Application Designer

PDF and FOP Services

Version 8.2 (2013-03-18)

March 2013

Application Designer

This document applies to Application Designer Version 8.2 (2013-03-18).

Specifications contained herein are subject to change and these changes will be reported in subsequent release notes or new editions.

Document ID: CIT-SERVICES-82-20130318

Table of Contents

Preface v
I 1
1 Introduction 3
FOP - Formatting Objects Project 4
Output Formats 6
Printing 6
Typical Example 6
Browser Output of PDF Documents 8
2 The First PDF Document 11
Overview of Required Steps 12
Creating the XML Form 14
Creating the Java Class 16
Calling the APIs at Runtime 20
Integrating the PDF into an Application Designer Page 21
II Page Layout Controls 25
3 Typical Page Layout 27
4 CISFO:FOPPAGE2 29
Properties 30
5 CISFO:HEADER2 33
Properties 34
6 CISFO:FOOTER2 41
Properties 42
7 CISFO:PAGENUMBER2 49
Properties 50
8 CISFO:BODY2 51
Properties 52
III Container Controls 57
9 General Information 59
10 CISFO:ROWTABLE2 61
Properties 62
11 CISFO:COLTABLE2 69
Properties 94
IV Distance Controls 77
12 CISFO:VDIST2 79
Properties 80
13 CISFO:HDIST2 81
Properties 82
14 CISFO:HLINE2 87
Properties 88
V Text Output Controls 89
15 General Information 91
16 CISFO:ROWTEXTBLOCK2 93
Properties 94

17 CISFO:CELLTEXT2 101
Properties 102
18 CISFO:TEXT2 109
Properties 110
19 CISFO:TEXTBLOCK2 115
Properties 116
20 CISFO:TEXTREPLACE2 123
Properties 124
21 CISFO:REPLACE2 129
Properties 130
VI Other Controls 137
22 Lists of Data 139
23 Style Control 145
Properties 146
24 Image Control 153
Properties 154
25 Barcode Control 155
Properties 156
26 New Page Control 159
Properties 160
27 Controls that you Know from Application Designer HTML 161
VII Printing 163
28 Printing 165

Preface

The PDF standard (portable document format, defined by Adobe) is a commonly accepted way for producing high-quality documents for printable output. By the usage of plug-in modules (Adobe Reader), the output can be displayed and printed in any browser.

Creating PDF is a standard issue when providing user interfaces through the browser. This is the reason why Application Designer introduced a couple of PDF services. The Application Designer ideas which are applied in the area of HTML-based GUIs for applications are transferred into the generation of PDF output:

- High-quality output.
- Simple usage but still making control possible in a fine granular way on low level.
- Simple binding of PDF output to application data objects (adapters).
- XML as base for defining templates.

This documentation is organized under the following headings:

Introduction	General information on Application Designer's FOP (Formatting Objects Processor) services.
The First PDF Document	Explains the steps for creating a PDF document that is rendered inside the browser.
Page Layout Controls	Describes the controls used for the layout of a print form.
Container Controls	Describes the controls for table definitions.
Distance Controls	Describes the controls for keeping the distance in a table.
Text Output Controls	Describes the controls for the output of text.
Other Controls	Describes the controls for lists of data, and for style, image, barcode and new pages.
Printing	Explains how to produce paper printouts.

Note: An Application Designer form contains an arrangement of controls. The controls are represented by XML tags with corresponding properties. Each control tag and each tag property is described in this documentation. The descriptions can also be reached by accessing the control wizards inside the Layout Painter.

I

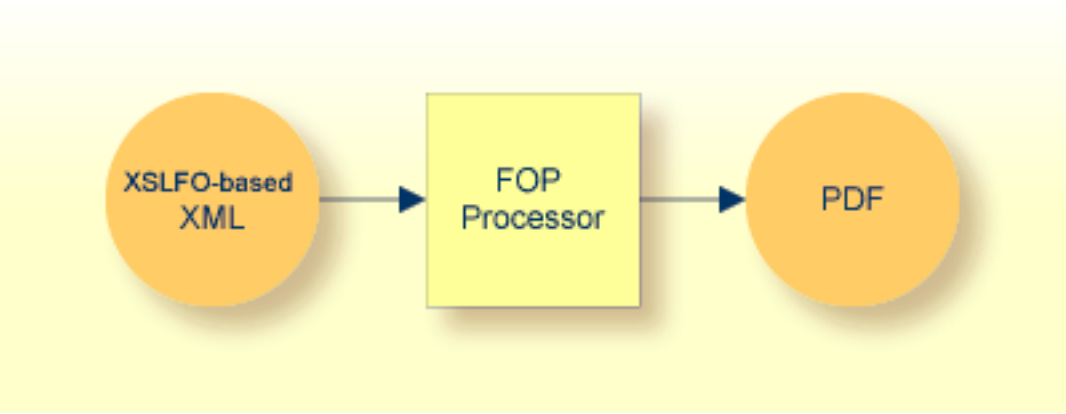
- 1 Introduction 3
- 2 The First PDF Document 11

1 Introduction

- FOP - Formatting Objects Project 4
- Output Formats 6
- Printing 6
- Typical Example 6
- Browser Output of PDF Documents 8

FOP - Formatting Objects Project

There is a series of accepted standard modules coming from the Apache Group that form the base of Application Designer's PDF services. The so-called FOP standard (Formatting Objects Processor) is - though still in evolution - a healthy base for generating PDF documents through a certain XML-based specification: XSLFO. XSLFO consists out of a number of XML tag definitions by which a document is described. The XML definition is processed by a transform processor that produces PDF as output.

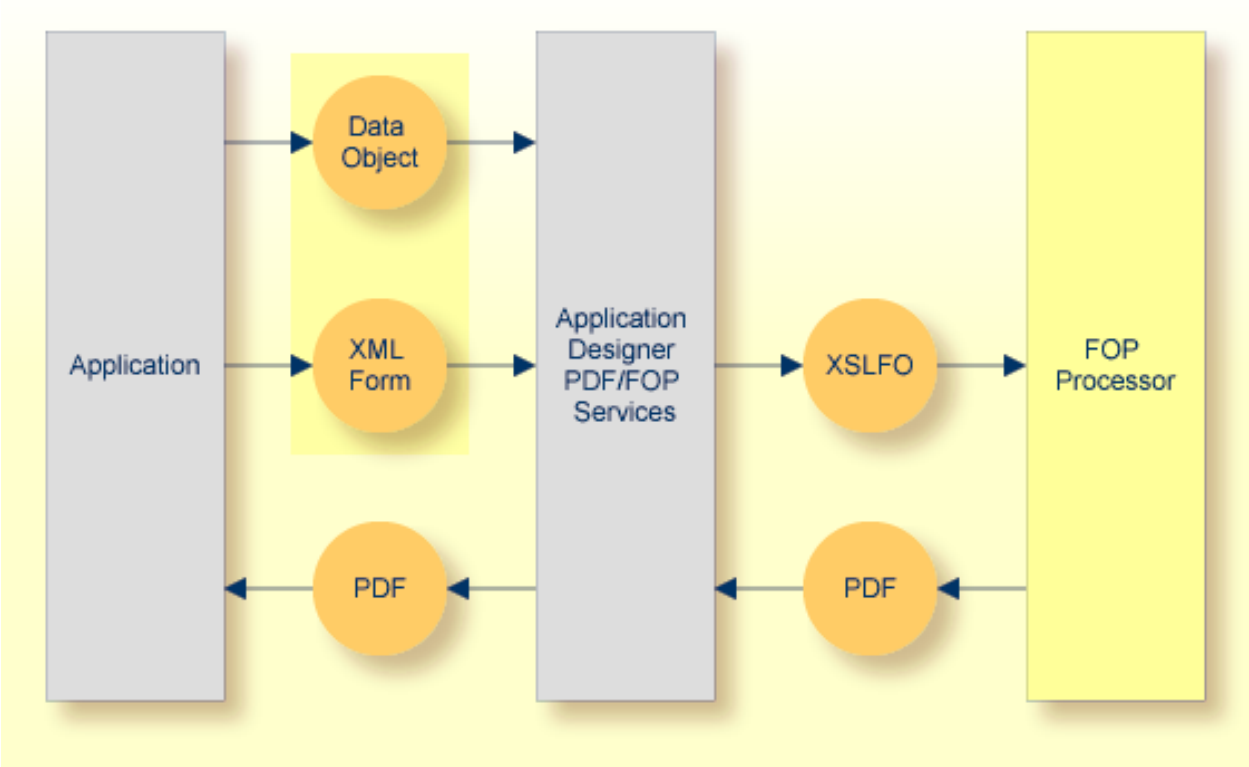


Though clear in concept and flexible in usage, XSLFO has a certain complexity: the number of tags and the number of properties inside the tags is quite high. Therefore, the effort for writing XSLFO-based XML documents is high and requires a profound understanding of FOP.

In addition, XSLFO is a pure output protocol - it describes a document before being transferred to PDF. The task of building the document and filling the right data in is not covered by FOP. This is where Application Designer's services come in.

Application Designer Services in Front of FOP

The following image summarizes the architecture of Application Designer's FOP services:



An application passes an XML form and a data object to the PDF/FOP services.

The XML form contains:

- Special abbreviation tags provided by Application Designer that simplify the creation of standard documents.
- Special tags provided by Application Designer that represent certain output controls (such as a grid) and contain binding information to a data object.
- Any XSLFO tag that you require. This means, you can directly use XSLFO statements and embed them into the form definition.

The data object (adapter) contains:

- All the “net data” that is mixed into the form. Net data is provided in the same way as you may be used to from working with Application Designer HTML pages: by corresponding properties that are referenced from the XML form definition.

The Application Designer PDF/FOP services build proper XSLFO out of the form and the data object. The XSLFO is passed to the normal FOP processing. The result is a PDF stream that is passed back to the calling application.

Output Formats

FOP is an abstract document definition. It is not bound to PDF. Application Designer offers both interfaces to create FOP-XSLO documents by the use of forms and it offers interfaces for printing FOP to a printer.

You can use other FOP output formats as well. Application Designer offers to you to directly generate the XSLFO result and pass it back to you as string.

Printing

Printing of application forms can be done in two ways:

- Server side PDF creation.
- Server side printing.

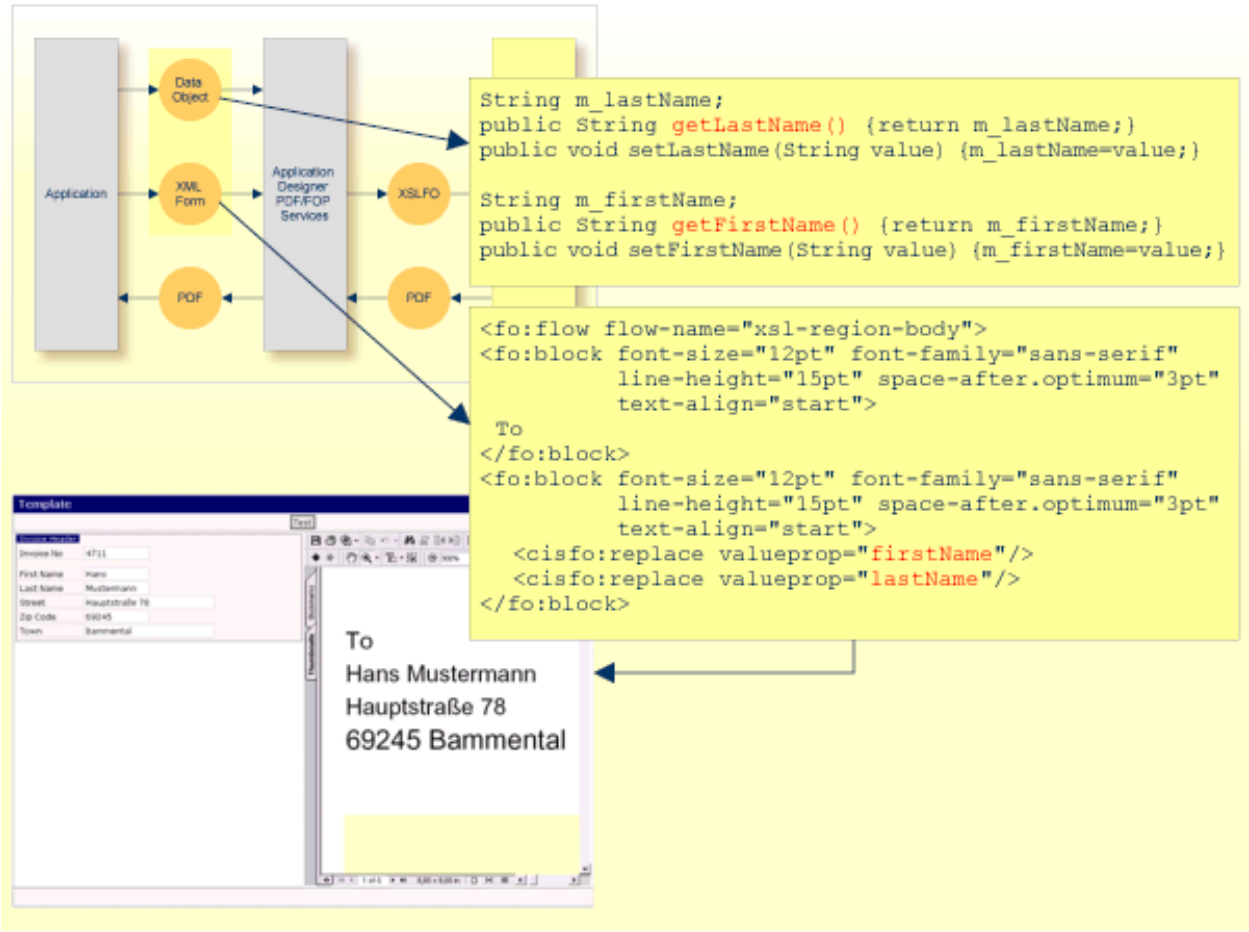
The server side PDF creation is what this document is about. Your server program uses the Application Designer APIs for creating PDF via FOP. The PDF files can be made available for the client (i.e. reachable by URL) and the client can integrate the files into its pages. Typically, the Adobe Reader is started on the client and the user can decide whether to print the document.

The server side printing is available by using a corresponding API. You can create an FOP document and call the print API. You have to define the printer on which you want to print.

Note: Server side printing requires JDK 1.4 to be used inside your application server. (JDK 1.3. only allows to print on the standard printer of your application server.)

Typical Example

The following image shows an example of how the framework is applied:



You see that the binding between the form's XML and the data object is the same as you know from the binding between Application Designer HTML pages and adapter objects. In fact, it is the same technology now transferred to PDF output scenarios.

The following topics are covered below:

- Data Object (Adapter)
- Ways of Usage

Data Object (Adapter)

At runtime, the data object provides the data that is filled into the form. The same rules that you already know from the Application Designer adapter objects also apply to the data object:

- Properties can be nested: `address.street` means that first `address` is accessed, then `street` within the address object.
- Properties can be implemented - at any level - by corresponding `set/get` methods and/or by implementing the interface `IDynamicAccess`.

For detailed information, see *Binding between Page and Adapter* in the *Special Development Topics*.

Ways of Usage

There are several ways of using the Application Designer PDF/FOP services.

- You may store the XML forms somewhere in the database or in the file system. By this you can, for example, allow your customer to set up own forms that bind to the same data objects but contain different rendering information (e.g. different logos, different order of columns).
- You may generate the XML form dynamically. If having a flexible grid of information, then the number of columns depends, for example, on certain parameters.

It is up to you to decide.

Browser Output of PDF Documents

After having created a PDF document, you typically want to output it somewhere in the browser. There are two questions that you have to consider:

- How do you store the generated PDF document in your server so that it is reachable by URL?
- How do you embed the document into Application Designer HTML pages?

The following topics are covered below:

- Storing the Generated PDF
- Embedding the PDF into your Application Designer Page

Storing the Generated PDF

The Application Designer PDF APIs will generate a PDF byte stream. You have to store this byte stream in a certain way so that it can be reached by a URL from the browser. There are several ways to do so:

- You can store the document in the file system of the application server, for example, somewhere inside the file system belonging to your web application. However, this is a “quick and dirty” way and only works properly for some application servers (e.g. Tomcat, Jetty, etc.): it assumes that there is a file system in which your application is stored. Some application servers do not store their web applications in the file system at all, but hold them within databases.
- You can store the data in a way that it is not reachable by a “fixed URL to a file” but by a “soft, servlet-based” URL.

If using the second option and embedding PDF into Application Designer page processing, then Application Designer already provides a so-called “session buffer”. Inside the session buffer, you can store any object under a certain name. It will return a URL that you can pass to the browser as the URL to the stored content.

An example is provided in the section *The First PDF Document*.

Embedding the PDF into your Application Designer Page

- You can define a SUBPAGE control. A SUBPAGE control is a frame that can be dynamically loaded with a URL. This means you can store the PDF result somewhere in the file system so that it is accessible via a URL. The URL is passed to the SUBPAGE control - and the PDF is shown inside the frame accordingly.
- You can define an own frame in your frameset arrangement and send the URL of the PDF document to this frame.

An example is provided in the section *The First PDF Document*.

