

Oracle® Tuxedo System and Applications Monitor Plus

Users Guide

12c Release 2 (12.2.2)

January 2017

Copyright © 2013, 2017, Oracle and/or its affiliates. All rights reserved.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

This software or hardware and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

Contents

Using Oracle TSAM Plus Console

Overview	1-1
User Accessibility	1-2
Top Menu Bar	1-2
Component Tree	1-3
Search Panel	1-8
Domains Dashboard	1-9
Using Oracle TSAM Plus Console	1-14
Monitoring Policy	1-14
Tuxedo Metrics Monitoring	1-33
CICS/IMS/Batch Runtime Metrics Monitoring	1-60
CICS/IMS/Batch Management	1-72
User/Data Management	1-80
Load Generator	1-95
Alert Monitoring	1-95

Using the Enterprise Manager for Oracle Tuxedo

Using Web-Based Console	2-1
Logging into Enterprise Manager Cloud Control	2-2
Viewing Tuxedo Summary	2-2
Target Home Page Introduction	2-4
Monitoring Tuxedo Targets	2-9
Viewing TUXCONFIG / BDMCONFIG	2-12
Viewing Configuration Topology	2-13
TuxJES Monitoring	2-13

Oracle Tuxedo-Side Configuration	2-14
Discovery	2-14
Login to TuxJES Monitoring Homepage	2-15
Monitoring Metrics	2-15
Operations on UI Console	2-16
Creating an Enterprise Manager Job for Tuxedo Batch Operation	2-16
GDG Management	2-17
Prerequisites	2-17
Operations on UI Console	2-18
Tuxedo Event Monitoring	2-18
Reporting for Tuxedo Targets	2-24
Using Enterprise Manager SLA	2-24
Creating Enterprise Manager Services	2-25
Using Resource Broker	2-28
About Resource Broker	2-29
Application Package Organization and Content	2-29
Application Package Management	2-39
Domain UBBCONFIG Editor	2-40
Deploying Application Packages to Tuxedo Domain	2-47
Deploying an Application Package Dynamically	2-58
Updating a Deployed Application Package	2-58
Removing Package-Associated Groups	2-59
Dynamic Resource Broker	2-59
Controls on Tuxedo Domain Target	2-64
Using Resource Broker for ART Batch	2-65
Using Resource Broker for ART CICS	2-75
Dynamic Request Broker	2-81
Enabling/Disabling Dynamic Request Broker	2-82

Viewing Dynamic Load Information.	2-82
Integration with OVAB	2-82
Supported Tuxedo Version	2-82
Supported Platform	2-82
Scaling Oracle Tuxedo Machines Number with OVAB Script	2-83
DM_MIB MBean Interface	2-88
Creating the TuxedoDomainGatewayTargetMBean Object	2-88
Application Programming Interfaces.	2-91
Security Requirement	2-97
User Cases	2-97
Integration with Exalogic Elastic Cloud.	2-110
Viewing Tuxedo Domains in Exalogic Elastic Cloud Targets	2-110
Easy Configuration for Multiple Tuxedo Domains.	2-112
Easy Applying Oracle Tuxedo Patches	2-113
Oracle Tuxedo Exalogic Compliance Standard.	2-114

Business Transaction Management (BTM) Integration

Overview.	3-1
Prerequisite	3-2
Enable BTM Monitoring from Tuxedo	3-3
BTM Discovery	3-3
Tuxedo Container in BTM	3-4
Dependency	3-6
BTM Transaction for Tuxedo Operation	3-6
Call Path in TSAM Console	3-7
Tuxedo Service in BTM	3-7

Oracle Tuxedo Scripting Tool

Using the Tuxedo Scripting Tool	4-1
What is the Tuxedo Scripting Tool?	4-1
Modes of Operation.	4-2
Main Steps for Using TXST	4-4
Navigating MBeans	4-10
Navigating and Interrogating Mbeans.	4-10
Browsing Runtime MBeans	4-15
Finding MBeans	4-15
Managing the Server Life Cycle	4-16
Starting and Stopping	4-16
Suspending and Resuming	4-16

TSAM Plus Accounting and Chargeback

Accounting Information Output	5-1
Prerequisite	5-1
Configuration	5-1
Accounting Information Analyzing	5-4
Configuration	5-4
Output	5-6
Samples	5-6
Known Issues	5-7

User Payload Collection

Overview	6-9
Supported User Payload Buffer Types.	6-9
Determine User Payload Storage	6-10
User Payload Record in Oracle Database	6-10

User Payload Record in Hadoop	6-12
User Payload Record in Local File	6-14
Specifying User Payload Storage from TSAM Plus Console	6-14
Configuring User Payload Collection.	6-15
. Defining Payload Collection Data Criteria	6-15
Defining Payload Collection Result Filter.	6-18
Exporting Collected User Payload Data.	6-18
APIs for Data Export	6-18
. Payload API Samples	6-37
Payload Dump Tool.	6-40

Using Test Load Generator

Overview.	7-1
Creating Call Path Policy for a Replay.	7-1
Creating a Replay Definition	7-2
Setting Replay Definition Attributes.	7-2
Running a Replay	7-4
. Execution Plan File	7-5
Replay Security Profile	7-7
. Replay Execution Result	7-7
Running a Replay in Auto-Created Test Domain.	7-9
Configuration File	7-9
Executing tlgdomgen.py	7-10

Setting Up Oracle Tuxedo as a Cloud Service

Overview.	8-1
Setting Up the Tuxedo Self Service Portal	8-2
Creating a Tuxedo PaaS Infrastructure Zone.	8-2

Creating a Tuxedo Pool	8-4
Configuring Tuxedo Profiles	8-4
Setting Up Quotas	8-7
Creating a Service Template	8-7
Configuring Request Settings	8-9
Configuring Chargeback	8-9
Using the Tuxedo Self Service Portal	8-10
Requesting a Tuxedo Service	8-11
Controlling the Tuxedo Service	8-12
Viewing the Tuxedo Service Home Page	8-12
Chargeback	8-13
Defining Policies for Tuxedo Services	8-13
Introspecting an Existing Domain	8-15
Introspecting an Existing Domain From Console	8-15
Introspecting an Existing Domain Using Command Line	8-16
Introspecting ART Applications	8-29
ART CICS Introspection	8-29
ART Batch Introspection	8-30
ART IMS Introspection	8-32

Using Oracle TSAM Plus Console

This chapter contains the following topics:

- [Overview](#)
- [Using Oracle TSAM Plus Console](#)

Overview

The Oracle TSAM Plus monitoring console allows you to specify the Oracle Tuxedo components you want to monitor, as well as track events and alerts. The console has six major sections:

- [User Accessibility](#)
- [Top Menu Bar](#)
- [Component Tree](#)
- [Search Panel](#)
- [Domains Dashboard](#)
- [Using Oracle TSAM Plus Console](#)

Note: To get the best experience and full functionality of Oracle TSAM Plus Console, Oracle recommends you use the following Web browsers:

- Internet Explorer 7 and later versions.
- Firefox 4, 5, 7, 8, and later versions.

User Accessibility

User accessibility settings can be adjusted from the log-in screen and the console page.

- Log-in Screen

In the upper left-hand corner of the log-in screen, click the **Settings** drop-down menu. You can select the following three options; the settings take effect immediately:

- I use a screen reader

Accessibility-specific constructs are added to improve screen reader behavior.

- I use high contrast colors

Application display uses high-contrast instead of the default contrast.

- I use large fonts

Application display uses large fonts instead of the default size fonts.

- Console Page

In the upper right-hand corner of the Oracle TSAM Plus Console page, click **Accessibility**; the **Accessibility Preferences** page appears. It has the same three user accessibility options as the log-in screen.

When you have selected your options, click **OK**; the settings take effect immediately.

Top Menu Bar

The top menu bar contains the following Oracle TSAM Plus monitoring console functionality:

Policy: Define and manage system policies.

Tuxedo Metrics: Query Oracle Tuxedo monitoring metrics.

Tuxedo Application Runtime Metrics: Query Oracle Tuxedo Application Runtime monitoring metrics.

Load Generator: Create the replay definition from call paths.

Management: Define user management, data management, global parameter settings, and blackout.

Alert: Define and query alerts.


Help: On-line help page.

Component Tree

Tuxedo Domains Component Tree

Tuxedo Domains tree displays the Tuxedo Domain Group -> Domain -> Machine -> Group -> Server -> Service hierarchy information.

All service nodes are hidden in the Tuxedo Component Tree by default. You can click **Show**

Service () to display them.


By default, the component tree displays the first domain. You can specify the domain to be displayed in the component tree from the Domain list. Your preference is remembered until the selected domain does not exist anymore.

The domain structure in the component tree and relevant pages will not automatically change

after you logged in. Click **Refresh** () to display the current domain structure.

In the component tree, different icon colors indicate different nodes status:

- Green: the domain and all its components are active
- Yellow: at least one child node under current node is in inactive or unhealthy status
- Red: the domain or component is inactive

Particularly, if an alert icon  arises, it indicates the domain contains unread alerts.

You can filter the component tree to display only domains with unread alerts or inactive components, domains in particular group, or particular domain using the **Domain** list.

You can view the overall status for all domains, a Tuxedo domain group, or a Tuxedo domain by clicking corresponding node in the component tree. For more information, see [Domains Dashboard](#).

CICS Regions Component Tree

CICS Regions tree displays the CICS Region -> CICS Transaction/CICS Terminals hierarchy information.

Besides hierarchy information, other information is displayed (such as Oracle Tuxedo version, Domain model, server status, etc.). Certain tasks (for example, Create Policy) can be performed directly on the tree nodes.

Batch Systems Component Tree

Batch Systems tree displays the JES application -> JES nodes -> Tuxedo JES servers hierarchy information.

Note: Similar to the `artjesadmin changeconcurrent` command, the maximum concurrent job number can also be changed in TSAM Plus console. Right click **ARTJESINITIATOR** in the JES component tree, there is an menu item named **Change Concurrent Jobs**. If the server is active, the menu item is available. On this option is selected, a popup window appears with the current max concurrent job number displayed. Input a number between 1 and 32767, the concurrent job number is modified.

IMS Systems Component Tree

IMS Systems tree displays the IMS systems ->Tuxedo Domain ->Transactions/Programs hierarchy information.

IMS System Summary

In the IMS Systems tree, you can right-click the root node and select **Summary** to view the system summary page.

The ART IMS system summary page contains two panels.

IMS System Summary Table

The system summary table lists all the ART IMS systems (domains) currently monitored by TSAM Plus. [Table 1-1](#) describes each column:

Table 1-1 IMS System Summary

Column Name	Description
Sysman Name	IMS system name. It is the first part string in the domain ID before the first ":" character. You can click the system name to go to the corresponding ART IMS Region Summary page.
Machine Names	The comma-separated list of all machine names in the domain.
Status	Domain status, Active or Inactive.
Last Start Time	The last startup time of the domain.
Number of Machines	Machine number in the domain.

IMS System Details Panel

When you select an IMS system line in the Summary table, the related system information is displayed in the Details panel below. The three tabs, MPP, BMP, and BMPT, represent three types of the ART IMS region monitored by TSAM Plus. The information shown for each region type are very similar. [Table 1-2](#) describes the MPP region type details.

Table 1-2 MPP Region Details

Metrics		Description
Number of MPP Regions		The total number of MPP regions (ARTIMPP processes) in the domain.
Totals Since Startup	CPU Consumption	Total CPU consumption. The system and user CPU of all transactions since the system startup.
	System CPU	Total system CPU consumption of all transactions since the system startup.
	User CPU	Total user CPU consumption of all transactions since the system startup.
	Transaction Count	Total transaction count since the system startup.
Totals Since Last Hour	CPU Consumption	Total CPU consumption. The system and user CPU of all transactions in the last hour.
	System CPU	Total system CPU consumption of all transactions in the last hour.
	User CPU	Total user CPU consumption of all transactions in the last hour.
	Transaction Count	Total transaction count in the last hour.

Table 1-2 MPP Region Details

Metrics	Description	
Averages in the Last Hour	Transaction Time	The average transaction elapse time in the last hour (total transaction time / transaction count).
	Transaction CPU Consumption	The average transaction CPU time in the last hour (system and user CPU).
	Transaction System CPU	The average transaction system CPU time in the last hour.
	Transaction User CPU	The average transaction user CPU time in the last hour.
	Transaction Rate	The transaction count in each 15 minutes time window in the last hour.

IMS Region Summary

In the IMS Systems tree, you can right-click the domain node and select Summary to view the region summary page.

The ART IMS region summary page contains two panels.

IMS Region Summary Table

The IMS Region Summary table lists all the ART IMS regions (servers) currently monitored by TSAM Plus in this domain. [Table 1-3](#) describes each column:

Table 1-3 IMS Region Summary

Column Name	Description
Region	The region name is in the format of "process name/group name/server id", for example, ARTIMPP/GROUP1/6 Note that each UBB server entry is considered as one region, no matter what MIN and MAX values are configured. One region can actually contains multiple server instances.
Region Type	The region type is one of the following values: <ul style="list-style-type: none">• BMP (Batch-oriented): The process is ARTIBMP• BMP (Transaction-oriented): The process is ARTIBMPT• IFP: IMS Fast Path Region. The process is ARTIMPP, and "-p" is not configured in the CLOPT of the server.• MPR: Message Processing Region. The process is ARTIMPP, and "-p" is configured in the CLOPT of the server.
Status	Tuxedo server status, Active or Inactive.
Number of Servers	The started server instances number.
Last Start Time	The last server startup time.

IMS Region Details Panel

When you select an IMS region line in the Summary table, the related IMS region information is displayed in the Details panel below.

[Table 1-4](#) describes the metrics shown in the main area.

Table 1-4 IMS Region Details

Metrics		Description
Totals Since Startup	CPU Consumption	Total CPU consumption. The system and user CPU of all transactions in this region since the server startup.
	System CPU	Total system CPU consumption of all transactions in this region since the server startup.
	User CPU	Total user CPU consumption of all transactions in this region since the server startup.
	Transaction Count	Total transaction count in this region since the server startup.
Totals Since Last Hour	CPU Consumption	Total CPU consumption, system + user CPU of all transactions in this region in the last hour.
	System CPU	Total system CPU consumption of all transactions in this region in the last hour.
	User CPU	Total user CPU consumption of all transactions in this region in the last hour.
	Transaction Count	Total transaction count in this region in the last hour.
Averages in the Last Hour	Transaction Time	The average transaction elapse time in the last hour (total transaction time / transaction count).
	Transaction CPU Consumption	The average transaction CPU time in the last hour (system and user CPU).
	Transaction System CPU	The average transaction system CPU time in the last hour.
	Transaction User CPU	The average transaction user CPU time in the last hour.
	Transaction Rate	The transaction count in each 15 minutes time window in the last hour.

Search Panel

Search component tree elements.

Domains Dashboard

When you log into Oracle TSAM Plus monitoring console, the Domains Dashboard is displayed with overall Tuxedo domains and domain groups metrics. It presents the following information:

- Severity-level unread alert count. To read the alerts of a severity level, click the number after the level.
- An aggregated view of all domains components availability. When clicking the inactive (red) part of a pie chart, a window is popped up showing all the inactive items in detail.
- All domains and domain groups metrics charts

The metrics charts show the related metrics data captured in the selected time range. When you move the cursor over a data node, the data value is shown up. For the metrics data collection interval, you can set it from the Management-> Global Parameters page. For more information, see [Global Parameters](#).

The domains dashboard always displays the first three metrics charts positioned in individual domain or domain group dashboard for a quick glimpse. By default, these charts are displayed with the data for all services in last 24 hours.

- Throughput

The number of Tuxedo service calls per minute.

- Response Time

The average response time (in microseconds) of Tuxedo service calls.

- Failure Number

The number of Tuxedo service call failures per minute.

For the domain groups presented in the dashboard, you can manage them from the **Management** -> **Domain Group Management** page. For more information, refer to [Domain Group Management](#).

From any domain or domain group dashboard, you can set the dashboard to refresh the data periodically by choosing the **Auto Refresh** box on the top and specifying the refresh interval from the list. The refresh interval you set affects all dashboard pages.

Note: This feature is only available when Oracle database is used, and is not available if you are using Derby database.

Viewing Domain Group Dashboard

To view a domain group dashboard, click the domain group name in the dashboard or left component panel.

- Unread Alerts

Displays severity-level unread alert count. To read the alerts of a severity level, click the number after the level.

- Availability

Displays aggregated availability of all domains belonging to the group. When clicking the inactive (red) part of a pie chart, a window is popped up showing all the inactive items in detail.

- Metrics Charts

The throughput, response time, and failure number charts are displayed with the data in last 24 hours by default. For any metrics chart, you can change the time range to show the chart from the drop-down list.

If you have set the chart to show metrics per service, you get multiple lines on the chart representing each selected service in different colors. You can hide or unhide certain services lines by checking or unchecking the short bar (-) icon before the desired services name under the chart.

You can also customize the charts in dashboard. For more information, refer to [Customizing a Dashboard](#).

Click the group name on the top of the dashboard, you enter the domain group details page that presents the availability and the first three metrics charts of the domain group for each individual member domain. You can sort the domain list in ascending or descending order by the metrics selected in the **Order By** list.

To go further to view and edit individual domain dashboard, click the domain name. For more information, refer to [Viewing Individual Domain Dashboard](#).

Viewing Individual Domain Dashboard

To view individual domain dashboard, click the domain name in any dashboard or left component panel.

- Service Statistics

Click this link to enter the Service Statistics page, which shows you the last-time collected metrics data breakdown for each services.

You can filter the list to display desired services. Type the name keywords (case-insensitive) in the Filter textbox and click the button. You can input multiple keywords separated by commas, semicolons, or spaces.

You can also export the service statistics table as a CVS file by clicking **Export to CVS**.

To view detailed metrics data collected for a service, right-click the service row and select one of the time range:

- 24 Hours: displays the aggregated data of every collection interval in 24 hours.
- 7 Days: displays the aggregated data for every hour in the past six days.
- 31 Days: displays the aggregated data for every day in the past month.

- Unread Alerts

Displays severity-level unread alert count. To read the alerts of a severity level, click the number after the level.

- Availability

Displays availability of servers in the domain. When clicking the inactive (red) part of a pie chart, a window is popped up showing all the inactive items in detail.

- Metrics Charts

The throughput, response time, and failure number charts are displayed with the data in last 24 hours by default. For any metrics chart, you can change the time range to show the chart from the drop-down list.

If you have set the chart to show metrics per service, you get multiple lines on the chart representing each selected service in different colors. You can hide or unhide certain services lines by checking or unchecking the short bar (-) icon before the desired services name under the chart.

You can also personalize the charts in dashboard. For more information, refer to [Customizing a Dashboard](#).

Customizing a Dashboard

In a domain group or domain dashboard, you can chart metrics over any time period you choose, create your own charts to suit your need, and define the key services for regular monitoring. Click the **Customize** button in the upper right page to activate customization mode.

Note: Any customization configuration made in the group dashboard does not affect the underlying domain dashboard definition and vice versa.

Defining Key Services

In many cases, you may want to monitor a fixed set of services regularly. You just need to define them as key services to avoid repeatedly specifying target services. The key services can then be used when you edit or create a chart.

To define the key services for a domain, click the **Key Services** in the upper right of the domain dashboard page and add the desired services to the left box.

To define the key services for a group, refer to [Domain Group Management](#).

Editing a Metrics Chart

To customize the chart that appears, click **Edit** in any metrics chart field and define the parameters listed in [Table 1-5](#).

