WebADI for R12 SubLedger Accounting

John Peters

JRPJR, Inc

john.peters@jrpjr.com

http://www.jrpjr.com

Session 8620

- This is Revision 2 of the presentation
- New features are being released for Web ADI a quick pace, this presentation is based on 12.1.3
- Please check the OAUG papers database for updates
- I will also have updates on my web site:
  http://jrpjr.com
  Click on Paper Archives on the right hand side, then
  Collaborate11- WebADI for R12 SubLedger Accounting

Agenda

- Introduction
- What is WebADI, Settings/Versions
- WebADI Processing of Data
- Custom SubLedger Accounting JE Interface
- Demo
- Steps To Create Custom WebADI Integrators
- Create the Integrator & Interface
- Create the Layout
- Save a Document as a Menu Function
- Recent Enhancements
- Additional Resources

Introduction

- John Peters, JRPJR, Inc
- Independent Consulting in the San Francisco Bay Area
- I have been working with Oracle E-Business Suite since 1992, early days of Release 10 character mode
- I have presented many technical papers at numerous OUAG, IOUG and OpenWorld conferences
- Founding member of the Northern California OAUG GEO
- Current Coordinator for the Workflow SIG

Introduction

- I have worked on multiple Journal Entry Interfaces into SubLedger Accounting for several R12 Clients
- The most recent one involved integrating various external accounting and transactional systems with R12.1.3 SLA
- The client was a large shipping services company that has gone through many mergers and acquisitions that did not fully integrate their information systems
- This was a new R12 implementation with a new COA and business processes all the way through Procure to Pay, Order to Cash and Fixed Assets

Introduction

- SubLedger Accounting has a public PL/SQL API for integrating in Journal Entries

  XLA_JOURNAL_ENTRIES_PUB_PKG

- What SubLedger Accounting does not have is a WebADI interface like GL has for interfacing in Journal Entries

- This presentation will go over the creation of a custom WebADI interface for SubLedger Accounting to support the integration of accounting transactions from the clients various external systems

What is WebADI

What is Web ADI?

- How many of you are now using Web ADI as opposed to Desktop ADI?
- Not the same old Desktop ADI (Client ADI)
- Nothing to pre-install on the PC, all downloaded from the web browser
- Works with current Microsoft Excel, Word, and Project (32 bit versions)
- Only ADI tool available in R12, but also works with 11i
- Only ADI tool available for Vista and Windows 7 OS's
- Good resource for supported versions and other info: ML Note: 417692.1 Installing, Configuring and Troubleshooting Web ADI

Web ADI requires security setting changes

- ML Note: 1077728.1 Using Microsoft Office 2007 and 2010 with Oracle E-Business Suite 11i and R12
- These settings essentially loosen the security of your PC
- You can also now digitally sign your Desktop Integrators ML Note: 1095155.1, which allows you to keep your Excel security settings high

IE Browser Settings

- Select 'Tools' -> 'Internet Options -> 'Security' (Tab) from the browser menu
- Select the appropriate zone i.e. 'Trusted Sites' -> 'Custom Level' (button)
- Under the 'Downloads' section ensure 'File download' is enabled
- Set 'Allow Status bar updates via script' to 'Enable‘
- I also identify the server as a Trusted Site

Excel 2010 Settings

- Click File -> Options
- Click 'Trust Center' link in the 'Excel Options' window
- Click 'Trust Center Settings...' button
- In the 'Trust Center' window click 'Macro Settings' link from the menu on the left
- Under 'Macro Settings' check 'Disable all macros with notification'
- Under 'Developer Macro Settings' check 'Trust Access to the VBA project object model'
- Click 'OK' button to close the 'Trust Center' window
- Click 'OK' button to return to the spreadsheet

Web Application Desktop Integrators

- It is a set of meta data in the DB that define an “Integrator”
- You can optionally specify “Contents” which are extracts of data to download into your spreadsheet template
- Various “Layouts” can be applied on top of one “Integrator” to create specific tools for different groups of users
- Ability to create the Web ADI “Document” from a user menu item and download it to the users desktop
- Web ADI Documents can be both
  - Data extracted from the E-Business Suite
  - Data interfaced into the E-Business Suite

E-Business Suite Uses of Web ADI

- Web ADI Integrators are now used by many modules in the E-Business Suite, it has become a tool that all development groups use.
- Over 240 Integrators predefined by Oracle in 12.1.3

APPLICATION_NAME	COUNT(*)
APAC Consulting Localizations	9
Advanced Benefits	4
Advanced Product Catalog	5
Application Report Generator	1
Assets	5
CRM Foundation	6
Custom Development	1
Enterprise Performance Foundation	6
Financial Consolidation Hub	11
Financial Intelligence	2
Financials Common Modules	1
General Ledger	5
Human Resources	83
Incentive Compensation	1
Information Technology Audit	1
Internal Controls Manager	9
Learning Management	2
Marketing	3

APPLICATION_NAME	COUNT(*)
Operations Intelligence	3
Oracle Price Protection	3
Payroll	32
Projects	7
Public Sector Financials International	6
Public Sector HR	5
Public Sector Payroll	3
Report Manager	2
Sales	2
Sales Offline	1
Scheduler	1
Site Management	2
Student System	9
Supply Chain Intelligence	5
Trade Management	4
Transfer Pricing	2
Web Applications Desktop Integrator	1
iSupplier Portal	1

WebADI Processing of Data

Pre-Web ADI Method of Interfacing Data to E-Business Suite

- Perform Some Steps to create data
- Create a delimited file
  (comma, tab, pipe, etc)
- Upload file to server
  ftp, scp, custom OA Framework Upload Page
- SQL*Loader Concurrent Program to load into table
  Handle SQL*Loader Parsing Errors
  File Handling, good, discards, bad
- Concurrent Program to validate, derive and I/F data into the E-Business Suite

Custom Web ADI Method of Interfacing Data

- Perform Some Steps to create data
- Cut and Paste data into Web ADI Template
- Upload from Excel
- Concurrent Program to validate, derive and I/F data into the E-Business Suite
- Web ADI elminates the steps associated with:
  - Flat file transport
  - Parsing/loading
  - File handling after loading

Overall Custom SLA JE Process

- Web ADI uploads Journal Entries from Excel to a custom staging table
- Custom concurrent program performs derivations and validations against the data in the staging table
- Then the Journal Entries are interfaced to the base Oracle E-Business Suite XLA tables using the PL/SQL API XLA_JOURNAL_ENTRIES_PUB_PKG

Custom SubLedger Accounting JE Interface

Business Requirement

- The client is a shipping services company, new R12.1.3 implementation, financial modules are being used, new COA
- The client has well over three dozen external systems that will continue to feed transactions and data into R12 in Legacy formats (COA, Customers, Suppliers, etc)
- Some data interfaces will be eventually automated with extracts, transfers and loads
- For now the JE interfaces need human intervention to “tweak” the data prior to load into R12

Solution Design

- Custom staging table to hold Legacy data
- For now data will be uploaded from Excel to the staging table using a Custom WebADI process, eventually we can replace a WebADI process with a more traditional File or DB Link I/F
- Concurrent program is run to validate and convert to R12 then interface into SubLedger Accounting using a PL/SQL API
- Error correction form on the custom staging table to allow users to fix errors and resubmit

SubLedger Accounting JE PL/SQL API Calls

- Create an SLA JE Header using:
  XLA_JOURNAL_ENTRIES_PUB_PKG.CREATE_JOURNAL_ENTRY_HEADER
- Create an SLA JE Line(s) using:
  XLA_JOURNAL_ENTRIES_PUB_PKG.CREATE_JOURNAL_ENTRY_LINE
- Complete the SLA JE Header using:
  XLA_JOURNAL_ENTRIES_PUB_PKG.COMPLETE_JOURNAL_ENTRY

Custom Staging Table

- Custom Header Level Data Table
  XXPAG_XLA_JE_HEADERS_IF
- Custom Line Level Data Table
  XXPAG_XLA_JE_LINES_IF
- Custom Errors Table
  XXPAG_XLA_JE_IF_ERRORS

Demo

Steps To Create Custom WebADI Integrators

Steps to Create the Custom Web ADI

1. Create an Integrator
   (Resp = Desktop Integration Manager)
   Table or API to insert into
   - Integrator, Interface, Content, Uploader, Importer

2. Create a Component (optional)
   (Resp = Desktop Integration Manager)
   List of Values to be used by the Integrator

3. Create a Layout
   (Resp = Desktop Integration)
   The display characteristics of the columns from the Integrator

4. Create a Document
   (Resp = Desktop Integration)
   This is the actual Web ADI Template
   Optionally Save as a Menu Function

Creating Custom Web ADI

- You can use the old PL/SQL API method to define your Integrator
  - PL/SQL Procedure BNE_INTEGRATOR_UTILS
    - ✓ CREATE_INTEGRATOR
    - ✓ CREATE_INTERFACE_FOR_API
    - ✓ CREATE_DEFAULT_LAYOUT
    - ✓ UPDATE_INTERFACE_COLUMN_TEXT
    - ✓ CREATE_TABLE_LOV

