

APEX INTERACTIVE REPORTS: THE GOOD, THE BAD AND THE UGLY

*Karen Cannell, Principal,
Integra Technology Consulting*

INTRODUCTION

Oracle Application Express Interactive Reports (APEX IR) offer amazing functionality for both developers and end users, but this wealth of features comes with limitations. This paper discusses the Good, the Bad and the Ugly of APEX IR. The “Good” is the wealth of built-in features for developers and end users. The “Bad” refers to the limitations of those features. The “Ugly” refers to the default APEX IR appearance.

The good aspects of APEX IR are widely publicized and well documented: APEX IR was the major feature of the APEX 3.1 release and is the topic of numerous Oracle whitepapers and Oracle-By-Example features. The bad aspects, the limitations, are documented, but not well publicized. The appearance is rarely if ever mentioned. One can assume that most applications just settle for the default appearance. Accordingly, this paper includes a brief review the “Good” (the wealth of built-in features), then devotes more attention to and suggests workarounds for the “Bad” (the limitations) and the “Ugly” (the appearance).

As with all things APEX, with a bit of planning and design it is possible for developers to work around any out-of-the-box “Bad” or “Ugly” to deliver all “Good” for the end users.

THE GOOD

The obvious and notable “Good” of APEX IR is the wealth of built-in features. A standard APEX IR includes an impressive list of end user features – column selection and ordering, advanced search capability, filters, highlighting, sorting, breaks, aggregates, charting, computations and flashback - all delivered with literally a handful of developer keystrokes. To manually code even a subset of these features on a classic APEX report would require significant developer time and significant developer knowledge of SQL, HTML, CSS, Javascript and AJAX. The use of APEX IR enables developers to deliver advanced functionality to the end user without investment in advanced skill sets.

The following section gives a brief description of the major APEX IR features. Refer to the APEX on-line documentation and the APEX Oracle by Example articles (see the References section for links) for more complete information on APEX IR features.

APEX IR ANATOMY

A high-level IR anatomy lesson is in order. The differentiator between a classic APEX report and an interactive report is the enhanced toolbar. The APEX IR toolbar is comprised of a Search Bar and an Action Menu. The Search Bar, the left portion of the toolbar, provides built-in search capability, by single columns (with type checking) and across all columns. A drop-down menu in the Search bar allows the user to search by column name. The Actions Menu, accessed by the gear pull-down on the right side of the toolbar, provides a list of actions that a user may perform on the data set. A Control Panel region appears when needed to enable selections specific to the selected action. A Control Summary region displays action-specific options that are currently in effect. Column sorting and filtering is achieved through an interactive toolbox that opens adjacent to each column heading upon focus on the column heading. Small arrow icons on the column heading visually indicate any sorting in effect. If enabled, a page icon provides a link to a single row view of the data, or a custom URL or page setting. Figures 1 and 2 illustrate the major APEX IR components.

GOOD USER FEATURES

This section provides a brief description of the major “Good” APEX IR features. Refer to the APEX on-line documentation and the APEX Oracle by Example articles (see the References section for links) for more complete information on APEX IR features.

The highlights of the APEX IR “Good” features are:

In the Search bar:



IMPROVED SEARCH CAPABILITY

Allows a user to search by specific columns or by all columns. The user does not need to be aware of data type – number or character – the Search interface takes care of implicit conversions where necessary.

In the Actions Menu:



SELECT AND ORDER COLUMNS

Allows a user to select and order the displayed columns



FILTERING

Allows a user to define simple or complex filters on selected columns.



SIMPLIFIED COLUMN SORTING

Allows a user to sort the selected columns by a selected order. A user may also sort by a double-click on the column heading, which displays column-specific sorting options.



IMPROVED BREAK LEVELS AND BREAK FORMATTING

Allows a user to define break up to five columns, and to easily format break headings. Try that in a classic APEX report!



HIGHLIGHT

Allows a user to highlight rows or specific columns according to a defined filter.



AGGREGATES

Allows a user to select column(s) to declaratively aggregate and the function to apply: SUM, COUNT, AVERAGE, etc.



COMPUTE

Allows a user to build a new column as a computation involving one or more other columns, as one may build for a standard deviation, a percentage discount or a salary bonus.



CHARTING

Allows users to define and view data in a chart: horizontal or vertical bar, pie or line chart. Once created, the user may switch between the data view and the chart view.



FLASHBACK

Allows a user to view the data as it was at a point in time up to 3 hours previous (depending on database settings).



SAVE REPORT

Allows users to save a report in its current configuration. The saved report appears in a new tab. All defined filters, highlights, breaks, sorting, etc are preserved in the saved report. All saved reports will be there when the user logs back in.



HELP

Displays help information for each of the actions in the Action Menu.