2 The First PDF Document

- Overview of Required Steps 12
- Creating the XML Form 14
- Creating the Java Class 16
- Calling the APIs at Runtime 20
- Integrating the PDF into an Application Designer Page 21

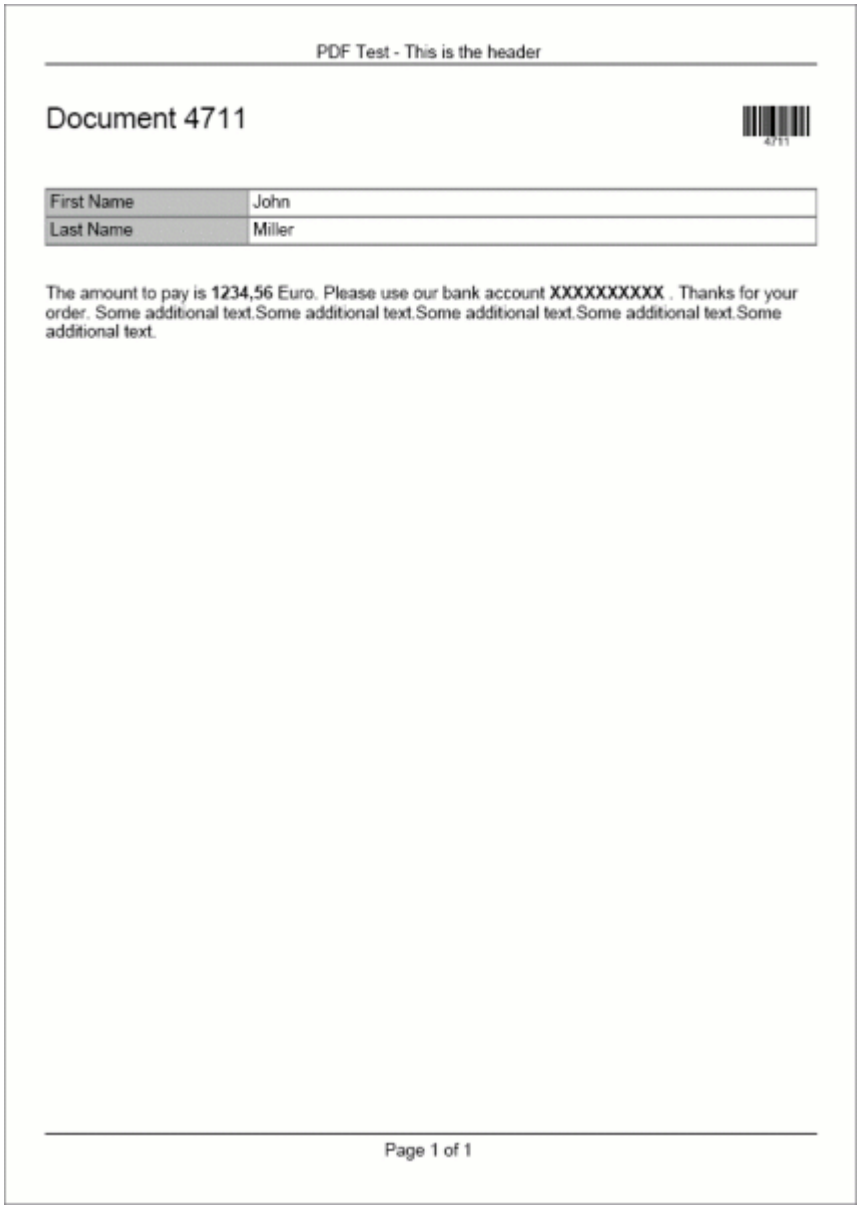
This chapter provides a brief tutorial in which you get to know all important steps for creating a PDF document that is rendered inside the browser. The tutorial describes how to do this using the development workplace. It is required that you have a basic understanding of Application Designer.

Overview of Required Steps

The following steps are required to create a PDF document:

1. Create the XML form.
2. Create the Java class that provides the data objects for the XML form.
3. Call the APIs to generate PDF.
4. Integrate the PDF into an Application Designer page.

The result of this tutorial will be the following document:



The document contains the following:

- Header, footer, body.
- Differently formatted texts.
- Barcode.
- Table of data.
- Some flat text containing dynamic data.

Creating the XML Form

Start the Application Designer development workplace (***http://localhost:51000/cis/HTML-BasedGUI/workplace/ide.html***) and open a project in the navigation frame (this tutorial uses the project "cisyourfirstproject").

Create a new layout. In the resulting dialog, enter a name (for example, "pdftest.xml") for the XML form definition and select the layout template for the PDF output. A new layout will be opened and preloaded with template information.

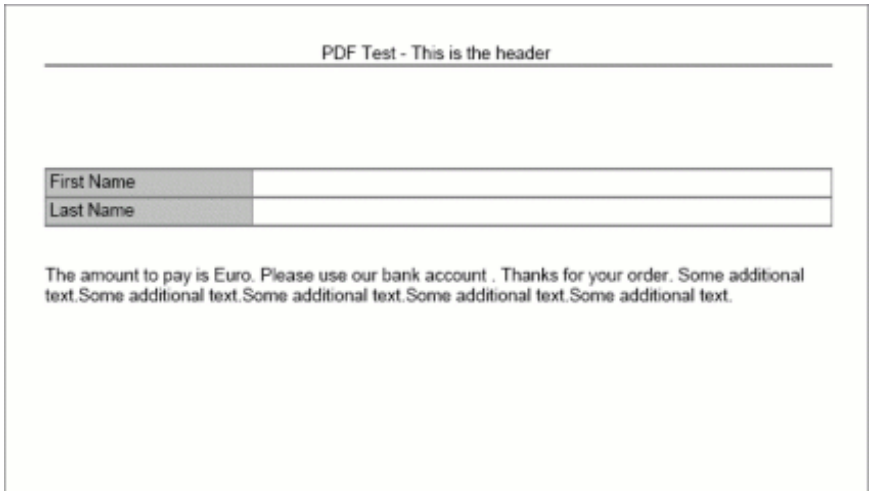
The Layout Painter is now used in the same way as you normally use it for creating Application Designer HTML pages. When you preview the layout in the Layout Painter, Adobe Reader is opened in the preview area, displaying the current XML form definition.

By adding controls into the template's control structure, you can modify the rendering result accordingly. Use the editor functions to create the following XML form definition:

```
<cisfo:foppage2 objectclass="PDFDataObject" pageheight="29.7cm" pagewidth="21cm" ↩
margintop="1cm" marginbottom="0.5cm" marginleft="1cm" marginright="1cm" ↩
headerheight="1.5cm" footerheight="1.3cm">
    <cisfo:styleclasses2>
        <cisfo:style2 stylename="THICK" fontsize="20pt" fontstyle="normal">
        </cisfo:style2>
        <cisfo:style2 stylename="LABEL" backgroundcolor="#C0C0C0" ↩
bordercolor="#808080" borderstyle="solid" borderwidth="0.1mm" padding="1mm">
        </cisfo:style2>
        <cisfo:style2 stylename="DEFAULTOUT" bordercolor="#808080" ↩
borderstyle="solid" borderwidth="0.1mm" padding="1mm">
        </cisfo:style2>
    </cisfo:styleclasses2>
    <cisfo:header2>
       <cisfo:rowtextblock2 text="PDF Test - This is the header" textalign="center">
        </cisfo:rowtextblock2>
        <cisfo:hline2>
        </cisfo:hline2>
    </cisfo:header2>
    <cisfo:footer2>
        <cisfo:hline2>
        </cisfo:hline2>
        <cisfo:vdist2 height="2mm">
        </cisfo:vdist2>
        <cisfo:rowtextblock2 text="Page" textalign="center">
            <cisfo:pagenumber2 separator=" of ">
            </cisfo:pagenumber2>
        </cisfo:rowtextblock2>
    </cisfo:footer2>
    <cisfo:body2>
        <cisfo:rowtable2 columnwidths="17cm; 2cm">
```

```
            <cisfo:row2>
                <cisfo:replace2 valueprop="docNumberText" stylename="THICK">
                </cisfo:replace2>
                <cisfo:barcode2 valueprop="docNumber" barcodetype="CODE39" ↩
height="10">
                </cisfo:barcode2>
            </cisfo:row2>
        </cisfo:rowtable2>
        <cisfo:vdist2 height="10mm">
        </cisfo:vdist2>
        <cisfo:rowtable2 columnwidths="5cm; 14cm">
            <cisfo:row2>
                <cisfo:celltext2 text="First Name" stylename="LABEL">
                </cisfo:celltext2>
                <cisfo:replace2 valueprop="firstName" stylename="DEFAULTOUT">
                </cisfo:replace2>
            </cisfo:row2>
            <cisfo:row2>
                <cisfo:celltext2 text="Last Name" stylename="LABEL">
                </cisfo:celltext2>
                <cisfo:replace2 valueprop="lastName" stylename="DEFAULTOUT">
                </cisfo:replace2>
            </cisfo:row2>
        </cisfo:rowtable2>
        <cisfo:vdist2 height="10mm">
        </cisfo:vdist2>
        <cisfo:rowtextblock2>
            <cisfo:text2 text="The amount to pay is ">
            </cisfo:text2>
            <cisfo:textreplace2 valueprop="amount" fontweight="bold">
            </cisfo:textreplace2>
            <cisfo:text2 text=" Euro. Please use our bank account ">
            </cisfo:text2>
            <cisfo:textreplace2 valueprop="account" fontweight="bold">
            </cisfo:textreplace2>
           <cisfo:text2 text=". Thanks for your order. Some additional text.Some ↩
additional text.Some additional text.Some additional text.Some additional text.">
            </cisfo:text2>
        </cisfo:rowtextblock2>
    </cisfo:body2>
</cisfo:foppage2>
```

After having entered all this (you can also cut and paste the XML from this documentation into your XML document) and when previewing the page, the top of the page should look as follows:

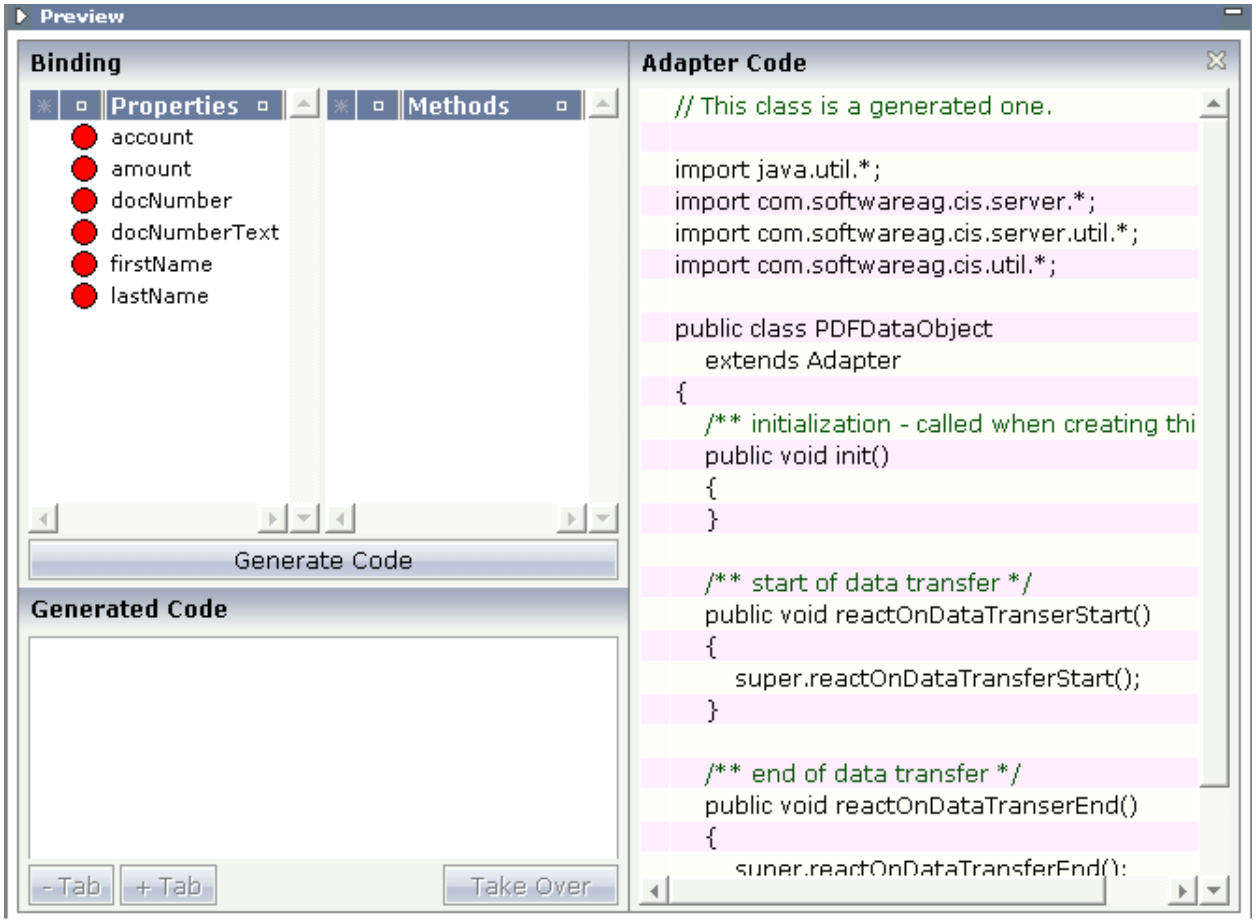


The document structure is already visible, but some data is still missing - the data that is provided by a corresponding data object.

Creating the Java Class

In the first tag of the XML form (CISFO:FOPPAGE2) there is a property `objectclass`. This is the class that provides the data that will be mixed into the document. In the example, the name of this class is `PDFDataObject`.

In order to create the class, you may use the Code Assistant which is part of the Layout Painter:



You should create some Java code that looks as follows:

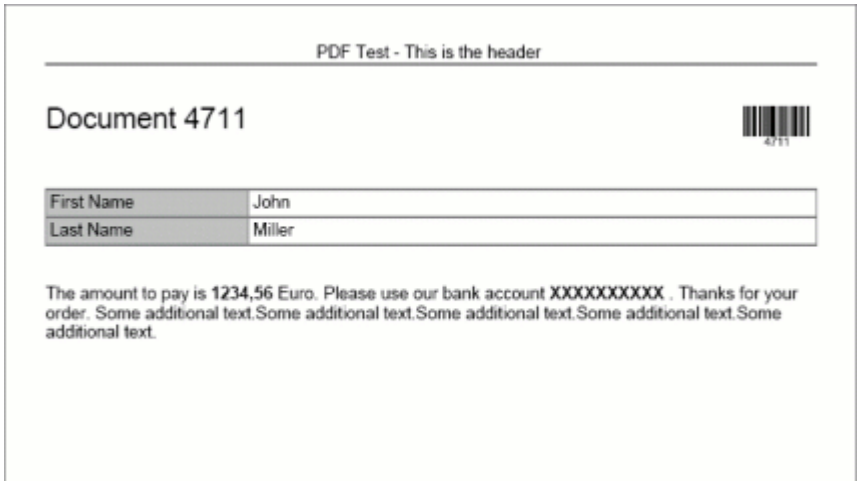
```
import java.util.*;

import com.softwareag.cis.context.SessionContext;
import com.softwareag.cis.print.fop.*;
import com.softwareag.cis.server.util.*;
import com.softwareag.cis.util.*;

// class for PDF creation with CISFO2 tags

public class PDFDataObject
    implements IFOPDataObject
{
    // -----------------------------------------------------------------------
    // constructor
    // -----------------------------------------------------------------------

    // default constructor
    public PDFDataObject()
    {
    }
```

```
// Called if the corresponding print form is previewed within the Layout Painter
public PDFDataObject(PDFFOPPreviewOnly previewObject)
{
    m_docNumber = "4711";
    m_firstName = "John";
    m_lastName = "Miller";
    m_account = "XXXXXXXXXX";
    m_amount = "1234,56";
}

// ------------------------------------------------------------------------
// property access
// ------------------------------------------------------------------------

// property >account<
String m_account;
public String getAccount() { return m_account; }
public void setAccount(String value) { m_account = value; }

// property >amount<
String m_amount;
public String getAmount() { return m_amount; }
public void setAmount(String value) { m_amount = value; }

// property >firstName<
String m_firstName;
public String getFirstName() { return m_firstName; }
public void setFirstName(String value) { m_firstName = value; }

// property >lastName<
String m_lastName;
public String getLastName() { return m_lastName; }
public void setLastName(String value) { m_lastName = value; }

// property >docNumberText<
public String getDocNumberText() { return "Document " + m_docNumber; }

// property >docNumber<
String m_docNumber;
public String getDocNumber() { return m_docNumber; }
public void setDocNumber(String value) { m_docNumber = value; }

// ------------------------------------------------------------------------
// Internationalization for this data object
// ------------------------------------------------------------------------

String m_dateFormat = SessionContext.DATE_MMSDDSYYYY;
public String getDateFormat() { return m_dateFormat; }

String m_dayTimeFormat = SessionContext.TIME_HHCMMCSS;
public String getDayTimeFormat() { return m_dayTimeFormat; }
```

```
    String m_decimalSeparator = SessionContext.DEC_SEPARATOR_POINT;
    public String getDecimalSeparator() { return m_decimalSeparator; }

    String m_language = "E";
    public String getLanguage() { return m_language; }

    String m_timeStampFormat = SessionContext.DATE_MMSDDSYYYY;
    public String getTimeStampFormat() { return m_timeStampFormat; }


}
```

Compile this code so that the generated class is available in the directory *…/cisyourfirstproject/app-classes/classes*.

When now previewing the page, the top of the page should look as follows:



Let us have a closer look at the code:

- It implements an interface `com.softwareag.cis.print.fop.IFOPDataObject`. This interface provides some methods that the Application Designer PDF/FOP framework requires in order to provide internationalisation support.
- It provides the properties that are required by the page.
- It provides a special constructor by which the properties are filled with some dummy data.

The special constructor is used by Application Designer when previewing the form in the Layout Manager. (Later on you will - as part of your programs - fill the data object inside your application and pass it to the PDF/FOP processing together with the form.)

Calling the APIs at Runtime

In the previous steps you created:

- an XML form (stored in an XML file within a certain Application Designer project),
- a Java class representing the net data behind the XML form.

Creating a PDF file now is quite simple: you just have to load the XML form and create an instance of the data object, and pass both to the PDF/FOP processor framework. The code might look like:

```
String xmlForm = WebResourceReader.readTextFileIntoStringWithLinebreak
      (findServletContext(),"/cisyourfirstproject/xml/pdftest.xml");
 PDFDataObject pdo = new PDFDataObject();
 pdo.setDocNumber("4711");
 pdo.setAccount("1234567890");
 pdo.setFirstName(m_firstName);
 pdo.setLastName(m_lastName);
 pdo.setAmount(m_amount);
 IPDFFOPService pdffop = PDFFOPServiceFactory.createPDFFOPService(xmlForm,pdo);
 byte[] pdf = pdffop.generatePDF();
```

In the code, the XML file is read by using a `WebResourceReader` class which is part of the Application Designer runtime environment. This class uses the servlet context's mechanism to access files that are part of your web application. As a consequence, the file is accessed in such a way that is compatible to any cluster structures of the application server you might use. Of course, you can also read the XML form in any other way. Maybe you store your forms inside a database and read the data from there.

