

VERSION 2.0

**You Can't do THAT
With a Browser!**

Scott Spendolini
President & Founder, Sumner Technologies, LLC

Agenda

- Intro
 - Recap of Version 1
- Version 2
 - MS Outlook
 - dhtmlx Tree
 - FusionGadgets & FusionMaps
 - PL/PDF
- Q & A

About Sumner Technologies, LLC

- Founded in October 2005 by Scott Spendolini
 - Ex-Product Manager for Oracle APEX for 3 1/2 years
 - 10 years total tenure at Oracle
 - Supported Federal and Commercial clients
 - eBusiness Suite
- Currently 5 employees
 - All specializing in APEX and Oracle Technologies
 - Located all over the US

Introduction

- APEX provides a TON of functionality out of the box
 - PL/SQL Dynamic Content Region
 - Flash-based Charting
 - Export to Excel/PDF/Word
 - PDF & Word only when used with BI Publisher
 - Tree Report

Introduction

- However, some of the APEX components have limitations
 - Typical solution is to use Ajax to extend APEX's functionality
 - Requires learning JavaScript, XML and likely CSS
- There are a number of methods and third party components which allow you to extend APEX's functionality
 - All with little more than PL/SQL

Recap of Version 1.0

- Ajax Select List
- Ajax Memory Tree
- FCKEditor
- PL/PDF

Ajax Select List

- Dynamically retrieve some data when a value of a select list changes
- Four Components:
 - Page Zero
 - Application Process
 - JavaScript Event
 - JavaScript Function
- Working Example
 - <http://apex.oracle.com/pls/otn/f?p=33867:1>

Ajax Memory Tree

- OTN Tech Note written in 2005
- Combined Carl Backstrom's example DHTML Tree with the ability to "remember" which nodes were expanded
 - "Memory" capability now in APEX 3.0 Tree
 - Future APEX Tree components likely to be based on this concept
- Read the Tech Note:
 - <http://www.oracle.com/technology/pub/articles/spendolini-tree.html>

FCKEditor

- Rich, HTML-editor based on JavaScript
- Prior to APEX 3.0, could be integrated manually
- Now included in APEX as an Item Type
 - **HTML Editor Minimal** or **HTML Editor Standard**
 - Not **Textarea with HTML Editor**
- Works with both IE & Firefox

PL/PDF

- Still an excellent reporting alternative
- New feature in Release allow for easier report creation and add new capabilities
 - PDF Forms
 - Template Support
 - E-Mail PDF Files



Version 2.0

- Topics to be covered
 - MS Outlook “Integration”
 - dhtmlx Tree
 - FusionGadgets & FusionMaps
 - PL/PDF
- Almost all of the code required will be PL/SQL
 - Most JavaScript needed can simply be copied & pasted

MS Outlook “Integration”

- Two Points:
 - Address Book
 - Calendar
 - Will also work with other contact management tools
- How it Works:
 - Generate Address Book or Calendar Files from APEX via PL/SQL Process
 - Outlook will automatically import into its Address Book or Calendar
 - Need to understand vCard and iCalendar formats
 - Nothing more than a text file



Synchronizing Contacts

- vCard Standard
 - File format for electronic business cards
 - Contains name, address, phone, email, URLs, etc.
 - Readable by most contact manager applications
- Using PL/SQL, you can generate this file with as much or as little flexibility as you like
 - Rigid example:
 - Assume that all contacts only have one type of contact point
 - Flexible example:
 - Dynamically determine how many contact points of a specific type each contact has and account for that

Synchronizing Contacts

- We can track when a user downloads a vCard for a specific contact
- And then compare the download date to the `LAST_UPDATE` date of the contact record to determine who has the newer version
- If the server has a newer version, then we can indicate that visually by displaying an icon, for instance

Synchronizing Contacts

- BEFORE UPDATE & BEFORE INSERT triggers on the CONTACTS table
- Trigger should automatically populate the UPDATED_ON column with SYSDATE

BEFORE UPDATE & INSERT Triggers

```
CREATE OR REPLACE TRIGGER bi_contacts
  BEFORE INSERT ON contacts
  FOR EACH ROW
BEGIN
  IF :NEW.contact_id IS NULL THEN
    SELECT contacts_seq.NEXTVAL INTO :NEW.contact_id FROM dual;
  END IF;
  :NEW.last_updated := SYSDATE;
END;
/
```

```
CREATE OR REPLACE TRIGGER bu_contacts
  BEFORE UPDATE ON contacts
  FOR EACH ROW
BEGIN
  :NEW.last_updated := SYSDATE;
END;
/
```

Synchronizing Contacts

- BEFORE UPDATE & BEFORE INSERT triggers on the CONTACTS table
- Second table (CONTACT_DOWNLOADS) to track who has downloaded which contact and when
 - This table is used to track which user has downloaded which contact and when
 - Each time a contact is downloaded, a record should be inserted into CONTACT_DOWNLOADS
 - APEX Process should INSERT only CONTACT_ID into this table
 - Trigger will handle APP_USER and LAST_UPDATED values

CONTACTS_DOWNLOAD Table

```
CREATE TABLE contact_downloads
(
  contact_id          NUMBER,
  app_user            VARCHAR2 (255) ,
  downloaded_on      DATE
)
/

CREATE OR REPLACE TRIGGER bi_contact_downloads
  BEFORE INSERT ON contact_downloads
  FOR EACH ROW
BEGIN
  DELETE FROM contact_downloads WHERE
    contact_id = :NEW.contact_id
    AND app_user = v('APP_USER');
  :NEW.app_user := v('APP_USER');
  :NEW.downloaded_on := SYSDATE;
END;
/
```

Synchronizing Contacts

- BEFORE UPDATE & BEFORE INSERT triggers on the CONTACTS table
- Second table (CONTACT_DOWNLOADS) to track who has downloaded which contact and when
- **APEX Report which joins the two tables together**
 - If the actual contact data is newer than the corresponding row in CONTACT_DOWNLOADS or there are no rows in CONTACT_DOWNLOADS, then the report will display an icon indicating that the data in the database is newer than on the user's local directory