Table 1-5 Metrics Chart Parameters

Options	Description
Chart Name	Specifies a name to identify the chart.
Service List	Specifies the services you want to monitor on the chart: <ul style="list-style-type: none">• All Services• Key Services: The key services you defined before for regular monitoring purpose. For more information, refer to Defining Key Services.• Select Manually: Select particular services you want to monitor in the list.

Table 1-5 Metrics Chart Parameters

Options	Description
Metrics	<p>Specifies the metric chart type:</p> <ul style="list-style-type: none">• Throughput (per minute): Number of Tuxedo service calls per minute• Number of Successes (per minute): Number of success Tuxedo service calls per minute• Number of failures (per minute): Number of failed Tuxedo service calls per minute• Number of System Failures (per minute): Number of Tuxedo service calls with system failure per minute• Number of User Failures (per minute): Number of Tuxedo service calls with user failure per minute• Response Time (microsecond): Average Tuxedo service call response time• Wait Time (microsecond): Average Tuxedo service call wait time• Execution Time (microsecond): Average Tuxedo service execution time• CPU Time (microsecond): Average Tuxedo service CPU consumption time
Range	<p>Specifies the time range to present the data on the chart. You can choose to display the data in:</p> <ul style="list-style-type: none">• Last Hour: the aggregation interval is every metrics collection interval• Last 24 Hour: the aggregation interval is every metrics collection interval• Last Week: the aggregation interval is one hour• Last 31 Day: the aggregation interval is one day <p>By default, the chart presents data in last 24 hours.</p>
Per Service	<p>This option is available when Service List is not set to All Service. When selected, the chart shows lines of each service separately in different colors.</p>
Chart Type	<p>Specifies the chart type: Line, Bar, Area, Combination, or Line with Area.</p>
Chart Size	<p>Specifies the width of the chart.</p>

Adding a Metrics Chart

You can create up to 12 new charts by clicking **Add** in the upper right page. For descriptions of parameters in the window that appears, refer to [Table 1-5](#).

Moving or Removing a Panel

You can move up, move down, or delete any chart by clicking related function buttons at the top of the chart field.

Only the top three charts positioned in an individual domain dashboard are presented in the group or all domains dashboard.

Using Oracle TSAM Plus Console

This section contains the following topics:

- [Monitoring Policy](#)
- [Tuxedo Metrics Monitoring](#)
- [CICS/IMS/Batch Management](#)
- [User/Data Management](#)
- [Alert Monitoring](#)

Monitoring Policy

Oracle TSAM Plus provides comprehensive monitoring control of Oracle Tuxedo infrastructure behavior. Policy Management allows you to do the following:

- Organize monitoring requirements into significant and useful monitoring policy solutions.
- Manage monitoring policy (including import, export and usage tracking).
- Define monitoring policy entries for Oracle Tuxedo components and dynamic conditions.
- Selectively enable and disable real-time communication with monitored Oracle Tuxedo back-end components.

This section contains the following topics:

- [Tuxedo Monitoring Policy](#)
- [CICS Runtime Monitoring Policy](#)

- [IMS Runtime Monitoring Policy](#)

Tuxedo Monitoring Policy

On the menu bar, click **Policy** and select **Tuxedo Monitoring Policy** from the drop-down menu; the **Monitoring Policy List** page appears. It displays the existing defined policies and allows you to view, add, edit, or delete policies. This section contains the following topics:

- [Monitoring Policy List Page](#)
- [Create/Edit Policy Page](#)

Monitoring Policy List Page

The Monitoring Policy List page allows you to create and manage the Oracle Tuxedo component monitoring policies.

Button Bar

[Table 1-6](#) lists the **Monitoring Policy List** button bar functions.

Table 1-6 Monitoring Policy List Button Bar

Button	Description
View	From the drop-down menu, select the following: <ul style="list-style-type: none"> • Columns: Select the columns displayed in the table. • Detach: Displays the table in a separate window. • Reorder Columns: Change how the column order is displayed.
Add	Creates a new monitor policy. The maximum policy name character length is 255. Note: You can also add a new policy by right-clicking a Domain listed in the Oracle Tuxedo Components panel.
Delete	Deletes selected monitoring policies.
Enable	Enables selected monitoring policies.
Disable	Disables selected monitoring policies.
Edit	Edits selected monitoring policies.

Table 1-6 Monitoring Policy List Button Bar

Button	Description
Clone	Clones a new policy from a selected policy.
Import	Imports previously exported policies from a user-specified location.
Export	Exports selected policies in an .xml file to a user-specified location.
Refresh	Updates the policies displayed in the Policy List table.
Detach	Displays the table in a separate window.

Policy List Table

The Policy List table displays the following columns:

- **Select:** Allows you to select all or individual policies.
- **Name:** Displays the current policy names.
- **Status:** Displays the policy status.
 - **enabled:** The corresponding request is sent to Oracle Tuxedo.
 - **disabled:** The corresponding request is not sent to Oracle Tuxedo.
- **Note:** Policy monitoring does not take effect until the impacted Oracle Tuxedo processes have started.
- **Domain ID:** Displays the domain identifier attached to the policy as follows:
DOMAINID:Master:IPCKEY.
- **Tuxedo Components:** String that represents policy impact scope of Tuxedo resources in selected domain. It is FML32 boolean expression compliant. For example:
(TA_P MID%' . *BOXBANK* ') indicates using an FML32 boolean expression for machine selection.

Policy for PeopleSoft Domains

TSAM Plus provides out-to-box policies for monitoring Oracle PeopleSoft Applications. The PeopleSoft domain is displayed different from common Tuxedo domains in the list. Its Domain ID is separated by three colons, e.g. APPDOM_217989:LOCALHOST.COM:PS_DOMAI:PS. To monitor PeopleSoft domains, select the policy and click **Enable**.

Create/Edit Policy Page

The Create/Edit Policy page contains the following sections:

- [Tuxedo Component Panel](#)
- [Monitoring Policy Panel](#)

Tuxedo Component Panel

The Tuxedo Component panel contains the following selections:

- **Domain:** Required.
- **Machine:** Optional.
- **Group:** Optional.
- **Server:** Optional.

Machine, Group, and Server can also accept FML32 boolean expressions. Click the radio button to select an input method (from the drop-down list or manually enter an FML32 boolean expression). The value of each level is determined by its parent level (except for Domain). One monitoring policy must be specified for one domain.

Monitoring Policy Panel

The Monitoring Panel contains the following tabs:

[Call Path Tab](#)

[Service Tab](#)

[XA Transaction Tab](#)

[Domain Gateway Tab](#)

[BRIDGE Tab](#)

[GWWS Tab](#)

To use the listed tab options, click the **Enable** check box. If enable is not selected, all tab options are disabled. One monitoring policy can contain multiple monitoring categories.

Call Path Tab

[Table 1-7](#) lists the **Call Path** tab options.

Table 1-7 Call Path Tab Options

Options	Description
Basic Policy Options	
Enable	Enables/disables call path monitoring. Note: The first policy deployed to the Tuxedo domain takes effect when there are more than one policies enabled on a Tuxedo target. The first policy attributes, such as extended, bridge decode, affect the whole call path.
Ratio/Interval	Selects Ratio or Interval policy. If multiple policies cover one initiator process and the ratio or interval policy are different, the smaller value is used. Applies to other monitoring policy definitions as well. <ul style="list-style-type: none">• Ratio: Accepts a value range of 1-65535. It indicates that monitoring starts for a certain number of requests. The default value is "1" (indicating each request is monitored).• Interval: Accepts a value range of 1-65535. It indicates a period of time (in seconds) that monitoring can be started. It is exclusive of ratio settings. The default value is "1". The ratio/interval control only applies to the call path initiator.
Alert Only	Specifies alert evaluation only. No metrics are sent to manager.
Bridge Decode	Allows all BRIDGES in the call path to decode messages. By default, BRIDGES do not decode messages and the call path representation does not show BRIDGE points. Note: Use this option carefully. Decode/encode impacts application performance.

Table 1-7 Call Path Tab Options

Options	Description
Enable Extended Monitoring	<p>When this option is enabled, the extended monitoring data reported by the <code>tsambegin()</code> and <code>tsamend()</code> API are collected by call path. These API are invoked automatically in Oracle Tuxedo ART for CICS and Batch when a CICS command, EXEC SQL, or IMS DLI is invoked.</p> <p>Note: To monitor Pro*C sql by OCI , OCI library must be loaded in TSAM Plus Agent. Specify <code>libclntsh.so</code> location in the lib path when starting up the domain</p> <p>Following SQL are supported by OCI:</p> <ul style="list-style-type: none">• EXEC SQL ALTER• EXEC SQL AUDIT• EXEC SQL COMMENT• EXEC SQL CREATE• EXEC SQL DROP• EXEC SQL GRANT• EXEC SQL NOAUDIT• EXEC SQL RENAME• EXEC SQL REVOKE• EXEC SQL TRUNCATE• EXEC SQL DELETE• EXEC SQL EXPLAIN PLAN• EXEC SQL INSERT• EXEC SQL LOCK TABLE• EXEC SQL OPEN• EXEC SQL SELECT• EXEC SQL UPDATE• EXEC SQL SAVEPOINT• EXEC SQL EXECUTE• EXEC SQL PREPARE• EXEC SQL SET ROLE• EXEC SQL EXECUTE END-EXEC
Enable Extended Monitoring Limit	<p>When this option is enabled, it specifies the maximum extended call path segments that can be generated in a call path when "Enable Extended Monitoring" is enabled. The acceptable value range is 1~2147483647. Default value is 1000.</p>

Table 1-7 Call Path Tab Options

Options	Description
Enable Service Contract Discovery	When this option is selected, the monitored services contract information is collected. For more information, see User Payload Collection .
Define Payload Collection	Click this button to specify the payload collection policies. For more information, see User Payload Collection .
Define Alert	Displays the alert definition page. For more information, see Tuxedo Alert Definition .
Define Conditional Filter	<p>Click the button to define the conditional call path filters. When specified, only the call path metrics matched are collected and sent to TSAM Plus manager side. This feature saves network and database resources and allows administrators to focus on exceptional cases.</p> <p>To specify the filter, select Enable.</p> <ul style="list-style-type: none">• Report Previous/Next Callpath Number <p>When it is enabled, the pervious and next call paths of the matched one are also collected. You can specify the amounts of the associated call paths. The available values range is 0~100.</p> <ul style="list-style-type: none">• Metrics Evaluation Expression(s) <p>You can select the metric, operator and threshold value to compose an expression. The logic for the expression can be "and" or "or".</p> <p>Following conditional filters are available:</p> <ul style="list-style-type: none">– Elapse Time exceeds a numeric value– Execution Status is or is not a value– URcode (tpreturn) is or is not a numeric value– Call depth is greater than or equal to a numeric value– Message Number on Request Queue is greater than (or equal to) a numeric value <ul style="list-style-type: none">• FML Boolean Expression(s) <p>You can also input FML Boolean expressions directly.</p>

Table 1-7 Call Path Tab Options

Options	Description
Dynamic Filter Panel	<p>Dynamic filter conditions are independent of each other. If multiple conditions are configured, the evaluation is <code>true</code> for all conditions and the monitoring can be initiated.</p> <p>For single condition items by default, if one of the item evaluations is <code>true</code>, then the condition is true. This applies to other monitoring policy definitions unless there is a special comment.</p>
Initiator Type List	<p>Allows you to choose the following Initiator types:</p> <ul style="list-style-type: none">• Native Client• Workstation Client• Jolt Client• Domain Gateway• Application Server• Web Service Client• TMQFORWARD <p>Note: "Domain Gateway" acts as a call path initiator for local service requests. For imported services, the initiator type is set to the original caller process. The "Web Service Client" initiator point is established from the Web service gateway process (GWWS).</p> <p>To enable the call path monitoring for GWSNAX inbound request, you must select "Domain Gateway" when defining the policy.</p> <p>The WTC module is not supported for call path monitoring.</p>
Service List	<p>Specifies the initiator location monitored call path services. Select a service from the drop-down list. If Machine, Group, or Server is selected using an FML32 boolean expression, you must edit/input manually.</p>
IP Address List	<p>Applies to workstation client and Jolt client initiator types. Specifies the IP address that initiates call path monitoring. The value can be an IPV4 or IPV6 address, or an FML32 boolean expression.</p>
Client Name List	<p>Lists the native client, workstation client, and Jolt client initiator type client names.</p>
User Name List	<p>Lists the initiator type user names.</p>

Service Tab

Table 1-8 lists the **Service** tab options.

Table 1-8 Service Tab Options

Options	Description
Basic Policy Options	
Enable	Enables/disables service monitoring.
Ratio/Interval	Selects Ratio or Interval policy. If multiple policies cover a single process and the ratio or interval policy are different, the smaller value is used. Applies to other monitoring policy definitions as well. <ul style="list-style-type: none">• Ratio: Accepts a value range of 1-65535. It indicates that monitoring is started among how many requests. The default value is "1" (indicating each request is monitored).• Interval: Accepts a value range of 1-65535. It indicates a period of time (in seconds) that monitoring can be started and is exclusive of ratio settings. The default value is "1".
Alert Only	Specifies alert monitoring only. No metrics are sent to manager.
Enable Service Contract Discovery.	When this option is selected, the monitored services contract information is collected. For more information, see User Payload Collection .
Enable Payload Collection	When this option is selected, the user payload data of the monitored services is collected. For more information, see User Payload Collection .
Define Alert	Displays the alert definition page. For more information, see Tuxedo Alert Definition .
Dynamic Filter Options	
Service List	Specifies the monitored services. Select a service from the drop-down list. If Machine, Group, or Server is selected using an FML32 boolean expression, you must edit/input manually. Note: Oracle TSAM Plus can also monitor CORBA interfaces. For CORBA, the interface name is the same as Oracle Tuxedo ATMI services; however, "Enable Service Contract Discovery" does not apply the CORBA interface.

XA Transaction Tab

[Table 1-9](#) lists the **XA Transaction** tab options.

Table 1-9 XA Transaction Tab Options

Options	Description
Basic Policy Options	
Enable	Enables/disables XA Transaction monitoring.
Ratio	<p>Selects Ratio policy. If multiple policies covering a single initiator process and the ratio policy are different, the smaller value is used. Applies to other monitoring policy definitions as well.</p> <p>It accepts a value range of 1-65535. It indicates that monitoring is started among how many requests. The default value is “1” (indicating each request is monitored).</p> <p>If transaction policy is applied to the transaction initiator process, then (similar to call path monitoring), the ratio is enforced during the initiator process. If it applies to a non-initiator transaction participator, the ratio impacts the monitored XA routines.</p> <p>Note: Monitoring from the transaction initiator is recommended.</p>
Define Alert	Displays the alert definition page. For more information, see Tuxedo Alert Definition .

Table 1-9 XA Transaction Tab Options

Options	Description
Dynamic Filter Options	
Initiator Type List	<p>Allows you to choose the following Initiator types:</p> <ul style="list-style-type: none"> • Native Client • Workstation Client • Jolt Client • Domain Gateway • Application Server <p>Transaction path monitoring from an initiator is currently supported. If monitoring is implemented during the initiator process, all XA Transaction points are monitored during the transaction propagation.</p>

Domain Gateway Tab

[Table 1-10](#) lists the **Domain Gateway** tab options.

Table 1-10 Domain Gateway Tab Options

Options	Description
Enable	Enables/disables Domain Gateway (GWTDOMAIN) monitoring.
Interval	<p>Selects Interval policy. If multiple policies cover a single GWTDOMAIN process and the interval policy are different, the smaller value is used. Applies to other monitoring policy definitions as well.</p> <p>It indicates a period of time (in seconds) that monitoring can be started. It accepts a value range of 1-65535. The default value is 300.</p>
Define Alert	Displays the alert definition page. For more information, see Tuxedo Alert Definition .

BRIDGE Tab

[Table 1-11](#) lists the **Bridge** tab options.

Table 1-11 Sub Controls of BRIDGE Panel

Options	Description
Enable	Enables/disables Bridge monitoring.
Interval	<p>Selects Interval policy. If multiple policies cover a single BRIDGE process and the interval policy are different, the smaller value is used. Applies to other monitoring policy definitions as well.</p> <p>It indicates a period of time (in seconds) that monitoring can be started. It accepts a value range of 1-65535. The default value is “300”.</p>
Define Alert	Displays the alert definition page. For more information, see Tuxedo Alert Definition .

GWWS Tab

[Table 1-12](#) lists the **GWWS** tab options.

Table 1-12 GWWS Panel Tab Options

Options	Description
Enable	Enables/disables GWWS monitoring.
Interval	<p>Selects Interval policy. If multiple policies cover a single GWWS process and the interval policy are different, the smaller value is used. Applies to other monitoring policy definitions as well.</p> <p>It indicates a period of time (in seconds) that monitoring can be started. It accepts a value range of 1-65535. The default value is “300”.</p>
Define Alert	Displays the alert definition page. For more information, see Tuxedo Alert Definition .

Notes: A policy must belong to a single domain. Oracle TSAM Plus does not support cross-domain policies. When an invalid Oracle Tuxedo domain is removed, the policies are also removed.

All policies are global configurations that are visible for all users and editable for users with policy management administrator privileges.

It is possible for a policy to be modified *simultaneously* by multiple users. The *final changes* may not be seen by other users depending on the console refresh interval.

CICS Runtime Monitoring Policy

Oracle TSAM Plus provides comprehensive monitoring control of Oracle Tuxedo Application Runtime infrastructure behavior.

This section contains the following topics:

- [Monitoring Policy List Page](#)
- [Create/Edit Policy Page](#)

Monitoring Policy List Page

On the menu bar, click **Policy** and select **Tuxedo Application Runtime Monitoring Policy** from the drop-down menu. The **Monitoring Policy List** page appears. It displays existing defined Tuxedo Application Runtime monitoring policies and allows you to view, add, edit, or delete policies.

Button Bar

[Table 1-13](#) list the button bar functions.

Table 1-13 Button Bar

Button	Description
View	From the drop-down menu, you can select the following: <ul style="list-style-type: none">• Columns: Select the columns to shown in the table.• Detach: Displays the table in a separate window.• Reorder Columns: Change how the column order is displayed.
Add	Create a new monitor policy. The maximum policy name character length is 255. You can also add a new policy by right-clicking a Domain listed in the Oracle Tuxedo Components panel.
Delete	Selected monitoring policies are deleted.
Enable	Selected monitoring policies are applied.
Disable	Selected monitoring policies are cancelled.
Edit	Edit selected monitoring policy.
Clone	Clone a new policy from a selected policy.
Import	Imports previously exported policies from a user-specified location.
Export	Exports selected policies in an <code>.xml</code> file to a user-specified location.
Refresh	Updates the policies shown in the Policy List Table.

Policy List Table

The Policy List table displays the following columns:

- **Select:** Allows you to select all or individual policies.
- **Name:** Displays the current policy names.
- **Status:** Displays the policy status.
 - **enabled:** The corresponding request is sent to Oracle Tuxedo.
 - **disabled:** The corresponding request is not sent to Oracle Tuxedo.

Note: Policy monitoring does not take effect until the impacted Oracle Tuxedo processes have started.

- **CICS Region ID:** Displays the CICS region identifier (CICS region name) attached to the policy.

Create/Edit Policy Page

The Create/Edit Policy page contains the following sections:

- [Policy Property Panel](#)
- [Tuxedo Application Runtime Monitoring Policy Panel](#)

Policy Property Panel

The Tuxedo Application Runtime Policy panel contains the following:

- **Name:** Policy name input text box. Required.
- **CICS Region:** Selection of region that the current policy attaches to. Required.