RESET

Clears all APEX IR Action and Search settings, restoring the report to the default display. Users may also temporarily disable individual customization settings by unchecking the checkbox next to setting, or may remove it by clicking on the red X.



DOWNLOAD

Allows a user to download the data in one of several data formats: Excel, Word, HTML or PDF, as defined by the developer.

GOOD USER CONTROL

With APEX IR, the end user is in control of how the data appears. For some users, all this can be overwhelming. We will discuss that later. For those ready to take on these features, the simple freedom from repeated data requests for a new column order, sort or break format, or from massaging data in Excel is welcome. Features such as charting, computation and flashback open a whole new territory in reporting customizations. It is now possible for end users to be active data consumers rather than passive data recipients. This is good!

Figure 3 – Developer Control: APEX IR Report Attributes, Search Bar Developer Settings

GOOD DEVELOPER FEATURES

Developers can deliver all that dynamic, user-controlled functionality in a few keystrokes: creation of an APEX IR is wizard-driven and requires no more effort than creation of a classic APEX report. A developer does not need to learn advanced HTML, CSS, JavaScript or AJAX, or anything more than how to build a query, and even that can be done declaratively using the Query Builder. With APEX IR, more developers can deliver greater functionality with minimal investment. This means SQL-savvy end users could perhaps be building their own applications, freeing the development team for more complex tasks. Developers may also control the breadth of features available to users, clear the IR cache, and reference IR columns by name.

GOOD DEVELOPER CONTROL

A developer may control the APEX IR features by turn various IR Search and Action Menu features on and off through the IR Builder interface, the Report Attributes tab, in the Search Bar element, as in Figure 4. This developer declaration applies *for all users, all the time*. It is not currently possible to declaratively control which users have access to which toolbar features; it is one-size-fits-all.

In addition to the Report Attribute, Search Bar settings, there are a few more important APEX IR features for developers for clearing and referencing columns of an APEX IR

The screenshot shows the 'Create Button' dialog box. At the top, it says 'Set: none'. Below that, it indicates 'Page: 2 - Customers' and 'Region: Customers'. The 'Button Name' is 'RESET'. The 'Target is a' dropdown is set to 'Page in this Application'. The 'Page' is set to '2'. There is a checked box for 'reset pagination for this page'. The 'Request' field is empty. The 'Clear Cache' field contains '2,RIR', with a blue arrow pointing to it from the text 'Reset Interactive Report'. Below that are fields for 'Set these items' and 'With these values', both empty. At the bottom, there is a 'URL Target' field, also empty. Navigation buttons include 'Cancel', '< Previous', 'Next >', and 'Create Button'.

Figure 4 – APEX IR Clear IR Cache Setting

CLEAR IR CACHE

In cases where the Action Menu is not displayed, as in the above example, the user often needs a way to reset the interactive report. A simple Clear Cache setting on the page does not reset the APEX IR. To specifically reset an APEX IR, one needs to a clear cache request of “**RIR**”, for **Reset Interactive Report**. For example, to reset the APEX IR on page 2, the Clear Cache setting would be 2,RIR, as in Figure 5. A Reset directive restores the interactive report to its default settings.

Similarly, to clear ALL filters and settings for an interactive report, use the “**CIR**”, for **Clear Interactive Report**. The CIR directive clears *all* filter and control settings. After a Clear, there is no automatic way for the user to return to the default report settings short of logging out then back in, unless the developer builds one in.

REFERENCE IR COLUMNS AND FILTER VALUES

A developer can define declarative filters on an APEX IR by passing filter criteria using the URL item values. This allows for passing filter criteria from one page to another, and for building drill downs from a report to an interactive report. To reference an APEX IR report column, use IR_<column_name>.

To define and reference filter boundaries for an APEX IR, such as “Less than <value>” or “Greater Than <value>”, use these references:

- IREQ_<column_alias> Equals (this is the default) same as IR_<column_alias>
- LT_<column_alias> Less than
- IRLTE_<column_alias> Less than or equal to
- IRGT_<column_alias> Greater Than
- IRGTE_<column_alias> Greater than or equal to
- IRLIKE_<column_alias> SQL Like operator
- IRN_<column_alias> Null
- IRNN_<column_alias> Not Null
- IRC_<column_alias> Contains
- IRNC_<column_alias> Not Contains

This ability to directly reference the APEX IR columns and filter parameters by name makes it simple to control the APEX IR data set in drill downs, or in popup pages. This ability to reference the APEX IR column by name is a powerful tool.

To summarize the “Good” of APEX IR: they save developer time (budget), offer end users increased functionality (user acceptance), and ultimately make managers look good when their team delivers more for less (keep the boss happy!). So what’s the catch?

