Creating a Form-Based Process (Adobe)
## Typographic Conventions

<table>
<thead>
<tr>
<th>Type Style</th>
<th>Represents</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example Text</strong></td>
<td>Words or characters quoted from the screen. These include field names, screen titles, pushbuttons labels, menu names, menu paths, and menu options. Cross-references to other documentation.</td>
</tr>
<tr>
<td><strong>Example text</strong></td>
<td>Emphasized words or phrases in body text, graphic titles, and table titles.</td>
</tr>
<tr>
<td><strong>EXAMPLE TEXT</strong></td>
<td>Technical names of system objects. These include report names, program names, transaction codes, table names, and key concepts of a programming language when they are surrounded by body text, for example, SELECT and INCLUDE.</td>
</tr>
<tr>
<td><strong>Example text</strong></td>
<td>Output on the screen. This includes file and directory names and their paths, messages, names of variables and parameters, source text, and names of installation, upgrade and database tools.</td>
</tr>
<tr>
<td><strong>Example text</strong></td>
<td>Exact user entry. These are words or characters that you enter in the system exactly as they appear in the documentation.</td>
</tr>
<tr>
<td><strong>&lt;Example text&gt;</strong></td>
<td>Variable user entry. Angle brackets indicate that you replace these words and characters with appropriate entries to make entries in the system.</td>
</tr>
<tr>
<td><strong>EXAMPLE TEXT</strong></td>
<td>Keys on the keyboard, for example, F2 or ENTER.</td>
</tr>
</tbody>
</table>

## Icons

<table>
<thead>
<tr>
<th>Icon</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>!</td>
<td>Caution</td>
</tr>
<tr>
<td>📕</td>
<td>Example</td>
</tr>
<tr>
<td>📝</td>
<td>Note</td>
</tr>
<tr>
<td>🧑‍💻</td>
<td>Recommendation</td>
</tr>
<tr>
<td>🕹️</td>
<td>Syntax</td>
</tr>
</tbody>
</table>
# Contents

Scenario........................................................................................................................................... 2

About This Document ......................................................................................................................... 2
  General Prerequisites....................................................................................................................... 2
  Applicable Releases......................................................................................................................... 2
  Disclaimer....................................................................................................................................... 2

The Step By Step Solution .................................................................................................................. 3
  How to Check the Configuration of the J2EE Engine ................................................................. 3
    Prerequisites ................................................................................................................................. 3
    Description of the Scenario ......................................................................................................... 3
  How to Create Adobe Interactive Forms with Adobe Designer ................................................. 5
    Prerequisites ................................................................................................................................. 5
    Description of the Scenario ......................................................................................................... 5
  How to Use a Personalized Offline Interactive Form via E-mail .................................................. 10
    Description of the Scenario ........................................................................................................ 11
  How to Create and Use Callable Objects to Fill the Adobe Forms .......................................... 21
    Description of the Scenario ........................................................................................................ 21
  How to Create and Use Callable Objects to Validate the Adobe Forms .................................... 31
    Description of the Scenario ........................................................................................................ 31

Examples for a GP-Enabled Adobe Document .................................................................................. 39
Scenario

An Adobe form can be used to gather information as part of a Business Process. Interactive forms can be incorporated into a Guided Procedure or you can use them as standalone objects.

You are allowed to send personalized and impersonalized interactive forms, as well as online and offline interactive forms.

For example, you can:

- create an offline form from a form template and publish it to an URL
- send a form to an e-mail
- trigger a GP process using an interactive form, and so on

About This Document

General Prerequisites

To implement an Interactive Form based process using Adobe interactive forms you should perform the configuration steps described in the document “Error! Reference source not found.”.

To use Adobe interactive forms you should install the newest version of Adobe Reader using www.adobe.com

In case of Problems concerning ADOBE Reader please refer to SAP Note 934981

Applicable Releases

This tutorial is compatible with the following release” Free from SAP NetWeaver ’04s SPS07”.

Disclaimer

Any software coding and/or code lines / strings (“Code”) included in this documentation are only examples and are not intended to be used in a productive system environment. The Code is only intended better explain and visualize the syntax and phrasing rules of certain coding. SAP does not warrant the correctness and completeness of the Code given herein, and SAP shall not be liable for errors or damages caused by the usage of the Code, except if such damages were caused by SAP intentionally or grossly negligent.
The Step By Step Solution

How to Check the Configuration of the J2EE Engine

Prerequisites

For a detailed discussion about the configuration of your J2EE engine see also [1].