Tuxedo Application Runtime Monitoring Policy Panel

The Tuxedo Application Runtime Monitoring Panel contains the following tabs:

[CICS Transaction Tab](#)

[CICS Terminals Tab](#)

To use the options listed in each tab, click the **Enable** check box. If enable is not selected, all options in the tab are disabled.

CICS Transaction Tab

[Table 1-14](#) lists the **CICS Transaction** tab options.

Table 1-14 CICS Transaction Tab Options

Options	Description
Enable	Enables/disables CICS Transaction monitoring.
Ratio/Interval	<p>Selects Ratio or Interval policy. If multiple policies cover a single process and the ratio or interval policy are different, the smaller value is used. Applies to other monitoring policy definitions as well.</p> <ul style="list-style-type: none"> • Ratio: Accepts a value range of 1–65535. It indicates that monitoring is started among how many requests. The default value is “1” (indicating each request is monitored). • It indicates a period of time (in seconds) that monitoring can be started and is exclusive of ratio settings. Interval: Accepts a value range of 1–65535. The default value is “1”.
Alert Only	Specifies alert monitoring only. No metrics are sent to manager.
Define Alert	Displays the alert definition page. For more information, see Tuxedo Alert Definition .
Transaction List	Select monitored CICS Transaction(s) from the drop-down list or input the FML boolean expression manually.

CICS Terminals Tab

[Table 1-15](#) lists the CICS Terminal tab options.

Table 1-15 CICS Terminals Tab Options

Options	Description
Enable	Enables/disables CICS Terminal monitoring.
Interval	Indicates a period of time (in seconds) that monitoring can be started. It accepts a value range of 1–65535. The default value is 300.
Define Alert	Displays the alert definition page. For more information, see CICS Alert Definition .

Notes: A policy must belong to a single CICS region. Oracle TSAM Plus does not support cross-region policies. When an invalid CICS region is removed, the policies are also removed.

All policies are global configurations that are visible for all users and editable for users with policy management administrator privileges.

It possible for a policy to be modified *simultaneously* by multiple users. The *final changes* may not be seen by other users depending on the console refresh interval.

If the `UBBCONFIG` group setting or the `tuxgroups` in the file `Region-Group.mapping` has been modified after the ART transaction policy is defined, the resource filter might be expired. In that case, policy should be redefined.

IMS Runtime Monitoring Policy

Oracle TSAM Plus provides comprehensive monitoring control of Oracle Tuxedo Application Runtime IMS infrastructure behavior.

This section contains the following topics:

- [Monitoring Policy List Page](#)
- [Create/Edit Policy Page](#)

Monitoring Policy List Page

On the menu bar, click **Policy** and select **IMS Runtime Policy** from the drop-down menu. The **Monitoring Policy List** page appears. It displays existing defined Tuxedo Application Runtime monitoring IMS policies and allows you to view, add, edit, or delete policies.

Button Bar

[Table 1-16](#) list the button bar functions.

Table 1-16 Button Bar

Button	Description
View	From the drop-down menu, you can select the following: <ul style="list-style-type: none">• Columns: Select the columns to shown in the table.• Detach: Displays the table in a separate window.• Reorder Columns: Change how the column order is displayed.
Add	Create a new monitor policy. The maximum policy name character length is 255. You can also add a new policy by right-clicking a Tree Node in IMS System component tree panel.
Delete	Selected monitoring policies are deleted.
Enable	Selected monitoring policies are applied.
Disable	Selected monitoring policies are cancelled.
Edit	Edit selected monitoring policy.
Clone	Clone a new policy from a selected policy.
Import	Imports previously exported policies from a user-specified location.
Export	Exports selected policies in an .xml file to a user-specified location.
Refresh	Updates the policies shown in the Policy List Table.

Policy List Table

The Policy List table displays the following columns:

- **Select:** Allows you to select all or individual policies.
- **Name:** Displays the current IMS policy names.
- **Status:** Displays the IMS policy status.
- **Domian ID:** Displays IMS policy Domain ID.

Create/Edit Policy Page

Click Add, Edit, or Clone, go to the Create/Edit Policy page. At the top of page, enter the name for IMS policy to be created and select a IMS Tuxedo domain.

The Create/Edit Policy page contains the following tabs:

[IMS Transaction Tab](#)

[IMS Program Tab](#)

To use the options listed in each tab, click the **Enable** check box. If enable is not selected, all options in the tab are disabled.

IMS Transaction Tab

This page is used to define policy to report IMS Transaction metric generated by MPP server. [Table 1-17](#) lists the **IMS Transaction** tab options.

Table 1-17 IMS Transaction Tab Options

Options	Description
Enable	Enables/disables IMS Transaction monitoring.
Ratio/Interval	Selects Ratio or Interval to collect metrics.
Alert Only	Specifies alert monitoring only. No metrics are sent to manager.
Define Alert	Displays the alert definition page. For more information, see IMS Alert Definition .
Transaction List	Select monitored IMS Transaction(s) under the Tuxedo domain from the drop-down list or input IMS transactions separated by ";" manually. If no Filter is set, this policy is available for all IMS Transactions under this IMS domain.

IMS Program Tab

This page is used to define policy to report IMS Program metric generated by BMP server. [Table 1-9](#) lists the IMS Program tab options.

Table 1-18 IMS Program Tab Options

Options	Description
Enable	Enables/disables IMS Program metric collection.
Ratio/Interval	Selects Ratio or Interval to collect metrics.
Alert Only	Specifies alert monitoring only. No metric is sent to manager.
Define Alert	Displays the alert definition page. For more information, see IMS Alert Definition .
Program List	Select monitored IMS program(s) under the Tuxedo domain from the drop-down list or input IMS programs separated by ";" manually. If no Filter is set, this policy is available for all IMS programs under this IMS domain.

Tuxedo Metrics Monitoring

Oracle Tuxedo Metrics collection is driven by policy monitoring. Once metrics are collected, you can view them on the related metric pages. Click **Tuxedo Metrics** on the menu bar; the **Tuxedo Metrics** page appears. From the drop-down menu you can select the following:

- [Call Path](#)
- [Call Pattern](#)
- [Service](#)
- [XA Transaction](#)
- [Domain Gateway](#)
- [BRIDGE](#)
- [GWWS](#)

Call Path

Call Path metrics monitoring provides a quick way for you to view the latest call path information. For long-running call path situations, Oracle TSAM Plus allows you to view runtime executions dynamically in real time.

The Call Path Metric window contains the following panels:

- [Call Path Query by Filter Panel](#)
- [Call Path Results Panel](#)

Call Path Query by Filter Panel

[Table 1-19](#) lists the **Call Path Query by Filter** panel options. Results are displayed in the **Call Path Results List** panel.

Table 1-19 Call Path Query By Filter Options

Options	Descriptions
Initiator	<p data-bbox="422 430 1225 482">Specifies the call path initiator type to limit the scope. You can select the following values:</p> <ul data-bbox="422 496 704 840" style="list-style-type: none"><li data-bbox="422 496 516 520">• Type:<ul data-bbox="462 534 704 840" style="list-style-type: none"><li data-bbox="462 534 529 558">– All<li data-bbox="462 572 704 597">– Workstation Client<li data-bbox="462 611 610 635">– Jolt Client<li data-bbox="462 649 637 673">– Native Client<li data-bbox="462 687 690 711">– Domain Gateway<li data-bbox="462 725 704 749">– Web Service Client<li data-bbox="462 763 704 788">– Application Server<li data-bbox="462 802 690 826">– TMQFORWARD <p data-bbox="462 854 717 878">The default value is "All".</p> <p data-bbox="422 906 1225 930">Note: The "Initiator Type" supports multiple selections for specific initiator types.</p> <ul data-bbox="422 944 1225 1076" style="list-style-type: none"><li data-bbox="422 944 999 968">• Domain: Lists all Domains. The default value is "Any".<li data-bbox="422 982 1225 1006">• Machine: Lists all machines for the selected domain. The default value is "Any".<li data-bbox="422 1020 1201 1045">• Server: Lists all servers for the selected machine. The default value is "Any".<li data-bbox="422 1058 986 1083">• Initial Called Service: The first service of the call path. <p data-bbox="422 1104 1225 1156">Note: If the initiator type is set to "Application Server", the "Initial Called Service" value must be host service name.</p> <ul data-bbox="422 1170 1188 1229" style="list-style-type: none"><li data-bbox="422 1170 932 1194">• Client Process Name: The initiator process name.<li data-bbox="422 1208 1188 1232">• User Name: The Tuxedo user name is provided to call initial called service.
Message Tag	<p data-bbox="422 1260 1008 1284">Specifies the Tuxedo Message Tag in the call path segments.</p> <p data-bbox="422 1305 1126 1329">Note: To use this function, the following product versions are required:</p> <ul data-bbox="422 1343 844 1402" style="list-style-type: none"><li data-bbox="422 1343 744 1367">• Oracle Tuxedo 12.1.3 RP013<li data-bbox="422 1381 844 1406">• Oracle TSAM standalone 12.1.3 RP006 <p data-bbox="422 1426 1008 1451">This filter only takes effect when the database type is Oracle.</p>

Table 1-19 Call Path Query By Filter Options

Options	Descriptions
Status	<p>Checks the call path status. You can select the following values:</p> <ul style="list-style-type: none">• Any• Success• Running• Application Failure• System Failure• Any Failure <p>The default value is "Any".</p> <p>Note: "Application Failure" indicates tperrno is TPESVCFAIL. Other errors situations belong to "System Failure"</p>
Time	<p>Specifies a specific time period.</p> <ul style="list-style-type: none">• Any.• Recent: Specifies the recent time period between 0~60 minutes/hours/days. Default value is 1 hour.• Time Period: Specifies a particular time span.
Min. Elapsed	<p>Displays call paths with minimum microseconds spent (0 means all).</p>

Table 1-19 Call Path Query By Filter Options

Options	Descriptions
Max Record	Specifies the call path max number. If the query results are larger than this number, the latest max records are retrieved.
Correlation ID / ECID (Optional)	<p>This option can be used to query call path by Correlation ID or ECID.</p> <p>If the Execution Context ID is inputted, Oracle TSAM Plus use it to filter related call path. ECID Query works only if related Tuxedo application enables ECID.</p> <p>If the call path Correlation ID is inputted, Oracle TSAM Plus uses it to filter. The default value is empty (which indicates a Correlation ID is not used in the query).</p> <p>Note: "Correlation ID" query is exclusive with "Filtering Parameters".</p> <p>The Correlation ID consists of the following field values separated by a space:</p> <ul style="list-style-type: none"> • Tuxedo domain ID: <DOMAINID>:<master machine name>:<IPCKEY> in the UBBCONFIG *RESOURCE section. • Logical Machine ID: LMID in the UBBCONFIG *MACHINES section. • Process Name: <ul style="list-style-type: none"> – “client”: Used for native clients. – “JSH”: Used for Jolt clients. – “WSH”: Used for /WS clients. – <server name>: Used for the server name. • Process ID (pid) • Thread ID • A counter in range of 1..99999999 (starting from 1) • Timestamp <p>Correlation ID Example:</p> <pre>TUXEDO:lcsol18:200401 SITE1 client 18505 1 1 1259031468</pre> <p>ECID Example:</p> <pre>004u3D6nvTI5qYSLMEo2ye0003KL000000</pre>

Saving and Reusing Queries

You can save the frequently-used queries for reusing or as a base for creating similar queries. The Save/Reuse Queries area in the bottom provides the following functions:

- **Load:** Loads a saved query. When you click this button, a Saved Queries list is displayed. Select the desired query, the related conditions are loaded in the Query by Filter panel and are editable. If a historical domain, group, machine, server, or service is already removed or renamed, it is displayed in a read-only text field beneath the corresponding list or shuttle selection. It can be cleared by changing the value of the corresponding list or shuttle selection.
- **New:** Starts a new query. When you click this button, all query condition fields are reset to the default status.
- **Save:** Saves a brand new query or changes in a loaded query. Before saving a query, you must execute the query first. For a new query, when you click this button, type a unique query name and click **OK**. The saved query name is displayed as Current Query Name.
- **Save As:** Saves a loaded query to another name. Before saving a query, you must execute the query first.
- **Delete:** When a query is loaded or saved, this button is activated to delete the loaded query.

Notes:

- If a Tuxedo node (domain/machine/group/server/service) specified in a saved query does not exist anymore (e.g. removed or renamed), it can still be enquired and is displayed in the historical field in the Query by Filter panel when being enquired.
- When a user account is deleted, all his/her queries saved in the system are removed.

Call Path Results Panel

This Call Path Results panel displays server-side call path queries.

There are two sub-panels:

- [Call Path Results List Panel](#)
- [Call Path Details Panel](#)

Call Path Results List Panel

[Table 1-20](#) lists the **Call Path Results List** panel options.

Table 1-20 Call Path Results List Panel

Results	Description
View	From the drop-down menu, you can select the following: <ul style="list-style-type: none">• Columns: Select the columns displayed in the table.• Detach: Displays the table in a separate window.• Reorder Columns: Change how the column order is displayed.
Refresh	Updates the user information list.
Detach	Displays the table in a separate window.

Table 1-20 Call Path Results List Panel

Results	Description
Call Path Data Grid	<p>The data grid holds the entire call path information server-side query. One call path occupies one row. The columns are listed as follows:</p> <ul style="list-style-type: none">• Correlation ID. Click to display single call path details.• Root Service The first ATMI service made in the call path.• Start Time The time stamp for the start of the call path query in the following format: <code>mmdd, yyyy, hh:mm:ss:ms GMT</code>• Status<ul style="list-style-type: none">– Succeeded– Failed• Return Code<ul style="list-style-type: none">– TPOK/0– TPEXXX/tperrno TPEXXX is the error code string format and tperrno is the corresponding numeric value, for example, TPESVCFAIL/11.• Elapsed Time (microsecond) The amount of time passed during the call path query. The unit is seconds with granularity to millisecond. If a call path is completed, the value is the total elapsed time used.• CPU Time (microsecond) The total CPU time used on the call path. The CPU time is the service execution CPU time summary (CPU time for multi-threaded servers cannot be accurately calculated).• User CPU Time (microsecond) The total User CPU time used on the call path. The User CPU time is the service execution User CPU time summary (User CPU time for multi-threaded servers cannot be accurately calculated).• System CPU Time (microsecond) The total System CPU time used on the call path. The System CPU time is the service execution System CPU time summary (System CPU time for multi-threaded servers cannot be accurately calculated).

Table 1-20 Call Path Results List Panel

Results	Description
	<ul style="list-style-type: none">• User return code• GTRID The XA Transaction identifier if the call path is involved in an XA Transaction. It provides the XA Transaction query results if XA Transaction monitoring is enabled.• ECID The minimum ECID if one call path has more than one ECID.• Message Tag The minimum Tuxedo Message Tag in the call path.• Client Address Client IP Address (if available).• User Name The Tuxedo user name is provided to call initial called service.
Status Bar Summary	<p>The summary lists the following information:</p> <ul style="list-style-type: none">• Export to Excel button places the table contents into an MS Excel compatible file.• Total call path in this collection.• Running call path in this collection.• Successfully done call path number.• Application failed call path number.• System failed call path number.

Call Path Details Panel

[Table 1-21](#) lists the **Call Path Details** panel options.

Table 1-21 Call Path Details Panel Options

Results	Description
Call Path Details Data Grid	<p>The data grid holds the entire call path server-side queries. One call path occupies one row. The columns are as follows:</p> <ul style="list-style-type: none">• Message Flow: Message Flow format: ICON Server (tpcall/tpacall/tpforward/tpconnect Root Service) ICON Server (ARTIMS/DLI_OPERATION) <p>Where:</p> <ul style="list-style-type: none">– ICON:<ul style="list-style-type: none">• Send request• Get request• Send reply• Get reply– Server: Server name <p>Note:</p> <ul style="list-style-type: none">• If an Oracle Tuxedo server has invoked tpcall/tpacall/tpforward/tpconnect, the content is displayed in parentheses. For example, BROKER (tpcall DEPOSIT).• Following DLI operations are supported: GU, GHU, GN, GHN, GNP, GHNP, ISRT, DLET, CHNG, PURG, REPL, FLD, OPEN, CLSE, INQY, POS, and PING.• If OCI SQL is monitored, the ORACLE SQL/CMD is appended to Message flow, such as simpserv1 (ORACLE SQL/SELECT).• If it is a tsambegin/tsamend node, "server process name(type/subtype)" is displayed. The type is Extended Monitoring Type specified by tsambegin(), and the subtype is Extended Subsidiary Monitoring Type specified by tsambegin(). For other columns, the detailed CICS command or SQL is displayed. <ul style="list-style-type: none">• Domain: Domain call path step located.• Machine: Machine call path step located.• Group: Group call path step located.• Server: Server call path step located. Oracle TSAM Plus Users Guide• Timestamp: The logging point timestamp.

Table 1-21 Call Path Details Panel Options

Results	Description
	<ul style="list-style-type: none"> • Wait time (microsecond): The IPC queue wait time (in seconds). • Call Flag: The <code>tpcall/tpacall</code> call flags. • Message Size (bytes): The buffer size in bytes. • Message Queued: Number of messages in queue. • IPC queue ID: Identifies the service request queue. The default value is hidden. • Return Code • User Return Code • Extended Monitoring Type: specified by <code>tsambegin()</code>. • Extended Subsidiary Monitoring Type: specified by <code>tsambegin()</code>. • ECID: Execution Context ID. • Tuxedo Message Tag • Others: <ul style="list-style-type: none"> – LDOM – RDOM – Local GTRID – MSGCVTTIME: Message Web service conversion time. – SQL, AMOUNT, and RETURNCODE. <ul style="list-style-type: none"> • SQL: Original SQL content after <code>EXEC SQL</code> • AMOUNT: This number makes sense for <code>insert/update/delete/select</code> only. For <code>insert/delete/update</code>, it indicates the number of rows affected; for <code>select</code>, it indicates the number of rows putted into the user buffers. AMOUNT is 0 when <code>sql open cursor</code> is monitored. • RETURNCODE: The return code retrieved from the OCI call return code by TSAM Plus agent. Unlike <code>SQLCODE</code>, RETURNCODE can only be got from TSAM Plus agent code. • Web service conversion time

Table 1-21 Call Path Details Panel Options

Results	Description
Status Bar	<p>Displays the current selected call path Correlation ID.</p> <p>The Export to Excel button allows you to place the table contents into an MS Excel compatible file.</p> <p>Note: When exporting to Excel, if the callpath/callpattern tree is collapsed, only the nodes that are not collapsed are exported to excel file. If you want to export the full callpath/callpattern tree, you must fully expand it and then click Export to Excel.</p>

Notes: If multiple async calls are made in one process (for example, in a service two `tpacalls` are made), the order is based on the service invocation sequence for the reply instead of the timestamp. This allows you to easily correlate requests and reply in an "ordered" manner.

For GWWTDOMAIN, both local domain and remote domain are displayed. LDOM/RDOM information is displayed in the domain gateway cell.

For BRIDGE, the message full stages are supported (same as GWTDOMAIN).

Call Pattern

Call Pattern monitoring provides a quick way for you to view the Call Pattern information. The Call Pattern Metric window contains the following panels:

- [Call Pattern Query by Filter](#)
- [Call Pattern Results Panel](#)

Call Pattern Query by Filter

[Table 1-22](#) lists the **Call Pattern Query by Filter** panel options.