THE BAD

With all that “Good” comes a bit of “Bad”. There are a few limitations. Some are important for developers. Others affect both developers and end users. Others are less-known and perhaps of less impact. Education is the key. “Know thy limitations,” if you will. For developers, this means understanding the APEX IR limitations in order to work around them to meet requirements. For end users, this means knowing the limitations in order to avoid unexpected results or error messages. Note that these limitations are not always clearly defined in the Help interface – end users must be educated to these limitations if there is any chance they may be encountered. For this reason, developers should always know (or at least have some idea) how the end users use their data, and specifically how they use the APEX IR features.

BAD FOR DEVELOPERS

The most important limitations for developers to master are that only one APEX IR is allowed per page, and, ironically, how to limit all that end user functionality.

ONE PER PAGE

The most significant limitation is one APEX IR per page. The APEX Known Limitations documentation states: “You can only have one Interactive Report display on a page at any given time.” The word “display” here is misleading. It implies that one could have multiple IR’s on a single page, conditionally displaying one at a time. In fact, APEX will not let you create more than one IR on a page. This limitation is due to how the IR processes work, similar to the limitation of one set of data manipulation processes per page.

Sample Application

ADMIN [Print](#) [Logout](#)

Home Customers **Customers & Orders** Products Orders Charts Admin

Rows 15 Go [Create Customer >](#)

Customer Name	Address	City	State	ZIP Code
<input checked="" type="checkbox"/> Bradley, Eugene	Schoephoester Road	Windsor Locks	CT	06096
<input checked="" type="checkbox"/> Dulles, John	45020 Aviation Drive	Sterling	VA	20166
<input checked="" type="checkbox"/> Hartsfield, William	6000 North Terminal Parkway	Atlanta	GA	30320
<input checked="" type="checkbox"/> LaGuardia, Fiorello	Hangar Center, Third Floor	Flushing	NY	11371
<input checked="" type="checkbox"/> Lambert, Albert	10701 Lambert International Blvd.	St. Louis	MO	63145
<input checked="" type="checkbox"/> Logan, Edward	1 Harborside Drive	East Boston	MA	02128
<input checked="" type="checkbox"/> OHare, Edward "Butch"	10000 West OHare	Chicago	IL	60666

1 - 7

This is a 2nd IR on the same page!

Rows 5 Go

Name	Category	Avail?	List Price	Unit	Sales	# Cust	Dt Sold	Img
3.2 GHz Desktop PC	Computer	Y	1200	4	4800	2	24-MAR-2009 02:29PM	
MP3 Player	Audio	Y	199	1	199	1	19-MAR-2009 02:29PM	
Bluetooth Headset	Phones	Y	40	2	80	2	04-MAR-2009 02:29PM	
PDA Cell Phone	Phones	Y	250	3	750	3	09-MAR-2009 02:29PM	

Figure 5 – Workaround for One IR per Page - APEX IR in an Iframe

One may argue that this is just plain smart, that only so much functionality belongs on a single web page, and that an APEX IR combined with one or more classic reports will meet most requirements. However, some situations truly require more than one APEX IR per page, as in a dashboard-type interface that includes several reports with full search and action menu features. For these cases, it is possible to give the *appearance* of more than one IR on a page by using iframes.

IFRAME FOR > 1 PER PAGE

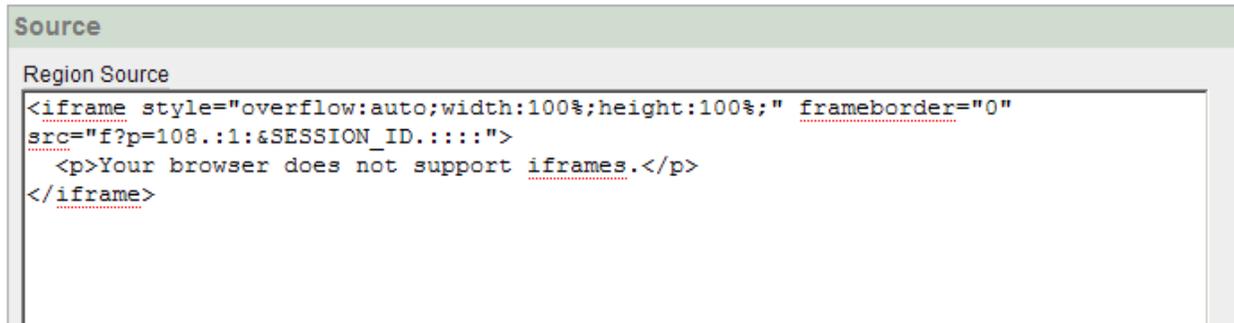
An iframe is an HTML tag that defines an inline frame that contains another document. In this case, the embedded document is another APEX page that contains an APEX IR – thus the appearance of two (or more) APEX IR on a single page. Before implementing an iframe, consider why. What is the intent of the web page? What is the level of expertise of the end users? How much functionality should be on one page? If requirements honestly mandate more than one IR on one page, consider the use of iframes .