Description of the Scenario

1. Open the Visual Administrator of your J2EE engine.
   (Start -> Programs -> SAP J2EE Engine -> Visual Administrator)
   Select your connection configuration and click the Connect button.

2. The Login dialog appears. Type in your password and then click Connect.
3. The main window of the Visual Administrator opens. On the left-hand side of the window select the Cluster tab if needed and then expand the server’s node.

4. Expand the Services node.

5. Select the caf/eu/gp/model node.
6. Check the entries on the right-hand side of the window. If you adjust any of the entries, restart the engine when you have changed the configuration values.

### How to Create Adobe Interactive Forms with Adobe Designer

#### Prerequisites

You have already installed the SAP NetWeaver Developer Studio 7.0.0. The SAP NetWeaver Developer Studio 7.0.0 includes the Adobe Designer as well, so you should not install it separately.

#### Description of the Scenario

This section provides a high level description of how Adobe Interactive Forms can be created using either the Adobe Designer or the Adobe Plug-In for the NetWeaver IDE.

You do not have to prepare a new template here as the template used in the following chapters is attached in the appendix of this document.

1. Start the Adobe Designer application as a standalone application or as part of the SAP NetWeaver Developer Studio.
If you want to start the Adobe Designer from within the SAP NetWeaver Developer Studio, you should:

a. Add an InteractiveForm element to one of your views
b. Select the InteractiveForm element
c. Right-click it and choose the Edit menu item from its context menu

2. When you have started the Adobe Designer you will find the Library panel on the right-hand side of its window. The Library panel contains the list of the available GUI elements. You can select the appropriate element and drag and drop it into the body page.

Master pages are currently not supported by Guided Procedures; you should use only the body pages.
If you are working from within the SAP NetWeaver Developer Studio the Data View tab contains the context elements of your Web Dynpro view and you can drag and drop them into the body page of Adobe form as well. In this case the binding between the context element and the Adobe GUI element will be created automatically too.

Please ensure that every single data element has a unique binding name! If, for example, employee and approver data is included, please choose a different binding name for the name of the employee (for example eName) from the name of the approver (for example aName). Otherwise you may get an error when sending the data.

3. A script editor panel can be found on the upper side of the Adobe Designer window. Here you can attach scripts to the GUI elements. The GUI elements located on the WebDynpro tab already contain some predefined scripts, for example:

```javascript
-- form1.#subform[0].submitToSAP1::click - (JavaScript, client) -
// DO NOT MODIFY THE CODE BEYOND THIS POINT - SubmitToSAP.xfo
app.eval("event.target.SAPSubmit();");
// END OF DO NOT MODIFY
```

You can use JavaScript and FormCalc script languages
A Web Dynpro page is also available in the library; it contains several SAP specific elements.

If you have no WebDynpro tab page, you can create it by right-clicking the tabs, then choosing Add Group and specifying its name in the Add Library Group dialog.

When you have added the new tab, you have to select it, right click on it, and choose the Group Properties menu item from its context menu.

The following dialog appears:

Correct the ending of the string typed into the Location field as above.
4. When you have created the form you can use the PDF Preview tab to check its layout.

5. Save the file you have created as an Adobe XML Form File (XDP file), if you are working with a standalone Adobe Designer application. Later you can use this file in your GP process when you create an Interactive Form callable object. For example, see step 7 in section 0.

How to Use a Personalized Offline Interactive Form via E-mail

In this scenario we modify the Time-Off Process example from the GP installation by replacing its 
*Book Request* action with an Adobe interactive form; that is, the HR consultant receives an Adobe Interactive Form containing the approved time-off request data by e-mail.
Description of the Scenario

1. Create a new process named *Vacation Request Process*.

2. Set its built-in roles to *Initiator*.

3. Create a new sequential block named *Vacation Request*. This block is a modified version of the *Time-Off Process* block located in the folder *Examples->Time-Off Process->Blocks* of the GP.
4. Add the actions Create Request, Set 1st Approver, Set Second Approver, Set HR Consultant, 1st Approval, 2nd Approval and Summary to the Vacation Request block; that is, all actions from the Time-Off Process except the Book Request action.