Table 1-22 Call Pattern Filter Panel

Options	Description
Initiator	<ul style="list-style-type: none">• Domain: Lists all Domains. The default value is "Any".• Machine: Lists all machines for the selected domain. The default value is "Any".• Group: Lists all groups for the selected machine. The default value is "Any".• Server: Lists all servers for the selected machine. The default value is "Any".• Initial Called Service: The first service of the call path.• Client Process Name: The initiator process name.• User Name: The Tuxedo user name is provided to call initial called service.
From/To	Time Span. This option is available only when "Recent" is not specified.
Recent	Specifies the recent time period between 0~60 minutes/hours/days. The default value is 1 hour. This option is available only when "From/To" is not specified.
Minimum Request Number	The minimum number of requests that form a pattern. Usually a call pattern contains a large number of calls. This threshold prevents listing a pattern with a small number of calls.
Service Pattern	Checked: Displays only service call path pattern Unchecked: Displays service call path pattern including detailed location and GWTDOMAIN information

Click the **Query** button; the **Call Pattern Result List** panel appears.

Note: For Oracle Database, there is a lag time of minutes specified in Call Pattern Calculation Delay before you can query call patterns in one hour or day. That is, you can only query the call patterns occurred in one hour or day after the Call Pattern Calculation Delay minutes of this hour or day. For example, suppose the Call Pattern Calculation Delay is 30 minutes, if you select to query the call patterns occurred from 9 to 10 o'clock, you can

actually query the desired information after 10:30, similarly, if you select to query the call patterns occurred yesterday, you can actually query the desired information after today 00:30.

Saving and Reusing Queries

You can save the frequently-used queries for reusing or as a base for creating similar queries. For more information, refer to [Saving and Reusing Queries](#).

Call Pattern Results Panel

The Call Pattern Results panel displays server-side Call Pattern queries. There are two sub-panels:

- [Call Pattern Results List Panel](#)
- [Call Pattern Details Panel](#)

Call Pattern Results List Panel

[Table 1-23](#) lists the **Call Pattern Results List** panel options.

Table 1-23 Call Pattern Results List Panel Options

Options	Description
View	From the drop-down menu, you can select the following: <ul style="list-style-type: none">• Columns: Select the columns displayed in the table.• Detach: Displays the table in a separate window.• Reorder Columns: Change how the column order is displayed.
Refresh	Updates the user information list.
Detach	Displays the table in a separate window.

Table 1-23 Call Pattern Results List Panel Options

Options	Description
Call Pattern Data Grid	<p>The data grid holds the entire call path server-side queries. One call path occupies one row. The columns are as follows:</p> <ul style="list-style-type: none">• Description: Brief information for this call pattern. Clicking it displays call pattern details as follows.• Total Number: Total number of call paths in a call pattern.• Success Number: Successful number of call paths in a call pattern.• Application Failure Number: Application failure number of call paths in a call pattern.• System Failure Number: System failure number of call paths in a call pattern.• Average Elapsed Time (microsecond): Average call path elapsed time in a call pattern.
Status Bar Summary	<ul style="list-style-type: none">• Total call pattern in this collection. The Export to Excel button allows you to place the table contents into an MS Excel compatible file. <p>Note: When exporting to Excel, if the callpath/callpattern tree is collapsed, only the nodes that are not collapsed are exported to excel file. If you want to export the full callpath/callpattern tree, you must fully expand it and then click Export to Excel.</p>

Call Pattern Details Panel

Click any row in the **Call Pattern** list, the selected **Call Pattern** details are displayed. [Table 1-24](#) lists the **Call Pattern Details** panel options.

Table 1-24 Call Pattern Details Panel

Options	Description
Call Pattern Details Data Grid	<p>The tree table data grid holds the entire call path details queried from server side. The columns are as follows:</p> <ul style="list-style-type: none">• Message Flow: Message Flow format:<ul style="list-style-type: none">– ICON Service <p>Where:</p> <ul style="list-style-type: none">– ICON:<ul style="list-style-type: none">• Send request Root Service: Initial service at Call Pattern For example, BR_ACNT_SUM <ul style="list-style-type: none">• Domain: Domain call pattern step located. Displayed when Service Pattern is unchecked.• Machine: Machine call pattern step located. Displayed when Service Pattern is unchecked.• Group: Group call pattern step located. Displayed when Service Pattern is unchecked.• Server: Server call pattern step located. Displayed when Service Pattern is unchecked.
Status Bar	<p>Status Bar displays the current selected call pattern index in the Call Pattern List panel.</p> <p>The Export to Excel button allows you to place the table contents into an MS Excel compatible file.</p>

Service

Service Monitoring allows you to monitor Oracle Tuxedo services. Click **Tuxedo Metrics** and select **Service** from the drop-down menu; the **Service** page appears.

The filtering options panel allows you to specify supply information to the services you want monitored. On the left, there are drop-down lists of Domain, Machine, Group and Server. There are two modes for the service selection, select the most active services or particular services on the right. [Table 1-25](#) lists the **Service Selection** options.

Table 1-25 Service Selection Options

Options	Description
Services Selection	<ul style="list-style-type: none">• Domain: Lists all Domains. The default value is “Any”.• Machine: Lists all machines for the selected domain. The default value is “Any”.• Group: Lists all server groups for the selected machine. The default value is “Any”.• Server: Lists all servers for the selected machine. The default value is “Any”.• User Name: The Tuxedo user name is provided to call initial called service.

Table 1-25 Service Selection Options

Options	Description
Monitoring Mode	<p data-bbox="357 388 727 413">Four monitoring modes are supported:</p> <ul data-bbox="357 427 1170 1576" style="list-style-type: none"><li data-bbox="357 427 1170 479">• Most Active (Live): Allows you to query the most active services in the latest refresh-time window<li data-bbox="357 493 1170 545">• Most Active (Historical): Allows you to query the most active service over a long time interval and displays the service metrics distribution during that time span.<li data-bbox="357 558 1170 1576">• Selected Services (Live): Allows you to monitor service dynamic execution status in the latest refresh-time window. When selected, the Available Services/Selected Services panel appears.<ul data-bbox="395 656 1170 1506" style="list-style-type: none"><li data-bbox="395 656 1170 708">– Available Services/Selected Services: Selects the services to be monitored.<li data-bbox="395 722 1170 843">– Show services with same name as one service: If checked, specifies the query only uses the service name as the query key. If unchecked, the services are queried using the service location for same name service.<li data-bbox="395 857 1170 909">– Aggregation Time: Applies to alive related queries. The algorithm is time span.<li data-bbox="395 923 1170 1010">– Refresh Interval: Specifies the refresh aggregation interval for service metrics computing. It only applies to live monitoring related query.<li data-bbox="357 1024 1170 1576">• Selected Services (Historical): Allows you to query services execution status during a long time span. A time window is needed for aggregation purpose. When selected, the Available Services/Selected Services panel appears.<ul data-bbox="395 1121 1170 1506" style="list-style-type: none"><li data-bbox="395 1121 1170 1173">– Available Services/Selected Services: Selects the services to be monitored.<li data-bbox="395 1187 1170 1308">– Show services with same name as one service: If checked, specifies the query only uses the service name as the query key. If unchecked, the services are queried using the service location for same name service.<li data-bbox="395 1322 1170 1374">– From, To: Time period. For historical query. This option is available only when “Recent“ is not specified.<li data-bbox="395 1388 1170 1475">– Recent: Specifies the recent time period between 0~60 minutes/hours/days. The default value is 1 hour. This option is available only when “From/To“ is not specified.<li data-bbox="395 1489 1170 1541">– Time Window: Specifies the time increment (in seconds). The default value is 3600.

Click **Query**; the query results are displayed in the **Service Monitoring Results** panel. [Table 1-26](#) lists the **Service Monitoring Results** panel options.

Table 1-26 Service Monitoring Results Panel Options

Options	Description
Metrics	<p>From the drop-down list you can select the following service monitoring metrics:</p> <ul style="list-style-type: none"> • Average Execution time (microsecond) • Success Number • Failure Number • Max Message Size (bytes) • Min message size (bytes) • Average CPU time (microsecond) • Average System CPU time (microsecond) • Average User CPU time (microsecond) • Average Message Queue Length • Average Message Wait Time in Queue (millisecond) <p>For "CPU time", only single threaded servers are calculated (CPU time for multi-threaded servers cannot be accurately calculated).</p> <p>The Pause/Resume button allows you to stop the data refresh temporarily. The default is Pause (which indicates that a current refresh is in progress). Clicking it will Resume and refresh stops. Clicking it again restores the status.</p>
Chart/Table	<p>Displays results in either a chart or table. The default view is Chart (either line or bar chart). One service is one series. The horizontal axis represents the time series; the vertical axis represents the selected metrics.</p> <p>In table view, the Export to Excel button allows you to place the table contents into an MS Excel compatible file.</p>

Saving and Reusing Queries

You can save the frequently-used queries for reusing or as a base for creating similar queries. For more information, refer to [Saving and Reusing Queries](#).

XA Transaction

There are two ways to initiate XA Transaction queries: you can specify an XA Transaction identifier, or you can query using filtering parameters. These two ways are exclusive. [Table 1-27](#) lists the **XA Transaction Query by Scope** filtering options.

Table 1-27 “XA Transaction Query by Scope Panel Options

Options	Description
Initiator Type	<p>There XA Transaction monitoring scenarios:</p> <ol style="list-style-type: none">1. If the monitoring policy is applied to an XA Transaction initiator, monitoring initiated with the XA Transaction path propagation (similar to Call Path).2. If monitoring is initiated for particular processes only (such as TMS without initiator involved), propagation does not take place and only the monitored XA Transaction calls are reported. The "Initiator Type" applies to the first situation, so that a specific XA Transaction can be retrieved. It has the following values:<ul style="list-style-type: none">– All– Native Client– Workstation Client– Jolt Client– Domain Gateway– Application Server <p>The default value is "All" and the second monitoring scenario must use the default setting.</p>
Status	<p>Allows you to select the following values:</p> <ul style="list-style-type: none">• Any• Succeeded• Failed
Query for latest	Query under latest time (in seconds).

Table 1-27 “XA Transaction Query by Scope Panel Options

Options	Description
Query during	Query XA transactions during a specific time period.
Query by GTRID	Query an XA transaction by using an XA transaction identifier. For example x0x46524a28 x1. For more information, see How the System Tracks Distributed Transaction Processing in the Oracle Tuxedo User Documentation. The Exact Match checkbox is under the text field. If checked, the whole identifier string must be matched, otherwise all the XA transaction IDs that contain the specified string are listed.

Click **Submit**; the XA Transaction results appears in the **XA Transaction Result List** panel.

[Table 1-28](#) lists the **XA Transaction Results List** panel options.

Table 1-28 XA Transaction Results List Panel Options

Options	Description
GTRID	Global Transaction ID.
Initiator	The initiated process where the transaction starts.
Status	Successful transactions indicated by Failed transactions indicated by
Start Time	Time when the XA Transaction started.
Duration (microsecond)	Total transaction time span.
Execution Time (microsecond)	Total transaction time per Oracle Tuxedo service.
Parent GTRID	Displays the GTRID of the previous domain if the transaction was propagated from another Oracle Tuxedo domain.

When you click on a transaction item in the list, the transaction path, related call path ID and transaction details are displayed in the XA Transaction Path, Corresponding Call Path Correlation ID and XA Transaction Detail panels respectively.

The XA Transaction Path shows the XA Transaction network in a tree structure similar to the Component Tree. In the XA Transaction Path tree, each node represents one domain XA Transaction. If multiple /T domains are involved in the XA Transaction, the XA transaction spread path can be easily observed.

In the Corresponding Call Path Correlation ID panel, the call path(s) related to the XA transaction are listed. You can include multiple call path correlation IDs in an XA Transaction. To do so, you must define an XA Transaction policy element and a call boundary call path policy element. If the two elements are applied to one process, one XA Transaction monitoring process includes multiple call paths. Clicking on the call path ID displays the call path detail in the Call Path Metric.

Table 1-29 lists the **XA Transaction Detail** panel options.

Table 1-29 XA Transaction Detail Panel Options

Options	Description
XA Routine	The transaction routine names are as follows: <ul style="list-style-type: none"> • tpbegin • tpcommit • tprollback • tpabort • xa_commit • xa_start • xa_prepare • xa_rollback • xa_end • tms_msg_xxx (for GWTDOMAIN XA Transaction activities) • xa_msg_xxx (for GWTDOMAIN XA Transaction activities)
Return Code	The XA Routine return code.
Machine/Group/Process	The XA Routine process information. The process can be an Oracle Tuxedo server or a client.
Start Time	The time when the XA Routine started.
Execution Time (microsecond)	The elapse time used for the XA Routine.

Domain Gateway

A monitoring project may contain multiple domain gateways; however, one monitoring chart can only monitor one local domain gateway (due to data source consistency). Click **Tuxedo Metrics** and select **Domain Gateway** from the drop-down menu; the **Domain Gateway Metrics** page appears. The Domain Gateway page contains two panels:

- [Domain Gateway Selection](#)
- [Domain Gateway Monitoring](#)

Domain Gateway Selection

[Table 1-30](#) lists the **Domain Gateway Selection** panel options.

Note: Domain Gateway only supports GWTDOMAIN.

Table 1-30 Domain Gateway Selection Panel Options

Options	Description
Domain	Lists all Domains. Required.
Local Access Point	This drop-down list contains the local domain gateway configured in "Gateway" monitoring. The format is as follows: domain id/group/server id. Note: One group can only have one gateway instance.
Available Remote Access Point/Selected Remote Access Point	The remote domain gateway link connected with the selected local domain gateway. They are the RDOM in DMCONFIG. Multiple remote domains can be selected.
Monitoring Mode	<ul style="list-style-type: none">• Live<ul style="list-style-type: none">– Aggregation Time: Applies to alive related query. The algorithm is time span (in seconds).– Refresh Interval. Specifies the refresh interval for live monitoring (in seconds).• Historical<ul style="list-style-type: none">– From/To: time period for historical query.– Time Window. Specifies the aggregation window for historical query (in seconds).

Domain Gateway Monitoring

Table 1-31 lists the **Domain Gateway Monitoring** panel options.

Table 1-31 Domain Gateway Monitoring Panel Options

Options	Description
Metrics	From the drop-down list you can select the following metrics: <ul style="list-style-type: none">• Network Message Number• Network Message Bytes• Network Pending Number• Network Pending Bytes• Network Outstanding Requests
Chart/Table	Displays results in either a chart or table. The default view is Chart (either line or bar chart). One service is one series. The horizontal axis represents the time series; the vertical axis represents the selected metrics. In table view, the Export to Excel button allows you to place the table contents into an MS Excel compatible file.

BRIDGE

Similar to [Domain Gateway](#).

Note: There is no "Network Outstanding Requests" metric.

GWWS

Click **Tuxedo Metrics** and select **GWWS** from the drop-down menu; the **GWWS Metric** page appears. The GWWS Metric page contains two panels:

- [GWWS Query by Filter](#)
- [GWWS Live/History Monitoring](#)

GWWS Query by Filter

Table 1-31 lists the **GWWS Query by Filter** options.

Table 1-32 GWWS Query by Filter Panel Options

Options	Description
Domain	Lists all Domains. Required.
GWWS	GWWS server list. The default is "All".
Monitoring Mode	<ul style="list-style-type: none">• Live<ul style="list-style-type: none">– Aggregation Time: Applies to alive related query. The algorithm is time span (in seconds).– Refresh Interval. Specifies the refresh interval for live monitoring (in seconds).• Historical<ul style="list-style-type: none">– From/To: time period for historical query.– Time Window. Specifies the aggregation window for historical query (in seconds).

GWWS Live/History Monitoring

[Table 1-33](#) lists the **GWWS Live/History Monitoring** options:

Table 1-33 GWWS Live/History Monitoring Panel Options

Options	Descriptions
Metrics	<p data-bbox="548 427 1147 453">From the drop-down list you can select the following metrics:</p> <ul data-bbox="548 465 1147 939" style="list-style-type: none"><li data-bbox="548 465 1147 491">• Active Thread Numbers (average active thread numbers)<li data-bbox="548 499 1147 526">• Average Inbound Process Time (microsecond)<li data-bbox="548 534 1147 560">• Average Outbound Process Time (microsecond)<li data-bbox="548 569 1147 595">• Inbound One-Way Failure Number<li data-bbox="548 604 1147 630">• Inbound One-Way Success Number<li data-bbox="548 638 1147 664">• Inbound RPC Failure Number<li data-bbox="548 673 1147 699">• Inbound RPC Success Number<li data-bbox="548 708 1147 734">• Outbound One-Way Failure Number<li data-bbox="548 743 1147 769">• Outbound One-Way Success Number<li data-bbox="548 777 1147 803">• Outbound RPC Failure Number<li data-bbox="548 812 1147 838">• Outbound RPC Success Number<li data-bbox="548 847 1147 873">• Inbound Pending Request<li data-bbox="548 881 1147 907">• Outbound Pending Request

CICS/IMS/Batch Runtime Metrics Monitoring

Metrics collection is driven by policy monitoring. Once metrics are collected, you can view them on the related metric pages. The Tuxedo Application Runtime Metrics page includes four panels:

To access the Tuxedo Application Runtime Metrics page, click **Tuxedo Application Runtime Metrics** on the menu bar. From the drop-down menu you can select the following:

- [CICS Transactions](#)
- [CICS Terminals](#)
- [Batch Jobs](#)
- [IMS Transactions](#)
- [IMS Programs](#)
- [IMS System Traces](#)

CICS Transactions

CICS Transactions monitoring provides a quick way for viewing the latest CICS Transaction information.

The CICS Transactions page contains the following panels:

- [CICS Transactions Query by Filter Panel](#)
- [CICS Transactions Live/History Monitoring Panel](#)

CICS Transactions Query by Filter Panel

[Table 1-34](#) lists the **CICS Transactions Query by Filter** options.

Table 1-34 CICS Transactions Query By Filter Options

Options	Description
CICS Region	Lists all CICS regions. The default value is "All".
Transaction Class	Lists all Transaction classes. The default value is "All".
Transaction	Lists all Transactions. The default value is "All".
Monitoring Mode	<ul style="list-style-type: none">• Live<ul style="list-style-type: none">– Aggregation Time: Applies to alive related query. The algorithm is time span (in seconds).– Refresh Interval. Specifies the refresh interval for live monitoring (in seconds).• Historical<ul style="list-style-type: none">– From/To: time period for historical query.– Time Window. Specifies the aggregation window for historical query (in seconds).

CICS Transactions Live/History Monitoring Panel

[Table 1-35](#) lists the **CICS Transactions Live/History Monitoring Graph** tab metrics.

Table 1-35 CICS Transactions Live/History Monitoring Options

Options	Description
Chart Control	From the drop-down list you can select the following metrics: <ul style="list-style-type: none">• Number of Transaction Calls• Average Execution Time (microsecond)• Average CPU Time (microsecond)• Average System CPU Time (microsecond)• Average user CPU Time (microsecond)
Chart/Table	Displays results in either a chart or table. The default view is Chart (either line or bar chart). One service is one series. The horizontal axis represents the time series; the vertical axis represents the selected metrics. In table view, the Export to Excel button allows you to place the table contents into an MS Excel compatible file.

CICS Terminals

CICS Terminals metrics provides a quick way for you to view the latest CICS Terminals information.

The CICS Terminals window contains the following panels:

- [CICS Terminals Query by Filter Panel](#)
- [CICS Terminals Live/History Monitoring Panel](#)

CICS Terminals Query by Filter Panel

[Table 1-36](#) lists the **CICS Terminals Query by Filter** options.