To incorporate an APEX IR in an iframe:

1. Create a separate application that uses a plain (no-header) page template.
2. Create the APEX IR to be embedded in the plan-page template application, using the plain page template.
3. In the “parent” application, create the “main” page.
4. Create the APEX IR on the main page.

5. Create an HTML region to contain the iframe.

For the Region Source for this page, enter something like:



The screenshot shows a text area titled "Region Source" containing the following HTML code:

```
<iframe style="overflow:auto;width:100%;height:100%;" frameborder="0"
src="f?p=108.:1:&SESSION_ID.:::">
  <p>Your browser does not support iframes.</p>
</iframe>
```

Figure 5 –HTML Region, Iframe Source

where 108 is the application number of the plan-page-template application and 1 is the page of the APEX IR to be embedded.

6. Repeat as needed.

It is also possible to open an APEX IR in a separate page, by a standard popup window using `popupURL`, or using `ExtJS` (see Mark Lancaster, <http://oracleinsights.blogspot.com> and <http://apex.oracle.com/pls/otn/f?p=200801:2022:0>). Many thanks to Mark Lancaster for suggesting the iframe!

Too Much Good?

The second major limitation of APEX IR's is actually one of its strengths: all that functionality. In a web environment, for many users, more is *not* always better. There is such a thing as too much functionality, particularly for inexperienced or infrequent users.

All APEX IR features are enabled by default. For inexperienced users, this can be intimidating. For users who do not need to do anything besides enter a value to get some data back, the extra features, and extra interface elements, are unnecessary. For experienced, SQL-savvy users, all this “Good” can be dangerous. Should all users be able to build their own columns? Should all users be able to Select Columns and reorder them to suit their needs? Think of these two features in combination. For example, one could create an adjusted Sales column, then use the Select Columns option to hide the original Sales column, expose the adjusted sales column, and then publish the report. Is this a “Good” feature, or “Bad” one?

The one-size-fits-all nature of APEX IR features raises some questions:

- What if users need some of the features, some of the time, but not all features, all of the time?
- What if some users should have all the features and others only a few? For example, what if only privileged users should be able to Compute Columns, while other users should never have that feature?
- What if developers want to leverage the features in developing the report, but users should never modify the report at all?

These are all viable situations, and all are achievable, with a bit of developer planning and creativity.

GOOD FOR SOME BUT NOT ALL

To enable some Action Menu options for some users, and none or a different set for others, consider building two (or more) separate APEX IRs (separate pages), and conditionally directing users to the appropriate page based on distinct authorizations, distinct conditions or a combination of the two.

NOT GOOD AT ALL?

What if users do not want *any* of the extra functionality APEX IRs offer? One client wanted the entire Search Bar and Action Menu turned off, since those interfaces were “too complicated” and “too confusing” for their users. The developer chose an APEX IR for the improved Break feature and the convenience of Search feature, which eliminates the need to code specific number-to-character conversions into the WHERE clause..

