

ADVANCED CHARTING IN APPLICATION EXPRESS

*Dimitri Gielis
Apex Evangelists*

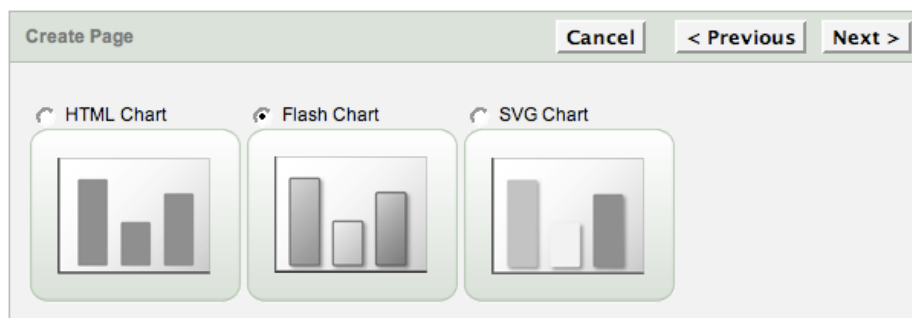
PURPOSE OF THIS PAPER

More and more people prefer to have a graphical representation about their data. A picture shows more than thousand words, some say... So which possibilities do you have in Oracle Application Express?

This paper will discuss the charting implementation in APEX 3.0.1. How it works, how to create a chart and how to extend it. HTML and SVG charts are not covered in this presentation as I believe the future is for Flash Charts.

CREATING A FLASH CHART IN APEX

Let's say we want to create a chart in our application that shows which pages of that application are most accessed. We start by creating a new Page with a Flash Chart region: Create Page > Chart > Flash Chart



Give the Page and Region a name and if needed assign a tab to the page.

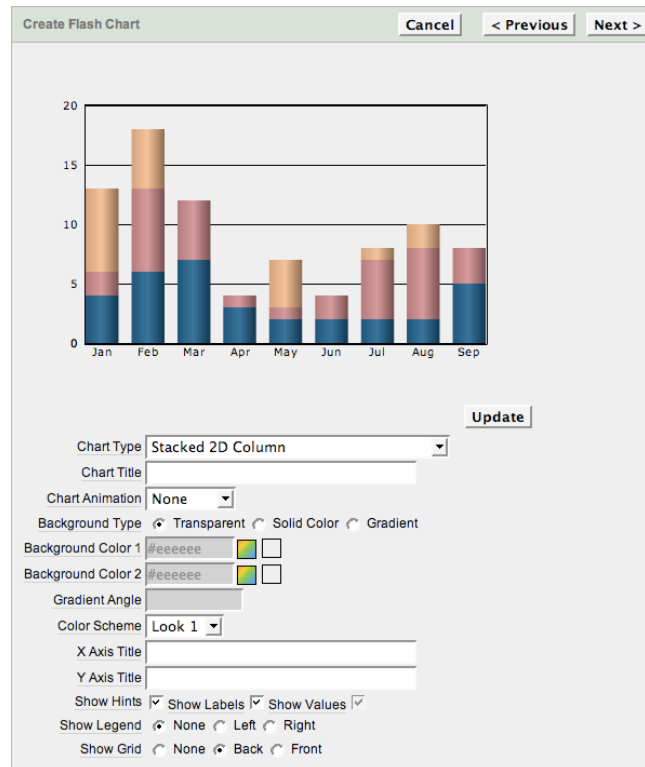
When the wizard shows the "Create Flash Chart" screen, there're different choices of charts. By selecting another Chart Type in the select list and hitting the Update button you see immediately how such a chart looks like.

Select Chart Type: Stacked 2D Column

It's also possible to customize the chart with

- a title
- an animation
- color and background color
- X and Y axis titles
- hints, labels and values
- a legend and a grid

See a screenshot on the next page



Next to do, adding a Serie to the Chart. A Serie contains the data to create the chart and if formed by a SQL Query or PL/SQL function returning a SQL Query. Eventually you can use the Query Builder to help you creating the correct select statement. At the end of the page there're also some examples which might be useful to you as depending the chart type the query can be different.