![Process: Vacation Request Process](image)

5. Create a new action named Adobe Book Request and move it up to between the 2nd Approval and the Summary actions.

![Process: Vacation Request Process](image)

6. Create a new Interactive Form type callable object under the Adobe Book Request action.

   Enter the following basic data:
   - **Name**, for example Adobe Interactive Form CO
   - **Description**
   - **Language**
   - Choose a location for your callable object (Folder)

   Choose [Next ▶]
7. Set the XDP document located in section 0 of the "Error! Reference source not found." into the Define Object part of the callable object. Click Next.

8. Skip the Define Input and Define Output screens, and then set the configuration as shown.
9. Consolidate the parameters as shown.

<table>
<thead>
<tr>
<th>Context Parameter</th>
<th>Object Type</th>
<th>Parameter Defined For</th>
<th>List</th>
</tr>
</thead>
<tbody>
<tr>
<td>HR Consultant Data</td>
<td>Structure</td>
<td>Summary</td>
<td></td>
</tr>
<tr>
<td>HR Consultant Data Mapped</td>
<td>Structure</td>
<td>&lt;Group&gt;</td>
<td></td>
</tr>
<tr>
<td>HR Consultant Data</td>
<td>Structure</td>
<td>Create Request</td>
<td></td>
</tr>
<tr>
<td>HR Consultant Data</td>
<td>Structure</td>
<td>1st Approval</td>
<td></td>
</tr>
<tr>
<td>HR Consultant Data</td>
<td>Structure</td>
<td>2nd Approval</td>
<td></td>
</tr>
<tr>
<td>Employee Data Mapped</td>
<td>Structure</td>
<td>&lt;Group&gt;</td>
<td></td>
</tr>
<tr>
<td>Employee Data</td>
<td>Structure</td>
<td>Create Request</td>
<td></td>
</tr>
<tr>
<td>Employee Data</td>
<td>Structure</td>
<td>1st Approval</td>
<td></td>
</tr>
<tr>
<td>Employee Data</td>
<td>Structure</td>
<td>2nd Approval</td>
<td></td>
</tr>
<tr>
<td>Employee Data</td>
<td>Structure</td>
<td>Summary</td>
<td></td>
</tr>
<tr>
<td>Approver Data Mapped</td>
<td>Structure</td>
<td>&lt;Group&gt;</td>
<td></td>
</tr>
<tr>
<td>Approver Data</td>
<td>Structure</td>
<td>Create Request</td>
<td></td>
</tr>
<tr>
<td>Approver Data</td>
<td>Structure</td>
<td>1st Approval</td>
<td></td>
</tr>
<tr>
<td>Approver Data</td>
<td>Structure</td>
<td>2nd Approval</td>
<td></td>
</tr>
<tr>
<td>Approver Data</td>
<td>Structure</td>
<td>Summary</td>
<td></td>
</tr>
<tr>
<td>Second Approver Mapped</td>
<td>Structure</td>
<td>&lt;Group&gt;</td>
<td></td>
</tr>
<tr>
<td>Processor of Second Approval</td>
<td>Structure</td>
<td>Create Request</td>
<td></td>
</tr>
<tr>
<td>User List</td>
<td>Structure</td>
<td>Set Second Approver</td>
<td></td>
</tr>
<tr>
<td>Processor of First Approval Mapped</td>
<td>Structure</td>
<td>&lt;Group&gt;</td>
<td></td>
</tr>
<tr>
<td>Processor of First Approval</td>
<td>Structure</td>
<td>Create Request</td>
<td></td>
</tr>
<tr>
<td>User List</td>
<td>Structure</td>
<td>Set 1st Approver</td>
<td></td>
</tr>
<tr>
<td>HR Consultant Mapped</td>
<td>Structure</td>
<td>&lt;Group&gt;</td>
<td></td>
</tr>
<tr>
<td>Processor for HR consultant</td>
<td>Structure</td>
<td>Create Request</td>
<td></td>
</tr>
<tr>
<td>User List</td>
<td>Structure</td>
<td>Set HR Consultant</td>
<td></td>
</tr>
<tr>
<td>Time-Off Data Mapped</td>
<td>Structure</td>
<td>&lt;Group&gt;</td>
<td></td>
</tr>
<tr>
<td>Time-Off Data</td>
<td>Structure</td>
<td>Create Request</td>
<td></td>
</tr>
<tr>
<td>Time-Off Data</td>
<td>Structure</td>
<td>1st Approval</td>
<td></td>
</tr>
<tr>
<td>Time-Off Data</td>
<td>Structure</td>
<td>2nd Approval</td>
<td></td>
</tr>
<tr>
<td>Time-Off Data</td>
<td>Structure</td>
<td>Summary</td>
<td></td>
</tr>
<tr>
<td>employee</td>
<td>Structure</td>
<td>Adobe Booking Request</td>
<td></td>
</tr>
</tbody>
</table>
Do not expose any input parameters at process level:
10. Consolidate the roles as shown.