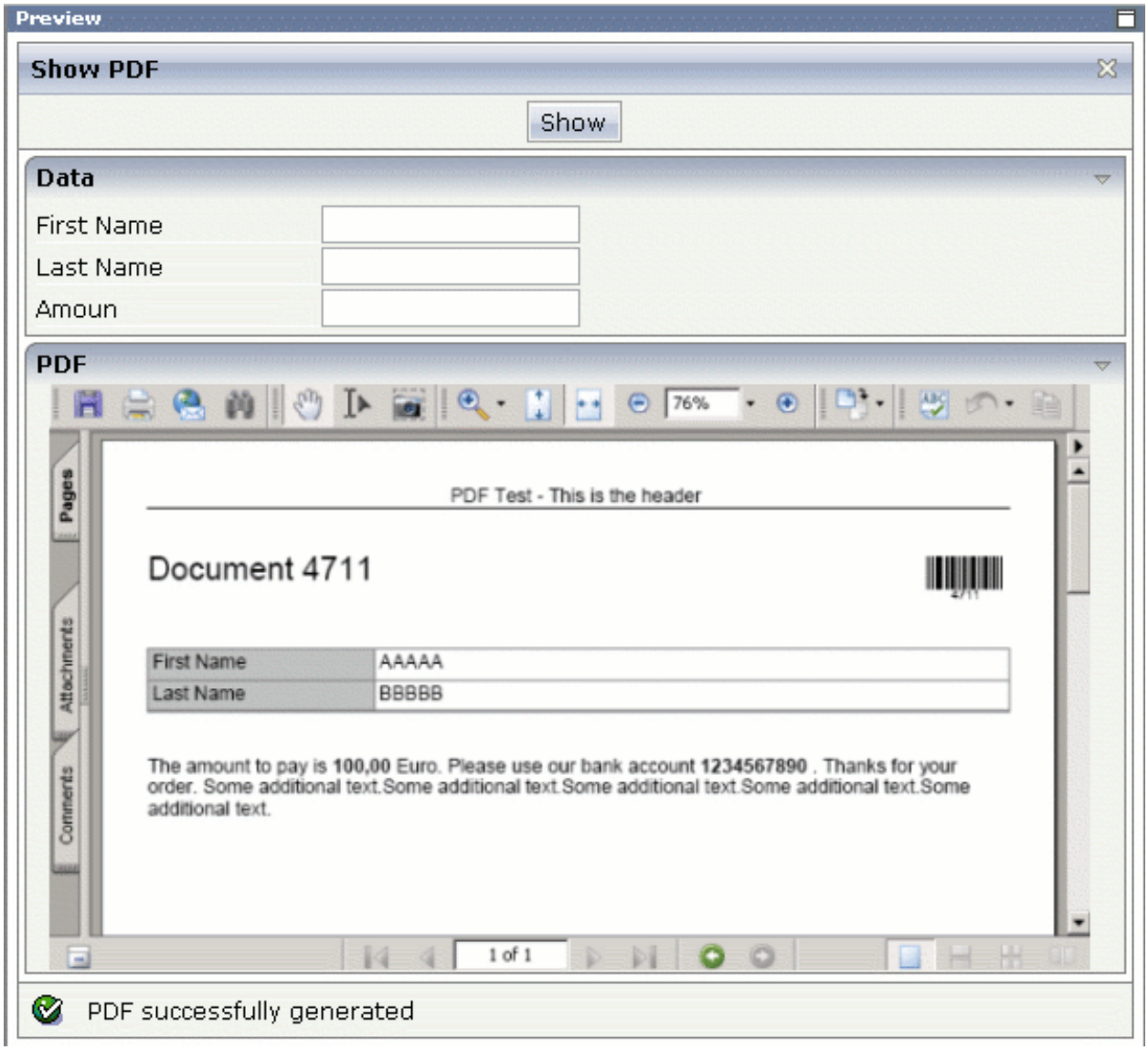
The data object is simply created and filled with certain data. Part of the data are passed as fixed literals, other parts are dynamically passed by corresponding member variables.

The PDF processor is created by calling the `PDFFOPServiceFactory` class and requesting an instance. Finally, the instance is invoked and PDF is passed back as a byte array.

It is important to note that in principle the creation of PDF is completely independent from the normal GUI adapter processing inside the Application Designer; i.e. you can use it within your application at any point - not only inside Application Designer adapter objects that represent Application Designer HTML pages.

Integrating the PDF into an Application Designer Page

Now, let us build an Application Designer HTML page that looks as follows:



You can enter some parameters and choose the **Show** button. As a result, the corresponding PDF page will be displayed as part of the page.

The layout of the page looks as follows:

```
<page model="ShowPDFAdapter">
    <titlebar name="Show PDF">
    </titlebar>
    <header withdistance="false">
        <button name="Show" method="onShow">
        </button>
    </header>
    <pagebody takefullheight="true">
        <rowarea name="Data">
            <itr>
                <label name="First Name" width="120">
                </label>
                <field valueprop="firstName" width="200">
                </field>
            </itr>
            <itr>
                <label name="Last Name" width="120">
                </label>
                <field valueprop="lastName" width="200">
                </field>
            </itr>
            <itr>
                <label name="Amount" width="120">
                </label>
                <field valueprop="amount" width="200">
                </field>
            </itr>
        </rowarea>
        <rowarea name="PDF" height="100%">
            <itr width="100%" height="100%" fixlayout="true">
                <subpage valueprop="pdfURL" height="100%" width="100%">
                </subpage>
            </itr>
        </rowarea>
        <vdist>
        </vdist>
    </pagebody>
    <statusbar withdistance="false">
    </statusbar>
</page>
```

There are three fields to do the input and there is one SUBPAGE control which renders a URL that is dynamically passed by the property `pdfURL`.

Let us have a look at the corresponding adapter class code:

```
// This class is a generated one.

import java.util.*;

import com.softwareag.cis.print.fop.IPDFFOPService;
import com.softwareag.cis.print.fop.PDFFOPServiceFactory;
import com.softwareag.cis.server.*;
import com.softwareag.cis.server.util.*;
import com.softwareag.cis.util.*;

public class ShowPDFAdapter
    extends Model
{
    // -----------------------------------------------------------------------
    // property access
    // -----------------------------------------------------------------------

    // property >amount<
    String m_amount;
    public String getAmount() { return m_amount; }
    public void setAmount(String value) { m_amount = value; }

    // property >firstName<
    String m_firstName;
    public String getFirstName() { return m_firstName; }
    public void setFirstName(String value) { m_firstName = value; }

    // property >lastName<
    String m_lastName;
    public String getLastName() { return m_lastName; }
    public void setLastName(String value) { m_lastName = value; }

    // property >pdfURL<
    String m_pdfURL = "about:blank";
    public String getPdfURL() { return m_pdfURL; }
    public void setPdfURL(String value) { m_pdfURL = value; }

    // -----------------------------------------------------------------------
    // public adapter methods
    // -----------------------------------------------------------------------

    public void onShow()
    {
     try
 {
      // build pdf
  String xmlForm = WebResourceReader.readTextFileIntoStringWithLinebreak
                   (findServletContext(),"/cisyourfirstproject/xml/pdftest.xml");
  PDFDataObject pdo = new PDFDataObject();
  pdo.setDocNumber("4711");
  pdo.setAccount("1234567890");
  pdo.setFirstName(m_firstName);
```

```
  pdo.setLastName(m_lastName);
  pdo.setAmount(m_amount);
  IPDFFOPService pdffop = PDFFOPServiceFactory.createPDFFOPService(xmlForm,pdo);
  byte[] pdf = pdffop.generatePDF();
  // build dynamic URL
  m_pdfURL = findCISSessionContext().getSessionBuffer().addPDF("test" + (new ↩
Date()).getTime(),pdf);
  outputMessage(MT_SUCCESS,"PDF successfully generated");
 }
     catch (Throwable e)
 {
      outputMessage(MT_ERROR,e.toString());
 }
    }
}
```

The critical method is the `onShow` method that is invoked when the user chooses the **Show** button. In the method, the API is called and a PDF byte array is created as mentioned in the previous section.

The result is not stored in the file system but is passed to a so-called “session buffer” that is made available by Application Designer. The session buffer is a transient store for temporary data that is made available to the browser via a URL. The data you pass to the session store is passed by using an API; in the API, you specify a certain name for the data. The URL that is passed back to the application can be used inside the browser to access the transient data. Internally, the URL is a link to certain servlet provided by Application Designer that accesses the transient data.

As a consequence, the PDF bytes are not stored in the file system in order to be displayed inside the browser. This particularly means that your application is able to be operated in a clustered environment.

Note: Of course, there may be different scenarios in your environment: maybe you have one central file server which also serves as a web server, and you park all your PDF documents on this server. In this case, you do not need the session buffer.

The session buffer is automatically removed when the user's session ends. If you have long-lasting user sessions, make sure that the information in the session buffer is removed when it is not required anymore. Removing is done by using the session buffer's Java API.

II Page Layout Controls

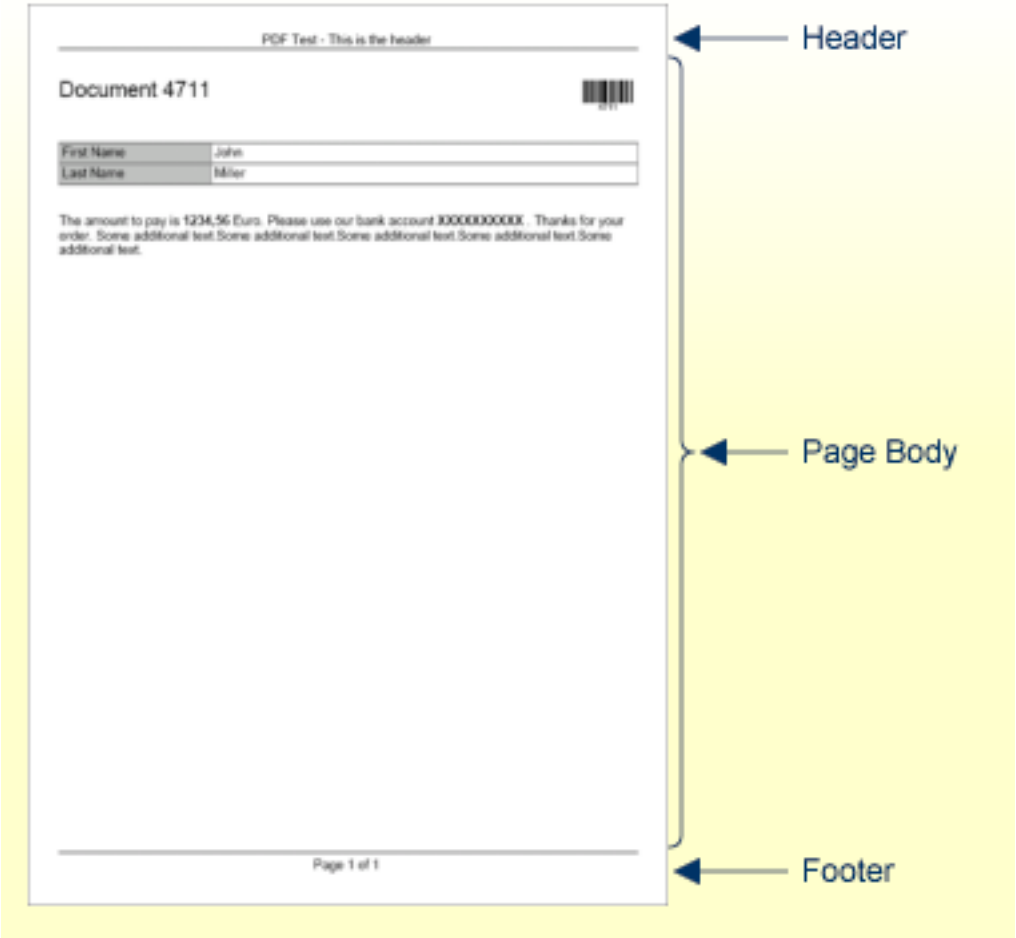
The information provided in this part is organized under the following headings:

Typical Page Layout
CISFO:FOPPAGE2
CISFO:HEADER2
CISFO:FOOTER2
CISFO:PAGENUMBER2
CISFO:BODY2

3 Typical Page Layout

The layout of a print form typically contains the following elements:

- page
- header
- footer
- body



Header, footer and body open container areas that may hold any other controls inside.

4 CISFO:FOPPAGE2

- Properties 30

The CISFO:FOPPAGE2 control is always the top node of a layout definition. The most important definition of the PAGE control is the definition of the paper size. Set the height and width e.g. to 29.7cm/21cm to use the German DIN A4 paper size. Like with the PAGE control in the HTML part, the FOP page is bound to a Java class. The Java class is the counterpart of the print form.

Properties

Basis			
objectclass	Name of the class which is the adapter class for the document. The class represents the logical counter part of the document. While the document specifies the layout, the class (at runtime the object...) holds the content data that the document binds to.	Sometimes obligatory	
project	The name of the CIS project for the document. It is needed for the multi language management in order to find the language dependent translation files.	Sometimes obligatory	
translationreference	The name of the document without extensions, which is needed for translation purposes.	Sometimes obligatory	
pageheight	Specifies the height of a page. Some valid values are 'auto', 'indefinite', '29cm' or '12in'. The width of the page is determined from the size of the laid-out content. 'pagewidth' and 'pageheight' properties may not both be set to 'indefinite'. Should that occur, the dimension that is parallel to the block-progression-direction, as determined by the 'referenceorientation' and 'writing-mode' on the fo:simple-page-master, of the page-reference-area will remain 'indefinite' and the other will revert to 'auto'.	Optional	21cm 29.7cm 8.5in 11in indefinite
pagewidth	Specifies the width of a page. Some valid values are 'auto', 'indefinite', '29cm' or '12in'. The width of the page is determined from the size of the laid-out content. 'pagewidth' and 'pageheight' properties may not both be set to 'indefinite'. Should that occur, the dimension that is parallel to the block-progression-direction, as determined by the 'referenceorientation' and 'writing-mode' on the fo:simple-page-master, of the page-reference-area will remain 'indefinite' and the other will revert to 'auto'.	Optional	21cm 29.7cm 8.5in 11in indefinite
headerheight	Height of the header area.	Optional	1pt 12pt 0.5in 1in 1mm

			5mm 1cm 1.5cm
footerheight	Height of the footer area.	Optional	1pt 12pt 0.5in 1in 1mm 5mm 1cm 1.5cm
columncount	When the COLUMNCOUNT is greater than one, then the region-body will be subdivided into multiple columns. If not set then the count ist 1.	Optional	1 2 3
columngap	Specifies the width of the separation between adjacent columns in a multi-column region. If a negative value has been specified a value of 0pt will be used.	Optional	1pt 5pt 1mm 5mm
Margin			
margintop	Top margin between page border and content.	Optional	1pt 5pt 1mm 5mm
marginbottom	Bottom margin between page border and content.	Optional	1pt 5pt 1mm 5mm

marginleft	Left margin between page border and content.	Optional	1pt 5pt 1mm 5mm
marginright	Right margin between page border and content.	Optional	1pt 5pt 1mm 5mm

5 CISFO:HEADER2

- Properties 34

Printouts typically have a header and/or footer to indicate the page content. You may use the header to output the document's title, the page number, the date of creation or other information. It opens a container area that holds any other controls inside.

Properties

Color			
backgroundcolor	This attribute sets the background color of an element, either a andlt;colorandgt; value or the keyword "transparent", to make the underlying colors shine through.	Optional	#000000 #808080 #FFFFFF #FF0000
color	Color in which the element is output.	Optional	#000000 #808080 #FFFFFF #FF0000
Border			
bordercolor	Color for all borders (bottom, top, left, right). Values are any valid color specification.	Optional	#000000 #808080 #FFFFFF #FF0000
borderstyle	This attribute specifies the line style of all box's border. Some valid values are "solid", "double" or "dashed".	Optional	solid none
borderwidth	The border width attribute specifies the width of all borders. Some valid values are "thin", "medium", "thick" or a concrete size (e.g. "2pt").	Optional	1pt 5pt 1mm 5mm
bordertopcolor	Color for the border on the top. Values are any valid color specification.	Optional	#000000 #808080

			#FFFFFF #FF0000
bordertopstyle	This attribute specifies the line style of a box's top border. Some valid values are "solid", "double" or "dashed".	Optional	solid none
bordertopwidth	The border width attribute specifies the width of the top border. Some valid values are "thin", "medium", "thick" or a concrete size (e.g. "2pt").	Optional	1pt 5pt 1mm 5mm
borderleftcolor	Color for the border on the left side. Values are any valid color specification.	Optional	#000000 #808080 #FFFFFF #FF0000
borderleftstyle	This attribute specifies the line style of a box's left border. Some valid values are "solid", "double" or "dashed".	Optional	solid none
borderleftwidth	The border width attribute specifies the width of the left border. Some valid values are "thin", "medium", "thick" or a concrete size (e.g. "2pt").	Optional	1pt 5pt 1mm 5mm
borderbottomcolor	Color for the border on the bottom. Values are any valid color specification.	Optional	#000000 #808080 #FFFFFF #FF0000
borderbottomstyle	This attribute specifies the line style of a box's bottom border. Valid values are 'solid', 'double' or 'dashed'.	Optional	solid none
borderbottomwidth	The border width attribute specifies the width of the bottom border. Valid values are "thin", "medium", "thick" or a concrete size (e.g. "2pt").	Optional	1pt 5pt

			1mm 5mm
borderrightcolor	Color for the border on the right side. Values are any valid color specification.	Optional	#000000 #808080 #FFFFFF #FF0000
borderrightstyle	This attribute specifies the line style of a box's right border. Some valid values are "solid", "double" or "dashed".	Optional	solid none
borderrightwidth	The border width attribute specifies the width of the right border. Some valid values are "thin", "medium", "thick" or a concrete size (e.g. "2pt").	Optional	1pt 5pt 1mm 5mm
Font			
fontfamily	This attribute specifies a priorized list of font family names and/or generic family names. To deal with the problem that a single font may not contain glyphs to display all the characters in a document, or that not all fonts are available on all systems, this property allows authors to specify a list of fonts, all of the same style and size, that are tried in sequence to see if they contain a glyph for a certain character. This list is called a "font set".	Optional	Helvetica Times Courier Symbol ZapfDingbats
fontsize	This attribute describes the size of the font when set as "solid". The font size corresponds to the em square, a concept used in typography. Note that certain glyphs may bleed outside their em squares. Some valid values are "xxsmall", "medium", "smaller" or a concrete size, e.g. "110%".	Optional	8pt 10pt 16pt
fontstyle	This attribute requests normal (sometimes referred to as "roman" or "upright"), italic, and oblique faces within a font family. Valid values are "normal", "italic" and "oblique".	Optional	normal italic oblique
fontvariant	In a small-caps font, the glyphs for lowercase letters look similar to the uppercase ones, but in a smaller size and with slightly different proportions. The "font-variant" property requests such a font for bicameral (having two cases, as with Roman script). This property has no visible effect for scripts	Optional	normal small-caps

	that are unicameral (having only one case, as with most of the world's writing systems). Valid values are "normal" and "small-caps".		
fontweight	This attribute specifies the weight of the font. Some valid values are 'normal', 'bold', 'bolder', 'lighter', '100' or '900', where 'normal' is the same as '400', 'bold' is the same as '700'.	Optional	normal bold 100 200 300 400 500 600 700 800 900
Margin			
margin	Sets the margin of a box. Negative values for margin properties are allowed, but there may be implementation-specific limits.	Optional	1pt 5pt 1mm 5mm
margintop	Sets the top margin of a box. Negative values for margin properties are allowed, but there may be implementation-specific limits.	Optional	1pt 5pt 1mm 5mm
marginleft	Sets the left margin of a box. Negative values for margin properties are allowed, but there may be implementation-specific limits.	Optional	1pt 5pt 1mm 5mm

marginbottom	Sets the bottom margin of a box. Negative values for margin properties are allowed, but there may be implementation-specific limits.	Optional	1pt 5pt 1mm 5mm
marginright	Sets the right margin of a box. Negative values for margin properties are allowed, but there may be implementation-specific limits.	Optional	1pt 5pt 1mm 5mm
Padding			
padding	Specifies the width of the padding of a block-area or inline-area. Unlike margin properties, values for padding properties cannot be negative.	Optional	1pt 5pt 1mm 5mm
paddingtop	Specifies the width of the padding on the top of a block-area or inline-area. Unlike margin properties, values for padding properties cannot be negative.	Optional	1pt 5pt 1mm 5mm
paddingleft	Specifies the width of the padding on the left of a block-area or inline-area. Unlike margin properties, values for padding properties cannot be negative.	Optional	1pt 5pt 1mm 5mm
paddingbottom	Specifies the width of the padding on the bottom of a block-area or inline-area. Unlike margin properties, values for padding properties cannot be negative.	Optional	1pt 5pt 1mm 5mm
paddingright	Specifies the width of the padding on the right of a block-area or inline-area. Unlike margin properties, values for padding properties cannot be negative.	Optional	1pt 5pt

			1mm 5mm

6 CISFO:FOOTER2

- Properties 42

Printouts typically have a header and/or footer to indicate the page content. You may use the footer to output the document's title, the page number and the date of creation. It opens a container area that holds any other controls inside.