or

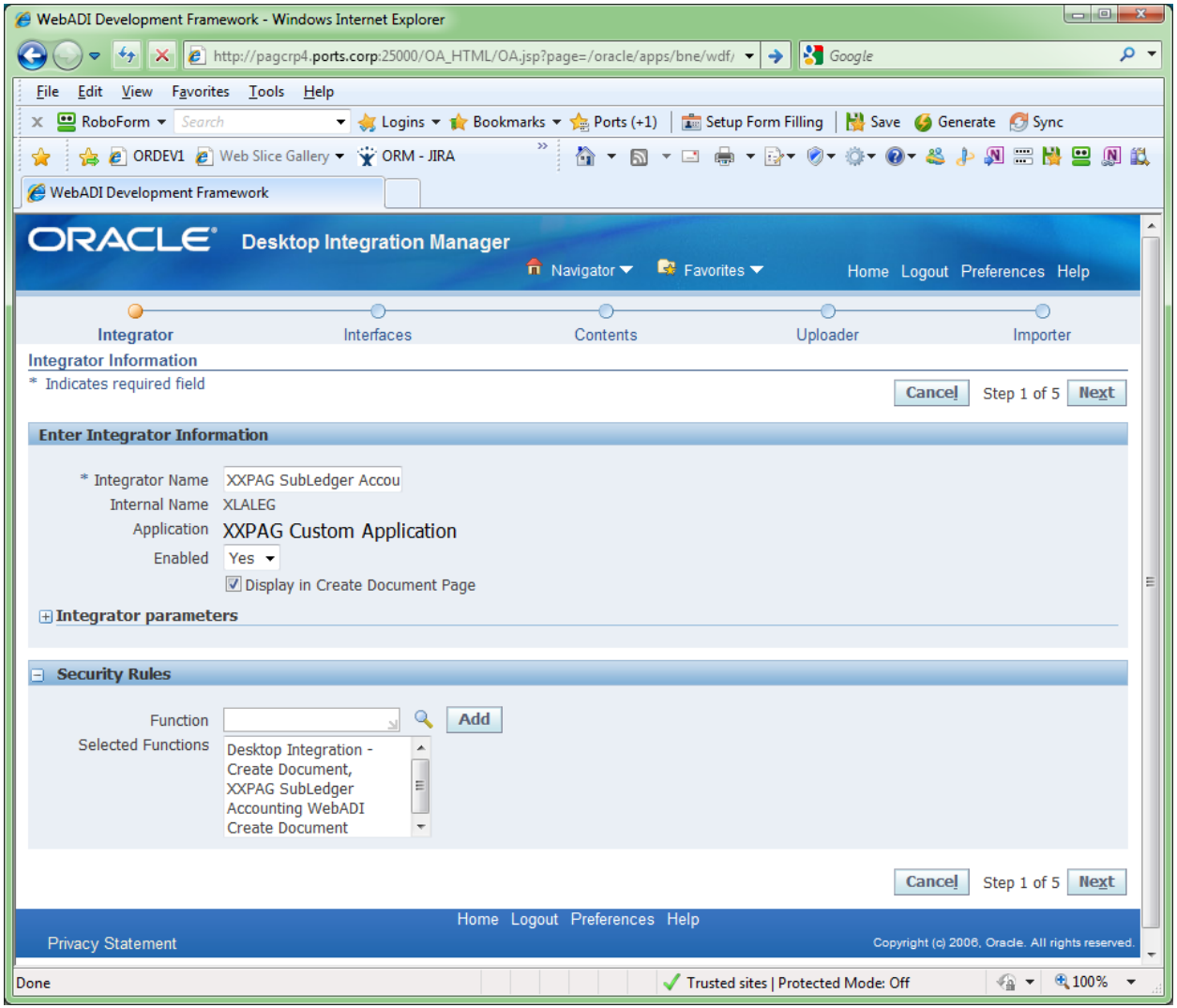
- You can use the new User Interface (Wizards)
  - **Integrator & Interface**
    Responsibility: Desktop Integration Manager
  - **Layout**
    Responsibility: Desktop Integration

Create the Integrator & Interface

Create the Integrator & Interface

- I will show the wizard based UI method of creating these
- The Integrator is the essentially the name that holds all of the meta data together for you spreadsheet integration
- The Interface is how the data is uploaded from the desktop appication to the Oracle E-Business Suite
  - Table
  - PL/SQL API Procedure
  - PL/SQL API Function

Create the Integrator

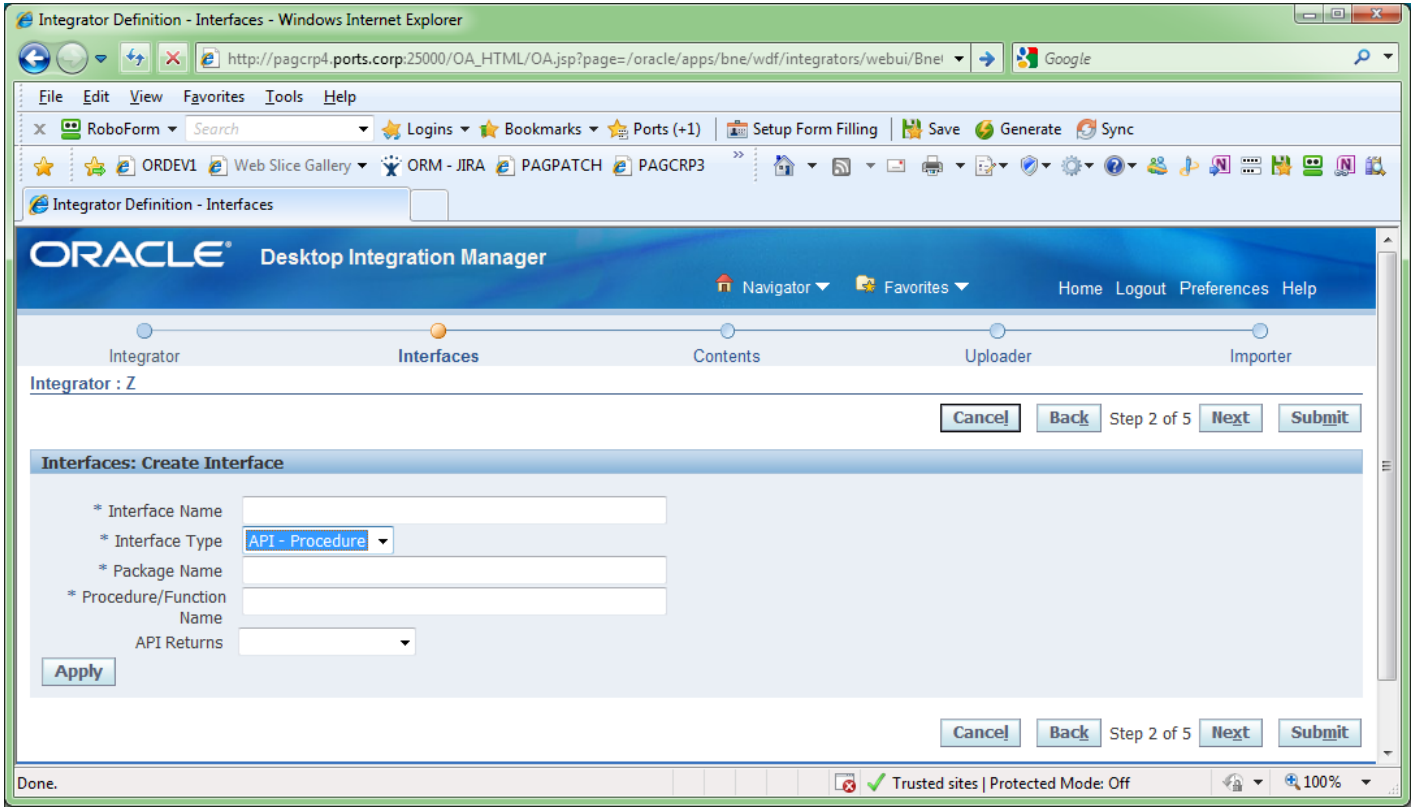


Integrator Fields

- **Integrator Name** – name you will see if the forms
- **Internal Name** – keep this name short
- **Application** – create under your custom application name
- **Enabled** – this appears to be the only way to disable an integrator
- **Display in Create Document Page** – I allow this, more on this later
- **Integrator Parameters** – you can create a parameters page required to specify constraints on a download document
- **Security Rule** – I create my own function and assign this, along with the seeded ‘Desktop Integration – Create Document’ which allows the users to use the Create Document Page

- Click **Next**

Create the Interface – step 1



Interface Fields – step 1

- **Interface Name** – name you will see if the forms, keep this name short (6 to 8 characters) since it is used by the wizard (or PL/SQL API) to autocreate column definitions
- **Interface Type** – select an interface type
  - Table
    - Table Name
  - API – Procedure
    - Package Name
    - Procedure/Function Name
    - API Returns
  - API – Function
    - Package Name
    - Procedure/Function Name
    - Returns error information
    - API Returns
- Click **Apply**

Interface Caveats

- Oracle currently only supports one interface per integrator

- Oracle currently only supports interfaces to a single level data structure
  Example: there are only lines, no header and lines

- This is kind of frustrating because the Excel Spreadsheet has Header and Line data structures. WebADI combines them when uploading the data.

- To get around this you have to create a PL/SQL Procedure that:
  - ✓ The parameters must be header columns, then line columns
  - ✓ Procedure finds the header record
  - ✓ If not found it creates the header record
  - ✓ Then insert a line record referencing the existing header
  - ✓ Repeat for each row

My Interface PL/SQL

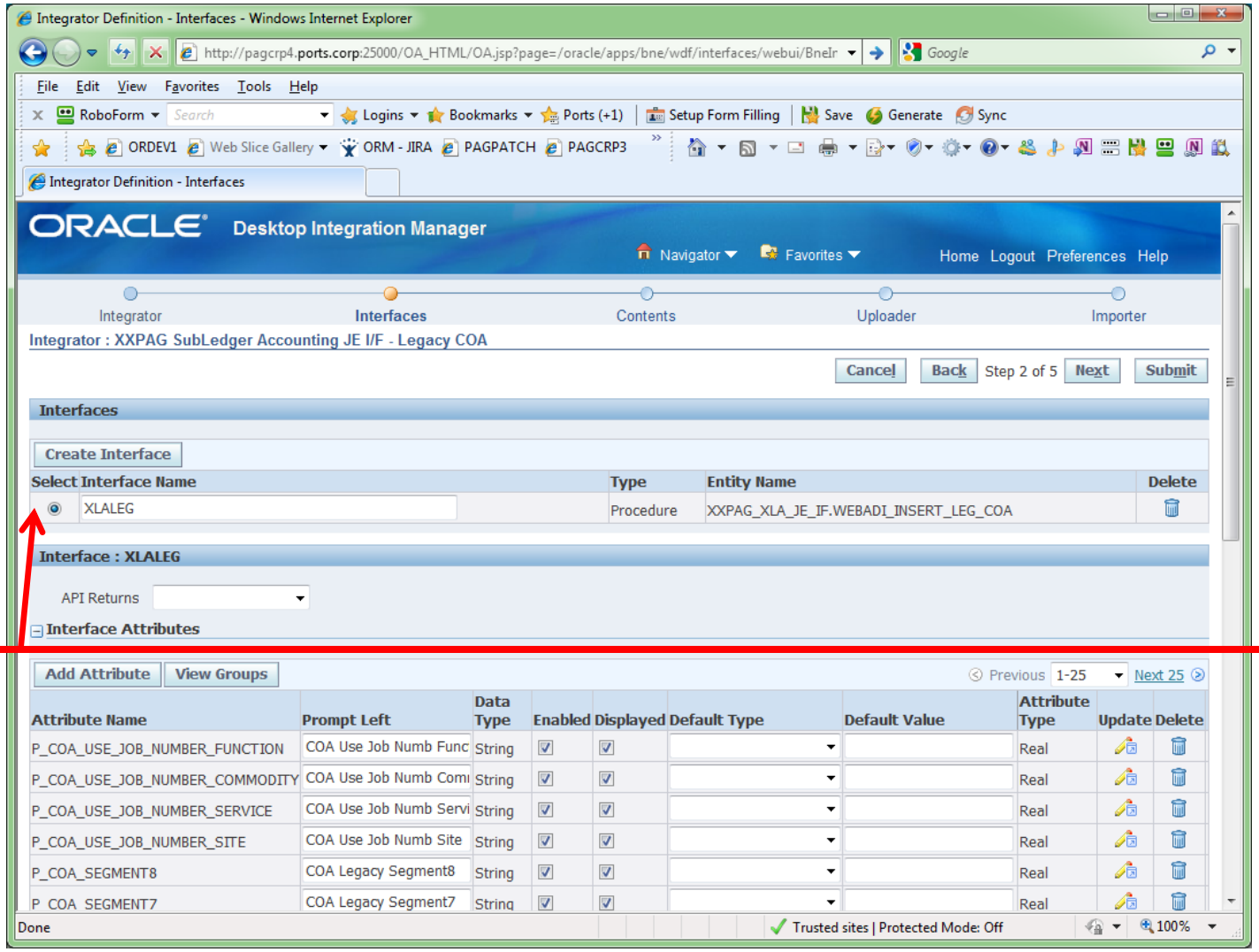
- PL/SQL Procedure to insert New COA data
  XXPAG_XLA_JE_IF.WEBADI_INSERT_NEW_COA

