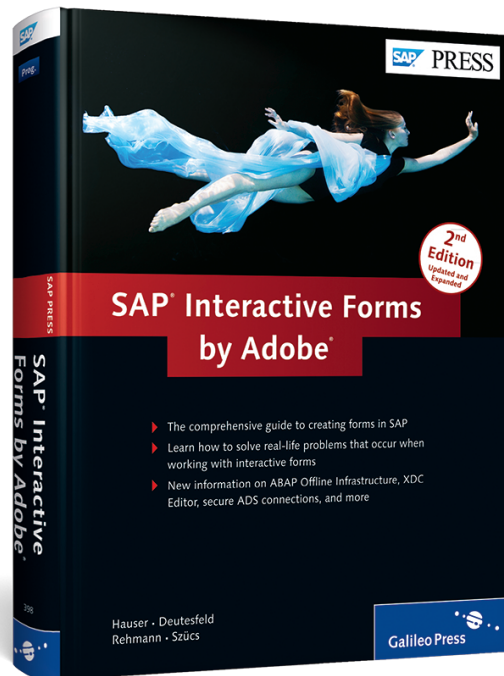


Jürgen Hauser, Andreas Deutesfeld, Stephan Rehmann,
and Thomas Szücs

SAP® Interactive Forms by Adobe®



 Galileo Press

Bonn • Boston

Contents at a Glance

1	Introduction	17
2	Use of SAP Interactive Forms by Adobe	23
3	Installation and Configuration	63
4	Interface and Form Context	115
5	Creating Form Templates	175
6	Form Output	229
7	Advanced Form Template Design	289
8	Integration with Web Dynpro ABAP	373
9	Internal Service Request	501
10	ABAP Offline Infrastructure	605
11	ABAP PDF Object	635
12	Offline Scenarios via Web Services	659
13	Advanced Installation and Configuration	695

Contents

1	Introduction	17
2	Use of SAP Interactive Forms by Adobe	23
2.1	PDF, PDF-Based Print Forms, and Interactive PDF Forms	23
2.1.1	PDF File Format	23
2.1.2	PDF-Based Print Forms	24
2.1.3	Interactive PDF Forms	27
2.1.4	Accessibility of PDF Forms	28
2.1.5	Using Interactive PDF Forms	29
2.2	Adobe Reader and Adobe Acrobat Family	29
2.2.1	Comparing Adobe Reader and Acrobat Professional	30
2.2.2	Selecting the Appropriate Adobe Reader Version	33
2.2.3	Useful Settings of Adobe Reader	34
2.3	Examples of PDF-Based Print Forms and Interactive PDF Forms ...	34
2.3.1	Example for Print Output	34
2.3.2	Creating an Interactive PDF Form	38
2.4	Using Interactive Forms in Business Processes	40
2.4.1	Online Scenarios	41
2.4.2	Offline Scenarios	41
2.4.3	Combining Online and Offline Scenarios	43
2.4.4	When Do You Use Online and Offline Scenarios?	43
2.4.5	Notes on the Use of Interactive Forms	45
2.5	Software Components and Architecture	46
2.5.1	Adobe LiveCycle Designer	47
2.5.2	Adobe Document Services	52
2.5.3	Interaction of Components	56
2.6	Summary	60
3	Installation and Configuration	63
3.1	Prerequisites	63
3.1.1	Credentials and Password for SAP Interactive Forms by Adobe	64

3.1.2	Adobe LiveCycle Designer 8.2	64
3.2	Basic Configuration	64
3.2.1	ADSUSER	65
3.2.2	ADSCaller	68
3.2.3	Web Service Test	68
3.3	Scenario-Dependent Configuration	72
3.3.1	RFC Destination	73
3.3.2	Destination and ICF Service	75
3.3.3	Web Service Proxy	82
3.3.4	ReaderRights Credential	85
3.3.5	FPCONNECT	88
3.4	Configuration for Parallelization	89
3.4.1	Web Service for Parallelization	91
3.4.2	Web Service for Monitoring	93
3.4.3	Connection Test for the Two Web Services	94
3.4.4	Assigning the Logical Port to the ADS Connection	95
3.5	Optional Configuration	96
3.5.1	Licensing	96
3.5.2	Credentials, Trusted Anchors, Certificate Revocation Lists	98
3.5.3	Service Properties	101
3.5.4	Number of Parallel Print Jobs	103
3.5.5	Fonts	105
3.5.6	XDCs	105
3.6	Error Analysis	105
3.7	Using ADS	107
3.7.1	Monitoring and Performance Tracing	107
3.7.2	Log Configurator	108
3.8	Installation of Adobe LiveCycle Designer	110
3.9	Automatic Configuration via CTC	110
3.10	ADS Hub	111
3.11	Summary	114

4 Interface and Form Context 115

4.1	Structure of a Form	115
4.2	Interface of a Form	116
4.2.1	ABAP Dictionary-Based Interface	119

4.2.2	XML Schema-Based Interface	129
4.2.3	Smart Forms-Compatible Interface	131
4.2.4	Documentation of a Form Interface	132
4.3	Context of a Form	134
4.3.1	Creating a Form	134
4.3.2	Structure of a Form	137
4.3.3	Structures	141
4.3.4	Data Nodes	144
4.3.5	Text Nodes	146
4.3.6	Graphic Nodes	153
4.3.7	Address Nodes	157
4.3.8	Alternatives	160
4.3.9	Tables or Loops	163
4.3.10	Single Record Nodes	167
4.3.11	Conditions	170
4.3.12	Folders	171
4.4	Documentation of a Form	172
4.5	Summary	173

5 Creating Form Templates 175

5.1	Adobe LiveCycle Designer	175
5.1.1	Menu Bar and Toolbar	177
5.1.2	Script Editor	177
5.1.3	Hierarchy and Data View	178
5.1.4	Library and Object Palette	180
5.1.5	Form Design Area	182
5.2	The Basic Principles of the Form Template Design	182
5.2.1	Getting Started	184
5.2.2	Using Text Fields	187
5.2.3	Formatting Texts	193
5.3	Data Binding for Form Fields	195
5.3.1	Defining the Data Binding	195
5.3.2	Using Preview Data	198
5.4	Structuring Form Templates with Subforms	199
5.4.1	Adding a Hierarchy	200
5.4.2	Extending the Form Template	203
5.5	Using Master Pages and Rich Text Fields	209
5.5.1	Customizing the Master Page	209

- 5.5.2 Rich Text Fields 210
- 5.5.3 Alternative Method for Adding the Footer 213
- 5.5.4 Activating Forms 213
- 5.6 Reusing Form Objects 213
 - 5.6.1 Creating Your Own Libraries 214
 - 5.6.2 Adding Form Objects 215
 - 5.6.3 Using Form Objects 216
- 5.7 Implicit Data Binding 217
 - 5.7.1 Creating Copies of Forms 217
 - 5.7.2 Implementing the Implicit Data Binding 218
- 5.8 Useful Functions in Adobe LiveCycle Designer 222
 - 5.8.1 Customizing the Toolbars 222
 - 5.8.2 Object Editor 222
 - 5.8.3 Spelling 223
 - 5.8.4 Hyphenation 223
 - 5.8.5 Managing Multiple Form Fields 225
 - 5.8.6 Managing Data Bindings 225
 - 5.8.7 Adobe Reader Target Version for a Form Template 226
 - 5.8.8 Defining the Tab Order 227
 - 5.8.9 PDF Structure 228
- 5.9 Summary 228

6 Form Output 229

- 6.1 Print Program 230
 - 6.1.1 Data Selection 230
 - 6.1.2 Determining the Generated Function Module 232
 - 6.1.3 Opening a Print Job 234
 - 6.1.4 Calling the Generated Function Module 237
 - 6.1.5 Closing the Print Job 239
 - 6.1.6 Form Output 240
- 6.2 Spool System 244
- 6.3 Device Types for the Output 246
- 6.4 Special Output Scenarios 249
 - 6.4.1 Archiving 249
 - 6.4.2 Returning Documents 250
- 6.5 Error Analysis 254
 - 6.5.1 Trace and PDF with Additional Information 254

6.5.2	Activating and Retrieving the Trace in the Print Program	258
6.5.3	More Detailed Error Messages	258
6.5.4	ICF Recorder	258
6.6	Performance Optimization Through Bundling	263
6.6.1	Bundling Forms	263
6.6.2	Activating Bundling	264
6.6.3	Deactivating Bundling	265
6.6.4	Bundling Example	266
6.6.5	Bundling the PDF Return	269
6.7	Parallelization of Print Jobs	270
6.7.1	Activating Parallelization in the Print Program	272
6.7.2	Monitoring Requests	273
6.7.3	Tracing Using the SOA Manager	276
6.8	Translating Forms	281
6.8.1	Translating Short Texts	282
6.8.2	Translating Long Texts	283
6.8.3	Output of a Translated Form	284
6.9	XFP Output	285
6.10	Summary	287

7 Advanced Form Template Design 289

7.1	Advanced Techniques for Print Forms	289
7.1.1	Preparations	289
7.1.2	Floating Fields	290
7.1.3	Using Tables	292
7.1.4	Page Breaks	296
7.1.5	Subform Set	300
7.1.6	Using Patterns for Display Formatting	302
7.1.7	Conditional Page Breaks	307
7.1.8	Form Output with Duplex Printing	309
7.1.9	Output of the Completed Form Template	311
7.2	Interactive PDF Forms	312
7.2.1	Preparations	313
7.2.2	From PDF-Based Print Forms to Interactive PDF Forms ...	317
7.2.3	Form Fields of Interactive PDF Forms	330
7.2.4	Dynamic Properties	338

7.3	Using Script Programs	341
7.3.1	First Script Program	342
7.3.2	Tips for Debugging Script Programs	346
7.3.3	Calculations via Script Programs	348
7.3.4	Input Validation via Script Programs	349
7.3.5	Controlling the Ready-for-Input Status and Visibility of Form Fields	350
7.3.6	Determining the Recipient Address for an Email Message at Runtime	352
7.3.7	Querying Reader Version Information	353
7.3.8	Access to Data Nodes and Form Fields	356
7.3.9	Dynamic Tables	357
7.4	Using Barcodes	361
7.4.1	Properties of Barcodes	362
7.4.2	Notes on the Output of Barcodes with Zebra Label Printers	363
7.5	Tips for Increasing Performance	363
7.5.1	Merging Static Texts	363
7.5.2	Resolution of Used Images	364
7.5.3	Ensuring Correct Data Bindings	364
7.5.4	Nesting Depth of Subforms	364
7.5.5	Gradient Fills for PostScript Level 2	365
7.5.6	Using Script Programs	365
7.5.7	Enforcing Strict Scoping Rules in JavaScript	365
7.5.8	Limiting the Volume of Transferred Data for the Form Output	366
7.5.9	Using Fonts	366
7.6	Introduction to Accessible PDF Forms	366
7.6.1	Creating Accessible PDF Forms	367
7.6.2	Tips for Creating Accessible PDF Forms	368
7.7	Additional Information	369
7.7.1	Additional Information on the Internet	369
7.7.2	Adobe Designer Scripting Reference	369
7.7.3	XFA	370
7.7.4	Acrobat JavaScript Reference	370
7.8	Summary	371

8 Integration with Web Dynpro ABAP 373

8.1	Web Dynpro ABAP	374
8.1.1	Initial Overview	374
8.1.2	Step by Step: Hello World	378
8.2	Scenarios for the Integration of Interactive Forms	385
8.2.1	Print Scenario	385
8.2.2	Online Scenario	387
8.2.3	Offline Scenario	388
8.3	The InteractiveForm UI Element	389
8.4	Interaction with Form Builder	391
8.4.1	Using Form Templates	391
8.4.2	The pdfOnly Case	404
8.5	Implementing Scenarios Using Web Dynpro ABAP	404
8.5.1	Print Scenario	405
8.5.2	Online Scenario	410
8.5.3	Offline Scenario	469
8.5.4	Migrating ACF to ZCI	497
8.5.5	Specialized Topics	498
8.6	Summary	500

9 Internal Service Request 501

9.1	Introduction	502
9.1.1	Concepts in the ISR Framework	502
9.1.2	Prerequisites	504
9.1.3	What Can Be Configured and Where Can It Be Programmed?	505
9.2	SAP NetWeaver Portal Scenarios	506
9.2.1	Grouping of the Provided Scenarios	507
9.2.2	Calling the Request Form in the Portal	507
9.2.3	Applicant Role	508
9.2.4	Approver Role	514
9.2.5	Processor Role	517
9.2.6	Workflow or Worklist?	518
9.3	Getting Started—Creating a Plain Scenario	518
9.3.1	Preliminary Considerations for a New ISR Scenario	519
9.3.2	ISR Wizard	520

9.4	Customizing and Programming a Scenario	529
9.4.1	General Details	529
9.4.2	Application and Version	532
9.4.3	Notification Type	533
9.4.4	Entry Type	534
9.4.5	Characteristics	535
9.4.6	Activities and Automatic Updates in Notifications	537
9.4.7	The Processor's Activities from the Form	541
9.4.8	Processor Determination (BAI QISR3) and Workflow ...	543
9.4.9	Service Costs and BAI QISR2	548
9.4.10	Testing the Scenario	550
9.5	Request Forms as Adobe Forms	551
9.5.1	Specific Properties	551
9.5.2	Structure of the Interface and Form Context	552
9.5.3	Generating Adobe Forms from Characteristics	554
9.5.4	Generating New Characteristics from the Form	556
9.5.5	ISR Library—Special Screen Elements	556
9.6	Form Flow Logic	564
9.6.1	Data Structures within the ISR Framework	567
9.6.2	Default Values for the MODE, USER_COMMAND, and FORM_VIEW Parameters	568
9.6.3	Programming Examples	568
9.6.4	Reading and Setting the Special Request Fields in BAI Methods	572
9.6.5	Error Handling During Checks	574
9.6.6	Controlling the Field Layout	574
9.7	ISR Architecture	575
9.7.1	ISR as a Toolset	575
9.7.2	Web Dynpro ABAP Component and How to Use It	577
9.7.3	Usage Options in SAP NetWeaver Portal	580
9.7.4	ISR_PROCESS_EVENT RFC	588
9.7.5	Additional Interfaces and Auxiliary Functions	590
9.8	ISR in the Application Basis (ABA)	592
9.8.1	Basic Concept	592
9.8.2	Defining a Scenario	594
9.8.3	Form Flow Logic	596
9.8.4	Architecture	600
9.9	Summary	603

10 ABAP Offline Infrastructure 605

10.1	Offline Scenarios Using Email Messages	605
10.2	Address Updating in the Flight-Booking Model as a Sample Scenario	606
10.3	Configuring the Email Inbound	608
10.4	Using ABAP Offline Infrastructure	609
10.4.1	Creating the Offline Handler	609
10.4.2	Interface and Context	610
10.4.3	Creating the Form Template	612
10.4.4	Generating and Sending the Form	618
10.4.5	Processing Forms Received	623
10.5	Testing the Offline Scenario	628
10.6	Summary	633

11 ABAP PDF Object 635

11.1	Overview	635
11.2	Instantiating the PDF Object	636
11.3	Creating a PDF Document	639
11.3.1	Retrieving a Form	640
11.3.2	Transferring a Form via a Reference	641
11.3.3	Creating an Interactive PDF Document	642
11.3.4	Usage Rights	645
11.4	Processing an Interactive PDF Document	647
11.4.1	Filling Out an Interactive PDF Document	648
11.4.2	Data Extraction	650
11.4.3	Validation of a Digital Signature	651
11.5	Certifying PDF Documents	653
11.6	Job Profiles	657
11.7	Additional Methods of the PDF Object	657
11.8	Summary	658

12 Offline Scenarios via Web Services 659

12.1	Overview	659
12.2	Using WSDL Data Connections	660
12.2.1	Creating a Web Service Based on a BAPI	660

12.2.2	Creating a Form Template	668
12.2.3	Testing the Form Template	674
12.3	Using the SOAP Object in JavaScript	678
12.3.1	Creating a Web Service on the Basis of a Function Module	678
12.3.2	Creating a Form Template	684
12.3.3	Implementing the Web Service Calls	685
12.3.4	Testing the Form Template	690
12.4	Tips for Using Web Services	692
12.4.1	Calling the Web Service in the Click Event	692
12.4.2	Calling the Web Service Automatically When Opening the Form	692
12.4.3	Calling Multiple Web Services Using a Button	692
12.5	Summary	693
13 Advanced Installation and Configuration		695
13.1	Configuring a Secure Connection to ADS	695
13.1.1	Setting Up Your Own Certification Authority	696
13.1.2	Creating an SSL Client Identity in the ABAP System	698
13.1.3	Configuring the Java System and ADS	709
13.1.4	Creating a Secure Connection in the ABAP System	727
13.2	Managing and Creating Job Profiles	730
13.2.1	Calling the Job Profile Management	730
13.2.2	Managing Job Profiles	731
13.2.3	Creating a New Job Profile	731
13.3	Installing and Using the XDC Editor	734
13.4	Summary	735
The Authors		737
Index		739

This chapter introduces SAP Interactive Forms by Adobe and discusses various application scenarios. It also specifies the software components used and the architecture for integration with SAP NetWeaver.