Contacts Query

```
SELECT
  c.contact_id,
  CASE
    WHEN
      cd.last_download IS NULL
      OR cs.last_download < SYSDATE THEN
      ''
    ELSE
      NULL
  END last_download_flag,
  c.first_name,
  c.last_name
FROM
  contacts c,
  contact_downloads cd
WHERE
  c.contact_id = cd.contact_id(+)
  AND cd.app_user = :APP_USER
```

Synchronizing Contacts

- BEFORE UPDATE & BEFORE INSERT triggers on the CONTACTS table
- Second table (CONTACT_DOWNLOADS) to track who has downloaded which contact and when
- APEX Report which joins the two tables together
- **PL/SQL Procedure to generate the corresponding vCard File**

DEMO

CONTACTS SYNCHRONIZATION

vCard PL/SQL

-- Record that a user downloaded a specific vCard

```
INSERT INTO contact_downloads(contact_id)
VALUES (p_contact_id);
```

```
FOR x IN (SELECT * FROM contacts c
WHERE c.contact_id = p_contact_id)
```

LOOP

```
l_full_name := x.first_name || ' ' || x.last_name;
l_vcard :=
'BEGIN:VCARD' || l_crlf ||
'VERSION:3.0' || l_crlf ||
'N:' || x.last_name || ';' || x.first_name || ';;; '
|| l_crlf ||
'FN:' || l_full_name || l_crlf ||
'ORG:' || x.company || ';' || l_crlf ||
'TITLE:' || x.title || l_crlf ||
'EMAIL;type=INTERNET;type=WORK;type=pref:' || x.email
|| l_crlf ||
'TEL;type=WORK;type=pref:' || x.office_phone || l_crlf ||
'TEL;type=CELL:' || x.mobile_phone || l_crlf ||
'item1.X-ABADR:us' || l_crlf ||
'X-ABUID:' || l_guid || '\:ABPerson' || l_crlf ||
'END:VCARD';
```

END LOOP;

vCard PL/SQL

```
-- Set the MIME header for a text file
owa_util.mime_header('application/text', FALSE );

-- Set the Content-Length to the corresponding file size
http.p('Content-length: ' || LENGTH(l_vcard));

-- Set the file name of the vCard based on the Full Name
http.p('Content-Disposition: filename="' || l_full_name || '.vcf"');

-- Close the HTTP header
owa_util.http_header_close;

-- Print the vCard
http.p(l_vcard);

-- Prevent anything else in APEX from processing
wwv_flow.g_unrecoverable_error := TRUE;
```

Events Synchronization

- Same general approach as Contacts
 - Use iCalendar or vCalendar specification instead of vCard
- Need two tables and a view:
 - EVENTS
 - EVENTS_CAL
 - EVENTS_V
- This is due to how APEX calendars handle events
 - Only support an Event Date, not a Start and End Date

EVENTS_CAL Trigger

```
CREATE OR REPLACE TRIGGER ai_events
  AFTER UPDATE OR INSERT ON events
  FOR EACH ROW
DECLARE
  l_date_diff NUMBER;
BEGIN
  IF UPDATING THEN
    DELETE FROM events_cal WHERE event_id = :NEW.event_id;
  END IF;
  l_date_diff := :NEW.end_date - :NEW.start_date;
  FOR x IN 0..l_date_diff
  LOOP
    INSERT INTO events_cal
      (event_id, event_date)
    VALUES
      (:NEW.event_id, TO_DATE (:NEW.start_date + x));
  END LOOP;
END;
/
```

EVENTS_V

```
CREATE OR REPLACE VIEW events_v
AS
SELECT
  e.event_id,
  '<a href="f?p=&APP_ID.:11:&APP_SESSION.:::P11_EVENT_ID:'
  || e.event_id || '">'
  || e.event_name
  || '</a>&nbsp;&nbsp;&nbsp;<a href="f?p=&APP_ID.: '
  || '12:&SESSION.:::P12_EVENT_ID:'
  || e.event_id
  || '"></a>' event_name,
  ec.event_date
FROM
  events e,
  events_cal ec
WHERE
  e.event_id = ec.event_id
/
```

DEMO

EVENTS SYNCHRONIZATION

dhtmlx Tree

- 3rd Party Tool by DHTMLX
 - <http://www.dhtmlx.com>
 - DHTMLeXtensions
- Specialize DHTML & XML controls
- Relatively easy to integrate with APEX

Available DHTML Controls

- **Tree**
- **Grid**
- **TreeGrid**
- **Menu**
- **Calendar**
- **Tabbar**
- **Combo**
- **Toolbar**
- **Folders**

Pricing

	Standard	Commercial	Enterprise
dhtmlx Tree	FREE	\$149	\$399
dhtmlx Grid	FREE	\$199	\$449
dhtmlx TreeGrid	N/A	\$299	\$549
dhtmlx Menu	FREE	\$49	\$99
dhtmlx Calendar	FREE	\$49	\$99
dhtmlx Tabbar	FREE	\$69	\$199
dhtmlx Combo	FREE	\$49	\$99
dhtmlx Toolbar	FREE	N/A	N/A
dhtmlx Folders	FREE	\$149	\$399

Pricing

- **Standard**
 - Allows for use of Standard Edition of code in your applications as per GPL
- **Commercial**
 - Allows for use of Professional Edition in one project
- **Enterprise**
 - Allows for use of Professional Edition in unlimited projects for a single company

APEX Tree vs. dhtmlx Tree

	APEX	dhtmlx Tree
Declarative	✓	
Requires PL/SQL Code		✓
Complete UI Control		✓
AJAX Support		✓
Additional Server Req'd		
Drag & Drop		✓
Partial Page Refresh		✓
Extra Cost		✓

How does it work?