- PL/SQL Procedure to insert Legacy COA data
  XXPAG_XLA_JE_IF.WEBADI_INSERT_LEG_COA

- Both of these insert records into the custom staging table using a common routine. This was done to narrow down the parameters to just those that are required for each type of action.

- For each row of data we need to check if there is a header record if not create it for the first row

Create the Interface – step 2



Interface Fields – step 2

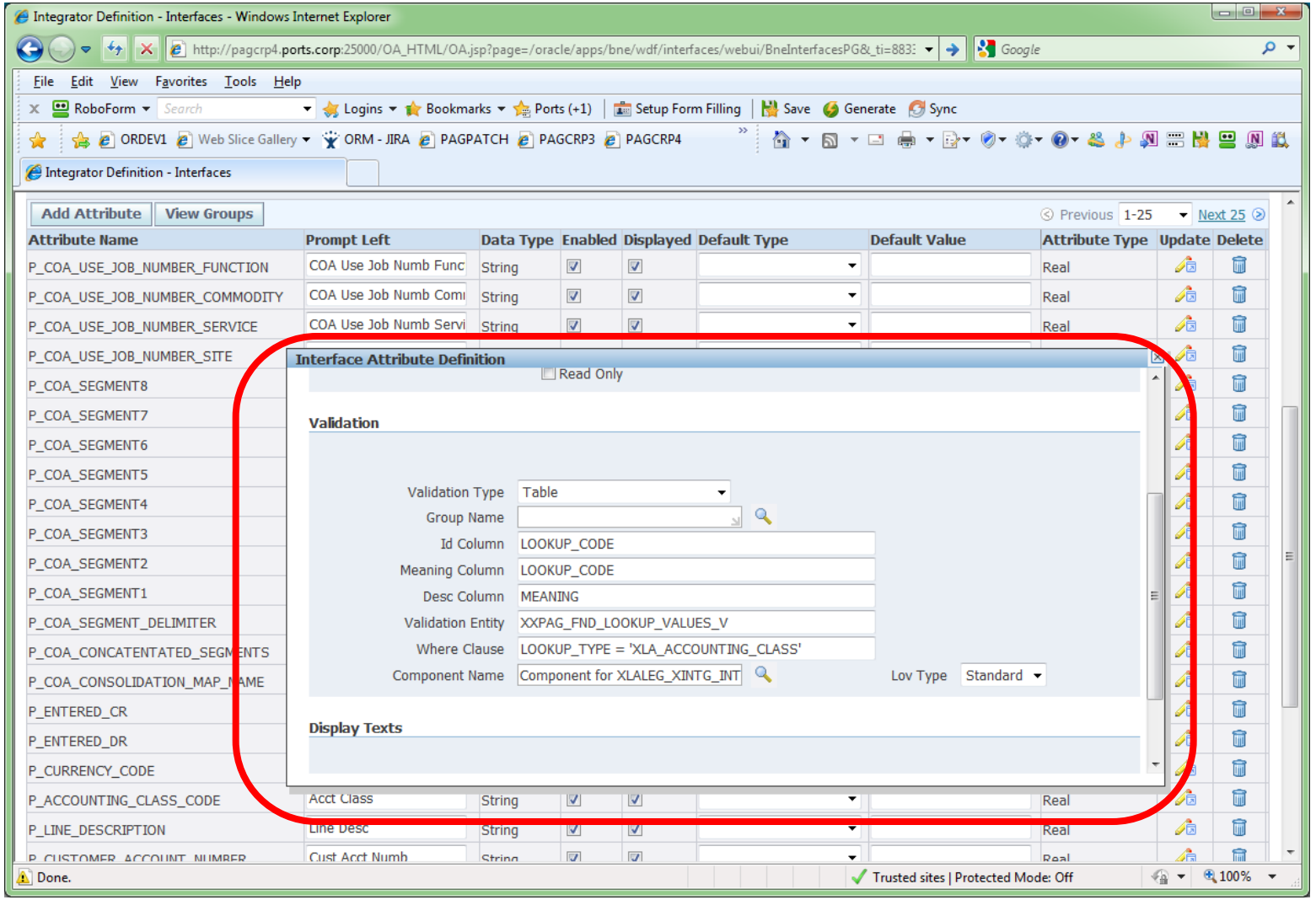
- You must select the Interface Name
- Wait a few seconds for the bottom region to become visible
- I have found this form only works in IE, when ever I try to use it in Firefox it is way too slow to be usable

Interface Fields – step 2

- **Attribute Name** – RO the PL/SQL Parameter, or table column name
- **Prompt Left** – left hand prompt in the document
- **Data Type** – RO from the parameter or column defintion
- **Enabled** – should the attribute be used
- **Displayed** – should the attribute be displayed
- **Default Type** – Constant, Environment Variable, Parameter, Profile Option, SQL Query, Lookup
- **Attribute Type** – RO describing where it originated from

- Click **Update**

Create the Interface – step 2



Interface Fields – step 2

- **Required** – Should this attribute be included in all laoyouts
- **Enabled for Mapping** – left hand prompt in the document
- **Available for Summary** –
- **Not NULL** – Prevent leaving the attribute NULL
- **Read Only** – Prevent the attribute from being changed
- **Validation Type** – Descriptive FlexField, Descriptive FlexField Context, Descriptive FlexField Segment, Group, Java, Key FlexField, Key FlexField Segment, Table
- **Group Name** –

Interface Fields – step 2

Table Validation

- **ID Column** – LOOKUP_CODE
- **Meaning Column** – MEANING
- **Desc Column** – DESCRIPTION
- **Validation Entity** – View you want query against
- **Where Clause** – Your where clause for the view
- **Component Name** – Prior defined component name
- **LOV Type** – None, Pop List, Standard

- Click **Save**
- Click **Submit**

Create Validations Using the PL/SQL APIs

- I found that the UI was way too cumbersome to create the WebADI LOV's that show up in the spreadsheet.
- So I used the old PL/SQL API's for defining the WebADI LOV's.

Create Validations Using the PL/SQL APIs

- I found that the UI was way too cumbersome to create the WebADI LOV's that show up in the spreadsheet.
- So I used the old PL/SQL API's for defining the WebADI LOV's.
- First create the required views:

```
create or replace view XXPAG_FND_FLEX_VALUES_V as
   SELECT ffvs.FLEX_VALUE_SET_NAME,
          ffv.FLEX_VALUE_SET_ID,
          to_char(ffv.FLEX_VALUE) flex_value,
          to_char(ffvtl.DESCRIPTION) description
     FROM FND_FLEX_VALUES_TL ffvtl,
          FND_FLEX_VALUES ffv,
          FND_FLEX_VALUE_SETS ffvs
    WHERE ffvtl.LANGUAGE = 'US'
      AND ffv.FLEX_VALUE_ID = ffvtl.FLEX_VALUE_ID
      and ffv.ENABLED_FLAG = 'Y'
      and SYSDATE between nvl(ffv.START_DATE_ACTIVE, SYSDATE)
and nvl(ffv.END_DATE_ACTIVE, SYSDATE)
      AND ffvs.FLEX_VALUE_SET_ID = ffv.FLEX_VALUE_SET_ID;
```

Create Window LOV's

```
declare
  v_interface_code varchar2(200) := 'XLANEW_XINTG_INTF1';
begin
  BNE_INTEGRATOR_UTILS.CREATE_TABLE_LOV
   (P_APPLICATION_ID        => 20003,
    P_INTERFACE_CODE        => v_interface_code,
    P_INTERFACE_COL_NAME    => 'P_SOURCE_SYSTEM',
    P_ID_COL                => 'FLEX_VALUE',
    P_MEAN_COL              => 'FLEX_VALUE',
    P_DESC_COL              => 'DESCRIPTION',
    P_TABLE                 => 'XXPAG_FND_FLEX_VALUES_V',
    P_ADDL_W_C              => 'FLEX_VALUE_SET_NAME = ''XXPAG_SOURCE_SYSTEMS''',
    P_WINDOW_CAPTION        => 'Source Systems',
    P_WINDOW_WIDTH          => 400,
    P_WINDOW_HEIGHT         => 500,
    P_TABLE_BLOCK_SIZE      => 10,
    P_TABLE_SORT_ORDER      => 'FLEX_VALUE',
    P_USER_ID               => 1110,
    P_POPLIST_FLAG          => 'N',
    P_TABLE_COLUMNS         => 'FLEX_VALUE,DESCRIPTION');
end;
```

Create Pop-List LOV's

```
declare
  v_interface_code varchar2(200) := 'XLANEW_XINTG_INTF1';
begin
  BNE_INTEGRATOR_UTILS.CREATE_TABLE_LOV
   (P_APPLICATION_ID        => 20003,
    P_INTERFACE_CODE        => v_interface_code,
    P_INTERFACE_COL_NAME    => 'P_PRODUCT_CODE',
    P_ID_COL                => 'PRODUCT_CODE',
    P_MEAN_COL              => 'PRODUCT_CODE',
    P_DESC_COL              => 'APPLICATION_NAME',
    P_TABLE                 => 'XXPAG_XLA_PRODUCT_CODES_V',
    P_ADDL_W_C              => NULL,
    P_WINDOW_CAPTION        => 'Product Codes',
    P_WINDOW_WIDTH          => 400,
    P_WINDOW_HEIGHT         => 500,
    P_TABLE_BLOCK_SIZE      => 10,
    P_TABLE_SORT_ORDER      => 'PRODUCT_CODE',
    P_USER_ID               => 1110,
    P_POPLIST_FLAG          => 'Y',
    P_TABLE_COLUMNS         => 'PRODUCT_CODE,APPLICATION_NAME');
end;
```

Optional Elements

- **The following are optional elements. You don't have to define them to get your Integrator working. Play with them later once you have defined your first Integrator.**

- **Contents**
  The definition of what data the integrator should initially populate the desktop document with. Effectively the query.