2 Use of SAP Interactive Forms by Adobe

This chapter introduces you to the basic terminology associated with the use of SAP Interactive Forms by Adobe, such as Portable Document Format (PDF)-based print forms and interactive forms. A classification of online and offline application scenarios is outlined for interactive forms; additionally, you are provided with support for deciding in which cases you should use interactive forms. Furthermore, this chapter compares Adobe Acrobat® and Adobe Reader and explains why it is sufficient to deploy Adobe Reader for the use of SAP Interactive Forms by Adobe. Finally, this chapter introduces the software components of Adobe, Adobe LiveCycle Designer, and Adobe Document Services and their integration with the SAP NetWeaver stack.

2.1 PDF, PDF-Based Print Forms, and Interactive PDF Forms

The following sections detail PDFs and the two main concepts of this book—PDF-based print forms and interactive PDF forms.

2.1.1 PDF File Format

In the early 1990s, Adobe Systems launched the first version of PDFs. At that time, Adobe was well known for its PostScript printer language, Adobe Illustrator (a vector-based drawing tool), and Adobe Photoshop (image editing). The idea behind PDF was to develop a platform-independent file format that enabled high-quality exchange and printout of documents. The primary focus was on the exchange of documents and not on an editable document format.

The specification of PDF is published; therefore, everyone can get an overview of this file format. The International Organization for Standardization (ISO) lists the

PDF format under Standard 32000. The PDF/A format for archiving documents is already an ISO standard; however, it has not yet become a part of SAP Interactive Forms by Adobe.

PDFs enable you to flexibly describe the structure of individual pages of a document. PDF is a flat file format and knows only the concept of a page as the structuring element. Compared to PostScript, it is a pure descriptive language and therefore contains no programming language constructs.

The most commonly known use of PDF is high-quality print output: A document is converted into a PDF file and can then be displayed or printed on different platforms and operating systems via Adobe Reader. The presentation of a document as a PDF ensures that the document always has the same layout regardless of the user's operating system or platform. The user requires only a PDF display tool, for example, Adobe Reader.

In addition to the pure display and printout, you can also use PDF documents for data entry. To do this, forms that are based on PDF contain interactive form fields that enable the user to interact with the PDF document. This way, you can enter data and execute calculations that are based on this data directly during entry. For this reason, Adobe Reader supports JavaScript as one of the scripting languages.

2.1.2 PDF-Based Print Forms

Forms are a well-known structuring method for data output (for example, a telephone bill) or data entry (for example, official application forms). One of the essential aspects of forms is a uniform layout that enables and supports the recognition of forms. The structure of each telephone bill is identical and the customer knows where to find certain information on the bill—just like annual tax forms.

This feature requires the form designer to use a clear and comprehensible layout for the form. The layout describes what information is located in which position on the form. It must also specify how to handle a large number of data records, which is usually done by distributing them throughout multiple pages.

The task of a form designer is to create form templates. Based on the form template, a specific form is generated and—in most cases—filled with data during the generation process. Consequently, the form template plays a central role in all form-based processes. Therefore, the form designer must be optimally supported in his task with a user-friendly and intuitive tool.

Forms can be output both on paper and electronically; however, the electronic output has significant advantages (for example, electronic transmission or archiving). For this reason, SAP Interactive Forms by Adobe enables you to output forms in different printer languages or as PDF files.

Supported Printer Languages

The printer languages supported include Adobe PostScript® (PS), Printer Command Language® (PCL) developed by Hewlett-Packard®, and Zebra Programming Language® (ZPL) for printing barcodes on Zebra label printers. Printers must be able to process one of the aforementioned printer languages to ensure that they are supported by SAP Interactive Forms by Adobe.

Supported Printer Languages

SAP Interactive Forms by Adobe supports the following printer languages:

- ▶ Adobe PostScript Levels 2 and 3, where Level 3 is the default setting. To create Level 2, the corresponding XML Device Configuration (XDC) file must be copied and customized (see Section 3.5.6, XDCs).
- ▶ PCL 5 in monochrome and color.
- ▶ ZPL-II is supported in different resolutions, for example, 200, 300, and 600 Dots per Inch (DPI).

The print output and the PDF file are created based on Adobe XML Forms Architecture (XFA) technology, which is a specification published by Adobe. This specification describes the XML-based structure of a form template and the behavior of display and conversion programs at runtime. The XFA specification is a very good source of information and can be downloaded from the Adobe website (http://partners.adobe.com/public/developer/xml/index_arch.html).

Support of Barcodes

Another important area supported by SAP Interactive Forms by Adobe is barcodes. A barcode encodes information that can be read by a scanner and then processed. You can find barcodes on virtually every product package read at a checkout counter. You can determine the scanned product and its price based on the information encoded in the barcode.

SAP Interactive Forms by Adobe supports most of the common barcodes. Currently, there are 34 different barcodes, including EAN8, EAN13, and Code 128, for example.

There are also barcodes that can be supported only by the printer itself. If this is the case, the output is done as a PDF file, and a gray rectangle will be displayed instead of the barcode. It is also possible to exclusively create a barcode and print it on a barcode printer. Currently, this is supported for Zebra label printers.

There is no way to define barcodes yourself. For barcodes that are not supported, you have to rely on their support in one of the more recent releases.

Simplex and Duplex Printing

The double-sided print output and paper tray selection are supported as of SAP NetWeaver 7.0 Support Package 13. Simplex and duplex printing, that is, single-sided and double-sided printing, are controlled by means of the page sets and master pages. The form designer assigns the individual form pages to master pages to specify whether a page appears on the front side or the back side. Very complex scenarios are possible here (for example, combining simplex and duplex prints in one form) that require skillful use of master pages and conditional page breaks.

Paper Tray Control

The selection for the paper tray control requires multiple configuration steps because the corresponding control codes need to be determined for the printer and assigned to the available paper types. The control codes are contained in the XDC files, and their assignment is carried out by editing the XDC files located on the server.

PDF-Based Print Forms, SAP Smart Forms, and SAPscript

The print output, which is based on PDF-based print forms, provides an alternative to the SAPscript and SAP Smart Forms technologies provided by SAP. There is no full equivalence with regard to functions and properties between the technologies due to technological differences. The option to navigate to a callback of an ABAP function during the layout creation is not possible for PDF-based print forms due to technological reasons.

SAP Smart Forms versus SAP Interactive Forms by Adobe

SAP Note 1009567 describes the functional differences between SAP Smart Forms and SAP Interactive Forms by Adobe.

SAP Interactive Forms by Adobe provides advanced and complex innovations, such as enhanced layout options (for example, rectangles with rounded corners) or support of a scripting language (for instance, implementing calculations and validations in interactive PDF forms). These enhanced options can impact performance and file sizes; for example, shadings affect the output in PCL format.

Another innovation compared to the already-existing technologies is the option to create interactive PDF forms. These are described in greater detail in the following section.

2.1.3 Interactive PDF Forms

PDF documents provide much more than just the known display and print functionality: They can contain interactive form fields. These fields are called “interactive” because the user can make changes by means of the mouse or the keyboard.

Possibilities

Interactive form fields include input fields for text and numbers that are entered via the keyboard. Moreover, you are provided with radio buttons or checkboxes, which are primarily operated using the mouse. In addition, more complex fields, such as the DATE SELECTION field (a calendar appears for simplified selection), list boxes, and dropdown lists are available. List fields or dropdown lists are particularly helpful if the user of the form is supposed to select an input value from a predefined list. The list can display all list entries so that the user does not have to remember all possible input options.

Interactive PDF forms are used for structured data entry by the user. If multiple users enter data, this is called *mass data entry*. In this case, electronic support is especially beneficial because the processing and saving of data can be automated.

Interactive form fields enable the user to interact directly with the document. Data that are stored in the interactive PDF form are changed during the interaction. For integration with electronic processes, you can extract or insert data on the server side. In specific cases, this is also possible at the user's work station, for example, if only the data contained in the PDF form are to be sent by email.

For SAP Interactive Forms by Adobe, the XFA-based technology is used. Here, the form template is stored in XML format and the data contained in the PDF form are stored as an XML data record.

Static and Dynamic PDF Forms

For interactive PDF forms, you can differentiate between static and dynamic PDF forms depending on whether the form layout changes during the interaction or not. An example is dynamic tables in which users can insert new rows or delete old rows. Another example is to hide parts of the form depending on the previous entries. This is possible only for dynamic PDF forms.

For dynamic PDF forms, the form layout is created directly in Adobe Reader. This is done based on the form template and the data contained therein. As of Adobe Reader version 8.1, the display of dynamic forms is considerably faster than in older versions. It is recommended that you use Adobe Reader version 8.1 or higher if you utilize dynamic functions intensively. It is recommended that you use the current Adobe Reader version. Nevertheless, it is possible that the use of frameworks (for example, Web Dynpro frameworks) may result in restrictions or cause delays in the support. If this is the case, the corresponding chapters of this book refer to the relevant SAP note.

2.1.4 Accessibility of PDF Forms

Because you can use interactive PDF forms as a part of user interfaces, the accessibility of PDF forms is of significance. Interactive forms enable you to create accessible PDF forms. Accessible means, for example, that visually handicapped users can also utilize the PDF form. Although Adobe Reader provides a function for reading PDF documents out loud (VIEW • READ OUT LOUD), third-party software, such as JAWS® from Freedom Scientific®, is usually used. Accessibility also means that mobility-impaired users can work with an application or a document (for example, by means of the voice control).

Accessibility impacts the size of the PDF file because additional information needs to be added. This information must be generated on the server during the creation of the PDF document, which also requires extra time. Therefore, you should consider whether a PDF form should always be accessible or not. A user-specific control is one option to avoid additional work. For example, you can use the user settings for the print preview in SAP GUI.

The form layout does not always result in an accessible form. The design tool to form templates supports you in the creation of accessible forms—for example, for the positioning of form fields, which influences the sequence. However, if the layout of the form is so complex that it is difficult to use, it is predictable that accessibility

problems will occur. In such cases, it is extremely helpful to reconsider and revise the form layout.

2.1.5 Using Interactive PDF Forms

You have been introduced to several significant properties of PDF forms:

- ▶ A PDF form may or may not be interactive. If not, it is a PDF-based print form.
- ▶ In addition, a PDF form can be static or dynamic.

These properties are either determined by the individual frameworks that integrate the PDF forms, for example, in Web Dynpro ABAP it is a preset that creates dynamic PDF forms, or are available as parameters in the programming interfaces.

The benefit of interactive PDF forms in electronic business processes in the SAP environment is that you just need Adobe Reader and a Web browser or email program to use them. The wide distribution of Adobe Reader and the common email programs and Web browsers facilitates their use considerably.

2.2 Adobe Reader and Adobe Acrobat Family

Adobe offers two products for displaying PDF files: the free Adobe Reader and the Adobe Acrobat family that is subject to charge. The Adobe Acrobat 9 family comprises three separate products: Adobe Acrobat Standard, Adobe Acrobat Pro, and Adobe Pro Extended.

Note on the Usage of SAP Interactive Forms by Adobe

Adobe Reader is sufficient for the use of SAP Interactive Forms by Adobe! Most notably, it is not necessary to purchase Adobe Acrobat licenses. Adobe Reader is distributed by Adobe Systems and not by SAP; that is, Adobe Reader can be downloaded only from the Adobe website (<http://www.adobe.com/reader>).

Adobe Reader 9.4.2 is used throughout this book. If you use an older version of Adobe Reader, you should update Adobe Reader to this version or a higher version after you have read this chapter.

The fee-based products of the Adobe Acrobat family provide more functionality than the free Adobe Reader. Section 2.2.1, Comparing Adobe Reader and Acrobat Professional, details why it is still sufficient to use Adobe Reader.

Interactive Forms on Mobile Devices

SAP Note 1002905 describes the supported mobile devices and the existing restrictions from the point of view of SAP Interactive Forms by Adobe. Currently, interactive forms are supported by Adobe Reader for Pocket PC.

2.2.1 Comparing Adobe Reader and Acrobat Professional

Adobe Acrobat products provide more functionality than the free Adobe Reader. This functionality includes, for example, the creation of PDF documents and their manipulation. Adobe Acrobat Pro Extended enables the conversion of different 3D formats into a PDF document.

Parts of this functionality are also included in Adobe Reader; however, they are hidden and must first be activated. For their activation, a PDF document must be provided with a corresponding ID, called usage rights, during their creation.

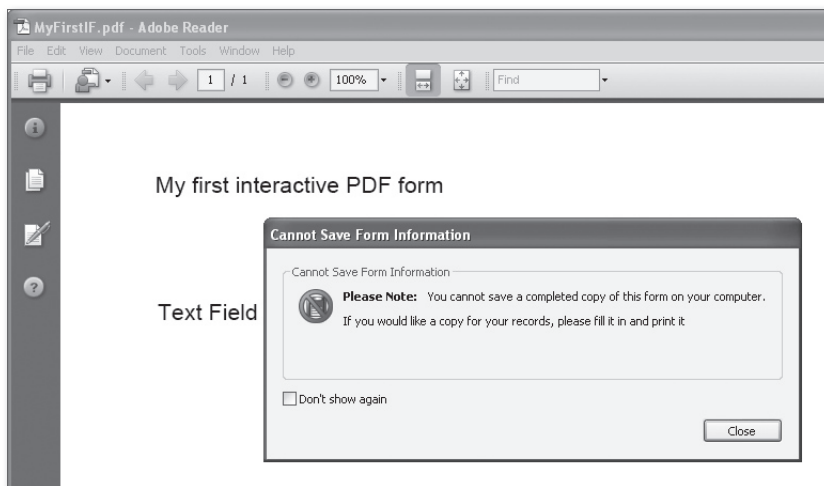


Figure 2.1 Interactive Forms in Adobe Reader (Without Embedded Usage Rights)

Figure 2.1 shows a simple interactive PDF form in Adobe Reader 9.4, which contains interactive form fields. After a change has been made in the first form field,

the dialog shown in the figure appears. It notifies the user that the data entered cannot be saved using Adobe Reader. In Adobe Acrobat, this functionality is always activated.

Usage Rights for Adobe Reader

By embedding usage rights into the PDF itself, you can activate the following additional functions in Adobe Reader:

► **Saving of interactive PDF forms**

This eliminates the behavior of Adobe Reader described earlier, in which data that were entered in an interactive PDF form cannot be saved.

► **Digital signature**

If the PDF form includes a signature field, the user can digitally sign the PDF, provided that this functionality has been activated.

The qualified digital signature required in Germany, for example, is not supported in the standard version of Adobe Reader. This functionality can be added by using a plug-in.

► **Add comments and markups**

This functionality enables you to comment on the opened PDF document. To do this, you are provided with sticky notes that allow you to enter free text (see Figure 2.2). In addition, you can use custom stamps.

Commenting on dynamic PDF forms is not possible presently.

► **Add PDF file attachments**

You can add file attachments to PDF documents. Two types of file attachments are distinguished here: document-wide file attachments and file attachments on specific pages and in specific positions (also referred to as *file attachment comments*). A setting exists for these file attachments that prevents the file attachment from opening at the click of a mouse. This setting can be found under EDIT • PREFERENCES • TRUST MANAGER.

For dynamic PDF forms, you are provided with only document-wide file attachments.

► **Call Web services**

Calling Web services is also a functionality that can only be used with usage rights.

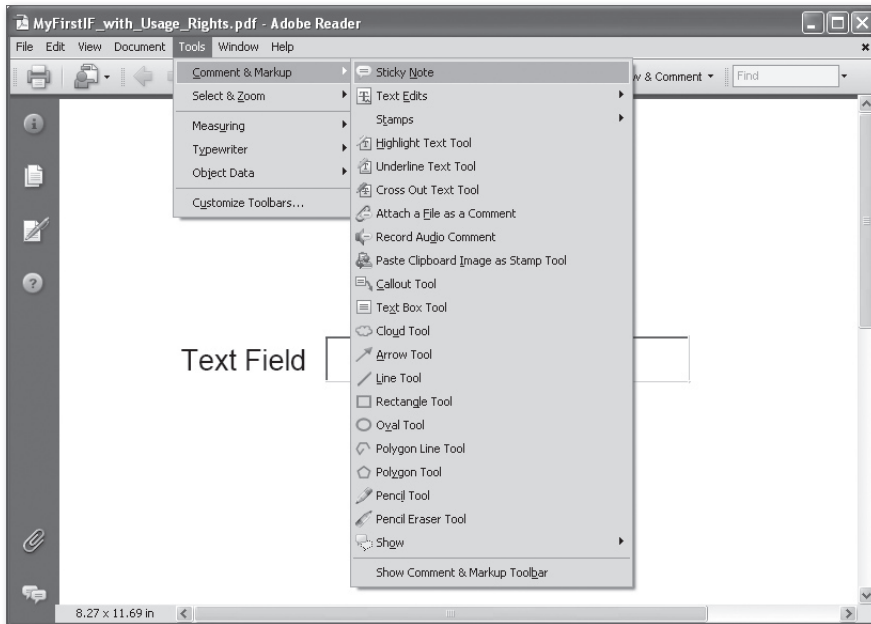


Figure 2.2 Enabled Menu for Comment & Markup

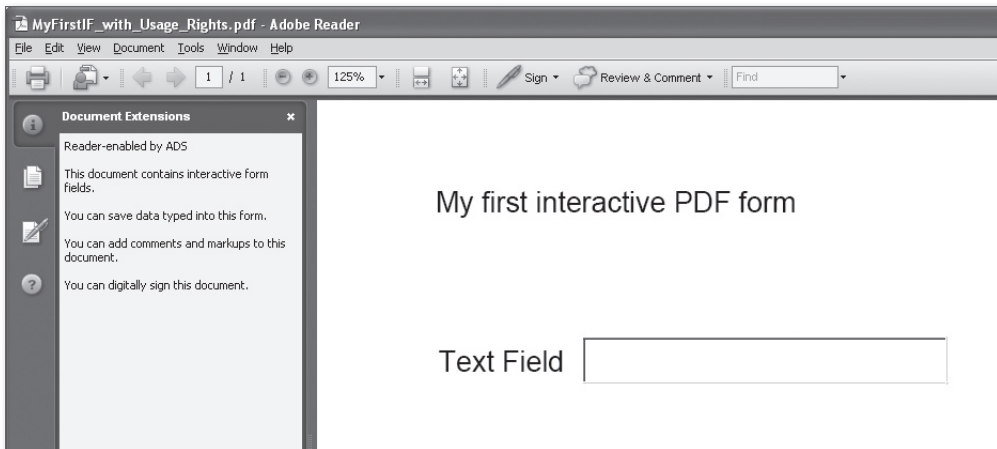


Figure 2.3 Information About the Opened PDF Document

Figure 2.3 shows the same PDF form that was displayed in Figure 2.1. The difference is that usage rights are now embedded. You are provided with additional

information about the currently opened document if you click on the blue I symbol in the top left corner. The information for the example shown in this figure indicates that the document contains interactive form fields and lists the embedded usage rights in descriptive form.