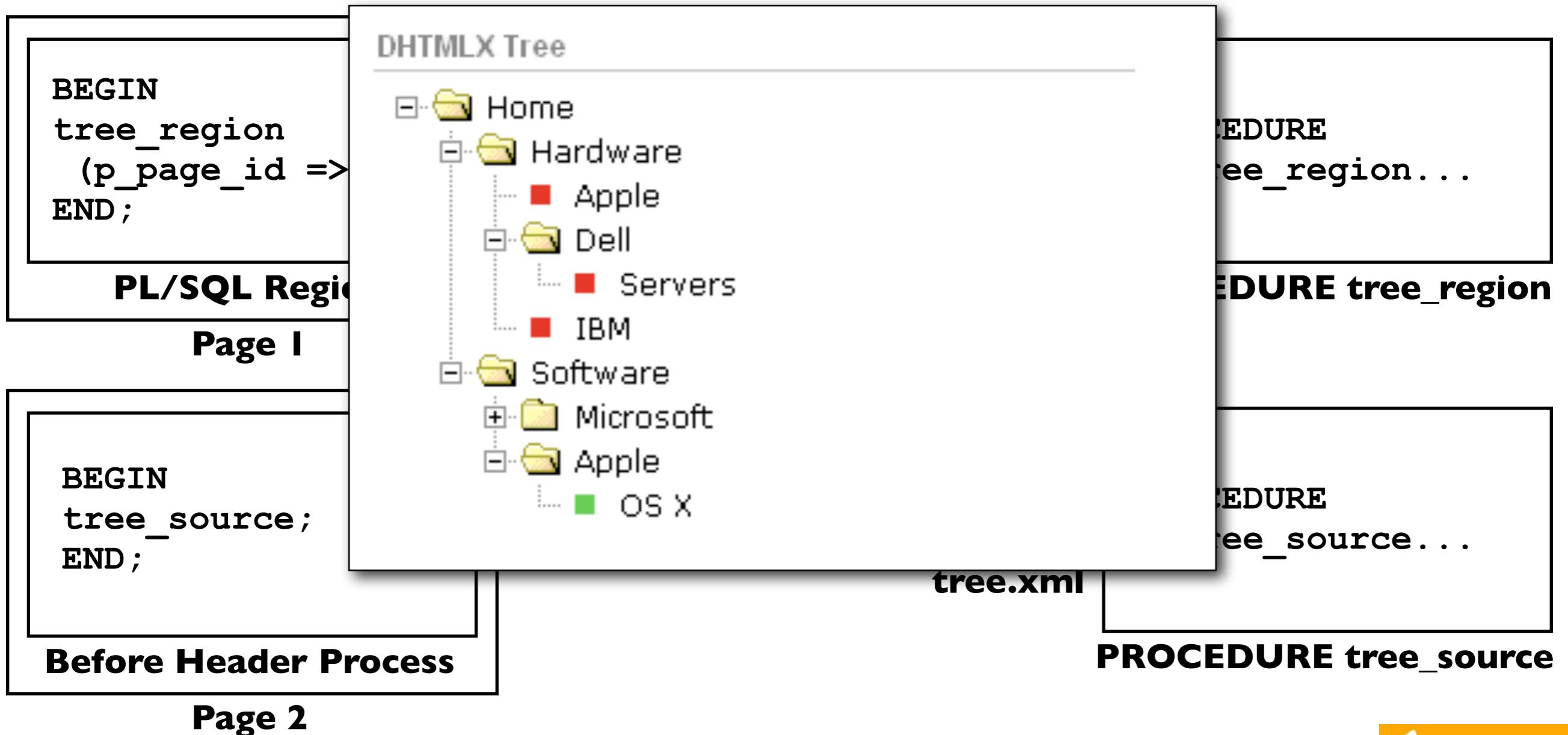
- Tree is DHTML & XML based
 - JavaScript libraries manage all aspects of it
 - Can be configured with a number of different parameters
- Uses an XML file as its data source
 - Can be static or more likely, dynamically generated
 - Using PL/SQL most likely
- Simple XML file structure:

```
<?xml version="1.0" encoding="iso-8859-1" ?>
<tree id="0" radio="1">
  <item text="Home" id="1" im0="closed.gif" open="Y">
    <item text="Hardware" id="2" im0="closed.gif">
      </item>
    <item text="Software" id="3" im0="closed.gif">
      </item>
    </item>
  </tree>
```

How does it work with APEX?



tree.js



DEMO

DHTMLX TREE

PROCEDURE tree_region

PROCEDURE region

(

```
p_codebase_path      IN VARCHAR2,  
p_images_path        IN VARCHAR2,  
p_width              IN VARCHAR2,  
p_height             IN VARCHAR2,  
p_div_id             IN VARCHAR2 DEFAULT 'treebox_tree'
```

)

AS

```
l_div                VARCHAR2(1000);
```

BEGIN

```
l_div := '  
<link rel="stylesheet" type="text/css" href="' || p_codebase_path || 'codebase/dhtmlxtree.css">  
<link rel="stylesheet" type="text/css" href="' || p_codebase_path || 'codebase/dhtmlxtabbar.css">  
<script src="' || p_codebase_path || 'codebase/dhtmlxcommon.js"></script>  
<script src="' || p_codebase_path || 'codebase/dhtmlxtabbar.js"></script>  
<script src="' || p_codebase_path || 'codebase/dhtmlxtabbar_start.js"></script>  
<script src="' || p_codebase_path || 'codebase/dhtmlxcommon.js"></script>  
<script src="' || p_codebase_path || 'codebase/dhtmlxtree.js"></script>  
<div id="' || p_div_id || '" style="width:' || p_width || ';height:' || p_height || '"></div>  
<script>  
function toggleOC(pID) { return true }  
tree=new dhtmlXTreeObject("' || p_div_id || '", "100%", "100%", 0);  
tree.setImagePath("' || p_images_path || '");  
tree.setOnOpenHandler(toggleOC);  
tree.loadXML("/apex/f?p=' || nv('APP_ID') || ':41:' || v('APP_SESSION') || '");  
</script>';  
htp.p(l_div);  
END region;
```

