Oracle Enterprise Manager
Oracle Fusion Middleware Management

Business Transaction Management Lab

Session S318968
Oracle Enterprise Manager 11g
Business Transaction Management
Hands-on Lab

Introduction to Enterprise Manager 11g

Oracle Enterprise Manager 11g is the centerpiece of Oracle's integrated IT management strategy, which rejects the notion of management as an after-thought. At Oracle, we design manageability into each product from the start, enabling Oracle Enterprise Manager to then serve as the integrator of manageability across the entire stack encompassing Oracle and non-Oracle technologies. Fueled by this unique vision, Oracle Enterprise Manager 11g has introduced business-driven IT management to help IT deliver greater business value through three highly differentiated capabilities:

- **Business-driven application management**, which combines industry-leading capabilities in real user experience management, business transaction management and business service management to improve application users' productivity while enhancing business transaction availability
- **Integrated application-to-disk management**, which provides deep management across the entire Oracle stack to reduce IT management complexity and eliminate disparate point tools
- **Integrated systems management and support**, which utilizes industry-first technology to bring support services into the IT management console; enabling proactive IT administration, increased application and system availability, and improved customer satisfaction

Oracle Business Transaction Management (OBTM)

More and more, organizations are turning to composite applications to implement critical business activities. Common examples include: Cellular Provisioning and Activation, Insurance Claims Processing, B2B Procurement etc.

Composite applications are difficult to manage. They do not respect the traditional boundaries between networks, middleware, and infrastructure. Instead, they span a variety of application components and networks to execute ‘business transactions’. Inherently brittle, composite applications are veritable black-boxes in an operational environment. Operations and Application Support staff responsible for these business systems face a variety of challenges, such as:

- Tracking transaction status and health
- Monitoring performance and SLAs
- Managing transaction errors and fallout

Neither IT nor business teams have traditionally had the visibility and control needed to efficiently manage composite applications. Business transactions can vanish at any point in their flow. IT and business staff usually first discover transaction issues through cryptic log entries or calls from vexed customers. Resource-intensive fire-drills follow, significantly increasing costs. Business Transaction Management alleviates these problems.

OBTM is an integral component of the following Oracle Enterprise Manager Management Packs:
• SOA Management Pack Enterprise Edition
• WebLogic Server Management Pack Enterprise Edition
• Management Pack for Non-Oracle Middleware

This lab will demonstrate:
• Discovering services and transactions in a distributed application
• Navigating the user interface for Oracle Business Transaction Management
• Monitoring transactions and services for performance and availability
• Monitoring SLA’s and usage
• Monitoring transactions for errors and exception
• Transaction analysis for identifying fallouts, and failures

Please feel free to seek assistance from the instructor or Oracle Demo staff at any point in time.

Logins and Passwords:
When you begin the Lab image, you will be logged into the image as well as web applications that you will use. However, you may be logged out after a period of inactivity.

If you are asked to login to the image via the Windows Login, use Administrator/weblogic1
If you are asked to login to Oracle BTM use weblogic/weblogic1

How to Start the Demo:
To launch Oracle BTM, click the desktop shortcut icon “Start Oracle BTM”.

When the image opens, two servers are running: One for BTM, the other for the application being managed, called “Toystore”.

If you need to restart the servers, you can do so using the two shortcuts on the desktop called “Start BTM” and “Start Toystore”.

If necessary, you can also start the servers from the Windows Start Menu:
To Start BTM:
Start>Oracle WebLogic>User Projects>BTM>Start Admin Server
To start Toystore:
Start>Oracle WebLogic>User Projects>pqedomain>Start Admin Server

If you are not using one of the designated BTM machines, and you want to start up the VM, first, close the two command prompt windows. Then start the 2 servers via the shortcuts on the desktop.
**Generate Business Transactions to Monitor**

**Exercise Walkthrough:**
Estimated Duration: 1-2 minutes

When you launch the BTM image, you should see a browser with two open tabs—one for BTM (on the left) and the other for Toystore (on the right).

You will use Toystore to generate transactions to be monitored in BTM.

If you need to open Toystore at any time, do the following:

1. Launch the Mozilla Firefox Browser. Note two bookmark links in the browser: Oracle Business Transaction Management and Toyco Purchasing Client. (To make things easier, you may wish to open two tabs in the browser (CTRL + T)--one for each bookmark.)

2. Select the “Toyco – Purchasing Client” bookmark.


[Note: If you see the error “Session timeout”, simply close the tab running Toystore, open a new Tab, and then click on the bookmark for the “Toyco Purchasing Client”.]