- **Uploader**
  The definition of what parameters the user should be prompted for during the upload action.

- **Importer**
  The definition of what to do once the data is uploaded, example, run a concurrent program.

Create the Layout

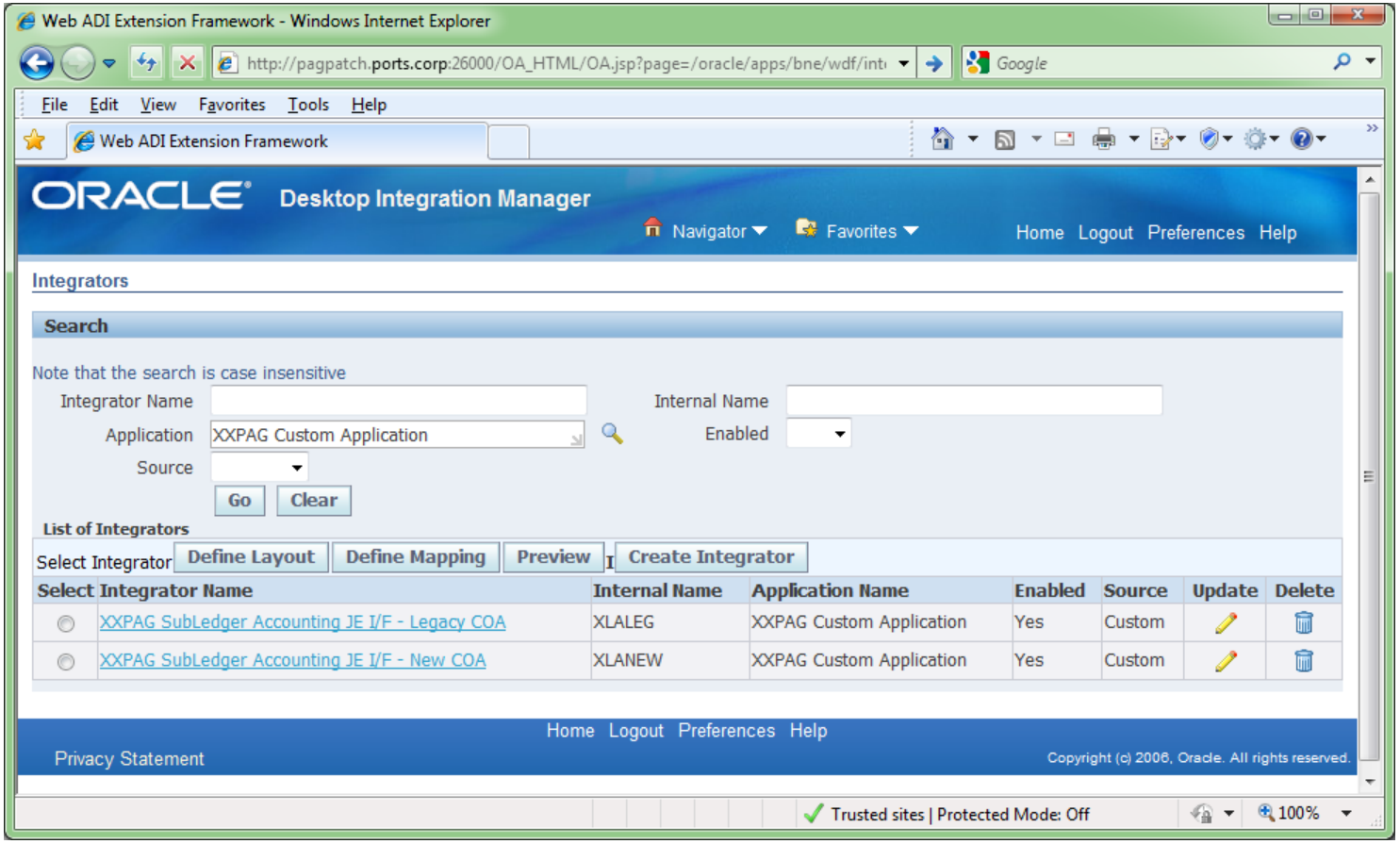
Create the Layout

- Layouts define how the Integrator will look in the Desktop Document
- An Integrator can have multiple Layouts, this allows you to create user business process specific Web ADI's

  Examples:
  - ✓ One for Payables SubLedger Accounting Journal Entries
  - ✓ One for Receivables SubLedger Accounting Journal Entries
  - ✓ One for a each External System
- You can easily duplicate an existing Layout then make the changes you need
- This is can easily be done by a non-technical user
- Use the responsibility:
  Desktop Integration