The intention behind SAP Interactive Forms by Adobe is to add usage rights to interactive PDF forms. Therefore, Adobe Reader is sufficient for viewing.

For adding usage rights, you must install a *credential* in SAP NetWeaver together with SAP Interactive Forms by Adobe. This credential is not required for PDF-based print forms. Chapter 3, Installation and Configuration, describes how you can obtain this credential and how it is installed in the SAP NetWeaver stack.

2.2.2 Selecting the Appropriate Adobe Reader Version

New releases of SAP Interactive Forms by Adobe usually include new functions. Because of this, there is a recommended minimum version number for every release of SAP Interactive Forms by Adobe to utilize the full functionality.

For a project that uses SAP Interactive Forms by Adobe, you must always determine the Adobe Reader version(s) to be used. If you use different Adobe Reader versions, it is recommended that you test them (for example, a new version of Adobe Reader may correct errors in older versions).

Selecting the Adobe Reader Version

If you use functionality that was introduced together with a specific SAP NetWeaver support package, you must consider the lowest version number. New Adobe Reader versions are downward compatible; that is, you can always use a newer version of Adobe Reader.

- ▶ If you use Adobe LiveCycle Designer 8.1 and SAP NetWeaver 7.0 Enhancement Package 1 (EHP1), Adobe Reader 8.2 is the minimum version.
- ▶ As of SAP NetWeaver 7.0 EHP2, and if you use Adobe LiveCycle Designer 8.2, you should use Adobe Reader 9.4 or higher. You must also observe whether an integration environment (such as Web Dynpro) specifies the Adobe Reader version to be used.

It is also useful to visit the Adobe Systems Web site (<http://www.adobe.com/reader>) to obtain information about the latest Adobe Reader version. There you can find new versions if, for example, a security problem occurred that was remedied by a new version.

2.2.3 Useful Settings of Adobe Reader

In Adobe Reader, there are some preferences that enable you to customize the behavior of Adobe Reader for the use of interactive forms. These preferences can be found under the menu path **EDIT • PREFERENCES...**

- ▶ Under the **Forms** category you can find multiple options with regard to the display and behavior for filling out an interactive PDF form. The **ALWAYS HIDE FORMS DOCUMENT MESSAGE BAR** option is particularly interesting. You can select this option if document messages are sensitive (for example, if the document contains interactive form fields) or if you want to provide more space for the form itself.
- ▶ In the **INTERNET** category you can find the **DISPLAY PDF IN BROWSER** option. This option specifies whether Adobe Reader is started embedded in the browser. Users frequently search for this option when they find out that Adobe Reader always opens externally and want to change this setting.

2.3 Examples of PDF-Based Print Forms and Interactive PDF Forms

If you have an SAP system at hand that is already installed and configured for the use of SAP Interactive Forms by Adobe, you can continue to the following two examples. If you do not have such a system, you can set up a test system as described in Chapter 3 and then follow the examples. The first example demonstrates the print output; the second example creates an interactive PDF form that is prefilled with data.

2.3.1 Example for Print Output

This example is intended to demonstrate the integration of the print output with the SAP NetWeaver ABAP stack via PDF-based print forms. To do this, you call a print program, enter the parameters, and then create a print output. You can display the print output as a print preview instead of sending it to a printer.

The example assumes that the flight database is filled with data. If required, the flight database can be filled by executing the `SAPBC_DATA_GENERATOR` program in Transaction SE38.

The print output is viewed at runtime when the print program and the form design have already been created. The print program performs the data retrieval and preparation and then initiates the creation of the print output. The print output can be either a printer language (PS, PCL, or ZPL) or a PDF.

1. Log on to the SAP system via the SAP GUI.
2. Start Transaction SE38 to open the ABAP Editor.
3. Enter `FP_TEST_03` as the name of the program. Figure 2.4 shows what the screen should look like.

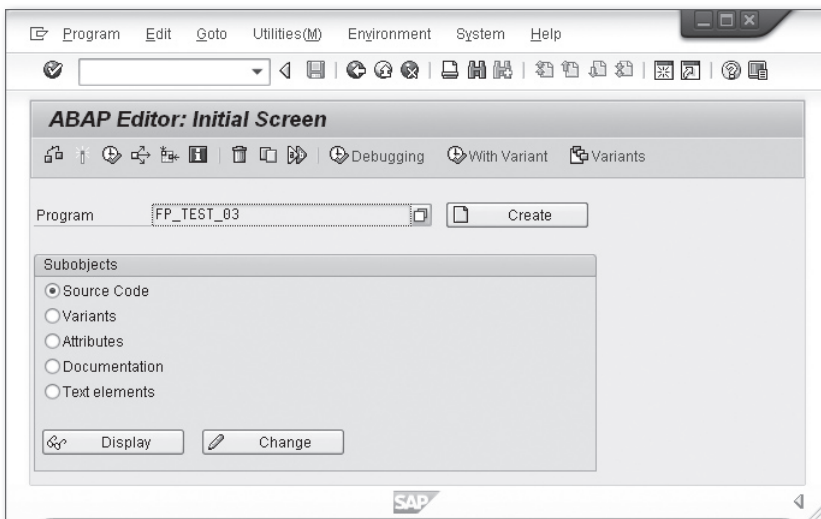


Figure 2.4 Selecting and Starting a Print Program

4. Then, start the program by clicking on the corresponding icon in the toolbar (the clock with the green checkmark) or by pressing the `[F8]` key.
5. The print program is now executed and the screen shown in Figure 2.5 is displayed. You can accept the displayed parameters. After you have run the example, you can change the parameters.
6. You can also select another form by changing the `FORM` parameter in this screen. However, you cannot select just any form—only forms with the same interface. Chapter 4, Interface and Form Context, provides detailed information on this topic.

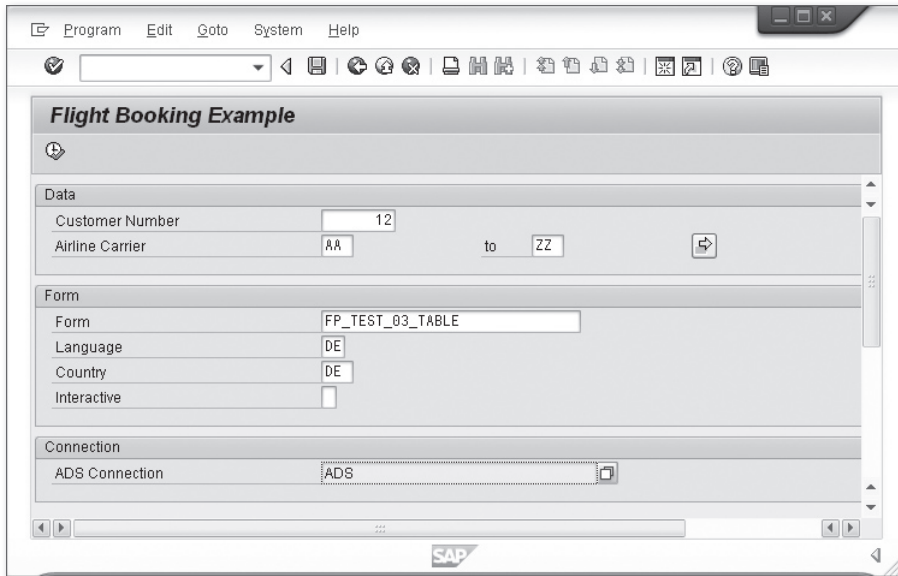


Figure 2.5 Entering Parameters for the Print Program Example

For example, select `FP_TEST_03` or `FP_TEST_03_TABLE`. Again, click on the clock icon or press the `[F8]` key to run the program. This opens the PRINT dialog for selecting the output device (see Figure 2.6).

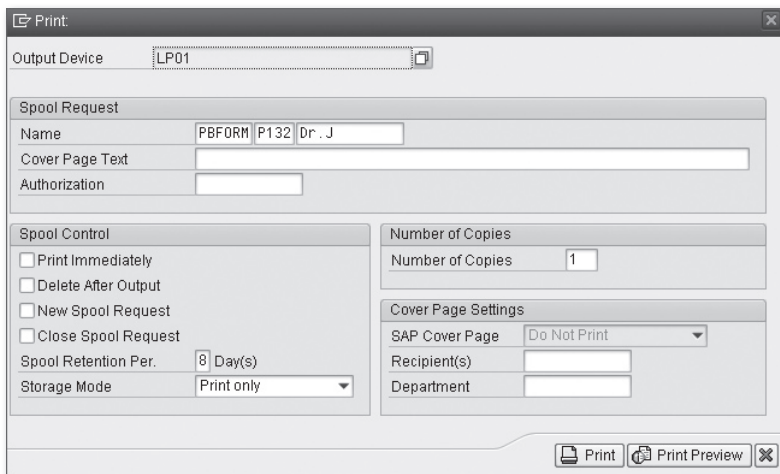


Figure 2.6 Dialog for the Print Output

7. In this dialog, you must select an output device. You can enter "LP01" as text; LP01 is a sample printer present in many SAP systems. If this printer is not available on your SAP system, you must search for an output device via the input help or contact your system administrator.
8. Next, click on the PRINT PREVIEW button (not PRINT) to obtain a preview of the print output.
9. If all steps were performed correctly, Adobe Reader will be displayed in the SAP GUI as shown in Figure 2.7. In this case, Adobe Reader displays the print output as a PDF document. You can see an excerpt from the flight-booking database in accordance with the selected form.

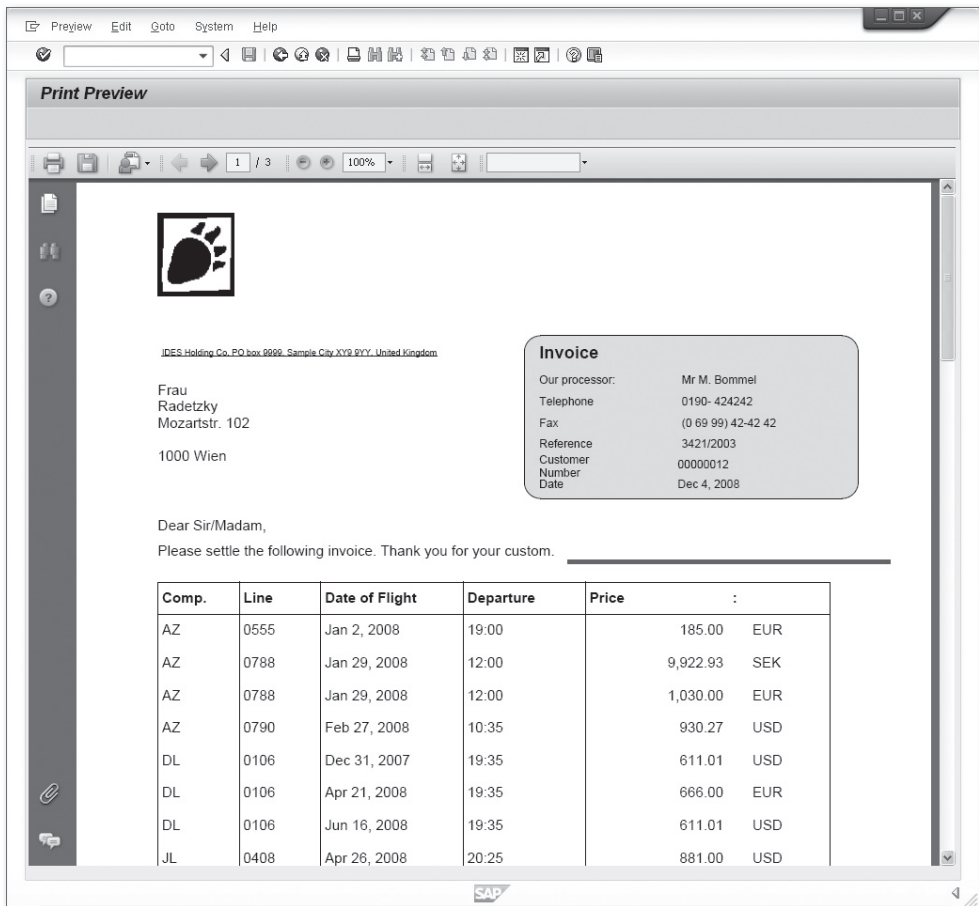


Figure 2.7 Print Preview of the Print Output via Adobe Reader

This example indicates which parts of the print output are identical for all forms. This includes the form layout that describes the appearance of the formatted output and the structure of data. The data quantity per form output is the only variable that is changeable and thus changes at runtime during the various calls. This is based on parameters and therefore different data are selected from the database and processed.

2.3.2 Creating an Interactive PDF Form

The second example considers the creation of an interactive PDF form. SAP provides a test program for this as well. You can test it as described here:

1. Go back to the initial screen of the ABAP Editor (see Figure 2.4).
2. In contrast to the print example, start the `FP_TEST_IA_01` program. You can change the parameters for the test in the following screen.
3. Do not make any changes in the FORM area for the initial test.
4. In the DATA area you can set individual parts of the form to EDITABLE or LOCKED (READ-ONLY).
5. You can change the values that are used for filling the individual form fields during creation. You can accept these defaults or change them as shown in Figure 2.8.

Generate Interactive PDF	
Form	FP_TEST_IA_01
ADS Connection	ADS
<input type="checkbox"/> PDF for Preprinted Paper?	
<input checked="" type="checkbox"/> Header: Read-Only?	
<input checked="" type="checkbox"/> Old Address: Read-Only?	
<input type="checkbox"/> New Address: Read-Only?	
URL	www.adobe.com
Effective Date	20050101
Employee ID	Dr.J
Name	Jürgen Hauser
Department	Software Testing
Old Address: Apartment	233a
Old Address: Street	Main Street
Old Address: City	Stuttgart
Old Address: Province	Arizona
Old Address: Postcode	12345
Old Address: Telephone Number	123456789

Figure 2.8 Screen for Entering Parameters

6. Start the program.
7. In the subsequent PRINT dialog, select a configured printer—just as in the print example—and click on PRINT PREVIEW to confirm.

An interactive PDF form as shown in Figure 2.9 is displayed in the preview of the print output. This form is used for a change of address for an employee. Accordingly, the form includes an area that provides general information, such as name and department, and an area with the old, currently saved address. The NEW ADDRESS area contains interactive form fields for entering a new address.

The screenshot shows a 'Print Preview' window with a toolbar and a form titled 'Please fill out the following form. You can save data typed into this form.' The form contains the following fields:

- Name:** Jürgen Hauser
- Department:** Software Testing
- Employee (Banner) #:** Dr.J
- Please change address/telephone effective:** 01.01.2005
- Old Address Section:**
 - Apt #:** 0233
 - Street & #, P.O.Box or R.R.#:** Main Street
 - City:** Stuttgart
 - Province:** Arizona
 - Old Postal Code:** 12345
 - Old Phone No.:** 000123456789
- New Address Section:**
 - Apt #:** 0000
 - Street & #, P.O.Box or R.R.#:**
 - City:**
 - Province:**
 - New Postal Code:** 00000
 - New Phone No.:** 000000000000
- Date:**
- Signature:**

At the bottom of the form, it says 'Please forward to the Human Resources Department' and the SAP logo is visible in the footer.

Figure 2.9 Preview of the Interactive PDF Form Example

Below the NEW ADDRESS area, there are two more form fields that you can test (you may have to scroll down the visible area of the PDF form to see them).

- ▶ The first field is a date field that provides a convenient date selection screen.
- ▶ The second field is the field for a digital signature.

At this point, you may test the procedure for applying a digital signature. To do this, a certificate must be installed on your PC. Adobe Reader indicates the certificates available. You can also check which usage rights have been added to the form (see Section 2.2.1 for more details).

2.4 Using Interactive Forms in Business Processes

The creation of interactive forms is the second important application area in addition to the print output. Interactive forms can be used to optimize individual steps of a business process. In this case, optimizing processes means the following:

▶ **Increasing the data integrity**

Errors can occur whenever data are transmitted or processed manually. If, for example, the handwriting on a paper form is illegible or cannot be read entirely after transmission by fax, this may result in problems during further processing.

So what are you supposed to do if you want to enter the data into the system? In most cases, you would simply guess. Guessing may result in higher costs, if, for example, electronic bank transfers fail or a second manual revision is necessary. Consistent electronic entry of data can remedy this problem.