Create Region Cancel < Previous Next > Create Region

Page: 5 - Chart 5
Region Title: e

* Enter SQL Query or PL/SQL function returning a SQL Query:

```
select null as link, application_name as label, count(*) as clicks
from apex_workspace_activity_log
group by application_name
order by application_name
```

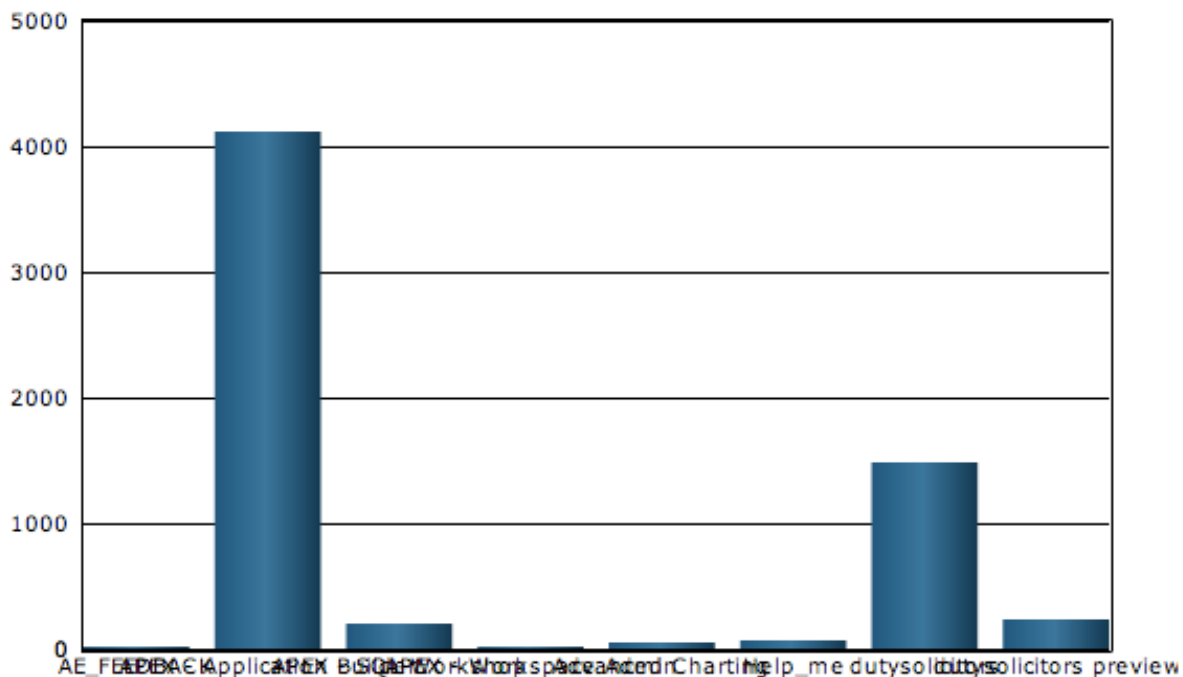
Build Query

When No Data Found Message
No data found.

➤ **Chart Query Example for Stacked 2D Column**

➤ **Items**

The result can be something like the below chart



The above chart isn't that nice, is it? How we make that chart look better further one, but first how APEX generates the chart.

HOW CHARTING WORKS IN APEX

In APEX 3.0.1 the Flash charts are based on Anychart 3. Anychart is a third party product integration in Oracle Application Express. At the moment of writing, Anychart 4.2 has already seen the light and an upgrade package for APEX is available on the Anychart website (<http://www.anychart.com>). However I wouldn't suggest you to upgrade the version of Anychart in an APEX environment, as Oracle might not support that configuration anymore (until it's officially in the product).

Future versions of Oracle Application Express will probably include Anychart 4.

Where do we find the Anychart files? On the server where APEX is installed you'll find the directory `$APEX_HOME/images/flashchart/` with an swf-file for every chart type.

A chart exists roughly out of two parts: a swf-file and data in xml format. The swf-files will use the xml data to draw the graph. Depending which swf-file is used the graph will look like a bar chart, a donut, a stacked bar etc.

We also call the swf-file a Flash file. The extension of (a compiled) Adobe/Macromedia Flash is swf. Anychart, and following out of that, APEX, uses Flash to show the charts.