PROCEDURE tree_source

```
PROCEDURE tree_source
AS
  l_xml CLOB;
  l_count INTEGER;
  l_prev_level INTEGER := 0;
BEGIN
  -- Start with the XML header and open the TREE
  l_xml := '<?xml version="1.0" encoding="iso-8859-1"?>
<tree id="0" radio="1">';
  -- Loop through all items in the tree
  FOR x IN
  (SELECT
    id, pid, name, status, level
  FROM tree_v
  CONNECT BY PRIOR id = pid
  START WITH id = 1)
  LOOP
```

PROCEDURE tree_source

LOOP

-- If the current level is less than the previous level, then close the ITEM tag

```
IF x.level < l_prev_level THEN
  FOR y IN x.level..l_prev_level-1
  LOOP
    l_xml := l_xml || '</item>';
  END LOOP;
END IF;
```

-- Determine if there are any child nodes for a given node and render the
-- corresponding icon

```
SELECT count(*) INTO l_count FROM tree_t WHERE pid = x.id;
IF l_count = 0 THEN
  l_xml := l_xml || print_node(p_text => x.name, p_id => x.id, p_icon => 'icon',
    p_status => x.status) || '</item>';
ELSE
  l_xml := l_xml || print_node(p_text => x.name, p_id => x.id,
    p_icon => 'folder');
END IF;
```

-- Capture the previous level, which is used at the top of the loop

```
l_prev_level := x.level;
END LOOP;
```

PROCEDURE tree_source

```
-- Close out any ITEMS that may not yet be closed
```

```
FOR x IN 1..l_prev_level-1  
LOOP  
  l_xml := l_xml || '</item>';  
END LOOP;
```

```
-- Close the TREE
```

```
l_xml := l_xml || '</tree>';
```

```
-- Set the MIME header for an XML file
```

```
owa_util.mime_header('text/xml', FALSE );
```

```
-- Set the Content-Length so that the browser knows how much data to expect
```

```
htp.p('Content-length: ' || LENGTH(l_xml));
```

```
-- Set the file name of the Tree's data source
```

```
htp.p('Content-Disposition: filename="tree.xml"');
```

```
-- Close the HTTP header
```

```
owa_util.http_header_close;
```

```
-- Print the XML file
```

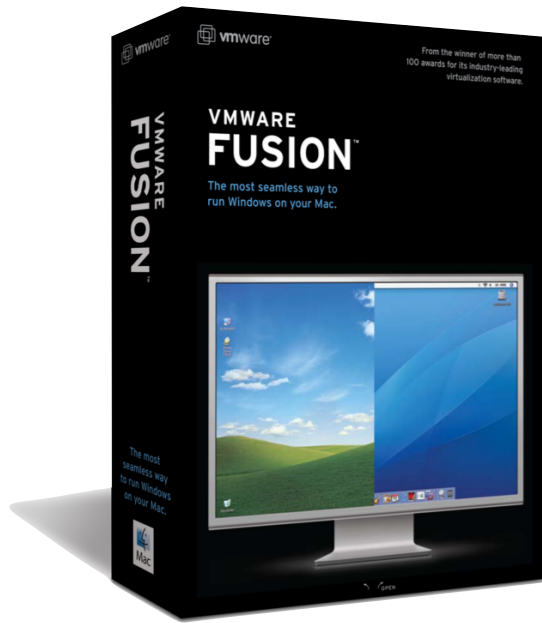
```
htp.p(l_xml);
```

```
-- Prevent anything else in APEX from processing
```

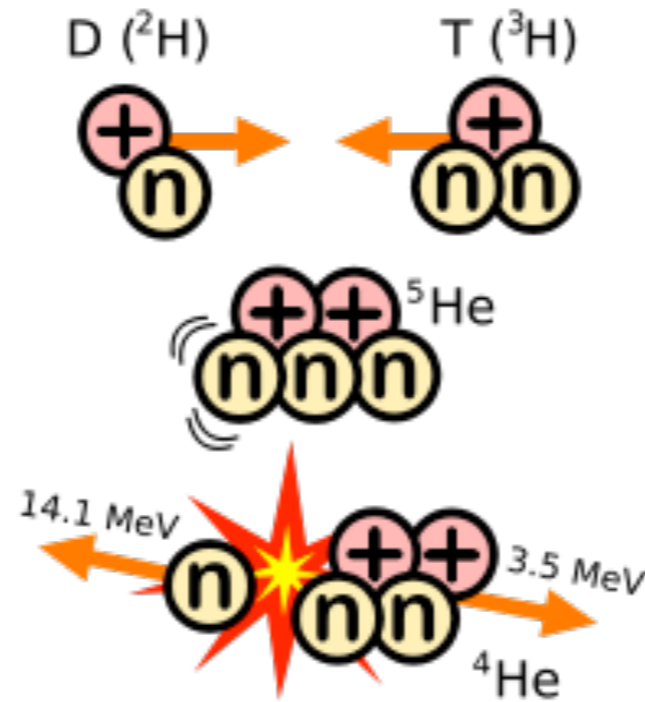
```
wwv_flow.g_unrecoverable_error := TRUE;
```

```
END tree_source;
```

What is Fusion?



ORACLE



Deuterium-tritium (D-T) fusion reaction



FusionGadgets & FusionMaps

- Developed by Infosoft Global
 - <http://infosoftglobal.com>
- Five products
 - FusionCharts
 - FusionCharts Free
 - Subset of Fusion Charts at no cost
 - **FusionGadgets**
 - **FusionMaps**
 - PowerCharts

FusionGadgets & FusionMaps

- FusionGadgets
 - Gantt Chart
 - Gauges
 - Pyramid Charts
 - Process Flow Diagrams
- FusionMaps
 - International
 - United States
 - State
 - County

Pricing

	User	Developer	Professional	Site	Enterprise
FusionCharts	\$69	\$199	\$499	\$999	\$1999
FusionGadgets	N/A	\$199	\$499	\$999	\$1999
FusionMaps	N/A	\$199	\$499	\$999	\$1999
PowerCharts	N/A	\$199	\$499	\$999	\$1999

Pricing

- **User**
 - Designers or general users who want to plot charts for static web pages and PowerPoint Presentations
- **Developer**
 - Web Developers and Flash Developers who want to include dynamic flash charting capability in their web applications.
- **Professional**
 - Professionals and freelancers who develop multiple client web applications.
- **Site**
 - Hosted ASP/ Subscription based sites, which use FusionCharts for their commercial modules.
- **Enterprise**
 - Businesses having multiple developers and websites.