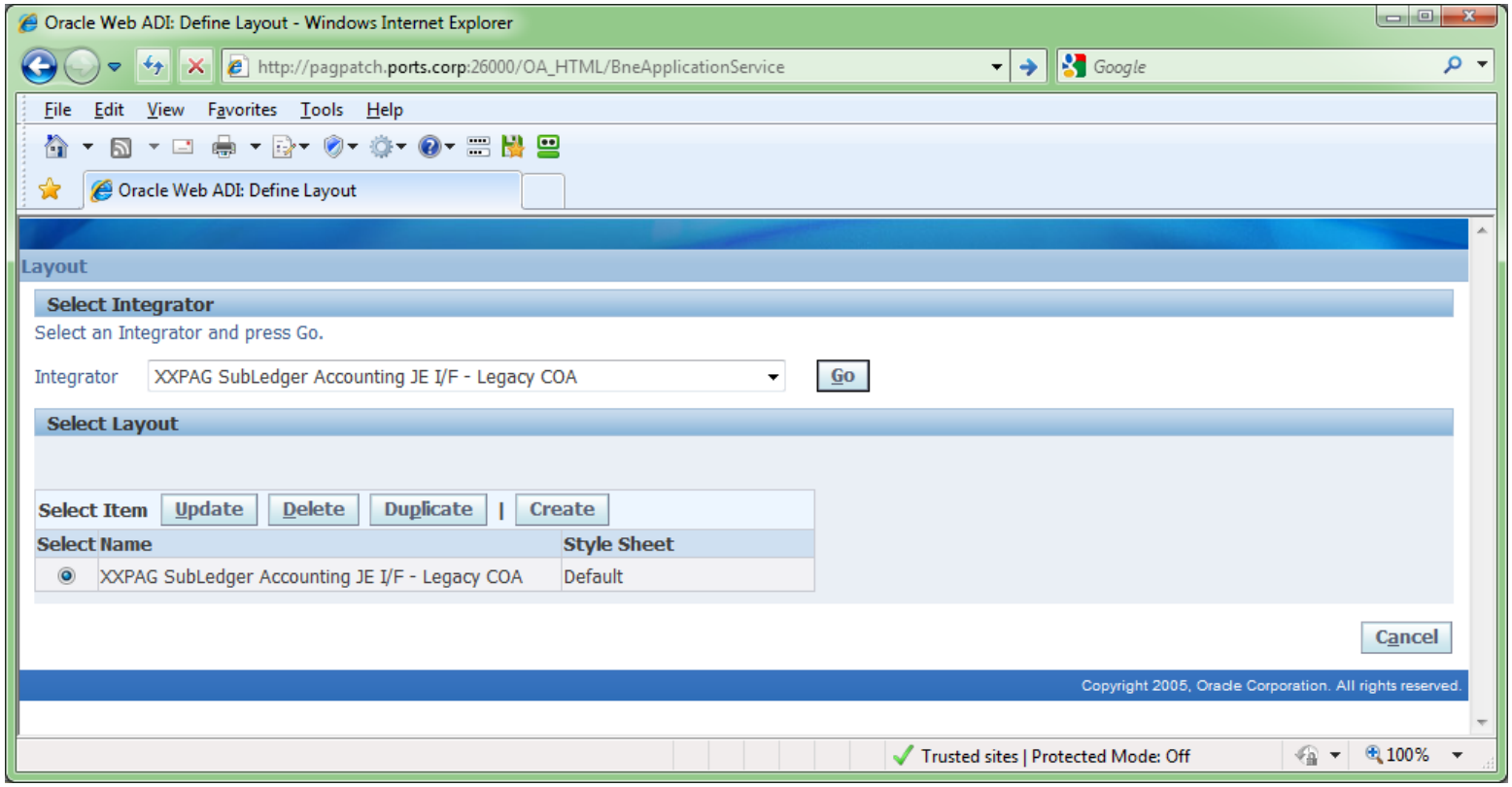
Create the Layout

- This can also be accessed from the Integrators page



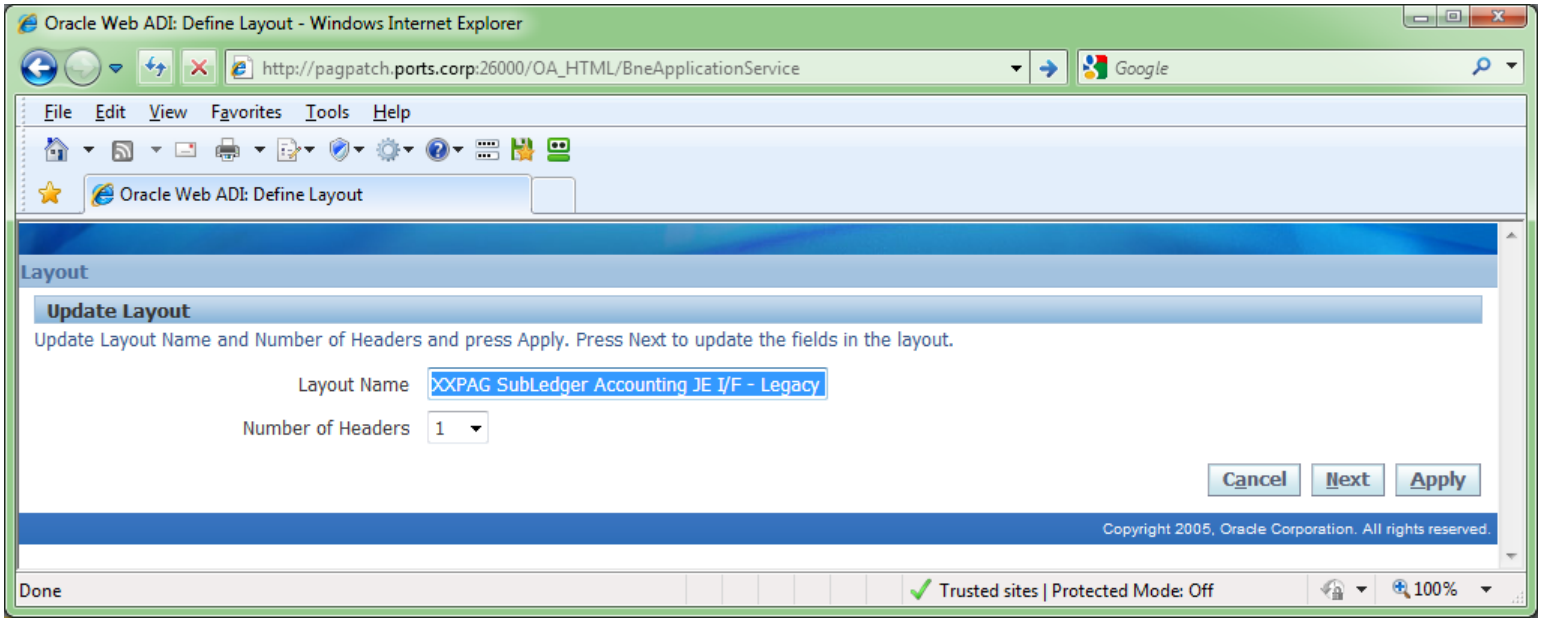
Create the Layout

- Menu Item: Manage Layout
- Select the Integrator, then an existing Layout and Action, or Create



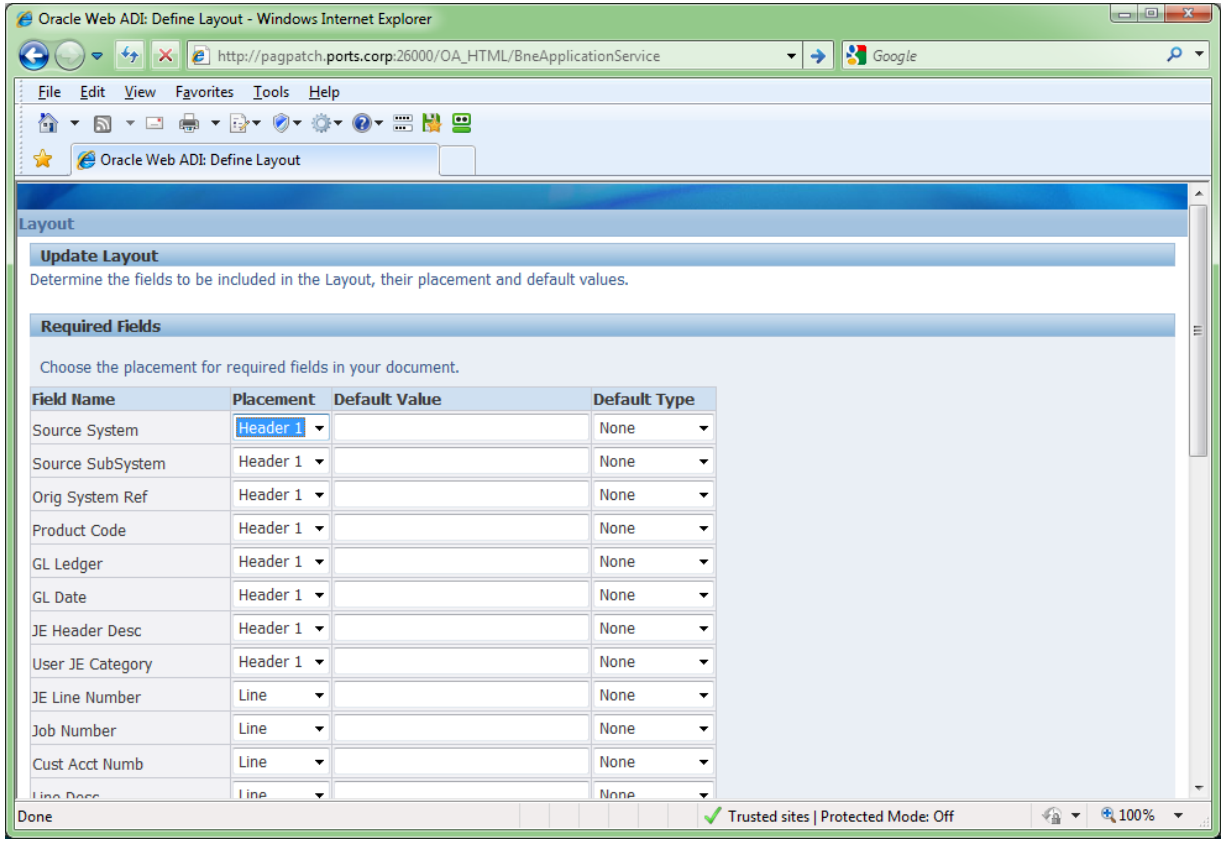
Create the Layout

- Name the Layout
- Specify the Number of Header sections in the spreadsheet



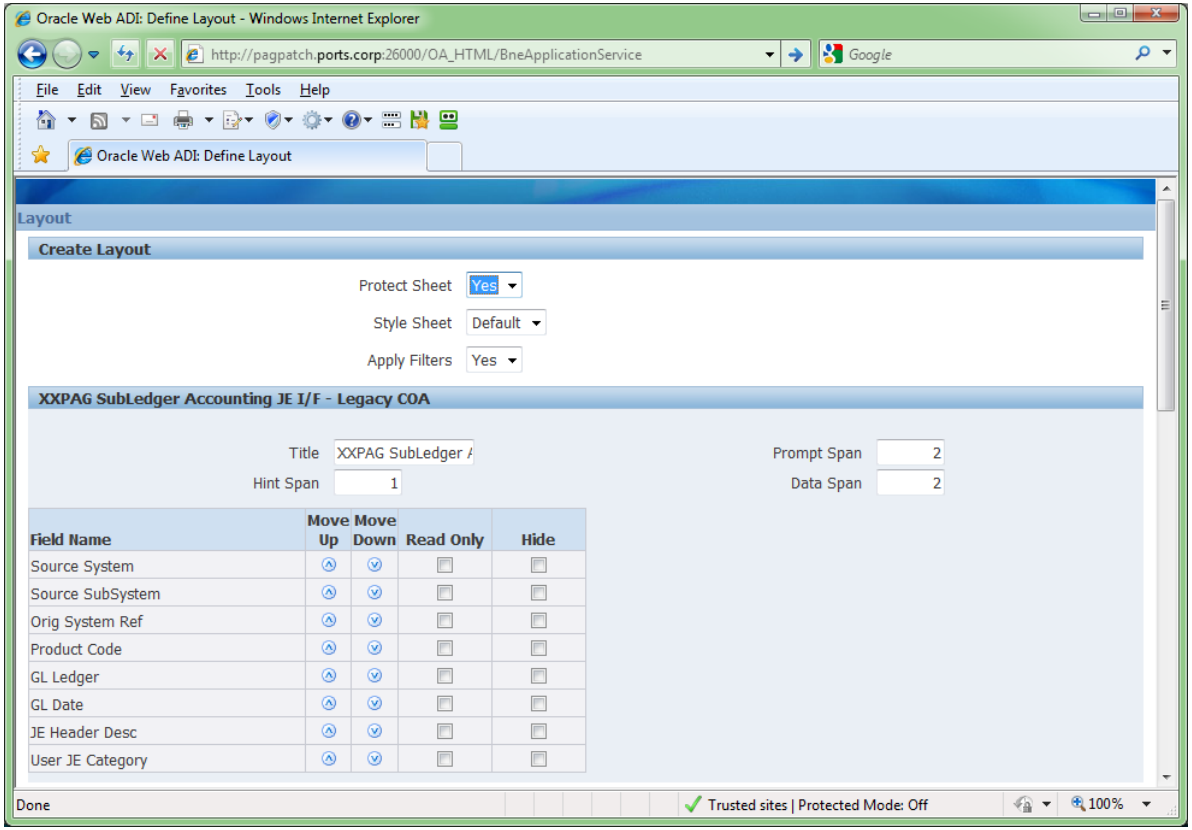
Create the Layout

- Specify which Fields of the Interface go in which section of the document HeaderN or Lines
- Specify the Default Values and Default Type