11. (Optional) Assign a default role to the Employee, SecondApprover, FirstApprover, HRConsultant roles.
12. Make sure that the **HRConsultant** user has a valid e-mail address.

13. Once you have saved and activated your process, instantiate it via its URL.

14. Complete the required fields in the **Vacation Request Process**.

For example:

* **Absence Type**: Vacation
  Paid
* **Start Date**: 01/18/2006
  **End Date**: 01/19/2006
15. Complete the *Create Request, 1st Approval* and *2nd Approval* actions. After that you will receive the message shown.

16. After a while an e-mail will be sent by the Guided Procedure. Open the mail and its attachment.
17. Fill out the form and then choose the *Submit* button.

18. You will get a confirmation.
19. Start a new browser instance, choose the GP runtime URL and login as the user who is assigned to the data display step. Select the running Vacation Request Process work item. Choose Complete.
How to Create and Use Callable Objects to Fill the Adobe Forms

In the following scenario we implement a *Background Execution* callable object to fill the interactive forms. Note that any other background callable object, for example, an *External Services* callable object, could be used as well.

**Description of the Scenario**

1. Create a new Java development component for your background object, which has to perform the pre-filling of the interactive forms.
2. Add the following used DCs with Build-Time dependency
   - sap.com_SAP-EU_1 - caf/eu/gp/api (external)
   - sap.com_SAP-JEE_1 - com.sap.exception
   - sap.com_SAP-JEE_1 - com.sap.security.api.sda
   - sap.com_SAP-JEE_1 - tc/logging

3. Create a new package and then a new class in your DC. Remember that your class must implement the IGPBackgroundCallableObject interface. See Error! Reference source not found..

4. Create another class in this package called “PrefillRessourceAccessor”. As super class of this class the “GPStandardResourceAccessor” has to be chosen.

5. Add a simple file to the “Prefill.properties” package.
6. Please add the following text to the new “Prefill.properties” file:

CO_NAME=Prefill Background
CO_DESCRIPTION=Background Callable Object for prefilling booking form

7. Implement the getDescription method as follows:

```java
public IGPTechnicalDescription getDescription(Locale originalLocale) {
    try {
        PrefillResourceAccessor resourceAccessor = new PrefillResourceAccessor(
            "com.sap.tst.co.Prefill.properties";

        //create technical description instance
        IGPTechnicalDescription technicalDescription =
            GPCallableObjectFactory.createTechnicalDescription(
                "CO_NAME",
                "CO_DESCRIPTION",
                resourceAccessor,
                originalLocale);

        //get root structure for output parameters
        IGPStructureInfo output =
            technicalDescription.getOutputStructureInfo();

        //define attributes for output parameters
        output.addAttribute("beginDate",
            IGPAtributeInfo.BASE_DATE);
        output.addAttribute("endDate", IGPAtributeInfo.BASE_DATE);
    }
    return technicalDescription;
}
```
//add a result state for successful execution
IGPCOResultStateInfo success =
    technicalDescription.addResultState("Success");
//add result state description
success.setDescriptionKey("Success_desc");
return technicalDescription;
}
}
catch (GPInvocationException e) {
    com.sap.tc.logging.Location logger =
        com.sap.tc.logging.Location.getLocation(PrefillCOClass);
    logger.logT(
        Severity.ERROR,
        Category.APPLICATIONS,
        "Incorrect technical name");
    logger.traceThrowableT(
        Severity.ERROR,
        "Exception while creating technical description: ",
        e);
    return null;
}