Properties

Color			
backgroundcolor	This attribute sets the background color of an element, either a andlt;colorandgt; value or the keyword "transparent", to make the underlying colors shine through.	Optional	#000000 #808080 #FFFFFF #FF0000
color	Color in which the element is output.	Optional	#000000 #808080 #FFFFFF #FF0000
Border			
bordercolor	Color for all borders (bottom, top, left, right). Values are any valid color specification.	Optional	#000000 #808080 #FFFFFF #FF0000
borderstyle	This attribute specifies the line style of all box's border. Some valid values are "solid", "double" or "dashed".	Optional	solid none
borderwidth	The border width attribute specifies the width of all borders. Some valid values are "thin", "medium", "thick" or a concrete size (e.g. "2pt").	Optional	1pt 5pt 1mm 5mm
bordertopcolor	Color for the border on the top. Values are any valid color specification.	Optional	#000000 #808080

			#FFFFFF #FF0000
bordertopstyle	This attribute specifies the line style of a box's top border. Some valid values are "solid", "double" or "dashed".	Optional	solid none
bordertopwidth	The border width attribute specifies the width of the top border. Some valid values are "thin", "medium", "thick" or a concrete size (e.g. "2pt").	Optional	1pt 5pt 1mm 5mm
borderleftcolor	Color for the border on the left side. Values are any valid color specification.	Optional	#000000 #808080 #FFFFFF #FF0000
borderleftstyle	This attribute specifies the line style of a box's left border. Some valid values are "solid", "double" or "dashed".	Optional	solid none
borderleftwidth	The border width attribute specifies the width of the left border. Some valid values are "thin", "medium", "thick" or a concrete size (e.g. "2pt").	Optional	1pt 5pt 1mm 5mm
borderbottomcolor	Color for the border on the bottom. Values are any valid color specification.	Optional	#000000 #808080 #FFFFFF #FF0000
borderbottomstyle	This attribute specifies the line style of a box's bottom border. Valid values are 'solid', 'double' or 'dashed'.	Optional	solid none
borderbottomwidth	The border width attribute specifies the width of the bottom border. Valid values are "thin", "medium", "thick" or a concrete size (e.g. "2pt").	Optional	1pt 5pt

			1mm 5mm
borderrightcolor	Color for the border on the right side. Values are any valid color specification.	Optional	#000000 #808080 #FFFFFF #FF0000
borderrightstyle	This attribute specifies the line style of a box's right border. Some valid values are "solid", "double" or "dashed".	Optional	solid none
borderrightwidth	The border width attribute specifies the width of the right border. Some valid values are "thin", "medium", "thick" or a concrete size (e.g. "2pt").	Optional	1pt 5pt 1mm 5mm
Font			
fontfamily	This attribute specifies a priorized list of font family names and/or generic family names. To deal with the problem that a single font may not contain glyphs to display all the characters in a document, or that not all fonts are available on all systems, this property allows authors to specify a list of fonts, all of the same style and size, that are tried in sequence to see if they contain a glyph for a certain character. This list is called a "font set".	Optional	Helvetica Times Courier Symbol ZapfDingbats
fontsize	This attribute describes the size of the font when set as "solid". The font size corresponds to the em square, a concept used in typography. Note that certain glyphs may bleed outside their em squares. Some valid values are "xxsmall", "medium", "smaller" or a concrete size, e.g. "110%".	Optional	8pt 10pt 16pt
fontstyle	This attribute requests normal (sometimes referred to as "roman" or "upright"), italic, and oblique faces within a font family. Valid values are "normal", "italic" and "oblique".	Optional	normal italic oblique
fontvariant	In a small-caps font, the glyphs for lowercase letters look similar to the uppercase ones, but in a smaller size and with slightly different proportions. The "font-variant" property requests such a font for bicameral (having two cases, as with Roman script). This property has no visible effect for scripts	Optional	normal small-caps

	that are unicameral (having only one case, as with most of the world's writing systems). Valid values are "normal" and "small-caps".		
fontweight	This attribute specifies the weight of the font. Some valid values are 'normal', 'bold', 'bolder', 'lighter', '100' or '900', where 'normal' is the same as '400', 'bold' is the same as '700'.	Optional	normal bold 100 200 300 400 500 600 700 800 900
Margin			
margin	Sets the margin of a box. Negative values for margin properties are allowed, but there may be implementation-specific limits.	Optional	1pt 5pt 1mm 5mm
margintop	Sets the top margin of a box. Negative values for margin properties are allowed, but there may be implementation-specific limits.	Optional	1pt 5pt 1mm 5mm
marginleft	Sets the left margin of a box. Negative values for margin properties are allowed, but there may be implementation-specific limits.	Optional	1pt 5pt 1mm 5mm

marginbottom	Sets the bottom margin of a box. Negative values for margin properties are allowed, but there may be implementation-specific limits.	Optional	1pt 5pt 1mm 5mm
marginright	Sets the right margin of a box. Negative values for margin properties are allowed, but there may be implementation-specific limits.	Optional	1pt 5pt 1mm 5mm
Padding			
padding	Specifies the width of the padding of a block-area or inline-area. Unlike margin properties, values for padding properties cannot be negative.	Optional	1pt 5pt 1mm 5mm
paddingtop	Specifies the width of the padding on the top of a block-area or inline-area. Unlike margin properties, values for padding properties cannot be negative.	Optional	1pt 5pt 1mm 5mm
paddingleft	Specifies the width of the padding on the left of a block-area or inline-area. Unlike margin properties, values for padding properties cannot be negative.	Optional	1pt 5pt 1mm 5mm
paddingbottom	Specifies the width of the padding on the bottom of a block-area or inline-area. Unlike margin properties, values for padding properties cannot be negative.	Optional	1pt 5pt 1mm 5mm
paddingright	Specifies the width of the padding on the right of a block-area or inline-area. Unlike margin properties, values for padding properties cannot be negative.	Optional	1pt 5pt

			1mm 5mm

7 CISFO:PAGENUMBER2

- Properties 50

The CISFO:PAGENUMBER2 control represents the running number of the page within the PDF document. You typically place the page number within the header and/or footer of the document. By default, the control also shows the total number of pages. Use the property `withtotalpagenumber` to suppress the total page number.

Properties

Basic			
withtotalpagenumber	By default this control ouptuts - besides the running page number - the total page number. Use this attribute to control the output of the total page number. Default is TRUE.	Optional	true false
textid	Textid for the multi language management.	Optional	
separator	By default this control ouptuts - besides the running page number - the total page number. Use this attribute to separate the two figures. Example: a value of " / " outputs "2 / 16". Use attribute TEXTID in order to have language dependent separator ("2 of 16").	Optional	

8 CISFO:BODY2

- Properties 52

The page body is the main area where you place the body part of your layout. It opens a container area that holds any other controls inside.

Properties

Color			
color	Color in which the element is output.	Optional	#000000 #808080 #FFFFFF #FF0000
backgroundcolor	This attribute sets the background color of an element, either a andlt;colorandgt; value or the keyword "transparent", to make the underlying colors shine through.	Optional	#000000 #808080 #FFFFFF #FF0000
Border			
bordercolor	Color for all borders (bottom, top, left, right). Values are any valid color specification.	Optional	#000000 #808080 #FFFFFF #FF0000
borderstyle	This attribute specifies the line style of all box's border. Some valid values are "solid", "double" or "dashed".	Optional	solid none
borderwidth	The border width attribute specifies the width of all borders. Some valid values are "thin", "medium", "thick" or a concrete size (e.g. "2pt").	Optional	1pt 5pt 1mm 5mm
bordertopcolor	Color for the border on the top. Values are any valid color specification.	Optional	#000000 #808080 #FFFFFF

			#FF0000
bordertopstyle	This attribute specifies the line style of a box's top border. Some valid values are "solid", "double" or "dashed".	Optional	solid none
bordertopwidth	The border width attribute specifies the width of the top border. Some valid values are "thin", "medium", "thick" or a concrete size (e.g. "2pt").	Optional	1pt 5pt 1mm 5mm
borderleftcolor	Color for the border on the left side. Values are any valid color specification.	Optional	#000000 #808080 #FFFFFF #FF0000
borderleftstyle	This attribute specifies the line style of a box's left border. Some valid values are "solid", "double" or "dashed".	Optional	solid none
borderleftwidth	The border width attribute specifies the width of the left border. Some valid values are "thin", "medium", "thick" or a concrete size (e.g. "2pt").	Optional	1pt 5pt 1mm 5mm
borderbottomcolor	Color for the border on the bottom. Values are any valid color specification.	Optional	#000000 #808080 #FFFFFF #FF0000
borderbottomstyle	This attribute specifies the line style of a box's bottom border. Valid values are 'solid', 'double' or 'dashed'.	Optional	solid none
borderbottomwidth	The border width attribute specifies the width of the bottom border. Valid values are "thin", "medium", "thick" or a concrete size (e.g. "2pt").	Optional	1pt 5pt 1mm

			5mm
borderrightcolor	Color for the border on the right side. Values are any valid color specification.	Optional	#000000 #808080 #FFFFFF #FF0000
borderrightstyle	This attribute specifies the line style of a box's right border. Some valid values are "solid", "double" or "dashed".	Optional	solid none
borderrightwidth	The border width attribute specifies the width of the right border. Some valid values are "thin", "medium", "thick" or a concrete size (e.g. "2pt").	Optional	1pt 5pt 1mm 5mm
Font			
fontfamily	This attribute specifies a priorized list of font family names and/or generic family names. To deal with the problem that a single font may not contain glyphs to display all the characters in a document, or that not all fonts are available on all systems, this property allows authors to specify a list of fonts, all of the same style and size, that are tried in sequence to see if they contain a glyph for a certain character. This list is called a "font set".	Optional	Helvetica Times Courier Symbol ZapfDingbats
fontsize	This attribute describes the size of the font when set as "solid". The font size corresponds to the em square, a concept used in typography. Note that certain glyphs may bleed outside their em squares. Some valid values are "xxsmall", "medium", "smaller" or a concrete size, e.g. "110%".	Optional	8pt 10pt 16pt
fontstyle	This attribute requests normal (sometimes referred to as "roman" or "upright"), italic, and oblique faces within a font family. Valid values are "normal", "italic" and "oblique".	Optional	normal italic oblique
fontvariant	In a small-caps font, the glyphs for lowercase letters look similar to the uppercase ones, but in a smaller size and with slightly different proportions. The "font-variant" property requests such a font for bicameral (having two cases, as with Roman script). This property has no visible effect for scripts that are unicameral (having only one case, as with most of	Optional	normal small-caps

	the world's writing systems). Valid values are "normal" and "small-caps".		
fontweight	This attribute specifies the weight of the font. Some valid values are 'normal', 'bold', 'bolder', 'lighter', '100' or '900', where 'normal' is the same as '400', 'bold' is the same as '700'.	Optional	normal bold 100 200 300 400 500 600 700 800 900
Margin			
margin	Sets the margin of a box. Negative values for margin properties are allowed, but there may be implementation-specific limits.	Optional	1pt 5pt 1mm 5mm
margintop	Sets the top margin of a box. Negative values for margin properties are allowed, but there may be implementation-specific limits.	Optional	1pt 5pt 1mm 5mm
marginleft	Sets the left margin of a box. Negative values for margin properties are allowed, but there may be implementation-specific limits.	Optional	1pt 5pt 1mm 5mm
marginbottom	Sets the bottom margin of a box. Negative values for margin properties are allowed, but there may be implementation-specific limits.	Optional	1pt 5pt

			1mm 5mm
marginright	Sets the right margin of a box. Negative values for margin properties are allowed, but there may be implementation-specific limits.	Optional	1pt 5pt 1mm 5mm
Padding			
padding	Specifies the width of the padding of a block-area or inline-area. Unlike margin properties, values for padding properties cannot be negative.	Optional	1pt 5pt 1mm 5mm
paddingtop	Specifies the width of the padding on the top of a block-area or inline-area. Unlike margin properties, values for padding properties cannot be negative.	Optional	1pt 5pt 1mm 5mm
paddingleft	Specifies the width of the padding on the left of a block-area or inline-area. Unlike margin properties, values for padding properties cannot be negative.	Optional	1pt 5pt 1mm 5mm
paddingbottom	Specifies the width of the padding on the bottom of a block-area or inline-area. Unlike margin properties, values for padding properties cannot be negative.	Optional	1pt 5pt 1mm 5mm
paddingright	Specifies the width of the padding on the right of a block-area or inline-area. Unlike margin properties, values for padding properties cannot be negative.	Optional	1pt 5pt 1mm 5mm

III Container Controls

The information provided in this part is organized under the following headings:

General Information
CISFO:ROWTABLE2
CISFO:COLTABLE2

9 General Information

There are certain container controls:

- CISFO:ROWTABLE2
- CISFO:COLTABLE2

If you want to position certain information, then you have to open a container. A container is a table definition. Inside the container you place rows and inside the rows you place controls.

```
<cisfo:body2>
        <cisfo:rowtable2 columnwidths="17cm; 2cm">
            <cisfo:row2>
                <cisfo:replace2 valueprop="docNumberText" stylename="THICK">
                </cisfo:replace2>
                <cisfo:barcode2 valueprop="docNumber" barcodetype="CODE39" ↩
height="10">
                </cisfo:barcode2>
            </cisfo:row2>
            <cisfo:row2>
               ...
               ...
               ...
            </cisfo:row2>
            ...
            ...
            ...
        </cisfo:rowtable2>
       ...
       ...
       ...
```

The size of the columns is part of the container definition. While CISFO:ROWTABLE2 opens a container inside a row, CISFO:COLTABLE2 opens a container inside a column.

Note: The column size within the XML form definition is a fixed definition in centimeters, inches or other units of measurement. Unlike the HTML size definitions, it cannot be a percentage value.

A row inside a container can either be a “flexible row” that itself holds other controls

- CISFO:ROW2

or a row control with a predefined rendering. A row control spans all columns of the container definition. Available row controls are:

- CISFO:HLINE2 - a horizontal line
- CISFO:ROWTEXTBLOCK2 - a text output

Containers can be nested, i.e. inside a column of a container, you can position a CISFO:COLTABLE2 container. Or inside a row, you position a CISFO:ROWTABLE2 container.

10 CISFO:ROWTABLE2

- Properties 62

The CISFO:ROWTABLE2 control is a container. A container is a table definition. Inside the container, you place rows, and inside the rows, you place controls. It is placed into a table container (e.g. CISFO:BODY2) and spans one row.