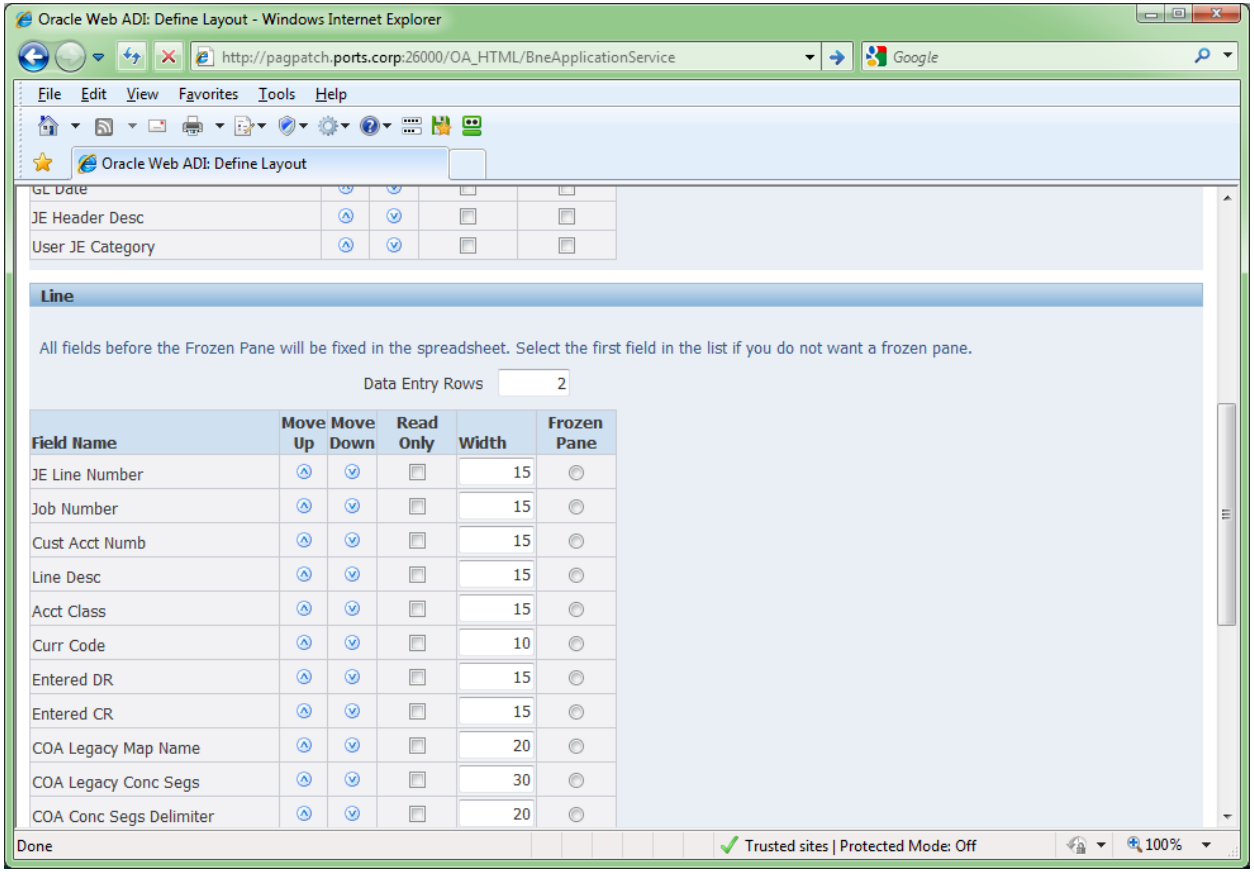
Create the Layout

- Should the sheet be protected when downloaded
- Style Sheet identifies the colors and fonts in the spreadsheet
- Order of Header Fields and if they are Read Only or Hidden



Create the Layout

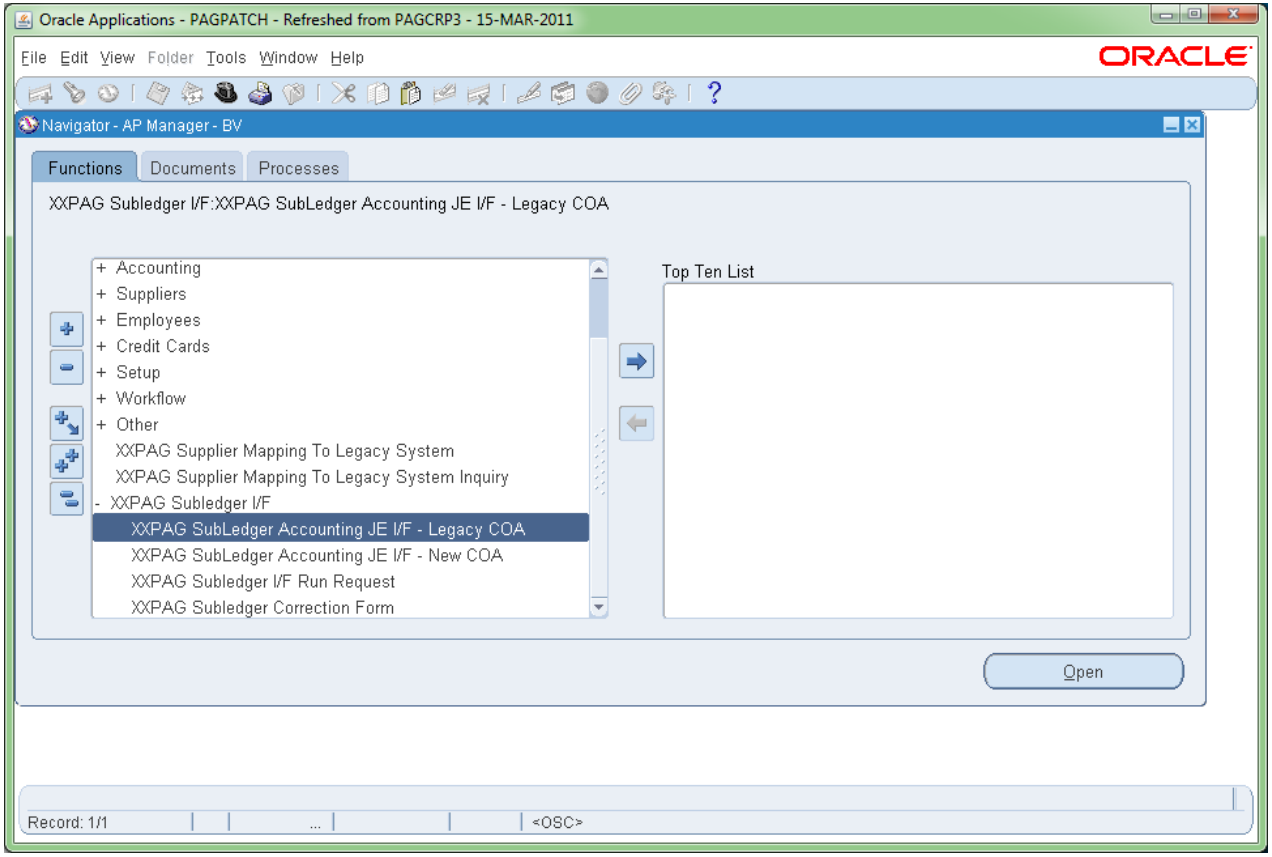
- Number of rows for user data entry (users can insert and delete)
- Order of Line Fields, if they are Read Only, Width
- Identify where the Frozen Pane ends



Save a Document as a Menu Function

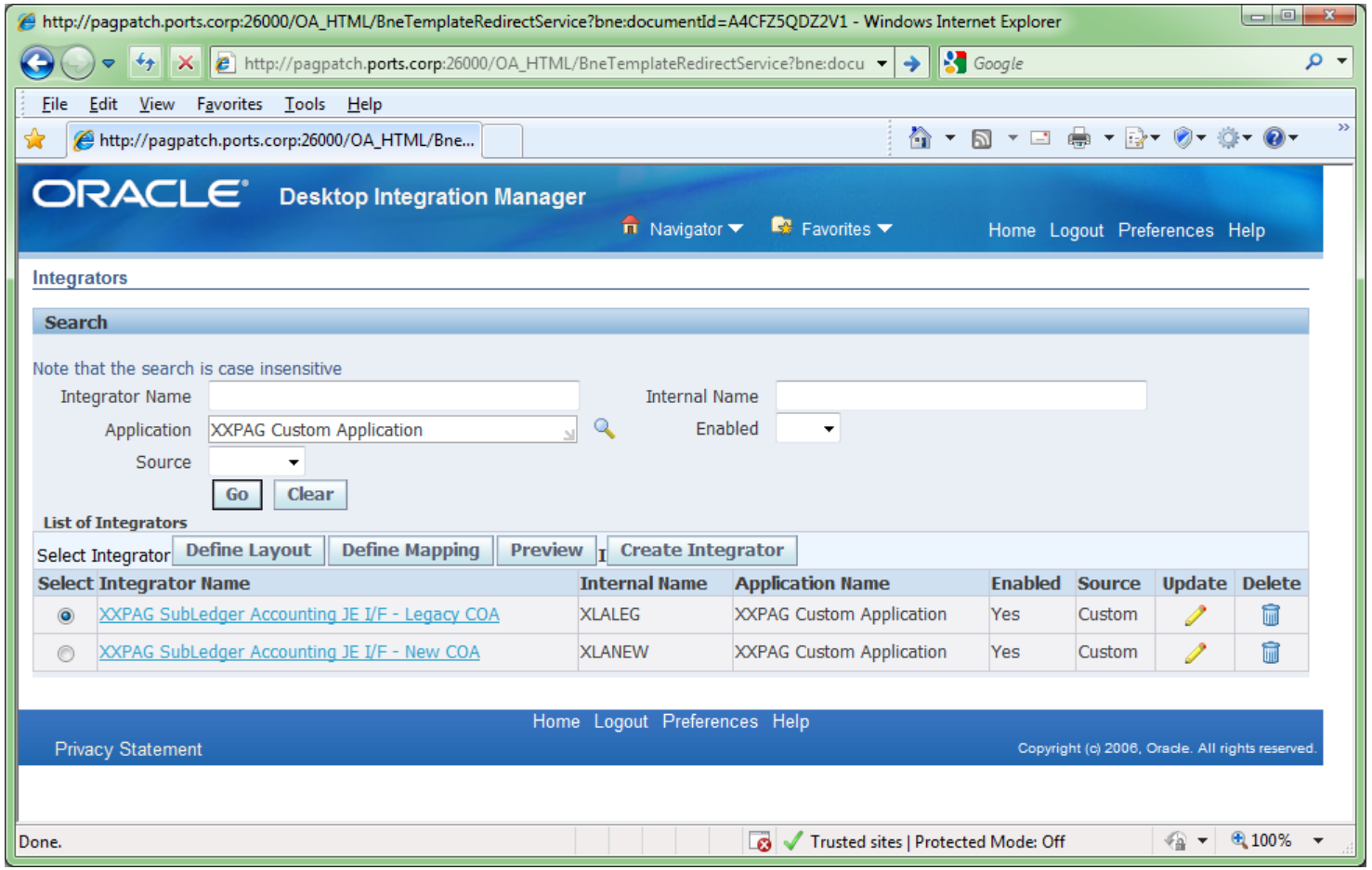
Save a Document as a Menu Function

- You can assign a function to the user's menu to allow them to download a specific WebADI Layout Template Spreadsheet
- Makes it far easier for the users to get the WebADI Document Template