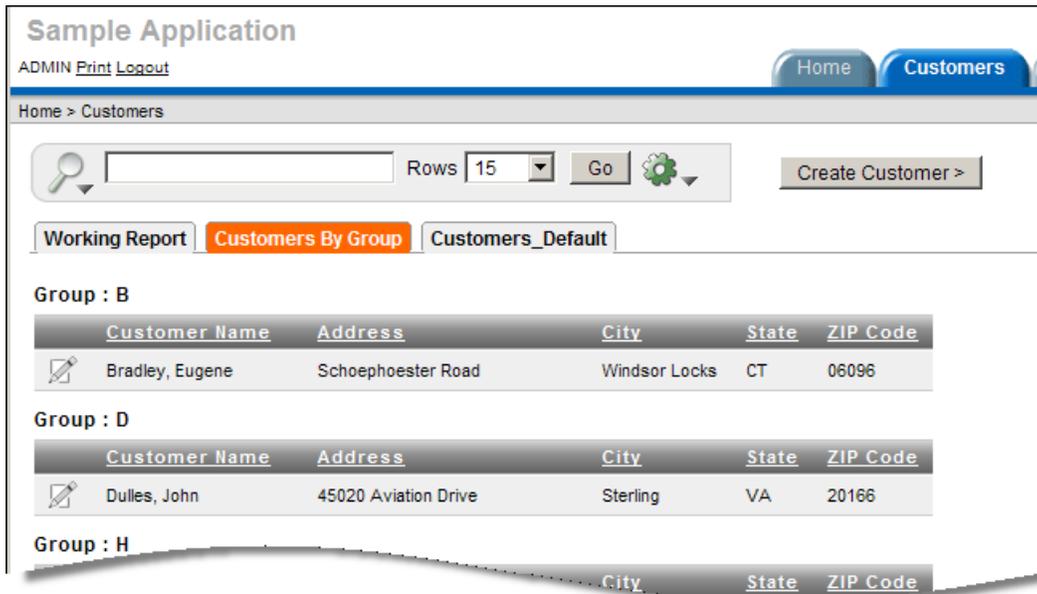


Figure 6 – APEX IR with Hidden Control Summary

In this case, the developer, per the client request, turned off the Action Menu altogether by unchecking the Report Builder, Search Element Action Menu checkbox. The developer then used CSS to turn off the Control Panel Summary by adding CSS for this element in the page header with `display:none`. This effectively hid the display of current report settings – breaks and sorts, in this case - that would normally appear below the toolbar. These settings needed to be hidden to prevent a user from removing them by clicking on the red “X”. The developer then added a separate Reset button, so users were able to clear settings in cases where a link passed a specific column value to that IR. The result looks more like a classic report with improved formatting. Behind the scenes, the report leverages the full capability of an APEX IR.

MINOR “BAD” FOR DEVELOPERS

The following limitations are important for developers to know, but are less significant than the one-per-page and one-size-fits-all feature settings. The developer needs to be aware of these issues, and plan accordingly to meet user requirements and expectations. In some cases this means developer planning, in others it means end user education.

DATABASE VERSION

APEX IRs do not work with Oracle Database 10.1.0.3. Aside from upgrading, there is not a workaround for this limitation.

SUPPORTED BROWSERS

APEX IRs only work with supported browsers: IE (6.0 or later) and Firefox (1.0 or later). If one has no control over the end user browser, this may be an issue.

UNIQUE COLUMN

The Unique Column declaration is not validated when the APEX IR is created. The developer needs to ensure the selected column is truly unique, or the user will receive an error when selecting the single column link.

For queries that contain UNIONS and GROUP BYs, such that a ROWID pseudo column is not defined, the developer must specify a unique column or disable the single-row view.

ORDER BY IN THE IR QUERY

ORDER BY in the query will not work when a single-row link is enabled. Instead, use a Sort to order columns. Remember that users will be able to see and remove the sort setting in the control summary panel (by clicking in the red 'X'), so one may need to consider whether it is possible to disable display of the control summary

32K ROW LENGTH LIMIT

The same 32K characters-per-row limit as in classic APEX reports is in place. There can be more columns in the report, but APEX can only display 32K characters in a row at a time. This includes all HTML markup, so be particularly aware of this limitation when there is embedded HTML in the report cells. If the row limit is exceeded, the user receives the "ORA-06502: PL/SQL: numeric or value error: character string buffer too small" error. Developers should be careful not to leave end users open to this error, by either limiting the report columns such that this error can never occur, or by fully educating end users of the 32K limit and how to avoid it.

HIGHLIGHTING

Developers cannot declaratively disable highlighting for a column.

BAD FOR DEVELOPERS AND USERS

The following known limitations are important for developers to be aware of, but also important for end users to be educated about. These issues are not dramatic, and in most cases, knowledge of the constraint will suffice to avoid any unexpected results or error messages.

32K ROW LENGTH LIMIT

If requirements are such that the developer is unable to ensure that the 32K row length limit can never be reached, no matter what combination of columns a user selects, users must be educated about this limit, how to avoid it, and how to calmly adjust and move on if it does occur.

DOWNLOAD FORMATS

Downloads of all types do not retain formatting settings. Data content, column order, sort settings and filters are retained, but breaks, aggregates and highlighting are not. Users need to be educated that what they see on the screen is not always what they will get in their download.

If it is essential that breaks, aggregates and highlighting are retained, the developer can build a Named Column report or XSL-FO or RTF equivalent or the desired format, and provide access to that report instead of the standard Download action. Of course, now the developer is into PDF printing, ensuring that report templates and buttons for all desired formats are provided, and the spontaneity of having the end user create and download exactly what they want is lost.

SEARCH BAR NOT DATE SAVVY

The Search Bar All Columns search is not DATE savvy but IS NUMBER savvy. For example, in the Sample Application, on the Orders page, a search on 200 will return rows, where a search on 04-MAR will not, even though rows exist where the Last Sale date is 04-MAR-09. End users need to know to use the Named Column search for DATE searches.

FILTER VALUES

IN and NOT IN filter values cannot have commas (comma is the IR delimiter).

When a single-row link is enabled, not all filter options are available. Specifically, CONTAINS, BETWEEN, IN and NOT IN are not there.

When a single-row link is enabled, the LTE option is broken: it returns GTE or GT instead of LTE.

CHARTING

The APEX IR displays the chart *or* the data set. There is no way for the end user to declaratively display both at once. If the requirement is for a chart and an APEXIR, the developer will need to build a separate chart region.

The maximum number of elements in chart is 1000 for horizontal, vertical and line charts, and 20 for pie charts. If these limits are prohibitive, consider presenting aggregated data and charting the aggregate values.