8. Implement the execute method as follows:

    public void execute(IGPExecutionContext executionContext)
        throws GPTechnicalCallableObjectException {
    com.sap.tc.logging.Location logger =
        com.sap.tc.logging.Location.getLocation(PrefillCOClass);
        try {
            //retrieve the runtime representation of the output structure
            IGPStructure output = executionContext.getOutputStructure();

            //set values to output parameters
            output.setAttributeValue("beginDate", new java.util.Date(System.currentTimeMillis()));
            output.setAttributeValue("endDate", new java.util.Date(System.currentTimeMillis()));

            //set result state
executionContext.setResultState("Success");
//complete object execution
executionContext.processingComplete();
}

} catch (GPInvocationException e) {
    PrefillResourceAccessor resourceAccessor =
        new PrefillResourceAccessor("com.sap.tst.co.Prefill.properties");
    throw new GPTechnicalCallableObjectException(
        logger, resourceAccessor, "ERROR_PARAMETERS", e);
}

} catch (GPEngineException e) {
    PrefillResourceAccessor resourceAccessor =
        new PrefillResourceAccessor("com.sap.tst.co.Prefill.properties");
    throw new GPTechnicalCallableObjectException(
        logger, resourceAccessor, "INTERNAL_ERROR", e);
}

9. Define a public part for your DC. Check “Can be packaged into other build results (e.g. SDAs)”
10. Create a new Library development component for your background object which has to encapsulate your Java development component.

11. Add the (sap.com_SAP-EU_1) caf/eu/gp/api (external) and the (sap.com_SAP-JEE_1) com.sap.security.api.sda development components and your Java DC created in the steps 1-9 to the Used DCs of your Library development component.

⚠️ You must also specify deploy time and runtime dependencies for the caf/eu/gp/api and the com.sap.security.api.sda development components. After you have added the used DCs, please check that these flags are really set!
12. Build and deploy your Library DC.

13. Login in to Guided Procedures Design Time and create a Background Execution callable object. In the Define Object dialog specify your Library DC deployed in step 12 and the class name of your implementation class.

14. Test, save and activate your callable object.
15. Open your GP process using interactive forms.

16. Open your interactive form CO. Switch to edit mode and navigate to the Input tab.
   Click the Define Prefilling link.

17. Choose Select a prefilling service.
18. Select the line of your callable object and then click `Select`.

19. As we have no input parameters in our callable object we continue with `Done`.
20. Define the mappings as follows:

Highlight the corresponding line of the left table. Then select the row to be mapped in the right table. This will create a new mapping.

Then choose **Done**.

21. Click the **Save** button.

The **Prefilling service has been saved** message appears.

22. Save all changes and activate your GP process again.
23. After instantiating your *Vacation Request Process*, the *Start Date* and the *End Date* specified by the Initiator will be overwritten to the current date by the prefilling callable object. Note that functionality makes no sense in this particular case, it was implemented only as an example; but if you use the same Adobe form to start the *Vacation Request Process*, then it could make sense to use such a callable object.

**How to Create and Use Callable Objects to Validate the Adobe Forms**

In the following scenario we implement an *External Service* callable object to validate the interactive forms. Note that any other background callable object, for example, a *Background Execution* callable object, could be used as well.

**Description of the Scenario**

1. Logon to your SAP external system.
2. Open the SE80 transaction.

3. Create a new ZADOBE_TEST function group and a new ZVALIDATE_ADOBE_FORM function in it.
   The following screenshots demonstrate the input parameters, exceptions and source code of our example function.

    FUNCTION ZVALIDATE_ADOBE_FORM.
      """Local Interface:"""
      """ IMPORTING
      """  VALUE(IN_BEGIN_DATE) TYPE D
      """  VALUE(IN_END_DATE) TYPE D
      """ EXCEPTIONS
      """  FAILED
      """
      if in_begin_date < sy-datum or in_end_date < sy-datum.
          raise FAILED.
      endif.
    ENDFUNCTION.
4. In the attributes tab check the “Remote-Enabled Module” flag!

5. Save and activate your function.

6. Login to Guided Procedures Design Time and create an **External Service** callable object.
7. In the Define Object dialog specify the endpoint alias of your external server and then the name of your validation service.

