BC481

SAP Interactive Forms by Adobe in SAP Environments

SAP NetWeaver

Date _____

Training Center _____

Instructors _____

_____

Education Website _____

Participant Handbook

Course Version: 73
Course Duration: 2 Day(s)
Material Number: 50090874

An SAP course - use it to learn, reference it for work

Internal Use SAP Partner Only

Copyright

Trademarks

Disclaimer

Internal Use SAP Partner Only

About This Handbook

This handbook is intended to complement the instructor-led presentation of this course, and serve as a source of reference. It is not suitable for self-study.

Typographic Conventions

American English is the standard used in this handbook. The following typographic conventions are also used.

Type Style	Description
Example text	Words or characters that appear on the screen. These include field names, screen titles, pushbuttons as well as menu names, paths, and options. Also used for cross-references to other documentation both internal (in this documentation) and external (in other locations, such as SAPNet).
Example text	Emphasized words or phrases in body text, titles of graphics, and tables
EXAMPLE TEXT	Names of elements in the system. These include report names, program names, transaction codes, table names, and individual key words of a programming language, when surrounded by body text, for example SELECT and INCLUDE.
`Example text`	Screen output. This includes file and directory names and their paths, messages, names of variables and parameters, and passages of the source text of a program.
`Example text`	Exact user entry. These are words and characters that you enter in the system exactly as they appear in the documentation.
`<Example text>`	Variable user entry. Pointed brackets indicate that you replace these words and characters with appropriate entries.

Icons in Body Text

The following icons are used in this handbook.

Internal Use SAP Partner Only

Icon	Meaning
	For more information, tips, or background
	Note or further explanation of previous point
	Exception or caution
	Procedures
	Indicates that the item is displayed in the instructor's presentation.

Contents

Course Overview

In this course, you'll learn the key aspects of SAP Interactive Forms by Adobe in interactive scenarios.

What are the objectives and advantages of the Interactive Forms, what is the architecture required, what are the environments in which they can be integrated are questions which will be answered here.

This technical course put emphasis on development in several SAP environments (ABAP, ABAP OO and Web Dynpro ABAP) and provides development information, techniques and best practices with the Adobe LiveCycle Designer.

Target Audience

This course is intended for the following audiences:

- Project team members, developers and consultants who are responsible for business scenarios involving interactive forms

Course Prerequisites

Required Knowledge

- ABAP programming experience
- Web Dynpro ABAP experience
- UI interfaces programming knowledge

Recommended Knowledge

- Adobe LiveCycle Designer experience

Course Goals

This course will prepare you to:

- Understand the SAP Interactive Forms by Adobe capabilities and use cases in SAP environments
- Understand and develop SAP Interactive Forms by Adobe within an ABAP report
- Understand and develop SAP Interactive Forms by Adobe with Web Dynpro ABAP
- Understand and use the advanced features of the Adobe LiveCycle Designer

Internal Use SAP Partner Only

Course Objectives

After completing this course, you will be able to:

- Understand the SAP Interactive Forms by Adobe capabilities and use cases in SAP environments.
- Understand and develop SAP Interactive Forms by Adobe within an ABAP report
- Understand and develop SAP Interactive Forms by Adobe with Web Dynpro ABAP
- Understand and use the advanced features of the Adobe LiveCycle Designer

SAP Software Component Information

The information in this course pertains to the following SAP Software Components and releases:

- SAP NetWeaver 7.0

Unit 1

Product Overview

Unit Overview

This unit focuses on scenarios and motivations by giving ideas and overview on product objectives, advantages and key capabilities.

Unit Objectives

After completing this unit, you will be able to:

- Understand reasons for partnership between SAP and Adobe
- Understand the added value of SAP Interactive Forms by Adobe
- Compare SAP Interactive Forms by Adobe to other UI technologies
- Compare SAP Interactive Forms by Adobe to other SAP printing technologies
- Provide an overview of SAP Interactive Forms by Adobe integration in SAP NetWeaver
- Understand and explain the technical capabilities of SAP Interactive Forms by Adobe
- Understand and explain the technical integration of SAP Interactive Forms by Adobe in SAP technologies
- Understand the 3-year roadmap for SAP Interactive Forms by Adobe

Unit Contents

Lesson: SAP and Adobe Partnership

Lesson Overview

This lesson gives insights on motivation for SAP Interactive Forms by Adobe, past and future evolution.

Lesson Objectives

After completing this lesson, you will be able to:

- Understand reasons for partnership between SAP and Adobe
- Understand the added value of SAP Interactive Forms by Adobe

Business Example

You're considering the usage of SAP Interactive Forms by Adobe in a project. This lesson underlines the main aspects of SAP and Adobe partnership: the reasons and motivations for the product creation as well as objectives for this product.

Reasons for a partnership

- **Business Issues**
  - Overwhelming array of data and information
  - Geographic and organizational boundaries stop people from working together
  - IT silos don't support real-world processes and work styles
- **IT Issues**
  - Push from business to deliver more solutions, faster and cheaper
  - High cost and low ROI of existing information systems
  - Worker mobility, new audiences and communication channels

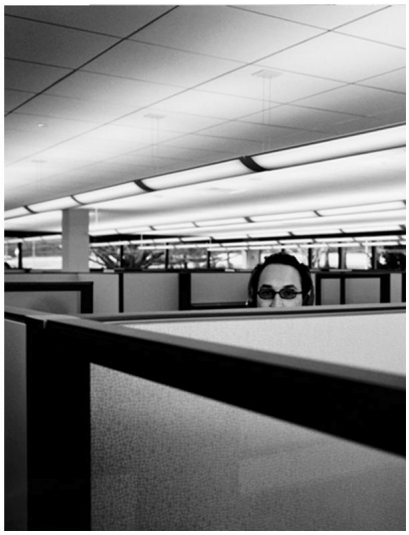


Figure 1: Users are contrained, overwhelmed and unproductive

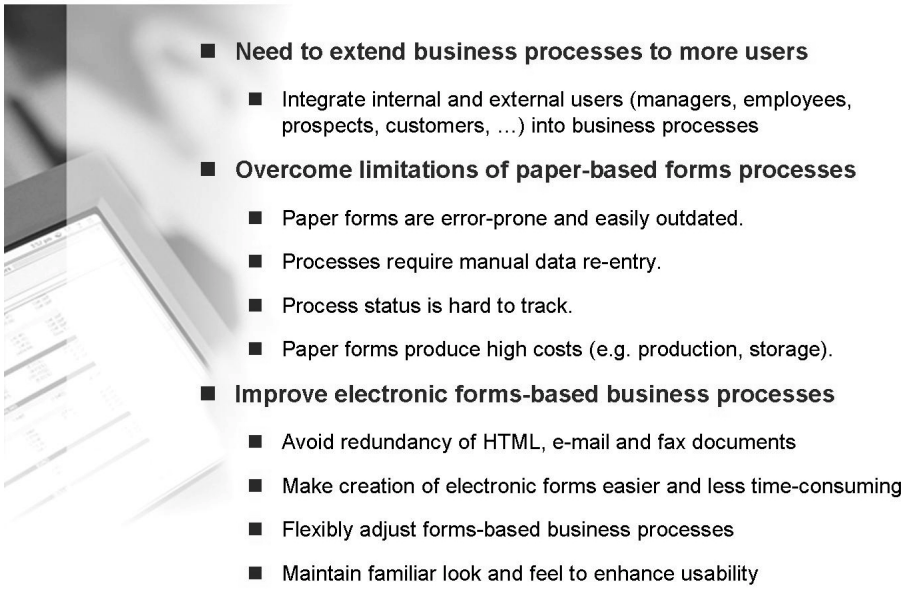


Figure 2: Requirements for forms-based processes

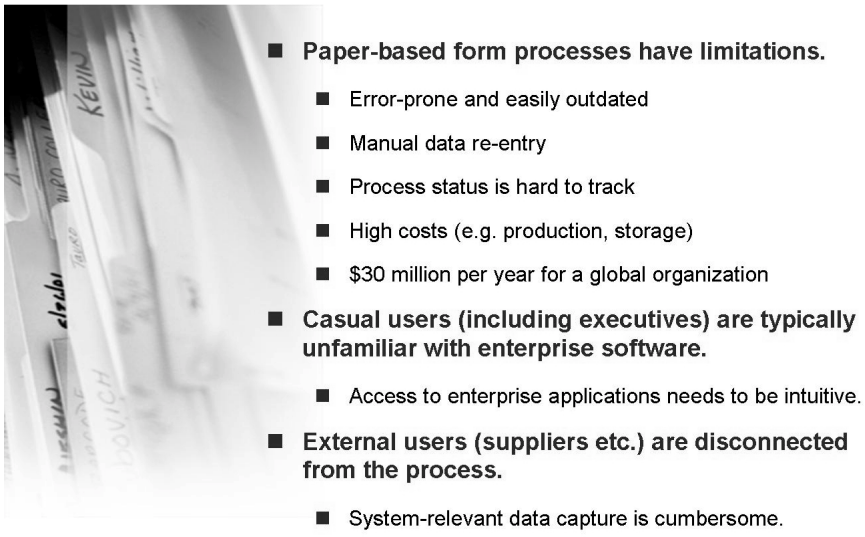
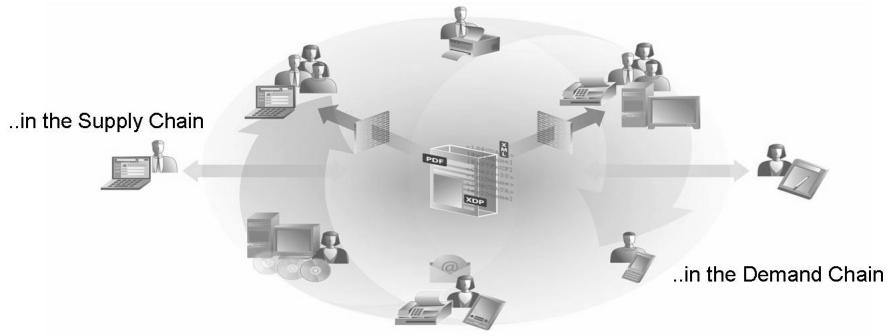


Figure 3: Challenges in forms-based processes

Figure 4: SAP and Adobe Partnership Overview

- **700 Million users with the Adobe Reader**



- Interactive Forms targets the estimated 220 Million employees within SAP's installed base that have Adobe Reader but no SAP access today
- Interactive Forms enables employees quickly without training

Figure 5: Drive data to new SAP users at the edge of the enterprise

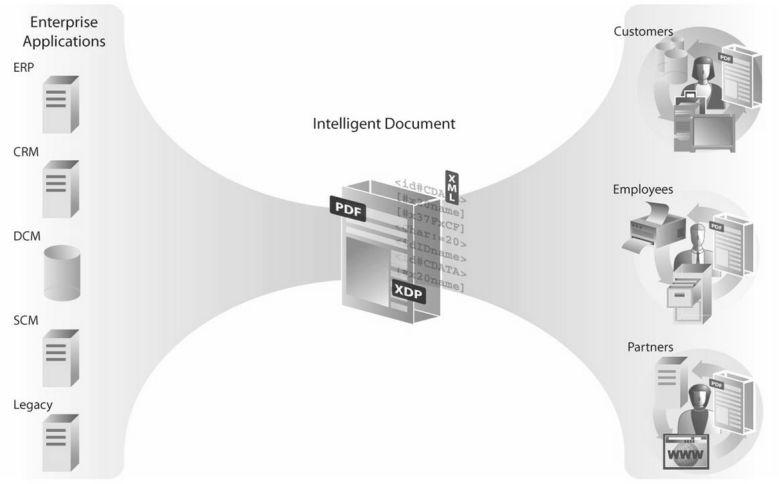


Figure 6: Best run businesses employ end-to-end processes

Lesson Summary

You should now be able to:

- Understand reasons for partnership between SAP and Adobe
- Understand the added value of SAP Interactive Forms by Adobe

Related Information

- SAP on Adobe: http://www.sap.com/solutions/solutionextensions/interactiveforms/index.epx
- Adobe on SAP: http://www.adobe.com/enterprise/partners/sap.html

Lesson: Product Advantages

Lesson Overview

This lesson compares SAP Interactive Forms by Adobe with other UI technologies and underlines the product advantages.

Lesson Objectives

After completing this lesson, you will be able to:

- Compare SAP Interactive Forms by Adobe to other UI technologies
- Compare SAP Interactive Forms by Adobe to other SAP printing technologies
- Provide an overview of SAP Interactive Forms by Adobe integration in SAP NetWeaver

Business Example

You're considering the usage of various UI technologies, for input screens as well as for print purposes. This lesson will help you to compare pros and cons of these technologies with SAP Interactive Forms by Adobe

SAP Interactive Forms by Adobe advantages

- Online applications, web based
- Offline applications
- No learning time
- Same look and feel across applications
- Security
- Design tool
- Industry standards: XML and PDF

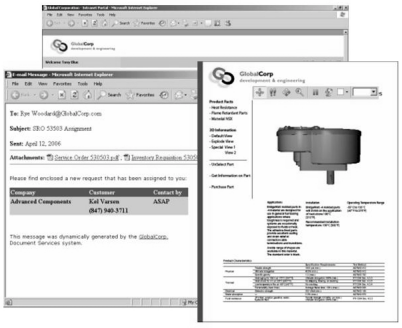


Figure 7: Web, email and advanced forms

- **Web Dynpro / BSP screens : HTML**
  - How do I print my form ?
  - How do I save my form ?
- **Traditional HTML form**
  - HTML frames: how do I print and save my data ?
- **SAPGUI**
  - Heavy client
  - Requires learning time
- **Offline: Excel or Word documents**
  - How do I implement data error control ?

Figure 8: SAP Interactive Forms by Adobe vs. other UI technologies

- *"The SAP forms strategy is based on SAP Interactive Forms by Adobe (IFbA). This applies to both print and interactive forms. In this context, most Business Suite solutions have already been converted to the IFbA technology (more than 2300 forms), and new standard SAP forms will be created using IFbA instead of Smart Forms or SAPscript.*
  SAPscript and Smart Forms shall continue to be supported inline with SAP's maintenance strategy, thereby ensuring the ability to protect existing customer investment."
- Technologies which support PDF natively as principal UI technology (example: HR scenarios and processes with ISR)

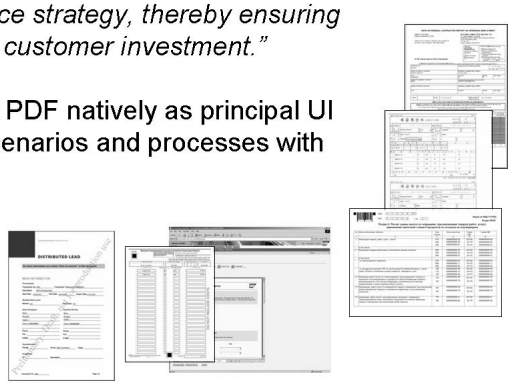


Figure 9: Sample forms delivered by SAP

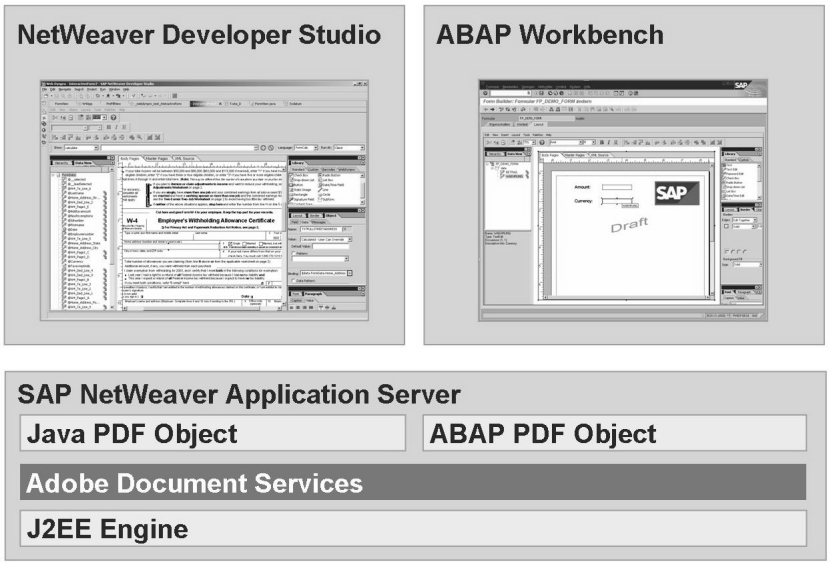


Figure 10: Interactive Forms: Design time and Runtime

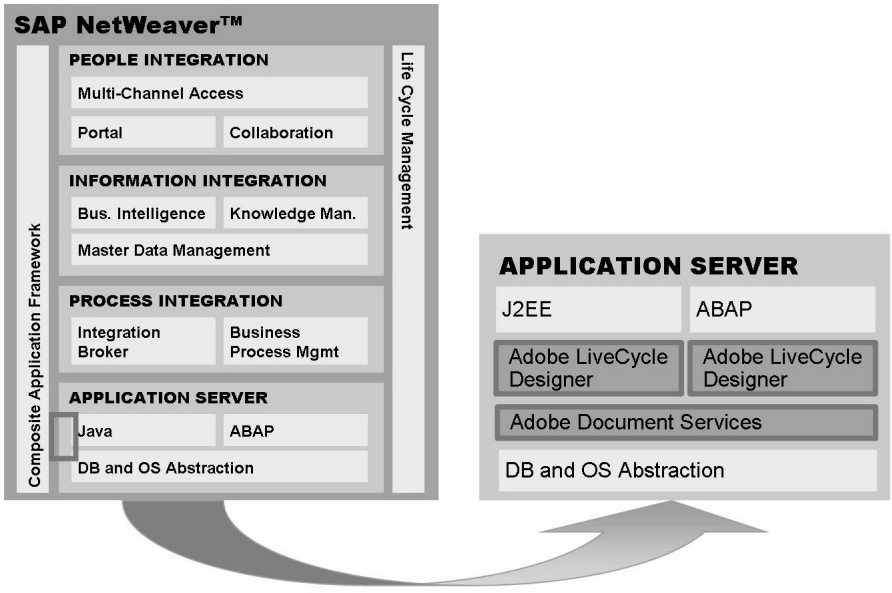


Figure 11: Full integration in SAP NetWeaver

Internal Use SAP Partner Only

- **Design Time: Development**
  - Layout is always problematic...
  - ALD easier to learn
  - ALD: custom objects
  - ...
- **Runtime: Performance**
  - Comparable performance
  - PDF Forms bundling

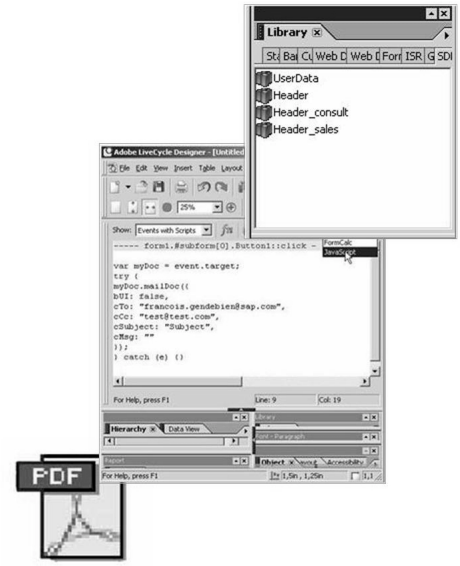


Figure 12: SmartForms vs. PDF-based print forms (1)

- **Front-end and clients**
  - Reader is required for PDF, SAPGUI only for SmartForms
- **Compatibility**
  - Legal templates
  - Accessibility
  - Email sending
  - ...

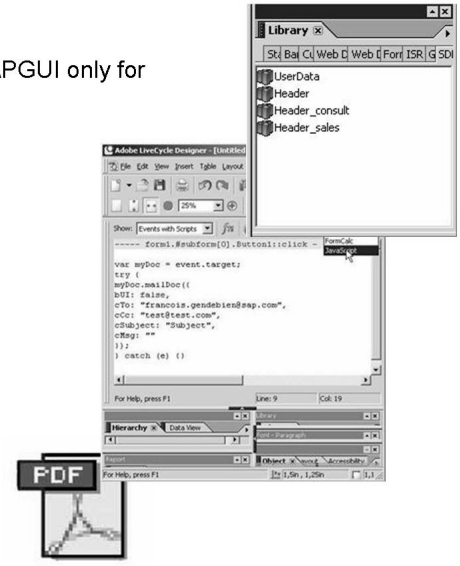


Figure 13: SmartForms vs. PDF-based print forms (2)

- Highly configurable migration wizard: SMARTFORM > Utilities > Migration > PDF-Based Form > Export

- Interface and form can be automatically created with text elements, tables and loops, layout, conditions etc.