In the step, you create several transactions using the client that invokes the composite application monitored by Oracle BTM.

You will see an empty form displayed. You can enter the sample information you see below, or any information you would like. The BTM system will watch for users named Chris, James, Rajiv, Dhruv, and Andrew. Enter amounts larger than 10 in the “Quantity” field. You can create errors by leaving any field blank. Once you have completed the form, select Submit. You should enter two or three transactions with different user names, but one is enough.
<table>
<thead>
<tr>
<th>ID</th>
<th>Product</th>
<th>Item Price</th>
<th>Quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>1001</td>
<td>hsqldb_add more products_1</td>
<td>10.01</td>
<td>99</td>
</tr>
<tr>
<td>1002</td>
<td>hsqldb_add more products_2</td>
<td>10.02</td>
<td>99</td>
</tr>
<tr>
<td>1003</td>
<td>hsqldb_add more products_3</td>
<td>10.03</td>
<td>99</td>
</tr>
<tr>
<td>1004</td>
<td>hsqldb_add more products_4</td>
<td>10.04</td>
<td>99</td>
</tr>
</tbody>
</table>

**Ship Using:**
- FedEx TextMessage (SOAP)
- UPS TextMessage (non-SOAP)
- DHL DirectMessage
- USPS Express
**Discovery Services & Transactions**

**Exercise Walkthrough:**
Estimated Duration: 5-7 minutes

1. If you have not done so, launch the Mozilla Firefox Browser
2. Select the bookmark for “Oracle Business Transaction Management”.

The screen you see upon login is the entry point into the OBTM. OBTM extends visibility into application components including Enterprise Java Beans, RMI calls, or JDBC invocations. Any SOA component can be deemed as a service and graphically depicted as such with dependency diagrams.

![Diagram of application components and call paths](image)

This view, called **Explorer**, shows a birds-eye view of all application components discovered by OBTM, by monitoring live traffic flowing within the application.

For this exercise, the sample application running in the background simulates placing orders using a composite application comprised of different types of services. The **Explorer** shows all components in this application and the call paths between these components. It also displays some operational characteristics of the message traffic using color-coding schemes and icons. For instance, red colored node and lines indicate that an SLA is being violated by that node, or that exceptions are being thrown.

Click on OrderService node in the graph to select that service. The bottom pane of the view will display a set of performance views designed to give an overview of the performance of a component,
at different time intervals, at the operation level of granularity. You also have access to links that drill directly down to provide detail on SLA violations and exceptions.

3. Switching to the “List View” (by clicking on the icon) of the same information, you are able to customize and sort a table based on all of the information available in this system.
4. Now, click on Network tab, and then select the Summary sub-tab to navigate to Network->Summary view.

This view gives you high level summary or inventory of what your SOA landscape looks like. You can see numbers that represent the Services that you have, their status (up, down, managed or not), Types and numbers of Policies (pending, applied), the rate of change to those policies, policy capable intermediaries, number of containers that contain web services, UDDI registries, etc.

5. Click on Services sub-tab. This view will display all services discovered, in a list format. Select OrderService to see details for the OrderService.

6. Click Dependencies sub-tab in the details area to view all possible transaction flows through this service.

By observing messages flowing between different components of an application, OBTM will detect dependencies between the various services at a fine-grained level. OBTM is able to visualize that dependency in a logical (which is what we see here) as well as a physical fashion.
For example, in this graph, you can see that the PurchasingService (Web service) invokes the OrderService (Web service), which in turn calls a database using PurchasingDB (JDBC connection) as well as via an EJB. This is the path taken by a single ordering transaction in our sample application running in the background.

7. Click on the Profile sub-tab.

The Profile view displays the metadata associated with this service. Every managed object within the system has a "profile". The profile is extensible, and contains metadata that is used to categorize the object. Parts of the profile are automatically populated by OBTM using discovered data. Others can be filled in manually or programmatically, as necessary.
Monitoring Performance, Availability, SLAs, Usage

Exercise Walkthrough:
Estimated Duration: 10-15 minutes

Now let us examine how OBTM helps you understand the performance and use of services and transactions.