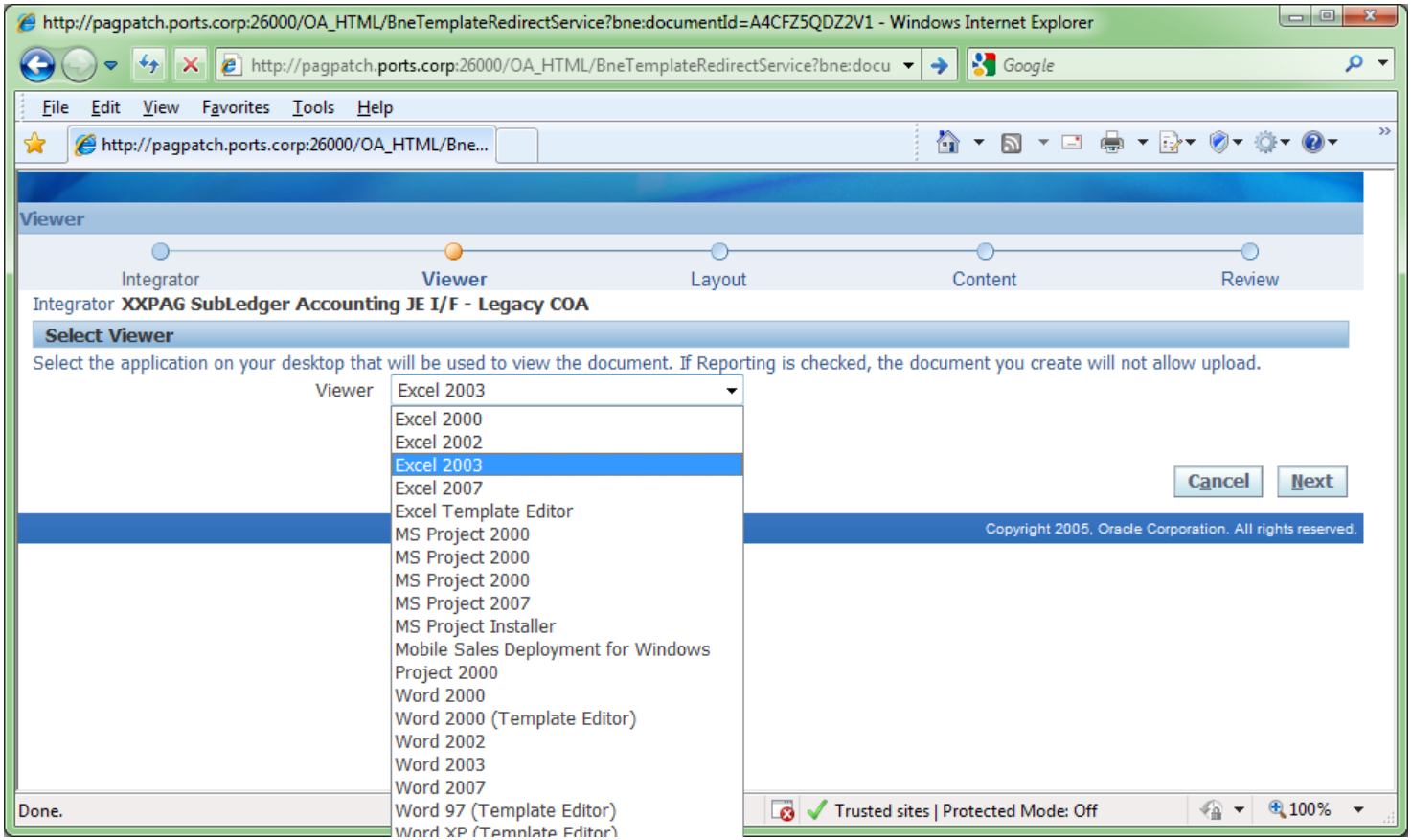
Save a Document as a Menu Function

- Select Integrator, Click on Preview



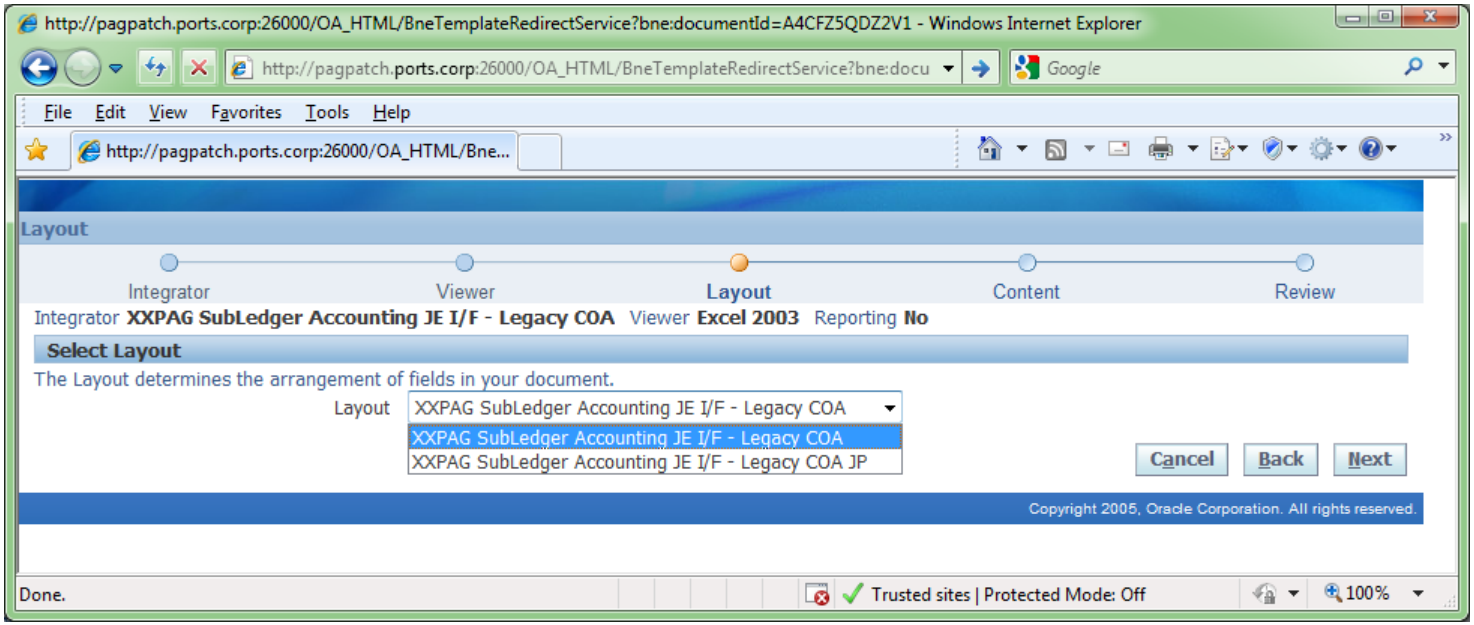
Save a Document as a Menu Function

- Select the Viewer you wish to create a Document for



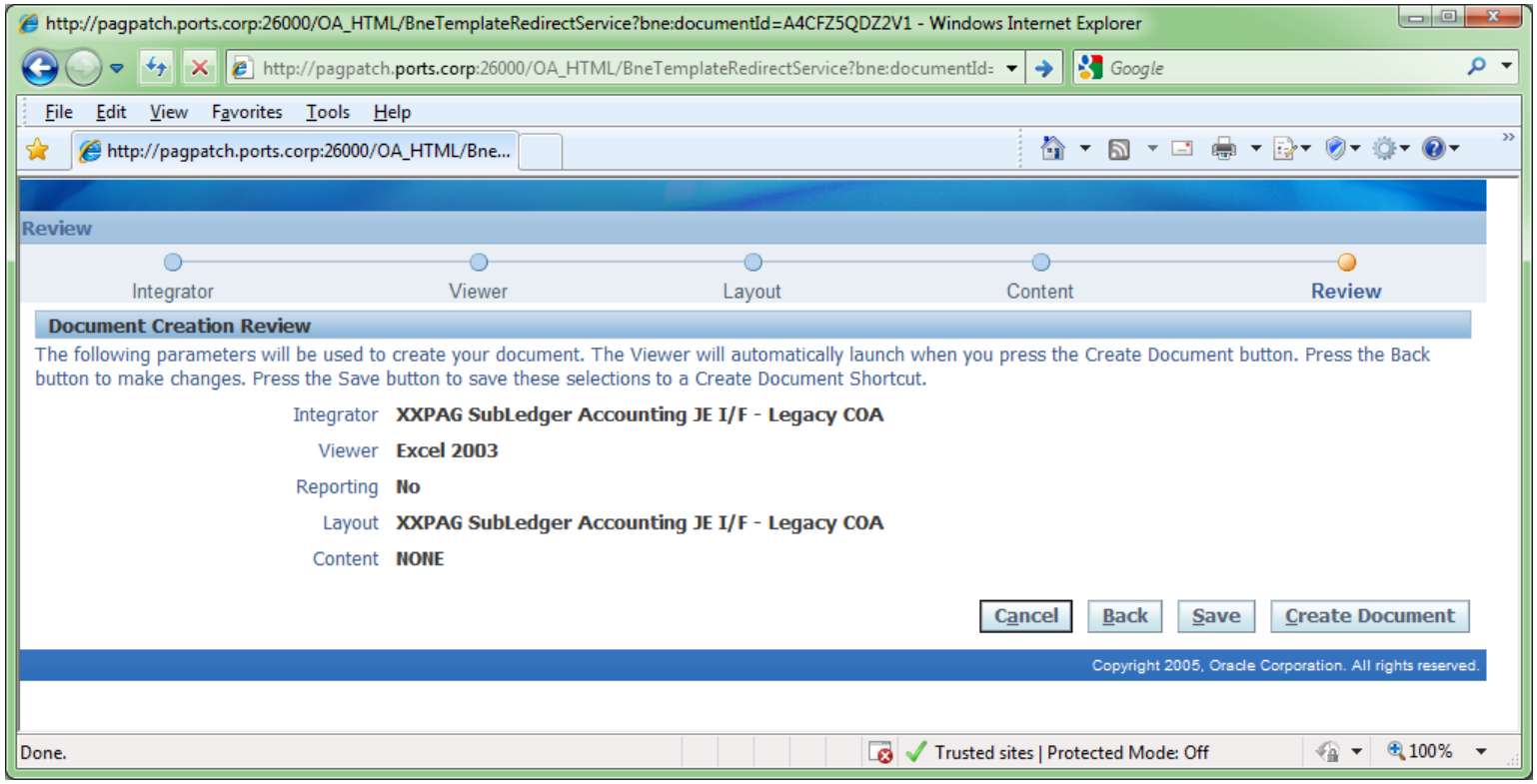
Save a Document as a Menu Function

- Select the Layout
- Optionally select a Content



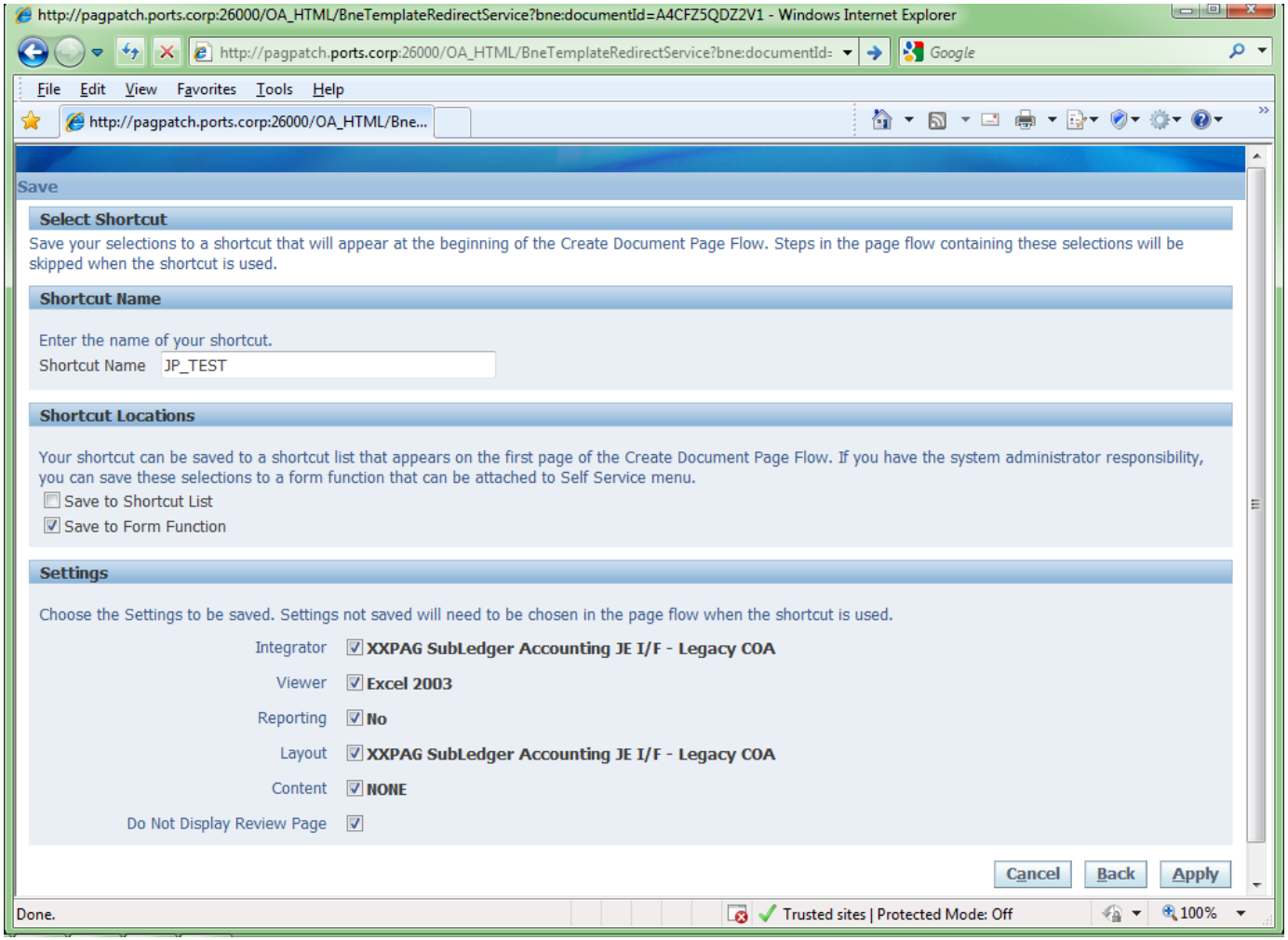
Save a Document as a Menu Function

- Click on Save to create Menu Function
- Click on Create Document to actually view the WebADI Document



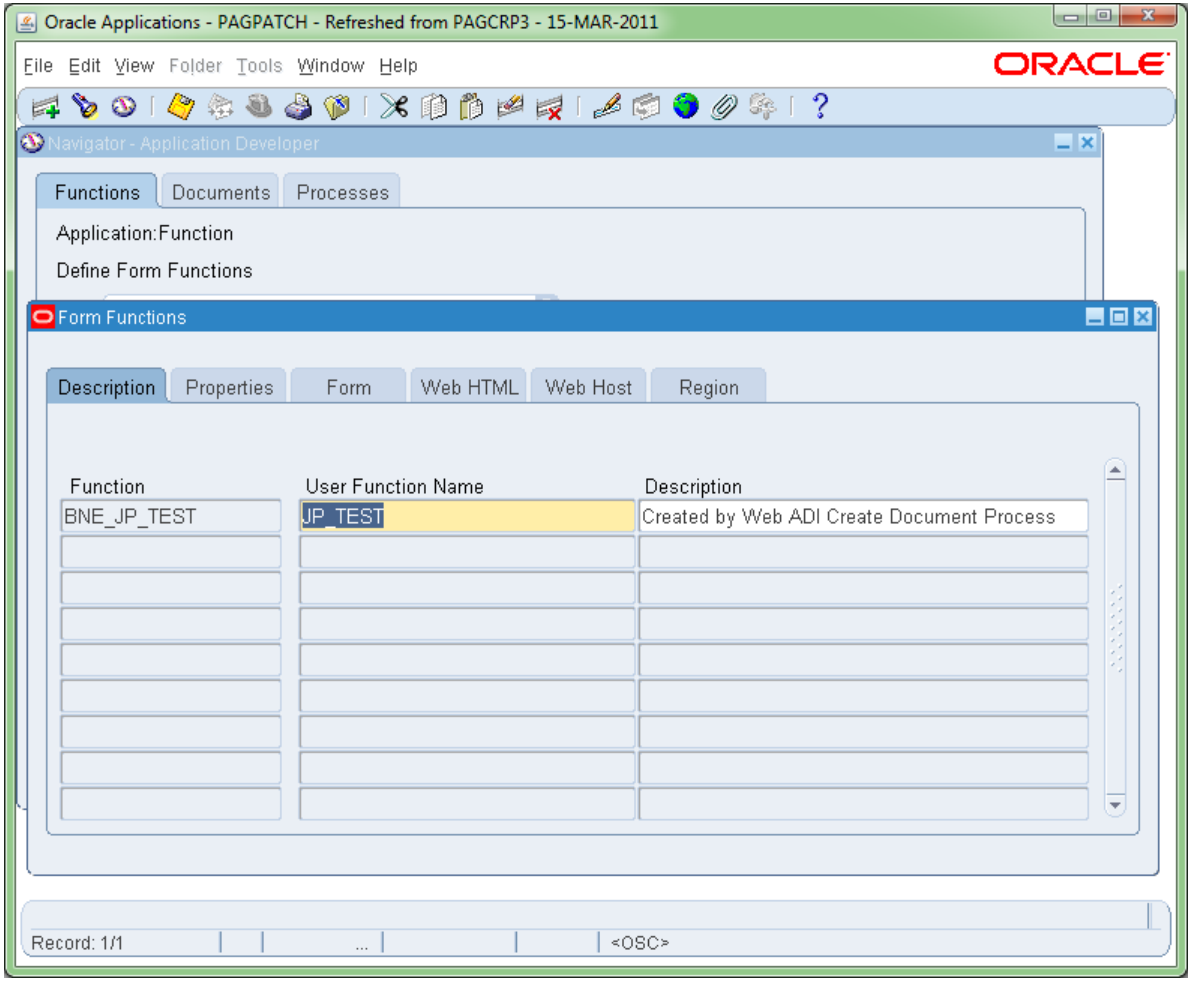
Save a Document as a Menu Function

- Provide the Shortcut Name, Select Save to Form Function, Apply



Save a Document as a Menu Function

- Function was created with the Shortcut Name as the User Function Name



Recent Enhancements

12.1.2 Enhancements

- **Oracle E-Business Suite Desktop Integration Framework**
  - User Interface placed on top of the WebADI API's to provide the starting functionality for creating a Custom WebADI.
  - Wizard based setup
  - You will still have to jump out to the API's for some tasks.

12.1.3 Enhancements

- **Oracle E-Business Suite Desktop Integration Framework Enhancements**
  - Upload parameters support
  - Importer rules support

- **Digital Signature Support**
  - You can lock down those risky Excel settings

12.2 Enhancements

- **Oracle E-Business Suite Desktop Integration Framework Enhancements**
  - Links to common pages (home, logout, diagnostics, preferences, help)
  - Logo on Spreadsheets
  - Expanders/Compressors on Spreadsheet
  - Ability to switch responsibilities prior to upload
  - Upload commit granularity on validations
  - Compress upload

- **Microsoft Office 2010 32bit and 64bit Certification**

- **WebADI Diagnostics**

Additional Resources

- **Oracle E-Business Suite Desktop Integration Framework Developer's** Guide
  Release 12.1, Part No. E15877-02

- **Oracle Web Applications Desktop Integrator Implementation and Administration Guide**
  Release 12.1, Part No. E12902-04

- **ML Note: 396181.1, Oracle Web Applications Desktop Integrator Documentation Resources, Release 12**
  Transfer of Information OnLine Training

Questions and Answers

My Contact Information

This was session # 8620, please complete the surveys

John Peters
JRPJR, Inc
john.peters@jrpjr.com

This paper is posted on my web site along with many others I have done over the years:
http://jrpjr.com/

Click on Paper Archives on the right hand side, then Collaborate11-WebADI for R12 SubLedger Accounting