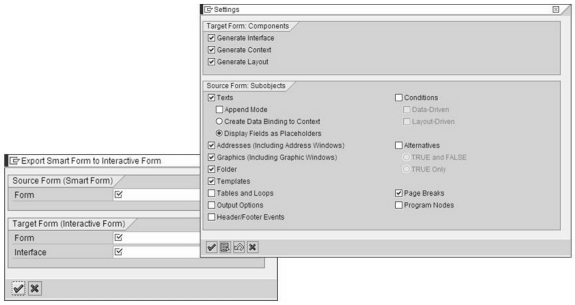


Figure 14: Migration wizard

Lesson Summary

You should now be able to:

- Compare SAP Interactive Forms by Adobe to other UI technologies
- Compare SAP Interactive Forms by Adobe to other SAP printing technologies
- Provide an overview of SAP Interactive Forms by Adobe integration in SAP NetWeaver

Internal Use SAP Partner Only

Lesson: Product Capabilities Overview

Lesson Overview

This lesson aims to provide information on product capabilities: integration with various SAP technologies, usage types, integration in business solutions etc.

Lesson Objectives

After completing this lesson, you will be able to:

- Understand and explain the technical capabilities of SAP Interactive Forms by Adobe
- Understand and explain the technical integration of SAP Interactive Forms by Adobe in SAP technologies
- Understand the 3-year roadmap for SAP Interactive Forms by Adobe

Business Example

You're at the beginning of a project and you're responsible for the selection of the solution, which features various UI elements and several forms. You're asked to make a study and check whether SAP Interactive Forms by Adobe are a relevant altenative.

SAP Interactive Forms by Adobe in SAP technical environments

The focus of this lesson in the technical integration of SIFbA.

Internal Use SAP Partner Only

- **Online Interactive Form Scenario**
  - SAP system access needed
  - Full integration into Web Dynpro for Java and ABAP frameworks
  - Context-sensitive value help, online checks
- **Offline Interactive Form Scenario**
  - No SAP system access needed
  - Static value help, static checking, simple arithmetic calculations
  - Self-contained PDF (XML data+layout)
- **Form Printing Scenario**
  - 'Classic' form processing, not interactive
  - Output documents are printed, e-mailed, archived or faxed

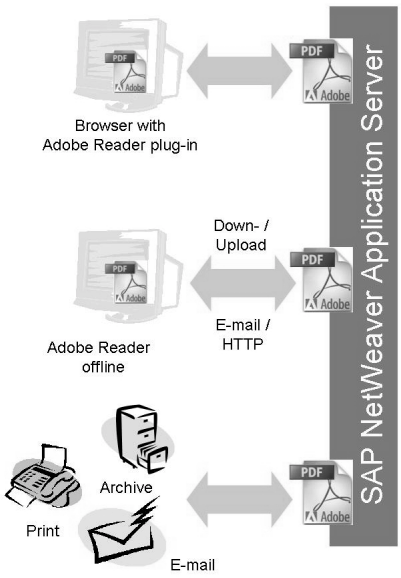


Figure 15: Forms scenarios: Technical View

- **Determine your usage scenario for Interactive Forms:**
  - Interactive PDF use (online or offline – Java and/or ABAP)
  - Interactive use in Manager Self-Services (Java+ABAP)
  - Backend printing only (ABAP)

Keep in mind that Interactive Forms is integrated in six (6!) different ways into the SAP world.

Environment	Availability
Guided Procedures (for offline forms-based workflow processes)	SAP NetWeaver 2004s
Internet Service Request (for online forms-based workflow processes)	mySAP ERP 2004
Web Dynpro for Java (for application development with standard UI technology)	SAP NetWeaver 2004
Web Dynpro for ABAP (for interactive application development in ABAP)	SAP NetWeaver 2004s
ABAP Workbench (for high-volume printing, transaction SFP)	SAP NetWeaver 2004
Enterprise reporting (for printing dynamic BI tables)	SAP NetWeaver 2004s

Figure 16: Integration into SAP environments

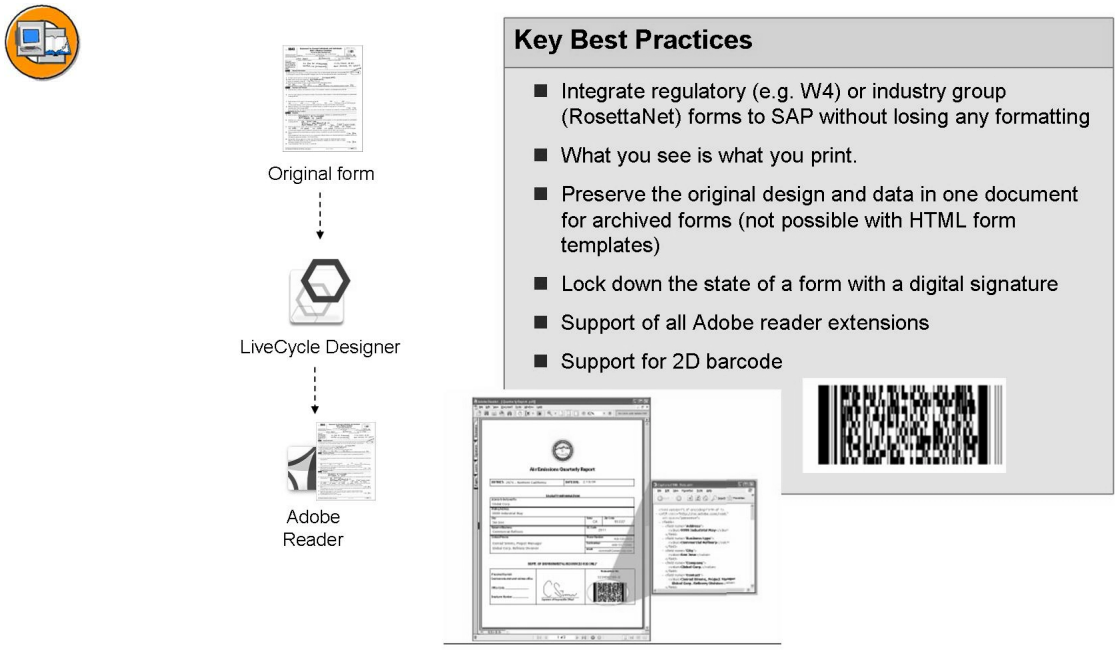


Figure 17: Maintain the visual integrity of a document

- Adobe is an active supporter of SAP's Enterprise Services Architecture
- Interactive Forms is designed to be fully compliant with the Enterprise Service Architecture
- Interactive PDF forms can act as a client to access SAP enterprise services
- Adobe document services in SAP Web AS are accessed through SAP's Web Services technology

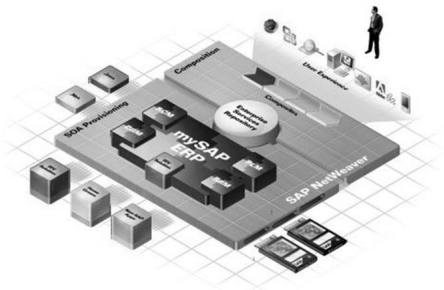


Figure 18: Interactive Forms and Enterprise Services Architecture

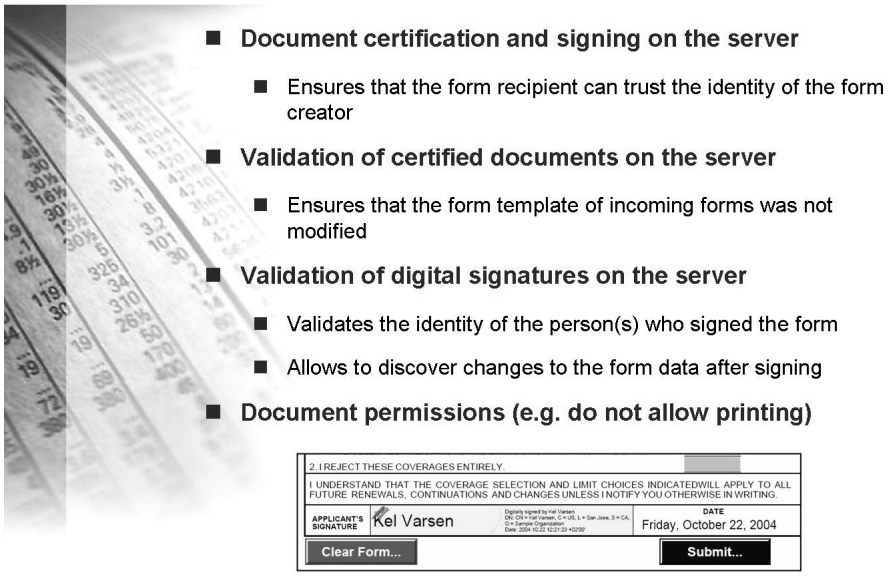


Figure 19: New in SAP NetWeaver 7.0: security

Installation & Configuration

- The administration and configuration of the ADS will be made easier through automatically computed default values (for example Pool Max) and input help and checks on the properties.
- Administration of digital signatures and key storage will be improved.

User Productivity

- The design of forms will be made easier through usability improvements in the LiveCycle Designer, the use of form fragments and with corporate style sheets.
- The development and deployment of offline form scenarios will beaccelerated.

Accessibility

- Enable impared users access for use of interactive and print forms

Performance

- The performance of IFbA for printing will be further improved. For very high-volume scenarios, it is intended to enable several processors in parallel to handle a single print job.

Figure 20: Roadmap details (1)

- **Globalization**
  - Solutions for local product requirements such as right-to-left languages need to be improved.
  - Support for printers will be widened beyond the current PCL, PS,& ZPL compatible printers.
- **Extended Print Features**
  - Improvements will be made to the generated page layout, such as formatting of numbered lists.
  - It will be possible to 'Watermark'PDF print forms.

Figure 21: Roadmap Details (2)

- **Interactive Forms based on Adobe software provide substantial benefits, because they …**
  - Automate and streamline forms-based business processes
  - Are fully integrated into SAP NetWeaver
  - Are available to any SAP solution
  - Combine the benefits of the open technology standards XML and PDF
  - Can be deployed in online and offline scenarios
  - Cover high-volume print requirements

Figure 22: SAP Interactive Forms by Adobe: Summary

Lesson Summary

You should now be able to:

- Understand and explain the technical capabilities of SAP Interactive Forms by Adobe
- Understand and explain the technical integration of SAP Interactive Forms by Adobe in SAP technologies
- Understand the 3-year roadmap for SAP Interactive Forms by Adobe

Related Information

- https://www.sdn.sap.com/irj/sdn/adobe

Unit Summary

You should now be able to:

- Understand reasons for partnership between SAP and Adobe
- Understand the added value of SAP Interactive Forms by Adobe
- Compare SAP Interactive Forms by Adobe to other UI technologies
- Compare SAP Interactive Forms by Adobe to other SAP printing technologies
- Provide an overview of SAP Interactive Forms by Adobe integration in SAP NetWeaver
- Understand and explain the technical capabilities of SAP Interactive Forms by Adobe
- Understand and explain the technical integration of SAP Interactive Forms by Adobe in SAP technologies
- Understand the 3-year roadmap for SAP Interactive Forms by Adobe

Related Information

- SDN Adobe homepage: https://www.sdn.sap.com/irj/sdn/adobe
- SAP Solutions on SIFbA: http://www.sap.com/solutions/solutionextensions/interactiveforms/index.epx

Test Your Knowledge

1. Can you explain the benefits of SAP and Adobe collaboration ?

2. Can you give 3 advantages on the SIFbA over HTML screen ? Over the SAPGUI ?

3. Can you explain the technical integration of SIFbA in SAP NetWeaver at Design time and at Runtime ?

4. Can you cite and explain 3 improvements which are on the SIFbA roadmap ?

5. Can you cite 3 different SAP environments in which SIFbA are integrated ?

Answers

1. Can you explain the benefits of SAP and Adobe collaboration ?

   Answer: Refer to information above.

2. Can you give 3 advantages on the SIFbA over HTML screen ? Over the SAPGUI ?

   Answer: Possibility to save, print and use the form offline; no learning time and no heavy client.

3. Can you explain the technical integration of SIFbA in SAP NetWeaver at Design time and at Runtime ?

   Answer: Refer to slides above.

4. Can you cite and explain 3 improvements which are on the SIFbA roadmap ?

   Answer: Improvements on the roadmap are the following:

   - Installation and configuration
   - User productivity
   - Accessibility
   - Performance
   - Globalization
   - Extended print features

5. Can you cite 3 different SAP environments in which SIFbA are integrated ?

   Answer: There are 6 SAP environments in which SIFbA are integrated: Guided Procedures, Internet Service Request, WD Java, WD ABAP, ABAP workbench, enterprise reporting.

Internal Use SAP Partner Only

Internal Use SAP Partner Only

Unit 2

SAP Interactive Forms by Adobe in SAP environments

Unit Overview

This unit is more technical and focuses on architecture, integration in ABAP programming and provides information and guidelines on the usage of the Adobe LiveCycle Designer.

Unit Objectives

After completing this unit, you will be able to:

- Understand the architecture used at design time and runtime of SAP Interactive Forms by Adobe
- Know and use the best practices and SAP recommendations in terms of architecture
- Troubleshoot an existing installation and find test programs in an SAP ECC system
- Understand and know where to locate the most important SAP Interactive Forms by Adobe in SAP ECC
- Understand the design and runtime required to generate an interactive form from a traditional ABAP report.
- Write your own report to generate an Interactive Form.
- Understand and make use of the main functionalities and elements of the Adobe LiveCycle Designer

Unit Contents

Lesson: SAP Interactive Forms in SAP Environments: Architecture

Lesson Overview

This lesson provides information on architecture at design time and at runtime of the SAP Interactive Forms by Adobe in the ABAP environments.

Lesson Objectives

After completing this lesson, you will be able to:

- Understand the architecture used at design time and runtime of SAP Interactive Forms by Adobe
- Know and use the best practices and SAP recommendations in terms of architecture
- Troubleshoot an existing installation and find test programs in an SAP ECC system

Business Example

You're part a of a company with a complex SAP landscape, featuring various SAP solutions and NetWeaver components. You're searching more information on installation, configuration, architecture possibilities and troubleshooting guidelines. This lesson focuses on these topics.

SAP Interactive Forms by Adobe Architecture

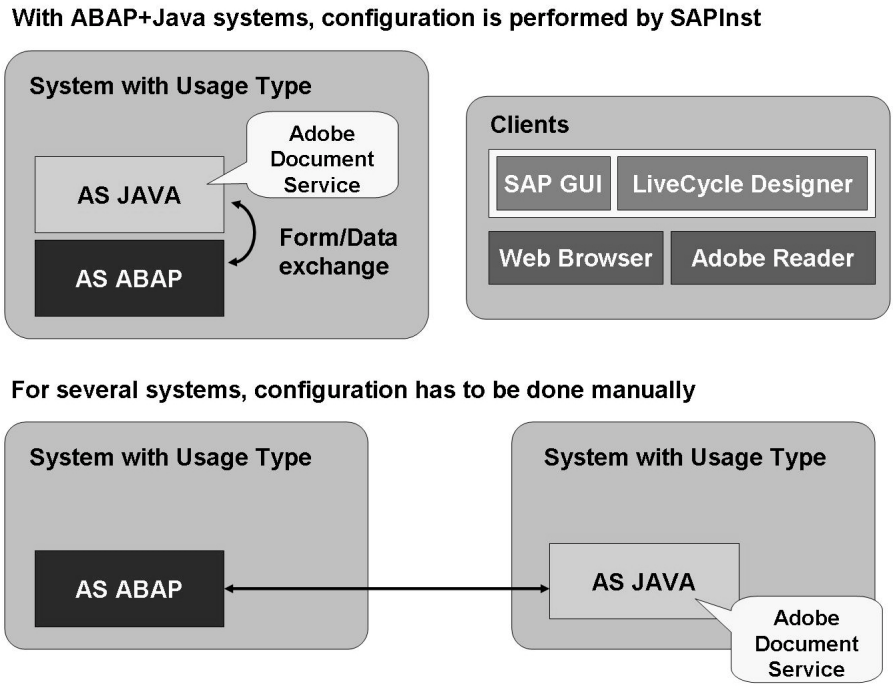


Figure 23: System landscapes for PDF forms in WebAS ABAP

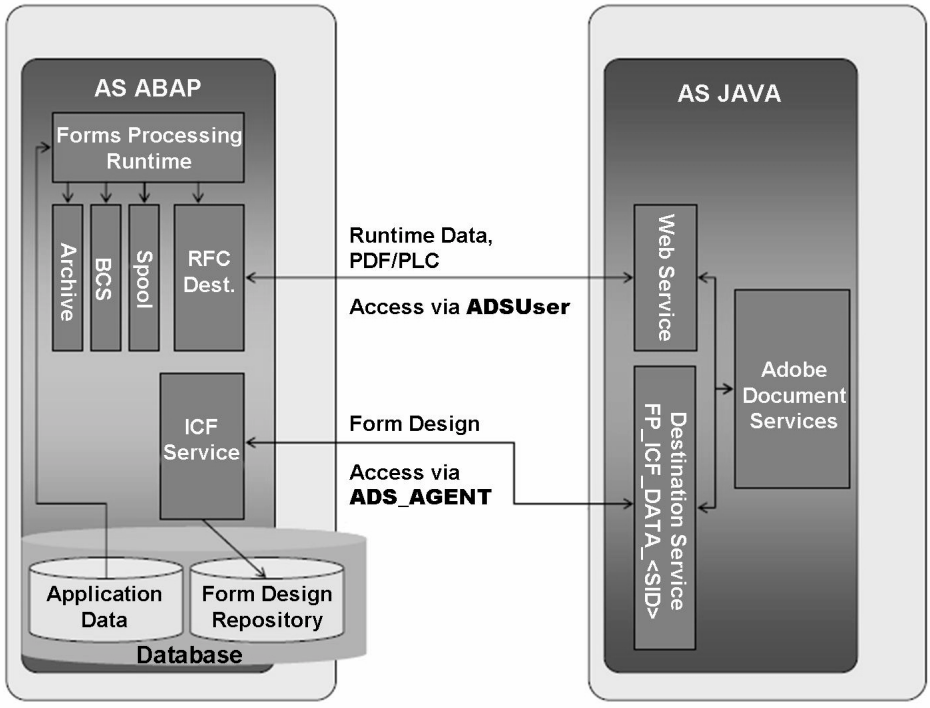


Figure 24: Communication in WebAS ABAP: Adobe Document Services

- Everything runs on the same stack: no specific config to make to run Web Dynpro Java applications
- (Integration is similar close from the one used in WD ABAP)

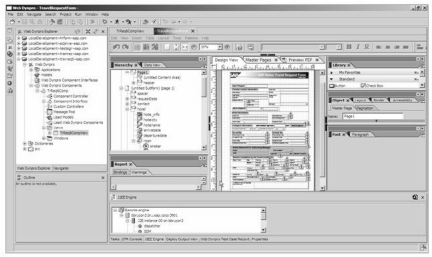


Figure 25: Web Dynpro Java

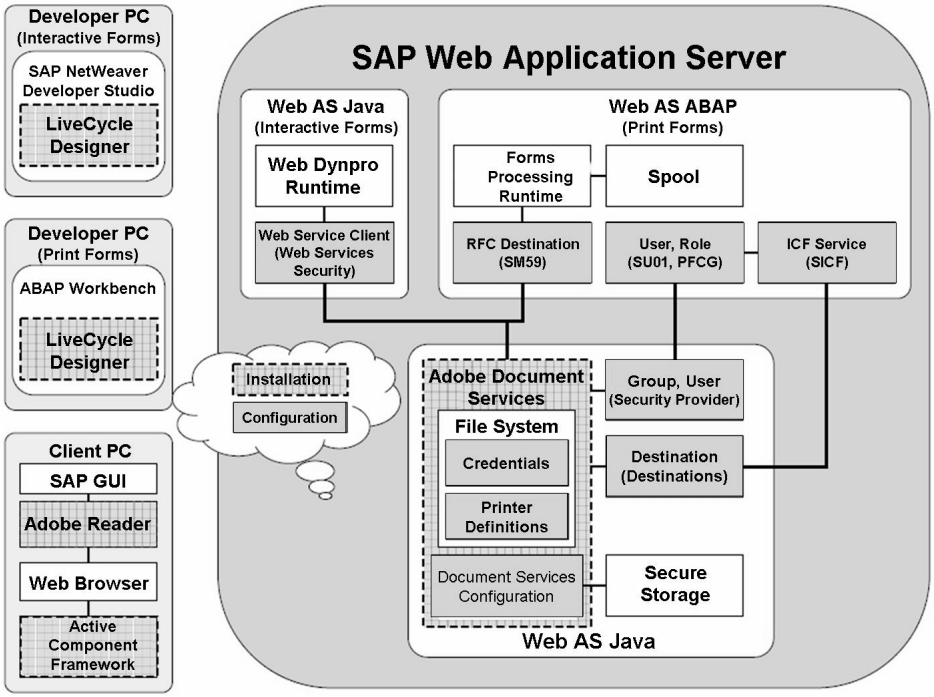


Figure 26: SAP Interactive Forms by Adobe Detailed Architecture

- Support Package Stack level
  - Check on J2EE System Info Page
- ADS web service
  - Check on J2EE web services navigator page
- ADSAgent and ADSUser
  - Check on ABAP SU01 and J2EE UME