8. Leave the input parameters unchanged.

9. Choose Next. (Our service has no output parameters.)

10. Specify the error handling mode, and then click Next.
11. Choose **Finish and Open**.

12. Test, save and activate your callable object.
13. Open your GP process using interactive forms.

14. Open your CO using the interactive form. Switch to edit mode, navigate to the Output tab and click the Define Validation link.

15. Choose Select a validation service.
16. Select the row of your callable object and then click the **Select** button.

![Callable object: AddInfoTaskF (Callable object)](image)

17. Define the mappings as follows:

<table>
<thead>
<tr>
<th>Parameter of Service AddInfoTaskF</th>
<th>Mapped To</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter</td>
<td>AddInfo</td>
</tr>
<tr>
<td>Parameter</td>
<td>Add</td>
</tr>
</tbody>
</table>

Then choose **Done**.

18. For error handling, select an email template then click **Done**.

For more information about e-mail templates see **Error! Reference source not found.**
19. Choose **Save**. The validation service has been saved; the message appears.

20. (Optional) If you have set a prefilling service in section 4.4, then remove its mappings.

21. Save all changes and activate your GP process again.
Examples for a GP-Enabled Adobe Document

The following screenshot shows an example XDP file containing a request form for the Time-Off process.

![Figure 1: travelRequest.xdp in the Adobe Designer](image)

The corresponding code example can be saved into an XDP file and then you can use it in your GP process.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<?xfa generator="AdobeDesigner_V6.0_SAP" APIVersion="1.4.4183.0"?>
```
Book Request Form

First Name
<field name="firstname" y="6.35mm" x="31.75mm" w="31.75mm" h="13.7068mm" access="readOnly">
  <ui>
    <TextEdit>
      <border>
        <?templateDesigner StyleID aped3?>
        <edge stroke="lowered"/>
        <edge stroke="lowered"/>
        <edge stroke="lowered"/>
        <edge stroke="lowered"/>
      </border>
      <margin/>
    </TextEdit>
  </ui>
  <font typeface="Arial"/>
  <caption reserve="0.185306in" placement="top">
    <font typeface="Arial"/>
    <para vAlign="middle"/>
    <value>
      <text>First Name</text>
    </value>
  </caption>
</field>

<field name="lastname" y="6.35mm" x="31.75mm" w="31.75mm" h="13.7068mm" access="readOnly">
  <ui>
    <TextEdit>
      <border>
        <?templateDesigner StyleID aped3?>
        <edge stroke="lowered"/>
        <edge stroke="lowered"/>
        <edge stroke="lowered"/>
        <edge stroke="lowered"/>
      </border>
      <margin/>
    </TextEdit>
  </ui>
  <font typeface="Arial"/>
  <caption reserve="0.185306in" placement="top">
    <font typeface="Arial"/>
    <para vAlign="middle"/>
    <value>
      <text>Last Name</text>
    </value>
  </caption>
</field>

<field name="userid" y="6.35mm" x="63.5mm" w="25.4mm" h="13.7068mm">
  <para vAlign="middle"/>
  <caption reserve="0.185306in" placement="top">
    <font typeface="Arial"/>
    <para vAlign="middle"/>
    <value>
      <text>User ID</text>
    </value>
  </caption>
</field>
<para vAlign="middle"/>
<caption placement="right" reserve="22mm">
  <para vAlign="middle"/>
  <font typeface="Arial"/>
  <value>
    <text>Unpaid</text>
  </value>
</caption>
<items>
  <text>false</text>
</items>
<value>
  <text xmlns:xsi="http://www.w3.org/2000/10/XMLSchema-instance" xsi:nil="true"/>
</value>
</field>
<?templateDesigner expand 1?></exclGroup>

<field h="9.525mm" name="BEGDA" w="114.3mm" x="3.175mm" y="73.025mm">
  <ui>
    <dateTimeEdit>
      <border>
        <edge stroke="lowered"/>
        <edge stroke="lowered"/>
        <edge stroke="lowered"/>
        <edge stroke="lowered"/>
      </border>
      <margin/>
    </dateTimeEdit>
  </ui>
  <font typeface="Arial"/>
  <margin bottomInset="1mm" leftInset="1mm" rightInset="1mm" topInset="1mm"/>
  <para vAlign="middle"/>
  <caption reserve="43.13mm">
    <font typeface="Arial"/>
    <para vAlign="middle"/>
<value>
    <text>Start Date</text>
</value>
</caption>
<value>
    <date/>
</value>
</field>

<field h="9.525mm" name="ENDDA" w="114.3mm" x="3.175mm" y="82.55mm">
    <ui>
        <dateTimeEdit>
            <border>
                <edge stroke="lowered"/>
                <edge stroke="lowered"/>
                <edge stroke="lowered"/>
                <edge stroke="lowered"/>
            </border>
            <margin/>
        </dateTimeEdit>
    </ui>
    <font typeface="Arial"/>
    <margin bottomInset="1mm" leftInset="1mm" rightInset="1mm" topInset="1mm"/>
    <para vAlign="middle"/>
    <caption reserve="43.13mm">
        <font typeface="Arial"/>
        <para vAlign="middle"/>
        <value>
            <text>End date</text>
        </value>
    </caption>
</field>

<field h="9.525mm" name="complete" w="44.45mm" x="3.175mm"
Complete

**THE BEST-RUN BUSINESSES RUN SAP**