Vertical Bar Chart Headings will adjust for display purposes, such that not every element will be labeled. If this is not acceptable, use a horizontal bar chart.

Charting in general is not always intuitive. As with all charting (not just APEX IR charts), users may need to be educated on how to select data columns for displaying in the various types of charts.

AGGREGATES

When aggregating data in combination with other Actions, the behavior may not be as expected. Consider this example from the OTN APEX Forum:

<http://forums.oracle.com/forums/thread.jspa?messageID=2635534>

Result #1:

```
Region: West
*****
State County           City           Population
CA     Orange County   Irvine         100
CA     Orange County   Orange        200
CA     Los Angeles       Hollywood     300
CA     Los Angeles       Universal City 400
                                     Sum           1,000
```

Intended result:

Result #2:

```
Region: West
*****
State County           Population
CA     Orange County   300
CA     Los Angeles     700
                                     Sum           1,000
```

The actual result:

Result #3:

```
Region: West
*****
State County           Population
CA     Orange County   100
CA     Orange County   200
CA     Los Angeles     300
CA     Los Angeles     400
                                     Sum           1,000
```

The APEX IR is simply a tabular report, with many end user options to rearrange the data. By default, there is not grouping or order intelligence built in. To effect the summary adjustment desired in this example, the developer needs to add a GROUP BY clause to the IR query.

This example underlines the fact that even though APEX IR give the end user great power to format and arrange reports to their needs, the developer needs to be aware of those needs to prepare and deliver the appropriate APEX IR query to support the end user needs.

SEARCH COLUMNS

The maximum number of columns in the Column Header dropdown is 1000.

The maximum number of columns displayed in the Report Attributes table is 100.

These are hard limits; there is no workaround to either. In most cases, these limits are more than reasonable for an interactive web-based report.

HIGHLIGHTING

Cell Highlighting does not work on Theme 20 – the background color is ignored. Use Row Highlight or a blank background color. As a workaround, the developer can employ a different theme.

To summarize the Bad, some of these limitations are significant, most are relatively minor and all can be worked around with some developer planning and adequate end user education.

THE UGLY

The default style for interactive reports across all templates is a basic report template with grey background and black heading background. Let's face it, it's ugly. Ok, it's arguably ugly. In some themes, as in Theme 12, used by default in the Sample Application, this appearance is acceptable. In other themes, such as Theme 11, Round Green, it just looks out of place. In one's corporate theme, it may be downright unacceptable.

While it may be tempting to concede that for all that functionality, "we'll just live with it", you don't have to.

Remember that all APEX pages are HTML pages with styles controlled by HTML attributes and CSS style settings. The APEX IR components each have a CSS style definition that may be changed. Changes to the default APEX IR styles may be made by applying standard CSS techniques to override the defaults. Such changes may be applied to a single interactive report, to a page template to effect changes across several APEX IR's, or to all page templates of a theme to enforce a common look-and-feel for all APEX IR's in an application.

The screenshot shows an APEX Interactive Report titled 'Customers'. The page has a green header with navigation links: Home, Customers, Products, Orders, Charts, Admin. Below the header, the breadcrumb 'Home > Customers' is visible. The report includes a search bar, a 'Rows' dropdown set to 15, and a 'Go' button. A 'Create Customer >' button is also present. The report is currently displaying 'Customers By Group' under the 'Working Report' tab. A 'Saved Report = "Customers By Group"' is listed with a 'Group' checkbox checked. The main data is presented in a table with columns: Address, Customer Name, City, State, and ZIP Code. The data is grouped by letter (B, D, H, L, O). A 'Top Customers' summary table is shown on the right, listing customer names and their corresponding values.

Address	Customer Name	City	State	ZIP Code
Group : B				
Schoephoester Road	Bradley, Eugene	Windsor Locks	CT	06096
Group : D				
45020 Aviation Drive	Dulles, John	Sterling	VA	20166
Group : H				
6000 North Terminal Parkway	Hartsfield, William	Atlanta	GA	30320
Group : L				
1 Harborside Drive	Logan, Edward	East Boston	MA	02128
Hangar Center, Third Floor	LaGuardia, Fiorello	Flushing	NY	11371
10701 Lambert International Blvd.	Lambert, Albert	St. Louis	MO	63145
Group : O				
10000 West OHare	OHare, Edward "Butch"	Chicago	IL	60666

Top Customers	
LaGuardia, Fiorello	\$3,800.00
Hartsfield, William	\$2,598.00
Dulles, John	\$1,200.00
Logan, Edward	\$790.00
Bradley, Eugene	\$500.00
Lambert, Albert	\$490.00
OHare, Edward "Butch"	\$250.00

1 - 7

Figure 7 – APEX IR, the “Ugly” Default Appearance

The suggestions that follow do NOT constitute a CSS lesson; the techniques presented here are simplified to demonstrate what can be achieved through CSS. The reader is advised to review and optimize all custom CSS for your application or template and follow optimal CSS practices prior to deployment, including compression and testing in all supported browsers. See the CSS references in the References section for more information.