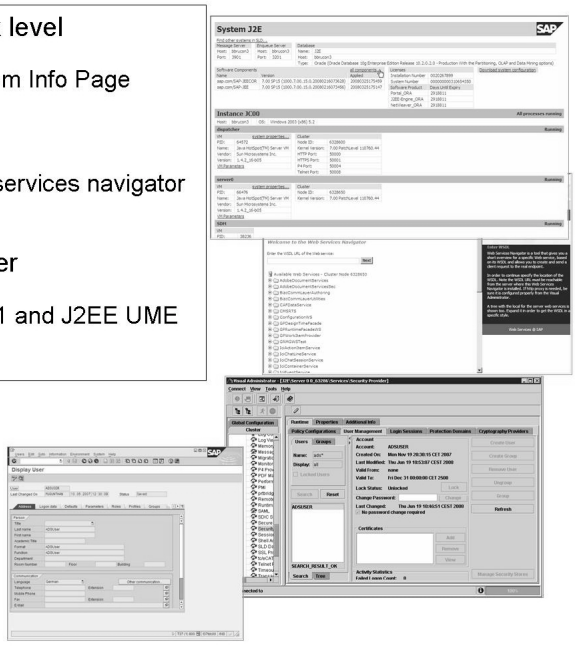


Figure 27: Installation Troubleshooting (1)

- SM59
  - Default created connection: ADS
- J2EE Visual Administrator
  - Web Services security
  - Destinations > HTTP
  - Document Services Configuration: Check credential validity

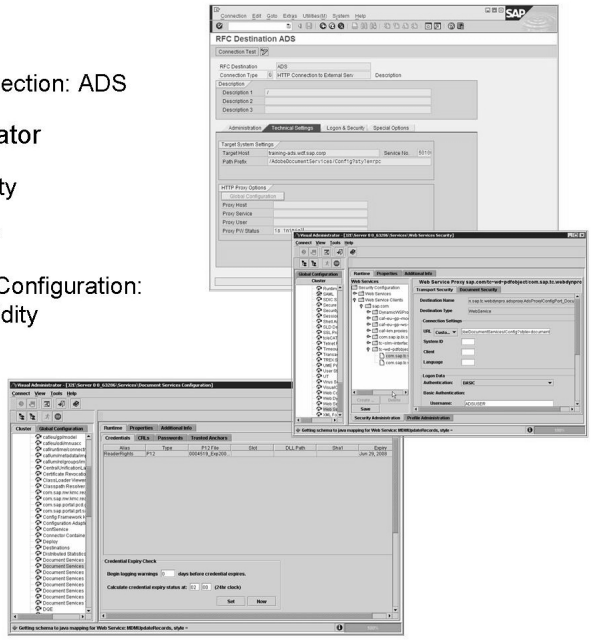


Figure 28: Installation Troubleshooting (2)

- **Client installations**
  - Recommended Adobe Reader versions: 8.1 for WDJava; 7.0.9 for ABAP
  - xACF if required (check SAP Note 766191)

- **Important SAP Notes**
  - 766191: ACF central note
  - 1150277: Prerequisites for ACF
  - 834573: Adobe Acrobat & Adobe Reader version
  - 1043531: Designer and Reader support on Windows Vista
  - 750784: SAP Interactive Forms Licenses

Figure 29: Installation Troubleshooting (3)

SE38 programs:

- FP_TEST*: central test programs
  - FP_TEST_00: Central Test program
- FP_PDF_TEST*: PDF utils programs
  - FP_PDF_TEST_03: Data Extraction

Packages:

- SAFP (SAP Form Processing)
- SFPT (Forms Processing Test Objects)

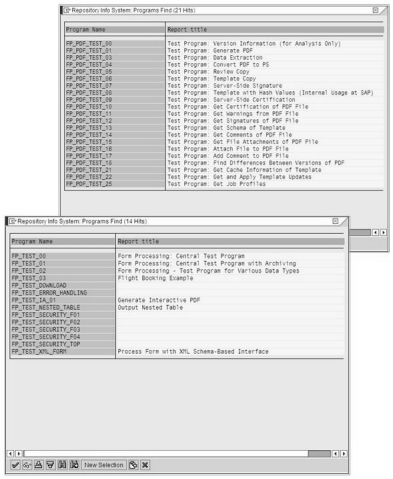


Figure 30: Installation Troubleshooting (4)

SDN SAP Interactive Forms by Adobe homepage:
https://www.sdn.sap.com/irj/sdn/adobe

- Installation & Configuration guides
- Tutorials
  - Adobe LiveCycle Designer
  - ABAP, Web Dynpro ABAP
  - Web Dynpro Java
  - Guided Procedures
  - BSP

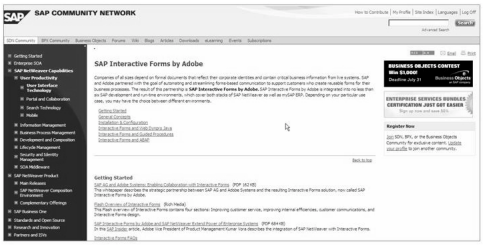


Figure 31: Installation and configuration guides: SDN

Internal Use SAP Partner Only

Lesson Summary

You should now be able to:

- Understand the architecture used at design time and runtime of SAP Interactive Forms by Adobe
- Know and use the best practices and SAP recommendations in terms of architecture
- Troubleshoot an existing installation and find test programs in an SAP ECC system

Related Information

- SDN Adobe homepage: https://www.sdn.sap.com/irj/sdn/adobe

Internal Use SAP Partner Only

Lesson: Business Applications Featuring SAP Interactive Forms by Adobe

Lesson Overview

This lesson goes through several examples of existing standard SAP Interactive Forms by Adobe integrated in various components of the ECC.

Lesson Objectives

After completing this lesson, you will be able to:

- Understand and know where to locate the most important SAP Interactive Forms by Adobe in SAP ECC

Business Example

You're part of an SAP team considering the usage of SAP Interactive Forms by Adobe. Your project will involve various SAP components and solutions so you're searching information on modules which make use of nteractive forms as UI tehcnology..

General Overview

- Guided Procedures is the process layer of SAP's Composite Application Framework (CAF) in SAP NetWeaver
  - Offers flexible, highly functional workflow environment
  - Enables users to easily set up and execute collaborative business processes
  - Provides reusable templates for actions
  - Leverages existing systems
  - Uses Web Dynpro user interfaces

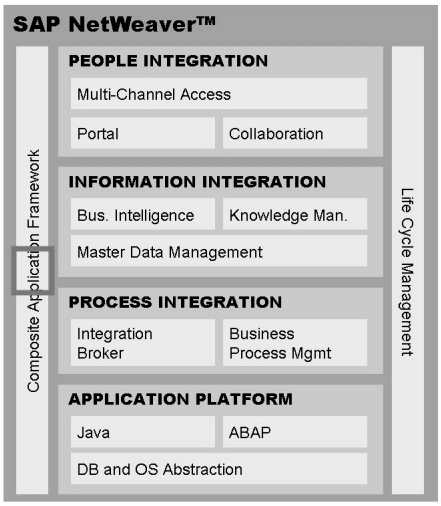
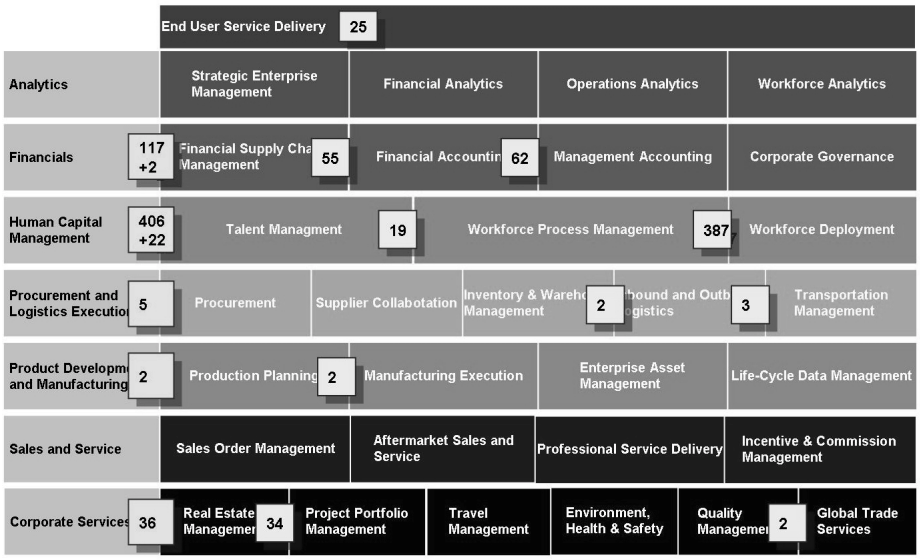


Figure 32: Guided Procedures



Note: some forms cannot be assigned a specific process ('+x' in yellow boxes)

Figure 33: Interactive forms in business solutions

HCM Processes and Forms

HR Scenarios (Personnel Change Requests)

- Change Employee Group and Subgroup
- Change Personnel Area and Sub Area
- Request for Promotion
- Request for Special Payment
- Change of Working Time
- Request for Internal Transfer
- Request for Internal Transfer (Enhanced)
- Request for Separation
- Request for Separation (Enhanced)
- Request for Transfer
- Request for Transfer (Enhanced)
- Simple Requisition Request
- Extended Requisition Request

Figure 34: Manager Self Services (1)

Financials Scenarios

- Request Creation of Cost Center
- Request Change to Cost Center
- Request Creation of Internal Order
- Request Change to Internal Order
- Request Creation of Profit Center
- Request Profit Center Change
- Request Equipment Repair
- Request Equipment Change
- Notify Equipment Loss
- Request Adjustment Posting
- Request Budget Change
- Express Planning

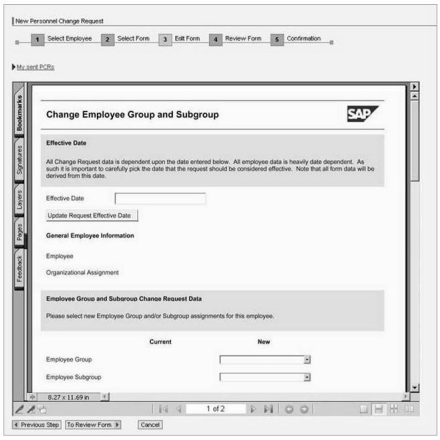


Figure 35: Manager Self Services (2)

Scenarios:

- Hiring
  - Germany, US
  - 2 scenarios each
- Transfer
- Maternity Leave
- Birth of a Child
  - Germany, US
- Termination
  - Germany, US
  - 4 scenarios each
- Organizational Change

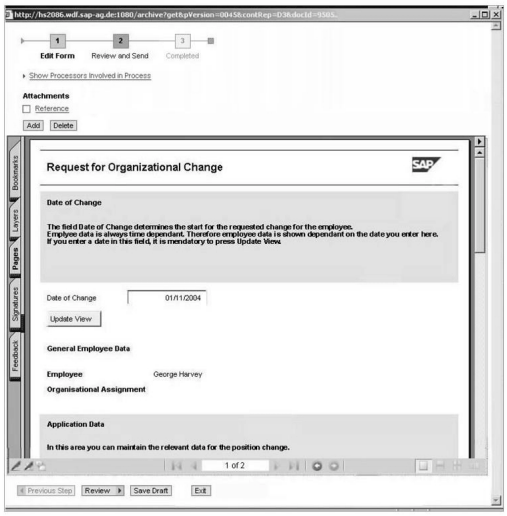


Figure 36: HCM Processes and Forms

Financials Scenarios

- Request Budget Change
- Request Budget Transfer
- Request Adjustment Posting
- Explanation of a Variance
- Message Regarding a Variance
- Request New Key Figure
- Express Planning: Strategy and Targets
- Express Planning: Targets and Tasks
- Cost Center Review
- Request Assignment of Equipment

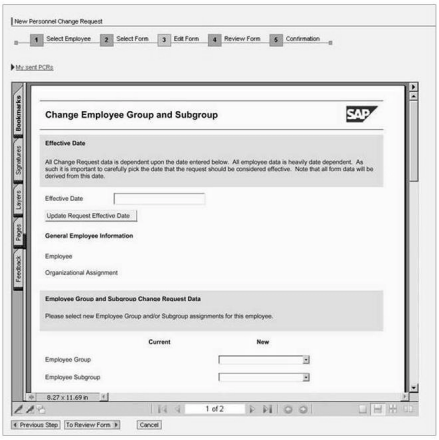


Figure 37: Manager Self Services (3)

Other SAP applications

- **Based on Internet Service Request:**
  - HCM Talent Management
    - Performance Appraisal Management
  - Operations / Product Life-Cycle Management
    - Create PSP Element
    - Change PSP Element
- **Other**
  - Financials: Invoice Management System
    - Invoice Exception Handling (2 scenarios)
- **Based on Guided Procedures**
  - Operations / Product Life-Cycle Management
    - Create Material Master
    - Change Existing Material Master
    - (not generally released as of May 2006)

Figure 38: Miscellaneous

- **SAP CRM (as of SAP CRM 5.0)**
  - Lead Management
    - Offline scenario involving external channel partner
- **SAP for Higher Education & Research**
  - University Student Admission (based on Internet Service Request)

Figure 39: Other SAP Business Applications

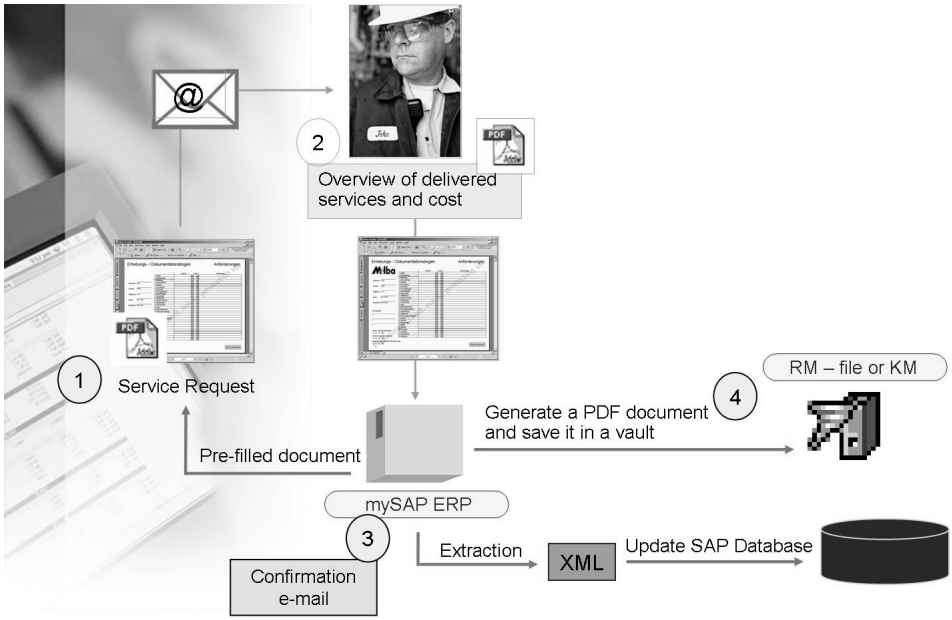


Figure 40: Scenario for Service Request where Transaction = Document

Lesson Summary

You should now be able to:

- Understand and know where to locate the most important SAP Interactive Forms by Adobe in SAP ECC

Related Information

- SDN Adobe homepage: https://www.sdn.sap.com/irj/sdn/adobe

Internal Use SAP Partner Only

Lesson: ABAP Reports and Form Interfaces

Lesson Overview

This lesson focuses on ABAP programming to develop and deploy custom SAP Interactive Forms by Adobe.

Lesson Objectives

After completing this lesson, you will be able to:

- Understand the design and runtime required to generate an interactive form from a traditional ABAP report.
- Write your own report to generate an Interactive Form.

Business Example

Your business case requires custom interactive forms for various usage. You're a part of an ABAP development team and you need to learn how to develop those forms.

Create an Interactive Form with an ABAP report

- To create an Interactive Form, a form interface must be created first, then the form itself, based on this interface.
- The interface represents data structures to be displayed, or which will be filled in by the end-used
- An interactive form makes use of the data elements of the interface.
- The Adobe LiveCycle Designer is used to make the layout of the form.

Figure 41: Create an Interactive Form with an ABAP report

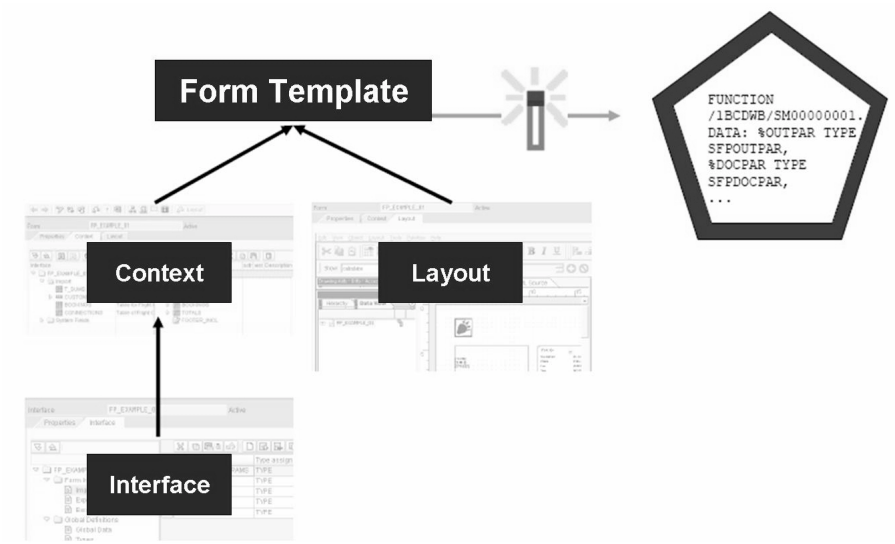


Figure 42: Tools involved at Design Time

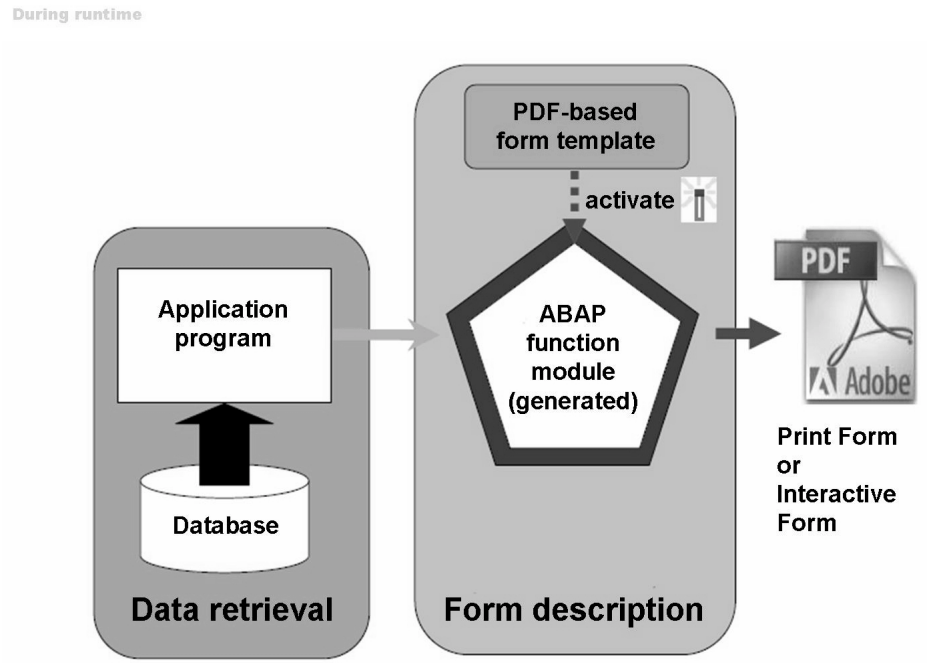


Figure 43: During Runtime

- **Create an interface (transaction SFP)**
- **Add your data elements to the import parameters list**
  - Import parameters are passed from the application program to the form at runtime
  - ... and vice-versa for export parameters (typical export parameter: FORMOUTPUT)
- **Create a form based on the interface**
  - The elements which will be displayed on the form must be present in the context of the form
- **The layout tab of the form opens the Adobe LiveCycle Designer**
  - Interactive forms and print forms are designed the same way

Figure 44: Step by Step

```
* (1) Data retrieval and processing
SELECT ... FROM ...
...

* (2) Find out name of generated function module
CALL FUNCTION 'FP_FUNCTION_MODULE_NAME'...
* (3) Start form processing
CALL FUNCTION 'FP_JOB_OPEN'...

LOOP AT ...
* (4) Call function module dynamically
   CALL FUNCTION <generated function module> ...
ENDLOOP.

* (5) End form processing
CALL FUNCTION 'FP_JOB_CLOSE'...
```

Figure 45: PDF Print Form program: section of the application program

- **Interfaces for print forms: ABAP dictionary based or SmartForms compatible (check BC480 course)**
- **Interfaces for WD ABAP: XML based interface, used when designing the PDF in the view**
  - **Can be created automatically from the Web Dynpro Context**
- **Form Context**
  - **Created from the Interface data**
  - **Can contain conditions**

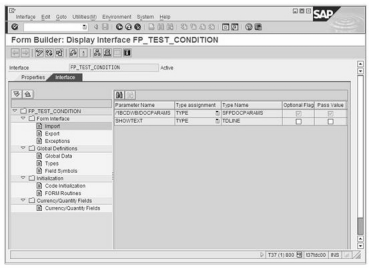


Figure 46: Interfaces and Form Context

Output structures

- Structure **sfpdocparams**
  - Fillable
  - Dynamic
- Structure **sfpoutputparams** (spool control)
  - Dest: spool output device
  - PDFCHANGESRESTRICTED: restricts copy of text elements or images in the PDF
  - ..