```xml
<ui>
  <button/>
</ui>

<font typeface="Arial"/>
<caption>
  <value>
    <text>Complete</text>
  </value>
  <para hAlign="center" vAlign="middle"/>
  <font typeface="Arial"/>
</caption>

<border hand="right">
  <edge stroke="raised"/>
  <fill>
    <color value="212,208,200"/>
  </fill>
</border>

<bind match="none"/>
<event activity="click">
  <submit format="xml" target="%com.sap.caf.gp.if.PostProcessorUrl%" textEncoding="UTF-8"/>
</event>
</field>
</subform>

<?templateDesigner DefaultLanguage FormCalc?>
<?templateDesigner DefaultRunAt client?>
<?templateDesigner Grid show:1, snap:1, units:0, color:ff8080, origin:(0,0), interval:(125000,125000)?

<?templateDesigner Rulers horizontal:1, vertical:1, guidelines:1, crosshairs:0?>
<?templateDesigner Zoom 103?>
<?templateDesigner DefaultPreviewDynamic 0?>
<?templateDesigner XDCFile acrobat6.xdc?>

<config xmlns="http://www.xfa.org/schema/xci/1.0/"
```
<agent name="designer">
  <!-- [0..n] -->
  <destination>pdf</destination>
  <pdf>
    <!-- [0..n] -->
    <fontInfo/>
  </pdf>
</agent>

<present>
  <!-- [0..n] -->
  <destination>pdf</destination>
  <pdf>
    <!-- [0..n] -->
    <fontInfo>
      <embed>0</embed>
    </fontInfo>
    <tagged>1</tagged>
    <compression>
      <compressLogicalStructure>1</compressLogicalStructure>
    </compression>
    <interactive>1</interactive>
  </pdf>
</present>

<url>D:\usr\sap\C11\SYS\global\AdobeDocumentServices\lib\acrobat6.xdc</url>

<xdc>
  <coolType>
  </coolType>
  <unicodeDirectory>D:\usr\sap\C11\JC00\j2ee\os_libs\adssap\FontManagerService\fonts\adobe\typespt\unicode</unicodeDirectory>
  <commonFontsDirectory>D:\usr\sap\C11\JC00\j2ee\os_libs\adssap\FontManagerService\fonts\adobe\requrd\cmaps</commonFontsDirectory>
  <additionalFontsDirectory>D:\usr\sap\C11\JC00\j2ee\cluster\server0\.persistent\com.adobe.FontManagerService\fonts\customer;D:\usr\sap\C11\JC00\j2ee\os_libs\adssap\FontManagerService\fonts\adobe;D:\usr\sap\C11\JC00\j2ee\os_libs\adssap\FontManagerService\fonts\adobe\pdfl\6.0\fonts</additionalFontsDirectory>
<coolType>
</coolType>
<pcl>
<fontInfo>
<embed>1</embed>
</fontInfo>
</pcl>
<ps>
<fontInfo>
<embed>1</embed>
</fontInfo>
</ps>
<zpl>
<fontInfo>
<embed>0</embed>
</fontInfo>
</zpl>
<xdp>
<packets>*</packets>
</xdp>
</present>
<trace>
<area name="font" level="1"/>
</trace>
<acrobat>
<common>
<locale/>
<data>
<adjustData>1</adjustData>
<attributes>preserve</attributes>
<xsl>
<uri/>
<debug>
<uri/>
</debug>
</xsl>
</data>
</common>
</acrobat>
<outputXSL>
  <uri/>
</outputXSL>
</data>
<template>
  <relevant/>
  <base/>
  <uri/>
</template>
</common>
</acrobat>
</config>
<xfa:datasets xmlns:xfa="http://www.xfa.org/schema/xfa-data/1.0/">
  <xfa:data xfa:dataNode="dataGroup"/>
</xfa:datasets>
<localeSet xmlns="http://www.xfa.org/schema/xfa-locale-set/2.1/">