CUSTOMIZE IR CSS

As a general approach to all APEX CSS edits, do not touch the core APEX CSS files. Make copies, and edit the copies, then edit the page template(s) to include the modified CSS file(s). Remember that upon an APEX upgrade, one will need to crosscheck the custom settings against the new APEX CSS, to check for any changes. For APEX IR, the core CSS is in the apex_3_1.css file. For learning and editing purposes, it is easiest to locate and copy the APEX IR CSS from the uncompressed version of the CSS file, in the <images>\css\uncompressed directory.

Most sections of the APEX IR are included in the apexir_ CSS directives. The most important exception is the apex_finderbar. The CSS for the finderbar is located in the same apex_3_1.css file, but apart from the apexir_ settings. Search for apex_finderbar to locate the finderbar settings. Copy the finderbar section to the top of the custom css file. Note that override of the finderbar style settings will override for all instances of apex_finsderbar, not just those in your APEX IR.

```

...
/**
 * This is the actual data table for worksheets
 */
table.apexir_WORKSHEET_DATA{border:0px #ffffff solid;}
table.apexir_WORKSHEET_DATA tfoot tr td{background:#E1EFE0;}
table.apexir_WORKSHEET_DATA th{
    background:#91C58E;
    font-weight:bold;
    color:#ffffff;
    border-top:1px #ccc solid;
    border-bottom:1px #aaa solid;
    font-size:11px;
    white-space:nowrap;
    vertical-align:center;
    letter-spacing:1;
    background-
image:url(file:///C:/oracle/oid1014/Apache/Apache/images/ws/report_bg.gif);
    background-repeat:repeat-x;
}
table.apexir_WORKSHEET_DATA td{
    background:#E1EFE0;
    border-top:1px #fff solid;
    border-bottom:1px #ccc solid;
    font-size:8pt;
    padding:3px 9px;
    empty-cells:show !important;
}
...

```

Code Listing 1 – APEX IR, Excerpt from Custom CSS File

APPROACH

1. Do not edit the apex_3_1.css file.
2. Locate the APEX IR CSS in apex_3_1.css. (It is easiest to do this in the uncompressed version) A search for “apexir_” will find the first apexir component.
3. Copy the APEX IR CSS into a separate CSS file, for example, my_custom_apexIR.css Do not edit the apex_3_1.css file. If desired, locate and copy the apex_finderbar settings in as well.
4. Include the new CSS file in the page template or the page header for the APEX IR page, taking care to include the new stylesheet after the #HEAD# substitution variable. Or, include the APEX IR CSS in a <style> section in the HTML Header element of the report builder interface.

```
<link rel="stylesheet"
href="#IMAGE_PREFIX#themes/theme_11/my_themell_apexIR.css" type="text/css" />
```

5. Edit the custom APEX IR CSS to achieve the desired result. A CSS editor or web development tool is helpful for this process. The Firefox Web Developer, the Edit CSS option, is ideal.
6. Test you changes in all supported browsers. Note that the APEX supported browsers are IE 6 and higher, and Firefox 1.5 and higher.
7. Test on the end users’ browsers. End user browser settings are often significantly different than developers’.

As a general recommendation, simpler is better. The fewer changes made, the least impact to existing APEX styles, and the more consistent colors applied, the better.

APEX IR – CSS ANATOMY

Before you modify the APEX IR CSS, it is helpful to understand the major APEX IR CSS components. Figure 8 lists the major components of interest for customizations.

APEX IR CSS Element	Description
apexir_WORKSHEET_REGION	The entire APEX IR worksheet region.
apexir_CONTROL_PANEL	The Control Panel area where the user sets action options.
apex_finderbar	The standard APEX search toolbar.
apexir_SEARCH_ICON	The Search bar icon.
apexir_SEARCH	The Search (leftmost) region of the toolbar.
apexir_ACTIONSMENU	The Action Menu (rightmost) region of the toolbar.
apexir_CHART	The Chart region of the APEX IR. This region alternates with the report region.
apexir_WORKSHEET_DATA	The data table. This contains th and td settings.
apexir_REPORT_TAB	The report tab, for Saved Reports.

Figure 8 – Major APEX IR CSS Elements

All of the APEX IR CSS elements may be modified. Often it is only necessary to edit a few, such as the apexir_WORKSHEET_DATA th and td backgrounds, or the finderbar and Ccontrol panel backgrounds, to produce the intended style.

DIFFERENT SHADES OF GRAY

When editing the APEX IR CSS, one will notice many shades of gray: #efefef, #eeeeee, #ddd, #909090 to name a few. If the intent is to change the default gray background, be careful to look at each apexir component – a simple search for #efefef or #eeeeee will not suffice. Note the different shades of gray – one may or may not want to retain the subtle differences.

