Tuning oracle application express performance using advisor

For applications having a large number of concurrent users, maintaining optimal performance is critical.
**Tuning Performance**

For applications having a large number of concurrent users, maintaining optimal performance is critical. To optimize your application’s performance, remember to utilize the following features:

- Use bind variables within your application whenever possible. You can reference session state values using bind variable syntax in SQL queries and application logic such as PL/SQL executed from processes and validations. Accessing session state using bind variables is the most efficient way to reference session state. See "About Bind Variable Syntax".
- Include a #TIMING# substitution string in the region footer so that you can view the timing of each region. See "Understanding Substitution Strings".

**Running Advisor to Check Application Integrity**

The Oracle Application Express Advisor (Advisor) enables you to check the integrity and quality of your Oracle Application Express application. Advisor functions like a compiler or LINT and flags suspicious behavior or errors. Running Advisor checks the integrity of your application based on the underlying metadata.

**Topics:**

- About Oracle Application Express Advisor
- Running Advisor on an Entire Application
- Running Advisor on a Single Page

**About Oracle Application Express Advisor**

Before deploying your application, you can use the Oracle Application Express Advisor to perform various sanity checks on your application. The Advisor checks for errors, security issues, performance bottlenecks, usability and quality assurance.

**Running Advisor on an Entire Application**

To run Advisor on an entire application:

1. Navigate to the appropriate application:
   a. On the Workspace home page, click the Application Builder icon.

      The Application Builder home page appears.

   b. Select the application.
The Application home page appears.

2. Click **Utilities**.
3. Click **Advisor**.

The Advisor page appears.

**Description of the illustration debug_advisor.gif**
4. Under Checks to Perform, review the selected checks. Enable and disable check boxes as appropriate.
5. Click **Perform Check** at the top of the page.

   A Results page appears. Current Advisor settings are used next time the check is performed.

6. To remove any Advisor changes you have made, under Tasks select **Remove My Preferences**.
7. To run Advisor again, click **Perform Check**.

**Running Advisor on a Single Page**

To run Advisor on a single page:

1. Navigate to the appropriate application:
   a. On the Workspace home page, click the **Application Builder** icon.

      The Application Builder home page appears.

   b. Select the application.

      The Application home page appears.

   c. Select the page.

      The Page Definition page appears.

2. Click **Utilities**.

   The list of utility options displays.

3. Click **Check**.

   The Results page appears. Result reports are generated using the last Advisor settings.

4. To change settings, click **Change Settings**.

   The Advisor page appears.

5. Under Checks to Perform, review the selected checks. Enable and disable check boxes as appropriate.
6. To run the check again, click **Perform Check** at the top of the page.
A Results page appears. Current Advisor settings are used next time the check is performed.

7. To filter results, enable and disable check boxes under Filter Results and click **Apply Filter**.
8. To re-run the Advisor and see if reported problems are fixed, click **Check Again**.

**Reviewing Session State**

Many applications are based on data contained within application controls. For example, buttons can display conditionally based on a value stored in session state. You can view current session state for your application by clicking the **Session** link on the Developer Toolbar.

**See Also:**
"About the Developer Toolbar", "Viewing Session State", "Managing Session State Values", and "Managing Session State and User Preferences" in *Oracle Application Express Administration Guide*

**Accessing Debugging Mode**

The Debug mode is a built in mechanism used to track down unexpected application behavior. You enable and disable debug mode for the current page and session by clicking the **Debug** and **No Debug** links in the Developer Toolbar. To view Debug reports, click the **View Debug** link in the Developer Toolbar.

**Description of the illustration d_toolbar.gif**

**See Also:**
"About the Developer Toolbar"

**Topics:**

- **Enabling and Disabling Debugging**
- **Running an Application in Debugging Mode**
- **Using f?p Syntax to Access Debugging Mode**
- **Viewing Debug Deports**

**Enabling and Disabling Debugging**

You enable and disable debug mode at the application-level using the Debugging attribute on the Edit Application Definition page.
To enable or disable debugging:

1. On the Workspace home page, click the **Application Builder** icon.
2. Select an application.
3. Click the **Edit Application Properties** button to the right of the application name.
4. Scroll down to Properties.
5. For Debugging:
   - **Yes**. Enables the application debug mode using the browser at runtime.
   - **No**. Disables the application debug mode using the browser at runtime.

**See Also:**
"About Application Attributes"

**Running an Application in Debugging Mode**

You run and disable debug mode at runtime by clicking **Debug** and **No Debug** on the Developer Toolbar. By default, Debugging mode is disabled. See "Enabling and Disabling Debugging".

To enable or disable Debug mode:

1. On the Workspace home page, click the **Application Builder** icon.
2. Run the application.
3. On the Developer Toolbar, click **Debug**.

   The debugging mode is enabled. Debug data is captured in the background.