Figure 47: Generated Interactive form: parameters in output structures

Internal Use SAP Partner Only

Exercise 1: Discover and understand ABAP reports used to produce an Interactive Form.

Exercise Objectives

After completing this exercise, you will be able to:

- Understand the ABAP logic to produce PDF forms
- Locate and understand test programs

Business Example

You'd like to find examples and sample programs in SAP systems which you could use to learn how to generate Interactive Forms.

Task 1:

In this task, you'll find out which are the standard reports available in the SAP ECC.

1. Search for reports FP_PDF*
2. Search for reports FP_TEST*.

Task 2:

In this task you're asked to run and explore the coding required to generate an Interactive Form.

1. Run and analyze program FP_TEST_00.
2. Run and analyze program FP_TEST_03. Can you locate main parts of the program ? (collection of data, opening of the print program, findout and call of generated function module, close of the print program)

Solution 1: Discover and understand ABAP reports used to produce an Interactive Form.

Task 1:

In this task, you'll find out which are the standard reports available in the SAP ECC.

1. Search for reports FP_PDF*
   a) After login to your system, go to transaction SE38.
   b) Enter FP_PDF* in the search field and press F4. The list of available test programs are displayed
2. Search for reports FP_TEST*.
   a) In SE38, enter FP_TEST* in the search field and press F4. The list of available test programs are displayed.

Task 2:

In this task you're asked to run and explore the coding required to generate an Interactive Form.

1. Run and analyze program FP_TEST_00.
   a) In SE38, enter FP_TEST_00 and press F8.
   b) When prompting for an output device, enter LP01. Click on Print Preview. The generated PDF should be displayed.
   c) In the SE38 first screen, select Source Code > Display.
2. Run and analyze program FP_TEST_03. Can you locate main parts of the program ? (collection of data, opening of the print program, findout and call of generated function module, close of the print program)
   a) In SE38, enter FP_TEST_03 and press F8.
   b) When prompting for an output device, enter LP01. Click on Print Preview.
   c) In the SE38 first screen, select Source Code > Display. Here you can identify the building blocks for the print program.

Result

You are now able to locate and understand sample programs in SAP systems.

Lesson Summary

You should now be able to:

- Understand the design and runtime required to generate an interactive form from a traditional ABAP report.
- Write your own report to generate an Interactive Form.

Related Information

- SDN Adobe homepage: https://www.sdn.sap.com/irj/sdn/adobe

Lesson: Adobe LiveCycle Designer basics

Lesson Overview

This lesson focuses on the main features of the tool used at design time: the Adobe LiveCycle Designer. The most commonly used elements will be explored.

Lesson Objectives

After completing this lesson, you will be able to:

- Understand and make use of the main functionalities and elements of the Adobe LiveCycle Designer

Business Example

Your team is asked to produce more and more SAP Interactive Forms by Adobe and also to train people for quick and easy changes.

The Adobe LiveCycle Designer

- Designer is a tool used to create form designs from which you generate print, interactive, static, or dynamic forms.
- Form designs define the appearance and behavior of a form once it is processed and rendered by a client or server application.
- The term render describes the process of using a form design to generate the form in a particular file format and, possibly, present the form to the user.

Figure 48: Working with the LiveCycle Designer

- To create form designs, you drag objects, such as graphic elements, interactive fields, and images onto the layout editor.
- Palettes are areas within the Designer environment that contain information and options that enable you to configure object characteristics.

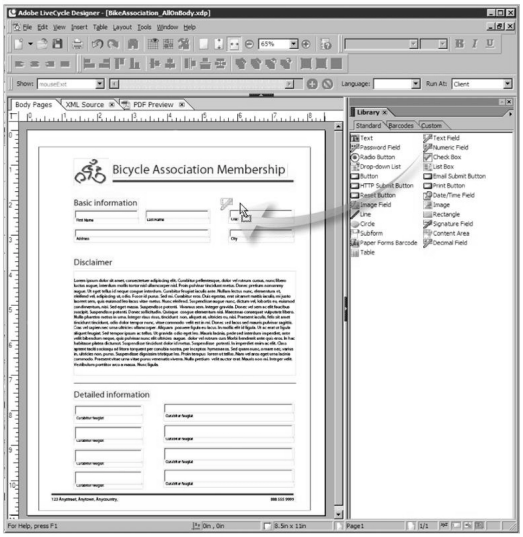


Figure 49: Designer Objects

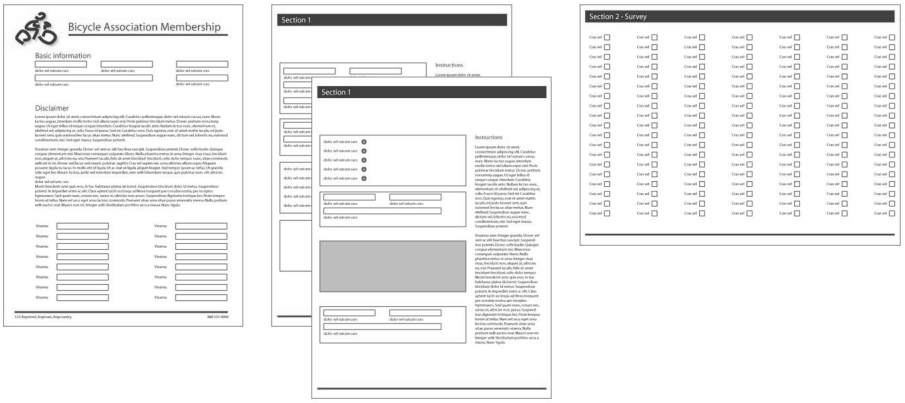


Figure 50: Master and Body Pages

- **Master pages**
  - Master pages define the working area of a page or set of pages and contain the common objects that must appear on those pages.
- **Body Pages**
  - Body pages contain elements that are unique to a particular page. You associate each body page with a master page.
- Using master pages can ensure:
  - Consistency across corporate forms.
  - Consistency within multi-page forms.
  - Efficiency and reduced design effort through reuse.

Figure 51: Master and Body Pages

- The Layout Editor, includes the following views:
  - Body Pages view
  - Master Pages view
  - PDF Preview

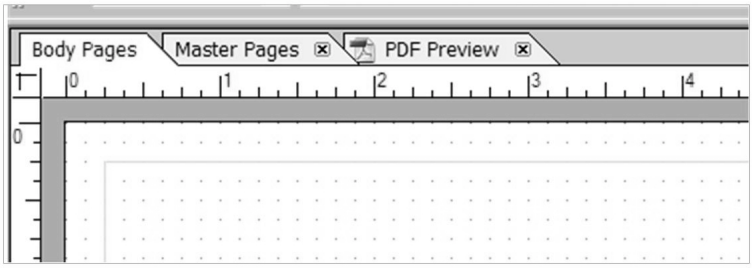


Figure 52: Form Design View

- The Hierarchy palette provides a tree view of the form objects and defines the relationship between objects.
- Objects selected in one view are reflected in the other views (except PDF Preview)

Figure 53: Form Design View

- Designer objects can be grouped into the following categories:
  - Draw objects
  - Field objects
  - Complex container objects
- You use the Object palette to configure each object uniquely.
- The Object palette displays different options for most objects, although many options are common within an object type.

Figure 54: Configuring and Organizing Designer Objects

- Draw objects enable you to add graphic elements and static text to a form design.
- You access the following Draw objects from the Standard tab of the Library palette:
  - Line
  - Circle
  - Rectangle
  - Text
  - Image

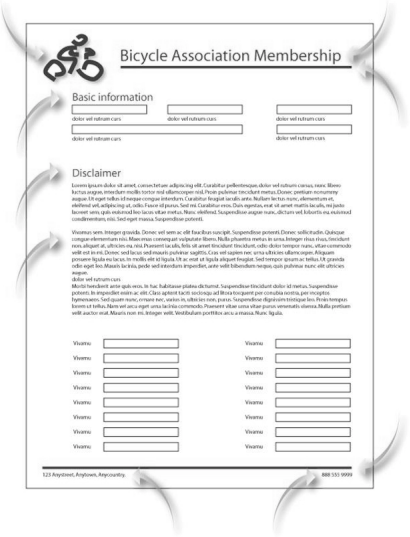


Figure 55: Draw Objects

- Field objects provide the means by which a user interacts with a form.
- The following field objects are available from the Standard tab of the Library palette.
  - Text Field
  - Check Box
  - Numeric Field
  - Signature Field
  - Drop-down List
  - Listbox
  - Paper Forms Barcode
  - Date/Time Field
  - Image Field
  - Radio Button
- Designer creates a composite object that can include the field object and a default caption.

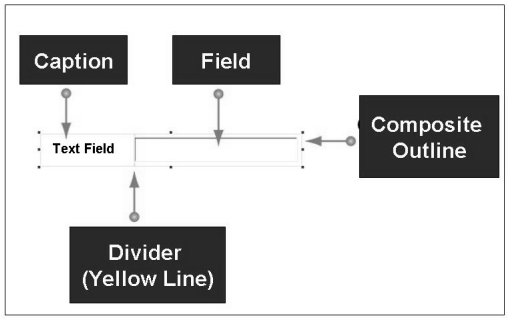


Figure 56: Field Objects

- Complex containers are objects that contain other objects, such as draw, field, or other container objects.
- Complex containers serve two key purposes. They:
  - Provide a means for organizing objects visually in the Hierarchy palette and in the layout editor.
  - Control the flow and occurrence of other objects in dynamic forms.
- The following complex container objects are available:
  - Master Pages and Master Page Objects
  - Subform Objects
  - Table Objects
  - Exclusion Group Objects
- Each of these complex container objects are visible in the Hierarchy palette after you create them.

Figure 57: Complex Container Objects

- Using Designer, you can create forms that faithfully replicate a form layout both in terms of object positioning and formatting.
- The following palettes provide the configuration options to achieve this fidelity.
  - Layout
  - Accessibility
  - Border
  - Font
  - Paragraph
- The Layout, Border, and Accessibility palettes behave similarly for all objects.
- The Font and Paragraph palettes are available for all objects that contain text or captions.

Figure 58: Object Positioning and Formatting Properties

- Designer enables you to add your own custom objects to the Library palette.
- Custom Objects enable you to assign properties, and add script, to a form object and save the configured object for reuse.
- You can save groups of objects as a custom object.
- You can create custom tabs in the Library palette for your own custom objects or for objects from a shared library.

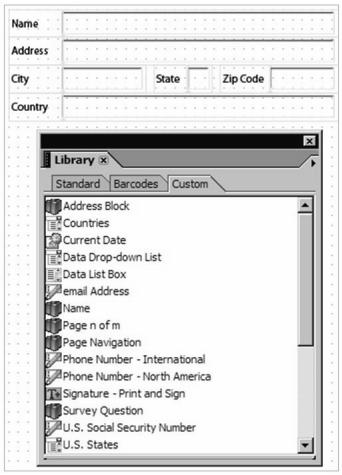


Figure 59: Custom Objects

- You can import forms and documents, that were created with third-party applications, in to Designer.
- Designer enables you to import third-party files of the following types:
  - Adobe Form Designer 5.0 Files (*.xft)
  - Adobe Output Designer Form (*ifd)
  - Microsoft Infopath (*.xsn)
  - Microsoft Word Document (*.doc)
- The purpose of this feature is to use these documents as a starting point for your form designs.
- You can import forms created in Acrobat, which are not XFA-based, in the same way as you import third-part forms.

Figure 60: Import and Open Documents and Forms

Exercise 2: Generate a custom interactive form

Exercise Objectives

After completing this exercise, you will be able to:

- Understand how to generate an SAP Interactive Form by Adobe with an ABAP report
- Write an ABAP report to generate an SAP Interactive Form by Adobe

Business Example

You're responsible for the development for an Interactive Form which should be used either in standalone mode, or integrated later on in the project in an online application.

For this, you decide to start the development of the form using an ABAP report.

Data	Date Value
Logo X position	`6.5in`
Logo Y position	`0.375in`
Logo width	`1.375in`
Logo height	`1in`
Firstname X position	`0.5in`
Firstname Y position	`1.125in`
Firstname width	`4in`
Firstname height	`0.25in`
Lastname X position	`0.5in`
Lastname Y position	`1.375in`
Lastname width	`4in`
Lastname height	`0.25in`

Task 1:

First task is to create a development package to centralize all your developments you'll make during this course.

1. Create a package with name ZBC481GROUP##, where ## is your group number.

Continued on next page

Task 2:

In this task, you prepare the data to be used by the Interactive Form.

1. Create a structure called ZPDF_DATA_##, where ## is your group number, with the following fields/type: MANDT/MANDT, FIRSTNAME/CHAR25, LASTNAME/CHAR25
2. Create a table type ZPDF_TDATA_##, where ## is your group number.
3. Create an interface called ZBC481_EX01_IF_##, where ## is your group number. Assign it as an optional import parameter the structure you've created.

Task 3:

In this task, you create a form ready to consume the interface and structures from task 1.

1. Create a form called ZBC481_EX01_FORM_##, where ## is your group number. Select the interface you've created as the interface.
2. Insert all the structure elements to the form context.
3. Create a simple layout for your PDF form with the FIRSTNAME and LASTNAME data. Enter respectively First Name and Last Name as field captions. Set the appearance of the field to Underlined. Check that paper size of your form is A4. Insert an SAP logo on the master page (logo can be found under My Documents > BC481 > saplogo.gif). Positioning requirements for the three elements are given in the data section above.

Task 4:

Finally, you create an ABAP report which will call created Interactive Form.

1. Create a program called ZBC481_GENERATE_PDF_##, where ## is your group number, to generate your interactive PDF form.
2. Generate your form. Save it locally and verify that it acts as a data container.

Solution 2: Generate a custom interactive form

Task 1:

First task is to create a development package to centralize all your developments you'll make during this course.

1. Create a package with name ZBC481GROUP##, where ## is your group number.

   a) Start transaction SE80.

   b) Select “Package” from the drop down list, enter ZBC481GROUP##, where ## is your group number and hit enter. Enter a description such as BC481 Package Group ##, select HOME as software component and hit the Save button.

   c) You're then prompted for the creation of a workbench request. Hit the create request button (or F8) and enter a short description such as BC481 Generic Request Group ##, where ## is your group number. Select this request for your package and hit continue.

   d) Your package is now created and ready for usage.

Continued on next page

Task 2:

In this task, you prepare the data to be used by the Interactive Form.

1. Create a structure called ZPDF_DATA_##, where ## is your group number, with the following fields/type: MANDT/MANDT, FIRSTNAME/CHAR25, LASTNAME/CHAR25

   a) Go to transaction SE11.

   b) In the Data Type field enter your structure name, ZPDF_DATA_## and hit create.

   c) Select Structure and enter a description such as BC481 Ex01 Structure Group ##, where ## is your group number.

   d) Enter the three mentioned fields stated above: MANDT of type MANDT, FIRSTNAME of type CHAR25 and LASTNAME of type CHAR25.

   e) Select an enhancement category: in the display screen of your structure, go to Extras > Enhancement category. Select Can be enhanced and hit continue. (Remark: the enhancement category is used to define wich type of enhancement can be made on the structure itself.)

   f) Save your object in your package, using the workbench request created above and activate it.

2. Create a table type ZPDF_TDATA_##, where ## is your group number.

   a) In the field Data Type, enter ZPDF_TDATA_##, where ## is your group number, and hit create.

   b) Select Table Type from the radio buttons. Enter a short text such as BC481 PDF Table Type Group ## and select ZPDF_DATA_## as a line type.

   c) Save and activate your table type using your package and the workbench request you've created earlier.

Continued on next page

3. Create an interface called ZBC481_EX01_IF_##, where ## is your group number. Assign it as an optional import parameter the structure you've created.

   a) Go to transaction SFP and select the radio button “Interface”.

   b) Enter interface name ZBC481_EX01_IF_##, select ABAP Dictionary-based interface and hit create. Enter a description such as BC481 Ex01 Interface Group ## and save it in your package, using the workbench request you've created earlier.

   c) In the import parameters, add a line by clicking on the white sheet (above the parameters lines) or on the green plus sign. Name your new parameter PDFDATA, which is of TYPE ZPDF_TDATA_##.

   d) Don't forget to check the two checkboxes “Optional” and “Pass Value. ”

Task 3:

In this task, you create a form ready to consume the interface and structures from task 1.

1. Create a form called ZBC481_EX01_FORM_##, where ## is your group number. Select the interface you've created as the interface.

   a) In transaction SFP, select the radio button “Form”.

   b) Enter your form name and click create. Enter a description such as BC481 Ex01 PDF Form Group ## and save it in your package, using the workbench request you're created in the previous task.

   c) Select the interface you've just created (ZBC481_EX01_IF_##) for this form. Save the form in your package and use the workbench request you've previously created. Activate your form.

2. Insert all the structure elements to the form context.

   a) In the change mode for your form, go to the Context tab.

   b) Select the structure PDFDATA from the import parameters of the interface (left part of the screen).

   c) Drag and drop the structure under the root context node (which has normally the same name as your form).

3. Create a simple layout for your PDF form with the FIRSTNAME and LASTNAME data. Enter respectively First Name and Last Name as field captions. Set the appearance of the field to Underlined. Check that paper size

Continued on next page

of your form is A4. Insert an SAP logo on the master page (logo can be found under My Documents > BC481 > saplogo.gif). Positioning requirements for the three elements are given in the data section above.

a) Go the the layout tab of the transaction SFP.

b) To check the paper size, select the Hierarchy palette and select the Page1 element (data > (Master Pages) > Page1). When the element is selected, go to the Object palette > Master Page. You can check the master page's properties, such as paper type. Set it to A4 instead of Letter.

c) In the left part of the screen, in the data view palette, you can find the three fields from the context. Be sure to be on the Design View in the LiveCycle Designer.

d) By drag and drop, place the fields FIRSTNAME and LASTNAME to the design view of your form. Select the field by clicking on it in the Design View, then go to the Object palette. Set the caption and appearance. Then go to the Layout palette and set the layout requirements.

e) Go to the Master Pages tab. By drag and drop, add an Image element from the Standard library. Select the element and in the layout tab set the layout requirements as indicated above. In the object tab of the image, check Embed image data and enter the URL of the image. Hint: the yellow folder icon next to the URL field opens an explorer to get the image. Hit Ok when 'Network Connection Not Found' opens up.

f) Once this is done, save and activate the form.

Task 4:

Finally, you create an ABAP report which will call created Interactive Form.

1. Create a program called ZBC481_GENERATE_PDF_##, where ## is your group number, to generate your interactive PDF form.

   a) Go to transaction SE38, enter ZBC481_GENERATE_PDF_## as name and hit create. Enter a description such as BC481 Generate Interactive Form - Group ##.

   b) Program is an executable program, status is Test Program. Save your object in your package, using the workbench request you've created earlier.

   c) Insert the following code to the program under your report name. Adapt the code with your group number. The source code of the program for group 00 can be found under My Documents > BC481 > ZBC481_GENERATE_PDF_00.txt.