  <locale desc="English (United States)" name="en_US">
    <calendarSymbols name="gregorian">
      <monthNames>
        <month>January</month>
        <month>February</month>
        <month>March</month>
        <month>April</month>
        <month>May</month>
        <month>June</month>
        <month>July</month>
        <month>August</month>
        <month>September</month>
        <month>October</month>
        <month>November</month>
        <month>December</month>
      </monthNames>
      <monthNames abbr="1">
        <month>Jan</month>
        <month>Feb</month>
        <month>Mar</month>
      </monthNames>
    </calendarSymbols>
  </locale>
</localeSet>
<month>Apr</month>
<month>May</month>
<month>Jun</month>
<month>Jul</month>
<month>Aug</month>
<month>Sep</month>
<month>Oct</month>
<month>Nov</month>
<month>Dec</month>
</monthNames>
<dayNames>
  <day>Sunday</day>
  <day>Monday</day>
  <day>Tuesday</day>
  <day>Wednesday</day>
  <day>Thursday</day>
  <day>Friday</day>
  <day>Saturday</day>
</dayNames>
<dayNames abbr="1">
  <day>Sun</day>
  <day>Mon</day>
  <day>Tue</day>
  <day>Wed</day>
  <day>Thu</day>
  <day>Fri</day>
  <day>Sat</day>
</dayNames>
<meridiemNames>
  <meridiem>AM</meridiem>
  <meridiem>PM</meridiem>
</meridiemNames>
<eraNames>
  <era>BC</era>
  <era>AD</era>
</eraNames>
<calendarSymbols/>
<datePatterns>
  <datePattern name="full">EEEE, MMMM D, YYYY</datePattern>
  <datePattern name="long">MMMM D, YYYY</datePattern>
  <datePattern name="med">MMM D, YYYY</datePattern>
  <datePattern name="short">M/D/YY</datePattern>
</datePatterns>
<timePatterns>
  <timePattern name="full">h:MM:SS A Z</timePattern>
  <timePattern name="long">h:MM:SS A Z</timePattern>
  <timePattern name="med">h:MM:SS A</timePattern>
  <timePattern name="short">h:MM A</timePattern>
</timePatterns>
<dateTimeSymbols>GyMdkHmsSEDFwWahKzZ</dateTimeSymbols>
<numberPatterns>
  <numberPattern name="numeric">z,zz9.zzz</numberPattern>
  <numberPattern name="currency">$z,zz9.99|($z,zz9.99)</numberPattern>
  <numberPattern name="percent">z,zz9%</numberPattern>
</numberPatterns>
<numberSymbols>
  <numberSymbol name="decimal">.</numberSymbol>
  <numberSymbol name="grouping">,</numberSymbol>
  <numberSymbol name="percent">%</numberSymbol>
  <numberSymbol name="minus">-</numberSymbol>
  <numberSymbol name="zero">0</numberSymbol>
</numberSymbols>
<currencySymbols>
  <currencySymbol name="symbol">$</currencySymbol>
  <currencySymbol name="isoname">USD</currencySymbol>
  <currencySymbol name="decimal">.</currencySymbol>
</currencySymbols>
</locale>
</localeSet>
<x:xmpmeta xmlns:x="adobe:ns:meta/" x:xmptk="3.1-701">
  <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
  </rdf:RDF>
</x:xmpmeta>
Figure 2: travelRequest.xdp Source Code