4. To disable debug mode, on the Developer Toolbar, click **No Debug**.

**Using f?p Syntax to Access Debugging Mode**

You can also use f?p syntax to run an application in Debugging mode. Simply call the page and set the Debug argument to YES. For example:

```plaintext
f?p=100:1:&APP_SESSION.::YES
```

**See Also:**
"Using f?p Syntax to Link Pages"

**Viewing Debug Deports**

To view a debug messages:

1. On the Workspace home page, click the **Application Builder** icon.
2. Select the desired application. 

You can view Debug messages when the application is running or in development mode.

3. To view a debug messages in development mode:
   - Click the **Find** icon and then the **Debug** tab. See "Using the Find Icon".
   - On the Application home page, click **Utilities**, and then **Debug Messages**.

4. To view Debug messages when the application is running, click **View Debug** on the Developer Toolbar.

A list of Debug reports displays.

5. Click the View Identifier to the left of the session you want to view.

The following illustration shows Debug messages viewed in the Item Finder.

Description of the illustration debug_report.gif

6. To view steps that took the longest, hover over that step on the graph above the report.
The step details display.

7. To go to that step in the report, click on the bar.

**Note:**

Debug sessions listed in the report live for at least 2 weeks and age out along with the activity log. This is usually more than adequate for debugging purposes.

---

**Enabling SQL Tracing and Using TKPROF**

Tracing your session can be a very effective way to debug an application. From a database perspective, each page request is a single database session. If you enable SQL tracing, then Oracle Application Express creates a temporary file you can then analyze using the TKPROF utility.

You enable SQL tracing in Oracle Application Express by using `f?p` syntax to set the argument `p_trace=YES`. For example, to trace the display of page 1 in application 100, you would use the syntax:

```
http://.../f?p=100:1&p_trace=YES
```

To use the TKPROF utility:

1. Log in to SQL*Plus as a privileged user.
2. Execute the following statement:
3. `
show parameter USER_DUMP_DEST`
4. Navigate to the directory in which the trace file is created.
5. Run the TKPROF utility from the operating system prompt using the following syntax:
6. `
tkprof filename1 filename2 [waits=yes|no] [sort=option] [print=n]`
7. `[aggregate=yes|no] [insert=filename3] [sys=yes|no] [table=schema.table]`
8. `[explain=user/password] [record=filename4] [width=n]`

The input and output files are the only required arguments.

9. To view online Help, invoke TKPROF without arguments.

**See Also:**

"Using Application Tracing Tools" in *Oracle Database Performance Tuning Guide* for information about using the TKPROF program.
Monitoring Application and Page Resource Use

Oracle Application Express facilitates the monitoring of resources used by applications and pages by calling the package DBMS_APPLICATION_INFO. Whenever the Application Express engine renders or processes a page, the module is set to APEX and includes the application ID and page number. Once set, you can query the V$SESSION and V$SQLAREA views to monitor transactions.

Viewing Page Reports

When isolating an issue within a page, it is important to clearly understand the functions it is performing. To accomplish this goal, Application Builder includes a number of page and application reports.

Viewing the Application Dashboard Report

The Application Dashboard provides a comprehensive view of your application from various perspectives. You can use this dashboard to view statistics and access reports of security settings, page and application components and templates.

To view the Application Dashboard:

1. Navigate to the Workspace home page.
2. Click the Application Builder icon.
3. Select an application.

   The Application home page appears.

4. Click Utilities.
5. Click Application Dashboard.

   The Application Dashboard appears. The first two regions on the left, Application Overview, and Pages by Types, list detailed information and statistics about the current application. The remaining regions are divided into the following sections:

   o **Security** lists the current authentication scheme, number of public and non-public pages, and the number of authorization schemes used within the current application.
   o **Application Components** contains links to reports of application-level controls and logic, including lists of values, tabs, lists, application items, application processes, and application computations.
   o **Templates** contains links to reports of templates used within the current application.
o **Page Components** contains links to reports of page-level controls and logic, including items, buttons, processes, regions, branches, validations, and dynamic actions.

6. To view a specific report, click the number to the right of the component type or template.

**Debugging Problematic SQL Queries**

If your query does not seem to be running correctly, try running it in SQL*Plus, SQL Developer, or in SQL Commands. Any of these approaches will test your query outside the context of your application, making it easier to determine what the problem is.

**Removing Controls and Components to Isolate a Problem**

If you have problems running a page, try removing controls and components one at a time. Using this approach, you can quickly determine which control or component may be the source of your problem. You can disable a control or component by selecting the Condition attribute **Never**.

To remove a control or component using conditional attributes:

1. Navigate to the appropriate Page Definition. See "Accessing the Page Definition".
2. Select the name of the control or component you want to disable.

   The appropriate attributes page appears.

3. Scroll down to Condition Type and select **Never** from the Condition Type list.
4. Click **Apply Changes** and return to the Page Definition.
5. Try running the page again.
6. Continue to remove controls or components until the page runs correctly.