APEX Charts vs. FusionCharts

	APEX	FusionCharts
Declarative	✓	
Requires PL/SQL Code		✓
Complete UI Control		✓
AJAX Support		✓
Additional Server Req'd		
Gadgets	Only 1	✓
Maps		✓
Extra Cost		✓

How do they work?

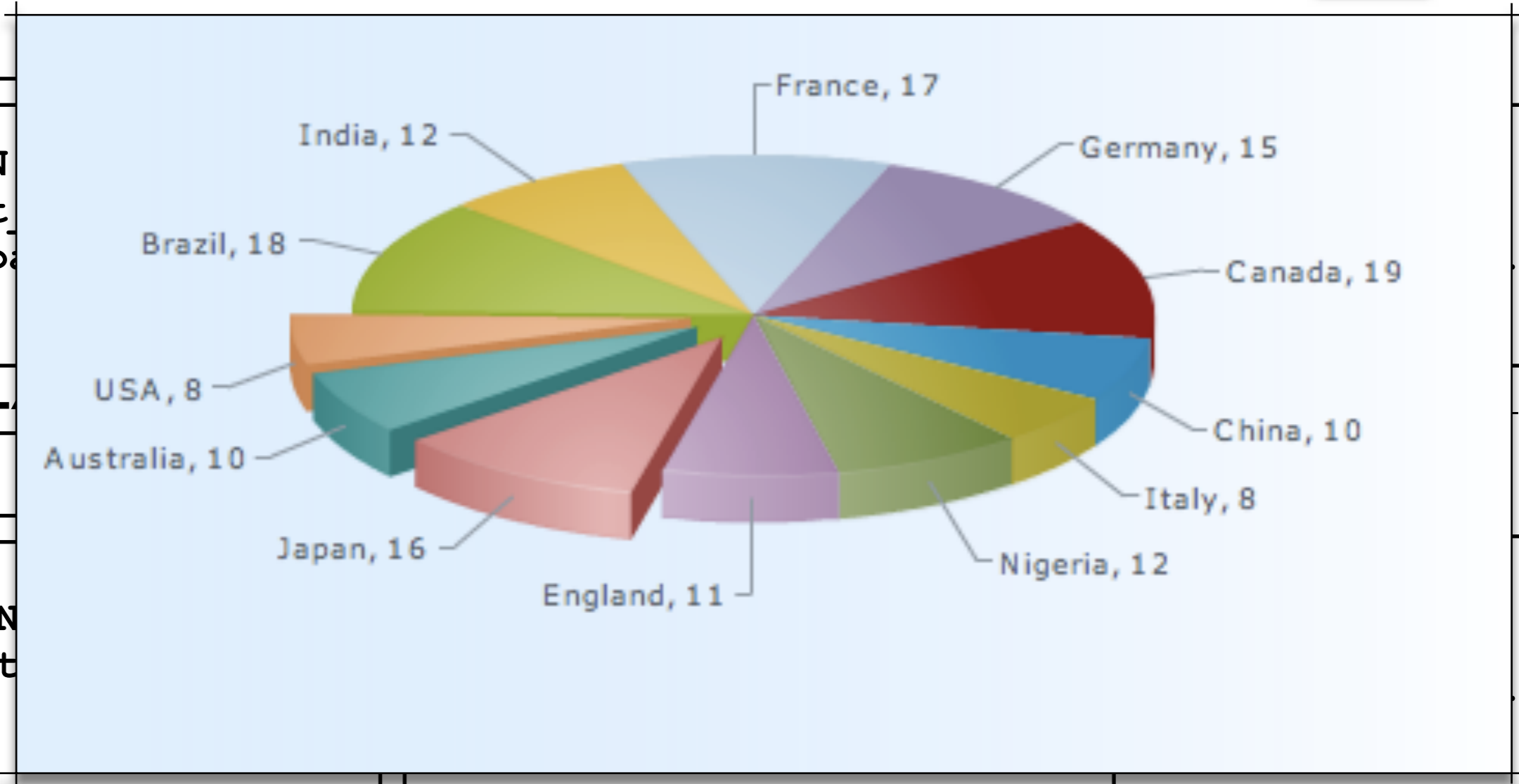
- Chart engine is Abode Flash
 - Downloaded to client via HTML/JavaScript reference
 - Can be configured with a number of different parameters
- Uses an XML file as its data source
 - Can be static or more likely, dynamically generated
 - Using PL/SQL most likely
- Simple XML file structure:

```
<chart yAxisName='Sales Figure' caption='Top 5 Sales Person'  
  numberPrefix='$' showBorder='1' imageSave='1'  
  imageSaveURL='FusionChartsSave.aspx'>  
  <set label='Alex' value='25000' />  
  <set label='Mark' value='35000' />  
  <set label='David' value='42300' />  
  <set label='Graham' value='35300' />  
  <set label='John' value='31300' />  
</chart>
```