To understand what's happening, we'll have a look at the source code of the chart of the above APEX page:

```
<object classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
  codebase="http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=6,0,0,0"
  width="600"
  height="400"
  id="c805338521098862158"
  align="">
<param name="movie" value="/i/flashchart/Stacked2DColumn.swf?XMLFile=http://apex.shellprompt.net/pls/apex/apex_util.flash?p=1100:5:121584562325720:FLOW_FLASH_CHART_R805338521098862158_en-us">
<param name="quality" value="high">
<param name="allowScriptAccess" value="sameDomain">
<param name="allowNetworking" value="all">
<param name="scale" value="noscale">
<param name="wmode" value="transparent">
<param name="FlashVars" value="waiting=Loading data. Please wait.&loading=Loading data...">
<embed src="/i/flashchart/Stacked2DColumn.swf?XMLFile=http://apex.shellprompt.net/pls/apex/apex_util.flash?p=1100:5:121584562325720:FLOW_FLASH_CHART_R805338521098862158_en-us"
  quality="high"
  width="600"
  height="400"
  name="c805338521098862158"
  scale="noscale"
  align=""
  allowScriptAccess="sameDomain"
  allowNetworking="all"
  type="application/x-shockwave-flash"
  pluginspage="http://www.macromedia.com/go/getflashplayer"
  wmode="transparent"
  FlashVars="waiting=Loading data. Please wait.&loading=Loading data...">
</embed>
</object>
<script type="text/javascript" language="javascript">
function chart_r805338521098862158_InitRefresh(pNow) {
  setTimeout("chart_r805338521098862158_InitRefresh(true)",10000);
  if (pNow){apex_RefreshChart (5, '805338521098862158', 'en-us');}
}
apex_SWFFormFix('c805338521098862158');
</script>
```

The first part (< object ...) of the source is needed so the browser understands a Adobe/Macromedia Flash object is included. If your browser doesn't have the Flash plugin installed, it will provide you with an option to download and install it. Adobe states that 97% of the browsers has a Flash plugin enabled.

Next we see some parameters so the swf file (the Anychart) knows how it needs to behave and look like.

The most important part is the embed-tag:

```
<embed src="/i/flashchart/Stacked2DColumn.swf?XMLFile=http://apex.shellprompt.net/pls/apex/apex_util.flash?p=1100:5:121584562325720:FLOW_FLASH_CHART_R805338521098862158_en-us"
  width="600"
  height="400"
  name="c805338521098862158"
  type="application/x-shockwave-flash"
  pluginspage="http://www.macromedia.com/go/getflashplayer"
  ...
</embed>
```

Next to the look and feel of the chart, it tells the swf-file where the data is coming from. In the example, the Stacked2DColumn.swf will be used (hence we picked a Stacked 2D Column Bar chart), the xml data will come from a typical APEX url, in fact the xml data will be generated on the fly. Application Express has a package called “apex_util” (which is a synonym for htmldb_util) and uses the procedure “flash” to provide the xml data.

To see the xml data, you can take the url http://apex.shellprompt.net/pls/apex/apex_util.flash?p=1100:5:121584562325720:FLOW_FLASH_CHART_R805338521098862158_en-us and run it from your browser.

The result is a xml data stream. If you run it for ex. from Firefox you’ll see the xml data in a more formatted way:

```
- <root>
- <type>
- <chart type="Stacked 2DColumn" minimum_value="0" maximum_value="5000">
  <animation enabled="no"/>
  - <hints auto_size="yes">
    <text>{NAME}, {VALUE}</text>
    <font type="Verdana" size="10" color="0x"/>
  </hints>
  <names show="no"/>
  <values show="no" prefix=" " postfix=" " decimal_separator="." decimal_places="0"/>
  <arguments show="no"/>
  - <column_chart column_space="3" block_space="12">
    <border enabled="no"/>
    - <block_names enabled="yes" placement="chart" position="bottom">
      <font type="Verdana" size="10" color="0x"/>
    </block_names>
    - <background type="gradient">
      + <alphas></alphas>
      + <ratios></ratios>
    </background>
  </column_chart>
</chart>
- <workspace>
  <background enabled="yes" type="solid" color="0xffffffff" alpha="0"/>
  <base_area enabled="no"/>
  - <chart_area enabled="yes" x="60" y="50" width="490" height="300" deep="0">
    <background enabled="no"/>
    <border enabled="yes" size="1"/>
  </chart_area>
  - <grid>
    - <values lines_step="1000">
      - <captions>
        <font type="Verdana" size="10" color="0x"/>
      </captions>
    </values>
  </grid>
</workspace>
<legend enabled="no"/>
</type>
- <data>
- <block name="AE_FEEDBACK">
  - <set name="Clicks" value="23" color="0x005381">
    + <background type="gradient"></background>
  </set>
</block>
```

The tree exists out of a root element with two big parts: type and data.

The type element describes the chart type and how the chart needs to look like inside.

The data element contains blocks of data that will be used to draw the chart.