▶ **Eliminating manual steps**

Manual entry of data must be carried out by a person. However, before this person can start entering data, the data must be sent to the person in paper form. The transfer of data to the entering person can be done by fax or by mail. Frequently, the documents are scanned in and the data are then provided in electronic form.

Another example is the entry of data during a telephone call (for instance, telephone orders). Usually, you cannot control the data entered on either side.

All of these manual steps result in processing delays. These delays ultimately prolong the overall duration of the business process, which means, for example, that products are delivered with a delay and suppliers receive their payment later.

Consistent electronic processing can considerably accelerate business processes by eliminating manual steps.

For the use of interactive forms in business processes, there are two different types of scenarios: online scenarios and offline scenarios. In this case, the concepts online and offline do not indicate whether the users have network access but whether the users are logged on to an SAP system or not.

2.4.1 Online Scenarios

Online scenarios mainly feature the following two characteristics:

► **Direct access to an SAP system is required and possible**

During the use of a PDF form, it is possible to call an SAP system, for example, to perform validations, implement a value help, or carry out complex calculations. This requires that the user is known in the system and that he is logged on to an SAP system.

► **Integration with other user-interface technology**

PDF forms are never displayed alone. They are displayed in a Web browser and are surrounded by other Web technology that calls Adobe Reader embedded in an (HTML) page.

A typical online scenario may look like the following: The user is logged on to SAP NetWeaver Portal. In the portal, he can view the pages of a Web Dynpro application in an iView. Web Dynpro enables you to integrate a PDF form with a view as a user-interface element and supports the implementation of value helps, validations, etc., for PDF forms.

2.4.2 Offline Scenarios

Offline scenarios may be characterized as follows:

► **Access to an SAP system is not possible or required**

When the user processes a form, he cannot (and does not have to) call an SAP system. Therefore, value helps, validations, and simple calculations must be contained in the PDF form by means of a scripting language. When the processing of the PDF form is completed, it is transferred to an SAP system. This transfer is done either indirectly (for example, via an email that contains the PDF form

as an attachment) or directly by logging on to the SAP system and uploading the PDF form to an application.

For example, if the user is not logged on to the company intranet, it is usually not possible to access an SAP system. If the user is logged on to the intranet, he can send the PDF form, or the data contained therein, to an SAP system via a Web service or HTTP call.

► **Independence of other user-interface technologies**

PDF forms are opened and processed with Adobe Reader exclusively. Therefore, integration with another technology for creating user interfaces is not necessary.

These two characteristics require only that the PDF form contain all of the necessary information (such as value helps and data). This may result in larger PDF files, which can impact the transfer time. This is important if the network connection is slow, for example.

The fact that all necessary information is already contained in the PDF form and that no access to an SAP system is required entails very interesting application options. Consider the following implementation example of an offline scenario:

A user receives an email including a request to increase the budget of a cost center. The details of this request (currently approved budget, requested budget, and already-spent budget) are contained in an interactive PDF form that has already been sent to the user as an email attachment. Therefore, this interactive PDF form provides all of the necessary information for making a decision. Additional access to an SAP system is not required to fulfill the task. After the decision has been made (approval, rejection, or approval of a different budget), the form, including all changes, can be transferred as follows:

- It is returned to the SAP system via *email* for further processing.
- Or the PDF form can be sent to the user as an *attachment to a proposal for an appointment*. At the customer's site, you can then enter all of the relevant data in the form. Subsequently, it is returned to the SAP system for updating.
- A *portal* can also provide the option to download an interactive PDF form, fill it out, and start a process when it is returned to the SAP system.

Example of Using a Portal for Transfer

An example of using a portal for transfer is when a customer places an order with a mail-order business. If the user logs on to the system, the form should already be filled in with important customer information. This information can be, for example, name, address, customer ID, and the last order as an initial value. Filling in business data in advance ensures that the user can conveniently use the interactive forms.

2.4.3 Combining Online and Offline Scenarios

Another approach is to provide the user with an option to go offline midway through the process. This constitutes a mixture of offline and online scenarios. Examples of this include the entry of travel expenses or working time.

When planning a business trip in the enterprise portal, you can prepare the entry of travel expenses. One step of the business process could be the display of a PDF form in which the user can enter the expenses. Depending on the process implementation, the user could download the form or receive it via email. During the business trip, he can fill out the PDF form on his laptop. The user interface for travel expenses enables the business traveler to upload the PDF form via a portal upon his return. It is then displayed for checking, the entries are validated, and possible confirmations are displayed. Corrections to the form can be done online.

2.4.4 When Do You Use Online and Offline Scenarios?

In addition to the question about the functioning and programming of the technology, you must also consider when you want to use interactive forms. To do this, the following lists provide some support. Interactive forms are suitable as a method for implementation if at least one of the following criteria applies:

- ▶ **Imitating the layout of well-known forms**

A well-known form implies that the user already knows how to handle it. This familiarity enables a fast transfer of the former paper-based process into an electronic process. The (re)use of a well-known form can reduce training costs.

- ▶ **Simple user interface for occasional SAP users**

Interactive forms allow for the creation of simple interfaces for occasional users of SAP systems. Instead of logging on to an SAP system, the user receives all of

the necessary information for the operation in a compact PDF form. This enables a fast and convenient participation in the business process.

▶ **Creating a document for archiving**

In contrast to other technologies (for example, all HTML-based Web technologies), completed PDF forms are available for further use. This way, they can be archived to fulfill regulatory requirements of traceability.

▶ **Local print output of the form**

PDF forms enable you to quickly create high-quality prints on site. For many user-interface technologies (for example, HTML) this is possible only with some difficulty and extra effort.

▶ **Support of digital signatures**

PDF forms can contain fields for digital signatures. However, country-specific requirements must be taken into account. You can insert multiple digital signatures into a form within workflows (the respective fields must be contained in the form).

▶ **Several persons are required for data entry**

PDF forms support data entry that cannot be implemented by one single person. Frequently, forms cannot be completed by one person only because the required information must be provided by several people (for example, information on products and finances).

You should consider the implementation of an offline scenario if one of the following criteria is met:

▶ **Integration of external users**

A person who is not an employee of the enterprise is supposed to participate in a business process but must not have access to the enterprise's SAP systems.

▶ **Offline use of a form**

The user group of the form is in the service field or on a business trip and therefore has no access to the intranet and hence no access to the SAP system.

An alternative approach is the analysis of existing business processes. If you can identify manual steps or detect intermediate statuses in processing that are similar to the following two examples, you should consider the use of interactive forms for process optimization:

▶ **Conversion of existing paper-based processes**

It is possible that today's paper-based forms can be replaced with electronic

versions. With the introduction of electronic forms, you can digitize parts of or complete processes.

► **Approval processes**

In approval processes, you either change or create new system data, for example, master data. However, this modification is supposed to become effective in the system only after it has been approved. In this case, interactive PDF forms are well suited as data containers. The data saved in the PDF form are transferred to the SAP system only after approval.

If you compare the various options available in online and offline scenarios, you will notice that the full efficiency of interactive PDF forms is utilized in offline scenarios. The need to integrate external users with business processes indicates that an implementation using interactive forms makes sense. This ultimately results in a faster execution of the business process and higher data quality—and therefore optimized process costs.

2.4.5 Notes on the Use of Interactive Forms

“The why and the how” play a decisive role in the use of interactive forms in the implementation of business processes.

Note for Offline Scenarios

In offline scenarios, the usability of interactive forms primarily depends on the forms' size. The larger the forms are, the more difficult it is for the user to navigate within the form. For this reason, it is useful to use the concepts of dynamic forms.

Depending on the data (for example, marital status, nationality, or answers to yes/no questions), some parts of the forms can be either hidden or displayed. This way, you can reduce the size and the complexity of the form. In extreme cases, you can implement navigation with tabs that are similar to user interfaces.

A welcome side effect is that the behavior with regard to performance sometimes considerably improves if, for example, large dynamic forms (more than ten pages) do not have to be constantly re-created in Adobe Reader.

Notes for Online Scenarios

Online scenarios entail that you ensure a balance between the user interface parts of the embedded technology (for example, Web Dynpro) and the interactive form. Consider some extreme examples that illustrate this concept:

- ▶ If an interactive form with multiple pages is embedded in Web Dynpro and if the Web Dynpro screen does not include any interface elements, the user faces a complex navigation in the interactive form.
- ▶ Another extreme example is multiple dynamic tables—tables in which you can insert or delete rows—which are distributed to multiple pages.

A possible solution for these cases is to reduce the complexity—just as for offline scenarios. In both cases, you should check why interactive forms are used (see Section 2.4.4, *When Do You Use Online and Offline Scenarios?*) to ensure that interactive forms are not used just because they are easy to implement.

Interactive Forms are not a Replacement for Complex User Interfaces

SAP Interactive Forms by Adobe is a form technology that is oriented toward the creation of forms. There are fewer functions provided, particularly in comparison to user-interface technologies; for example, it is not possible to program a form entirely in JavaScript without creating a form template in Adobe LiveCycle Designer.

Therefore, you should use SAP Interactive Forms by Adobe for form-based scenarios and not as a replacement for a user-interface technology. You will encounter difficulties if you still use forms as a replacement for user interfaces.

Notes for Combined Online/Offline Scenarios

If you combine online and offline scenarios, for example, to map the completing and processing of a form, you should carefully consider whether you should use one and the same form in both scenarios. Consider the notes provided in Section 2.4.4 to check whether an interactive form is required for the online scenario. Keep in mind that the effort saved for the design and the implementation of the application interface usually entails a greater development and maintenance effort.

2.5 Software Components and Architecture

Now that you have learned what PDF-based print forms and interactive forms are and when they can be used, let us look at the required software components and their integration with SAP NetWeaver. SAP Interactive Forms by Adobe mainly consists of three software components, which can be used only after they have been integrated with different SAP environments:

- ▶ **Adobe LiveCycle Designer**

Adobe LiveCycle Designer is used at design and implementation time.

- ▶ **Adobe Reader**

Adobe Reader is required on the user's PC at runtime.

- ▶ **Adobe Document Services (ADS)**

ADS must be installed and configured in the SAP NetWeaver stack at runtime.

Adobe Reader must be downloaded from the Adobe Website. ADS and Adobe LiveCycle Designer are available directly from SAP. They are provided via the SAP Service Marketplace (<http://service.sap.com>) or via SAP Solution Manager.

2.5.1 Adobe LiveCycle Designer

Adobe LiveCycle Designer is a tool that the form designer uses to create the layout of forms. The layouts of PDF-based print forms and interactive forms are created in the same way. The same tool, Adobe LiveCycle Designer, is used for both types of forms. Figure 2.10 shows how you can use Adobe LiveCycle Designer as a desktop application.

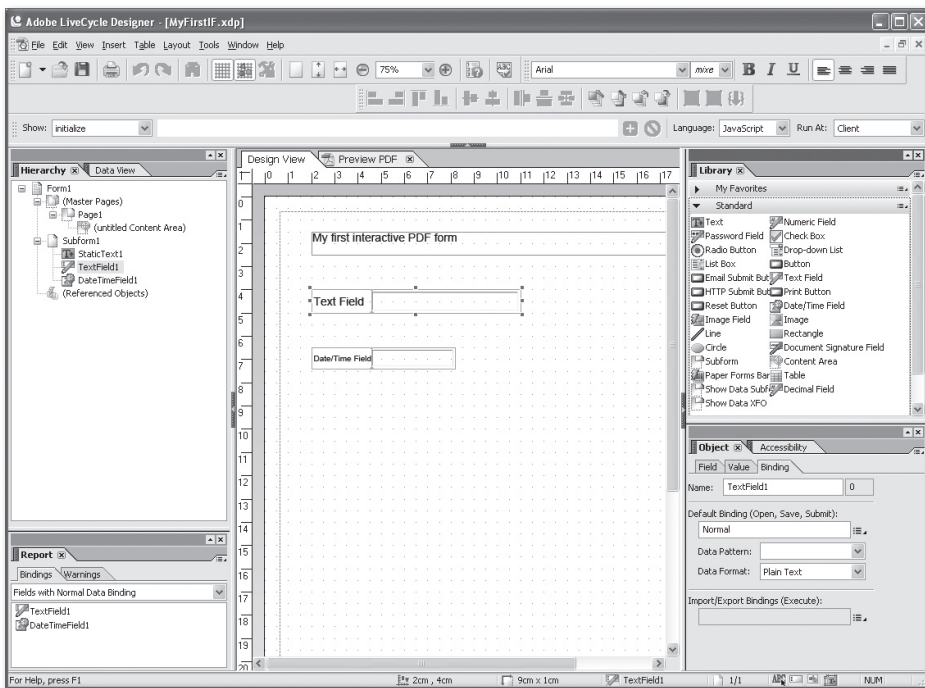


Figure 2.10 Adobe LiveCycle Designer

Adobe LiveCycle Designer is a graphical interactive tool; many of the required steps can be implemented using the mouse. The layout of the form is displayed graphically at design time (the window shown in the middle of Figure 2.10). Changes, such as the positioning of form fields, are carried out using drag and drop.

In addition to the layout, Adobe LiveCycle Designer graphically displays the form hierarchy and the data context:

- ▶ The form hierarchy indicates the structure of the pages that consist of the form fields used and the subforms. Subforms are structuring elements that enable you to create and maintain more complex forms. Nested subforms therefore constitute a form hierarchy.
- ▶ The data context describes the data structure of the form. Technically, a form is based on an XML data structure with a corresponding XML schema. In the SAP environment, Adobe LiveCycle Designer is used almost exclusively in the context of a framework. These frameworks create the XML schema at design time. During runtime, the XML data required for form output are generated automatically.

Adobe LiveCycle Designer provides all of the available form fields in a library. From this library, you can insert new form fields into the form layout using drag and drop. After a new form field has been added to the form, you can define additional properties using the palettes. Some examples are the appearance of buttons or edit and display patterns. The appearance determines whether a button has a border or a 3D effect. The edit and display pattern can be used to format figures, times, or a date.

Very complex behavior of forms can also be implemented using script programming. Forms are based on an event model and you can create a script program for many predefined events (for example, the `initialize` event for the time of initialization of a field).

You can choose between two different scripting languages:

- ▶ **JavaScript**

The first option is JavaScript, a scripting language from Web programming. It allows for easy access to script programming for form creation.

► **FormCalc**

The second option is Adobe's proprietary scripting language, FormCalc. FormCalc has a higher performance rate than JavaScript. In particular, calculations that are based on tabular data structures can be implemented more easily with FormCalc than with JavaScript.

Most PDF-based print forms can be used without any script programming. Interactive forms with dynamic behavior, however, often require scripts to implement this dynamic behavior (for example, inserting or deleting table rows).

SAP Version of Adobe LiveCycle Designer

You must use the SAP version of Adobe LiveCycle Designer for the development of PDF-based print forms and interactive forms, and not the version available from Adobe (for example, as part of Adobe Acrobat Professional or other Adobe LiveCycle products). Therefore, you must download Adobe LiveCycle Designer from the SAP Service Marketplace.

Embedding Adobe LiveCycle Designer in SAP Development Environments

Adobe LiveCycle Designer is already embedded in the development environments for the use in the SAP environment. This results in the following benefits:

- Adobe LiveCycle Designer is integrated with the development process and can be called at the right point and at the right time.
- Adobe LiveCycle Designer is always called in the context of the current development work. Therefore, the form to be processed is already opened and the data view already displays the underlying data structure.

As a result, the form design must be saved in the surrounding development environments. Therefore, the embedded Adobe LiveCycle Designer does not provide any FILE menu. You cannot find the menu item for file properties under FILE • FORM PROPERTIES... as is the case in the independent Adobe LiveCycle Designer; instead, it is available under EDIT • FORM PROPERTIES...

Figure 2.11 shows the embedding of Adobe LiveCycle Designer in the SAP GUI. Here, Adobe LiveCycle Designer is integrated with Form Builder. Form Builder is the environment in the ABAP Workbench that bundles all of the development tools required for the use of SAP Interactive Forms by Adobe.

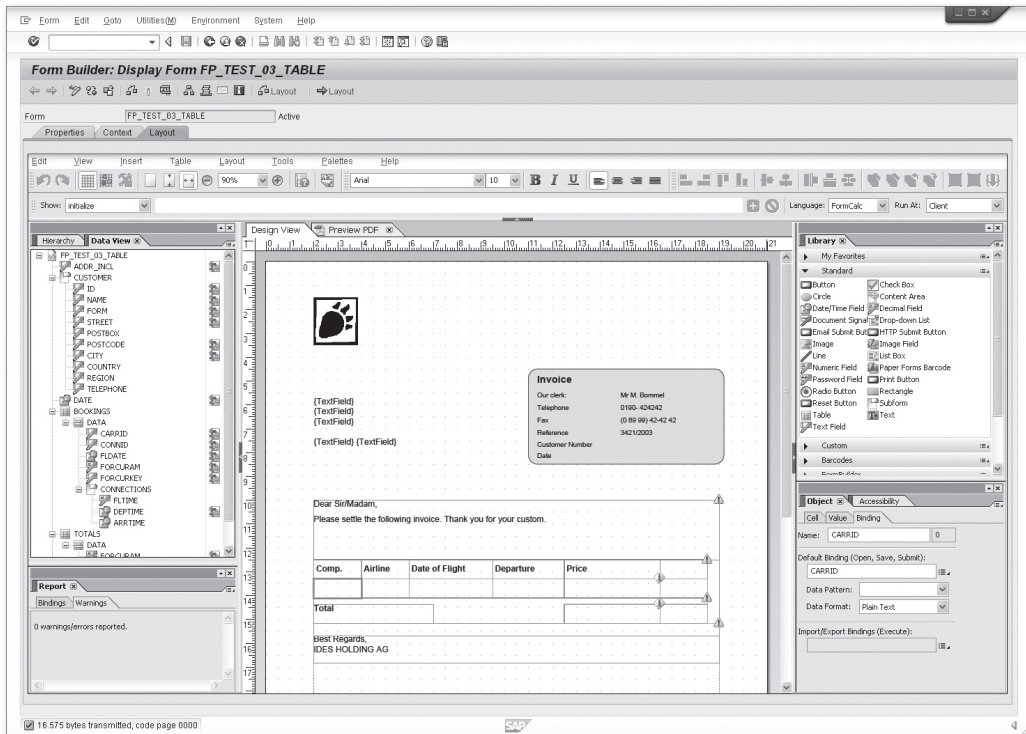


Figure 2.11 Adobe LiveCycle Designer Embedded in the SAP GUI

To check whether your installation is correct, you must call Form Builder and navigate to the form template shown in Figure 2.11.