Continued on next page


```
parameter:      p_form   type tdsfname   default 'ZBC481_EX
                p_conn   type rfcdest    default 'ADS',
                p_ia     type tdbool.

data:           fm_name   type rs38l_fnam,
                fp_docparams type sfpdocparams,
                fp_outputparams type sfpoutputparams,
                error_string type string.

* First get name of the generated function module
  call function 'FP_FUNCTION_MODULE_NAME'
    exporting
      i_name                    = p_form
    importing
      e_funcname                = fm_name.
*   exception handling ???

* Set output parameters and open spool job
  fp_outputparams-preview    = 'X'.    " launch print previ
  fp_outputparams-connection = p_conn.
  fp_outputparams-dest = 'LP01'.

  call function 'FP_JOB_OPEN'
    changing
      ie_outputparams       = fp_outputparams
    exceptions
      cancel                = 1
      usage_error           = 2
      system_error          = 3
      internal_error        = 4
      others                = 5.
  if sy-subrc <> 0.
    message id sy-msgid type sy-msgty number sy-msgno
            with sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.
  endif.

* Set form language and country (->form locale)
  fp_docparams-fillable = p_ia.

* Now call the generated function module
  call function fm_name
    exporting
      /1bcdwb/docparams        = fp_docparams
*   IMPORTING
*     /1BCDWB/FORMOUTPUT       =
```

Continued on next page


```
    exceptions
      usage_error              = 1
      system_error             = 2
      internal_error           = 3
      others                   = 4.
  if sy-subrc <> 0.
    call function 'FP_GET_LAST_ADS_ERRSTR'
      importing
        e_adserrstr       = error_string.
    if not error_string is initial.
*     we received a detailed error description
      write:/ error_string.
      exit.
    else.
      message id sy-msgid type sy-msgty number sy-msgno
              with sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.
    endif.
  endif.

* Close spool job
  call function 'FP_JOB_CLOSE'
*   IMPORTING
*     E_RESULT             =
    exceptions
      usage_error          = 1
      system_error         = 2
      internal_error       = 3
      others               = 4.
  if sy-subrc <> 0.
    message id sy-msgid type sy-msgty number sy-msgno
            with sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.
  endif.
```

d) Save and activate your program.

e) In the top menu, select Goto > Text elements > Selection texts. Enter the following values for the respective parameters: P_CONN: ADS Connection; P_FORM: Form Name; P_IA: Interactive ?. When this is done, save and activate the texts.

2. Generate your form. Save it locally and verify that it acts as a data container.

   a) Run your program. The interactive form appears in the SAPGUI. You can fill up the fields. Save it in local with the appropriate Reader icon, enter some data, close the form the reopen it. Data has been stored in the form.

Lesson Summary

You should now be able to:

- Understand and make use of the main functionalities and elements of the Adobe LiveCycle Designer

Related Information

- SDN Adobe homepage: https://www.sdn.sap.com/irj/sdn/adobe
- SDN ABAP homepage: https://www.sdn.sap.com/irj/sdn/abap

Unit Summary

You should now be able to:

- Understand the architecture used at design time and runtime of SAP Interactive Forms by Adobe
- Know and use the best practices and SAP recommendations in terms of architecture
- Troubleshoot an existing installation and find test programs in an SAP ECC system
- Understand and know where to locate the most important SAP Interactive Forms by Adobe in SAP ECC
- Understand the design and runtime required to generate an interactive form from a traditional ABAP report.
- Write your own report to generate an Interactive Form.
- Understand and make use of the main functionalities and elements of the Adobe LiveCycle Designer

Test Your Knowledge

1. Can you explain stacks required when one system is used ? When multiple systems are used ?

2. What are the commonly used test programs to test ADS installation and configuration ?

3. What is the central source of information for SIFbA ?

4. In which SAP solution can several SIFbA be used as input screens ?

Answers

1. Can you explain stacks required when one system is used ? When multiple systems are used ?

   Answer: SAPInst installs and configures automatically the ADS component when installing a system ABAP + Java.

   When multiple systems are used, communication needs to be setup between the AS ABAP and AS Java, with the SM59 transaction and the J2EE Visual Admin.

2. What are the commonly used test programs to test ADS installation and configuration ?

   Answer: All the programs FP_TEST* available within transaction SE38. The Web Services navigator available on the J2EE engine can also be used.

3. What is the central source of information for SIFbA ?

   Answer: The SDN Adobe homepage: https://www.sdn.sap.com/irj/sdn/adobe

4. In which SAP solution can several SIFbA be used as input screens ?

   Answer: The HCM module features a number of SIFbA within the HCM Processes and Forms functionality.

Internal Use SAP Partner Only

Internal Use SAP Partner Only

Unit 3

SAP Interactive Forms by Adobe in Web Dynpro ABAP environments

Unit Overview

This unit focuses on the most commonly used user interface technology which can embed an interactive form: the SAP Web Dynpro technology.

Unit Objectives

After completing this unit, you will be able to:

- Understand the MVC paradigm
- Understand the basics of the Web Dynpro for ABAP framework to work with interactive forms: controllers, context and actions
- Understand the steps required to integrate an Interactive Form in a Web Dynpro ABAP application
- Develop your own Web Dynpro for ABAP application with an embedded Interactive Form
- Undersand the differences between ZCI and ACF developed forms

Unit Contents

Lesson: The Web Dynpro For ABAP Framework

Lesson Overview

This lesson covers the basis of the Web Dynpro for ABAP framework required to integrate an SAP Interactive Form by Adobe.

Lesson Objectives

After completing this lesson, you will be able to:

- Understand the MVC paradigm
- Understand the basics of the Web Dynpro for ABAP framework to work with interactive forms: controllers, context and actions

Business Example

You project which requires Web Dynpro ABAP knowledge is about to start and you need to keep your team up to date with this technology basics.

The Web Dynpro Basics

What is Web Dynpro ?

A Programming Model for User Interfaces

- Defines a standard structure for user interface applications
  - Derived from the MVC ("model-view-controller") design pattern

A Set of Tools for User Interface Design

- Focus on graphical modelling
  - Code is generated from meta-model declarations
- Integrated in SAP NetWeaver Developer Studio

A Runtime Environment for Applications

- Framework running on SAP's WebAS (J2EE or ABAP) offers common services

A Technology for Software Modularization

- Components help structure projects and support pattern-based UIs

Figure 61: What is Web Dynpro ?

- The classic MVC design creates an architecture for decoupling a data supplier from a data consumer. E.G. It can be used to:
  - Decouple the presentation logic from application logic
  - Decouple the backend business system from the application logic

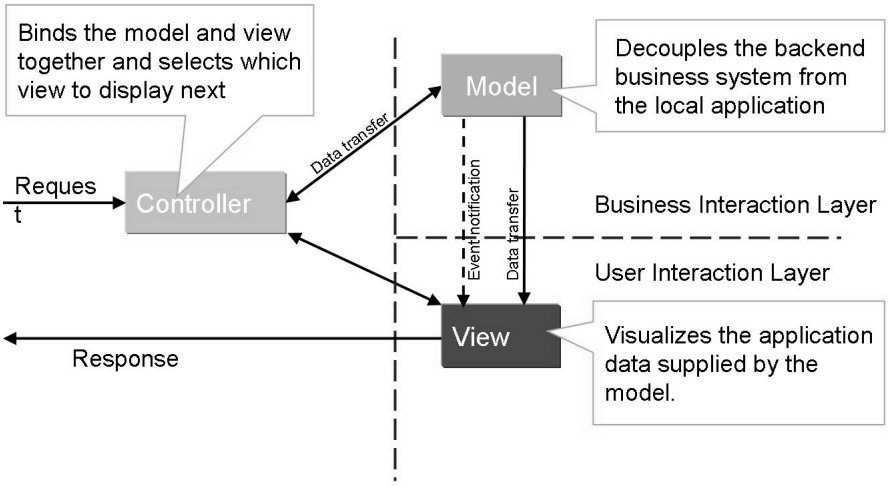


Figure 62: Classic Model View Controller (MVC)

- **Component controller**
  - The lifetime of the component controller equals the lifetime of the component. When starting a Web Dynpro application, the component controller is instantiated by the Web Dynpro runtime.
- **Custom controllers**
  - The instantiation of a custom controller is delayed until the first method of the controller is called. Custom controller instances can not be deleted explicitly.
- **Configuration controllers**
  - This controller is instantiated as the first controller of the component. It lives as long as the component lives.
- **View controllers**
  - The instantiation of a view controller is delayed until the first method of the controller is called.
- The lifetime of a view controller can be controlled by the views properties:
  - If framework controlled is selected, the view instance will be deleted with the component.
  - If when visible is selected, the view instance is deleted as soon as the view no longer belongs to the view assembly.

Figure 63: Web Dynpro Controllers

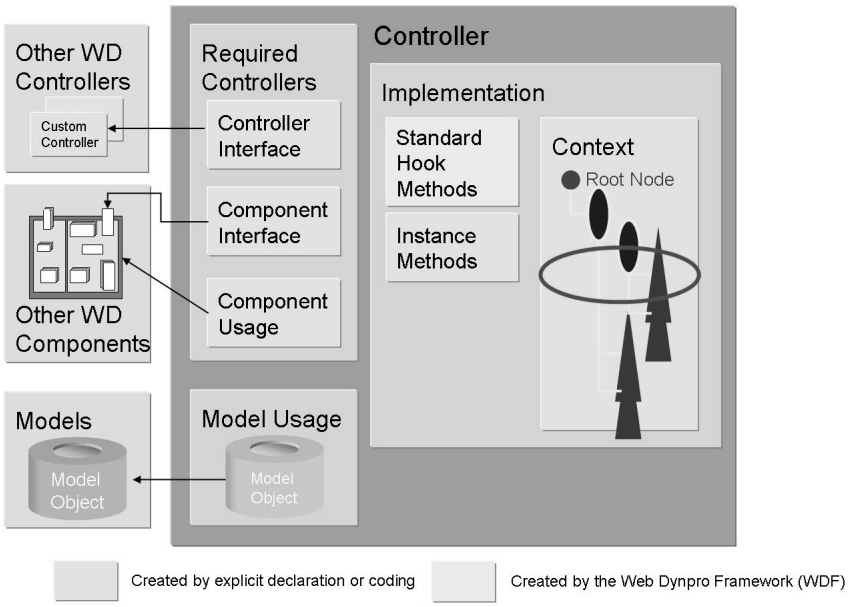
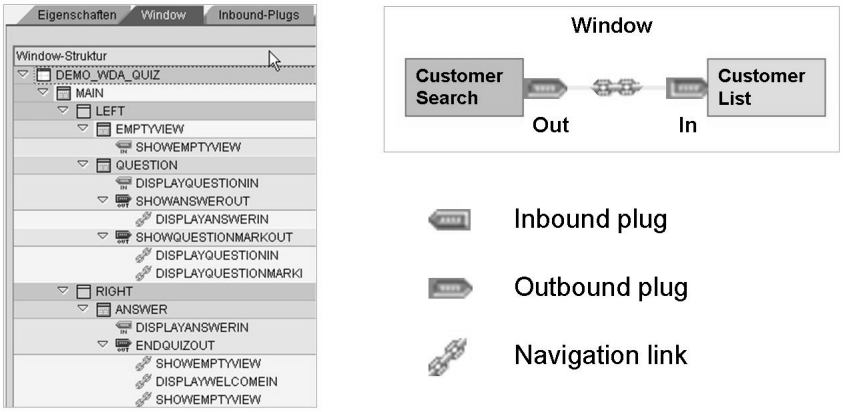


Figure 64: The Context - The heart of a controller



- To define the navigation between two views, you need to create exit and entry points for each view using outbound and inbound plugs.
- Only then you can specify the navigation flow using navigation links

Figure 65: Navigation between views

Lesson Summary

You should now be able to:

- Understand the MVC paradigm
- Understand the basics of the Web Dynpro for ABAP framework to work with interactive forms: controllers, context and actions

Related Information

- SDN Web Dynpro ABAP homepage: https://www.sdn.sap.com/irj/sdn/nw-wdabap

Lesson: The InteractiveForm UI element

Lesson Overview

This lesson focuses on one specific UI element of the Web Dynpro environment, the InteractiveForm UI element. It's a quite powerful but complex one and this lesson deals with its integration with the Web Dynpro for ABAP framework..

Lesson Objectives

After completing this lesson, you will be able to:

- Understand the steps required to integrate an Interactive Form in a Web Dynpro ABAP application
- Develop your own Web Dynpro for ABAP application with an embedded Interactive Form
- Undersand the differences between ZCI and ACF developed forms

Business Example

You're responsible for the development of a Web Dynpro ABAP application featuring SAP Interactive Forms by Adobe.

Prerequisites

The building blocks of the Web Dynpro application are supposed available. The minimum requires:

- A view to embed the PDF
- A window in which the view is embedded
- A component controller (created by default when application is created)
- The context, including dataSource node and pdfSource attribute

This is the most simple of a Web Dynpro application. Very often, multiple views are present, so this requires implementation of navigation between the views.

Integrate an SAP Interactive Form by Adobe in an Web Dynpro ABAP application

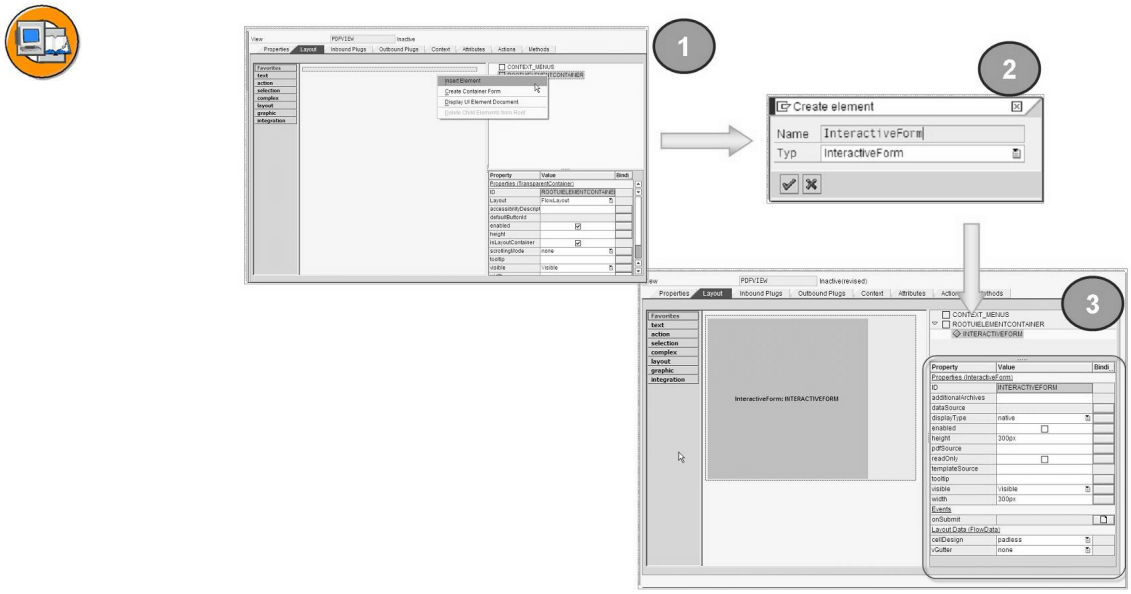


Figure 66: Integrate an Interactive Form in a WDA Application (1)

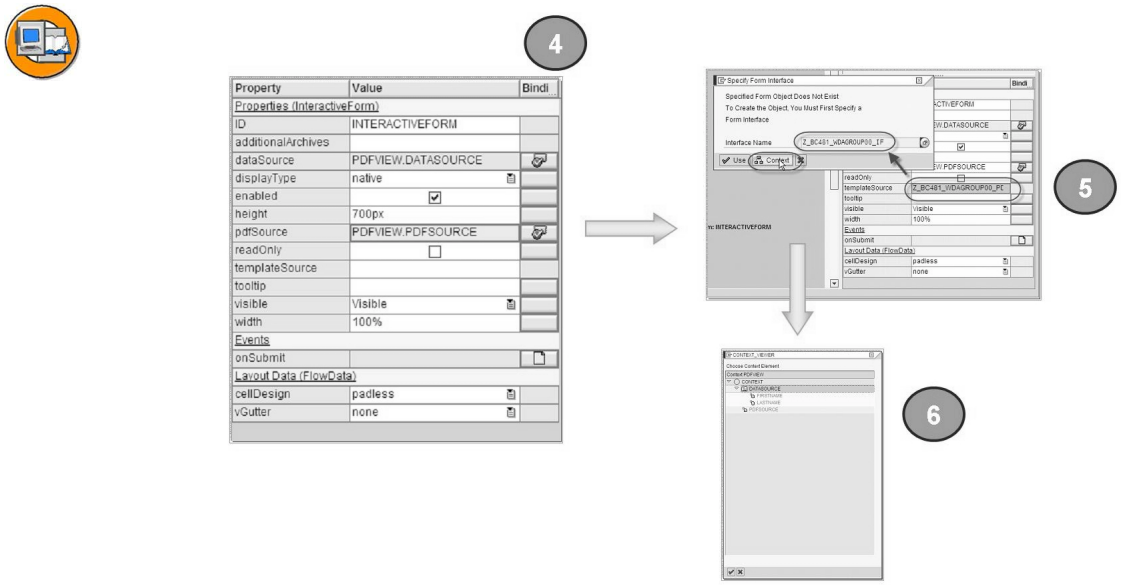


Figure 67: Integrate an Interactive Form in a WDA Application (2)

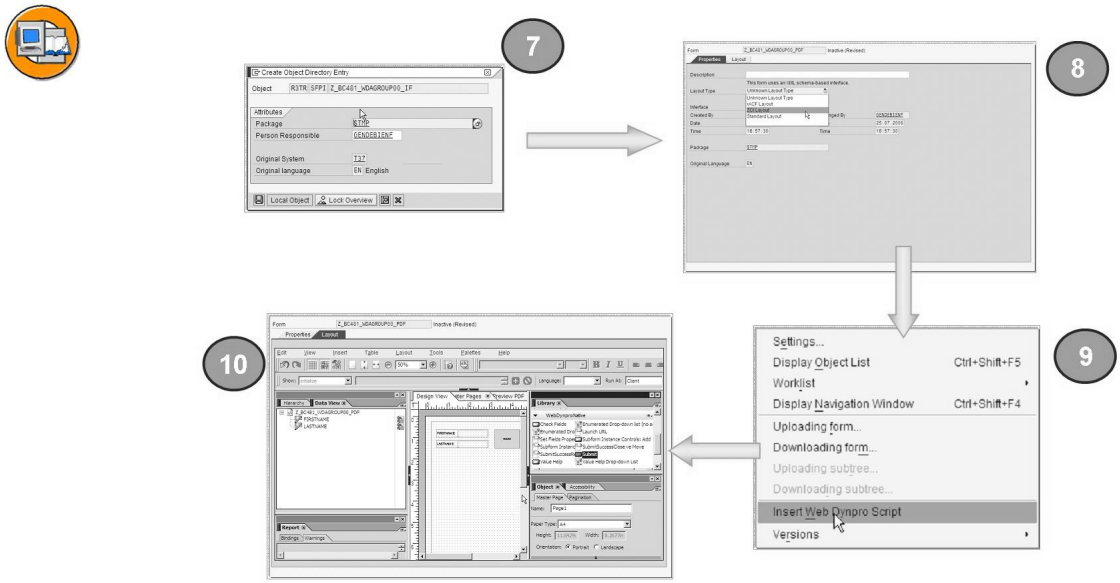


Figure 68: Integrate an Interactive Form in a WDA Application (3)

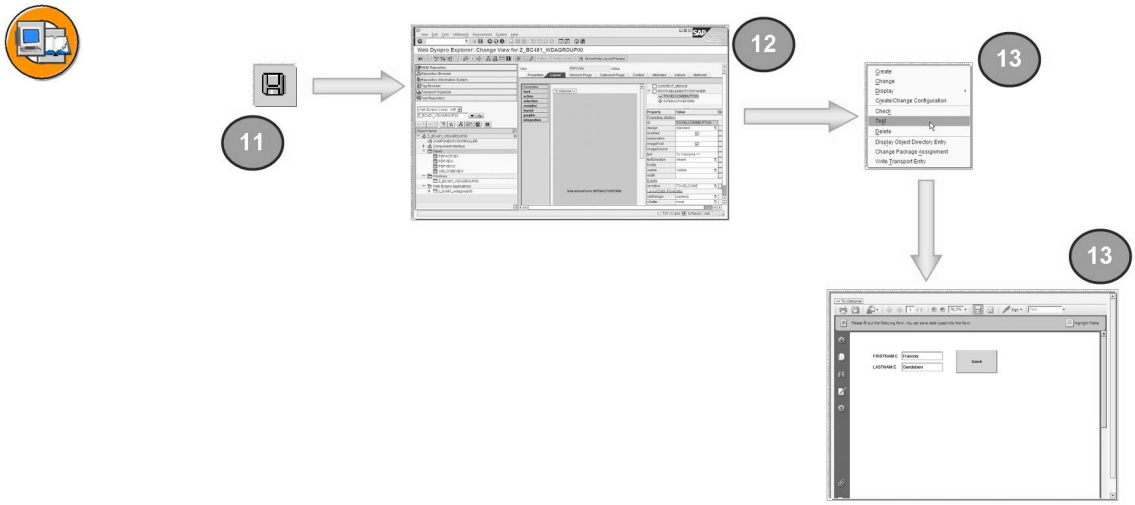


Figure 69: Integrate an Interactive Form in a WDA Application (4)

Every Web Dynpro view contains an element called “ROOTUIELEMENTCONTAINER”, in which all the elements must be inserted.