1. Click on Performance tab at the top of the console. This takes you to Performance->Summary view.

The Performance Dashboard in front of you is the “gateway” into service performance for all the components monitored by OBTM. This summary screen gives us a quick snapshot of the current state of services within our infrastructure. The traditional green, yellow, & red indicators will let us know immediately if we have any performance-related problems associated with a given service. The information that we see on the screen right now is based on a time range. If we roll back to a 1 hour segment, you will see the traffic applied to that 1 hour interval as well as any alerts that surfaced during that 1 hour segment of time. 6 hrs, 24 hours, in a summary fashion, generate reports based on that [report drop down], as well as interact with the graphs themselves.
2. Click on PurchaseTransaction in the Summary view, and select the Activity tab. Switch the time window to 6 hr view.

For a specific transaction, we can actually see information back to a 1 month period of time. The information that is captured in the graph is fully dynamic, so we can focus on a given type of metric. For example, if we want to just focus on response time only we can toggle on and off everything else and just focus on one type of instrument or measurement.

Click on Services sub-tab to open the Performance->Services view. This view displays all services in a list and details for each on the right hand side details pane. You can click each service to examine its performance and availability metrics.

3. OBTM can derive usage patterns for services that are part of a transaction or the front door for a Cloud API by examining message traffic.
Click on PurchasingService in the list and then select the Usage sub-tab in the details pane. Set up the filters to say “Show service usage by Customer in the last 6 hours and list top 10 with the most Throughput” and click Go.

We can see who the consumers of our services are, and see what kind of experience they’ve been having with that service from a performance perspective. We can see Response Time, Faults, Throughput, etc. The notion of a consumer for a service is highly flexible with OBTM. It can be tied to an actual individual, organization, department or subsidiary, as well as can be tied to physical attributes like Client IP Address, or we can segment the traffic based on the content of messages (or any business criteria)

Message based segmentation is a key investment for OBTM. We don’t force you to create different endpoints for each user segment to simply enable management of these users. We’re able to analyze a single message stream to identify usage patterns in real-time.

4. Whenever service performance starts to degrade, the OBTM system will generate alerts. One type of alert is based on performance levels that you establish for minimum acceptable service behavior, which we call Targets. OBTM allows you to set thresholds that define what the acceptable levels are for various performance metrics such as Throughput, Faults, Avg. Response Time, Availability etc. Click on Targets sub-tab for the PurchasingService object. OBTM allows you to establish
You can use this view to examine thresholds already established for this exercise. For instance, the system is set up to raise failure alerts if average response time for `createNewOrder` operation drops below 200 milliseconds and warning alerts when the response time crosses 50 milliseconds.

You can add multiple warning alerts to capture trends. Further, you can control the sensitivity of these alerts to prevent alert storms.

5. Targets are useful for actively controlling the system’s performance in general. But when you have business agreements in place with the service consumers such as partners, customers, divisions, business units etc., you must be able to model these complex agreements into a monitorable SLA. OBTM helps you monitor your services for compliance with such business agreements.

Click on Agreements sub-tab underneath Performance to navigate to the SLA view. Here, you can review all the SLAs already established for this exercise.

SLA’s combine multiple objectives into a single monitoring policy. SLAs are attached to groups of users and services. The group could be based on any of the segmentation mechanisms already discussed. The ability to model SLAs and multiple objectives is unique to OBTM. Most management vendors treat thresholds as SLAs, but that is insufficient when trying to manage enterprise class SLAs against multiple users.

6. OBTM gives you a business oriented view to quickly see SLA compliance for various consumers.

Click on Customers sub-tab underneath Performance.
The compliance view for each consumer shows which SLAs are attached to the consumer and how the system has performed with respect to those SLAs. The information is depicted graphically as well.

7. Click on Alerts sub-tab underneath Performance to view all alerts generated in the system thus far.

The Alerts view shows all alerts that have occurred. Alerts are also surfaced in the Explorer. Select any alert on PurchaseTransaction. From the alert, you can drill-down to the transactions that caused the alert!

To do so, click on See Message Log link.
This view displays all message instances that were processed in the time window during which the alert was raised. Select any one instance and click Inspect.
The transaction instances displays the time spent at each step as well as the message payload for each step.

Click on the magnifying glass icon to bring up the payload view! This can be used to reproduce the problem or debug with application developers. This deep visibility into transaction instances reduces MTTR (mean time to repair) from days to hours and minutes.
Monitoring Exceptions, and Analyzing Transactions

Exercise Walkthrough:
Estimated Duration: 10-15 minutes

This section examines Exception Management features of OBTM. Normally, we can classify exceptions into two - technical failures and business exceptions. Technical failures can be null pointer exceptions in code, SOAP faults thrown by services, and so on. Business exceptions describe a business situation of interest. E.g. you may have a business rule that states that no international airline tickets can be sold for less than USD200, or that no regulated equipment can be sold to customers outside the United States.