Properties

Basis			
columnwidths	Semicolon separated list of widths of contained columns. Example: "1cm; 3cm; 4cm". Column definitions are defined as part of the table definition - all contained column controls you add later on as content of the table are sized according to this definition.	Obligatory	10cm 8cm;8cm 4cm;5cm;4cm;5cm
stylename	Name of a CISFO style. The style must exist within the form definition.	Optional	
backgroundcolor	This attribute sets the background color of an element, either a andlt;colorandgt; value or the keyword "transparent", to make the underlying colors shine through.	Optional	#000000 #808080 #FFFFFF #FF0000
color	Color in which the element is output.	Optional	#000000 #808080 #FFFFFF #FF0000
textalign	The alignment of the text. Valid values are 'center', 'left' or 'right'.	Optional	left center right
textdecoration	Different values for printing the text, so as underline, overline or line-through.	Optional	overline line-through underline
Border			

bordercolor	Color for all borders (bottom, top, left, right). Values are any valid color specification.	Optional	#000000 #808080 #FFFFFF #FF0000
borderstyle	This attribute specifies the line style of all box's border. Some valid values are "solid", "double" or "dashed".	Optional	solid none
borderwidth	The border width attribute specifies the width of all borders. Some valid values are "thin", "medium", "thick" or a concrete size (e.g. "2pt").	Optional	1pt 5pt 1mm 5mm
bordertopcolor	Color for the border on the top. Values are any valid color specification.	Optional	#000000 #808080 #FFFFFF #FF0000
bordertopstyle	This attribute specifies the line style of a box's top border. Some valid values are "solid", "double" or "dashed".	Optional	solid none
bordertopwidth	The border width attribute specifies the width of the top border. Some valid values are "thin", "medium", "thick" or a concrete size (e.g. "2pt").	Optional	1pt 5pt 1mm 5mm
borderleftcolor	Color for the border on the left side. Values are any valid color specification.	Optional	#000000 #808080 #FFFFFF #FF0000
borderleftstyle	This attribute specifies the line style of a box's left border. Some valid values are "solid", "double" or "dashed".	Optional	solid none

borderleftwidth	The border width attribute specifies the width of the left border. Some valid values are "thin", "medium", "thick" or a concrete size (e.g. "2pt").	Optional	1pt 5pt 1mm 5mm
borderbottomcolor	Color for the border on the bottom. Values are any valid color specification.	Optional	#000000 #808080 #FFFFFF #FF0000
borderbottomstyle	This attribute specifies the line style of a box's bottom border. Valid values are 'solid', 'double' or 'dashed'.	Optional	solid none
borderbottomwidth	The border width attribute specifies the width of the bottom border. Valid values are "thin", "medium", "thick" or a concrete size (e.g. "2pt").	Optional	1pt 5pt 1mm 5mm
borderrightcolor	Color for the border on the right side. Values are any valid color specification.	Optional	#000000 #808080 #FFFFFF #FF0000
borderrightstyle	This attribute specifies the line style of a box's right border. Some valid values are "solid", "double" or "dashed".	Optional	solid none
borderrightwidth	The border width attribute specifies the width of the right border. Some valid values are "thin", "medium", "thick" or a concrete size (e.g. "2pt").	Optional	1pt 5pt 1mm 5mm
Font			
fontfamily	This attribute specifies a priorized list of font family names and/or generic family names. To deal with the problem that a single font may not contain glyphs to	Optional	Helvetica Times

	display all the characters in a document, or that not all fonts are available on all systems, this property allows authors to specify a list of fonts, all of the same style and size, that are tried in sequence to see if they contain a glyph for a certain character. This list is called a "font set".		Courier Symbol ZapfDingbats
fontsize	This attribute describes the size of the font when set as "solid". The font size corresponds to the em square, a concept used in typography. Note that certain glyphs may bleed outside their em squares. Some valid values are "xxsmall", "medium", "smaller" or a concrete size, e.g. "110%".	Optional	8pt 10pt 16pt
fontstyle	This attribute requests normal (sometimes referred to as "roman" or "upright"), italic, and oblique faces within a font family. Valid values are "normal", "italic" and "oblique".	Optional	normal italic oblique
fontvariant	In a small-caps font, the glyphs for lowercase letters look similar to the uppercase ones, but in a smaller size and with slightly different proportions. The "font-variant" property requests such a font for bicameral (having two cases, as with Roman script). This property has no visible effect for scripts that are unicameral (having only one case, as with most of the world's writing systems). Valid values are "normal" and "small-caps".	Optional	normal small-caps
fontweight	This attribute specifies the weight of the font. Some valid values are 'normal', 'bold', 'bolder', 'lighter', '100' or '900', where 'normal' is the same as '400', 'bold' is the same as '700'.	Optional	normal bold 100 200 300 400 500 600 700 800 900
Margin			

margin	Sets the margin of a box. Negative values for margin properties are allowed, but there may be implementation-specific limits.	Optional	1pt 5pt 1mm 5mm
margintop	Sets the top margin of a box. Negative values for margin properties are allowed, but there may be implementation-specific limits.	Optional	1pt 5pt 1mm 5mm
marginleft	Sets the left margin of a box. Negative values for margin properties are allowed, but there may be implementation-specific limits.	Optional	1pt 5pt 1mm 5mm
marginbottom	Sets the bottom margin of a box. Negative values for margin properties are allowed, but there may be implementation-specific limits.	Optional	1pt 5pt 1mm 5mm
marginright	Sets the right margin of a box. Negative values for margin properties are allowed, but there may be implementation-specific limits.	Optional	1pt 5pt 1mm 5mm
Padding			
padding	Specifies the width of the padding of a block-area or inline-area. Unlike margin properties, values for padding properties cannot be negative.	Optional	1pt 5pt 1mm 5mm
paddingtop	Specifies the width of the padding on the top of a block-area or inline-area. Unlike margin properties, values for padding properties cannot be negative.	Optional	1pt 5pt

			1mm 5mm
paddingleft	Specifies the width of the padding on the left of a block-area or inline-area. Unlike margin properties, values for padding properties cannot be negative.	Optional	1pt 5pt 1mm 5mm
paddingbottom	Specifies the width of the padding on the bottom of a block-area or inline-area. Unlike margin properties, values for padding properties cannot be negative.	Optional	1pt 5pt 1mm 5mm
paddingright	Specifies the width of the padding on the right of a block-area or inline-area. Unlike margin properties, values for padding properties cannot be negative.	Optional	1pt 5pt 1mm 5mm

11 CISFO:COLTABLE2

- Properties 94

The CISFO:COLTABLE2 control is a container. It is placed into a row (e.g. CISFO:ROW2) and spans one column.

Properties

Basis			
columnwidths	Semicolon separated list of widths of contained columns. Example: "1cm; 3cm; 4cm". Column definitions are defined as part of the table definition - all contained column controls you add later on as content of the table are sized according to this definition.	Obligatory	10cm 8cm;8cm 4cm;5cm;4cm;5cm
stylename	Name of a CISFO style. The style must exist within the form definition.	Optional	
backgroundcolor	This attribute sets the background color of an element, either a andlt;colorandgt; value or the keyword "transparent", to make the underlying colors shine through.	Optional	#000000 #808080 #FFFFFF #FF0000
color	Color in which the element is output.	Optional	#000000 #808080 #FFFFFF #FF0000
textalign	The alignment of the text. Valid values are 'center', 'left' or 'right'.	Optional	left center right
textdecoration	Different values for printing the text, so as underline, overline or line-through.	Optional	overline line-through underline
Border			

bordercolor	Color for all borders (bottom, top, left, right). Values are any valid color specification.	Optional	#000000 #808080 #FFFFFF #FF0000
borderstyle	This attribute specifies the line style of all box's border. Some valid values are "solid", "double" or "dashed".	Optional	solid none
borderwidth	The border width attribute specifies the width of all borders. Some valid values are "thin", "medium", "thick" or a concrete size (e.g. "2pt").	Optional	1pt 5pt 1mm 5mm
bordertopcolor	Color for the border on the top. Values are any valid color specification.	Optional	#000000 #808080 #FFFFFF #FF0000
bordertopstyle	This attribute specifies the line style of a box's top border. Some valid values are "solid", "double" or "dashed".	Optional	solid none
bordertopwidth	The border width attribute specifies the width of the top border. Some valid values are "thin", "medium", "thick" or a concrete size (e.g. "2pt").	Optional	1pt 5pt 1mm 5mm
borderleftcolor	Color for the border on the left side. Values are any valid color specification.	Optional	#000000 #808080 #FFFFFF #FF0000
borderleftstyle	This attribute specifies the line style of a box's left border. Some valid values are "solid", "double" or "dashed".	Optional	solid none

borderleftwidth	The border width attribute specifies the width of the left border. Some valid values are "thin", "medium", "thick" or a concrete size (e.g. "2pt").	Optional	1pt 5pt 1mm 5mm
borderbottomcolor	Color for the border on the bottom. Values are any valid color specification.	Optional	#000000 #808080 #FFFFFF #FF0000
borderbottomstyle	This attribute specifies the line style of a box's bottom border. Valid values are 'solid', 'double' or 'dashed'.	Optional	solid none
borderbottomwidth	The border width attribute specifies the width of the bottom border. Valid values are "thin", "medium", "thick" or a concrete size (e.g. "2pt").	Optional	1pt 5pt 1mm 5mm
borderrightcolor	Color for the border on the right side. Values are any valid color specification.	Optional	#000000 #808080 #FFFFFF #FF0000
borderrightstyle	This attribute specifies the line style of a box's right border. Some valid values are "solid", "double" or "dashed".	Optional	solid none
borderrightwidth	The border width attribute specifies the width of the right border. Some valid values are "thin", "medium", "thick" or a concrete size (e.g. "2pt").	Optional	1pt 5pt 1mm 5mm
Font			
fontfamily	This attribute specifies a priorized list of font family names and/or generic family names. To deal with the problem that a single font may not contain glyphs to	Optional	Helvetica Times

	display all the characters in a document, or that not all fonts are available on all systems, this property allows authors to specify a list of fonts, all of the same style and size, that are tried in sequence to see if they contain a glyph for a certain character. This list is called a "font set".		Courier Symbol ZapfDingbats
fontsize	This attribute describes the size of the font when set as "solid". The font size corresponds to the em square, a concept used in typography. Note that certain glyphs may bleed outside their em squares. Some valid values are "xxsmall", "medium", "smaller" or a concrete size, e.g. "110%".	Optional	8pt 10pt 16pt
fontstyle	This attribute requests normal (sometimes referred to as "roman" or "upright"), italic, and oblique faces within a font family. Valid values are "normal", "italic" and "oblique".	Optional	normal italic oblique
fontvariant	In a small-caps font, the glyphs for lowercase letters look similar to the uppercase ones, but in a smaller size and with slightly different proportions. The "font-variant" property requests such a font for bicameral (having two cases, as with Roman script). This property has no visible effect for scripts that are unicameral (having only one case, as with most of the world's writing systems). Valid values are "normal" and "small-caps".	Optional	normal small-caps
fontweight	This attribute specifies the weight of the font. Some valid values are 'normal', 'bold', 'bolder', 'lighter', '100' or '900', where 'normal' is the same as '400', 'bold' is the same as '700'.	Optional	normal bold 100 200 300 400 500 600 700 800 900
Margin			

margin	Sets the margin of a box. Negative values for margin properties are allowed, but there may be implementation-specific limits.	Optional	1pt 5pt 1mm 5mm
margintop	Sets the top margin of a box. Negative values for margin properties are allowed, but there may be implementation-specific limits.	Optional	1pt 5pt 1mm 5mm
marginleft	Sets the left margin of a box. Negative values for margin properties are allowed, but there may be implementation-specific limits.	Optional	1pt 5pt 1mm 5mm
marginbottom	Sets the bottom margin of a box. Negative values for margin properties are allowed, but there may be implementation-specific limits.	Optional	1pt 5pt 1mm 5mm
marginright	Sets the right margin of a box. Negative values for margin properties are allowed, but there may be implementation-specific limits.	Optional	1pt 5pt 1mm 5mm
Padding			
padding	Specifies the width of the padding of a block-area or inline-area. Unlike margin properties, values for padding properties cannot be negative.	Optional	1pt 5pt 1mm 5mm
paddingtop	Specifies the width of the padding on the top of a block-area or inline-area. Unlike margin properties, values for padding properties cannot be negative.	Optional	1pt 5pt

			1mm 5mm
paddingleft	Specifies the width of the padding on the left of a block-area or inline-area. Unlike margin properties, values for padding properties cannot be negative.	Optional	1pt 5pt 1mm 5mm
paddingbottom	Specifies the width of the padding on the bottom of a block-area or inline-area. Unlike margin properties, values for padding properties cannot be negative.	Optional	1pt 5pt 1mm 5mm
paddingright	Specifies the width of the padding on the right of a block-area or inline-area. Unlike margin properties, values for padding properties cannot be negative.	Optional	1pt 5pt 1mm 5mm

IV Distance Controls

There are two distance controls:

- CISFO:VDIST2 - keeps a certain vertical distance, i.e. it represents an empty row with a certain height.
- CISFO:HDIST2 - this is an empty column inside a container's row. There is no further width definition with HDIST2 because the width of the columns is part of the container definition.

In addition, the CISFO:HLINE2 control is available which creates a horizontal line.

The information provided in this part is organized under the following headings:

CISFO:VDIST2
CISFO:HDIST2
CISFO:HLINE2

12 CISFO:VDIST2

- Properties 80

The CISFO:VDIST2 control keeps a certain vertical distance, i.e. it represents an empty row with a certain height.

Properties

Basic			
height	The height of the control. Use unit 'mm', 'cm' or 'pt'.	Optional	1pt 12pt 0.5in 1in 1mm 5mm 1cm 1.5cm

13 CISFO:HDIST2

- Properties 82

The CISFO:HDIST2 control is an empty column inside a container's row. There is no further width definition with HDIST2 because the width of the columns is part of the container definition.

Properties

Basis			
stylename	Name of a CISFO style. The style must exist within the form definition.	Optional	
backgroundcolor	This attribute sets the background color of an element, either a andlt;colorandgt; value or the keyword "transparent", to make the underlying colors shine through.	Optional	#000000 #808080 #FFFFFF #FF0000
columnsspanned	A positive integer, specifying the number of columns spanned.	Optional	2 3
rowsspanned	A positive integer, specifying the number of rows taht this cell spans.	Optional	2 3
Border			
bordercolor	Color for all borders (bottom, top, left, right). Values are any valid color specification.	Optional	#000000 #808080 #FFFFFF #FF0000
borderstyle	This attribute specifies the line style of all box's border. Some valid values are "solid", "double" or "dashed".	Optional	solid none
borderwidth	The border width attribute specifies the width of all borders. Some valid values are "thin", "medium", "thick" or a concrete size (e.g. "2pt").	Optional	1pt 5pt 1mm 5mm
bordertopcolor	Color for the border on the top. Values are any valid color specification.	Optional	#000000

			#808080 #FFFFFF #FF0000
bordertopstyle	This attribute specifies the line style of a box's top border. Some valid values are "solid", "double" or "dashed".	Optional	solid none
bordertopwidth	The border width attribute specifies the width of the top border. Some valid values are "thin", "medium", "thick" or a concrete size (e.g. "2pt").	Optional	1pt 5pt 1mm 5mm
borderleftcolor	Color for the border on the left side. Values are any valid color specification.	Optional	#000000 #808080 #FFFFFF #FF0000
borderleftstyle	This attribute specifies the line style of a box's left border. Some valid values are "solid", "double" or "dashed".	Optional	solid none
borderleftwidth	The border width attribute specifies the width of the left border. Some valid values are "thin", "medium", "thick" or a concrete size (e.g. "2pt").	Optional	1pt 5pt 1mm 5mm
borderbottomcolor	Color for the border on the bottom. Values are any valid color specification.	Optional	#000000 #808080 #FFFFFF #FF0000
borderbottomstyle	This attribute specifies the line style of a box's bottom border. Valid values are 'solid', 'double' or 'dashed'.	Optional	solid none

borderbottomwidth	The border width attribute specifies the width of the bottom border. Valid values are "thin", "medium", "thick" or a concrete size (e.g. "2pt").	Optional	1pt 5pt 1mm 5mm
borderrightcolor	Color for the border on the right side. Values are any valid color specification.	Optional	#000000 #808080 #FFFFFF #FF0000
borderrightstyle	This attribute specifies the line style of a box's right border. Some valid values are "solid", "double" or "dashed".	Optional	solid none
borderrightwidth	The border width attribute specifies the width of the right border. Some valid values are "thin", "medium", "thick" or a concrete size (e.g. "2pt").	Optional	1pt 5pt 1mm 5mm
Margin			
margin	Sets the margin of a box. Negative values for margin properties are allowed, but there may be implementation-specific limits.	Optional	1pt 5pt 1mm 5mm
margintop	Sets the top margin of a box. Negative values for margin properties are allowed, but there may be implementation-specific limits.	Optional	1pt 5pt 1mm 5mm
marginleft	Sets the left margin of a box. Negative values for margin properties are allowed, but there may be implementation-specific limits.	Optional	1pt 5pt 1mm 5mm

marginbottom	Sets the bottom margin of a box. Negative values for margin properties are allowed, but there may be implementation-specific limits.	Optional	1pt 5pt 1mm 5mm
marginright	Sets the right margin of a box. Negative values for margin properties are allowed, but there may be implementation-specific limits.	Optional	1pt 5pt 1mm 5mm

14 CISFO:HLINE2

- Properties 88

CISFO:HLINE2 is a row control (spans all columns of the container definition) representing a horizontal line.

Properties

Basic			
height	Height of the line.	Optional	1pt 12pt 0.5in 1in 1mm 5mm 1cm 1.5cm
color	Color of the line.	Optional	#000000 #808080 #FFFFFF #FF0000

V Text Output Controls

The information provided in this part is organized under the following headings:

General Information

CISFO:ROWTEXTBLOCK2

CISFO:CELLTEXT2

CISFO:TEXT2

CISFO:TEXTBLOCK2

CISFO:TEXTREPLACE2

CISFO:REPLACE2

15 General Information

The output of plain text is done using the following controls:

- CISFO:ROWTEXTBLOCK2
- CISFO:CELLTEXT2

CISFO:ROWTEXTBLOCK2 is a row control (it spans all columns of the container definition). CISFO:CELLTEXT2 covers one table column (cell). With both controls, you either can define a text directly inside the control definition. Or you can set up a text containing the following elements to be arranged as one block of text:

- CISFO:TEXT2
- CISFO:TEXTREPLACE2
- CISFO:TEXTBLOCK2

With the above controls, you either define the text in a fixed way (using the `text` property) or language-dependent (using the `textid` property). The language is passed at runtime as property of the data object.

For variable text output, you can use the following controls:

- CISFO:REPLACE2 as column control
- CISFO:TEXTREPLACE2 inside a text definition

Both controls bind to a data object property that contains the text at runtime.

16 CISFO:ROWTEXTBLOCK2

- Properties 94

The CISFO:ROWTEXTBLOCK2 control outputs a block of plain text. It is placed into a table container (e.g. CISFO:BODY2 or CISFO:ROWTABLE2) and spans one row. You can either define a fixed text directly inside the control definition (properties `text` and `textid`). Or you can set up a text containing certain elements to be arranged as one block of text:

```
<cisfo:rowtextblock2>
    <cisfo:text2 text="The amount to pay is ">
    </cisfo:text2>
    <cisfo:textreplace2 valueprop="amount" fontweight="bold">
    </cisfo:textreplace2>
    <cisfo:text2 text=" Euro. Please use our bank account ">
    </cisfo:text2>
    <cisfo:textreplace2 valueprop="account" fontweight="bold">
    </cisfo:textreplace2>
   <cisfo:text2 text=". Thanks for your order. Some additional text.Some additional ↩
text.Some additional text.Some additional text.Some additional text.">
    </cisfo:text2>
</cisfo:rowtextblock2>
```

In this case, the text consists of four individual texts that are arranged as one block of text.