Table 1-36 CICS Terminals Query By Filter Options

Options	Description
CICS Region	Lists all CICS regions. The default value is "All".
Monitoring Mode	<ul style="list-style-type: none">• Live<ul style="list-style-type: none">– Minimum Policy Interval: Applies to alive related query. It is the minimum value among all Tuxedo Application Runtime monitoring policies defined (in seconds).– Refresh Interval. Specifies the refresh interval for live monitoring (in seconds).• Historical<ul style="list-style-type: none">– From/To: time period for historical query.– Time Window: Specifies the aggregation window for historical query (in seconds).– Since CICS Region Start Checkbox: If checked, From/To is replaced by Start Time/Shutdown Time.– Start Time/Shutdown Time: CICS region start/shutdown time.

CICS Terminals Live/History Monitoring Panel

[Table 1-37](#) lists the CICS Terminals Live/History Monitoring panel options.

Table 1-37 CICS Terminals Live/History Monitoring Graph Options

Options	Description
Chart Control	From the drop-down list you can select the following metrics: <ul style="list-style-type: none">• Successful Transaction Number• Failed Transaction Number• Session Number
Chart/Table	Displays results in either a chart or table. The default view is Chart (either line or bar chart). One service is one series. The horizontal axis represents the time series; the vertical axis represents the selected metrics. In table view, the Export to Excel button allows you to place the table contents into an MS Excel compatible file.

Batch Jobs

Batch Jobs allows you to monitor the running status of the Batch Jobs. It contains two panels: Query by filter and Monitor view.

JES Metrics Query by Filter Panel

The filtering condition panel let user supply information to specify the jobs to be observed in the JES system. On the left, there are dropdown lists for JES application name, Job classes, Job priorities and Job owners, and one edit box for Job name.

There are two modes for the JES metrics monitoring, live monitoring and history monitoring.

[Table 1-38](#) lists the UI components and their relationships.

Table 1-38 JES Metrics Query by Filter Panel

Elements	Description
JES application name	Specifies which JES system the Job is executed. The single choice dropdown list contains only live/active ones, mandatory. Same as the JES tree node name.
Job classes	Specifies one or more classes
Job name	Specifies a search a string contained in the job name.
Job priorities	Specifies one or more priorities

Table 1-38 JES Metrics Query by Filter Panel

Elements	Description
Job owners	Specifies one or more owners
Monitoring Mode	Two monitoring modes are supported: Live and Historical <ul style="list-style-type: none">• Live Monitoring. It allows user to monitor the JES system execution status in a latest refresh window.• History Monitoring. It allows user to query JES system execution status during a long time span. A time window is needed for aggregation purpose.
Aggregation Time	Applies to history monitoring mode. The algorithm is time span
Refresh Interval	Specifies the refresh aggregation interval for JES system metrics computing. It only applies to live monitoring.

When the conditions filing is complete, click **Submit**; the monitoring results page appears on the right.

JES Monitor View

Table 1-39 JES Monitor View

Elements	Description
Chart Control	<p>The "Metrics" drop down list let user select the metric aspect interested in the chart,</p> <ul style="list-style-type: none">• Successful job number in the Aggregation Time Window• Failed job number in the Aggregation Time Window• Waiting job number in different queues and the sum of all queues• Average CPU Time:<ul style="list-style-type: none">– Average user CPU time for successful jobs in milliseconds int the Aggregation Time Window– Average system CPU time for successful jobs in milliseconds int the Aggregation Time Window <p>Note: Failed jobs do not have "Average CPU Time"</p> <p>Button "Pause/Resume" allows you to temporarily stop the data refresh . The default is "Pause" which means the current refresh is in progress. Clicking it again will "Resume" and the refresh stops. Clicking it again restores the status.</p>
Chart/Table	<p>Chart/Table will show results in style either chart or table. The chart is can be line style or bar chart. One kind of job class is one series. The horizontal axis is time series and the vertical axis is for the metrics selected.</p>

The **Export** button allows you to save the table contents to an EXCEL compatible file.

IMS Transactions

IMS Transactions monitoring provides a quick way for viewing the latest IMS Transactions information and shows IMS transactions metrics generated by MPP server.

The IMS Transactions page contains the following panels:

- [IMS Transaction Query by Filter Panel](#)
- [IMS Transaction Live/History Monitoring Panel](#)

IMS Transaction Query by Filter Panel

[Table 1-40](#) lists the **IMS Transaction Query by Filter** options.

Table 1-40 IMS Transaction Query By Filter Options

Options	Description
Domain	Lists all IMS domains. The default value is empty.
Transaction Class	Lists all Transaction classes. The default value is empty.
Transaction	Lists all Transactions. The default value is empty.
Monitoring Mode	<ul style="list-style-type: none">• Live<ul style="list-style-type: none">– Aggregation Time: Applies to alive related query. The algorithm is time span (in seconds).– Refresh Interval. Specifies the refresh interval for live monitoring (in seconds).• Historical<ul style="list-style-type: none">– From/To: time period for historical query.– Time Window. Specifies the aggregation window for historical query (in seconds).

IMS Transaction Live/History Monitoring Panel

[Table 1-41](#) lists the **IMS Transaction Live/History Monitoring Graph** tab metrics.

Table 1-41 IMS Transaction Live/History Monitoring Options

Options	Description
Chart Control	<p>From the drop-down list you can select the following metrics:</p> <ul style="list-style-type: none"> • Number of Transaction Calls • Average Execution Time (microsecond) • Average CPU Time (microsecond) • Average System CPU Time (microsecond) • Average user CPU Time (microsecond) • Average user DLI Time (microsecond)
Chart/Table	<p>Displays results in either a chart or table. The default view is Chart (either line or bar chart). One transaction is one series. The horizontal axis represents the time series; the vertical axis represents the selected metrics.</p> <p>In table view, the Export to Excel button allows you to place the table contents into an MS Excel compatible file.</p>

IMS Programs

IMS Programs monitoring provides a quick way for viewing the latest IMS Programs information and shows IMS programs metrics generated by BMP server.

The IMS Programs page contains the following panels:

- [IMS Program Query by Filter Panel](#)
- [IMS Program Live/History Monitoring Panel](#)

IMS Program Query by Filter Panel

[Table 1-42](#) lists the **IMS Program Query by Filter** options.

Table 1-42 IMS Program Query By Filter Options

Options	Description
Domain	Lists all IMS domains. The default value is empty.

Table 1-42 IMS Program Query By Filter Options

Options	Description
Program	Lists all programs. The default value is empty.
Monitoring Mode	<ul style="list-style-type: none"> • Live <ul style="list-style-type: none"> – Aggregation Time: Applies to alive related query. The algorithm is time span (in seconds). – Refresh Interval. Specifies the refresh interval for live monitoring (in seconds). • Historical <ul style="list-style-type: none"> – From/To: time period for historical query. – Time Window. Specifies the aggregation window for historical query (in seconds).

IMS Program Live/History Monitoring Panel

[Table 1-43](#) lists the **IMS Program Live/History Monitoring** tab metrics.

Table 1-43 IMS Transaction Live/History Monitoring Options

Options	Description
Chart Control	<p>From the drop-down list you can select the following metrics:</p> <ul style="list-style-type: none"> • Number of Program Calls • Average Execution Time (microsecond) • Average CPU Time (microsecond) • Average System CPU Time (microsecond) • Average user CPU Time (microsecond) • Average user DLI Time (microsecond)
Chart/Table	<p>Displays results in either a chart or table. The default view is Chart (either line or bar chart). One program is one series. The horizontal axis represents the time series; the vertical axis represents the selected metrics.</p> <p>In table view, the Export to Excel button allows you to place the table contents into an MS Excel compatible file.</p>

IMS System Traces

The IMS System Traces page contains the following panels:

- [Search Panel](#)
- [Query Result List Table](#)
- [Detailed Information Panel](#)

Search Panel

[Table 1-44](#) lists the query filters on the Search panel.

Table 1-44 Query Filters

Options	Description
Domain	Lists all IMS domains. The default value is empty.
Machine	Lists all the machines belong to the selected domains. The default value is empty.
Group	Lists all the machines belong to the selected machines. The default value is empty.
Server	Lists all the machines belong to the selected groups. The default value is empty.
Transaction Class	1 ~ 999. The default value is empty.
Transaction	Lists all the transactions belong to the selected domains. The default value is empty.
Program Name	Lists all the programs belong to the selected domains. The default value is empty.
Region Type	MPP and BMP. The default value is empty.
From / To	The transaction/program ending time span. The default value is latest 24 hours.
Max Queried Out Record	Limits the returned record number. Default is 500. Too large value in this field may cause browser out of response

Clicking on the **Reset** button resets all the query filters (except Max Queried Out Record) to the initial status.

Query Result List Table

When clicking the **Query** button, all the Transaction/Program collect instances met the search criteria are listed in the Query Result List table. Click any of the entries listed in the Query Result List table to display the detailed information in the Detailed Information panel.

Clicking on the **Refresh** button executes the last query once more and loads the latest DB data in the list table.

Detailed Information Panel

This panel displays the detailed information of the entry selected in the Query Result List Table. [Table 1-45](#) lists the detailed information.

Table 1-45 Detailed Information

Name	Description
Domain	Tuxedo domain ID
Machine	Machine name
Group	Group name
Server	Server name
Server ID	Tuxedo server ID
PID	Server PID
Transaction Class	IMS transaction class
Transaction Name	IMS transaction name
Program Name	IMS program name
Logic Terminal	Logic terminal name
Region Type	The IMS region type, MPP or BMP
User ID	OS user ID
Start Time	IMS transaction/program start time point

Table 1-45 Detailed Information

Name	Description
End Time	IMS transaction/program stop time point
Elapsed Time	IMS transaction/program elapsed time from start to end (microseconds)
System CPU Time	System CPU time (microseconds)
User CPU Time	User CPU time (microseconds)
PSB Name	IMS PSB name
Program Elapsed Time	IMS program elapsed time (microseconds)
Current SPA Size	SPA size
Abend Code	Program abend code
Elapsed Time in DL/I	Total time elapsed in DL/I (microseconds)
Enqueue Number	Message number pushed in /Q
Dequeue Number	Message number popped from /Q
DLI Summary Information	The DLI summary information is presented by a table. One transaction/program may contain one or multiple DLI entries. There are four columns of the table, DL/I Event Name, DL/I Type, DL/I Invoked Count, and DL/I Total Duration (microsecond).

CICS/IMS/Batch Management

- [Batch Jobs Submit](#)
- [Batch Jobs Search/Display](#)
- [Batch Syslog Search/Display](#)
- [CICS Resources View/Edit](#)
- [IMS Resources View/Edit](#)

Batch Jobs Submit

JES Job submit provides a quick way for you to submit a job through the TSAM Plus console. For more information, see [Oracle ART Runtime documentation](#).

Notes: To enable JES job submit and monitoring, you must set the `JESMONITOR` environment variable to yes before you start the LMS server and set the "JOBREPOSITORY" value in the JES configuration file to specify the path of the job repository.

If the "JOBREPOSITORY" value is not in the JES configuration file, the list panel lists the JOB in the `APPDIR`.

The JES job submit window contains the following panels:

- [JES Application Selection Panel](#)
- [Job File Information List Panel](#)

JES Application Selection Panel

Click **List Available Jobs**, all available job script files are listed in the Job File Information List Panel" if the selected JES application is alive.

Job File Information List Panel

[Table 1-46](#) lists the JES Job File Information List Panel options.

Table 1-46 Job File Information List Panel Options

Options	Description
Script Name	Job script file name.
Name	Job name.
Priority	Job priority.
Class	Job class. Value may be [A-Z], [0-9]
Restart Option	The name of the step to use to restart the job

Table 1-46 Job File Information List Panel Options

Type run	Indicates what should be done with the job. Choose one of the following: COPY - Copy the job directly in an output stream to sysout. HOLD - The system should hold the job. JCLHOLD - JES should hold the job. SCAN - Scan JCL for syntax errors only.
Version	The EJ2 runtime engine version.
Start label	The label of the first phase to be started
EJ2 Option	Enter the EJ2 Option you want to start the job with.
Shell Option	Enter the Shell Option you want to start the job with.

Select an item, and click **Submit**. If the job is submitted successfully, the display message appears as follows:

"Submit of Job xxxx was successful", where "xxxx" is the job id. Click **View Job** to see a detailed message. If it fails, an error message is displayed.

Batch Jobs Search/Display

JES Jobs metrics provides a quick way for you to view the latest JES jobs information.

The JES jobs window contains the following panels:

- [JES Jobs Query by Filter Panel](#)
- [Jobs Query Results List Panel](#)
- [Job Detail Information Panel](#)

JES Jobs Query by Filter Panel

[Table 1-47](#) lists the **JES Jobs Query by Filter** options.

Table 1-47 JES Jobs Query By Filter Options

Elements	Description
JES application	Specifies which JES system the Job is executed,. The single choice dropdown list contains only live/active ones, mandatory. Same as the JES tree node name.
Job classes	Specifies one or more classes
Job name	Specifies a search a string contained in the job name.
Job priority	Specifies one or more priorities
Job owners	Specifies one or more owners separated by a semi-colon (;) or a coma (,).
Job status	Specifies one or more of the following status categories: <ul style="list-style-type: none">• EXECUTING• CONVING• WAITING• DONE• FAIL• HOLD_WAITING• HOLD_CONVING• INDOUBT• DISCARD
Submitted in latest _ seconds	Query the most recent Jobs submitted in latest a few seconds
Submit during	Query JES Jobs submitted during a specific time period.

You can also query the exact job ID by using 'Query by Job ID' subform. Fill in 'Job ID' and click **Query**; if found, the corresponding job is listed in the result.

Jobs Query Results List Panel

[Table 1-48](#) lists the **JES Jobs Query Results List** table columns.

Table 1-48 'Jobs Query Results List Table Columns

Sort Options	Description
Name	Job name.
ID	Job ID
Node	Machine name
Owner	The person that submitted the job
Priority	Job priority.
Current Queue	The current queue where the job is placed.
Class	Job class. Value may be [A-Z], [0-9]
Submit Time	Time when the job was submitted.
End Time	Time when job completes.
Status	Job status.

Certain actions can be made to the queried out job list entries. Besides the **Refresh** button, **Cancel**, **Purge**, **Hold**, and **Release** actions can be used on selected jobs. The job cancel and purge behavior is the same same as the `artjesadmin` command. The command return status is displayed in a popup window.

Notes: A job can be held only when the status is `CONVING` or `WAITING`.

A job can be released only when the job status is `HOLD_WAITING`.

A job can be purged only when the job is not in the `PURGE` queue.

Two additional actions/buttons (**Job Logs** and **Job Sysouts**) are located on the button bar that provide job log and job sysout information.

Notes: If job logs or job sysouts exceed 10, 000 bytes, the file is truncated (that is, only the last 10,000 bytes are retained).

Purged jobs do not have Job Logs and Sysouts to view

All records in the job list table can be exported to a `.xls` file by clicking the **Export to Excel** button below the list table.

Note: If you are using Internet Explorer, and IE security is set above medium, a warning message pops up when you click the **Export to Excel** button. Click the export button again to ignore the warning.

Choosing Monitoring Mode

You can change monitoring to auto refresh mode by specifying the Refresh Interval in the query result list panel toolbar; auto refresh is disabled by default. The refresh rate can be set as follows:

- 1 minute
- 5 minutes
- 10 minutes
- 30 minutes
- 60 minutes
- 120 minutes

Job Detail Information Panel

When you select a Job in the Job Query Results List table, detailed information for the selected job is displayed in the Job Detail Information panel. The job detailed information is self-explanatory and includes the following:

- Running step
- Running time
- Current queue
- Status
- Type Run
- Initiator
- End Time
- User CPU usage
- System CPU usage

Note: Job detail information is retrieved along with the Job Query List; the content may not reflect the latest information. To view the latest job details, click the refresh button.

Batch Syslog Search/Display

Oracle TSAM Plus allows you to query the Batch JES system log.

The Batch Syslog window contains the following panels:

- [JES Syslogs Query by Filter Panel](#)
- [JES Syslogs Query Results Panel](#)

JES Syslogs Query by Filter Panel

[Table 1-49](#) lists the JES Syslogs Query by Filter options.

Table 1-49 JES Syslogs Query by Filter Options

Options	Description
JES Application	Specifies the JES system by which the JES syslog is shown
Job Name	Specifies a search string contained in the job name
Generated in Latest	Query syslog generated in the latest few seconds
Generated During	Query syslog generated during a specific time period

You can also query the exact syslog using the related job ID. In the field **Query by Job ID**, fill in the job ID and click **Query**. The corresponding syslog is listed in the results if searched.

JES Syslogs Query Results Panel

By default, only the latest 5000 lines of Batch JES syslog are shown. If there are more logs in your system, you can click the Prev chunk or Next chunk button to view the previous or next 5000 lines (the size of chunk) of the system logs.

In the current chunk, you can click the First/Prev/Next/Last page button to browse the system logs.

CICS Resources View/Edit

Oracle TSAM Plus allows you to query and edit CICS resource configuration files on the Oracle Tuxedo master node

Note: To enable this function, you must make sure the ARTADM server runs on each node and the `KIXCONFIG` environment variable is set to the CICS resource files directory.

Querying CICS Resources

You can query CICS resources by region. All of the CICS Resource configuration files are listed after you select a CICS region and click the **Query** button. You can also click **Refresh** button to refresh the CICS Resource list for the selected CICS region.

Editing/Viewing CICS Resources

You can edit or view the CICS resource depending on the authority configured in User Management.

To edit the CICS Resource configuration file, you must do the following steps:

1. Select one CICS region from region dropbox, then click the **Query** button. All of the CICS configuration files belonging to the selected region are listed.
2. Select a configuration file then click the **Edit** button.
Note: The **Edit** button is enabled when the number of selected configuration file is 1.
3. Click the **Update** button to save the modification. You can also click the **Back** button to return to the list of CICS Resources without modifying the file.

Note: If there are any format errors in the listed files, an error message is displayed.

For more information, see [Oracle Tuxedo Application Runtime for CICS and Batch documentation](#).

IMS Resources View/Edit

Oracle TSAM Plus allows you to query and edit IMS resource configuration files on the Oracle Tuxedo master node.

Querying IMS Resources

You can query IMS resources by IMS domain. All of the IMS Resource configuration files are listed after you select an IMS domain and click the **Query** button. You can also click **Refresh** to refresh the IMS Resource list for the selected Tuxedo domain.

Editing/Viewing IMS Resources

You can edit or view the IMS resource depending on the authority configured in User Management.

To edit the IMS Resource configuration file, you must do the following steps:

1. Select one IMS domain from IMS Domain drop down list, and then click the **Query** button. All of the IMS configuration files belonging to the selected domain are listed.

2. Select a configuration file then click the **Edit** button.

Note: The **Edit** button is enabled when the number of selected configuration file is 1.

3. Click the **Update** button to save the modification. You can also click the **Back** button to return to the list of IMS Resources without modifying the file.

Note: If there are any format errors in the listed files, an error message is displayed.

For more information, see [Oracle Tuxedo Application Runtime for IMS documentation](#).

User/Data Management

Click **Management** and select the following from the drop-down list:

- [User Management](#)
- [Data Management](#)
- [Global Parameters](#)
- [Blackout](#)
- [Domain Group Management](#)

User Management

The User List page allows you to add, edit and delete users and groups. An Oracle TSAM Plus group is a set of privileges. The default groups are "administrator" and "viewer". They cannot be deleted.

An Oracle TSAM Plus user may belong to one or more groups. When Oracle TSAM Plus is installed, the default user is "admin" (configured during installation), and cannot be deleted.