1. To do this, start the SAP GUI and call Transaction SFP. The screen shown in Figure 2.12 should appear.
2. Enter the name of the form in the FORM input field—in this case FP_TEST_03_TABLE.
3. Then click on the DISPLAY button.
4. In the following screen, select the LAYOUT tab to go to the screen shown in Figure 2.11.

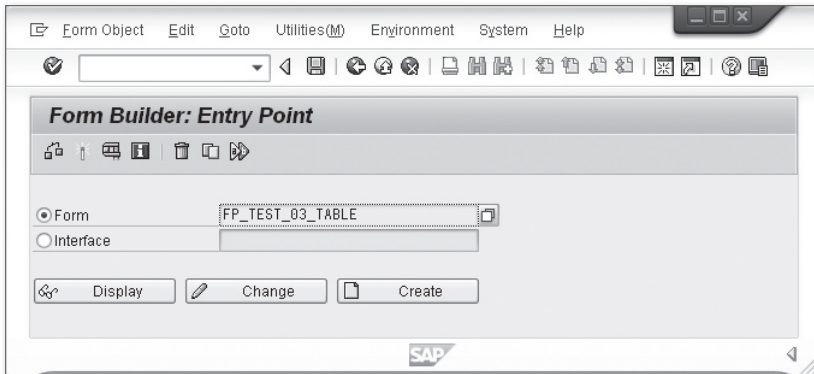


Figure 2.12 Initial Screen of Transaction SFP

For the sake of completeness, Figure 2.13 shows how Adobe LiveCycle Designer is embedded in the SAP NetWeaver Developer Studio. The SAP NetWeaver Developer Studio is an Eclipse-based development tool that is usually used for development in Java.

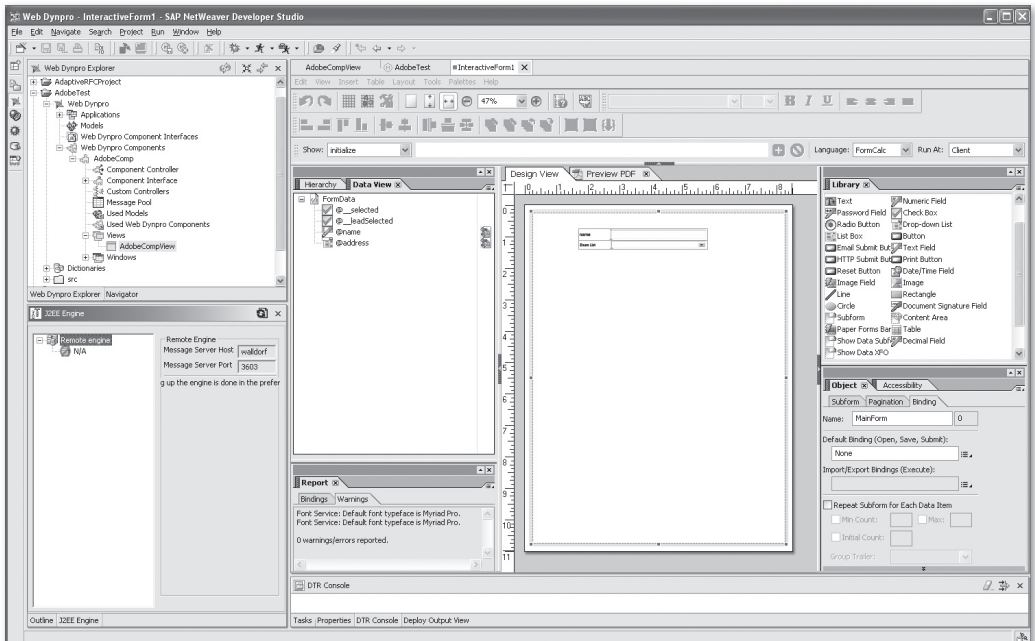


Figure 2.13 Adobe Designer Embedded in the SAP NetWeaver Developer Studio

In SAP's Java world, there is only one integration of SAP Interactive Forms by Adobe, that is, the integration with Web Dynpro Java. This book focuses on development using ABAP because multiple integrations exist here. For this reason, the main focus is on integration with Web Dynpro ABAP and not with Web Dynpro Java.

2.5.2 Adobe Document Services

ADS is the server component installed on the SAP NetWeaver stack; strictly speaking, it is installed on the Java stack and comprises Java and C++ coding. It must be configured after the installation and prior to the first usage. A configuration is also necessary on the ABAP side. In a nutshell, the ABAP stack must be notified of which ADS installation is to be used and how it can be reached. Only after these two configurations have been completed can ADS be used by ABAP.

ADS and Adobe Document Server

ADS is a software component of SAP Interactive Forms by Adobe. Do not confuse this component with Adobe Document Server.

Adobe Document Server is an Adobe product that is maintained only within the framework of warranty services and will be taken off the market soon. Adobe LiveCycle provides a similar scope of functions.

First let us consider how you can use ADS within the ABAP and Java world (see Figure 2.14). On both sides, there is a PDF object that represents the lowest usable interface. The PDF object calls ADS.

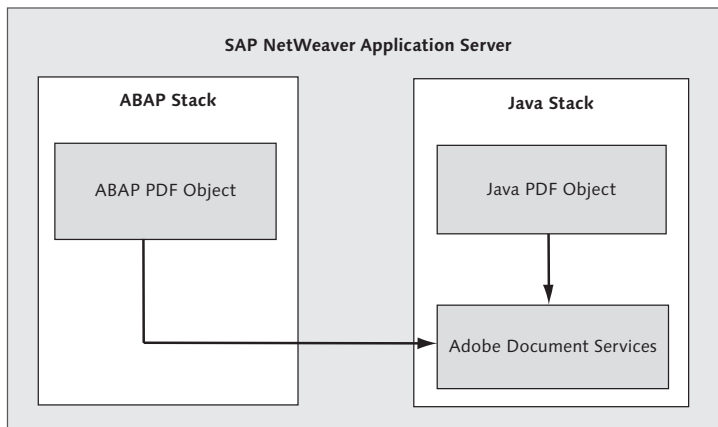


Figure 2.14 Main Components from the Developer's View

The PDF object is used exclusively whereby you can use only the functionality existing in the respective PDF object. Currently, the PDF object in ABAP provides considerably more functionality. The Java PDF object is mainly used in Web Dynpro runtime; the ABAP PDF object, however, calls ADS from the Java stack. The Java stack is still required for ADS for applications that are completely implemented in ABAP.

ADS Requires the Java Stack

ADS constitutes an SAP NetWeaver solution extension that can be installed and operated only on the Java stack of SAP NetWeaver Application Server. Therefore, you need a Java instance in your system landscape, even if the PDF forms are used only in the ABAP world.

Using ADS

Figure 2.15 illustrates which frameworks are based on the two PDF objects or on other frameworks.

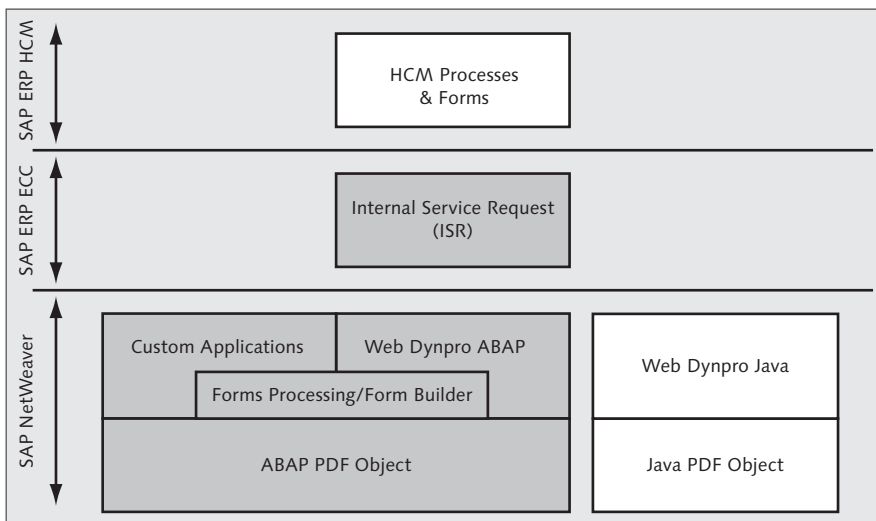


Figure 2.15 Hierarchy of Integration and Frameworks

The PDF object is the lowest layer. Web Dynpro is based on the PDF object in both worlds (ABAP and Java). In the ABAP world, you are also provided with Form Builder and the Forms Processing runtime. You can also develop custom

applications that use the PDF object directly. In this case, Form Builder is used to create the PDF forms and save them in the repository. This functionality is available in SAP NetWeaver directly.

SAP Enterprise Core Component (ECC) contains the Internal Service Request (ISR) framework, which is based on Web Dynpro. In more recent releases of SAP NetWeaver, it is based on Web Dynpro ABAP; in the past, it utilized Web Dynpro Java. ISR supports the implementation of online scenarios using a workflow integration.

This book considers all integrations that are displayed in gray in Figure 2.15. It imparts the necessary basics that also apply to the frameworks and application options not considered within the scope of this book. These are briefly explained in the following text for the sake of completeness.

- ▶ HCM Processes & Forms is a framework that has been developed specifically for personnel administration. It is part of SAP HCM and allows for the implementation of processes in personnel administration using interactive PDF forms. It is a specialization and further development of the ISR framework. For this reason, this book considers only the ISR framework because it provides all necessary basics.
- ▶ SAP NetWeaver Business Process Management is the second framework that is not considered. SAP NetWeaver Business Process Management is a complex tool for creating business processes without any programming. This framework includes an integration of interactive PDF forms for implementing individual offline steps of a process. A process is either started by processing and sending an interactive PDF form (to the SAP system) or an offline step is required midway through the process flow. In this case, an interactive PDF form is sent to the user, who then participates in the process. The process is stopped until the user has processed the PDF and returned it to the system. With regard to the integration of interactive PDF forms, SAP NetWeaver Business Process Management provides functions similar to the older Guided Procedures.
- ▶ There is also indirect integration. Indirect integration includes all options in which interactive PDF forms are not directly integrated with a technology or framework; however, they can be integrated with an existing framework using another technology. An example is the integration of a Web Dynpro application

with SAP Business Workflow, with SAP NetWeaver Business Process Management, or with Guided Procedures. This way, the mentioned frameworks can be used to support online or offline scenarios. For the implementation of interactive PDF forms in SAP Business Workflow, you need a working knowledge of the frameworks with which you want to integrate and of the integration of interactive PDF forms with Web Dynpro. This book provides details about the latter. However, it would go beyond the scope of this book to describe SAP Business Workflow, SAP NetWeaver Business Process Management, or Guided Procedures.

Communication between PDF Objects and ADS

PDF objects are implemented either in Java or ABAP. The communication structure with ADS is illustrated in Figure 2.16; it is identical for both languages.

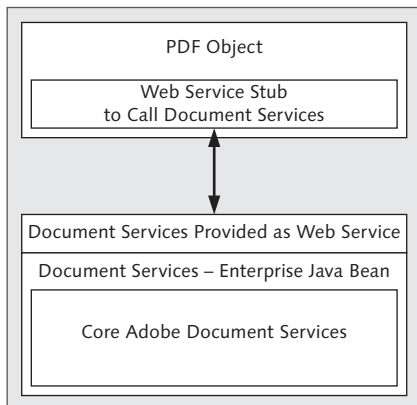


Figure 2.16 Interaction of PDF Object and ADS

The communication between the PDF object and ADS is carried out by calling a Web service. The PDF object implements a Web service stub, which is utilized to call ADS.

The PDF object itself does not contain a lot of logic. Its main task is to correctly call ADS via the Web service stub based on the parameters and the required functionality. Both the PDF objects and ADS are located on the SAP NetWeaver Application Server (AS). ADS is always installed on the Java stack.

Structure of ADS

ADS comprises two components: Document Services and Core Adobe Document Services (see the lower part of Figure 2.16):

- ▶ Core Adobe Document Services provide the main functionality for the server side. This includes, for example, the creation of a PDF or various printer languages, extraction of data from the PDF, insertion of a server-side signature, or adding of usage rights for a PDF. Functions, such as the administration of fonts or certificates stored on the server, are part of Core Adobe Document Services.
- ▶ Document Services are on the next abstraction level. These utilize the functions made available by Core Adobe Document Services; however, they provide an application programming interface that comprises more than just simple basic functions. This enables you to perform multiple tasks in one Document Services call. Document Services ensure that basic functions are used in the correct sequence and that you require only one Web service call. You can save a lot of administration and communication effort with just one single call. This is particularly significant if you call multiple fast functions individually.

Interface to ADS

Document Services are combined and managed in an Enterprise JavaBean. The interface of these JavaBeans is provided as a Web service and the interface of the Web service via a Web Service Definition Language (WSDL file). The Web service stub for the PDF objects can be created from the WSDL file. This is illustrated in Figure 2.16.

The Web service of ADS is a public Web service that cannot be used directly. You can find it in developer and administration tools within SAP NetWeaver without any problems—the documentation of the Web service and its parameters, however, is not published. Ultimately, it is the interface agreed upon by SAP and Adobe.

The PDF object abstracts from this Web service interface and is the lowest interface to ADS that is released for application development. This abstraction provides SAP and Adobe a wider scope for future developments.

2.5.3 Interaction of Components

Up to now, this chapter has described individual software components and the interaction of PDF objects and ADS. The following sections detail the overall

architecture of the integration at both development time and at runtime. You will need this information when reading the following chapters, for example, to understand and solve problems or to comprehend why certain decisions were made in frameworks.

Considering Development Time

First, development time is discussed. Figure 2.17 illustrates how you can use the development environment to create a form template (which is based on a specific context) and make it available for further use.

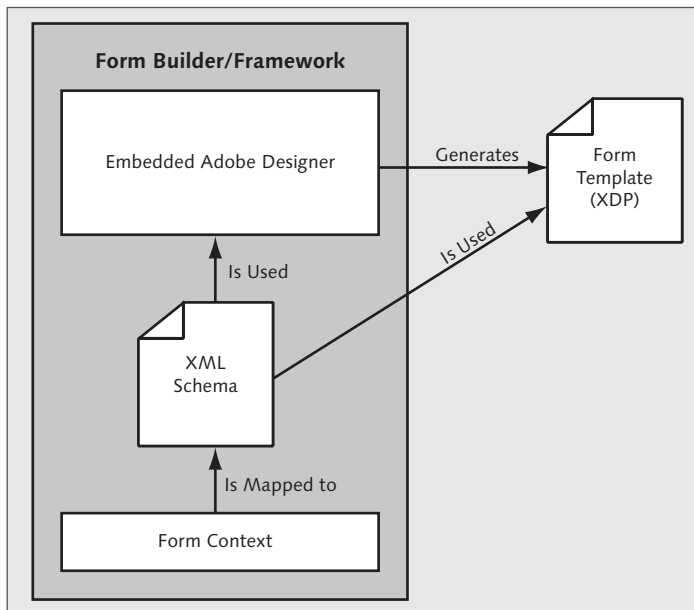


Figure 2.17 Interaction of Components at Development Time

1. At development time, Adobe LiveCycle Designer, which is integrated with the SAP environments, is used to create form templates, which are then converted into PDF documents or print outputs at runtime. This form template corresponds to the description of a data structure.
2. All development environments have a form context (often also referred to as context). In most cases, this context must be set up manually prior to the actual form template creation. The context is the intermediary element between the SAP framework and the Adobe components.

3. The development environment creates an XML schema based on the context. This XML schema includes the description of the data structure, which is based on the form template. This description is transferred to Adobe LiveCycle Designer and is displayed in the data view of the form designer's screen. This is done automatically.
4. Now, the form designer creates the form template. To do this, Adobe LiveCycle Designer, which is embedded in the development environment, is started. The form template includes the form layout; for interactive PDF forms, you must specify the behavior and integrate the form with the data structure. By integrating the form template with the data structure, you always create a form template for a specific data structure.

If you change the data structure, you may also have to change the form template; this is the case for structural changes of the data hierarchy. Adding new data nodes does not require a modification of the form template; however, if you want to use the data node, you must extend the form template accordingly.