Properties

Basis			
text	The text that is output.	Optional	
textid	Textid for the multi language management.	Optional	
stylename	Name of a CISFO style. The style must exist within the form definition.	Optional	
backgroundcolor	This attribute sets the background color of an element, either a andlt;colorandgt; value or the keyword "transparent", to make the underlying colors shine through.	Optional	#000000 #808080 #FFFFFF #FF0000
color	Color in which the element is output.	Optional	#000000 #808080 #FFFFFF #FF0000
textalign	The alignment of the text. Valid values are 'center', 'left' or 'right'.	Optional	left

			center right
textdecoration	Different values for printing the text, so as underline, overline or line-through.	Optional	overline line-through underline
Border			
bordercolor	Color for all borders (bottom, top, left, right). Values are any valid color specification.	Optional	#000000 #808080 #FFFFFF #FF0000
borderstyle	This attribute specifies the line style of all box's border. Some valid values are "solid", "double" or "dashed".	Optional	solid none
borderwidth	The border width attribute specifies the width of all borders. Some valid values are "thin", "medium", "thick" or a concrete size (e.g. "2pt").	Optional	1pt 5pt 1mm 5mm
bordertopcolor	Color for the border on the top. Values are any valid color specification.	Optional	#000000 #808080 #FFFFFF #FF0000
bordertopstyle	This attribute specifies the line style of a box's top border. Some valid values are "solid", "double" or "dashed".	Optional	solid none
bordertopwidth	The border width attribute specifies the width of the top border. Some valid values are "thin", "medium", "thick" or a concrete size (e.g. "2pt").	Optional	1pt 5pt 1mm 5mm

borderleftcolor	Color for the border on the left side. Values are any valid color specification.	Optional	#000000 #808080 #FFFFFF #FF0000
borderleftstyle	This attribute specifies the line style of a box's left border. Some valid values are "solid", "double" or "dashed".	Optional	solid none
borderleftwidth	The border width attribute specifies the width of the left border. Some valid values are "thin", "medium", "thick" or a concrete size (e.g. "2pt").	Optional	1pt 5pt 1mm 5mm
borderbottomcolor	Color for the border on the bottom. Values are any valid color specification.	Optional	#000000 #808080 #FFFFFF #FF0000
borderbottomstyle	This attribute specifies the line style of a box's bottom border. Valid values are 'solid', 'double' or 'dashed'.	Optional	solid none
borderbottomwidth	The border width attribute specifies the width of the bottom border. Valid values are "thin", "medium", "thick" or a concrete size (e.g. "2pt").	Optional	1pt 5pt 1mm 5mm
borderrightcolor	Color for the border on the right side. Values are any valid color specification.	Optional	#000000 #808080 #FFFFFF #FF0000
borderrightstyle	This attribute specifies the line style of a box's right border. Some valid values are "solid", "double" or "dashed".	Optional	solid none

borderrightwidth	The border width attribute specifies the width of the right border. Some valid values are "thin", "medium", "thick" or a concrete size (e.g. "2pt").	Optional	1pt 5pt 1mm 5mm
Font			
fontfamily	This attribute specifies a priorized list of font family names and/or generic family names. To deal with the problem that a single font may not contain glyphs to display all the characters in a document, or that not all fonts are available on all systems, this property allows authors to specify a list of fonts, all of the same style and size, that are tried in sequence to see if they contain a glyph for a certain character. This list is called a "font set".	Optional	Helvetica Times Courier Symbol ZapfDingbats
fontsize	This attribute describes the size of the font when set as "solid". The font size corresponds to the em square, a concept used in typography. Note that certain glyphs may bleed outside their em squares. Some valid values are "xxsmall", "medium", "smaller" or a concrete size, e.g. "110%".	Optional	8pt 10pt 16pt
fontstyle	This attribute requests normal (sometimes referred to as "roman" or "upright"), italic, and oblique faces within a font family. Valid values are "normal", "italic" and "oblique".	Optional	normal italic oblique
fontvariant	In a small-caps font, the glyphs for lowercase letters look similar to the uppercase ones, but in a smaller size and with slightly different proportions. The "font-variant" property requests such a font for bicameral (having two cases, as with Roman script). This property has no visible effect for scripts that are unicameral (having only one case, as with most of the world's writing systems). Valid values are "normal" and "small-caps".	Optional	normal small-caps
fontweight	This attribute specifies the weight of the font. Some valid values are 'normal', 'bold', 'bolder', 'lighter', '100' or '900', where 'normal' is the same as '400', 'bold' is the same as '700'.	Optional	normal bold 100 200 300 400 500

			600 700 800 900
Margin			
margin	Sets the margin of a box. Negative values for margin properties are allowed, but there may be implementation-specific limits.	Optional	1pt 5pt 1mm 5mm
margintop	Sets the top margin of a box. Negative values for margin properties are allowed, but there may be implementation-specific limits.	Optional	1pt 5pt 1mm 5mm
marginleft	Sets the left margin of a box. Negative values for margin properties are allowed, but there may be implementation-specific limits.	Optional	1pt 5pt 1mm 5mm
marginbottom	Sets the bottom margin of a box. Negative values for margin properties are allowed, but there may be implementation-specific limits.	Optional	1pt 5pt 1mm 5mm
marginright	Sets the right margin of a box. Negative values for margin properties are allowed, but there may be implementation-specific limits.	Optional	1pt 5pt 1mm 5mm
Padding			

padding	Specifies the width of the padding of a block-area or inline-area. Unlike margin properties, values for padding properties cannot be negative.	Optional	1pt 5pt 1mm 5mm
paddingtop	Specifies the width of the padding on the top of a block-area or inline-area. Unlike margin properties, values for padding properties cannot be negative.	Optional	1pt 5pt 1mm 5mm
paddingleft	Specifies the width of the padding on the left of a block-area or inline-area. Unlike margin properties, values for padding properties cannot be negative.	Optional	1pt 5pt 1mm 5mm
paddingbottom	Specifies the width of the padding on the bottom of a block-area or inline-area. Unlike margin properties, values for padding properties cannot be negative.	Optional	1pt 5pt 1mm 5mm
paddingright	Specifies the width of the padding on the right of a block-area or inline-area. Unlike margin properties, values for padding properties cannot be negative.	Optional	1pt 5pt 1mm 5mm

17 CISFO:CELLTEXT2

- Properties 102

The CISFO:CELLTEXT2 control outputs plain text. It is placed into a row container (e.g. CISFO:ROW2) and spans one table cell. You can either define a fixed text directly inside the control definition (properties `text` and `textid`). Or you can set up a text containing certain elements to be arranged as one block of text:

```
<cisfo:celltext2 stylename="LABEL">
    <cisfo:text2 text="Number">
    </cisfo:text2>
    <cisfo:textreplace2 valueprop="docNumber">
    </cisfo:textreplace2>
</cisfo:celltext2>
```

Properties

Basis			
text	The text that is output.	Optional	
textid	Textid for the multi language management.	Optional	
stylename	Name of a CISFO style. The style must exist within the form definition.	Optional	
backgroundcolor	This attribute sets the background color of an element, either a andlt;colorandgt; value or the keyword "transparent", to make the underlying colors shine through.	Optional	#000000 #808080 #FFFFFF #FF0000
columnsspanned	A positive integer, specifying the number of columns spanned.	Optional	2 3
color	Color in which the element is output.	Optional	#000000 #808080 #FFFFFF #FF0000
rowsspanned	A positive integer, specifying the number of rows taht this cell spans.	Optional	2 3
textalign	The alignment of the text. Valid values are 'center', 'left' or 'right'.	Optional	left center

			right
textdecoration	Different values for printing the text, so as underline, overline or line-through.	Optional	overline line-through underline
Border			
bordercolor	Color for all borders (bottom, top, left, right). Values are any valid color specification.	Optional	#000000 #808080 #FFFFFF #FF0000
borderstyle	This attribute specifies the line style of all box's border. Some valid values are "solid", "double" or "dashed".	Optional	solid none
borderwidth	The border width attribute specifies the width of all borders. Some valid values are "thin", "medium", "thick" or a concrete size (e.g. "2pt").	Optional	1pt 5pt 1mm 5mm
bordertopcolor	Color for the border on the top. Values are any valid color specification.	Optional	#000000 #808080 #FFFFFF #FF0000
bordertopstyle	This attribute specifies the line style of a box's top border. Some valid values are "solid", "double" or "dashed".	Optional	solid none
bordertopwidth	The border width attribute specifies the width of the top border. Some valid values are "thin", "medium", "thick" or a concrete size (e.g. "2pt").	Optional	1pt 5pt 1mm 5mm
borderleftcolor	Color for the border on the left side. Values are any valid color specification.	Optional	#000000 #808080

			#FFFFFF #FF0000
borderleftstyle	This attribute specifies the line style of a box's left border. Some valid values are "solid", "double" or "dashed".	Optional	solid none
borderleftwidth	The border width attribute specifies the width of the left border. Some valid values are "thin", "medium", "thick" or a concrete size (e.g. "2pt").	Optional	1pt 5pt 1mm 5mm
borderbottomcolor	Color for the border on the bottom. Values are any valid color specification.	Optional	#000000 #808080 #FFFFFF #FF0000
borderbottomstyle	This attribute specifies the line style of a box's bottom border. Valid values are 'solid', 'double' or 'dashed'.	Optional	solid none
borderbottomwidth	The border width attribute specifies the width of the bottom border. Valid values are "thin", "medium", "thick" or a concrete size (e.g. "2pt").	Optional	1pt 5pt 1mm 5mm
borderrightcolor	Color for the border on the right side. Values are any valid color specification.	Optional	#000000 #808080 #FFFFFF #FF0000
borderrightstyle	This attribute specifies the line style of a box's right border. Some valid values are "solid", "double" or "dashed".	Optional	solid none
borderrightwidth	The border width attribute specifies the width of the right border. Some valid values are "thin", "medium", "thick" or a concrete size (e.g. "2pt").	Optional	1pt 5pt

			1mm 5mm
Font			
fontfamily	This attribute specifies a priorized list of font family names and/or generic family names. To deal with the problem that a single font may not contain glyphs to display all the characters in a document, or that not all fonts are available on all systems, this property allows authors to specify a list of fonts, all of the same style and size, that are tried in sequence to see if they contain a glyph for a certain character. This list is called a "font set".	Optional	Helvetica Times Courier Symbol ZapfDingbats
fontsize	This attribute describes the size of the font when set as "solid". The font size corresponds to the em square, a concept used in typography. Note that certain glyphs may bleed outside their em squares. Some valid values are "xxsmall", "medium", "smaller" or a concrete size, e.g. "110%".	Optional	8pt 10pt 16pt
fontstyle	This attribute requests normal (sometimes referred to as "roman" or "upright"), italic, and oblique faces within a font family. Valid values are "normal", "italic" and "oblique".	Optional	normal italic oblique
fontvariant	In a small-caps font, the glyphs for lowercase letters look similar to the uppercase ones, but in a smaller size and with slightly different proportions. The "font-variant" property requests such a font for bicameral (having two cases, as with Roman script). This property has no visible effect for scripts that are unicameral (having only one case, as with most of the world's writing systems). Valid values are "normal" and "small-caps".	Optional	normal small-caps
fontweight	This attribute specifies the weight of the font. Some valid values are 'normal', 'bold', 'bolder', 'lighter', '100' or '900', where 'normal' is the same as '400', 'bold' is the same as '700'.	Optional	normal bold 100 200 300 400 500 600 700

			800 900
Margin			
margin	Sets the margin of a box. Negative values for margin properties are allowed, but there may be implementation-specific limits.	Optional	1pt 5pt 1mm 5mm
margintop	Sets the top margin of a box. Negative values for margin properties are allowed, but there may be implementation-specific limits.	Optional	1pt 5pt 1mm 5mm
marginleft	Sets the left margin of a box. Negative values for margin properties are allowed, but there may be implementation-specific limits.	Optional	1pt 5pt 1mm 5mm
marginbottom	Sets the bottom margin of a box. Negative values for margin properties are allowed, but there may be implementation-specific limits.	Optional	1pt 5pt 1mm 5mm
marginright	Sets the right margin of a box. Negative values for margin properties are allowed, but there may be implementation-specific limits.	Optional	1pt 5pt 1mm 5mm
Padding			
padding	Specifies the width of the padding of a block-area or inline-area. Unlike margin properties, values for padding properties cannot be negative.	Optional	1pt 5pt 1mm

			5mm
paddingtop	Specifies the width of the padding on the top of a block-area or inline-area. Unlike margin properties, values for padding properties cannot be negative.	Optional	1pt 5pt 1mm 5mm
paddingleft	Specifies the width of the padding on the left of a block-area or inline-area. Unlike margin properties, values for padding properties cannot be negative.	Optional	1pt 5pt 1mm 5mm
paddingbottom	Specifies the width of the padding on the bottom of a block-area or inline-area. Unlike margin properties, values for padding properties cannot be negative.	Optional	1pt 5pt 1mm 5mm
paddingright	Specifies the width of the padding on the right of a block-area or inline-area. Unlike margin properties, values for padding properties cannot be negative.	Optional	1pt 5pt 1mm 5mm

18 CISFO:TEXT2

- Properties 110

The CISFO:TEXT2 control outputs plain text. It is placed into a text block container (ROWTEXTBLOCK2 or CELLTEXT2). You either define the text in a fixed way (using the `text` property) or language-dependent (using the `textid` property). The language is passed at runtime as property of the data object.

Properties

Basis			
text	The text that is output.	Optional	
textid	Textid for the multi language management.	Optional	
stylename	Name of a CISFO style. The style must exist within the form definition.	Optional	
color	Color in which the element is output.	Optional	#000000 #808080 #FFFFFF #FF0000
textalign	The alignment of the text. Valid values are 'center', 'left' or 'right'.	Optional	left center right
textdecoration	Different values for printing the text, so as underline, overline or line-through.	Optional	overline line-through underline
Font			
fontfamily	This attribute specifies a priorized list of font family names and/or generic family names. To deal with the problem that a single font may not contain glyphs to display all the characters in a document, or that not all fonts are available on all systems, this property allows authors to specify a list of fonts, all of the same style and size, that are tried in sequence to see if they contain a glyph for a certain character. This list is called a "font set".	Optional	Helvetica Times Courier Symbol ZapfDingbats
fontsize	This attribute describes the size of the font when set as "solid". The font size corresponds to the em square, a concept used in typography. Note that certain glyphs may bleed outside their em	Optional	8pt 10pt

	squares. Some valid values are "xxsmall", "medium", "smaller" or a concrete size, e.g. "110%".		16pt
fontstyle	This attribute requests normal (sometimes referred to as "roman" or "upright"), italic, and oblique faces within a font family. Valid values are "normal", "italic" and "oblique".	Optional	normal italic oblique
fontvariant	In a small-caps font, the glyphs for lowercase letters look similar to the uppercase ones, but in a smaller size and with slightly different proportions. The "font-variant" property requests such a font for bicameral (having two cases, as with Roman script). This property has no visible effect for scripts that are unicameral (having only one case, as with most of the world's writing systems). Valid values are "normal" and "small-caps".	Optional	normal small-caps
fontweight	This attribute specifies the weight of the font. Some valid values are 'normal', 'bold', 'bolder', 'lighter', '100' or '900', where 'normal' is the same as '400', 'bold' is the same as '700'.	Optional	normal bold 100 200 300 400 500 600 700 800 900
Margin			
margin	Sets the margin of a box. Negative values for margin properties are allowed, but there may be implementation-specific limits.	Optional	1pt 5pt 1mm 5mm
margintop	Sets the top margin of a box. Negative values for margin properties are allowed, but there may be implementation-specific limits.	Optional	1pt 5pt 1mm

			5mm
marginleft	Sets the left margin of a box. Negative values for margin properties are allowed, but there may be implementation-specific limits.	Optional	1pt 5pt 1mm 5mm
marginbottom	Sets the bottom margin of a box. Negative values for margin properties are allowed, but there may be implementation-specific limits.	Optional	1pt 5pt 1mm 5mm
marginright	Sets the right margin of a box. Negative values for margin properties are allowed, but there may be implementation-specific limits.	Optional	1pt 5pt 1mm 5mm
Padding			
padding	Specifies the width of the padding of a block-area or inline-area. Unlike margin properties, values for padding properties cannot be negative.	Optional	1pt 5pt 1mm 5mm
paddingtop	Specifies the width of the padding on the top of a block-area or inline-area. Unlike margin properties, values for padding properties cannot be negative.	Optional	1pt 5pt 1mm 5mm
paddingleft	Specifies the width of the padding on the left of a block-area or inline-area. Unlike margin properties, values for padding properties cannot be negative.	Optional	1pt 5pt 1mm 5mm

paddingbottom	Specifies the width of the padding on the bottom of a block-area or inline-area. Unlike margin properties, values for padding properties cannot be negative.	Optional	1pt 5pt 1mm 5mm
paddingright	Specifies the width of the padding on the right of a block-area or inline-area. Unlike margin properties, values for padding properties cannot be negative.	Optional	1pt 5pt 1mm 5mm

19 CISFO:TEXTBLOCK2

- Properties 116

The CISFO:TEXTBLOCK2 control outputs plain text that starts within a new line. It is placed into a text block container (ROWTEXTBLOCK2 or CELLTEXT2). You either define the text in a fixed way (using the `text` property) or language-dependent (using the `textid` property). The language is passed at runtime as property of the data object.