You can enter the **Group List** menu bar by clicking the **Group List** button, and return to the **User List** menu bar by clicking the **User List** button. [Table 1-50](#) lists the **User List Menu Bar** options.

Table 1-50 User List Menu Bar

Elements	Description
View	<p>From the drop-down menu, you can select the following:</p> <ul style="list-style-type: none">• Columns: Select the columns displayed in the table.• Detach: Displays the table in a separate window.• Reorder Columns: Change how the column order is displayed.
Change Properties	<p>Allows you to change user properties (if proper permission is set) after login to the system.</p> <p>You can change all user properties from the User Management window if you have User Management administrator privileges. Select the user you want to edit and click the Change Properties button. If you are the selected user, you can change your own properties.</p> <p>The current property settings are shown in the corresponding text fields. Type new values for the following properties:</p> <ul style="list-style-type: none">• Full Name• Description• Group
Change Password	<p>Allows you to change user password (if proper permission is set) after login to the system. Enter your old password, then enter your new password.</p> <p>The password should be limited to 6~16 characters, and can contain the following characters: alphanumeric characters, underscores.</p> <p>You can change all user passwords from the User Management window if you have User Management administrator privileges. Select the user whose password you want to change and click the Change Password button. If you are the selected user, you can change your own password.</p> <p>Click Change Password to change the password. Click Back to cancel the operation and return to the User Management window.</p> <p>Note: The TSAM Plus user password is valid for 180 days. You need to change your user password before it is expired. If the password has expired, you can find related information recorded in the <code>tsam.log</code> file.</p>

Table 1-50 User List Menu Bar

Elements	Description
Create User	<p>Allows you to add a new user. You must enter the following:</p> <p>Note: The Create User button is enabled only when the current logged-in user has User Management administrator privileges.</p> <ul style="list-style-type: none">• Name (*) The user login name. User name should be limited to 5~16 characters, and can contain the following characters: "[a-z], [A-Z], [0-9], _". The initial character must be [a-z] or [A-Z].• Full Name The user full name. User full name should be limited to 128 characters, and can contain the following characters: alphanumeric characters, underscore, space, or period.• Password(*) User password. You need to provide password and confirm password. The password should be limited to 6~16 characters, and can contain the following characters: alphanumeric characters, underscores.• Description The users description. The description is limited to 255 characters, and can contain the following characters: alphanumeric characters, underscore, space, or period.• Group(*) A drop-down list. A created user can belong to on or more groups.
Delete User	Deletes the selected user.
Online User	The information of all online users is displayed in Online Users window. The available information of users includes Name, IP Address, Login Time and Session ID.
Refresh	Updates the user information list.
Detach	Displays the table in a separate window.

[Table 1-51](#) lists the **Group List** button bar options.

Table 1-51 Group List Button Bar

Elements	Description
View	<p>From the drop-down menu, you can select the following:</p> <ul style="list-style-type: none">• Columns: Select the columns displayed in the table.• Detach: Displays the table in a separate window.• Reorder Columns: Change how the column order is displayed.
Modify	<p>Allows you to change group properties</p> <p>The "Modify" button is enabled when the current login user has the administrator privilege of group management. Select the group you want to edit and click the Modify button.</p> <p>The current property settings are shown in the corresponding text fields of the window. Type new values for the following properties:</p> <ul style="list-style-type: none">• Description• Privileges
Create Group	<p>Allows you to add a new group. You must enter the following:</p> <p>Note: The Create Group button is enabled when the current logged-in user has User Management administrator privileges. The privileges you need to set are displayed below the input bar and listed in</p> <ul style="list-style-type: none">• Name(*) The group name. Group name should be limited to 5~16 characters, and can contain the following characters: "[a-z], [A-Z], [0-9], _". The initial character must be [a-z] or [A-Z].• Id The group Id. It should be an integer and not less than 0. Note: If you use LDAP authentication, the Group ID must be the same as the one of the corresponding user in the LDAP server.• Description The group description. The description is limited to 255 characters. It can contain the following characters: alphanumeric characters, underscore, space, or period.• Privileges(*) A table. Allows you to set group privileges.

Table 1-51 Group List Button Bar

Elements	Description
Delete Group	Deletes the selected group. A group cannot be deleted if there are any users who belongs to it.
Refresh	Updates the group information list.
Detach	Displays the table in a separate window.

Oracle TSAM Plus Group/User Privileges

[Table 1-52](#) lists the Oracle TSAM Plus Group/User privilege options.

Table 1-52 Oracle TSAM Plus Group/User Privileges

Category	Privilege(s)
User Management	view, administrate
Group Management	view, administrate.
Data Management	administrate
Tuxedo Component Tree	view, administrate
ART region Tree	view, administrate
JES Component Tree	view, administrate
Global Parameter Setting	view, administrate
Tuxedo Policy Definition	view, administrate
ART Policy Definition	view, administrate
Tuxedo Alert Definition	view, administrate
ART alert Definition	view, administrate
Tuxedo Alert	view, administrate
ART Alert	view, administrate

Table 1-52 Oracle TSAM Plus Group/User Privileges

Category	Privilege(s)
Tuxedo Metrics	View Service Metrics, view call path Metrics, view transaction Metrics, view gateway Metrics, view bridge Metrics, view GWWS Metrics
ART Metrics	view Terminal Metrics, view CICS transaction Metrics
JES Metrics Management	View JES Metrics, administrate

Notes: Note the following:

- For lower left **Unread Alert** notification panel:
 - If you have Tuxedo Alert view privilege, but do not have ART alert view privilege, only the numbers of the Tuxedo alerts are shown.
 - If you do not have Tuxedo alert view privilege, but have ART alert view privilege, only the numbers of the ART alerts are shown.
 - If you do not have both Tuxedo and ART alert view privileges, the entire panel is hidden.
- On the alert query page, if you have both Tuxedo alert view and ART alert view privileges, both types of alerts can shown in the query result table. However, without corresponding administrate privilege, administrator tasks (such as **Clear** and **Delete**) are not available.
For example, you have Tuxedo alert, ART alert and Tuxedo alert administrate privileges. When you select all the queried out alerts (including both Tuxedo alert and ART alert type), the **Clear** and **Delete** button above the query result table is disabled.
- Group privilege is based on the User View privilege. The user cannot enter the group management console without User Management view privileges.

Data Management

Data Management allows you to purge the Oracle TSAM Plus database data (including the monitoring data in the database and the invalid Oracle Tuxedo components in the database).

Purging Monitoring Metrics Data

You can purge all monitoring data including Server/Service data and Call Path/ XA Transaction data. To purge Monitoring data, do the following steps:

1. Select the Types you want purged.

- All
- Call Path
- Bridge
- GWTDOMAIN
- Service
- XA Transaction
- User payload metrics
- GWWS
- Alert
- Tuxedo CICS Terminals
- JES
- IMS

Note:

- If you want to purge Tuxedo CICS Transaction data, "Service" type can be selected.
- Only user payload metrics stored in TSAM plus database can be purged here. User payload metrics stored in Hadoop or local file can only be purged by the user manually.
- Purging call path or service metrics does not delete the corresponding user payload metrics.

2. Set the **Old Than** date, before which the monitoring data will be purged.

3. Click the **Purge** button to delete monitoring data from the database.

The statistics service metrics data are purged automatically:

- Raw data are purged after 7 days.
- Hourly aggregated data are purged after 32 days.
- Daily aggregated data are purged after 12 months.

Purging Invalid Tuxedo Components Nodes

Tuxedo components can be changed to "INValid" status due to management operations at Tuxedo application side. By default, the "INValid" Tuxedo components are not deleted from the database. Customers can purge all "INValid" Tuxedo components in the Data management window.

Note: Currently, only the purging all "INValid" Tuxedo components on the data management page is supported. Invalid components can be purged in the component tree.

Global Parameters

The Global Parameters page contains following tabs:

- TSAM Plus Global Properties
- TSAM Plus Manager Properties

[Table 1-53](#) lists the TSAM Plus Global properties.

Table 1-53 TSAM Plus Global Properties

Properties	Description
TSAM Plus User password Life Time(days)	The period of time, in days, that a user password is in effect. Its value can be 0~2147483647. The default value is 180 days. If specified to 0, the password will never expire.
Maximum Selected Services in Service Metrics	The maximum selected services in Service Metrics.
Call pattern calculation schedule	The period of statistics job run for Call Pattern. The valid value is DAILY or HOURLY. DAILY is the default value. The modifications of this parameter take effect immediately. This parameter is applicable for Oracle Database only. If TSAM database is Derby, it is invisible.

Table 1-53 TSAM Plus Global Properties

Properties	Description
Call pattern calculation delay (minutes)	<p>The delay minutes after one hour or day for statistics job run for Call Pattern. If schedule is HOURLY, the valid value is 10~59, and 30 is the default value. If schedule is DAILY, the valid value is 10~1439, and 120 is the default value.</p> <p>The modifications of this parameter take effect immediately.</p> <p>This Parameter is applicable for Oracle Database only. If TSAM database is Derby, it is invisible.</p>
Automatic data purge	<p>Automatically purge data or not. FALSE is the default value.</p> <p>The modifications of this parameter is stored in Oracle database immediately. If Automatic data purge is TRUE, the expired monitoring data is purged at time by the value of Daily Maintenance Every Day.</p> <p>This Parameter is applicable for Oracle Database only. If TSAM database is Derby, it is invisible.</p>
Reserve monitoring data for days	<p>Days for data to be reserved before purging. If purge is FALSE, it is set to -1 and is not editable. If purge is TRUE, its valid value is 0~2880.</p> <p>The modifications of this parameter is stored in Oracle database immediately. If Automatic data purge is TRUE, the expired monitoring data is purged at time by the value of Daily Maintenance Every Day.</p> <p>This Parameter is applicable for Oracle Database only. If TSAM database is Derby, it is invisible.</p>
Daily Maintenance Every Day	<p>Change the schedule when Oracle Database daily maintenance job (TSAM_JOB_DAILY) executes. TSAM_JOB_DAILY purges expired monitoring data if Automatic data purge is TRUE.</p>
Statistics Collection Interval (minutes)	<p>Specify the interval to collect the statistics metrics data presented in dashboards. For more information, refer to Domains Dashboard.</p>
Payload Storage	<p>Select one of the user payload data storage: Database, Hadoop, or local file.</p>
Hadoop URL	<p>Specify the Hadoop URL where you want to storage the user payload data. The Hadoop URL must include the hostname, port, and payload file directory. For example: <code>hdfs://localhost:8080/tmp/tuxPayload/</code></p> <p>If the URL is invalid or unreachable, a warning message is displayed.</p>

[Table 1-54](#) lists the TSAM Plus Manager properties for each panel.

Table 1-54 Manager Parameter Properties

Properties	Description
Default Top N of Services	The number of most active services on Service Monitoring page. The default value is 2.
Tuxedo Application Runtime Feature Enabled Check Box	Indicates if oracle Tuxedo Application Runtime related components/pages are displayed on TSAM Plus console. Note: If you check or unchecked Tuxedo Application Runtime Feature Enabled , you must restart the Oracle TSAM Plus application for the change to take effect.
Synchronous Query Timeout (seconds)	Allows you to set the JES Job query timeout value (in seconds). The default value is 30 seconds.
Authentication Type	This value specifies the authentication method used by TSAM Plus manager. The optional values are: LDAPOnly and LocalFirst. LDAPOnly specifies that authentication is performed through LDAP. LocalFirst specifies that TSAM Plus manager tries to do authentication with TSAM Plus database first. The default value is LocalFirst.
Configuration Files Backup Directory	The directory for configuration files backup. The default path is \$USER_INSTALL_DIR\$. Note: To back up configuration files successfully, properly set the file permission of the backup directory, or start up TSAM Plus Manager web server using the same account as the backup directory file owner
Max Persistence Threads	Denotes how many persistence threads in a persistence thread pool. This number should not be bigger than the CPU number of the host running the Data server of TSAM Plus manager. The default value is 2.
Data Persistence Queue Size	The maximum of the metric entity list in queue. The default value is 5000.

Table 1-54 Manager Parameter Properties

Properties	Description
Data Persistence Queue Size	The maximum of the metric entity list in the persistence queue before traffic control. The default value is 5000.
Queue Full Sleep Interval (milliseconds)	Denotes the sleep interval (in milliseconds) in case the <code>MAX_INQUEUE_RUNNER_NUM</code> value is crossed. The default value is 500.
Queue Full Sleep Interval (milliseconds)	Denotes the LMS sleep interval (in milliseconds) in case the value of Data Persistence Queue Size is exceeded. The default value is 500. If the value of Data Persistence Queue Size is exceeded, TSAM Plus Manager persistence threads process the data metric as normal, and at same sends LMS a response with sleep interval (in milliseconds). LMS sleeps for the interval before sending another data metric.
Timeout Shutdown (seconds)	Denotes the timeout (in seconds) to give metric entity persistence threads a chance to finish their job after you shutdown the Oracle TSAM Plus manager. The default value is 30.
Enable <code>DYNAMIC_RAP</code>	Denote if <code>DYNAMIC_RAP</code> (configure in <code>BDMCONFIG</code>) feature Oracle Tuxedo JCA Adapter should be supported or not. If this parameter is enabled, the remote JCA adapter domain connected by <code>DYNAMIC_RAP</code> can be selected from domain gateway monitored page.
TSAM Plus Manager Log Level	Log level control for TSAM Plus Manager. The default value is Info. (Trace, Debug, Info, Warn, Error). Log level modifications take effect immediately; you do not need to reboot the application server.
JPA SQL Log Level	Log level control for JPA SQL. The default value is Warn. (Trace, Debug, Info, Warn, Error). Log level modifications take effect immediately; you do not need to reboot the application server.

Clicking on the **Modify Command** button saves the modifications into DB or properties file. If you need to reboot the TSAM Plus Manager, you can get the information in popup box.

Clicking on the **Refresh** command loads current value of each property from DB or properties file.

Blackout

With the Blackout management feature, users can easily control the metric data collection schedule. Once a Blackout starts, there is no metric data communication between TSAM manager and agent. Blackouts can be started or ended immediately, or according to a specific schedule.

List Blackout Definitions

Blackout List panel lists all the defined blackout definitions. Blackout policy is enabled on domain level. The blackout list contains the following information for each blackout definition:

Table 1-55 Blackout List Table Content

Column	Description
Select	The selection checkbox.
Blackout Name	Globally unique Blackout name.
Domain ID	On which domain the Blackout defines.
Status	Enabled/Disabled This status indicates whether this blackout is enabled.
Schedule Summary	Show the Blackout schedule summary. Policy content is organized in xml, each policy have following child nodes. <ul style="list-style-type: none">• starttime: start time in seconds.• duration: duration time in seconds. -1 for infinite• endtime: end time in seconds.-1 for no end time• repeattype: repeat type.• repeatvalue: repeat value. Default value is 0.• heartbeat: heart beat interval. -1 for stopping heartbeat

Note: Blackout Name and Policy Name must be different as they share the same naming space.

[Table 1-56](#) lists the actions you can take:

Table 1-56 Blackout List Table Controls

Control	Description
Add	Adds Blackout schedule on specific domain(s). The Blackout creation is done in the Edit Blackout page.
Delete	Removes defined Blackout on the selected domain(s). Blackout on this domain is removed if the connection between agent and manager is OK.
Enable	Enables the selected Blackout(s) immediately.
Disable	Disables the selected Blackout(s) immediately.
Edit	Edits the Blackout definition. The control is disabled when multiple Blackouts are selected.
Clone	Adds a new Blackout with the same schedule of the selected one. The control is disabled when multiple Blackouts are selected.
Refresh	Refreshes the table.

Add/Edit a Blackout Definition

On clicking of the **Create** or **Edit** button in the **Blackout List** table, the **Create/Edit Blackout** page is shown. [Table 1-57](#) lists the fields that should be filled to define a Blackout schedule.

Table 1-57 Create/Edit Blackout Fields

Field	Options	Description	Validation
Name		Globally unique Blackout name.	It cannot exceed 255 characters.
Domain		The corresponding Domain.	
Heartbeat Frequency		Agent send heart beat frequency. Agent do not send heart beat if this check box is not choosen.	

Table 1-57 Create/Edit Blackout Fields

Field	Options	Description	Validation
Start	Immediately	Blackout start time. It's converted to TSAM manager current time when user save this blackout.	
	Later	Specify the Blackout start date/time.	A valid date string
Duration	Infinite	No end Blackout.	
	Length	How many hours and minutes the Blackout last.	Hour should be a number and in range (0,1000000). Minutes should be a number and in range (0,59) Hours and Minutest should not be zero at the same time.
	Until	Until the designated time.	A valid date string
Repeating	Type	At what frequency the Blackout repeats. "By minutes", "By Hours", "By Days". "Weekly", "Monthly".	
	Frequency	Repeat period..	It should be a number and in range (0,1000000)
End Date	No End Date	Infinite repeating.	
	Specified End Date	Specify the repeating end date/time.	A valid date string

Note: TSAM Plus Manager should be synchronized with agent. All time inputted in TSAM Plus console is taken as the time zone of TSAM Plus Manager.

Add and **Add & Enable** buttons are shown when you create a new blackout.

Save and **Save & Enable** buttons are shown when you edit an existing blackout.

- Add

Save the Blackout schedule configuration, start the Blackout immediately if "Immediately" is configured, or schedule it for later run.

- Add & Enable

Save and enable blackout, then back to the Blackout List Page.

- Save

Save the modified Blackout schedule configuration, start the Blackout immediately if "Immediately" is configured, or schedule it for later run.

- Save & Enable

Save and enable the modified Blackout schedule configuration.

- Cancel

Cancel the Blackout creation/edition and back to the **Blackout List** page.

Domain Group Management

The Domain Group Management page allows you to create and manage domain groups. A domain group can be monitored in the domains dashboard.

Viewing Domain Group List

The Domain Group List shows all the domain groups created. [Table 1-58](#) lists the information you can view for each group.

Table 1-58 Domain Group List

Items	Description
Name	Domain group name
Domain Count	The number of domains included in the group
Domains	The name of domains belonging to the group
Key Services	Key services defined in the group. The key services is a set of services you want to monitor regularly from the domains dashboard.

Creating a New Domain Group

To create a new domain group, do the following:

1. From the Domain Group List, click **Add**.
2. In the window that appears, specify the following and click **OK**.
 - Domain Group Name: Define a group name
 - Domain Selection: Select the domains you want to include in the group
 - Key Services: Optional. Select up to seven services that you want to monitor regularly from the group dashboard. Only the services that are available across all domains can be selected.

Editing a Domain Group

To edit a domain group, do the following:

1. From the Domain Group List, highlight the group you want to edit.
2. Click **Edit**.
3. Specify the domains and key services, and then click **OK**.

Load Generator

You can create the replay definition from call paths by Clicking Load Generator ->Replay Definition. Based on the replay definition, you can execute the replay for standard and stress test modes. For more information, see [Using Test Load Generator](#).

Alert Monitoring

Click **Alert** and select one of the following:

- [Tuxedo Alert Definition](#)
- [CICS Alert Definition](#)
- [IMS Alert Definition](#)
- [Batch Alert Definition](#)
- [Alert Query](#)

- [Alert Metrics Tables](#)

Tuxedo Alert Definition

Click **Alert** and select **Tuxedo Alert Definition** from the drop-down list; the Tuxedo Alert Definition List Panel appears. This panel lists available alert definitions and allows you to create, edit, and manage alert definitions.

[Table 1-59](#) and [Table 1-60](#) list the **Tuxedo Alert Definition Menu Bar** and **Tuxedo Alert Column** options respectively.