Right-click on the “ROOTUIELEMENTCONTAINER” and select “Insert Element”. Enter the name and choose “InteractiveForm” as type. This inserts the InteractiveForm UI element in your Web Dynpro View.

When selected, all the properties of this element are displayed in the lower right frame. The following properties must be filled in:

- dataSource: this must refer to a context node.
- pdfSource: this must refer to a context attribute of type XSTRING, which acts as the PDF Container in the view.
- templateSource: this must refer to an actual PDF form associated with an XML-based Interface. (Note: those can be created automatically when designing the WDA application).
- enabled: means generally whether or not an event can be triggered by a user interaction. If the enabled property is set to true, form will be interactive and can act as an input screen.
- read-only: specifies whether the user can select the InteractiveForm element or whether it only is to be displayed as read-only.
- displayType: specifies whether the ActiveX component or the ZCI is used at runtime
- onSubmit event: describes the action to be executed when the user chooses the Submit pushbutton

  Note that the submit event is only evaluated when there a submit button is selected within the form. "Within" means that the special Adobe Web Dynpro submit button has been included in the Adobe Designer.

- **Zero Client Installation**
  - Platform & browser independence
  - No client specific installation
  - Easy maintainability
  - More and more supported features
  - USE_PDF mode, autoscript injection…
  - ZCI must be used for newly created forms
- **Active Components Framework**
  - See SAP Note 766191
  - Will be supported for existing forms
  - ACF forms will be abandoned

Figure 70: Zero Client Installation vs. Active Component Framework

The other properties are:

- ID
- additionalArchive: additional Gantt and Network extensions written by the applications that are packed into a Java archive and loaded in the client.
- Tooltip: quick info text that appears when the user passes the mouse pointer over the UI element
- Height and Width: actual height and width of the form as it will be displayed in the browser window at runtime
- Visible: specifies the visibility of the interface element

The other properties are related to the Web Dynpro view.

Here are the detailed steps to integrate a ZCI form in a Web Dynpro ABAP view as referred in the slides above.

1. Right click on the ROOTUIELEMENTCONTAINER and select Insert Element
2. Select element type InteractiveForm and give it a name (in this example, InteractiveForm)
3. The InteractiveForm UI element now appears under the ROOTUIELEMENTCONTAINER. Display and edit its properties by selecting it
4. Set its main properties (dataSource, displayType, enabled, pdfSource, height and width)
5. Enter a name for the new form template and hit enter. In the popup, enter a name for the new interface and hit the context button
6. Select the context node which must be used to create the interface. In this explanation, we've created a new form template and a new interface based on the context. It's possible to use existing form and templates, in this case the interface must be used within the view with normal ABAP coding
7. Save the newly created objects (form and interface)
8. You're sent to the layout of the form. First, go to the Properties tab and set the Layout Type to ZCI layout
9. Then go back to the layout tab and in the menu go to Utilities > Insert Web Dynpro Script. This command automatically adds JavaScript to the form which allows ZCI usage. This command must not be used in case you're using the xACF layout type. It's important to add the script before adding interactive buttons on the form
10. Create your form layout (in this example with one action button). Use the appropriate buttons in the library (Web Dynpro ActiveX if you use xACF display type and Web Dynpro Native for ZCI layout type).
11. Save and activate your form
12. go back to the Web Dynpro development environment; create a Web Dynpro application and activate it
13. Test your application (display of the form, action buttons and context usage)

- Create a full Web Dynpro ABAP application with data and navigation
- Integrate an Interactive Form in the application
  - Read and write data (from WD Context)
  - Call Web Dynpro actions

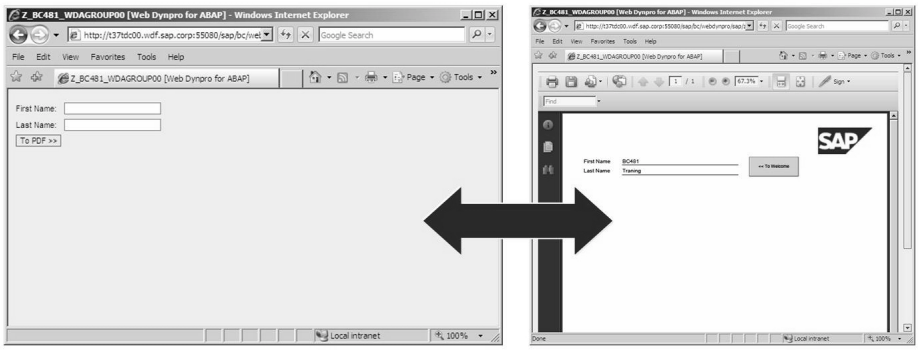


Figure 71: Overview Exercise 3

Exercise 3: Create a Web Dynpro ABAP application with an embedded SAP Interactive Form by Adobe

Exercise Objectives

After completing this exercise, you will be able to:

- Understand design time of Web Dynpro for ABAP with Interactive Form
- Create and run a Web Dynpro ABAP with an embedded SAP Interactive Form by Adobe

Business Example

You have to create an online application with an Interactive Form as input screen.

Task 1:

The first task prepares the views which are used for navigation and data transfer.

1. Create a Web Dynpro ABAP application called Z_BC481_WDAGROUP##, where ## is your group number.
2. Create two views in the application: WELCOMEVIEW and PDFVIEW. Embed it in the default window, the WELCOMEVIEW being the default view.
3. Implement navigation between those views with the help of navigation plugs.

Task 2:

In this task, you'll prepare the data which will be sent from one view to another as well as the PDF container node.

1. In the COMPONENTCONTROLLER, create a context node called DATASOURCE with two context attributes called FIRSTNAME and LASTNAME both of type STRING
2. Map all the context data to the two views, WELCOMEVIEW and PDFVIEW
3. In the view PDFVIEW, create a context attribute of type XSTRING called PDFSOURCE

Continued on next page

Task 3:

Now the navigation links and data have been built, you'll create an input form for the first view and you'll embed an actual PDF element in the second view, and set all the required properties for this particular UI element.

1. In the layout of the WELCOMEVIEW, add a container form for the two elements in the context and create a button to navigate to the PDFVIEW
2. In the PDFVIEW, insert an element of type InteractiveForm. Set the dataSource property to the context node DATASOURCE, the displayType to native, the enabled property to true, the pdfSource to the context attribute PDFSOURCE, the height to 700px and the width to 100%.

Task 4:

And XML-based interface needs to be created for a form used by the Web Dynpro application. The layout of the form still needs to be done with the help of the Adobe LiveCycle Designer.

1. Once this is done, the PDF can be designed. In the templateSource, enter Z_BC481_WDAPDF_## as value and hit enter. Specify Z_BC481_WDAIF_## name for the interface and create the interface from the context node DATASOURCE. Save the two created objects in your package using the previously created workbench request. This sends you to the actual design of the form: add the two context fields on the form and a submit button. Select the submit button from the Web Dynpro Native library. (IMPORTANT: before to start the design of the form, go to the properties tab and select ZCI Layout in the layout field. Then go back to the Layout tab and go to Utilities > Insert Web Dynpro script.)
2. Create an action for the onSubmit event in which you call the outbound plug TOWELCOME

Task 5:

In this task, you'll build an actual Web Dynpro application which you'll execute.

1. Create a Web Dynpro application called Z_BC481_WDAGROUP## and test it.

Solution 3: Create a Web Dynpro ABAP application with an embedded SAP Interactive Form by Adobe

Task 1:

The first task prepares the views which are used for navigation and data transfer.

1. Create a Web Dynpro ABAP application called Z_BC481_WDAGROUP##, where ## is your group number.

   a) Go to transaction SE80, select Web Dynpro Comp. / Intf. from the drop down box. Enter the name of the application and hit enter.

   b) Create the application mentioning the name, description (such as BC481 WDA and PDF Group ##), type (Web Dynpro Component).

   c) Correct the values for the automatically created View Name by changing the default name MAIN to WELCOMEVIEW. The Window name can stay the same, e.g. Z_BC481_WDAGROUP##.

   d) Save your application in your package using the workbench request you've created previously.

2. Create two views in the application: WELCOMEVIEW and PDFVIEW. Embed it in the default window, the WELCOMEVIEW being the default view.

   a) The WELCOMEVIEW has been automatically created on the previous step. You should see it in the views of your Web Dynpro component (bottom left side of the screen).

   b) On the left-hand navigation, right click on the application name and select Create > View. Another option is to go to the application structures, right click on Views and select Create

   c) Save your work when you're prompted to. Enter view name (PDFVIEW) and description.

   d) Open the Window (left hand navigation > Z_BC481_WDAGROUP## > Windows > Z_BC481_WDAGROUP## and double click on it). Right click on the Window, select Embed View then choose PDFVIEW from the drop down menu.

   e) Right click on the view WELCOMEVIEW and select set as default. Normally WELCOMEVIEW is already the default view.

Continued on next page

3. Implement navigation between those views with the help of navigation plugs.

   a) Open the WELCOMEVIEW. In the appropriate tabs (Outbound plugs and Inbound plugs), create an outbound plug called TOPDF and an inbound plug called FROMPDF

   b) Open the PDFView. In the appropriate tabs (Outbound plugs and Inbound plugs), create an outbound plug called TOWELCOME and an inbound plug called FROMWELCOME.

   c) Open the window (Z_BC481_WDAGROUP##) and link the outbound plugs to the inboung plugs accordingly by drag and drop. Leave the default inbound plug as it is.

Task 2:

In this task, you'll prepare the data which will be sent from one view to another as well as the PDF container node.

1. In the COMPONENTCONTROLLER, create a context node called DATASOURCE with two context attributes called FIRSTNAME and LASTNAME both of type STRING

   a) Open the COMPONENTCONTROLLER by double-clicking on it

   b) In the Context tab, right click on the context node and select Create > Node. Name the node DATASOURCE. Leave other values to the proposed ones.

   c) Under the newly created context node, create the two context attributes called FIRSTNAME and LASTNAME, both of type STRING. Leave other values to the proposed ones.

2. Map all the context data to the two views, WELCOMEVIEW and PDFVIEW

   a) Open the Context tab of the view WELCOMEVIEW.

   b) Drag and drop the DATASOURCE context node to the context of the view WELCOMEVIEW

   c) Do the same operation for the view PDFVIEW

3. In the view PDFVIEW, create a context attribute of type XSTRING called PDFSOURCE

   a) Right click on the root context node and add the PDFSOURCE attribute. Specify XSTRING as type.

Continued on next page

Task 3:

Now the navigation links and data have been built, you'll create an input form for the first view and you'll embed an actual PDF element in the second view, and set all the required properties for this particular UI element.

1. In the layout of the WELCOMEVIEW, add a container form for the two elements in the context and create a button to navigate to the PDFVIEW
   a) In the Layout tab of the WELCOMEVIEW, on the right hand, right click on ROOTUIELEMENTCONTAINER and select Create container form.
   b) Adjust the labelds for the input fields and set them to Firstname and Lastname accordingly.
   c) Click on the Context button and select the DATASOURCE node
   d) Right click in the ROOTUIELEMENTCONTAINER and select Insert Element. Name the element TOPDFBUTTON and choose Button as a type
   e) In the properties of this button, set the text property to To PDF >>
   f) On the onAction event, click on the white sheet to create a new action. Name it TOPDF and select the outbound plug TOPDF. After this, the WELCOMEVIEW is completed.
2. In the PDFVIEW, insert an element of type InteractiveForm. Set the dataSource property to the context node DATASOURCE, the displayType to native, the enabled property to true, the pdfSource to the context attribute PDFSOURCE, the height to 700px and the width to 100%.
   a) In the layout tab of the PDFVIEW, right click on the ROOTUIELEMENTCONTAINER and select Insert element. Name your element INTERACTIVEFORM and select InteractiveForm as a type.
   b) Set the dataSource property to the DATASOURCE context node by clicking on the button next to the dataSource property. A popup appears in which you can select your DATASOURCE node.
   c) Set the displayType property to native
   d) Set the enabled property to true by selecting the checkbox
   e) Set the context attribute PDFSOURCE for the pdfSource property by clicking on the button next to the pdfSource property. A popup appears in which you can select your PDFSOURCE attribute.
   f) Set the height property to 700px
   g) Set the width property to 100%

Continued on next page

Task 4:

And XML-based interface needs to be created for a form used by the Web Dynpro application. The layout of the form still needs to be done with the help of the Adobe LiveCycle Designer.

1. Once this is done, the PDF can be designed. In the templateSource, enter Z_BC481_WDAPDF_## as value and hit enter. Specify Z_BC481_WDAIF_## name for the interface and create the interface from the context node DATASOURCE. Save the two created objects in your package using the previously created workbench request. This sends you to the actual design of the form: add the two context fields on the form and a submit button. Select the submit button from the Web Dynpro Native library. (IMPORTANT: before to start the design of the form, go to the properties tab and select ZCI Layout in the layout field. Then go back to the Layout tab and go to Utilities > Insert Web Dynpro script.)

   a) Execute the above mentioned steps.

   b) You're automatically sent to the form design tool (the Adobe LiveCycle Designer), where you can insert by drag and drop the two context fields on the form. Use layout data from previous exercise.

   c) Add a submit button on the form. IMPORTANT: use the Submit button in the WebDynpro Native library. Use following layout properties: X: 4.75in; Y: 1.125in; width: 1.25; height: 0.5in. Change the label of the Submit button from Submit to << To Welcome.

   d) When this is done, activate the form. (You can aslo activate all your other objects).

Continued on next page

2. Create an action for the onSubmit event in which you call the outbound plug TOWELCOME

   a) Go back to the layout tab of the PDFVIEW and select your InteractiveForm UI element to display its properties.

   b) Click on the white sheet next to the onSubmit event (in the form properties) to create an action. Call your action TOWELCOME. Then don't click on continue but click on the Implementation button (or press F5). You're then sent to the implementation code of the TOWELCOMEACTION.

   c) Add the following code to the method:
   ```
   wd_this->fire_towelcome_plg(
     ).
   ```
   This will trigger the outbound plug to the WELCOMEVIEW.

   d) Another possibility is to generate this code with the Web Dynpro Code Wizard. When you're in the code, hit CTRL + F7. The wizard window appears. Check the Start Navigation radio button and select the TOWELCOME outbound plug with the help of the F4 key.

   e) The implementation of the PDFVIEW is now finished.

Task 5:

In this task, you'll build an actual Web Dynpro application which you'll execute.

1. Create a Web Dynpro application called Z_BC481_WDAGROUP## and test it.

   a) Right click on your Web Dynpro Component (called Z_BC481_WDAGROUP##) and select Create > Web Dynpro Application. Save the created object in your package using the previously created workbench request.

   b) Save the created application. Right click on the generated application and select Test. Your application appears in a new browser window.

   c) Fillup the fields in the first screen, then go to the PDF View. Your data are sent from one view to another. Modify the data and test the Submit button available on the form. You're sent back to the first view. Data have been sent from the PDF View to the Welcome View via the Web Dynpro context.

Lesson Summary

You should now be able to:

- Understand the steps required to integrate an Interactive Form in a Web Dynpro ABAP application
- Develop your own Web Dynpro for ABAP application with an embedded Interactive Form
- Undersand the differences between ZCI and ACF developed forms

Unit Summary

You should now be able to:

- Understand the MVC paradigm
- Understand the basics of the Web Dynpro for ABAP framework to work with interactive forms: controllers, context and actions
- Understand the steps required to integrate an Interactive Form in a Web Dynpro ABAP application
- Develop your own Web Dynpro for ABAP application with an embedded Interactive Form
- Undersand the differences between ZCI and ACF developed forms

Test Your Knowledge

1. What are the two typical Interactive Form UI element to be linked with the context ?

Answers

1. What are the two typical Interactive Form UI element to be linked with the context ?

   Answer: The DataSource and pdfSource properties.

Internal Use SAP Partner Only

Internal Use SAP Partner Only

Unit 4

Advanced Features

Unit Overview

This unit focuses on various commonly used SAP Interactive Forms by Adobe capabilities. It goes beyond SAP only for some parts: email, offline scenarios, complex layouts, performance aspects and PDFOBJECT ABAP API.

This unit can be seen as a toolset which can be used when asserting a new scenario, to deal with performance issues of existing forms or to explore new ways of developing forms.

Unit Objectives

After completing this unit, you will be able to:

- Understand concepts and possibilities in offline scenarios which can be used with SAP Interactive Forms by Adobe
- Understand considerations to have in mind when assessing an offline scenario
- Understand and make use of Interactive Form with email
- Develop this functionality on an existing Interactive Form
- Understand and handle complex layouts and frequently met form elements
- Estimate the complexity of an Interactive form
- Understand the main factors influencing the performance of a PDF form
- Understand and make use of the best practices for PDF forms design
- Know where to locate and how to use the PDFOBJECT API
- Show examples using the API in the SAP system
- Show how to use the API with comments, attachments and digital signatures
- Understand the possibilities offered by digital signatures within SIFbA
- Understand and explain what can be done with SIFbA digital signatures
- Identify configuration required to use digital signatures in an SAP environment

Unit Contents

Lesson: Offline Scenarios

Lesson Overview

This lesson shows the main offline scenarios which can be used with SAP Interactive Forms by Adobe.

Lesson Objectives

After completing this lesson, you will be able to:

- Understand concepts and possibilities in offline scenarios which can be used with SAP Interactive Forms by Adobe
- Understand considerations to have in mind when assessing an offline scenario

Business Example

Your company needs to make use of offline scenarios. As you know that SIFbA can be used for this purpose, you're studying the feasability of some specific scenarios.

Online and Offline Scenarios

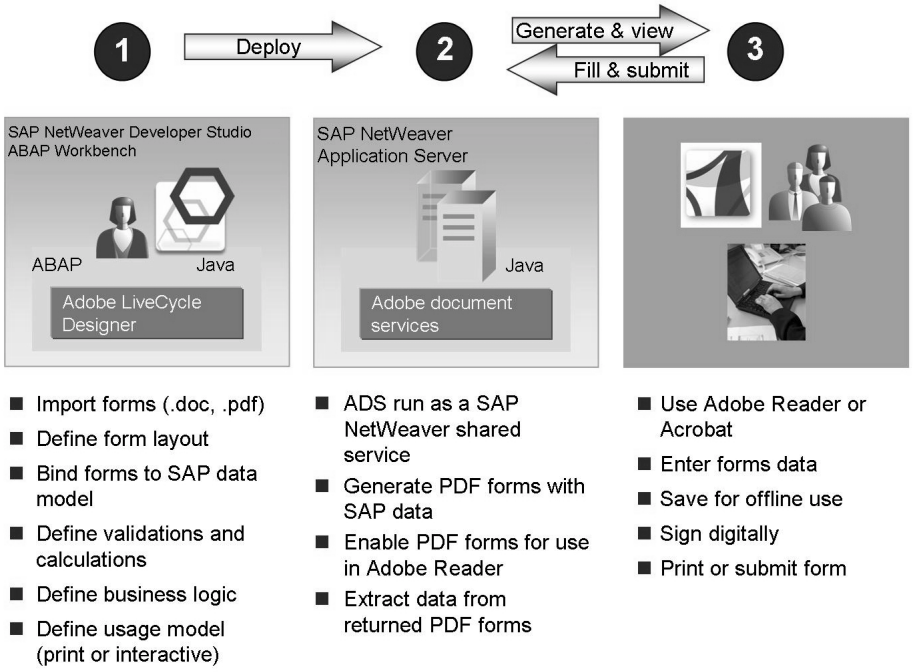


Figure 72: General scenarios featuring Interactive Forms

- **Web Dynpro Java and ABAP**
  - Link with ABAP backends, workflows, web services etc.
  - Embed in portal
- **Guided Procedures**
- **Internet Service Request**

Figure 73: Online mode

As discussed earlier in the course, the online usage of interactive forms allows you to use all the functionalities of the online applications themselves.