How do they work with APEX?



```
BEGIN  
chart  
(p_p  
END;  
  
PL
```



```
..
```

region

```
..
```

```
BEGIN  
chart  
END;  
  
Before Header Process
```

PROCEDURE chart_source

DEMO

FUSION GADGETS & MAPS

PROCEDURE chart_region

```
PROCEDURE chart_region
```

```
(
```

```
p_app_id          IN VARCHAR2 DEFAULT v('APP_ID'),  
p_app_page_id     IN VARCHAR2,  
p_app_session     IN VARCHAR2 DEFAULT v('APP_SESSION'),  
p_region_width    IN VARCHAR2 DEFAULT '500',  
p_region_height   IN VARCHAR2 DEFAULT '500',  
p_chart_debug     IN VARCHAR2 DEFAULT '0'
```

```
)
```

```
AS
```

```
l_request_protocol VARCHAR2(255) :=  
  LOWER(owa_util.get_cgi_env('REQUEST_PROTOCOL'));  
l_http_host         VARCHAR2(255) := owa_util.get_cgi_env('HTTP_HOST');  
l_script_name       VARCHAR2(255) := owa_util.get_cgi_env('SCRIPT_NAME');  
l_chart_source      VARCHAR2(1000);
```

```
BEGIN
```

```
l_chart_source := l_request_protocol || '://' || l_http_host || l_script_name || '/f?p='  
  || p_app_id || ':' || p_app_page_id || ':' || p_app_session;
```

```
HTP.p('  
<div id="chartdiv" align="center">FusionGadgets</div>  
<script language="JavaScript" src="/charts/JSCClass/FusionCharts.js"></script>  
<script type="text/javascript">  
  var myChart = new FusionCharts("/charts/Charts/Gantt.swf", "myChartId", "  
    || p_region_width || '", "' || p_region_height || '", "' || p_chart_debug || '", "100");  
  myChart.setDataURL("'" || l_chart_source || "'");  
  myChart.render("chartdiv");  
</script>  
' );
```

```
END chart_region;
```

PROCEDURE chart_source

```
PROCEDURE map_source IS
```

```
l_xml VARCHAR2 (32767) ;
```

```
BEGIN
```

```
l_xml :=  
'<map borderColor="FFFFFF" connectorColor="000000" fillAlpha="70"  
  hoverColor="FFFFFF" showBevel="0">' ;
```

```
l_xml := l_xml || '<colorRange>' ;
```

```
FOR x IN (SELECT * FROM sales_ranges ORDER BY min_val)  
LOOP
```

```
l_xml := l_xml || '<color minValue="' || x.min_val  
  || '" maxValue="' || x.max_val || '" displayValue="'  
  || x.name || '" color="' || x.color || '" />' ;
```

```
END LOOP ;
```

```
l_xml := l_xml || '</colorRange>' ;
```

```
l_xml := l_xml || '<data>' ;
```

```
FOR x IN (SELECT * FROM sales_figures ORDER BY state)  
LOOP
```

```
l_xml := l_xml || '<entity id="' || x.state || '" value="'  
  || x.value || '" />' ;
```

```
END LOOP ;
```

```
l_xml := l_xml || '</data>' ;
```

PROCEDURE chart_source

```
l_xml := l_xml || '  
  <styles>  
    <definition>  
      <style type="animation" name="animX" param="_xscale"  
        start="0" duration="1" />  
      <style type="animation" name="animY" param="_yscale"  
        start="0" duration="1" />  
    </definition>  
    <application>  
      <apply toObject="PLOT" styles="animX,animY" />  
    </application>  
  </styles>  
</map>  
';  
generate_xml(p_xml => l_xml);  
END map_source;
```

PROCEDURE generate_xml

```
PROCEDURE generate_xml
  (p_xml          IN VARCHAR2)
AS
BEGIN
  -- Set the MIME header for an XML file
  owa_util.mime_header('text/xml', FALSE );

  -- Set the Content-Length so that the browser knows the file size
  http.p('Content-length: ' || LENGTH(p_xml));

  -- Set the file name of the XML file
  http.p('Content-Disposition: filename="data.xml"');

  -- Close the HTTP header
  owa_util.http_header_close;

  -- Print the XML file
  http.p(p_xml);

  -- Prevent anything else in APEX from processing
  wwv_flow.g_unrecoverable_error := TRUE;
END generate_xml;
```

PL/PDF

- Third-party product
 - <http://plpdf.com>
- PL/SQL based set of APIs that will generate PDF files
- Simple to securely integrate with APEX
- Main features
 - Generate PDF reports based on a SQL query
 - Generate line, bar and pie charts
 - Generate Barcodes
 - E-Mail PDF files
 - Store PDF Files in the database



New Features

- TrueType Font Embedding
 - Can embed your own fonts into a PDF document
 - Thus, if the user does not have the fonts installed, the document will render using the embedded fonts
- PDF Templates
 - New feature which allows you to upload a PDF files and use that as a template for your report
 - You only need to write the code for the data, not the formatting of the actual document
 - PL/PDF will merge the two to create a single PDF document

Form I-9

Department of Homeland Security
U.S. Citizenship and Immigration Services

OMB No. 1615-0047; Expires 03/31/07
Employment Eligibility Verification

Please read instructions carefully before completing this form. The instructions must be available during completion of this form. **ANTI-DISCRIMINATION NOTICE:** It is illegal to discriminate against work eligible individuals. Employers **CANNOT** specify which document(s) they will accept from an employee. The refusal to hire an individual because of a future expiration date may also constitute illegal discrimination.

Section 1. Employee Information and Verification. To be completed and signed by employee at the time employment begins.

Print Name: Last	First	Middle Initial	Maiden Name
Address (Street Name and Number)		Apt. #	Date of Birth (month/day/year)
City	State	Zip Code	Social Security #
<p>I am aware that federal law provides for imprisonment and/or fines for false statements or use of false documents in connection with the completion of this form.</p>		<p>I attest, under penalty of perjury, that I am (check one of the following):</p> <input type="checkbox"/> A citizen or national of the United States <input type="checkbox"/> A Lawful Permanent Resident (Alien #) A _____ <input type="checkbox"/> An alien authorized to work until _____ (Alien # or Admission #) _____	
Employee's Signature			Date (month/day/year)

Preparer and/or Translator Certification. (To be completed and signed if Section 1 is prepared by a person other than the employee.) I attest, under penalty of perjury, that I have assisted in the completion of this form and that to the best of my knowledge the information is true and correct.