5. When you save the form template, it is returned to the SAP development environment that embeds Adobe LiveCycle Designer. The development environment then ensures that the form template is stored at the correct position with the correct name (for example, in ABAP Repository). There are several reasons for this automation, such as facilitating the form designer's work or ensuring that form templates can be referenced and retrieved.

Considering the Runtime

Every framework has a specific runtime environment in addition to a development environment—for PDF-based print forms, it is the Forms Processing runtime and for Web Dynpro the Web Dynpro runtime.

1. At runtime, form data must be created that corresponds to the data description and therefore to the context and the XML schema. The data can be read from the database or are already available in the framework, for example.
2. Before the data are transferred to the PDF object, they must be prepared and converted into XML format, which must correspond to the XML schema.
3. In the next step, the runtime must locate the form template required for the current task and transfer it to the PDF object.

4. For the actual call of the PDF object, the runtime environment decides on the required output format or whether usage rights should be added in case of an interactive PDF form. It then calls ADS via the PDF object.

Figure 2.18 summarizes the steps at runtime up to the call of the PDF object.

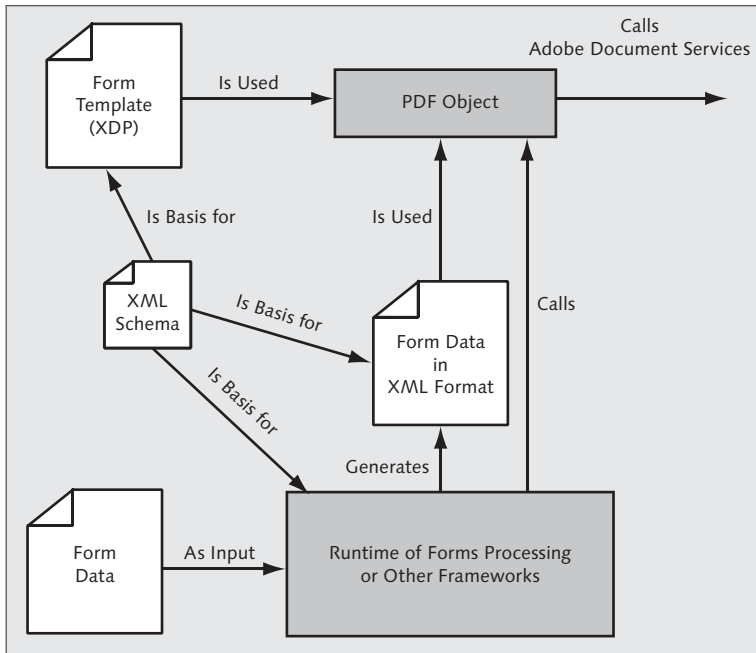


Figure 2.18 Interaction of Components at Runtime (Within the Framework)

Figure 2.19 summarizes how input data (form data and form templates) are transferred to ADS using the PDF object at runtime.

1. The PDF object uses the form data and form template that were provided by the runtime environment.
2. In this process, the PDF object calls ADS. The data, the form template, and the parameters are transferred to ADS in accordance with the interface definition.
3. ADS processes the data and form templates according to the parameters and thus creates the required output format.

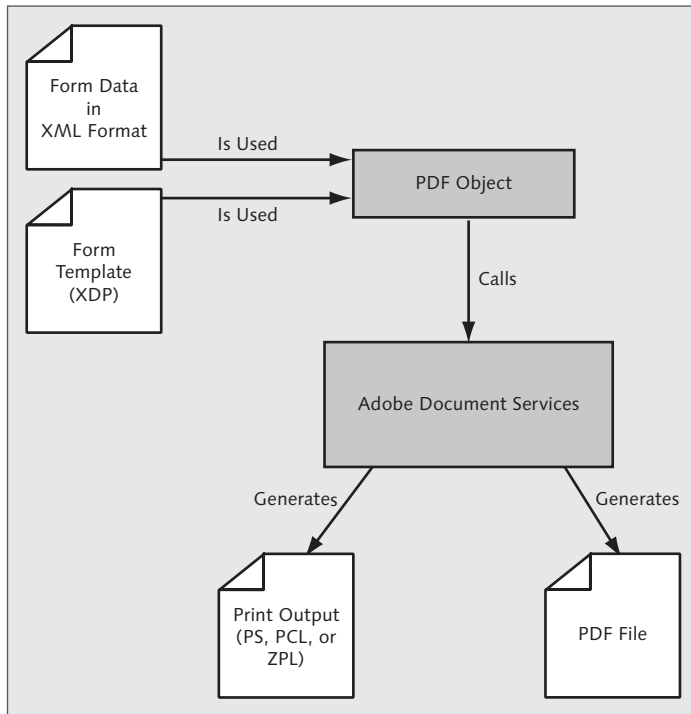


Figure 2.19 Interaction at Runtime (Call of ADS)

2.6 Summary

This chapter briefly introduced you to the terminology used for PDFs. In this context, two central concepts, PDF-based print forms and interactive forms, were detailed.

Knowledge of the classification of application scenarios in offline and online scenarios enables you to select the correct framework for your implementation. Furthermore, you were provided with decision support for assigning business processes to scenarios. This chapter also provided support to answer the question whether you should use interactive forms at all.

Users who participate in business processes comprising interactive forms must install Adobe Reader on their computers. In this context, this chapter compared Adobe Reader and Adobe Acrobat and pointed out the relevance of the various versions. Adobe LiveCycle Designer was presented as a tool that you can use to

create form templates during development. The form templates and data from the SAP systems are converted into different output formats (printer languages and PDF) at runtime using ADS.

Finally, you learned about the integration of ADS with SAP NetWeaver. This information is particularly useful for Chapter 3. It can also be beneficial for troubleshooting. If you already have an installed and configured system at your disposal, you can skip Chapter 3.

Index

\$record, 197
1D barcode, 362
2D barcode, 362

A

ABAP
 print program, 72, 89
ABAP Dictionary-based interface, 555
ABAP Editor, 38
ABAP Offline Infrastructure, 605
 availability, 606
ABAP PDF object, 625, 635
ABAP Workbench, 115
Accessibility, 28
 form, 369
Access permission, 503
Account assignment
 true, 550
Account assignment object, 549
ACF, 411
 integration with Web Dynpro, 427
Action launchpad, 542
Activation, 123, 523, 529
 status, 523
Active Component Framework -> see ACF, 411
Activity, 503, 518, 519, 530, 534, 537, 539, 541
 automatic, 518
 programming, 539
Activity bar, 538, 592
Activity list, 519, 541
Activity type, 548
ADDITIONAL_DATA, 589
Address, 157
 personal, 159
Address node, 157
Address number, 158
Address updating, 606

Administrator, 504
Adobe
 Developer Connection, 369
 integration, 46
 PostScript, 25
Adobe Acrobat, 29, 30
 JavaScript, 354
 JavaScript reference, 370
Adobe Document Server, 52
Adobe Document Services -> see ADS, 52
Adobe LiveCycle Designer, 47, 63, 175, 177, 616
 data view, 178
 embedding in the development environment, 49
 form design area, 182
 function, 222
 hierarchy, 178
 installation, 110
 library, 180
 menu bar, 177
 PDF preview, 193
 SAP version, 49, 176
 script editor, 177, 344
 scripting reference, 369
 toolbar, 177, 222
 version, 33
 ZCI script, 412
Adobe Reader, 29, 30, 33, 85, 88, 98, 504, 512, 576
 document message bar, 34
 error analysis, 414
 error tolerance, 414
 for pocket PC, 30
 height adjustment, 414
 rollout, 415
 target version, 226
 toolbar, 450
 Version 8.1.2, 29
 Web Dynpro feature, 413
 width adjustment, 414

- Adobe XML Forms Architecture -> see XFA, 25
 - ADS, 52, 55, 236, 579
 - basic functionality*, 56
 - certificate*, 713
 - configuration*, 695
 - connection*, 236
 - error analysis*, 105
 - installation*, 695
 - interaction with PDF object*, 56
 - interface*, 56
 - optional configuration*, 96
 - scenario-dependent configuration*, 72
 - secure connection*, 695
 - structure*, 56
 - usage*, 53, 107
 - ADS_AGENT, 75, 78, 106
 - ADSCaller, 107
 - ADSCerts, 713
 - AD_SMTPADR, 610
 - ADS_SSL, 652, 695, 727
 - ADSSSL, 700
 - ADS_SSL_Client.crt, 708, 719
 - ADS_SSL_Client.pem, 706, 707
 - ADSUSER, 65, 66, 106, 107, 713
 - ADSUser-cert.crt, 716, 719
 - Alternative node, 161
 - AND operator, 162
 - Annotation, 658
 - Appearance, 188
 - Applicant, 503, 504, 508, 510, 513, 518, 519, 530, 535, 548, 592
 - Application, 374, 532
 - Application log, 625
 - Application/pdf
 - MIME-type*, 471
 - Approval, 579
 - Approval form, 552
 - Approval step, 502, 519, 544
 - Approver, 503, 514, 519, 530, 538, 543, 544, 559, 579
 - Architecture
 - Adobe integration*, 46
 - Archive, 249
 - index*, 250
 - Archive information, 120
 - Archiving, 120, 249
 - Archiving mode, 236
 - Assignment, multiple, 564
 - Assistance class, 377
 - AttachFileToResponse, 486
 - Offline scenario*, 488
 - Attachment
 - Web Dynpro integration*, 454
 - Attribute, 375
 - Authentication
 - object*, 685, 689, 691
 - transport level*, 693
 - Web service*, 688
 - Authenticator, 689
 - Authorization object
 - B_NOTIF*, 505
 - I_QMEL*, 505
 - S_BDS_DS*, 505
 - S_USER_GRP*, 505
 - Automatic kerning, 195
 - Auxiliary function, 590
 - Availability query, 112
- ## B
-
- Background fill, 293
 - BAdI
 - QISR1*, 511, 526, 530, 537, 539, 552, 559, 561, 564, 567, 568, 576, 581, 591, 592, 596, 597, 600, 601
 - QISR2*, 548, 549, 592
 - QISR3*, 543, 545, 592
 - QISR4*, 517, 518, 539, 543, 592
 - QISR6*, 582, 592
 - BAPI, 660
 - Web Dynpro*, 461
 - Barcode, 25, 361
 - 1D*, 362
 - 2D*, 362
 - property*, 362
 - BAS, 157
 - Base64, 261
 - Baseline shift, 195
 - Basic Customizing, 564
 - BDS, 576, 577, 601
 - Binary format, 153
 - Binding
 - explicit*, 364

- expression*, 196, 207, 208, 209, 217, 295
 - implicit*, 364
 - Binding expression
 - \$*, 197
 - \$record*, 197
 - B_NOTIF, 505
 - Body image, 223
 - BSP, form, 577
 - BUA, 505, 508
 - Bundling, 263
 - bundling mode*, 235
 - Business Address Services -> see BAS, 157
 - Business Application Programming Interface
 - > see BAPI, 660
 - Business Communication Services, 605, 607
 - Business Document Service -> see BDS, 576, 601
 - Business logic, 377
 - Business object
 - BUS7051*, 575, 600
 - Business process, 40
 - optimization*, 40
 - Business Server Pages -> see BSP, 577
 - Business Unit Analyst -> see BUA, 505
 - Business Workplace, 514
 - Button, 332, 344
 - Send*, 423
- ## C
-
- CA, 696
 - cacert.pem, 707
 - Cache, 75, 101, 102
 - Caching, 642
 - Calculation, 348
 - Calendar window, 326
 - Call, 237
 - direct*, 508
 - dynamic*, 237
 - indirect*, 508
 - CA_private.p12, 709
 - Caption, 187
 - dynamic*, 341
 - CA_public.der, 698, 703, 710, 725, 726
 - CellEditor, 440
 - Certificate, 653
 - Certificate entry, 710
 - Certificate list, 702
 - Certificate revocation list, 98
 - Certification, 98, 101, 627, 653
 - Certification authority, 696
 - create*, 697
 - public key*, 698
 - trusted*, 698
 - Certification Authority -> see CA, 696
 - Change option, 642
 - Character format, 148
 - Characteristic, 530, 535, 551, 554, 555, 575, 590
 - uncategorized*, 536, 556
 - Character spacing, 195
 - Check, 124, 569, 582
 - request data*, 566, 598
 - Checkbox, 334
 - appearance*, 334
 - state*, 334
 - style*, 334
 - Check script syntax, 345
 - Choice
 - subform set*, 300
 - Choice section, 302
 - Class
 - CL_FP*, 637
 - CL_FP_PDF_OBJECT*, 636
 - CL_FP_WB_HELPER*, 641
 - CL_ISR_DATA*, 590
 - CL_WD_RUNTIME_SERVICES*, 486
 - CL_FP_FEATURE_TEST, 112
 - CL_FP_INBOUND_HANDLER, 608, 624
 - Click event, 685, 687, 689
 - Client, 149, 374
 - certificate*, 415
 - Client certificate, 700
 - CL_ISR_DATA, 590
 - Code
 - initialization*, 125
 - Collection, 614
 - create*, 614
 - Collection editor, 615
 - Collection list, 614
 - Comment, 31, 515, 558
 - Communication
 - PDF object and ADS*, 55

- Communication type, 608
 - Communication user, 665
 - Comparison
 - Adobe Reader and Acrobat Professional*, 30
 - Component, 374
 - ComponentController, 376
 - ComponentInterface, 380
 - Component, use, 581
 - Condition, 170, 308
 - Conditional page break, 307
 - ConfigPort_Document, 83
 - Configuration, 110
 - SSL*, 717
 - Connection, 637
 - SSL connection*, 652
 - standard connection*, 637
 - Console, 347, 348
 - activate*, 347
 - console.println*, 348
 - console.show*, 348
 - Consumer proxy, 90
 - Contact management
 - Web Dynpro*, 456
 - Contact partner address, 159
 - Container, 375
 - Container element, 546
 - ContainerFoundation_JS
 - version*, 418
 - Content area, 210
 - Context, 115, 134, 375, 555, 610
 - Context attribute, 375
 - Context editor, 375
 - Context element
 - DATA*, 441
 - Context evaluation, 286
 - Context field, 554
 - Context node, 375
 - ValueSet*, 432
 - Control command, 151
 - Control level, 165
 - Controlling, 503, 530, 550
 - Controlling area, 548
 - Control type, 332, 616
 - submit*, 616
 - Conversion routine, 169
 - Correspondence
 - manual*, 533
 - Cost center, 548
 - monitor*, 508
 - Cost factor, 548
 - Cost object, 548
 - Costs
 - estimated*, 548
 - request*, 503
 - Country, 120, 257
 - Credential, 33, 86, 98
 - CUKY, 127
 - CURR, 127
 - Currency, 127
 - Currency field, 127
 - Customer namespace, 531
 - Customizing
 - advanced*, 537
- ## D
-
- Data, 471
 - check*, 504
 - entry*, 24
 - extraction*, 650
 - formatting*, 233
 - global*, 124
 - initialization*, 510, 519, 552, 565
 - integrity*, 40
 - node*, 144
 - output*, 24
 - pattern*, 307
 - record*, 165
 - retrieval*, 230
 - scenario*, 544, 550
 - storage*, 576
 - structure*, 567
 - transfer*, 263
 - view*, 178, 204, 205
 - DATA, 589
 - Form Builder*, 441
 - Database access, 230
 - Data binding, 145, 195, 204, 205, 209, 364, 554, 555, 556
 - correct*, 364
 - define*, 195, 197, 205
 - explicit*, 217, 221
 - implicit*, 217, 221

- managing*, 225
- name*, 196
- performance comparison*, 221
- relative*, 364
- repeating data*, 209
- setting*, 225
- subform*, 206
- table*, 295
- Data connection, 179
 - define*, 669
 - OLEDB-based*, 180
 - WSDL-based*, 180, 659, 665, 672, 693
 - XML schema-based*, 180
- Data Dictionary, 528, 535
- Data quantity, 143
- dataSource, 390
 - automatic binding*, 391
- Data structure, 179, 221
 - XML-based*, 179
- Data type
 - CUKY*, 127
 - CURR*, 127
 - QUAN*, 127
 - UNIT*, 127
- Data view, 613
- Data volume
 - limitation*, 366
- Date
 - display*, 257
- Date field, 326
 - formatting*, 303
 - output formatting*, 303
- DDIC interface, 555
 - online scenario*, 425
 - Web Dynpro*, 399, 408
- Decimal field, 325
 - usage*, 325
- Decimal place, 127
- Default binding, 364
- Default value, 504
- Definition
 - global*, 124
- Design view, 182, 183
- Desktop email application, 629
- Destination, 72, 80, 106, 719
 - ADS_SSL*, 727
 - cache*, 101
 - create*, 719
- DestinationSi, 723
- Detail
 - general*, 529, 530, 594
- Development time
 - consider*, 57
- Device
 - mobile*, 30
- Device type, 246
- Dialog
 - document property*, 367
 - print*, 36
- Dialog box, 235
- Dictionary structure, 624
- DIR_GLOBAL, 88
- Display
 - error and message*, 347
- Display formatting, 302, 306
- Display mode, 513
- Display pattern, 305, 306, 307, 329
 - complex*, 329
- displayType
 - default value*, 427
- Docparams, 314
- Document
 - creation*, 229
 - signature*, 336, 337
- Documentation, 132, 172
 - form*, 172
 - form interface*, 132
- Document services
 - font manager*, 105
 - trust manager*, 98, 107
- Document Services Trust Manager, 88
- Document style, 667
- Domain, 169
- Dots per Inch -> see DPI, 25
- Downward compatibility, 111
- DPI, 25
- Drag and drop, 147
- Dropdown
 - <sap-vhlist>*, 438
- Dropdown list, 337
 - dynamic*, 438
 - dynamic population*, 339
 - Numbered*, 435
 - ZCI-based*, 434