Table 1-59 Tuxedo Alert Definition Menu Bar Options

Options	Description
View	From the drop-down menu, you can select the following: <ul style="list-style-type: none"> • Columns: Select the columns displayed in the table. • Detach: Displays the table in a separate window. • Reorder Columns: Change how the column order is displayed.
Add	Creates a new alert definition.
Delete	Deletes the alert definition.
Enable	Enables the alert definition.
Disable	Disables the alert definition.
Edit	Modifies the alert definition.
Clone	Creates a new alert with same definition as current alert but the name is left empty.
Alerts	See all alerts generated by this alert definition.
Refresh	Refresh the alert definition list.
Detach	Displays the table in a separate window.

Table 1-60 Tuxedo Alert Columns

Column	Description
Selection checkbooks	Select a set of alert definitions in current page for processing.
Name	The alert name specified in the alert definition.
Status	Displays alert status. The status could be enabled or disabled. Enable: The alert definition enabling request has been sent to Oracle Tuxedo. Disable: The alert definition disabling request has been sent to Oracle Tuxedo.
Type	The monitoring category and system are consistent with policy definition. The category types are as follows: <ul style="list-style-type: none">• Call Path• Service• XA Transaction• BRIDGE• GWTDOMAIN• GWWS• Service Statistics: a “default_adaptive_alert” is enabled by default.
Evaluation Condition	The FML Boolean expression of the alert triggering conditions
Tuxedo Components	The resource FML Boolean expression (similar to Policy definition).

Define New Alert Page

Click the **Add**, **Edit** or **Clone** button on the **Alert Definition List** Menu bar; the **Define New Alert** page appears. It allows you to define a concrete Oracle Tuxedo alert definition.

The Define New Alert page is divided into four sections:

- [Alert Properties](#)
- [Tuxedo Components](#)
- [Metrics Evaluation Expression\(s\)](#)
- [FML Boolean Expression\(s\)](#)

Alert Properties

Table 1-61 lists the **Alert Properties** options.

Table 1-61 Alert Properties Options

Options	Descriptions
Name	The name of the alert. It must be unique among the alerts globally. The name value is limited to 255 characters and must be unique within the project. The initial character must not be a ". ".
Type	<p>The type of the alert; it is based on monitoring policy. The drop-down list includes the following items:</p> <ul style="list-style-type: none">• Call Path• Service When you select a service alert, the Metrics Independent Checking check box appears. It allows Oracle TSAM Plus to check service timeouts independent of metrics collection. The alert can be generated while the service is running. If checked, "Drop Message" is removed from the "Action" option drop-down menu.• Service Statistics You can define one or more service statistics alerts for a Tuxedo domain or a domain group. When you select a service statistics alert, the configuration panel becomes different than other alert types. You can configure TSAM Plus to compare the instant metric data with a static threshold or historical data to evaluate and trigger the service statistics alerts. For more information, see Adaptive Settings (For Service Statistics Alerts Only).• XA Transaction• GWTDOMAIN• BRIDGE• GWWS
Severity	<p>The severity drop-down list contains the following severity levels:</p> <ul style="list-style-type: none">• Fatal• Critical• Warn• Information

Table 1-61 Alert Properties Options

Options	Descriptions
Interval	Accepts an integer value in seconds. The current alert is evaluated only once during this interval. The purpose is to throttle alert volume in similar scenario. If the interval value is 0, any alert is reported.
Action	<p>Specifies what kind of action is executed when an alert definition is set to "true". The following actions are supported:</p> <ul style="list-style-type: none">• Publish to Tuxedo Event Broker <p>An event is posted to the Oracle Tuxedo event broker. The event name is the alert name by default; you can also specify it manually. The buffer is an FML32 buffer containing the metric snapshot.</p> <ul style="list-style-type: none">• Drop Message <p>If this alert evaluation is true, the request message is dropped and the call fails (TPESYSTEM). This only applies to Call Path and Service type alerts. The evaluation points come before the request sent to the IPC queue (call path) and after it is retrieved from the IPC queue (call path, service). This alert can be used to avoid unneeded service processing for stagnant requests. For example, if the request message "Wait Time" in the IPC queue exceeds a particular threshold, the client may have already timed out. In this case, the action can drop the stagnant request and the application server will not process it.</p> <p>Caution: If the drop message action is executed in a embedded Tuxedo call, the whole call fails.</p>

Tuxedo Components

The Component Scope panel contains the following selections:

- Domain Group: For service statistics alerts only.
- Domain: Optional
- Machine: Optional
- Group: Optional
- Server: Optional

Note: Machine, Group, and Server can also accept FML32 boolean expressions.

Click the radio button to select input method (use the drop-down list or manually enter an FML32 boolean expression). The values of each level is determined by its parent level, except for Domain.

- Services: For service statistics alerts only.

Metrics Evaluation Expression(s)

You can select the metric, operator and threshold to compose one expression. The logic for the expression can be "and" or "or".

FML Boolean Expression(s)

You can also input FML Boolean expressions directly. For more information, see [Alert Metrics Tables](#). For call path alerts, if the elapsed time *only* is evaluated in the evaluation expression, the alert is triggered once along the entire call path. Other metrics combined with elapse time do not have this effect.

Note: The TSAM Plus Console cannot guarantee FML Boolean expression syntax correctness.

Adaptive Settings (For Service Statistics Alerts Only)

For service statistics alerts, when you select Adaptive Threshold in the Logic for Expression panel, you need to continue to configure the Adaptive Settings. It specifies a past time of which the metrics data are used for comparison with current time collected data. If the comparison result meets the criteria you set in the metrics evaluation expression, an alert is reported.

- **Daily:** the average or aggregated metrics data of one day is compared for evaluation. You can select one day of the past 30 days.
- **Hourly:** the average or aggregated metrics data in one hour of a day is compared for evaluation. You can specify how many hours ahead (-) or after (+) current hour in one day of the past 30 days.

For example, if you define an adaptive alert evaluation expression for the response time metrics and specify **Adaptive Settings** as “Hourly, 1 day ago, -5 hours”, the average response time between 1:00 pm to 2:00 pm yesterday will be used for comparison with each response time data collected during 6:00-7:00 pm today.

The **Per Service Instance** is only available for the time-related metrics, of which average data is calculated. When this option is checked, the metrics data collected from each service instance is evaluated against the historical average data.

CICS Alert Definition

Click **Alert** and select **CICS Alert Definition** from the drop-down list; the **CICS Alert Definition List** panel appears. This panel lists available CICS alert definitions and allows you to create, edit, and manage alert definitions.

CICS Alert Definition List Panel

[Table 1-62](#) and [Table 1-63](#) list the **CICS Alert Definition List** menu bar options and columns respectively.

Table 1-62 CICS Alert Definition List Menu Bar

Elements	Description
View	From the drop-down menu, you can select the following: <ul style="list-style-type: none">• Columns: Select the columns to shown in the table• Detach: Displays the table in a separate window• Reorder Columns: Change the order how the columns are displayed
Add	Creates a new alert definition.
Delete	Deletes the alert definition.
Enable	Enables the alert definition.
Disable	Disables the alert definition.
Edit	Modifies the alert definition.
Clone	Creates a new alert with same definition as current alert but the name is left empty.
Alerts	Identifies all alerts generated by this alert definition.
Refresh	Updates the alert definition list.
Detach	Displays the table in a separate window.

Table 1-63 Alert Definition List Columns

Column	Description
Selection checkbooks	Select a set of alert definitions in current page for processing.
Name	The alert name which is specified in alert definition
Status	Displays alert status. The status could be enabled or disabled. Enable: The alert definition enabling request has been sent to Tuxedo. Disable: The alert definition disabling request has been sent to Tuxedo.
Type	The monitoring category and system, consistent with policy definition. The category types are as follows: <ul style="list-style-type: none">• CICS Transaction• CICS Terminal
Evaluation Condition	The FML Boolean expression of the alert triggering conditions.

Define New Alert Panel

Click **Add** or **Clone** on the menu bar; the **Define New Alert** panel appears. This panel allows you to define concrete Tuxedo Application Runtime alert definitions.

The Define New Alert panel is divided into three sections:

- [Alert Properties](#)
- [Metrics Evaluation Expression\(s\)](#)
- [FML Boolean Expression\(s\)](#)

Alert Properties

[Table 1-64](#) lists the **Alert Properties** options.

Table 1-64 Alert Properties Options

Column	Description
Name	The name of the alert. It must be unique among the alerts globally. The name value is limited to 255 characters and must be unique within the project. The initial character must not be a " . " .
CICS Region	Defines the function scope of the defined alert.
Type	The type of the alert; it is based on monitoring policy. The drop-down list includes the following items: <ul style="list-style-type: none">• CICS Transaction• CICS Terminals
Severity	The severity drop-down list contains the following: <ul style="list-style-type: none">• Fatal• Critical• Warn• Information
Interval	It accepts an integer value (in seconds). The current alert is evaluated only once during this interval. The purpose is to throttle alert volume in a similar scenario.
Action	Specifies what kind of action is executed when an alert definition set to "true". It supports the following action: <ul style="list-style-type: none">• Publish to Tuxedo Event Broker An event is posted to the Oracle Tuxedo event broker. The event name is the alert name by default; you can also specify it manually. The buffer is an FML32 buffer containing the metric snapshot.

Metrics Evaluation Expression(s)

You can select the metric, operator and threshold to compose one expression. The logic for the expression can be "and" or "or".

FML Boolean Expression(s)

You can also input FML Boolean expressions directly. For more information, see [Alert Metrics Tables](#).

Note: The TSAM Plus Console cannot guarantee FML Boolean expression syntax correctness.

IMS Alert Definition

Click **Alert** and select **IMS Alert Definition** from the drop-down list; the **Alert Definition List** panel appears. This panel lists available Tuxedo Application Runtime IMS alert definitions and allows you to create, edit, and manage alert definitions.

IMS Alert Definition List Panel

[Table 1-65](#) and [Table 1-66](#) list the **IMS Alert Definition List** menu bar options and columns respectively.

Table 1-65 IMS Alert Definition List Menu Bar

Elements	Description
View	From the drop-down menu, you can select the following: <ul style="list-style-type: none">• Columns: Select the columns to shown in the table• Detach: Displays the table in a separate window• Reorder Columns: Change the order how the columns are displayed
Add	Creates a new alert definition.
Delete	Deletes the alert definition.
Enable	Enables the alert definition.
Disable	Disables the alert definition.
Edit	Modifies the alert definition.
Clone	Creates a new alert with same definition as current alert but the name is left empty.
Alerts	Identifies all alerts generated by this alert definition.
Refresh	Updates the alert definition list.
Detach	Displays the table in a separate window.

Table 1-66 Alert Definition List Columns

Column	Description
Selection checkbooks	Select a set of alert definitions in current page for processing.
Name	The alert name which is specified in alert definition
Status	Displays alert status. The status could be enabled or disabled. Enable: The alert definition enabling request has been sent to Tuxedo. Disable: The alert definition disabling request has been sent to Tuxedo.
Type	The monitoring category and system, consistent with policy definition. The category types are as follows: <ul style="list-style-type: none">• IMS Transaction• IMS Program
Evaluation Condition	The FML Boolean expression of the alert triggering conditions.

Define New Alert Panel

Click **Add** or **Clone** on the menu bar; the **Define New Alert** panel appears. This panel allows you to define concrete Tuxedo Application Runtime IMS alert definitions.

The Define New Alert panel is divided into three sections:

- [Alert Properties](#)
- [Metrics Evaluation Expression\(s\)](#)
- [FML Boolean Expression\(s\)](#)

Alert Properties

[Table 1-67](#) lists the **Alert Properties** options.

Table 1-67 Alert Properties Options

Option	Description
Name	The name of the alert. It must be unique among the alerts globally. The name value is limited to 255 characters and must be unique within the project. The initial character must not be a " . " .
Domain	Defines the function scope of the defined alert.
Type	The type of the alert; it is based on monitoring policy. The drop-down list includes the following items: <ul style="list-style-type: none">• IMS Transaction• IMS Program
Severity	The severity drop-down list contains the following: <ul style="list-style-type: none">• Fatal• Critical• Warn• Information
Interval	It accepts an integer value (in seconds). The current alert is evaluated only once during this interval. The purpose is to throttle alert volume in a similar scenario.
Action	Specifies what kind of action is executed when an alert definition set to "true". It supports the following action: <ul style="list-style-type: none">• Publish to Tuxedo Event Broker An event is posted to the Oracle Tuxedo event broker. The event name is the alert name by default; you can also specify it manually. The buffer is an FML32 buffer containing the metric snapshot.

Metrics Evaluation Expression(s)

You can select the metric, operator and threshold to compose one expression. The logic for the expression can be "And" or "Or".

FML Boolean Expression(s)

You can also input FML Boolean expressions directly. For more information, see [Alert Metrics Tables](#).

Batch Alert Definition

You can view, add, edit and delete JES Alert Definitions through Alert Definition

Click **Alert** and select **Batch Alert Definition** from the drop-down list; the **Batch Alert Definition List** panel appears. This panel lists available Batch alert definitions and allows you to create, edit, and manage alert definitions.

Batch Alert Definition List Panel

[Table 1-68](#) and [Table 1-69](#) list the **Batch Alert Definition List** menu bar options and columns respectively.

Table 1-68 Batch Alert Definition List Menu Bar

Elements	Description
View	From the drop-down menu, you can select the following: <ul style="list-style-type: none">• Columns: Select the columns to shown in the table• Detach: Displays the table in a separate window• Reorder Columns: Change the order how the columns are displayed
Add	Creates a new alert definition.
Delete	Deletes the alert definition.
Enable	Enables the alert definition.
Disable	Disables the alert definition.
Edit	Modifies the alert definition.
Clone	Creates a new alert with same definition as current alert but the name is left empty.
Alerts	Identifies all alerts generated by this alert definition.
Refresh	Updates the alert definition list.
Detach	Displays the table in a separate window.

Click **Add** or **Edit**; the **JES Alert Definition panel** appears.

Table 1-69 Alert Properties Options

Elements	Description
Name	The name of the alert. It must be unique among the alerts globally. The name value is limited to 255 characters and must be unique within the project. The initial character must not be a " . " .
Type	The type of the alert; it is based on monitoring policy. The drop-down list includes the following items: <ul style="list-style-type: none"> • JES Jobs • JES Metrics
Severity	The severity drop-down list contains the following: <ul style="list-style-type: none"> • Fatal • Critical • Warn • Information
Interval	It accepts an integer value (in seconds). The current alert is evaluated only once during this interval. The purpose is to throttle alert volume in a similar scenario.

Batch Alert Definition Filter Conditions

[Table 1-70](#) lists the **Batch Alert Definition Filter Conditions** options.

Table 1-70 Batch Alert By Filter Options

Options	Description
JES Application	Identifies which JES application the Job is executed, single choice dropdown list, required. Note: Job information can be only be obtained when the selected JES application is alive. If an inactive JES application is selected when clicking the 'Query' button, a warning message window is displayed.
Job Name	The string contained in the job name, input text box.
Job Owner	One or more owners of the job, input text box.

Table 1-70 Batch Alert By Filter Options

Options	Description
Job Priority	One or more priorities.
Job Class	The job class, multiple choice dropdown list.

Batch Alert Definition Details

If the **Type** is Batch Job, the Batch metric detail panel provides the information shown in [Table 1-71](#). Three levels of metric details can be selected and displayed: Job Status, Job Execution Time, and Job.

Table 1-71 JES Job Type Detailed Information

Elements	Description
Job Status	Job status is changed to the specified status. You can specify one of the following: <ul style="list-style-type: none"> ● EXECUTING ● CONVING ● WAITING ● DONE ● FAIL ● HOLD_WAITING ● HOLD_CONVING ● INDOUBT ● DISCARD
Job	Job is purged
Job Execution Time	Job execution time >= ? millisecond during the last Interval seconds

If the **Job Type** is JES Metrics, the JES metric detail panel provides the information shown in [Table 1-72](#)

Table 1-72 JES Metrics Type Detailed Metrics Information

Elements	Description
Jobs average execution time	Job average execution time >= ? millisecond during the last Interval seconds
Jobs fails number	Jobs fails number >= during the last Interval seconds
Jobs waiting number	Jobs waiting number >= during the last Interval seconds

Alert Query

Alert Query view displays the alerts that have transpired. It includes metric alerts and system alerts. Metrics alert are the events generated against the defined metrics alert. System alerts represent system events.

Click **Alerts** and select **Alert Query** from the drop-down menu; the **Alert Query** page appears. The Alert Query page contains two tabs:

- [Unread Alerts Tab](#): Displays all unread alerts.
- [Historical Alerts Tab](#) Provides historical alert query information.

Unread Alerts Tab

"Unread" alerts are most important for administrators because they represent events that they may not be aware of. The Unread Alerts tab contains two sections:

- Filtering Condition
- Unread Alert Columns

[Table 1-73](#) lists the **Alert Query** filtering condition options.

Table 1-73 Alert Query Filtering Conditions

Options	Description
Contains	Select a set of alerts in current page for processing.
Type	<p>The monitoring category and system are consistent with policy definition. The following category types are as follows:</p> <ul style="list-style-type: none">• Any• Call Path• Service• Service Statistics• XA Transaction• BRIDGE• GWTDOMAIN• GWWS• CICS Transaction• CICS Terminals• JES Job• JES Metrics• IMS Transaction• IMS Program• System <p>Note: System represents this is a system-level event generated by Oracle TSAM Plus.</p>
Severity	<p>Four severity levels are supported:</p> <ul style="list-style-type: none">• Any• Fatal• Critical• Warn• Information

[Table 1-74](#) lists the Alert Query columns.

Table 1-74 Alert Query Columns

Column	Description
Selection checkbox	Select a set of alerts in current page for processing.
Name	The alert name which is specified in alert definition.
Type	<p>The monitoring category and system are consistent with policy definition. The following category types are as follows:</p> <ul style="list-style-type: none">• Any• Call Path• Service• Service Statistics• XA Transaction• BRIDGE• GWTDOMAIN• GWWS• CICS Transaction (Tuxedo Application Runtime only)• CICS Terminals (Tuxedo Application Runtime only)• IMS Transaction• IMS Program• JES Job (Tuxedo Application Runtime only)• JES Metrics (Tuxedo Application Runtime only)• System <p>Note: System represents this is a system-level event generated by Oracle TSAM Plus.</p>
Severity	<p>Four severity levels are supported:</p> <ul style="list-style-type: none">• Any• Fatal• Critical• Warn• Information
Cleared	Indicates whether the alert has been marked as read.
Clear Time	Identifies the time an alert is “cleared.”

Table 1-74 Alert Query Columns

Column	Description
Log Time	The time an Alert occurs.
Description	The reason/cause for the alert. It uses the following format: <code>"Process [%s:%d] evaluates alert [%s] with true, metrics[key=value,key=value...]"</code>

Click **Clear** to clear all the selected alerts on current page.

Click **Delete** to delete all the selected events on current page.

Click **Detail**, a popup window with the following detailed Alert information appears:

- All information in the alerts summary data grid.
- The metrics detail and reason string for this alerts.

Once an alert is cleared as "Unread" status and not deleted, it still can be queried in the historical alert page, but not in the active alert page. The following filter conditions can be set:

- Alert Type
- Alert Severity
- A custom "search" text string

The search operation finds matched records from all current filtered "unread" alert results, no limited to the "unread" alerts shown on the current page. Matched record means any alert that is matching the specified alert category condition, alert severity condition and having the given "search" text string within any of the following value fields: "Alert Name", "Alert Reason String (Description)".

Click the **Reset** button to reset filter conditions and list all "unread" alerts.