- Web Dynpro ABAP and Java provides the development and runtime environment for online applications. The Web Dynpro framework handles connection between the online application and the backend system, SAP or non-SAP.

- **Standalone offline mode**
  - For simple workflows
- **Offline mode with upload functionality**
  - Receiver application
- **System handled offline mode**
  - Interfaces & data parser
  - Auto-send mode needs to be built

Figure 74: Offline mode

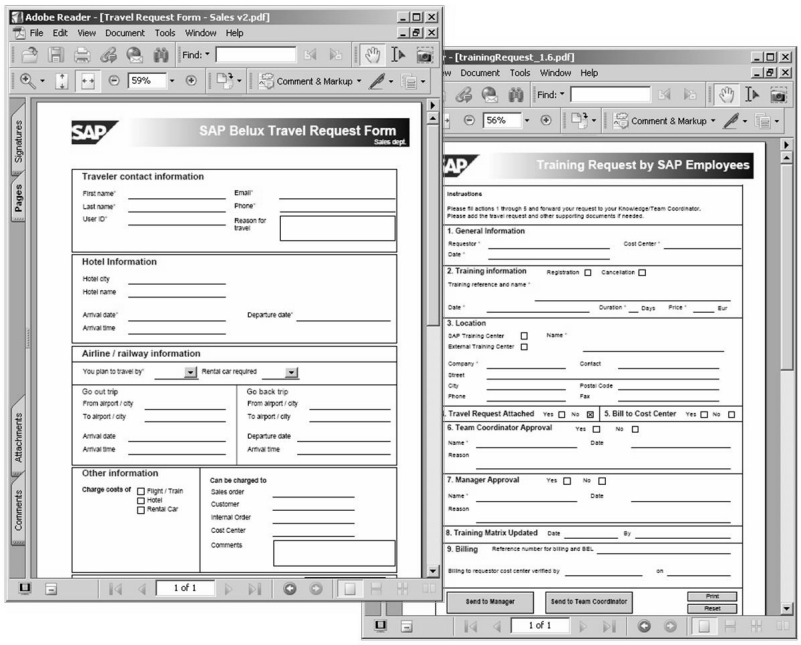


Figure 75: Offline Standalone Usage: simple workflows

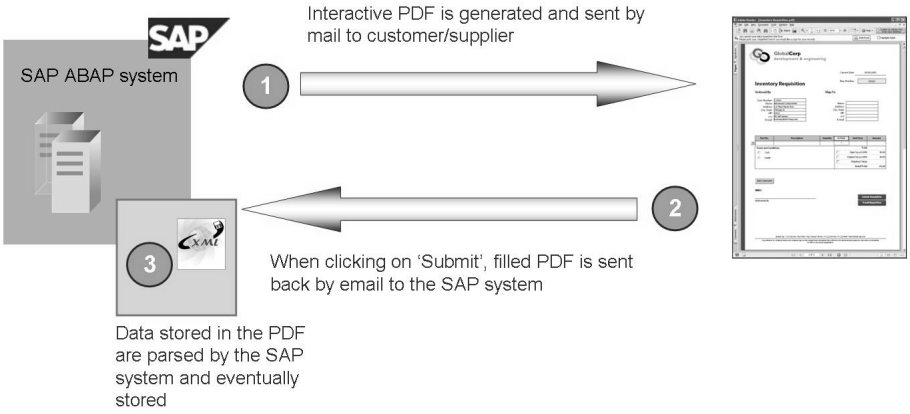


Figure 76: Offline scenario with backend intervention (1)

1. The PDF is sent by the SAP system
2. Make use of certification / digital signature(s) from SAP side (server side signature)
3. Prepare certification / digital signature(s) from the receiver (customer or supplier) side (client side signature)
4. Data extraction, certificates and signature validation are performed when the PDF is sent back to the SAP system

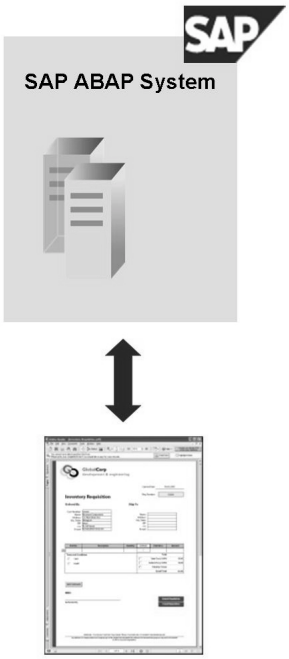


Figure 77: Offline scenario with backend intervention (2)

- **Regular mail: XML data parsing** (and PDF extraction)
- **Upload functionality**

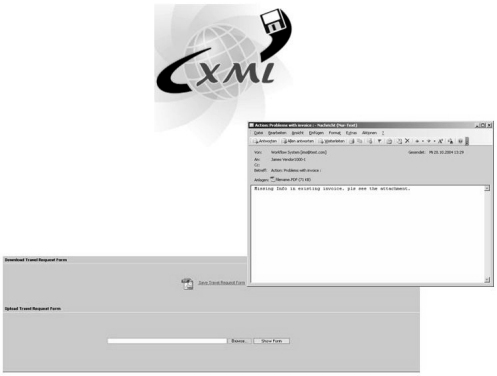


Figure 78: Offline mode with backend intervention (3)

Internal Use SAP Partner Only

- **Define clearly your usage and your scenario**
  - Online / Offline?
  - Interactive / Non interactive (Print)?
  - Workflows?
  - Architecture (sizing, costs...)?

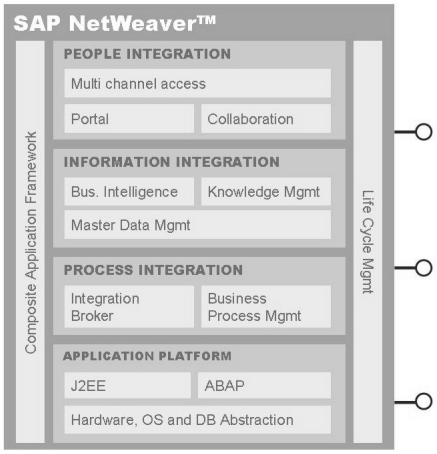


Figure 79: Scenarios: important considerations

Internal Use SAP Partner Only

Lesson Summary

You should now be able to:

- Understand concepts and possibilities in offline scenarios which can be used with SAP Interactive Forms by Adobe
- Understand considerations to have in mind when assessing an offline scenario

Related Information

- SDN Adobe homepage: https://www.sdn.sap.com/irj/sdn/adobe
- Adobe LiveCycle Devnet: http://www.adobe.com/devnet/livecycle/

Lesson: Email Sending

Lesson Overview

This lesson focuses on one of the most commonly used scenarios for offline usage: the standalone email sending, where the PDF itself acts as a data container and is exchanged by email.

Lesson Objectives

After completing this lesson, you will be able to:

- Understand and make use of Interactive Form with email
- Develop this functionality on an existing Interactive Form

Business Example

Your company wants to make use of Interactive Forms to replace existing Word documents or Excel sheets to ensure validity of data. You're responsible of this new practice.

Send an Interactive Form by email

- Open a form embedded in a WD View by double clicking on the property templateSource

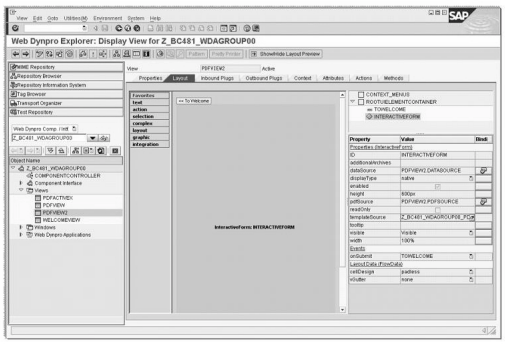
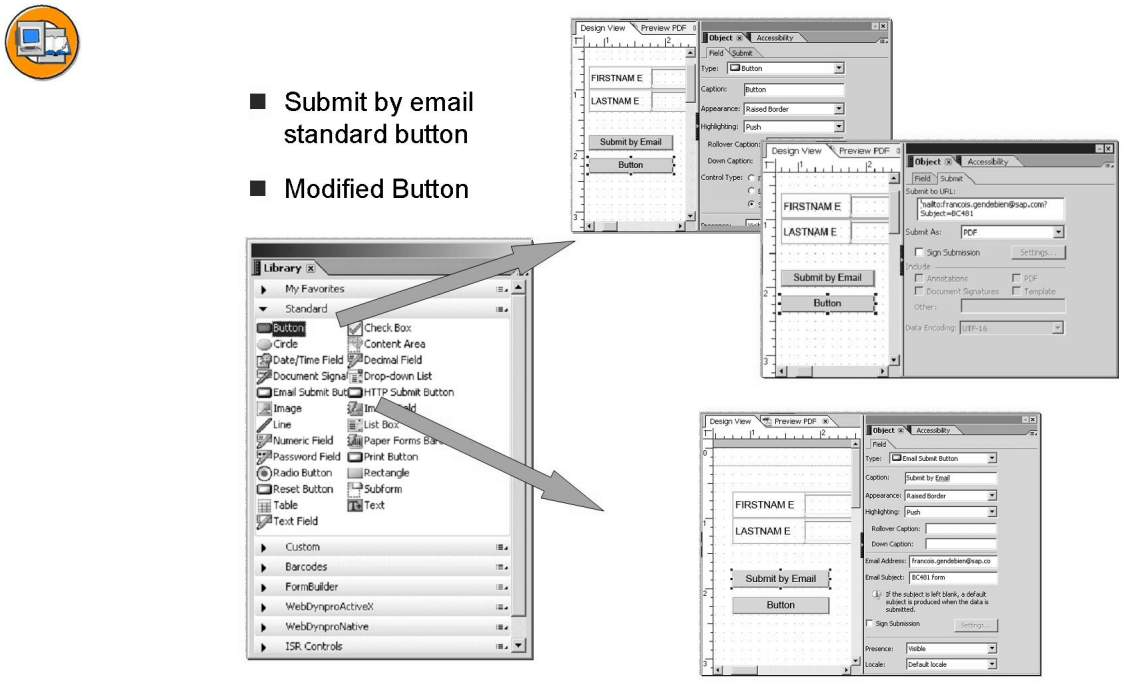


Figure 80: Edit the existing form



- Submit by email standard button
- Modified Button

Figure 81: Submit by email (1)

- Submit by email standard button
  - Submit XML content
- Standard Button, modified
  - Allows to submit to an http address, such as mailto:firstname.lastname@company.com?Subject=PDF%20Email
  - Allows to send XML or PDF file

Remark: form must be interactive (ADS-generated)

Figure 82: Submit by email (2)

Lesson Summary

You should now be able to:

- Understand and make use of Interactive Form with email
- Develop this functionality on an existing Interactive Form

Related Information

- SDN Adobe hompage: https://www.sdn.sap.com/irj/sdn/adobe
- Adobe LiveCycle Devnet: http://www.adobe.com/devnet/livecycle/
- Adobe LiveCycle Devnet scripting examples: http://www.adobe.com/devnet/livecycle/designer_scripting_samples.html

Lesson: Complex Layouts

Lesson Overview

This short lesson show classic complex layouts: the dynamic tables and the general UI elements which should be used for the vast majority of the forms.

Lesson Objectives

After completing this lesson, you will be able to:

- Understand and handle complex layouts and frequently met form elements
- Estimate the complexity of an Interactive form

Business Example

Your team faces some issues with development of complex forms. You're asked to support them in their tasks and to provide valuable input.

Dynamic Tables

- **Subforms are more suited to dynamic forms where the form design is not as easily defined within the confines of a grid structure.**
- **Subforms also provide the overall structure for your form. For example, to create a dynamic form, you need position content forms for static content and at least one flow content subform to enable the expansion of dynamic content.**
- **NOTE: The root node of the form is set to Flow and cannot be changed.**

Figure 83: Tables Or Subforms

- You can set the entire table or just the body row to repeat based on incoming data, or based on user interaction.

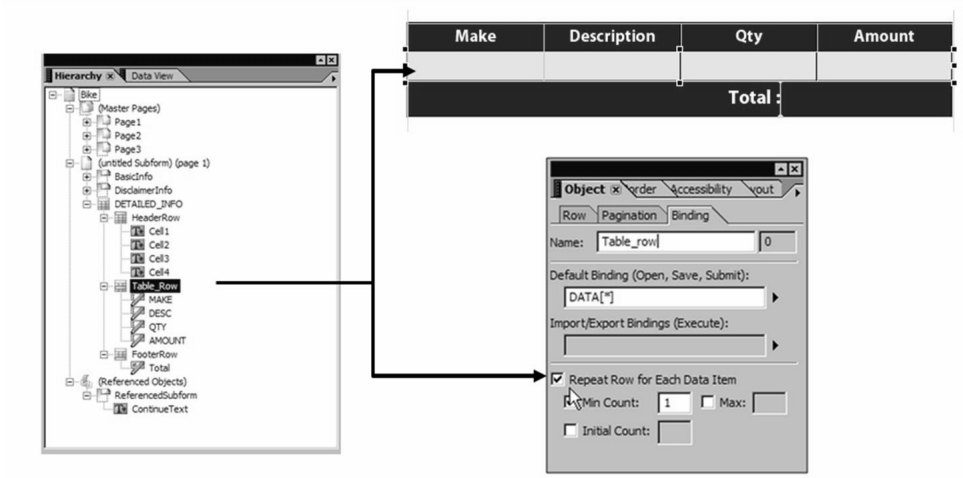


Figure 84: Dynamic Tables Properties

- Old fashioned way: subforms
- New way: tables
  - Example with form FP_TEST_NESTED_TABLE

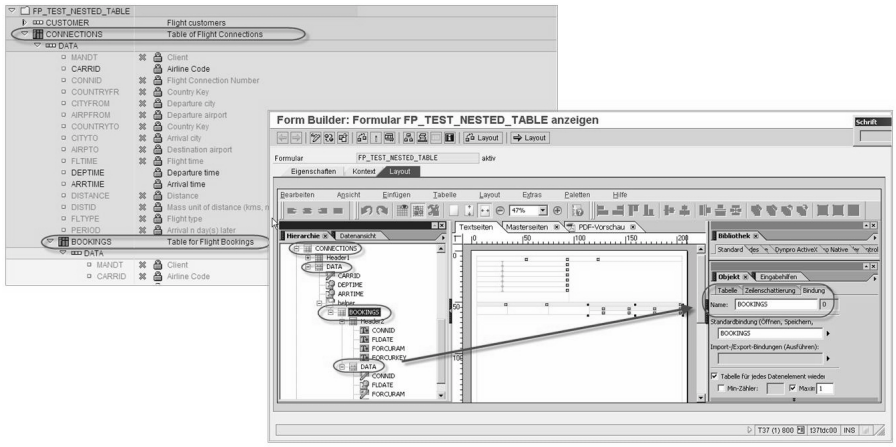


Figure 85: Dynamic Tables

Miscellaneous features

- Locale
- Display Pattern
- Floating fields
- Page numbering

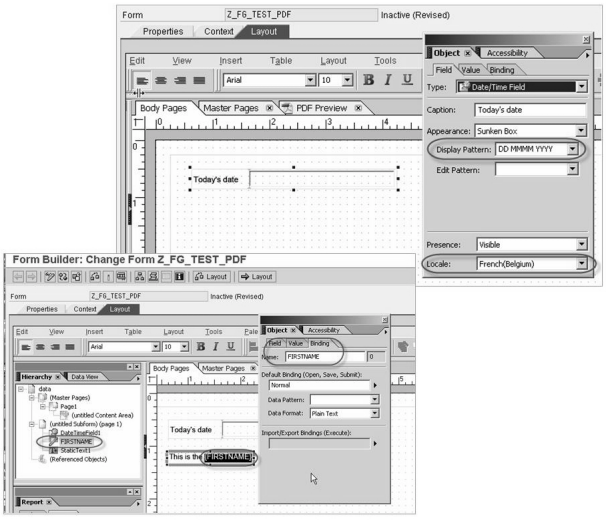


Figure 86: Miscellaneous features

Exercise 4: Adobe LiveCycle Designer Advanced Features

Exercise Objectives

After completing this exercise, you will be able to:

- Use typical and most commonly used UI elements on a PDF form
- Design Complex layouts with the Adobe LiveCycle Designer

Business Example

You're a part of the project team and you're asked to write a detailed specification for a PDF form (print or interactive), or to develop the form itself.

Task 1:

The form you're developing is quite complex and features various display elements. In this first task, focus is on usage of buttons on a form.

1. Open the form you've created on the first exercise and add a Send by Email button to send the PDF to your email address.
2. Add a button called Button1. Add a button to make it invisible and another one to make it visible, using FormCalc and the appropriate event.

   Test the functionality with the PDF Preview tab.

Task 2:

Data from your backend has to be displayed in various formats. In this task, focus is on displaying the data.

1. Create a static text with two floating fields for FIRSTNAME and LASTNAME. Text is the follwing:

   Dear M. {FIRSTNAME} {LASTNAME},

   this is an example of floating fields.

   Test your form from transaction SFP and see the result.
2. Create a dynamic table with columns FIRSTNAME and LASTNAME. Test the form by entering two lines in the Test Function Module Screen.

Continued on next page

Task 3:

In this task, the focus is on various LiveCycle Designer standard features.

1. Add a page numbering of type Page # / ## (page number / total number of pages) on the master page. (For the text element, X position is 4in, Y position is 0.5in, width is 1.5in and height is 0.25in)
2. Add a Date field. Test it with PDF Preview tab, then set the locale to English (Belgium) and set the display pattern so the month is written in full text.

Solution 4: Adobe LiveCycle Designer Advanced Features

Task 1:

The form you're developing is quite complex and features various display elements. In this first task, focus is on usage of buttons on a form.

1. Open the form you've created on the first exercise and add a Send by Email button to send the PDF to your email address.

   a) Go to transaction SFP, enter your form name, ZBC481_EX01_FORM##, where ## is your group number and hit the Change button.

   b) Delete the two existing fields Firstname and Lastname from the existing layout.

   c) In the layout tab, drag and drop an element of type Button from the Standard library. Change it label to Send by email. Change its name in the Hierarchy palette to SendByEmail.

   d) Use following layout properties for the button: X: 0.5in; Y: 1in; width: 1.125in; height: 0.25in

   e) In the Object palette for this element, set the Control Type property to Submit. In the Submit tab, fill in the Submit To URL field with the value: mailto:firstname.lastname@company.com?Subject=BC481%20PDF. Set also the Submit As field to PDF.

2. Add a button called Button1. Add a button to make it invisible and another one to make it visible, using FormCalc and the appropriate event.

Continued on next page

Test the functionality with the PDF Preview tab.

a) Drag and drop an element of type Button from the Standard library. In the hierarchy palette, name it Button1. Change its label to Button1.

b) Use following layout properties for the button: X: 0.5in; Y: 1.5in; width: 1.125in; height: 0.25in.

c) Drag and drop an element of type Button from the Standard library. In the hierarchy palette, name it Hide. Change its label to Hide. In the scripting editor, add the following FormCalc code to the click event: `Button1.presence = "invisible"`(no dot is inserted at the end of this instruction).

   Be sure that the language selected is FormCalc and that the Run At field is set to Client.

d) Use following layout properties for the hide button: X: 2in; Y: 1.5in; width: 1.125in; height: 0.25in.

e) Drag and drop an element of type Button from the Standard library. In the hierarchy palette, name it Show. Change its label to Show. In the scripting editor, add the following FormCalc code to the click event: `$.parent.Button1.presence = "visible"`(no dot is inserted at the end of this instruction).

   Be sure that the language selected is FormCalc and that the Run At field is set to Client.

f) Use following layout properties for the show button: X: 3.5in; Y: 1.5in; width: 1.125in; height: 0.25in.

g) Go to the PDF Preview tab in the Designer and test the change of presence of the button Button1.

Task 2:

Data from your backend has to be displayed in various formats. In this task, focus is on displaying the data.

1. Create a static text with two floating fields for FIRSTNAME and LASTNAME. Text is the follwing:

   Dear M. {FIRSTNAME} {LASTNAME},

   this is an example of floating fields.