Note that the apex_findebar is a table comprised of several images and text elements. The images have a gray background, so, if the requirement is to change the “background” color of the apex_finderbar, one needs to create images that use the same background color. The finderbar images both have a down arrow included. If these images are replaced, one should consider including an arrow as part of the replacement image, for end user clarity.

THE UGLY, BEAUTIFIED

Figure 7 shows the Sample Application in Theme 11, Round Green, Customers interactive report page in the default APEX IR appearance. Notice the black gradient table heading and grey table background somewhat clash with the mellow green of the theme. Figure 8 is a partial listing of the major APEX IR CSS elements. Code Listing 1 shows a partial listing of the custom CSS file, my_custom_apexIR.css. The modified settings are listed in **bold**. In this example, the only modifications are to the colors.

Home > Customers

Rows 15 Go ⚙️ Create Customer >

Working Report **Customers By Group** Customers_Default

Saved Report = "Customers By Group" ⊞ ⊞ ⊞

Group ⊞ ⊞ ⊞

Group : B

Address	Customer Name	City	State	ZIP Code
⊞ Schoephoester Road	Bradley, Eugene	Windsor Locks	CT	06096

Group : D

Address	Customer Name	City	State	ZIP Code
⊞ 45020 Aviation Drive	Dulles, John	Sterling	VA	20166

Group : H

Address	Customer Name	City	State	ZIP Code
⊞ 6000 North Terminal Parkway	Hartsfield, William	Atlanta	GA	30320

Group : L

Address	Customer Name	City	State	ZIP Code
⊞ 1 Harborside Drive	Logan, Edward	East Boston	MA	02128
⊞ Hangar Center, Third Floor	LaGuardia, Fiorello	Flushing	NY	11371
⊞ 10701 Lambert International Blvd.	Lambert, Albert	St. Louis	MO	63145

Group : O

Address	Customer Name	City	State	ZIP Code
⊞ 10000 West OHare	OHare, Edward "Butch"	Chicago	IL	60666

1 - 7

Top Customers

LaGuardia, Fiorello	\$3,800.00
Hartsfield, William	\$2,598.00
Dulles, John	\$1,200.00
Logan, Edward	\$790.00
Bradley, Eugene	\$500.00
Lambert, Albert	\$490.00
OHare, Edward "Butch"	\$250.00

Figure 9 – APEX IR with Custom CSS Settings Applied

Figure 9 shows the same Sample Application Customers page with some simple CSS edits applied in the file `my_theme_11_apexIR.css`. The changes took this author less than ½ hour, using Firefox Web Developer, Color Zilla and a simple text editor (Notepad or EditPad). The APEX IR now blends with the sidebar report on the same page. In this case, the finderbar was left gray, to be consistent with all APEX default search bars, but the Control Panel employs a soft green background. The end result is more harmonious with the theme: the ugly, beautified.

SUMMARY

APEX IRs offer a wealth of functionality for both developers and end users. There are a few limitations, and the default appearance is less-than-ideal. However, the time saved in development may be devoted to control access to features and for customizing the look-and-feel to meet corporate standards and application requirements.

Like most good things, APEX IR come with some bad, a few limitations, and some ugly, in this case the less-than-ideal appearance. As demonstrated above, there are workarounds for the major limitations, end user education can address the minor limitations, and, simple CSS edits improve the appearance. This makes the bad palatable, and makes the ugly irrelevant.

That leaves all good.

Enjoy!

RECOMMENDED READING

APEX OTN HOME PAGE AND ONLINE DOCUMENTATION

http://www.oracle.com/technology/products/database/application_express/index.html

Be sure to consult the documentation specific to your current release.

APEX INTERACTIVE REPORT OBE'S

[Using Interactive Report Regions with Oracle Application Express 3.1](#)

Making the most of an Interactive Report

[Building and Customizing an Interactive Report in Oracle Application Express 3.1](#)

Build, use and customize an Interactive Report

[Utilizing Advanced Interactive Report Region Techniques in Oracle Application Express 3.1](#)

Use some advanced techniques with Interactive Report Regions

[Creating Interactive Reports Over OLAP 11g Cubes](#)

Use Oracle Application Express to create an interactive sales analysis report that runs against OLAP 11g data

APEX FORUM

<http://forums.oracle.com/forums/forum.jspa?forumID=137>

APEX BLOG AGGREGATOR

<http://www.apexblogs.info/pls/apex/f?p=113>

CSS REFERENCE

<http://www.w3schools.com/css/default.asp>

ORACLE INSIGHTS

<http://oracleinsights.blogspot.com> and <http://apex.oracle.com/pls/otn/f?p=200801:2022:0>

Many thanks to Mark Lancaster for suggesting the iframe as one method of providing multiple APEX IRs on the same page.