Dropdown list box, 566, 571, 598
 Duplex printing, 26, 309, 310
 configuration, 310
 Dynamic, 314
 Dynamic texts, 152
 Dynamic XML form, 317

E

Easy Cost Planning, 548
 Edit pattern, 327
 Email address, 353
 Email inbound, 608, 623
 configuration, 608
 Email message, 352, 629
 Employee Self-Services -> see ESS, 505
 Enabled, 389
 default value, 407
 Endpoint type, 661
 Enhancement package (EHP), 17
 Entry type, 530, 534, 578
 Error, 124
 Error analysis, 254
 Error code, 260
 Error handling, 232, 234, 574
 Error message, 258, 511, 574
 Error text, 258
 ESS, 505
 Evaluation, 519
 Event, 342, 504, 566, 598
 generate, 559
 onSubmit, 447
 EVENT, 569
 Event type linkage, 544
 EV_FUNCNAME_INBOUND, 624
 Exception, 123, 637
 class-based, 234
 Exception class, 637
 Exclusion group, 335
 Expand
 to fit, 319
 Export binding, 673
 Export parameter, 122
 Express Planning, 532
 Extensible markup language -> see XML, 48

F

F4 help -> see Input help, 135
 Factory class, 637
 FALSE, 160
 Field
 floating, 290, 291
 initialization, 569
 ISR_NEW_LONG_TEXT, 537
 lock, 225
 UI attribute, 554
 Field layout control, 574
 Field list, 138
 Field symbol, 124
 File, 129
 File attachment, 254, 658
 File attachment comment, 31
 FileDownload, 471
 supply function, 479
 fileName, 471
 Fillable, 314
 Firefox
 ACF, 411
 ZCI, 411
 Flight booking model, 119, 242, 606
 Flight database, 34
 Flowed, 207
 Folder, 171
 Font, 105, 366
 Foreign key dependency, 163
 Form, 115, 386, 519, 523, 535, 551, 555,
 556, 559, 561, 571, 576, 577, 588, 601,
 602
 accessible, 28
 activate, 213
 API, 640
 bundle, 89
 check, 169
 context, 552, 554, 558
 copy, 217
 definition, 522, 527
 design area, 182
 documentation, 281
 embedding, 414
 event, 552, 569
 flow logic, 530, 539, 564, 565

- form object*, 115
- fragment*, 214
- generated*, 524
- hierarchy*, 221
- interactive*, 503, 504, 530, 579
- interface*, 611
- ISR_FORM_STANDARD*, 554
- language*, 230, 238
- layout*, 519, 522
- number of*, 240
- online and offline use*, 414
- output*, 229, 240
- parameter*, 553
- process logic*, 511
- property*, 316, 343
- runtime environment*, 232
- selection*, 230
- simple*, 532
- storage*, 674
- table data*, 517
- use*, 40
- Formatting, 120, 195, 230, 241, 263
 - date*, 241
 - date field*, 241
 - number fields*, 230, 241
- Form Builder, 54, 115, 551, 554, 555, 556
 - Web Dynpro*, 391
- FormCalc, 49, 342, 370, 556
- Form data, 578, 591
 - dropdown list*, 437
 - table*, 444
 - Web Dynpro*, 437
- Form designer, 24
 - task*, 24
- Form download
 - AttachFileToResponse*, 486
 - FileDownload*, 471
 - LinkToURL*, 470
- Form field, 180
 - access*, 356
 - align*, 190
 - arrange*, 186
 - button*, 332
 - changing the size*, 185
 - checkbox*, 334
 - date/time field*, 326
 - decimal field*, 325
 - document signature field*, 336
 - dropdown list*, 337
 - floating*, 290
 - image*, 184
 - image field*, 334
 - import*, 556
 - interactive*, 180
 - interactive PDF form*, 330
 - line*, 185
 - list box*, 337
 - name*, 191, 192
 - password*, 689
 - password field*, 331
 - radio button*, 335
 - ready-for-input status*, 322
 - rectangle*, 185
 - reference*, 356
 - rich text field*, 210
 - simultaneous editing*, 189
 - static*, 180
 - text*, 184
 - text field*, 187, 318
- Form fields
 - position*, 298
- Form hierarchy, 48, 178, 201
- Form interface, 119, 523, 535, 551, 554, 555
 - automatic update*, 402
 - create*, 668
 - Web Dynpro*, 391
 - Web Dynpro context*, 399
- Form object, 215
 - add*, 215
 - library*, 214
 - reuse*, 213
 - use*, 216
- Form properties
 - default*, 366
- Form routine, 126
- Form template, 386, 618
 - control*, 414
 - create*, 182, 668, 684
 - repository*, 58
 - test*, 690
 - ViewDesigner*, 395
 - Web Dynpro*, 391

- Web Dynpro component*, 400
- ZCI integration*, 411
- Form upload
 - confirmation of input*, 488
 - online review*, 492
 - Web Dynpro*, 488
- FORM_VIEW, 568
- FP_CHECK_DESTINATION_SERVICE, 82
- FPCONNECT, 88, 95
- FP_FUNCTION_MODULE_NAME, 233, 624
- FP_ICF_DATA_XXX, 721
- FP_JOB_CLOSE, 239
- FP_JOB_OPEN, 234
- FP_PDF_TEST_00, 74
- FpPrintRequestService, 89
- FpQueueAdminService, 89, 93
- FP_TEST_03, 36, 88
- FP_TEST_03_TABLE, 36
- FpXfaRequestProcessorBean, 103
- Function, 540
- Function group, 679
 - create*, 679
 - QISR5*, 591
- Function module, 232, 678, 679
 - call pattern*, 238
 - create*, 679
 - generated*, 232
 - interface*, 121, 642, 644
 - ISR_PROCESS_EVENT*, 551, 576, 588, 601, 602
 - remote-enabled*, 679

G

- Generation, 243, 552, 555
 - Adobe form*, 554
 - characteristic*, 556
- GET_HIDE_TOOLBARS, 451
- GET_INSTANCE, 623, 624
- Gradient, 365
- Gradient fill, 365
- Graphic, 153
 - content*, 156
 - format*, 157
 - node*, 153

- reference*, 155
- type*, 154
- Grid, 190
 - grid line*, 190
- gs_docpar-fillable, 619
- Guided Procedures, 54

H

- HANDLE_PDF, 623
- Header, 295
- height, 398
- Hierarchy, 115, 178
 - add*, 200
- HTTP/S
 - Web Dynpro integration*, 415
- Hub, 111
- Hyphenation, 223
- Hyphenation dictionary, 223

I

- ICF
 - ICF Recorder*, 258
 - ICF service*, 72, 77, 106
- ICM, 708, 716
 - restart*, 708, 716
- Identity management, 719
- IF_FP_OFFLINE, 609
- IF_FP_OFFLINE~GET_INSTANCE, 623, 624
- IF_FP_OFFLINE~HANDLE_PDF, 623, 624
- IF_FP_PDF_OBJECT, 636
- IF_FP_PDF_USAGE_RIGHTS, 645
- Image, 364
 - resolution*, 364
- Image field, 334
 - file format*, 334
- IMG activity, 529
- Implementation
 - interactive offline scenario*, 21
 - online scenario*, 21
 - print output*, 20
- Import binding, 673
- Import parameter, 119, 120, 610

- Information
 - additional*, 369
- Initial display, 658
- Initialization, 125
- Initialization code, 167
- Initial View, 733
- Input help, 135, 504, 511, 519, 560, 566, 570
 - context attribute*, 428
 - data binding*, 430
 - dropdown list*, 429
 - Dropdown list*, 435
 - enhanced*, 559, 571
 - programmed*, 537, 552, 561
 - visibility*, 428
- Input screen, 504
- Input validation, 349
- Installation
 - Adobe LiveCycle Designer*, 110
- Instance manager, 358
 - AddInstance*, 358
 - MoveInstance*, 358
 - RemoveInstance*, 358
- Integration
 - architecture*, 46
 - indirect*, 54
- Integrity, 653
- Interaction event, 342
- Interaction layer, 377
- Interactive, 639
- InteractiveForm, 389
 - onSubmit*, 447
- Interchange Text Format -> see ITF, 153
- Interface, 115, 610
 - ABAP Dictionary-based*, 117, 119, 555
 - DDIC*, 555
 - Smart Forms-compatible*, 117, 131
 - structure*, 552
 - XFP*, 285
 - XML*, 551
 - XML-based*, 555
 - XML schema-based*, 117, 129
- Internal order, 548
- Internal Service Request -> see ISR, 501
- International Organization for Standardization
 - > see ISO, 23
- Internet Communication Manager -> see ICM, 708
- Internet Explorer
 - ACF*, 411
 - ZCI*, 411
- INT_SERVICE_REQUEST_CHECK, 565, 566, 597, 598
- I_QMEL, 505
- IQS8, 518, 544
- IQS22, 516, 517, 544, 558
- ISO, 23
- ISR, 501
 - architecture*, 575
 - basic Customizing*, 527, 529, 537, 550, 551, 558, 571
 - Customizing*, 555, 560, 591
 - general data*, 553
 - general field*, 536
 - interface*, 592
 - Launchpad*, 508, 581, 583, 592
 - library*, 552, 554, 556, 559, 571
 - scenario*, 72, 501, 518, 529, 531, 547, 554, 555, 579, 580, 594, 595, 601
 - scenario group*, 586
 - status overview*, 513, 516, 537, 567, 591
 - toolset*, 575, 592, 594
 - wizard*, 520, 526
- ISR_ACTIVITY_EXECUTE, 539
- ISR_ACTOR_FOR_ROLE_GET, 591
- ISR_CHECK_AT_POST, 550
- ISR_GENERAL_DATA_GET, 590
- ISR_NEW_LONG_TEXT, 537
- ISR_NOTIF_REVERSE, 591
- ISR_PROCESS_EVENT, 588
- ISR_REQUEST_GET_RFC, 591
- ISR_SCENARIO_PARAMS, 580
- ISR_SPECIAL_DATA_GET, 572, 590
- ISR_SPECIAL_DATA_TO_STRUC, 573, 591
- ISR_STRUC_TO_SPECIAL_DATA, 591
- Item row, 163
- ITF, 153
- iView, 581
- iXML
 - data extraction*, 490

J

JavaScript, 48, 99, 342, 556, 659
console, 347
Developing Acrobat applications using JavaScript, 371
JavaScript for Acrobat API reference, 370
memory consumption, 365
strict scoping rule, 365

Java Server Pages -> see JSP, 577

Java system properties, 718

Job profile, 113, 236, 657, 695, 730
application example, 730
areas, 732
categories, 731
category overview, 731
create, 731
help, 734
manage, 731
management, 730
modify, 733

JSP, 577

JSP form, 577

K

Kerning, 195

Key Storage Views, 709

Knowledge Provider, 576, 601

L

Label printer, 247

Language, 120, 257, 281

Layout, 375, 576
element, 554, 570
information, 554
property, 557
type, 504, 552

Layout problem
solution, 299

Lead management, 18

legacyEditingEnabled, 426

Length restriction, 318

Library, 180
barcode library, 180, 361, 362
create, 214
Web Dynpro ActiveX, 423
Web Dynpro Native, 423

License, 96

License status, 97

Licensing, 96

LinkToURL, 470
FileDownload, 473

List box, 337

List entry, 337
define, 339

List item, 339

Locale, 257, 640

Lock icon, 142

Log configurator, 108

Logical Port, 723

Logo, 153

Log viewer, 109

Long text
field, 511, 518, 537, 558
formatting, 537

Loop, 163

LP01, 37

LPD_CUST, 542

M

Manager Self-Services -> see MSS, 505

Mandatory field, 557, 574

Markup, 31

Mass data entry, 27

Mass output, 264

Mass printing, 263

Master data monitor, 510

Master page, 182, 209, 299, 363
placement, 310
usage, 299, 309

Master pages, 309

Master password, 74

Memory requirement, 124, 143

Message box, 346

Method handler, 425
hide the Adobe Reader toolbar, 451

LegacyEditingEnabled, 426
 MIME object
 namespace, 406
 MIME Repository, 406
 MIME type, 157
 MIME-type
 Application/pdf, 471
 FileDownload, 471
 MODE, 568, 599
 Model View Controller -> see MVC, 374
 Modification, 125
 Monitoring, 93, 107, 273
 Mozilla Firefox -> see Firefox, 411
 MSS, 505
 MVC
 approach, 374

N

Namespace, 688
 Navigation, 122, 246
 PDF file, 407
 view, 474
 NetWeaver Administrator, 82, 109
 Network speed, 263
 Node, 375
 Node property
 active, 143
 generated, 142
 inactive, 143
 Node type, 140
 Notification, 503, 527, 537, 544, 566, 577
 action, 522
 notification number, 542, 543, 579, 580, 589, 590
 notification transaction, 534, 538, 542
 notification type, 530, 533, 534, 538, 550, 579, 595
 processing, 522
 Number format, 257
 Number range assignment, 534
 Numeric value
 formatting, 304
 output formatting, 304

O

Object
 palette, 613
 static, 225
 Object category
 BUS7051, 544
 Object editor, 222
 Object frame, 185, 194
 Object Navigator, 115
 Object palette
 pagination, 307
 submit, 616
 Object Value Selector -> see OVS, 557
 Occasional user, 503, 504
 Official certification authority, 696
 Offline handler, 609
 create, 609
 Offline mode, 502
 Offline scenario, 41, 42, 43, 45, 46, 388, 469, 605, 636, 643, 647
 form download, 470
 implementation, 469
 security, 627
 upload procedure, 495
 Web Dynpro use, 496
 Online form, 502
 Online scenario, 41, 43, 45, 46, 387, 410
 checklist, 416, 427
 contact management, 456
 page size, 417
 value help, 428
 onSubmit
 data change, 450
 difference, 450
 event, 449
 OpenSSL, 696
 Optimization
 business process, 40, 44
 Original language, 137, 281
 OR operator, 162
 Outline paragraph, 148, 151
 Output device, 235, 245
 virtual, 287
 Output pattern, 305

Output request, 245
 Overflow footer, 309
 Overflow header area, 309
 Overhead cost controlling, 548
 OVS, 557, 559, 571

P

Package assignment, 115, 117, 136, 137
 Page break, 296, 307
 conditional, 307, 310
 edit, 307
 notes, 309
 usage, 296
 Page break, 307
 Page numbering, 299
 Page protection, 148
 Pages, 239
 Page set, 309, 310
 Pagination, 309, 310
 Palette
 accessibility, 367
 border palette, 189
 drawing aids palette, 190
 expand layout to fit, 321
 font palette, 291
 Font palette, 194
 layout palette, 189
 library palette, 180
 object palette, 180
 paragraph palette, 305, 319, 320
 paragraph palette, 194
 PDF structure, 228
 report palette, 346
 report palette, 219
 tab order, 227
 warnings, 226
 Paper form
 barcode, 362
 online scenario, 428
 Paper tray control, 26
 Paper type, 363
 Paragraph format, 148
 Paragraph palette, 305
 Paragraph properties, 148
 Parallelization, 89, 91, 235, 270

Parameter
 default value, 568
 Part list, 245
 Partner, involved, 522
 Partnership
 Adobe/SAP, 17
 Part number, 268
 pass phrase, 697
 Password, 688
 field, 689
 Password field, 331
 Pattern, 126, 307, 327
 category num, 305
 date category, 304
 dialog, 304, 327
 time category, 306
 zero value, 330
 Pattern category, 330
 Pattern dialog, 304, 327
 Payload, 279
 PCL, 247
 PCL 5, 25
 PCR, 578
 PDF, 23, 386, 535, 576, 578
 accessible form, 366, 367, 368
 comment and markup, 31
 data container, 45
 digital signature, 31
 display, 34
 dropdown list, 432
 dynamic, 415
 dynamic form, 28
 dynpro file, 405
 file attachment, 31
 file format, 23
 hide toolbar, 453
 Magnification Mode, 733
 manipulation module, 88, 98, 102, 107
 offline scenario, 478
 PDF/A, 24
 prepare preview, 316
 preview, 182, 192, 316
 return, 270
 table, 439
 text field example, 420
 using the preview, 192
 value help, 428