Continued on next page

Internal Use SAP Partner Only

Test your form from transaction SFP and see the result.

a) Go to the library palette, drag and drop an element text from the Standard library.

b) Within the layout palette, set its X position to 0.5in, Y position to 2in, Width to 4in and Height to 2in.

c) Add the text above.

d) To add the floating fields, go to the Designer menu: Insert > Floating field.

e) Double click on the newly created field and go to the Object palette, in the tab Binding. In the Default binding field, using the appropriate icon, select the FIRSTNAME field from the context: Z_BC481_EX01_PDF_## > PDFDATA > FIRSTNAME. Select update all related properties when you're prompted to (after you've bound the element).

f) Create an additional floating field for the LASTNAME data element, using the same technique.

g) Test the form with the appropriate icon, or press F8. In the text function module screen, click on the table PDFDATA. Enter your Firstname and Lastname in the appropriate fields, then go gack using the green back arrow or the F3 button. Select LP01 as the output device when you're prompted to and hit the Print Preview button.

h) Check the difference in the display of the normal input fields (which are read only) and the floating fields.

Continued on next page

2. Create a dynamic table with columns FIRSTNAME and LASTNAME. Test the form by entering two lines in the Test Function Module Screen.

   a) Go to the library palette, drag and drop an element Table from the Standard library in the Design view of the form.

   b) This opens up the table wizard. Select Create simple table, with 2 columns, 1 body row and check Include header row in Table.

   c) Use following layout properties : X: 0.5in; Y: 4.25in

   d) Change the label of the header row to respectively Firstname and Lastname.

   e) In the row itself, in each cell add a floating field: double click in a cell and go to the Designer menu Insert > Floating Field. Then bind the fields respectively to FIRSTNAME and LASTNAME.

   f) In the hierarchy palette, select the table row: in your table (Table1 by default), Row1 by default. Go to the Object palette and check the following properties under the binding tab. Default binding must be set to $record.PDFDATA.DATA[*], and the checkbox Repeat Row for Each Data Item must be checked. Leave other properties to their defaults.

   g) Activate you form and test it using the Test or F8 button. In the Test function module initial screen, click on the PDFDATA table and create two rows (use the New Line button) which you can fill with some data. Go back using the back arrow or F3 button, run the function module using F8. Select LP01 as the output device and hit Print Preview. Check the result of your dynamic table. (Hint: you can also check how your previous fields behave when you add several additional lines).

Continued on next page

Task 3:

In this task, the focus is on various LiveCycle Designer standard features.

1. Add a page numbering of type Page # / ## (page number / total number of pages) on the master page. (For the text element, X position is 4in, Y position is 0.5in, width is 1.5in and height is 0.25in)

   a) Go to the library palette, drag and drop an element text from the Standard library.

   b) With the layout palette, set the positioning and layout as stated above.

   c) Delete the texts in the text element and go to the Designer menu: Insert > Current Page Number. Add a space and a / character after this element. Then go to the same menu and select Number of Pages.

   d) Test the form using the PDF preview tab, the appropriate Test icon or via the F8 button.

2. Add a Date field. Test it with PDF Preview tab, then set the locale to English (Belgium) and set the display pattern so the month is written in full text.

   a) Drag and drop an element of type Date/Time Field from the Standard library.

   b) Use following layout properties for the show button: X: 0.5in; Y: 2.25in; width: 4in; height: 0.25in

   c) Test the default result with the PDF Preview tab.

   d) In the Object palette, set the following value to the Display Pattern: DD MMMM YYYY

   e) In the Object palette, set the Locale property to English (Belgium).

   f) Test the result with the PDF Preview tab.

Lesson Summary

You should now be able to:

- Understand and handle complex layouts and frequently met form elements
- Estimate the complexity of an Interactive form

Related Information

- SDN Adobe homepage: https://www.sdn.sap.com/irj/sdn/adobe

Internal Use SAP Partner Only

Lesson: Performance Aspects

Lesson Overview

As forms are becoming more and more complex, users often face performance issues. So it's important to have in mind the main performance guidelines at the beginning of the development.

Lesson Objectives

After completing this lesson, you will be able to:

- Understand the main factors influencing the performance of a PDF form
- Understand and make use of the best practices for PDF forms design

Business Example

Your team is facing performance issues with its Interactive Forms and you need to provide support to improve rendering times and usability of the forms.

General Design Considerations

[Enter a title and the conceptual information about this lesson in this section. You can also include additional sections, graphics, demonstrations, procedures, and/or simulations.

- Avoid using fonts that must be embedded, particularly for field objects. Embedded fonts cannot be subset.
- Use embedded images instead of linked images. By default, images are linked.
- Use a compressed graphic format such as JPEG or GIF.
- Use Palette or Monochrome images for color. Avoid using 24-bit color formats.
- Be aware that generating accessible forms affects performance.
- Use only the minimum number of objects in forms and keep objects as simple as possible.

Figure 87: General Design Considerations (1)

- Group data elements in subforms
- Group display elements in subforms
- Limit amount of scripting (eg. Validation, business logic) and rather do this in the backend
- Limit amount of data to be available in the form
- A form is a form, not an application

Figure 88: General Design Considerations (2)

- When creating a box, use a single rectangle object instead of joining four individual lines.
- When creating a border for a field or subform object, use the object's border attributes instead of creating a separate box.
- When creating backgrounds for objects, use a background fill instead of creating a separate shaded box object.
- For captions, use an object's caption property instead of using a separate static text object.
- Combine multiple static text objects into a single text object. This strategy is particularly useful after importing forms into LiveCycle Designer.

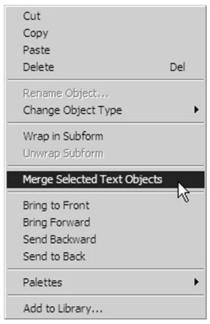


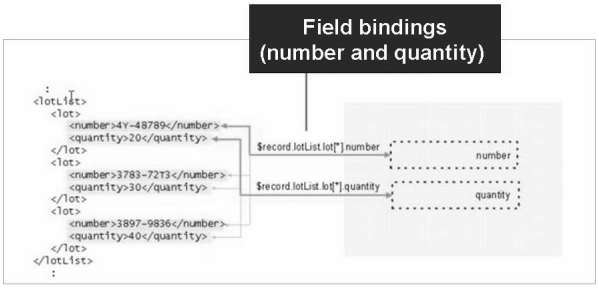
Figure 89: Tips for reducing the number of objects

- Keep the number of fonts, styles, and sizes to a minimum to take advantage of cached font information.
- Use fixed objects instead of dynamic objects that shrink and grow, if possible.
- Avoid building objects on top of each other, especially more than three layers deep.

Figure 90: Tips for reducing the complexity of objects

Data Binding

- Organize data elements into subforms and bind data structures to subforms



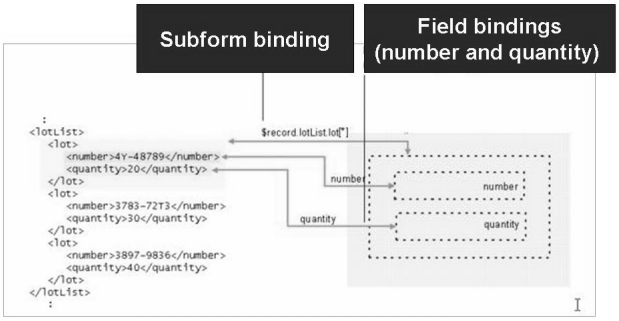


Figure 91: Data binding

- Repeating and nested subforms require additional processing to render the form. Avoid using them unless they offer better performance than alternative options. For example, a repeating subform may eliminate the need for a number of repeating objects.
- Allowing page breaks in subforms causes additional processing, even if Adobe document services do not apply page breaks. For example, the location, size, or content of a subform may prevent a page break. To optimize performance, turn off page breaks in subforms. By default, page breaks are allowed.

Figure 92: Subforms

- In general, scripts increase the processing required to render forms. Before using scripts, determine if a better performance alternative is available (typically in the ABAP logic)
- Avoid validation scripts if possible
- FormCalc executes simple calculations and validations more quickly.
- FormCalc interprets the XML Form Object Model syntax directly, which means it can evaluate reference syntax expressions more efficiently

Figure 93: Scripting

Lesson Summary

You should now be able to:

- Understand the main factors influencing the performance of a PDF form
- Understand and make use of the best practices for PDF forms design

Lesson: The PDFOBJECT API

Lesson Overview

This lesson explores the capabilities of the PDFOBJECT API and show examples on how to use it.

Lesson Objectives

After completing this lesson, you will be able to:

- Know where to locate and how to use the PDFOBJECT API
- Show examples using the API in the SAP system
- Show how to use the API with comments, attachments and digital signatures

Business Example

Your application requires a fully custom application with an Interactive Form with attachments, comments and digital signatures. This can be achieved by using the PDFOBJECT API.

PDFOBJECT ABAP Class

- Standard PDF object in ABAP system: CL_FP_PDF_OBJECT
- Methods:
  - Get/Set Attachments
  - Get/Set Comments
  - Get/Set Signature
  - Extract Data
  - …

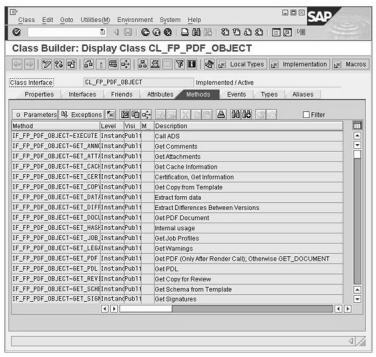


Figure 94: The PDJOBJECT ABAP class

Sample standard programs using the API

- Extract PDF data
  - Extracts filled in XML data structure

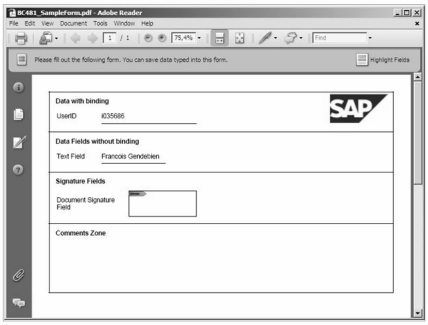


Figure 95: Program FP_PDF_TEST_03

- FP_PDF_TEST_14: Get attached comments of PDF file
- FP_PDF_TEST_15: Get attached files of PDF file

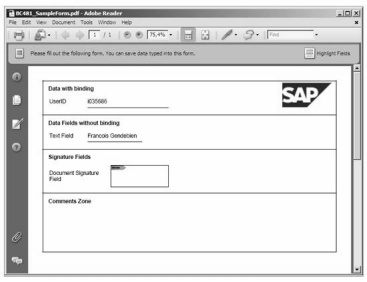


Figure 96: Programs FP_PDF_TEST_*

- SET_TEMPLATE(template, FILLABLE = ABAP_TRUE)
- SET_DATA(data)
- SET_USAGERIGHTS( )
- SET_TASK_RENDERPDF( )
- EXECUTE( )
- GET_PDF( )

Figure 97: Example: generate an Interactive Form

Lesson Summary

You should now be able to:

- Know where to locate and how to use the PDFOBJECT API
- Show examples using the API in the SAP system
- Show how to use the API with comments, attachments and digital signatures

Related Information

- Help.sap.com > Search on CL_FP_PDF_OBJECT
- SDN Adobe homepage: https://www.sdn.sap.com/irj/sdn/adobe
- SDN ABAP OO homepage: https://www.sdn.sap.com/irj/sdn/abap > ABAP OO

Lesson: Digital Signatures

Lesson Overview

This lesson covers aspects related to digital signatures embedded within SAP Interactive Forms by Adobe: what are they, what are the differences and possibilities, how can they be implemented in the PDF and how can the link with the SAP system be achieved.

Lesson Objectives

After completing this lesson, you will be able to:

- Understand the possibilities offered by digital signatures within SIFbA
- Understand and explain what can be done with SIFbA digital signatures
- Identify configuration required to use digital signatures in an SAP environment

Business Example

You're responsible for the setup of an application which needs to meet high security standards. The PDF technology has been chosen and you're asked to investigate the possibilities of integration of digital signatures for internal and external usage.

Digital Certifications and Signatures

You've received or downloaded a PDF form via email, which is supposed to be from one of your customers or suppliers. How can you be sure it's really from a trusted party ? The best way is to check whether the digital signatures present on the document are authentic. A PDF document can have two type of digital signatures:

- A certification signature, which can be applied by the document's author. Adobe Reader or Acrobat automatically checks the authenticity of this signature when you open the document, and then displays a window that indicates whether the signature is valid (that is, authentic and current). This guide also refers to the certification signature as the "author's digital signature."
- A standard signature, which can be applied by anyone who has permission to digitally sign the document. Adobe Reader or Acrobat can automatically check the authenticity of standard signatures when you open the document, or you can check them manually from within the application.

After opening the PDF document, the Reader automatically checks for unauthorized modifications to the document and checks the authenticity of the certification and signature.

There are three types of statuses for verification of certification signatures:

- Certification Valid, with a blue ribbon
- Validation Of Author Not Confirmed, with a blue question mark next to a person
- Certification Invalid, with a red X

On top of that, a PDF document can contain one or more standard digital signature. For each signature, the Adobe Reader also displays when the document was signed and whether it was modified after any of the signers had last signed the document.

The time and date of a digital signature, called a time stamp, can be important when you are working with time-sensitive documents (such as contracts, real estate offers, loan applications, and payments). For example, the time stamp can indicate when offers or counteroffers were made, or whether a document was signed before a deadline. In Adobe Reader or Acrobat, you can view the time stamps of signatures by opening the Signatures tab of the navigation pane, and then clicking the plus signs to expand the information for a particular signature.

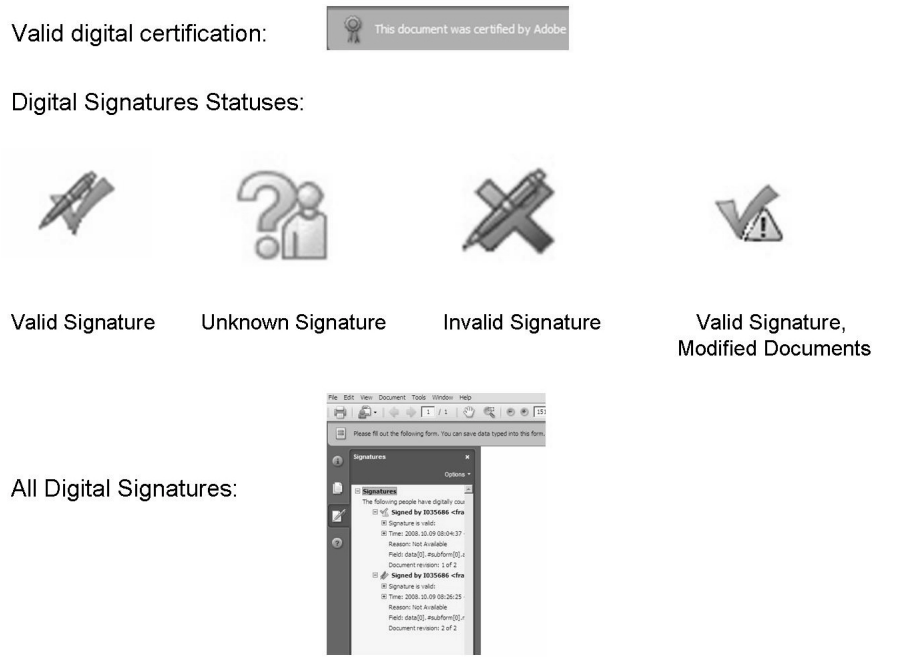


Figure 98: Certifications and Signatures Statuses

Any time that Adobe Reader or Acrobat reports that a digital signature has a status of Validity Of Author Not Confirmed or Signature Validity Is Unknown, you must decide whether to establish trust for that signature. This task involves three basic steps:

1. Obtain a certificate for the digital signature from a known, trusted individual or website. If you are at work, request this certificate from your company's IT department. A certificate is an electronic counterpart to driver licenses, passports, membership cards, etc. Certificates are electronic files containing information about an individual or organization that is used to establish their digital identity.
2. Add the certificate in Adobe Reader or Acrobat, and then set the trust level for the certificate in the application.
3. Revalidate the signature.

Signing PDF documents client side

- **Makes use of the PKI infrastructure in place**
- **Use existing digital ID from**
  - File
  - Roaming Digital ID
  - Connected Device
- **Create a new digital ID**

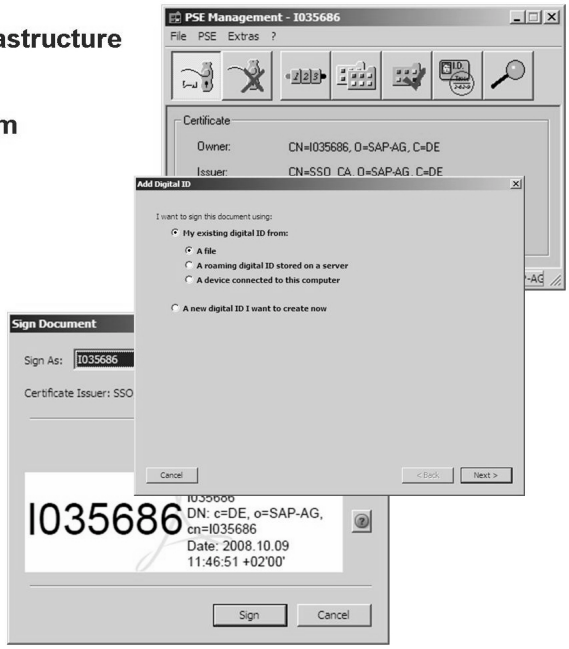


Figure 99: Client Side Signature

Server Side: Configuration in SAP systems

- **Server side**
  - SSL to J2EE is required
  - Certificates (private keys and public keys) installed
  - Certificate revocation list defined for ADS

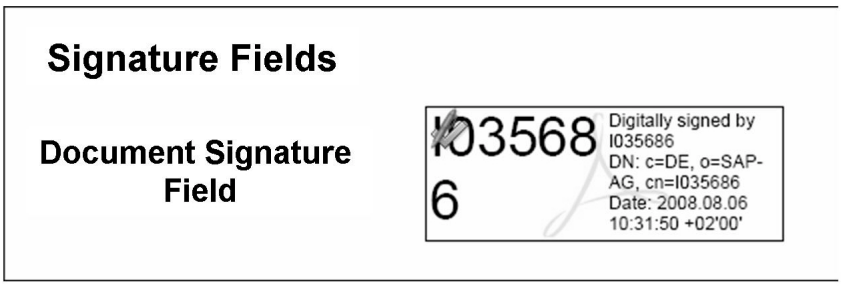


Figure 100: Digital Signatures in SAP systems

- SET_DOCUMENT(document)
- SET_TASK_EXTRACTDATA( )
- SET_TASK_GETSIGNATURES( )
- EXECUTE( )
- GET_DATA( )
- GET_SIGNATURES( )

Figure 101: Example: extract document data with PDFObject API

Lesson Summary

You should now be able to:

- Understand the possibilities offered by digital signatures within SIFbA
- Understand and explain what can be done with SIFbA digital signatures
- Identify configuration required to use digital signatures in an SAP environment

Unit Summary

You should now be able to:

- Understand concepts and possibilities in offline scenarios which can be used with SAP Interactive Forms by Adobe
- Understand considerations to have in mind when assessing an offline scenario
- Understand and make use of Interactive Form with email
- Develop this functionality on an existing Interactive Form
- Understand and handle complex layouts and frequently met form elements
- Estimate the complexity of an Interactive form
- Understand the main factors influencing the performance of a PDF form
- Understand and make use of the best practices for PDF forms design
- Know where to locate and how to use the PDFOBJECT API
- Show examples using the API in the SAP system
- Show how to use the API with comments, attachments and digital signatures
- Understand the possibilities offered by digital signatures within SIFbA
- Understand and explain what can be done with SIFbA digital signatures
- Identify configuration required to use digital signatures in an SAP environment

Test Your Knowledge

1. What are the four statuses of digital signature validation in the Adobe Reader ?

Answers

1. What are the four statuses of digital signature validation in the Adobe Reader ?

   Answer: Valid Signature, Unknown Signature, Invalid Signature and Valid signature, modified documents.

Internal Use SAP Partner Only

Internal Use SAP Partner Only

Internal Use SAP Partner Only

Course Summary

You should now be able to:

- Understand the SAP Interactive Forms by Adobe capabilities and use cases in SAP environments.
- Understand and develop SAP Interactive Forms by Adobe within an ABAP report
- Understand and develop SAP Interactive Forms by Adobe with Web Dynpro ABAP
- Understand and use the advanced features of the Adobe LiveCycle Designer

Feedback

SAP AG has made every effort in the preparation of this course to ensure the accuracy and completeness of the materials. If you have any corrections or suggestions for improvement, please record them in the appropriate place in the course evaluation.

Internal Use SAP Partner Only