Preparer's/Translator's Signature	Print Name
Address (Street Name and Number, City, State, Zip Code)	
Date (month/day/year)	

Section 2. Employer Review and Verification. To be completed and signed by employer. Examine one document from List A OR examine one document from List B and one from List C, as listed on the reverse of this form, and record the title, number and expiration date, if any, of the document(s).

List A	OR	List B	AND	List C
Document title: _____		_____		_____
Issuing authority: _____		_____		_____
Document #: _____		_____		_____
Expiration Date (if any): _____		_____		_____
Document #: _____		_____		_____
Expiration Date (if any): _____		_____		_____

CERTIFICATION - I attest, under penalty of perjury, that I have examined the document(s) presented by the above-named employee, that the above-listed document(s) appear to be genuine and to relate to the employee named, that the employee began employment on (month/day/year) _____ and that to the best of my knowledge the employee is eligible to work in the United States. (State employment agencies may omit the date the employee began employment.)

Signature of Employer or Authorized Representative	Print Name	Title
Business or Organization Name	Address (Street Name and Number, City, State, Zip Code)	
Date (month/day/year)		

Section 3. Updating and Reverification. To be completed and signed by employer.

A. New Name (if applicable)	B. Date of Rehire (month/day/year) (if applicable)
C. If employee's previous grant of work authorization has expired, provide the information below for the document that establishes current employment eligibility. Document Title: _____ Document #: _____ Expiration Date (if any): _____	
I attest, under penalty of perjury, that to the best of my knowledge, this employee is eligible to work in the United States, and if the employee presented document(s), the document(s) I have examined appear to be genuine and to relate to the individual.	
Signature of Employer or Authorized Representative	Date (month/day/year)

NOTE: This is the 1991 edition of the Form I-9 that has been rebranded with a current printing date to reflect the recent transition from the INS to DHS and its components.

Form I-9 (Rev. 05/31/05) Page 2

Form I-9 Filled out by PL/PDF

Department of Homeland Security
U.S. Citizenship and Immigration Services

OMB No. 1615-0047; Expires 03/31/07
Employment Eligibility Verification

Please read instructions carefully before completing this form. The instructions must be available during completion of this form. **ANTI-DISCRIMINATION NOTICE:** It is illegal to discriminate against work eligible individuals. Employers **CANNOT** specify which document(s) they will accept from an employee. The refusal to hire an individual because of a future expiration date may also constitute illegal discrimination.

Section 1. Employee Information and Verification. To be completed and signed by employee at the time employment begins.

Print Name: Last <u>lastname</u>	First <u>firstname</u>	Middle Initial <u>middleinitial</u>	Maiden Name <u>maidenname</u>
Address (Street Name and Number) <u>address</u>		Apt. # <u>apt#</u>	Date of Birth (month/day/year) <u>dateofbirth</u>
City <u>city</u>	State <u>state</u>	Zip Code <u>zipcode</u>	Social Security # <u>ssnum</u>

I am aware that federal law provides for imprisonment and/or fines for false statements or use of false documents in connection with the completion of this form.

I attest, under penalty of perjury, that I am (check one of the following):

A citizen or national of the United States

A Lawful Permanent Resident (Alien #) A aliennumber

An alien authorized to work until aliendate
(Alien # or Admission #)

Employee's Signature
employeeessig

Date (month/day/year)
signaturedate

Preparer and/or Translator Certification. (To be completed and signed if Section 1 is prepared by a person other than the employee.) I attest, under penalty of perjury, that I have assisted in the completion of this form and that to the best of my knowledge the information is true and correct.

Preparer's/Translator's Signature <u>prepareressignature</u>	Print Name <u>printname</u>
Address (Street Name and Number, City, State, Zip Code) <u>address2</u>	Date (month/day/year) <u>address2</u>

Section 2. Employer Review and Verification. To be completed and signed by employer. Examine one document from List A OR examine one document from List B and one from List C, as indicated on the form, and record the title, number and expiration date, if any, of the document(s).

List A	OR	List B	AND	List C
Document title: <u>documenttitle</u>		<u>listb1</u>		<u>listc1</u>
Issuing authority: <u>issuing_authority</u>		<u>listb2</u>		<u>listc2</u>
Document #: <u>docnumber</u>		<u>listb3</u>		<u>listc3</u>
Expiration Date (if any): <u>expirationdate</u>		<u>listb4</u>		<u>listc4</u>
Document #: <u>document2</u>				
Expiration Date (if any): <u>expirationdate2</u>				

CERTIFICATION - I attest, under penalty of perjury, that I have examined the document(s) presented by the above-named employee, that the above-listed document(s) appear to be genuine and to relate to the employee named, that the employee began employment on (month/day/year) and that to the best of my knowledge the employee is eligible to work in the United States. (State employment agencies may omit the date the employee began employment.)

Signature of Employer or Authorized Representative <u>certsignature</u>	Print Name <u>certprintname</u>	Title <u>certtitle</u>
Business or Organization Name <u>businessnameaddress</u>	Address (Street Name and Number, City, State, Zip Code) <u>businessnameaddress</u>	Date (month/day/year) <u>businessnameaddress</u>

Section 3. Updating and Reverification. To be completed and signed by employer.

A. New Name (if applicable)
newname

B. Date of Rehire (month/day/year) (if applicable)
dateofrehire

C. If employee's previous grant of work authorization has expired, provide the information below for the document that establishes current employment eligibility.
Document Title: doctitle Document #: documentnumber Expiration Date (if any): cexpirationdate

I attest, under penalty of perjury, that to the best of my knowledge, this employee is eligible to work in the United States, and if the employee presented document(s), the document(s) I have examined appear to be genuine and to relate to the individual.

Signature of Employer or Authorized Representative <u>lastsignature</u>	Date (month/day/year) <u>lstdate</u>
--	---

NOTE: This is the 1991 edition of the Form I-9 that has been rebranded with a current printing date to reflect the recent transition from the INS to DHS and its components.