Properties

Basis			
text	The text that is output.	Optional	
textid	Textid for the multi language management.	Optional	
stylename	Name of a CISFO style. The style must exist within the form definition.	Optional	
backgroundcolor	This attribute sets the background color of an element, either a andlt;colorandgt; value or the keyword "transparent", to make the underlying colors shine through.	Optional	#000000 #808080 #FFFFFF #FF0000
color	Color in which the element is output.	Optional	#000000 #808080 #FFFFFF #FF0000
textalign	The alignment of the text. Valid values are 'center', 'left' or 'right'.	Optional	left center right
textdecoration	Different values for printing the text, so as underline, overline or line-through.	Optional	overline line-through underline
Border			
bordercolor	Color for all borders (bottom, top, left, right). Values are any valid color specification.	Optional	#000000 #808080 #FFFFFF

			#FF0000
borderstyle	This attribute specifies the line style of all box's border. Some valid values are "solid", "double" or "dashed".	Optional	solid none
borderwidth	The border width attribute specifies the width of all borders. Some valid values are "thin", "medium", "thick" or a concrete size (e.g. "2pt").	Optional	1pt 5pt 1mm 5mm
bordertopcolor	Color for the border on the top. Values are any valid color specification.	Optional	#000000 #808080 #FFFFFF #FF0000
bordertopstyle	This attribute specifies the line style of a box's top border. Some valid values are "solid", "double" or "dashed".	Optional	solid none
bordertopwidth	The border width attribute specifies the width of the top border. Some valid values are "thin", "medium", "thick" or a concrete size (e.g. "2pt").	Optional	1pt 5pt 1mm 5mm
borderleftcolor	Color for the border on the left side. Values are any valid color specification.	Optional	#000000 #808080 #FFFFFF #FF0000
borderleftstyle	This attribute specifies the line style of a box's left border. Some valid values are "solid", "double" or "dashed".	Optional	solid none
borderleftwidth	The border width attribute specifies the width of the left border. Some valid values are "thin", "medium", "thick" or a concrete size (e.g. "2pt").	Optional	1pt 5pt 1mm

			5mm
borderbottomcolor	Color for the border on the bottom. Values are any valid color specification.	Optional	#000000 #808080 #FFFFFF #FF0000
borderbottomstyle	This attribute specifies the line style of a box's bottom border. Valid values are 'solid', 'double' or 'dashed'.	Optional	solid none
borderbottomwidth	The border width attribute specifies the width of the bottom border. Valid values are "thin", "medium", "thick" or a concrete size (e.g. "2pt").	Optional	1pt 5pt 1mm 5mm
borderrightcolor	Color for the border on the right side. Values are any valid color specification.	Optional	#000000 #808080 #FFFFFF #FF0000
borderrightstyle	This attribute specifies the line style of a box's right border. Some valid values are "solid", "double" or "dashed".	Optional	solid none
borderrightwidth	The border width attribute specifies the width of the right border. Some valid values are "thin", "medium", "thick" or a concrete size (e.g. "2pt").	Optional	1pt 5pt 1mm 5mm
Font			
fontfamily	This attribute specifies a priorized list of font family names and/or generic family names. To deal with the problem that a single font may not contain glyphs to display all the characters in a document, or that not all fonts are available on all systems, this property allows authors to specify a list of fonts, all of the same style and size, that are tried in sequence to see if they contain a glyph for a certain character. This list is called a "font set".	Optional	Helvetica Times Courier Symbol ZapfDingbats

fontsize	This attribute describes the size of the font when set as "solid". The font size corresponds to the em square, a concept used in typography. Note that certain glyphs may bleed outside their em squares. Some valid values are "xxsmall", "medium", "smaller" or a concrete size, e.g. "110%".	Optional	8pt 10pt 16pt
fontstyle	This attribute requests normal (sometimes referred to as "roman" or "upright"), italic, and oblique faces within a font family. Valid values are "normal", "italic" and "oblique".	Optional	normal italic oblique
fontvariant	In a small-caps font, the glyphs for lowercase letters look similar to the uppercase ones, but in a smaller size and with slightly different proportions. The "font-variant" property requests such a font for bicameral (having two cases, as with Roman script). This property has no visible effect for scripts that are unicameral (having only one case, as with most of the world's writing systems). Valid values are "normal" and "small-caps".	Optional	normal small-caps
fontweight	This attribute specifies the weight of the font. Some valid values are 'normal', 'bold', 'bolder', 'lighter', '100' or '900', where 'normal' is the same as '400', 'bold' is the same as '700'.	Optional	normal bold 100 200 300 400 500 600 700 800 900
Margin			
margin	Sets the margin of a box. Negative values for margin properties are allowed, but there may be implementation-specific limits.	Optional	1pt 5pt 1mm 5mm

margintop	Sets the top margin of a box. Negative values for margin properties are allowed, but there may be implementation-specific limits.	Optional	1pt 5pt 1mm 5mm
marginleft	Sets the left margin of a box. Negative values for margin properties are allowed, but there may be implementation-specific limits.	Optional	1pt 5pt 1mm 5mm
marginbottom	Sets the bottom margin of a box. Negative values for margin properties are allowed, but there may be implementation-specific limits.	Optional	1pt 5pt 1mm 5mm
marginright	Sets the right margin of a box. Negative values for margin properties are allowed, but there may be implementation-specific limits.	Optional	1pt 5pt 1mm 5mm
Padding			
padding	Specifies the width of the padding of a block-area or inline-area. Unlike margin properties, values for padding properties cannot be negative.	Optional	1pt 5pt 1mm 5mm
paddingtop	Specifies the width of the padding on the top of a block-area or inline-area. Unlike margin properties, values for padding properties cannot be negative.	Optional	1pt 5pt 1mm 5mm
paddingleft	Specifies the width of the padding on the left of a block-area or inline-area. Unlike margin properties, values for padding properties cannot be negative.	Optional	1pt 5pt

			1mm 5mm
paddingbottom	Specifies the width of the padding on the bottom of a block-area or inline-area. Unlike margin properties, values for padding properties cannot be negative.	Optional	1pt 5pt 1mm 5mm
paddingright	Specifies the width of the padding on the right of a block-area or inline-area. Unlike margin properties, values for padding properties cannot be negative.	Optional	1pt 5pt 1mm 5mm

20 CISFO:TEXTREPLACE2

- Properties 124

Use the CISFO:TEXTREPLACE2 control for variable text output. The most important definition of the TEXTREPLACE2 control is the definition which Java property is bound to obtain the text at runtime. The Java property is part of the data object - the counterpart of the page (see property `objectclass` of control FOPPAGE2).

Properties

Basis			
valueprop	Adapter property that provides the value that is output within the control.	Obligatory	
datatype	Data type of the output. Some valid values are "date, float, int, time, timestamp". Dependent from the data type the text may be converted before being rendered. Example: If specifying "date" then the output will be transferred from "YYYYMMDD" into a date display dependent format, e.g. "DD.MM.YYYY" or "MM/DD/YYYY".	Optional	date float int boolean time timestamp
decimaldigitsprop	Adapter property that passes back information how many decimal digits are to be displayed. Only used with the datat type "float".	Optional	
booleantextprop	Adapter property that passes back text displayed for "boolean" values. Only use this with the datatype "boolean". The property must provide a string containing the text for "true" and "false" separated by a comma. Example: "Yes	Optional	
stylename	Name of a CISFO style. The style must exist within the form definition.	Optional	
color	Color in which the element is output.	Optional	#000000 #808080 #FFFFFF #FF0000
textalign	The alignment of the text. Valid values are 'center', 'left' or 'right'.	Optional	left center right

textdecoration	Different values for printing the text, so as underline, overline or line-through.	Optional	overline line-through underline
Font			
fontfamily	This attribute specifies a priorized list of font family names and/or generic family names. To deal with the problem that a single font may not contain glyphs to display all the characters in a document, or that not all fonts are available on all systems, this property allows authors to specify a list of fonts, all of the same style and size, that are tried in sequence to see if they contain a glyph for a certain character. This list is called a "font set".	Optional	Helvetica Times Courier Symbol ZapfDingbats
fontsize	This attribute describes the size of the font when set as "solid". The font size corresponds to the em square, a concept used in typography. Note that certain glyphs may bleed outside their em squares. Some valid values are "xxsmall", "medium", "smaller" or a concrete size, e.g. "110%".	Optional	8pt 10pt 16pt
fontstyle	This attribute requests normal (sometimes referred to as "roman" or "upright"), italic, and oblique faces within a font family. Valid values are "normal", "italic" and "oblique".	Optional	normal italic oblique
fontvariant	In a small-caps font, the glyphs for lowercase letters look similar to the uppercase ones, but in a smaller size and with slightly different proportions. The "font-variant" property requests such a font for bicameral (having two cases, as with Roman script). This property has no visible effect for scripts that are unicameral (having only one case, as with most of the world's writing systems). Valid values are "normal" and "small-caps".	Optional	normal small-caps
fontweight	This attribute specifies the weight of the font. Some valid values are 'normal', 'bold', 'bolder', 'lighter', '100' or '900', where 'normal' is the same as '400', 'bold' is the same as '700'.	Optional	normal bold 100 200 300 400 500 600

			700 800 900
Margin			
margin	Sets the margin of a box. Negative values for margin properties are allowed, but there may be implementation-specific limits.	Optional	1pt 5pt 1mm 5mm
margintop	Sets the top margin of a box. Negative values for margin properties are allowed, but there may be implementation-specific limits.	Optional	1pt 5pt 1mm 5mm
marginleft	Sets the left margin of a box. Negative values for margin properties are allowed, but there may be implementation-specific limits.	Optional	1pt 5pt 1mm 5mm
marginbottom	Sets the bottom margin of a box. Negative values for margin properties are allowed, but there may be implementation-specific limits.	Optional	1pt 5pt 1mm 5mm
marginright	Sets the right margin of a box. Negative values for margin properties are allowed, but there may be implementation-specific limits.	Optional	1pt 5pt 1mm 5mm
Padding			
padding	Specifies the width of the padding of a block-area or inline-area. Unlike margin properties, values for padding properties cannot be negative.	Optional	1pt 5pt

			1mm 5mm
paddingtop	Specifies the width of the padding on the top of a block-area or inline-area. Unlike margin properties, values for padding properties cannot be negative.	Optional	1pt 5pt 1mm 5mm
paddingleft	Specifies the width of the padding on the left of a block-area or inline-area. Unlike margin properties, values for padding properties cannot be negative.	Optional	1pt 5pt 1mm 5mm
paddingbottom	Specifies the width of the padding on the bottom of a block-area or inline-area. Unlike margin properties, values for padding properties cannot be negative.	Optional	1pt 5pt 1mm 5mm
paddingright	Specifies the width of the padding on the right of a block-area or inline-area. Unlike margin properties, values for padding properties cannot be negative.	Optional	1pt 5pt 1mm 5mm

21 CISFO:REPLACE2

- Properties 130

The CISFO:REPLACE2 is very similar to CISFO:TEXTREPLACE2. It outputs variable text in the same way as CISFO:TEXTREPLACE2. The only difference: CISFO:TEXTREPLACE2 represents a piece of text within a text block - CISFO:REPLACE2 itself represents a whole text block.

The most important definition of the TEXTREPLACE2 control is the definition which Java Property is bound to obtain the text at runtime. The Java property is part of the data object - the counter part of the page (see property `objectclass` of control FOPPAGE2).

For some data types, you can customize the rendering via additional properties such as `decimaldigitsprop` and `booleantextprop`.

For example, to render the value of a CHECKBOX control as "YES" or "NO", proceed as follows:

1. In the CISFO:REPLACE2 control, set the property `booleantextprop`.

```
<cisfo:replace2 valueprop="myvalue" booleantextprop="mytextprop"
 datatype="boolean"></cisfo:replace2>
```

2. At runtime, provide a comma-separated list of values for the rendering of the Boolean value in the "mytextprop" field. The values for this example would be "YES;NO".

For example, to render a float value with no decimal digits, proceed as follows:

1. In the CISFO:REPLACE2 control, set the property `decimaldigitsprop`.

```
<cisfo:replace2 valueprop="myvalue" decimaldigitsprop="mydigitsprop"
 datatype="float"></cisfo:replace2>
```

2. At runtime, provide the number of digits in the "mydigitsprop" field. The value for this example would be "0".

Properties

Basis			
valueprop	Adapter property that provides the value that is output within the control.	Obligatory	
datatype	Data type of the output. Some valid values are "date, float, int, time, timestamp". Dependent from the data type the text may be converted before being rendered. Example: If specifying "date" then the output will be transferred from "YYYYMMDD" into a date display	Optional	date float int boolean time

	dependent format, e.g. "DD.MM.YYYY" or "MM/DD/YYYY".		timestamp
decimaldigitsprop	Adapter property that passes back information how many decimal digits are to be displayed. Only used with the datat type "float".	Optional	
booleantextprop	Adapter property that passes back text displayed for "boolean" values. Only use this with the datatype "boolean". The property must provide a string containing the text for "true" and "false" separated by a comma. Example: "Yes	Optional	
stylename	Name of a CISFO style. The style must exist within the form definition.	Optional	
backgroundcolor	This attribute sets the background color of an element, either a andlt;colorandgt; value or the keyword "transparent", to make the underlying colors shine through.	Optional	#000000 #808080 #FFFFFF #FF0000
columnsspanned	A positive integer, specifying the number of columns spanned.	Optional	2 3
color	Color in which the element is output.	Optional	#000000 #808080 #FFFFFF #FF0000
rowsspanned	A positive integer, specifying the number of rows taht this cell spans.	Optional	2 3
textalign	The alignment of the text. Valid values are 'center', 'left' or 'right'.	Optional	left center right
textdecoration	Different values for printing the text, so as underline, overline or line-through.	Optional	overline line-through underline
Border			
bordercolor	Color for all borders (bottom, top, left, right). Values are any valid color specification.	Optional	#000000

			#808080 #FFFFFF #FF0000
borderstyle	This attribute specifies the line style of all box's border. Some valid values are "solid", "double" or "dashed".	Optional	solid none
borderwidth	The border width attribute specifies the width of all borders. Some valid values are "thin", "medium", "thick" or a concrete size (e.g. "2pt").	Optional	1pt 5pt 1mm 5mm
bordertopcolor	Color for the border on the top. Values are any valid color specification.	Optional	#000000 #808080 #FFFFFF #FF0000
bordertopstyle	This attribute specifies the line style of a box's top border. Some valid values are "solid", "double" or "dashed".	Optional	solid none
bordertopwidth	The border width attribute specifies the width of the top border. Some valid values are "thin", "medium", "thick" or a concrete size (e.g. "2pt").	Optional	1pt 5pt 1mm 5mm
borderleftcolor	Color for the border on the left side. Values are any valid color specification.	Optional	#000000 #808080 #FFFFFF #FF0000
borderleftstyle	This attribute specifies the line style of a box's left border. Some valid values are "solid", "double" or "dashed".	Optional	solid none

borderleftwidth	The border width attribute specifies the width of the left border. Some valid values are "thin", "medium", "thick" or a concrete size (e.g. "2pt").	Optional	1pt 5pt 1mm 5mm
borderbottomcolor	Color for the border on the bottom. Values are any valid color specification.	Optional	#000000 #808080 #FFFFFF #FF0000
borderbottomstyle	This attribute specifies the line style of a box's bottom border. Valid values are 'solid', 'double' or 'dashed'.	Optional	solid none
borderbottomwidth	The border width attribute specifies the width of the bottom border. Valid values are "thin", "medium", "thick" or a concrete size (e.g. "2pt").	Optional	1pt 5pt 1mm 5mm
borderrightcolor	Color for the border on the right side. Values are any valid color specification.	Optional	#000000 #808080 #FFFFFF #FF0000
borderrightstyle	This attribute specifies the line style of a box's right border. Some valid values are "solid", "double" or "dashed".	Optional	solid none
borderrightwidth	The border width attribute specifies the width of the right border. Some valid values are "thin", "medium", "thick" or a concrete size (e.g. "2pt").	Optional	1pt 5pt 1mm 5mm
Font			
fontfamily	This attribute specifies a priorized list of font family names and/or generic family names. To deal with the problem that a single font may not contain glyphs to display all the	Optional	Helvetica Times

	characters in a document, or that not all fonts are available on all systems, this property allows authors to specify a list of fonts, all of the same style and size, that are tried in sequence to see if they contain a glyph for a certain character. This list is called a "font set".		Courier Symbol ZapfDingbats
fontsize	This attribute describes the size of the font when set as "solid". The font size corresponds to the em square, a concept used in typography. Note that certain glyphs may bleed outside their em squares. Some valid values are "xxsmall", "medium", "smaller" or a concrete size, e.g. "110%".	Optional	8pt 10pt 16pt
fontstyle	This attribute requests normal (sometimes referred to as "roman" or "upright"), italic, and oblique faces within a font family. Valid values are "normal", "italic" and "oblique".	Optional	normal italic oblique
fontvariant	In a small-caps font, the glyphs for lowercase letters look similar to the uppercase ones, but in a smaller size and with slightly different proportions. The "font-variant" property requests such a font for bicameral (having two cases, as with Roman script). This property has no visible effect for scripts that are unicameral (having only one case, as with most of the world's writing systems). Valid values are "normal" and "small-caps".	Optional	normal small-caps
fontweight	This attribute specifies the weight of the font. Some valid values are 'normal', 'bold', 'bolder', 'lighter', '100' or '900', where 'normal' is the same as '400', 'bold' is the same as '700'.	Optional	normal bold 100 200 300 400 500 600 700 800 900
Margin			

margin	Sets the margin of a box. Negative values for margin properties are allowed, but there may be implementation-specific limits.	Optional	1pt 5pt 1mm 5mm
margintop	Sets the top margin of a box. Negative values for margin properties are allowed, but there may be implementation-specific limits.	Optional	1pt 5pt 1mm 5mm
marginleft	Sets the left margin of a box. Negative values for margin properties are allowed, but there may be implementation-specific limits.	Optional	1pt 5pt 1mm 5mm
marginbottom	Sets the bottom margin of a box. Negative values for margin properties are allowed, but there may be implementation-specific limits.	Optional	1pt 5pt 1mm 5mm
marginright	Sets the right margin of a box. Negative values for margin properties are allowed, but there may be implementation-specific limits.	Optional	1pt 5pt 1mm 5mm
Padding			
padding	Specifies the width of the padding of a block-area or inline-area. Unlike margin properties, values for padding properties cannot be negative.	Optional	1pt 5pt 1mm 5mm
paddingtop	Specifies the width of the padding on the top of a block-area or inline-area. Unlike margin properties, values for padding properties cannot be negative.	Optional	1pt 5pt

			1mm 5mm
paddingleft	Specifies the width of the padding on the left of a block-area or inline-area. Unlike margin properties, values for padding properties cannot be negative.	Optional	1pt 5pt 1mm 5mm
paddingbottom	Specifies the width of the padding on the bottom of a block-area or inline-area. Unlike margin properties, values for padding properties cannot be negative.	Optional	1pt 5pt 1mm 5mm
paddingright	Specifies the width of the padding on the right of a block-area or inline-area. Unlike margin properties, values for padding properties cannot be negative.	Optional	1pt 5pt 1mm 5mm

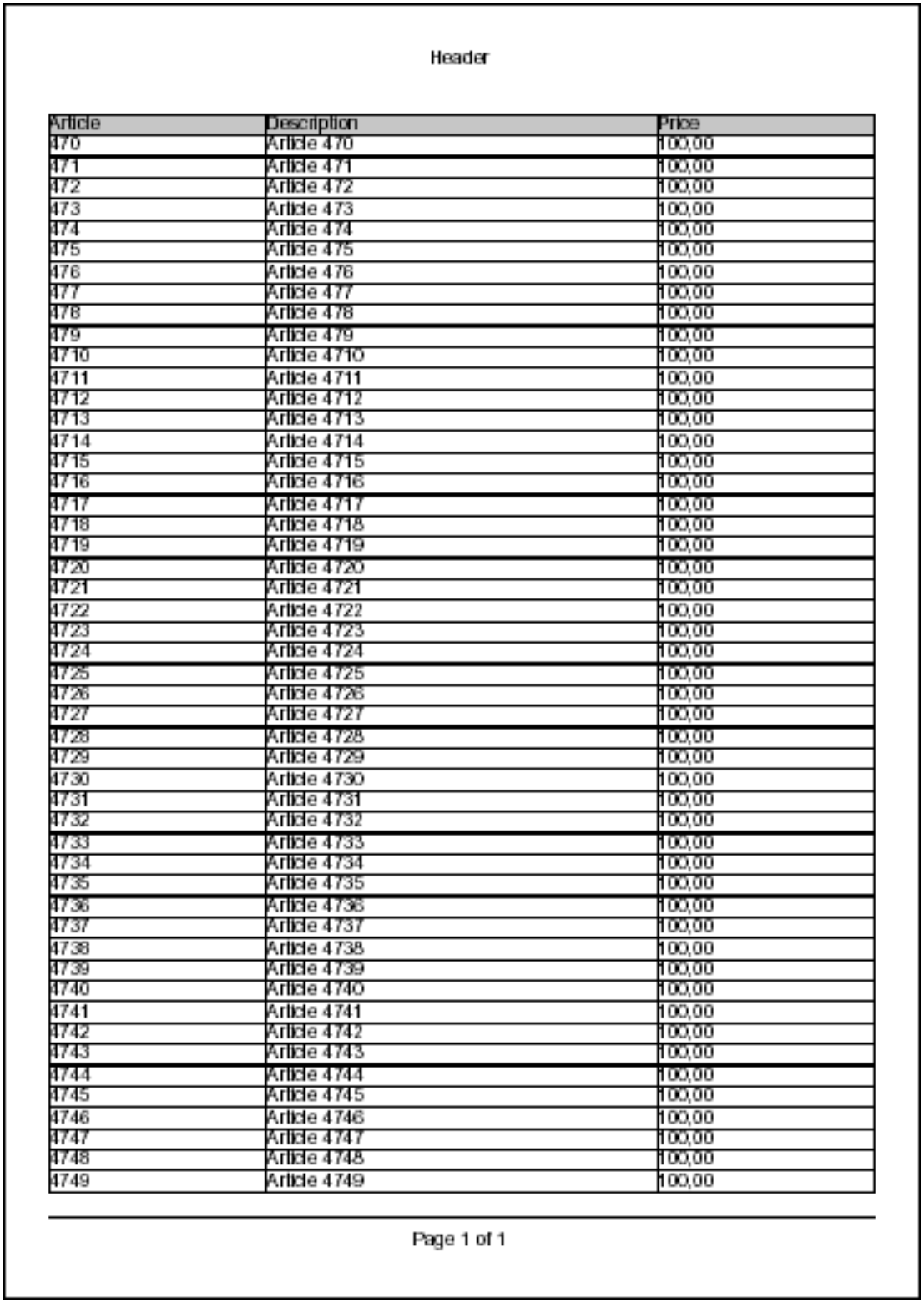
VI Other Controls

The information provided in this part is organized under the following headings:

Lists of Data
Style Control
Image Control
Barcode Control
New Page Control
Controls that you Know from Application Designer HTML

22 Lists of Data

The CISFO:ROWTABLEAREA2 control is used for rendering lists. Have a look at the following example:



Header

Article	Description	Price
470	Article 470	100,00
471	Article 471	100,00
472	Article 472	100,00
473	Article 473	100,00
474	Article 474	100,00
475	Article 475	100,00
476	Article 476	100,00
477	Article 477	100,00
478	Article 478	100,00
479	Article 479	100,00
4710	Article 4710	100,00
4711	Article 4711	100,00
4712	Article 4712	100,00
4713	Article 4713	100,00
4714	Article 4714	100,00
4715	Article 4715	100,00
4716	Article 4716	100,00
4717	Article 4717	100,00
4718	Article 4718	100,00
4719	Article 4719	100,00
4720	Article 4720	100,00
4721	Article 4721	100,00
4722	Article 4722	100,00
4723	Article 4723	100,00
4724	Article 4724	100,00
4725	Article 4725	100,00
4726	Article 4726	100,00
4727	Article 4727	100,00
4728	Article 4728	100,00
4729	Article 4729	100,00
4730	Article 4730	100,00
4731	Article 4731	100,00
4732	Article 4732	100,00
4733	Article 4733	100,00
4734	Article 4734	100,00
4735	Article 4735	100,00
4736	Article 4736	100,00
4737	Article 4737	100,00
4738	Article 4738	100,00
4739	Article 4739	100,00
4740	Article 4740	100,00
4741	Article 4741	100,00
4742	Article 4742	100,00
4743	Article 4743	100,00
4744	Article 4744	100,00
4745	Article 4745	100,00
4746	Article 4746	100,00
4747	Article 4747	100,00
4748	Article 4748	100,00
4749	Article 4749	100,00

Page 1 of 1

The XML form definition is:

```
<cisfo:foppage2 objectclass="PdfTableAdapter" pageheight="29.7cm" pagewidth="21cm" ↩
margintop="1cm" marginbottom="0.5cm" marginleft="1cm" marginright="1cm" ↩
headerheight="1.5cm" footerheight="1.3cm">
    <cisfo:styleclasses2>
       <cisfo:style2 stylename="HL" backgroundcolor="#C0C0C0" bordercolor="#000000" ↩
borderstyle="solid" borderwidth="0.1mm">
        </cisfo:style2>
        <cisfo:style2 stylename="IL" bordercolor="#000000" borderstyle="solid" ↩
borderwidth="0.1mm">
        </cisfo:style2>
    </cisfo:styleclasses2>
    <cisfo:header2>
        <cisfo:rowtextblock2 text="Header" textalign="center">
        </cisfo:rowtextblock2>
    </cisfo:header2>
    <cisfo:footer2>
        <cisfo:hline2>
        </cisfo:hline2>
        <cisfo:vdist2 height="0.3cm">
        </cisfo:vdist2>
        <cisfo:rowtextblock2 text="Page" textalign="center">
            <cisfo:pagenumber2 separator=" of ">
            </cisfo:pagenumber2>
        </cisfo:rowtextblock2>
    </cisfo:footer2>
    <cisfo:body2>
        <cisfo:rowtablearea2 columnwidths="5cm;9cm;5cm" gridprop="lines">
            <cisfo:headline2>
                <cisfo:row2>
                    <cisfo:celltext2 text="Article" stylename="HL">
                    </cisfo:celltext2>
                    <cisfo:celltext2 text="Description" stylename="HL">
                    </cisfo:celltext2>
                    <cisfo:celltext2 text="Price" stylename="HL">
                    </cisfo:celltext2>
                </cisfo:row2>
            </cisfo:headline2>
            <cisfo:repeat2>
                <cisfo:row2>
                    <cisfo:replace2 valueprop="atricle" stylename="IL">
                    </cisfo:replace2>
                    <cisfo:replace2 valueprop="description" stylename="IL">
                    </cisfo:replace2>
                    <cisfo:replace2 valueprop="price" stylename="IL">
                    </cisfo:replace2>
                </cisfo:row2>
            </cisfo:repeat2>
        </cisfo:rowtablearea2>
    </cisfo:body2>
</cisfo:foppage2>
```

Below the ROWTABLEAREA2 control, you see an arrangement of:

- CISFO:HEADLINE2 - represents the headline; the headline is repeated on every page of the list.
- CISFO: REPEAT2 - represents the part that is repeated as many times as items of the list are available.

The data object representation's class looks as follows:

```
public class PdfTableAdapter
    implements IFOPDataObject
{

    // default constructor
    public PdfTableAdapter()
    {
    }

    public PdfTableAdapter(PDFFOPPreviewOnly previewObject)
    {
        for (int i=0; i<50; i++)
        {
            LinesItem li = new LinesItem();
            li.setAtricle("47" + i);
            li.setDescription("Article 47" + i);
            li.setPrice("100,00");
            m_lines.add(li);
        }
    }

    // ---------------------------------------------------------------------
    // property access
    // ---------------------------------------------------------------------

    // class >LinesItem<
    public class LinesItem
    {
        // property >atricle<
        String m_atricle;
        public String getAtricle() { return m_atricle; }
        public void setAtricle(String value) { m_atricle = value; }

        // property >description<
        String m_description;
        public String getDescription() { return m_description; }
        public void setDescription(String value) { m_description = value; }

        // property >price<
        String m_price;
        public String getPrice() { return m_price; }
        public void setPrice(String value) { m_price = value; }
    }

    // property >lines<
```

```
    GRIDCollection m_lines = new GRIDCollection();
    public GRIDCollection getLines() { return m_lines; }
    ...
    ...
    ...
}
```

23 Style Control

- Properties 146

There is a variety of style parameters that can be defined with each control representing a text block: background color, foreground color, margins, padding definitions, borders, etc.

You either can define this parameter with every control or you can define explicit styles. By referencing the name of the style from the controls, the style parameters are taken over automatically. Of course you can still override style definitions locally.

The CISFO:STYLE2 control represents a certain style. It is placed into the CISFO:STYLECLASSES2 container, directly below the page control (CISFO:FOPPAGE2). In the FO form, you refer to this style by using the control's name (property `stylename`).

Properties

Basis			
stylename	Name for the style. Must be unique inside the form. Other controls reference this style name in order to take over the style's rendering information.	Obligatory	
backgroundcolor	This attribute sets the background color of an element, either a andlt;colorandgt; value or the keyword "transparent", to make the underlying colors shine through.	Optional	#000000 #808080 #FFFFFF #FF0000
columnsspanned	A positive integer, specifying the number of columns spanned.	Optional	2 3
color	Color in which the element is output.	Optional	#000000 #808080 #FFFFFF #FF0000
rowsspanned	A positive integer, specifying the number of rows taht this cell spans.	Optional	2 3
textalign	The alignment of the text. Valid values are 'center', 'left' or 'right'.	Optional	left center right

textdecoration	Different values for printing the text, so as underline, overline or line-through.	Optional	overline line-through underline
Border			
bordercolor	Color for all borders (bottom, top, left, right). Values are any valid color specification.	Optional	#000000 #808080 #FFFFFF #FF0000
borderstyle	This attribute specifies the line style of all box's border. Some valid values are "solid", "double" or "dashed".	Optional	solid none
borderwidth	The border width attribute specifies the width of all borders. Some valid values are "thin", "medium", "thick" or a concrete size (e.g. "2pt").	Optional	1pt 5pt 1mm 5mm
bordertopcolor	Color for the border on the top. Values are any valid color specification.	Optional	#000000 #808080 #FFFFFF #FF0000
bordertopstyle	This attribute specifies the line style of a box's top border. Some valid values are "solid", "double" or "dashed".	Optional	solid none
bordertopwidth	The border width attribute specifies the width of the top border. Some valid values are "thin", "medium", "thick" or a concrete size (e.g. "2pt").	Optional	1pt 5pt 1mm 5mm
borderleftcolor	Color for the border on the left side. Values are any valid color specification.	Optional	#000000 #808080 #FFFFFF

			#FF0000
borderleftstyle	This attribute specifies the line style of a box's left border. Some valid values are "solid", "double" or "dashed".	Optional	solid none
borderleftwidth	The border width attribute specifies the width of the left border. Some valid values are "thin", "medium", "thick" or a concrete size (e.g. "2pt").	Optional	1pt 5pt 1mm 5mm
borderbottomcolor	Color for the border on the bottom. Values are any valid color specification.	Optional	#000000 #808080 #FFFFFF #FF0000
borderbottomstyle	This attribute specifies the line style of a box's bottom border. Valid values are 'solid', 'double' or 'dashed'.	Optional	solid none
borderbottomwidth	The border width attribute specifies the width of the bottom border. Valid values are "thin", "medium", "thick" or a concrete size (e.g. "2pt").	Optional	1pt 5pt 1mm 5mm
borderrightcolor	Color for the border on the right side. Values are any valid color specification.	Optional	#000000 #808080 #FFFFFF #FF0000
borderrightstyle	This attribute specifies the line style of a box's right border. Some valid values are "solid", "double" or "dashed".	Optional	solid none
borderrightwidth	The border width attribute specifies the width of the right border. Some valid values are "thin", "medium", "thick" or a concrete size (e.g. "2pt").	Optional	1pt 5pt 1mm

			5mm
Font			
fontfamily	This attribute specifies a priorized list of font family names and/or generic family names. To deal with the problem that a single font may not contain glyphs to display all the characters in a document, or that not all fonts are available on all systems, this property allows authors to specify a list of fonts, all of the same style and size, that are tried in sequence to see if they contain a glyph for a certain character. This list is called a "font set".	Optional	Helvetica Times Courier Symbol ZapfDingbats
fontsize	This attribute describes the size of the font when set as "solid". The font size corresponds to the em square, a concept used in typography. Note that certain glyphs may bleed outside their em squares. Some valid values are "xxsmall", "medium", "smaller" or a concrete size, e.g. "110%".	Optional	8pt 10pt 16pt
fontstyle	This attribute requests normal (sometimes referred to as "roman" or "upright"), italic, and oblique faces within a font family. Valid values are "normal", "italic" and "oblique".	Optional	normal italic oblique
fontvariant	In a small-caps font, the glyphs for lowercase letters look similar to the uppercase ones, but in a smaller size and with slightly different proportions. The "font-variant" property requests such a font for bicameral (having two cases, as with Roman script). This property has no visible effect for scripts that are unicameral (having only one case, as with most of the world's writing systems). Valid values are "normal" and "small-caps".	Optional	normal small-caps
fontweight	This attribute specifies the weight of the font. Some valid values are 'normal', 'bold', 'bolder', 'lighter', '100' or '900', where 'normal' is the same as '400', 'bold' is the same as '700'.	Optional	normal bold 100 200 300 400 500 600 700

			800 900
Margin			
margin	Sets the margin of a box. Negative values for margin properties are allowed, but there may be implementation-specific limits.	Optional	1pt 5pt 1mm 5mm
margintop	Sets the top margin of a box. Negative values for margin properties are allowed, but there may be implementation-specific limits.	Optional	1pt 5pt 1mm 5mm
marginleft	Sets the left margin of a box. Negative values for margin properties are allowed, but there may be implementation-specific limits.	Optional	1pt 5pt 1mm 5mm
marginbottom	Sets the bottom margin of a box. Negative values for margin properties are allowed, but there may be implementation-specific limits.	Optional	1pt 5pt 1mm 5mm
marginright	Sets the right margin of a box. Negative values for margin properties are allowed, but there may be implementation-specific limits.	Optional	1pt 5pt 1mm 5mm
Padding			
padding	Specifies the width of the padding of a block-area or inline-area. Unlike margin properties, values for padding properties cannot be negative.	Optional	1pt 5pt 1mm

			5mm
paddingtop	Specifies the width of the padding on the top of a block-area or inline-area. Unlike margin properties, values for padding properties cannot be negative.	Optional	1pt 5pt 1mm 5mm
paddingleft	Specifies the width of the padding on the left of a block-area or inline-area. Unlike margin properties, values for padding properties cannot be negative.	Optional	1pt 5pt 1mm 5mm
paddingbottom	Specifies the width of the padding on the bottom of a block-area or inline-area. Unlike margin properties, values for padding properties cannot be negative.	Optional	1pt 5pt 1mm 5mm
paddingright	Specifies the width of the padding on the right of a block-area or inline-area. Unlike margin properties, values for padding properties cannot be negative.	Optional	1pt 5pt 1mm 5mm

24 Image Control

- Properties 154

The CISFO:IMAGE2 control outputs an image. Use property `image` to enter the image's URL in a static way. In order to derive the URL at runtime, use property `valueprop`. In both cases, the URL can either be relative (URL points to an image that is kept within your web application) or absolute (absolute directory path). Do not use absolute URLs if you plan to deploy your web application within a cluster.

The Application Designer demo workplace contains a demo on image URLs. Choose **Release 1.3.0** and then **CISFO2 Image include**. You can call the development workplace via the following link (standard installation):

http://localhost:51000/cis/HTMLBasedGUI/workplace/demo.html

Properties

Basic			
image	The URL of the image.	Optional	
valueprop	The name of the property from that the URL is derived at runtime.	Optional	
width	The width of the image control.	Optional	1pt 12pt 0.5in 1in 1mm 5mm 1cm 1.5cm
height	The height of the image control.	Optional	1pt 12pt 0.5in 1in 1mm 5mm 1cm 1.5cm

25 Barcode Control

- Properties 156

The CISFO:BARCODE2 control represents a barcode. The text/number of a barcode is derived from a property of the data object that is associated with the page. There are diverse types of barcodes with diverse parameters that are supported.

The barcode internally uses the Krysalis open source framework (*http://www.krysalis.org/*).

Properties

Basic			
valueprop	Name of adapter property that provides the value that is translated into a corresponding barcode output. Dependent on the barcode type you might only be able to return a limited set of characters as value (e.g. "only number"). The valid character codes are listed in the documentation for the barcodes - please read details there.	Obligatory	
barcodetype	Barcode type that is used for creating the barcode output. There are certain different industry standards that might be used for rendering the barcode.	Optional	INTERLEAVED2OF5 CODE39 USD3 3OF9 CODABAR USD4 NW7 2OF7 CODE128 EAN_UCC128 UPCA UPCE EAN13 EAN8 POSTNET

height	The height of the barcode. The unit is 'mm', 'cm' or 'pt'. The minmal and maximal height depends from the barcode type you use. Look at the description of the barcodes for details.	Optional	1pt 12pt 0.5in 1in 1mm 5mm 1cm 1.5cm
humandreadable	Definition if the barcode should contain human readbly characters. If set to "bottom" then readable characters are rendered below the barcode lines, if set to "top" then they are rendered above. Define "none" if no characters are to be output as part of the barcode.	Optional	BOTTOM NONE TOP
modulewidth	Width of a narrow element. The minmal and maximal width is dependent from the barcode. Look at the description of the barcodes for exact definition.	Optional	
quitezone	Size of the space left and right of the barcode. The minmal and maximal quite zone is dependent from the barcode. Look at the description of the barcodes for exact definition.	Optional	
Special			
checksummode	The checksum mode determines the behaviour for checksums. The posible values are: "add" for adding a checksum, "auto" for the barcode type's default, "check" for creating a checksum and "ignore" for no processing.	Optional	ADD AUTO CHECK IGNORE
igw	Intercharacter gap width. The intercharacter gap width is dependent from the barcode. Look at the description of the barcodes for exact definition.	Optional	
widefactor	A multiply-factor for the wide element. The wide factor is dependent from the barcode. Look at the description of the barcodes for exact definition.	Optional	
POSTNET			
baselinealignment	Determines the alignment of the line in POSTNET barcodes. Posible values are 'bottom' and 'top', as	Optional	BOTTOM TOP

	consequence short and long lines are aligned correspondingly.		
shortbarheight	The height of the short bars.	Optional	
tallbarheight	The height of the tall bars.	Optional	

26 New Page Control

- Properties 160

The CISFO:NEWPAGE2 control represents a page break. It is placed below the page control (CISFO:FOPPAGE2).

Properties

The CISFO:NEWPAGE2 does not have any properties.

27 Controls that you Know from Application Designer HTML

There are a couple of further controls that you already know from the HTML part of Application Designer:

- CISFO:TEXTGRID2 (for TEXTGRID)
- CISFO:REPORT2 (for REPORT)
- and others.

The rendering of these controls and the binding to the data object is 1:1 equivalent to the rendering and binding of their HTML counterparts.

VII Printing

28 Printing

There are two commonly-used ways for producing paper printouts:

- Create PDF via PDF/FOP processing and make it available in the user's browser. This is what the example in the section *The First PDF Document* does.
- Directly print FOP to a server side printer. This is what this section is about.

Server side printing is done via the interface `IPDFFOPService`:

```
public interface IPDFFOPService
{
    public String generateXSLFO();
    public byte[] generatePDF();
    public void print(String printerName) throws FOPPrintException;
    public byte[] generatePDFfromFOXML();
    public ILog getLog();
    public Protocol getProtocol();
    public String getTranslationFileName();
}
```

By using the method `print(...)`, you can send a document directly to a printer.