1. You can use Oracle BTM to trap all kinds of application errors – whether SOAP faults, HTTP error codes, or Custom Conditions that are based on application content or context.

   Click on Exceptions tab at the top of the console. By default, this will bring you to Exceptions-Summary view.

   ![Exceptions-Summary view](image)

   The summary page provides insight into the transactions that are having the most and the most recent exceptions. This enables you to quickly determine which transactions need attention right away and perform on the fly triage to address to most significant issues in priority order.

2. Click on All Exceptions sub-tab underneath Exceptions.

   This view gives you a central vantage point for visibility into all exceptions occurring in your application environment, in real-time. All open exceptions are listed in a tabular display, with details for each in the details pane underlying the table.
Without OBTM, you have to depend on after-the-fact log file scanning to obtain this information. Furthermore, depending on the logging characteristics of your development staff, you may or may not have this information logged.

On the other hand, Oracle BTM tracks the entire transaction from start to finish, captures all relevant messages and pinpoints the location of that exception. Once Oracle BTM is in place, operations staff can lean on the system to proactively alert them about errors as well as provide detailed insight into their root-cause.

3. Select any Large Order exception that you see in this view. You immediately get a visual depiction of where an error occurred within a given business process or transaction.

4. Click Inspect button, at the top of the screen, to get the message trace for that transaction and see exactly what went wrong. This Business Context is immensely helpful in debugging production issues.
In the inspection window, you can examine all requests, validate that all parameters that are being passed from one service request to another were all valid parameters [View XML] and can inspect the original payload in either structured or raw form. You can even view the SQL statements sent to the database.

Select any message sent to PurchasingDB and click the magnifying glass to see the SQL statement captured by the system. The statement is displayed in an html form as well as in an easy to read XML structure.
5. Proactive exception instrumentation is great news for your product environment. That said, many production issues are ad-hoc. Customers complain of problems, but cannot describe the exact issue. Here is when Business Context is most valuable. OBTM captures the business context for every transaction in the system, and makes it easy for you to search through that information to isolate the problem or its symptoms.

OBTM’s Message Search tool allows you to perform ad-hoc queries using business or technical data, such as customer name, order id, transaction id, time of day, etc.

To access this tool, click on Overview tab, and open the Explorer.

Click Search Messages button, located just below the graphical/tabular view.
6. For this lab, let’s imagine that users named “Rajiv” and “Chris” are complaining about lost orders and IT has asked for application support to investigate why some orders are being intermittently dropped by the application.

To solve this problem, let’s use Search Messages tool to find recent transactions submitted by “Rajiv” or “Chris”, and examine the issue.

In the search tool, Click Choose Content button and supply the text string to search for. Type “Rajiv” in the available text-field. Now select last 1 day from the drop down list. Click Search.

(The tool allows you to use free text search as well as regular expressions.)

The system will find all transactions submitted with the text content Rajiv in any of their constituent messages. All messages containing Rajiv are listed in a tabular form. Note that messages that have a SOAP fault are automatically flagged with an alert icon.)
7. Select the first transaction using a single mouse-click. Then click View->Related Correlation->PurchaseTransaction.
This will surface the end-to-end flow associated with that transaction, including all the messages, response times etc., as we saw in the previous drill-down to the transaction log. Also, note the business context - we have account info, order IDs and so on, all pulled from the content of the messages flowing through the system.

Thus, you can examine any of the transactions to understand what went wrong, and where in the complex environment did the transaction fall over.

Obviously, in this example, the transactions with the SOAP Fault highlighted are the culprits!

Thus OBTM helps you quickly identify the spot where you should focus all of your root-cause analysis efforts, instead of randomly digging up log files across your data center. The net savings in time and effort, further reduce your MTTR (mean time to repair), making you a more productive IT organization.
This concludes the Oracle Enterprise Manager Hands-on Lab.

Additional information can be found at:
Demo Booths located at {Location}
Additional Sessions:
Moscone South: Enterprise Manager # XXXX
Moscone West: Enterprise Manager # XXXX

For additional information, visit:

Oracle Enterprise Manager: SOA, WebLogic Server, and Middleware Management

Oracle Application Management
http://www.oracle.com/technology/products/oem/prod_focus/app_mgmt.html

Oracle Enterprise Manager
http://www.oracle.com/technetwork/oem/grid-control/overview/index.html