A complete XML reference for the Flash Charting Component used by Anychart:

- version 3: <http://3.anychart.com/products/docs/anychart/index.htm>
- version 4: <http://www.anychart.com/products/anychart/docs/xmlReference/index.html>

CUSTOMIZING A CHART IN APEX

In the above example we created a Stacked 2D Column bar chart, but it didn't look that great. We could hardly read the data, the colors were not that nice, we didn't have a legend etc.

Let's edit that chart. In the page, locate the Chart Region.

The Region Definition looks pretty normal, nothing special here. The Region Source contains the code to include the swf-file:

```
... <embed src="#IMAGE_PREFIX#flashchart/#CHART_TYPE#.swf?XMLFile=#HOST#apex_util.flash?p=&APP_ID.:5:&APP_SESSION.:FLOW_FLASH_CHART_R#REGION_ID#" ...
```

The most important parts are made dynamic (the words between # #) as we chose that in the Chart Attributes.

!! TIP : Some people have problems seeing the chart (in hosted environments). Depending the configuration of the APEX installation you'll need to replace the #HOST# by the actual url (for ex. <http://www.apexblogs.info>)

The "Chart Attributes" is the most important screen in APEX to customize your chart.

It allows you to specify the type of chart, the look and feel and most important the data that needs to be used. Every select is called a "Serie". You can have more than one serie in a chart, however the look and feel of all the series are the same. It's not possible to include a line and bar serie in one chart with this implementation of Anycharts 3 in APEX.

Chart Settings

Chart Type: Stacked 2D Column

Chart Title: Page Views by Application

* Chart Width: 600 * Chart Height: 400

Chart Margin: Top 10 Bottom 60 Left 5 Right 5

Color Scheme: Custom

Custom Colors: #009933

Chart Series Add Series >

Series Name	Query
<input checked="" type="checkbox"/> PageViewByApp	select null as link, application_name as label, count(*) as clicks from apex_workspace_activity_log group by application_name order by application_name

Display Settings

Animation: None

Background Type: Transparent Solid Color Gradient

Background Color 1: #CCFFCC

Background Color 2: #FFFFFF

Show Legend: None Left Right

Show Grid: None Back Front

Show Hints: Show Labels Show Values

Gradient Angle: 90 Degrees

Labels Rotation: 45 Degrees

Values Rotation: Degrees

Omit Label Interval:

Following previous screen

Axes Settings

X Axis Title:

Y Axis Title: Rotate Title

Y Axis: Min Max Grid Spacing

Y Axis: Prefix Postfix Show Group Separator Yes No Decimal Places

Font Settings

	Font Face	Font Size	Font Color
Labels	Verdana	10	#006600
Values	Verdana	10	#000000
Hints	Verdana	10	#000000
Legend	Verdana	10	#000000
Grid Label	Verdana	10	#009933
Chart Title	Verdana	14	#006600
X Axis Title	Verdana	14	#000000
Y Axis Title	Verdana	14	#000000

Chart XML

Use Custom XML: No Yes

Custom XML

```
<?xml version = "1.0" encoding="utf-8" standalone = "yes"?>
<root>
  <type>
    <chart type="Stacked 2DColumn" minimum_value="0" maximum_value="5000">
      <animation enabled="no"/>
      <hints auto_size="yes">
        <text><![CDATA[{NAME}, {VALUE}]]></text>
        <font type="Verdana" size="10" color="0x000000" />
      </hints>
      <names show="no"/>
      <values show="no" prefix="" postfix="" decimal_separator="." />
    </chart>
  </type>
</root>
```

The Chart XML is very interesting. Out of the box APEX provides you some functionalities, but if you want to get more out of your chart you can create a “Custom XML”. If you select Yes for Custom XML the Display, Axes and Font Settings will disappear from the screen and the Custom XML field will become editable and will include all changes you made through the standard APEX interface.

The APEX interface doesn’t let you change the color of the grid lines for ex. we would like to have a green lines instead of the black ones. To do that, go to the Custom XML field and search the <grid>-element in the xml and change it like:

```
<grid>
  <values>
    <lines size='1' color='0x15771A' alpha='100'/>
  </values>
</grid>
```

Another example is to include a Copyright text in the chart and our logo.

The following code adds two objects, a text area and an image. The result can be seen a bit further.

```
<objects>
  <text text="Copyright © 2008" x="250" y="3" auto_size="yes" width="600" height="25" url="">
    <font size="9" type="Verdana" color="0x000000" bold="" italic="" underline="" align="center" />
  </text>
  <image source="#WORKSPACE_IMAGES#ae_logo.png" x="490" y="5" alpha="100" url="http://apex-evangelists.com" width="100" height="30" />
</objects>
```

You can change almost any settings of the charts, but not the data section.