- Web Dynpro*, 408
- PDF document
 - additional information*, 254
 - create*, 639
 - data extraction*, 650
 - dynamic*, 642
 - fill out*, 648
 - interactive*, 253, 642
 - process*, 647
 - sign*, 648
- PDF form, 678
 - interactive*, 27, 29, 38, 39, 312, 317
 - save*, 31
 - static*, 28
 - template*, 386
- PDF object, 53, 55, 56, 65, 98, 636
 - ABAP, 72
 - EXECUTE, 638, 640, 650, 652
 - GET_CERTIFICATION, 656
 - GET_DATA, 650
 - GET method, 638
 - GET_PDF, 640
 - GET_SIGNATURES, 652
 - Java, 82
 - RESET method, 638
 - SET_CERTIFICATION, 654
 - SET_DATA, 640
 - SET_DOCUMENT, 650, 652, 654, 655
 - SET method, 638
 - SET_TASK_EXTRACTDATA, 650
 - SET_TASK_GETCERTIFICATION, 656
 - SET_TASK_GETSIGNATURES, 652
 - SET_TASK method, 638
 - SET_TASK_RENDERPDF, 640
 - SET_TEMPLATE, 640
 - SET_USAGERIGHTS, 646
- pdfOnly, 404
- pdfSource, 390
 - change*, 407
 - form template*, 410
 - offline scenario*, 494
 - online scenario*, 495
 - performance*, 410
- PDF structure, 228
- Performance, 107, 143, 240, 263, 363
 - attachment*, 455
 - dropdown list*, 438
 - online/offline form*, 414
 - pdfSource*, 410
 - table*, 446
 - table column*, 444
 - table row*, 444
 - Web Dynpro*, 415
- Personnel Change Request -> see PCR, 578
- Person number, 159
- PFCG, 76
- Placement
 - master pages*, 310
 - subforms*, 309
- Placement restriction, 310
- Platform independence, 374
- Plus sign, 318
- PoolMax, 102
- Portable document format -> see PDF, 23
- POST_NOTIF_POSITION, 566
- PostScript, 23, 25, 247
 - Level 2*, 186, 365
- Pretty printer, 126
- Preview data, 198
 - generate*, 199, 314
- Price
 - ISR*, 548
- Price list, 548
- Print, 243
- Print dialog, 241, 243
- Printer, 235
- Printer Command Language -> see PCL, 25
- Printer language, 23, 25, 247, 253
- Print form
 - PDF-based*, 24
 - provided*, 18
- Print job, 234
- Print output, 34, 38, 243
 - example*, 34
 - local*, 242
 - preview*, 37
 - server-based*, 243
- Print preview, 37, 235, 239, 243
- Print program, 230, 314, 618
 - data logic*, 230
 - selection screen*, 230
- Print scenario, 385
- Private key of the certification authority, 709
- Processing speed, 263

Processor, 503, 506, 517, 518, 519, 530, 534,
 538, 541, 542, 543
assign, 545
current, 591
determine, 519, 530, 543, 592

Program termination, 123

Property, 389
dynamic activation, 338, 339

Public key of certification authority, 703

Public sector, 533

Q

QISRACTIVITY, 543

QISRCONF, 527

QISRSCENARIO_OVS, 571

QISRSGENERAL_PARAM, 536

QISR_SM29, 534

QMNUM, 543

QUAN, 127

Quantity field, 127

Queue, 275

R

RABAX, 445

Radio button, 335
group, 335

Raw data interface, 285

Read condition, 167

ReaderRights credential, 64, 85, 107

ReaderRights Credential, 85

readOnly, 389

Read only, 323

Ready-for-input status, 325, 350, 557, 574
dynamic, 350
field, 558
form field, 322
procedure, 325

Recording, 260

Reference, 121, 641

Reference field, 127

Relational operator, 162

Replacement language, 120, 284

REPLACE_THIS, 436

Report
QISRACTIVITY, 543

Report Launchpad, 519, 542

Report palette, 346
warnings, 346

Request, 542
Adobe form, 551
change, 590
costs, 503, 530
form, 507, 519, 535, 565, 575, 581, 600
read, 590
start, 589
submit, 589
system user, 527

Request field
specific, 572

Request parameter, 659

ResolveNode, 357

Resource distribution, 271

Response parameter, 659

Responsibility, 547

Revenue management, 533

Reverse, 567, 591

REVERSE_NOTIF_EXTERNAL, 567

RFC destination, 72, 73, 88, 106

Rich text field, 210

Roadmap, 509, 579

Role, 503, 519, 542
applicant, 508
approver, 514
processor, 517
technical, 503

Root certificate, 99, 697

Root node, 138

ROOTUIELEMENTCONTAINER, 380

RSA private Key, 697

Runtime, 124, 556
consider, 58

S

SA38, 82

Sample data, 179

Sample form
FP_TEST_03, 36
FP_TEST_03_TABLE, 36

- Sample program
 - FP_TEST_03*, 35
 - FP_TEST_IA_01*, 38
- SAP_ADSCALLER, 68
- SAPBC_DATA_GENERATOR, 34, 242
- SAP Business Suite, 17, 18
- SAP Business Workflow, 55, 518, 575
- SAP ERP, 504, 533, 559, 562
- SAP GUI, 110
- SAP Help Portal, 101
- SAP Interactive Forms by Adobe, 17, 23, 579
 - VPN/Web Dynpro integration*, 418
 - Web Dynpro*, 415
 - Web Dynpro integration*, 415, 498
- SAP Management Console, 716
- SAP NetWeaver, 17, 575
 - solution extension*, 17
- SAP NetWeaver Administrator, 65, 730
- SAP NetWeaver application server
 - ABAP*, 63
 - Java*, 63
- SAP NetWeaver Application Server
 - Java*, 53
- SAP NetWeaver Business Process Management, 54
- SAP NetWeaver Developer Studio, 51
- SAP NetWeaver Portal, 506, 542, 580
- SAP Note 1122142, 735
- SAPscript, 17, 26
 - include text*, 146, 149
- SAP Service Marketplace, 49, 64
- SAP Smart Forms, 17, 26, 131
 - text module*, 146, 212
 - Web Dynpro interface*, 399
- SAP system ID, 98
- sap-vhlist, 438
- S_BDS_DS, 505
- Scaling, 195
- Scanner, 25
- Scenario, 523, 531, 555
 - activated*, 527
 - characteristic*, 528
 - Create Cost Center*, 508
 - Customizing*, 529, 551
 - definition*, 527
 - grouping*, 507
 - interactive*, 29
 - key account*, 388
 - test*, 550
- SCENARIO_AFTER_SUBMIT, 566
- SCENARIO_FINAL_BEFORE_SUBMIT, 566
- SCENARIO_PROCESS_USER_COMMAND, 566
- SCENARIO_SET_ADDITIONAL_VALUES, 566, 598
- SCENARIO_SET_FORM_VIEW, 566
- Schema, 134, 644
- Screen element, 556, 560
- Script editor -> see Adobe LiveCycle Designer, 177
- Script editor-> see Adobe LiveCycle Designer, 344
- Scripting language, 48
- Scripting Object Model -> see SOM, 357
- Script language, 342
- Script program, 341, 342, 365
 - abortion of execution*, 346
 - access to data nodes and form fields*, 356
 - calculation*, 348
 - controlling of the visibility*, 350
 - controlling the ready-for-input status*, 350
 - debug*, 346
 - validation*, 349
- Script programming, 48, 553, 554, 558
- SDN, 64, 369, 734
 - Web Dynpro area*, 498
- SE38, 35, 674
- SE80, 115
- Search help -> see Input help, 135
- Section, 302
- Secure connection, 695
 - create*, 727
- Secure Sockets Layer -> see SSL, 101
- Service
 - create*, 682
- Service costs, 520, 523, 548
- Service link, 508
- Service request, 502, 506, 530, 535, 538, 543, 548, 589, 590
- Service_SSL, 710
- SET_HIDE_TOOLBARS, 451
- SFP, 50, 115, 254, 639
- Short dump, 445
- Short text, 282

Index

- SICF, 77, 79, 259
- Sign, 648
- Signature, 98, 101, 651
 - determine*, 651
 - digital*, 40, 651
 - digital Web Dynpro integration*, 498
 - signature field*, 648
- Signature field, 614
- Simplex printing, 26, 310
- Single record node, 167
- Single test, 170, 237
- SM30, 88
- SM59, 73, 727
- SMICM, 709
- SOA management, 89
- SOA Manager, 276
- SOAP header, 688
- SOAP object, 659, 678, 685, 688, 693
 - connect*, 686, 688
 - request*, 688, 689
 - usage*, 685
 - use*, 678
- Software components, 46
- SOM
 - expression*, 357
- Sorting, 167
- Source, 129
 - generated*, 129
- Source code editor, 624
- Source language, 281
- SP01, 244
- SP02, 244
- SPAD, 246
- Special character, 148, 152
- Spelling, 223
- Spool administration, 246
- Spool request, 236, 243, 244
 - number*, 240, 245
- Spool system, 244
- SSL, 101
- SSL client identity, 698
- SSL connection, 101, 652, 698
- Standard
 - library*, 613
- Standard application, 532
- Standard connection, 637
- Standard text, 149
- Statement pattern, 126
- Status, 143, 523, 527, 580, 591
- Status field, 653
- Status overview, 581
- Step
 - completed*, 513
 - manual*, 40
 - process form*, 509
 - send*, 512
- Storage mode, 250
- Strict scoping rule, 365
- Structure, 141
 - QISRSGENERAL_PARAM*, 536
- Structuring, 172
- STRUSTSSO2, 698
- Style, 148
- SU01, 76
- Subform, 199
 - content*, 290, 296
 - data binding*, 207
 - edit*, 290
 - flow direction*, 207
 - instance*, 209
 - instance manager*, 358
 - management*, 290
 - name*, 201
 - nested*, 290
 - nesting depth*, 364
 - repetition*, 209
 - use*, 200, 203
- Subform set, 300, 301
 - choice*, 300
 - normal*, 300
- Subitem, 163
- Submit process, 619
- Subtree, 138
- Subwizard, 520, 527
- Supply function
 - FileDownload*, 486
 - PDF generation*, 474
- S_USER_GRP, 505
- Syntax check, 345
- Syntax error, 345, 346
 - remedy*, 346
- System event, 342
- System field, 140

T

Tab

- Draw*, 184, 185
- Execute*, 672
- Field*, 188
- Formatting*, 223
- Import/Export Bindings*, 673
- Layout*, 182
- Pagination*, 307, 310
- Preview PDF*, 192
- Subform*, 207

Tabbing

- order*, 190

Table, 163, 292, 298

- Adobe LiveCycle Designer help*, 292
- CellEditor*, 440
- choice section*, 302
- column*, 358
- column heading*, 293
- data binding*, 295
- data row*, 295
- display*, 295
- dynamic*, 357, 358, 359
- empty cell*, 296
- format*, 293
- FPCONNECT*, 88
- header*, 295
- LeadSelection*, 447
- master/detailed scenario*, 447
- navigation*, 292
- nested*, 295
- page*, 446
- page break*, 298
- row*, 298, 444
- section*, 302
- selection*, 447
- structure*, 294
- troubleshooting*, 296
- use*, 292
- wizard*, 440

Table name, 631

Table type

- generic*, 552

Tab order, 227

Tabulator, 148

Tag structure, 366

Target version, 226

Tax management, 533

Template, allocation, 548

Template certificate, 710

templateSource, 390

- form template*, 391

Test application

- ISR_LPD_START*, 582

Test phase, 172

Test program, 638, 639

Text

- dynamic*, 146, 341

- format*, 193

- formatted*, 211

- lock*, 225

- multiple line*, 321

- source*, 146

- static*, 290, 363

- text ID*, 149

- text module*, 146

- text node*, 146

- text object*, 149

Text field, 187, 318

- number of characters*, 318

- resizable*, 320, 321, 322

- use*, 187

TextView, 381

Time field, 326

Time output

- output formatting*, 306

TOA_DARA, 250

Tool tip, 367

Total page count, 299

Trace, 254

Trace configuration, 278

Tracing, 276

Transaction

- APB_ISR_CUST*, 594

- IQS8*, 518, 544

- IQS22*, 516, 517, 544, 558

- LPD_CUST*, 542

- PFCG*, 76

- QISRCONF*, 520, 527

- QISRLIST*, 520

- QISRSCENARIO*, 529

- QISRSCENARIO_OVS*, 557, 560, 571

- QISR_SM29*, 534

SA38, 82
 SCOT, 619
 SE16, 631
 SE24, 610
 SE38, 35, 623, 628, 674
 SE80, 115
 SFP, 50, 115, 254, 639
 SICF, 77, 79, 259
 SM30, 88, 95
 SM59, 73, 727
 SMICM, 709
 SO50, 608
 SP01, 244
 SP02, 244
 SPAD, 246
 STRUSTSSO2, 698
 SU01, 76
 Translation, 281
 long text, 283
 short text, 282
 Transport, 243
 Transport authentication, 678, 683, 685
 Transport request, 117
 Tray control, 310
 TRM, 533
 TRUE, 160
 Trusted anchor, 98, 99, 653, 725
 TrustedCAs, 709
 Type, 124
 Type assignment, 122

U

UI attribute
 field, 554
 UI element, 375
 UNIT, 127
 Unit of measurement, 190
 Universal Worklist -> see UWL, 514
 Update
 automatic, 537, 539
 UPDATE TASK, 243
 Upload
 input, 488
 online review, 492
 URL parameter, 542, 581

Usability
 FileDownload, 479
 Usage
 direct, 580
 Usage right, 31, 32, 40, 85, 102, 317, 639, 645
 User administration, 66
 USER_COMMAND, 568, 569, 600
 User default, 241
 User entered
 optional, 323
 required, 323
 User Management Engine, 718
 User name, 688
 UWL, 514

V

Validation, 651
 Validation pattern, 327, 328
 Validation script, 349
 Validity, 653
 VALIDITY_DATE, 532
 Validity period, 532
 Value, 121
 ValueSet, 432
 Variable
 naming, 349
 Version, 532, 653
 Version information, 353
 View, 375
 ViewController, 375
 ViewDesigner, 375
 Virtual Private Network -> see VPN, 418
 Visibility, 350, 557, 574
 Visual Administrator, 108
 VPN, 418
 Web Dynpro integration, 418

W

Warning, 169
 wdDoInit
 context, 392
 PDF file, 405

- wdDoModifyView
 - method handler*, 426
 - Web Dynpro, 374, 575, 578, 581
 - ActiveX, 423
 - application*, 374, 377
 - client*, 374
 - component*, 374
 - context*, 401
 - Native*, 423
 - PDF standard*, 414
 - Web Dynpro ABAP, 72, 577
 - QISR_UI component*, 578
 - Web Dynpro application
 - QISR_FORM*, 542
 - QISR_UI_DISPLAY_ONLY*, 513
 - QISR_UI_FORM*, 508, 579, 580
 - QISR_UI_FORM_APPROVE*, 579, 580
 - QISR_UI_FORM_DISPLAY_ONLY*, 542, 580
 - QISR_UI_STATUSOVERVIEW*, 513
 - Web Dynpro component
 - integrated*, 508
 - ISR_LPD_START*, 585
 - QISR_UI*, 587
 - QISR_UI_DISPLAY_ONLY*, 517
 - QISR_UI_FORM*, 575, 600
 - Web Dynpro explorer, 374
 - Web Dynpro Java, 52, 72, 82
 - Component PCUI_GP~ISR*, 578
 - Web service, 659, 660
 - authentication*, 660, 688
 - automatic authentication*, 691
 - binding*, 673
 - call*, 31, 685, 692
 - configuration*, 664, 683
 - copy*, 664
 - create*, 660, 661, 678, 682
 - document style*, 667
 - execute*, 692
 - implement*, 685
 - test*, 674, 690
 - transport authentication*, 664
 - URL*, 683
 - usage note*, 660
 - usage tip*, 692
 - Web service proxy, 72, 82, 83, 106
 - Web services navigator, 68
 - WHERE condition, 165
 - Where-used list, 116
 - width, 398
 - Window, 376
 - WindowController, 376
 - WITH KEY condition, 167
 - Workbench object, 555
 - Workflow, 503, 519, 542, 543, 551
 - rule*, 545
 - Workflow item, 579
 - Workflow template, 544, 591
 - WS03100019*, 544
 - WS31000009*, 544
 - Worklist, 518, 519, 522, 524, 526
 - WSADMIN, 665, 683
 - WSCONFIG, 665
 - WSDL data connection
 - enhanced function*, 660
 - use*, 660
 - WSDL file, 665, 667, 683, 688
- ## X
-
- XDC Editor, 695, 734
 - help*, 735
 - XDC -> see XFA Device Configuration, 105
 - XFA, 25, 370
 - technology*, 651
 - XFA device configuration, 105, 256
 - editor*, 248
 - file*, 25, 248
 - XFP interface, 285
 - XHTML, 211
 - XML, 48, 129
 - data structure*, 48
 - form module*, 102
 - XML Forms Architecture -> see XFA, 25
 - XML interface
 - ViewDesigner*, 409
 - Web Dynpro*, 399
 - XML structure, 644
 - XXXCTC, 110

Z

ZCI, 411

ZCI Layout, 504, 552, 556, 557

ZCI script, 411

Adobe LiveCycle Designer Cache, 420

ADS, 412

control, 413

current version, 418

form template, 412

PC, 420

project member, 412

support package, 412

version number, 419

ZCL_IFBA_BOOK_OFFLINE, 612

Zebra label printer, 26, 363

barcode, 363

Z_IFBA_BOOK_DDIC_02, 313

Z_IFBA_BOOK_FORM_01, 182

Z_IFBA_BOOK_FORM_01_IMPL, 217

Z_IFBA_BOOK_FORM_02, 290

Z_IFBA_BOOK_FORM_03, 313

Z_IFBA_BOOK_FORM_03_EMPTY, 317, 343

Z_IFBA_BOOK_FORM_04, 343

Z_IFBA_BOOK_FORM_05, 356

Z_IFBA_BOOK_INTERACTIVE, 314

z_ifba_book_offline, 628, 631

Z_IFBA_BOOK_OFFLINE_FORM, 611

Z_IFBA_BOOK_OFFLINE_INTERFACE, 610,
611

Z_IFBA_BOOK_PRINT, 292

Z_IFBA_BOOK_SQUARE, 679

Z_IFBA_BOOK_TIME, 682

Z_IFBA_BOOK_WS_FORM_01, 668

Z_IFBA_BOOK_WS_FORM_01.PDF, 676

ZPL, 247

ZPL-II, 25