Form I-9 (Rev. 05/31/05) Page 2

Pricing

	Per Oracle Database
PL/PDF	\$600

Pricing

- **Per Oracle Database**
 - PL/PDF is only priced per Oracle Database instance
 - Can be used in Unlicensed Mode
 - 5 pages of output maximum
 - PL/PDF Trial watermark on all pages

BI Publisher vs. PL/PDF

	BI Publisher	PL/PDF
Declarative	✓	
Requires PL/SQL Code		✓
Comprehensive UI Control	✓	✓
PDF	✓	✓
Word (RTF)	✓	
Excel (CSV)	✓	
J2EE Container Req'd	✓	
Save Files to DB		✓
E-Mail Files		✓
Secure APEX Integration	✓	✓
Price	\$40,000 per processor (list)	\$600 per database

DEMO

PL/PDF

PROCEDURE summary

```
PROCEDURE summary (p_project_id IN NUMBER) AS
  l_blob          BLOB;
  l_data          plpdf_chart.t_dg;
  l_sum           NUMBER := 0;
  l_start         NUMBER := 0;
  y              NUMBER := 1;
  l_project_name  projects.project_name%TYPE;
BEGIN
```

```
-- Get the Project Name
SELECT project_name INTO l_project_name FROM projects
WHERE project_id = p_project_id;
```

```
-- Initialize PL/PDF
plpdf.init(p_format => 'letter', p_orientation => 'L');
-- Create a new page
plpdf.NewPage;
-- Set the font
plpdf.SetPrintFont(p_family => 'Arial', p_size => 20);
-- Print the Title
plpdf.printCell(200,20,'Project Summary for ' || l_project_name);
-- Set the font
plpdf.SetPrintFont(p_family => 'Arial', p_size => 10);
-- Set the color of the cell background for the headers
Plpdf.SetColor4Filling(200,220,255);
-- Set the cursor to a specific X and Y coordinate
plpdf.setCurrentX(10);
plpdf.setCurrentY(25);
```

PROCEDURE summary

printCell

```
-- Pr
Plpdf -p_w number default 0: Rectangle width, if 0 then the rectangle will extend to the
Plpdf right margin
Plpdf -p_h number default 0: Rectangle height
Plpdf -p_txt varchar2 default NULL: Text
Plpdf -p_border varchar2 default '0': Rectangle border
-- Pr
Plpdf   o 0: no border
Plpdf   o 1: outline border
-- Set
Plpdf   o L: left side border
Plpdf   o T: top side border
-- Loc
Plpdf   o R: right side border
Plpdf   o B: bottom side border
FOR x   Or a combination of L, T, R and B
SELECT -p_ln number default 0: Cursor position after the cell is printed
FROM   o 0: Beside the cell
WHERE  o 1: New line
LOOP   o 2: Under the cell
-- L
Plpd   -p_align varchar2 default NULL: Text alignment
|      o L: left
|      o R: right
|      o C: center
|      o J: Force Justification
Plpd   -p_fill number default 0: Fill
Plpd   o 0: no fill
Plpd   o 1: fill with current fill color
Plpd   -p_link varchar2 default NULL: Link related to the text, can be a URL or an internal
Plpd   link ID
END LC -p_clipping number default 1: Clipping text
      o 0: no clipping
      o 1: clipping
```

PROCEDURE summary

```
-- Create the data for the pie chart
```

```
FOR x IN (  
SELECT c.category_name, count(*) cnt  
FROM tasks t, categories c  
WHERE t.category_id = c.category_id  
GROUP BY c.category_name)
```

```
LOOP
```

```
l_data(y).value := x.cnt;  
l_data(y).label := x.category_name;  
l_data(y).color := plpdf_util.set_color(y+200,y*20,y*30);  
y := y + 1;
```

```
END LOOP;
```

```
-- Initialize the chart
```

```
plpdf_chart.Init(  
p_x => 175,  
p_y => 25,  
p_w => 100,  
p_h => 100,  
p_title_position => 'B',  
p_frame => true,  
p_margin => 2);  
-- Set the center point of the chart  
plpdf_chart.SetArea(45,45);
```

PROCEDURE summary

```
-- Determine the total of all values of the chart
```

```
FOR l_i IN 1..l_data.COUNT LOOP  
  l_sum := l_sum + l_data(l_i).value;  
END LOOP;
```

```
-- Draw each slice of the pie chart
```

```
FOR l_i IN 1..l_data.COUNT LOOP  
  plpdf_chart.DrawSector(  
    p_r => 30,  
    p_a => l_start * 360,  
    p_b => (l_start + l_data(l_i).value / l_sum) * 360,  
    p_style => 'F',  
    p_legend_position => 'O',  
    p_legend => l_data(l_i).label,  
    p_fc => l_data(l_i).color  
  );  
  l_start := l_start + l_data(l_i).value / l_sum;  
END LOOP;
```

This code taken from online examples

```
-- Print the chart title and separator
```

```
plpdf_chart.PrintTitle(  
  p_text => 'Task Categories',  
  p_align => 'C',  
  p_left => 'M');  
plpdf_chart.DrawTitleSeparator;
```

```
-- End the chart region
```

```
plpdf_chart.EndDiagram;
```

PROCEDURE summary

```
-- Send the contents of the PDF document to l_blob
plpdf.SendDoc(l_blob);

-- Open the HTTP Header and set the type to PDF
owa_util.mime_header('application/pdf',false);

-- Set the file name of the XML file
http.p('Content-Disposition: filename="project_summary.pdf"');

-- Set the Content-Length so that the browser knows the file size
http.p('Content-Length: ' || dbms_lob.getlength(l_blob));

-- Close the HTTP Header
owa_util.http_header_close;

-- Send the BLOB to the browser
wpg_docload.download_file(l_blob);

end summary;
```

Summary

- Third Party “Black Box” components can save you time and money
 - Initial price of all components is relatively inexpensive
- Can also easily provide functionality that APEX lacks
- Add that level of “sizzle” to your applications that will impress management and your users alike
 - All without learning any significant Ajax, XML or JavaScript

Q & A

Sumner

TECHNOLOGIES, LLC

<http://sumnertech.com>