The complete XML reference for the Flash Charting Component used by Anychart:

- version 3: <http://3.anychart.com/products/docs/anychart/index.htm>
- version 4: <http://www.anychart.com/products/anychart/docs/xmlReference/index.html>

The last section of Chart Attributes

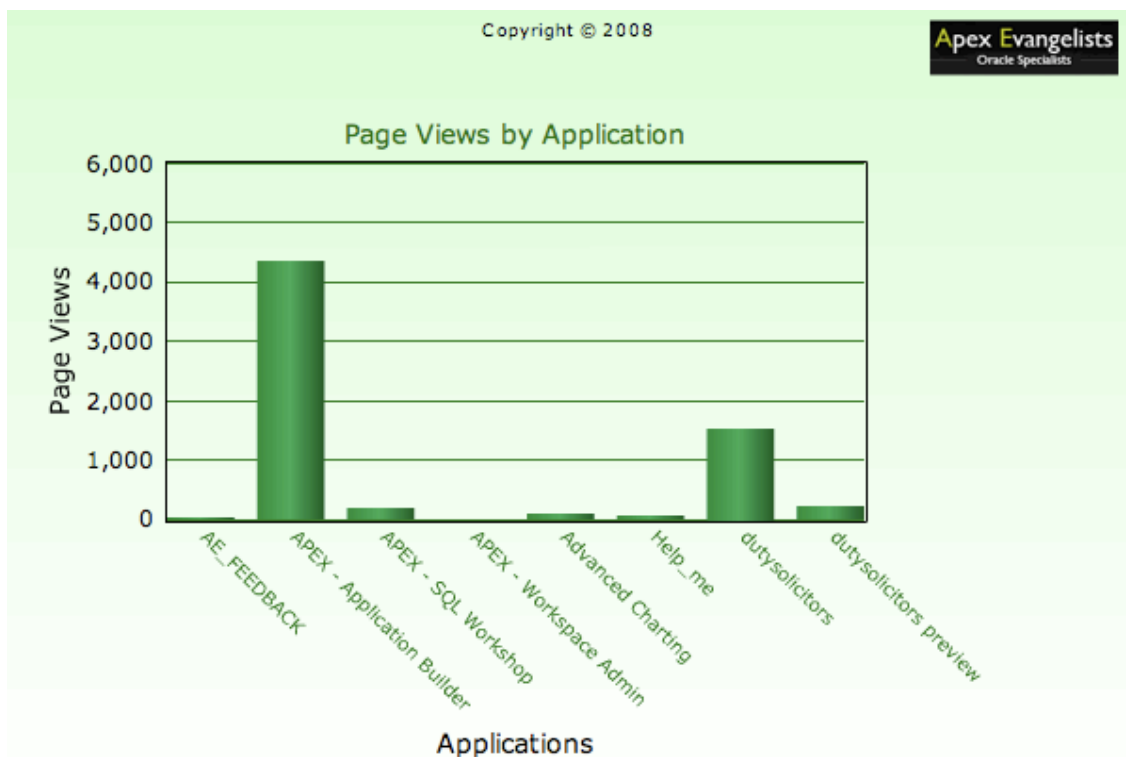
Refresh

Asynchronous Update Update Interval (Seconds)

The Asynchronous update feature allows you to refresh your chart automatically after a number of seconds. This makes use of AJAX and so doesn't refreshes the whole page again, instead it just refreshes the data of the chart.

!! TIP : If you click on the labels of the items in the above screen you'll get more info about that particular feature.

The same chart as before but now first a bit customized in the Chart Attributes section in APEX and later on modified in the Custom XML section.



GETTING MORE OUT OF CHARTS

If we want to create some advanced charts, the build-in chart functionality of APEX are not enough. In my view you've following options to nevertheless create more advanced charts:

- wait till the new version of Anychart is included in APEX (3.1 or 4.0 ?)
- upgrade Anychart 3 in APEX to 4 with the provided upgrade package of Anychart.
- integrate a third party charting in APEX next to the existing one.

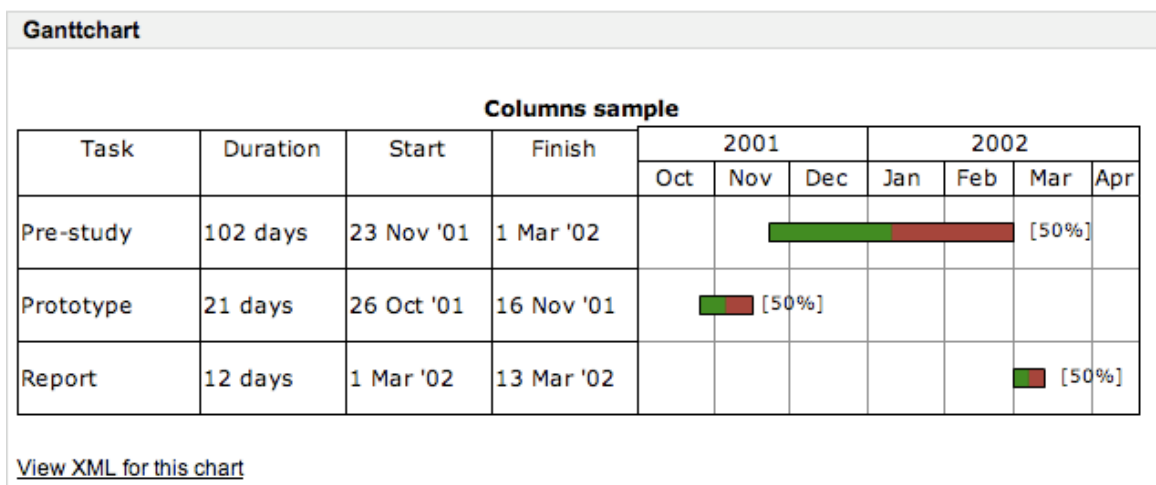
I guess for most people waiting for the new version is not an option. Upgrading could be a way to go, however this is not free of charge and it might be not supported by Oracle. So I only see one possibility and that's using another 3rd party charting tool like Anychart 4 (the new version with Gantt charts), Fusion Charts or Maani XML/SWF Charts to name a few.

The integration exists out of two parts: installing the third party charts, integrate it with APEX and write a package to provide the chart with XML data. Denes Kubicek (<http://deneskubicek.blogspot.com>) wrote a package to help you generate the xml needed by the Maani chart in a friendly way.

Some more advanced charts:

- a Gantt chart
- a drill down chart
- a chart with 2D column bars and a Line chart in one
- a dashboard

The principle of creating these advanced charts is in all cases the same. We show below how to create a Gantt chart in APEX.



This implementation is shown into the demo.

OTHER HINTS & TIPS

There're also other 3rd party charting facilities. Next to Anychart, Fusion Charts (or the new Fusion Gadgets) are also widely used to create nice looking charts and dashboards.

CONCLUSION

This paper outlines how Oracle integrated Anychart in Application Express. The paper shows you how you can create a Flash chart in APEX and how to enhance it. To get more out of the charting functionalities in APEX I briefly touch the integration of the new Anychart version which offers more possibilities for your charts and dashboards.

ABOUT THE AUTHOR

Dimitri has worked in the past as a consultant for Oracle Belgium for over 4 years, where he became involved with almost every Oracle product, his main area of expertise being with the database. He then moved to a smaller company to create an Oracle team and to concentrate on pre-sales and some high end Oracle projects. A few years later he started as Business Unit Manager Oracle at Ordina Belgium. In 2008 he took the decision to follow his heart and concentrate on Apex Evangelists (<http://apex-evangelists.com>), a company he founded in 2007 together with John Scott and which specializes in Oracle Application Express.

The author first began using the Application Express product in 2003, back then it was known as Project Marvel and later became known as HTMLDB and then Application Express. He shares his knowledge and experiences with Application Express through his blog (<http://dgielis.blogspot.com>), he wrote the "DG Tournament" application to show APEX can also be fun and he's the guy behind apexblogs.info, which aggregates all APEX blogs in one place. Dimitri participates in the APEX community and Oracle User Groups and is Vice President of the Application Express Special Interests Group.

APEX AND CHARTS RESOURCES

You can find more info on following sites:

- Oracle Application Express Homepage: <http://apex.oracle.com>
- Dimitri Gielis Blog: <http://dgielis.blogspot.com>
- Apex Evangelists website: <http://apex-evangelists.com>
- Anychart website: <http://www.anychart.com/home/>
- Fusion Charts: <http://www.fusioncharts.com/>
- Fusion Gadgets: <http://www.fusioncharts.com/gadgets/Gallery.asp>
- Maani XML/SWF Charts: http://www.maani.us/xml_charts/