Historical Alerts Tab

The Historical Alerts Tab allows you to query historical alerts with supplied filtering conditions. It contains two panels:

- Historical Alert Query Conditions
- Alerts Query Results

[Table 1-75](#) lists the **Historical Alert Query Conditions** panel options.

Table 1-75 Historical Alerts Query Conditions Options

Options	Description
Alert Name	The Alerts names configured in this project. "Any" is a reserved key word meaning all alerts.
Type	"Any" and the supported monitoring categories. System is in same level with other monitoring categories.
Severity	"Any" and supported severity levels. If Include Lower Level is checked, all alerts at this level or below are queried. The sequence is Fatal, Critical, Warn and Info.
Query During	The exact time of the previous day to the exact <i>same</i> time of the current day. For example: 7:45 Wednesday-to-7:45 Thursday.

Click **Submit**; the results are displayed in the **Alert Query Result** panel.

System Alerts Supported

Users cannot define System Alerts; they are generated automatically (for example, an invalid alert definition is encountered, or the database purging process is completed).

[Table 1-76](#) lists the supported system alerts.

Table 1-76 System Alerts Supported

Alert name	Alert Severity	Alert Reason String
.INVALIDEXP	warning	Alert definition [%ALERT_NAME] has invalid evaluation expression
DB.PURGE	information	Data purging (sequence:?) filed at [%DATE_TIME] by admin is done.

Alert Metrics Tables

The Alert Metrics tables contain the alert metrics for all alert types. Each metric has the corresponding FML32 field name and applicable operators. Referencing these tables will help you write the alert metric FML boolean expression directly when defining a new alert.

- [Call Path Alert Metrics Table](#)
- [Service Alert Metrics Table](#)
- [GWTDOMAIN/BRIDGE Alert Metrics Table](#)
- [GWWS Alert Metrics Table](#)
- [XA Transaction Alert Metrics Table](#)
- [CICS Transaction Alert Metrics Table](#)
- [CICS Terminals Alert Metrics Table](#)
- [IMS Transaction Alert Metrics Table](#)
- [IMS Program Alert Metrics Table](#)

Call Path Alert Metrics Table

Table 1-77 Call Path Alert Metrics Table

Metric	FML32 Field	Operators
Elapse Time	TA_MONELAPSETIME	>, >=
Execution Status	TA_MONERRNO	==, !=
URcode	TA_MONURCODE	==, !=
Depth	TA_MONDEPTH	>, ==, <
Message Number on Request Queue	TA_MONMSGQUEUED	>, >=

Service Alert Metrics Table

Table 1-78 Service Alert Metrics Table

Metric	FML32 Field	Operators
Execution Time	TA_MONEXEETIME	>, >=
Execution Status	TA_MONERRNO	==, !=
Request Message Size	TA_MONMSGSIZE	>, >=
Message Number on Request Queue	TA_MONMSGQUEUED	>, >=
URcode	TA_MONURCODE	==, !=
Waiting Time	TA_MONMSGWAITTIME	>, >=
CPU Time	TA_MONCPUPTIME	>, >=

GWTDOMAIN/BRIDGE Alert Metrics Table

Table 1-79 GWTDOMAIN/BRIDGE Alert Metrics Table

Metric	FML32 Field	Operators
Link Status (GWTDOMAIN/BRIDGE)	TA_MONLINKSTATUS	Lost, Reconnect
Outstanding Request (GWTDOMAIN)	TA_MONNUMWAITRPLY	>, >=

GWWS Alert Metrics Table

Table 1-80 GWWS Alert Metrics Table

Metric	FML32 Field	Operators
Inbound one way failed number	TA_MONINOWFAIL	>, >=
Outbound one way failed number	TA_MONOUTOWFAIL	>, >=

Table 1-80 GWWS Alert Metrics Table

Metric	FML32 Field	Operators
Inbound RPC failed number	TA_MONINRPCFAIL	>, >=
Outbound RPC failed number	TA_MONOUTRPCFAIL	>, >=
Inbound average processing time	TA_MONINTIME	>, >=
Outbound average processing time	TA_MONOUTTIME	>, >=
Inbound Pending Request	TA_MONINBOUNDPEND	>=
Outbound Pending Request	TA_MONOUTBOUNDPEND	>=

XA Transaction Alert Metrics Table

Table 1-81 XA Transaction Alert Metrics Table

Metric	FML32 Boolean Expression
Transaction Failed	TA_MONXANAME=='tpabort' TA_MONXANAME=='xa_rollback'
Heuristically Completed	TA_MONXANAME=='xa_commit' && TA_MONXACODE==8
Heuristically Commit	TA_MONXANAME=='xa_commit' && TA_MONXACODE==7
Resource Manager Failure	TA_MONXANAME%'xa_.*' && TA_MONXACODE==--3

CICS Transaction Alert Metrics Table

Table 1-82 CICS Transaction Alert Metrics Table

Metric	FML32 Field	Operators
Execution Time	TA_MONEXEETIME	>, >=
Message Number on Request Queue	TA_MONMSGQUEUED	>, >=

Table 1-82 CICS Transaction Alert Metrics Table

Metric	FML32 Field	Operators
Waiting Time	TA_MONMSGWAITTIME	>, >=
CPU Time	TA_MONCPUPTIME	>, >=

CICS Terminals Alert Metrics Table

Table 1-83 CICS Terminals Alert Metrics Table

Metric	FML32 Field	Operators
Average Session Number	TA_MONSESSIONNUM	>, >=
Number of CICS Transactions Failed	TA_MONTRANFAIL	>, >=

IMS Transaction Alert Metrics Table

Table 1-84 IMS Transaction Alert Metrics Table

Metric	FML32 Field	Operators
Execution Time	TA_IMS_EXECUSEC	>, >=
CPU Time	TA_IMS_CPUTIMEUSEC	>, >=
DLI Time	TA_IMS_DLIUSEC	>, >=

IMS Program Alert Metrics Table

Table 1-85 IMS Program Alert Metrics Table

Metric	FML32 Field	Operators
Execution Time	TA_IMS_EXECUSEC	>, >=

Table 1-85 IMS Program Alert Metrics Table

Metric	FML32 Field	Operators
CPU Time	TA_IMS_CPUTIMEUSEC	>, >=
DLI Time	TA_IMS_DLIUSEC	>, >=

Using the Enterprise Manager for Oracle Tuxedo

This chapter describes how you can use the Enterprise Manager for Oracle Tuxedo (EM) to monitor Tuxedo targets, integrate with EM functionality, and automatically deploy your Tuxedo applications on different remote machines using resource broker.

This chapter contains the following topics:

- [Using Web-Based Console](#)
- [Viewing Configuration Topology](#)
- [GDG Management](#)
- [Tuxedo Event Monitoring](#)
- [Reporting for Tuxedo Targets](#)
- [Using Enterprise Manager SLA](#)
- [Using Resource Broker](#)
- [Dynamic Request Broker](#)
- [Integration with OVAB](#)
- [Integration with Exalogic Elastic Cloud](#)

Using Web-Based Console

This section contains the following topics:

- [Logging into Enterprise Manager Cloud Control](#)
- [Viewing Tuxedo Summary](#)
- [Target Home Page Introduction](#)
- [Monitoring Tuxedo Targets](#)
- [Viewing Configuration Topology](#)

Logging into Enterprise Manager Cloud Control

The Enterprise Manager Cloud Control home page is displayed upon logging in. For information about the elements on the home page, click **Help** in the upper right of the page.

Viewing Tuxedo Summary

The Tuxedo Summary page filters all Tuxedo domain targets from a large number of Enterprise Manager monitoring targets, and allows you to centrally manage Tuxedo targets and easily navigate to each target home page.

To enter the Tuxedo Summary page, do the following:

1. Log in to Enterprise Manager Cloud Control as SYSMAN.
2. From the home page, click **Targets > Middleware** at the top.
3. In the Middleware page, click **Middleware Features > Tuxedo Summary**.

The **Tuxedo Summary** page is displayed, as shown in [Figure 1-1](#).

Figure 1-1 Tuxedo Summary Page

Name	Type	Status	Member Status Summary	Version	Compliance Score (%)
bej301142.on.oracle.com/bej301142/oracle/tuxedo12.1.1.0	Tuxedo Home	n/a	0 0 0 0	12.1.1.0	n/a
bej301142.on.oracle.com/16996/listen	Tuxedo Listener	🟢	0 0 0 0		n/a
bej301185.on.oracle.com/bej301185/oracle/tuxedo12.1.1.0	Tuxedo Home	n/a	0 0 0 0	12.1.1.0	n/a
bej301185.on.oracle.com/bej301185/oracle/tuxedo12.1.1.0	Tuxedo Application Home	n/a	0 0 0 0		n/a
bej301185.on.oracle.com/16996/listen	Tuxedo Listener	🟢	0 0 0 0		n/a
phv-bej301142-65931	Tuxedo Domain	🟢	3 14 0 0		n/a
TUXDOMAIN-bej301185-33336	Tuxedo Domain	🟢	0 0 0 4		n/a
bej301185	Tuxedo Machine	🟢	0 0 0 3		n/a
BRIDGE	Tuxedo Bridge	🟢	0 0 0 0		n/a
SYSGRP_SITE1	Tuxedo Group	🟢	0 0 0 1		n/a
LMS	Tuxedo LMS Server	🟢	0 0 0 0		n/a

Actions Bar

Table 1-1 lists the action bar functions.

Table 1-1 Actions Bar Description

Action	Description
Add	Adds Tuxedo domains to monitoring targets. You can choose to discover existing domains or create a new Tuxedo domain in Resource Broker Domain UBBCONFIG Editor .
Edit	Edits the highlighted Tuxedo domain in the Domain UBBCONFIG Editor .
Remove	Removes the Tuxedo target and all metrics associated with it from the Enterprise Repository. For Tuxedo domains, this action deletes all Tuxedo targets belonging to it. A Tuxedo <code>tlisten</code> target can be removed only if no Tuxedo Machine targets have "Managed_by" associated with them, otherwise a warning message pops up to prevent you from proceeding.

Table 1-1 Actions Bar Description

Action	Description
Configure	Configures the properties of the highlighted target.
Manage Application Package	Manages application packages that are specially used for resource broker feature.

Search Panel

You can quickly search for a target by specific fields using the Search panel.

Select the field to be searched in the drop-down list and enter an appropriate search string in the text box to run a standard search. You can also use Advanced Search to run the search with more extensive filtering.

Target Summary

[Table 1-2](#) lists descriptions to each summary column fields.

Table 1-2 Target Summary Description

Item	Description
Name	The target name.
Type	The target type.
Status	Shows the target status.
Member Status Summary	Shows the statistics of underlying member status.
Version	The target version.

Target Home Page Introduction

When you select a Tuxedo target from the Enterprise Manager Cloud Control home page or Tuxedo Summary page, the Home page for the target is displayed. For example, when you click on a Tuxedo Domain target, the screen shown in [Figure 1-2](#) appears.

Figure 1-2 Target Home Page

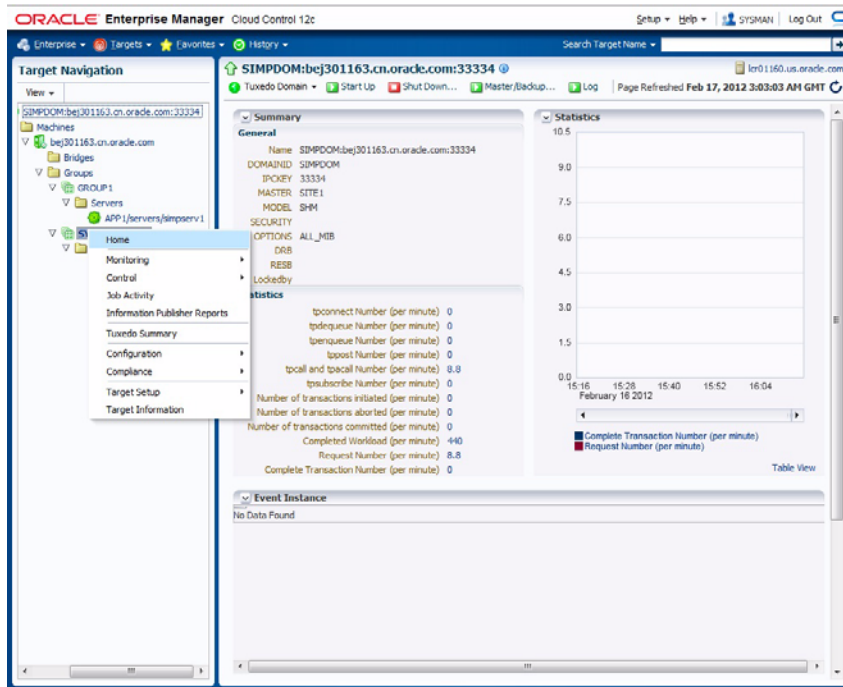


Figure 1-2 shows the target navigation pane on the left and the content page on the right. From the target navigation pane, you can expand or collapse the tree. When you select a target, the target home page is displayed in the content pane and the target menu is displayed at the top of the page. You can also view the menu for a target by right-clicking the target in the navigation pane.

A typical target home page consists of the following items:

- [Target Navigation Tree](#)
- [Target Menu](#)
- [Content Pane](#)
- [Right-Click Target Menu](#)
- [UBB Modification](#)
- [Credentials Page](#)

Target Navigation Tree

Lists all Tuxedo targets monitored by the OMS instance in a navigation tree. All Tuxedo targets belonging to a Tuxedo Domain are displayed in a tree hierarchy as shown in [Listing 1-1](#).

Listing 1-1 Tuxedo Domain Tree Hierarchy

```
[Tuxedo Domain Name]
  Tuxedo Machines (Category)
    [Tuxedo Machine 1 name]
      Tuxedo Groups (Category)
        [Tuxedo Group 1 name]
          Tuxedo Server (Category)
            [Tuxedo Server or System server name 1]
            [Tuxedo Server or System server name 2]
            [Tuxedo TMS name 1]
            [Tuxedo TMS name 2]
            ...
          [Tuxedo Group 2 name]
        ...
      [Tuxedo Machine 2 name]
    ....
```

There are two node types in the tree: category node and target node.

Category node does not correspond to a Tuxedo target; it is used to group all the same type targets under a higher level target. Clicking on the category node can expand or collapse all child nodes under it. Clicking on a target node in the tree shows its Target home page in the content pane.

Target Menu

Provides a list of operations that you can perform on the selected target. The menu that appears depends on the target you select. The menu for a specific target contains the same operations as those in the Right-Click Target Menu.

Content Pane

Shows the monitoring information of the selected target, including:

- Summary

Configuration information collected from target Monitoring Configuration.

- Statistics

In this panel, some important metrics are demonstrated using graphs and tables. By using graphs, you can easily watch for trends; by using tables, you can examine details of past metric severity history.

- Configuration

The configuration information collected from UBB.

Right-Click Target Menu

Provides a list of operations that you can perform on the currently selected target. The menu is displayed when you right-click the target name in the target navigation pane.

UBB Modification

Enterprise Manager for Oracle Tuxedo provides a special UBB page for several targets, allowing you to modify UBB configuration from Enterprise Manager Cloud Control console. To enter the Modify UBB page, click **Control->Modify UBB** from the target top menu.

For more information, see [Oracle Tuxedo Documentation](#).

Note: To make the UBB modification function work properly, a `DOMAINID` is required to be configured in the `UBBCONFIG *RESOURCES` section.

Credentials Page

For a security domain where the `UBBCONFIG` file `SECURITY` parameter is not `NONE`, when you first time perform high security-level operations (such as shutting up/down the domain, viewing log messages, modifying UBB, and so on in its target page or underlying targets page), the Credentials page appears as shown in [Figure 1-3](#).

Figure 1-3 Credentials

The screenshot shows a web-based configuration window titled "Credentials". At the top, it states "This operation might require Oracle Tuxedo Domain Administrator credentials." Below this is a section titled "Oracle Tuxedo Domain Application Credentials" with a sub-header "Select Credential". The instruction "Select credential from one of the following options." is followed by three radio button options: "Preferred", "Named", and "New". The "New" option is selected. Below these are three text input fields labeled "* Tuxedo Username", "* Tuxedo Password", and "* Tuxedo Application Password". At the bottom, there are two checkboxes: "Save As" (checked) and "Set As Preferred Credentials" (unchecked). The "Save As" checkbox has a text input field next to it containing the value "NC_SHMSECBE_2013-03-04-174828".

To configure credentials in [Figure 1-3](#), do one of the following:

- Select **New** to create a new one. Set the following parameters:
 - Tuxedo Username/Password/Application Password: These values are used to join Tuxedo applications.
 - Save As: If this checkbox is selected, an Enterprise Manager credential is generated as well.
 - Set As Preferred Credentials: If this checkbox is selected, the newly-created credential is saved as the Preferred one.
- Select **Preferred** or **Named** to load a previous-saved credential. Click OK.

Notes:

- If Tuxedo Authentication and Authorization is **APP_PW**, even though Tuxedo does not use the **Tuxedo Username** and **Tuxedo Password** fields value to authenticate or authorize, you must input the two fields as place holders.
- The new created/selected credential is applied to subsequent Tuxedo administrative operations in the domain during the current session. To change the credential you need to log out of the Enterprise Manager console first.
- All the Named/Preferred Credentials can be managed by clicking **Setup -> Security** in the Enterprise Manager console.

Monitoring Tuxedo Targets

Before monitoring Tuxedo targets from Enterprise Manager Cloud Control, note the following:

- Make sure the Enterprise Manager OMS, Enterprise Manager Agent, and the DB instance time zones are the same. Take the following SQL script as an example to select SQL and modify the DB instance time zone:

```
SQL> select dbtimezone from dual;
```

```
SQL> select sessiontimezone from dual;
```

```
SQL> alter database set time_zone='<new time zone>';
```

```
SQL> alter session set time_zone='<new time zone>';
```

- If you change the collection interval, make sure:
 - the collection interval for the `SERVICE_STATISTICS` metric of all targets in a domain are the same;
 - the collection interval for the `CLIENT_CONNECTION_NUMBER` metric of domain and machine targets in a domain are the same;
 - the collection interval for the domain metric `DOMAIN_OPERATION_STATISTICS` is same to the one of machine metric `MACHINE_OPERATION_STATISTICS` within the domain.
- You can easily change the collection interval for a batch of targets by clicking **Enterprise->Monitoring->Monitoring Templates** to create a template and applying it to multiple targets.
- A 5 or 15 minutes target collection interval is highly recommended.

[Table 1-3](#) summarizes the Tuxedo targets you can monitor from Enterprise Manager Cloud Control and the references to the metrics each target supports and specific administrative operations you can perform on each target home page.

Table 1-3 Tuxedo Targets

Product	Targets	Ref to Supported Metrics	Operations Provided on Target Homepage
Tuxedo	Tuxedo Home	Tuxedo Home	
	Tuxedo Application Home	Tuxedo Application Home	
	Tuxedo tlisten	Tuxedo tlisten	
	Tuxedo Domain	Tuxedo Domain	Switching Tuxedo machine work mode
	Tuxedo Machine	Tuxedo Machine	<ul style="list-style-type: none"> • Migrating servers between two machines • Detecting dead clients or servers • Managing native clients • Aborting transactions • Viewing Tuxedo ULOG
	Tuxedo Group	Tuxedo Group	Migrating servers between two machines
	Tuxedo Server	Tuxedo Server	Managing services

Table 1-3 Tuxedo Targets

Product	Targets	Ref to Supported Metrics	Operations Provided on Target Homepage
Tuxedo	Tuxedo System Server	Tuxedo System Server	
	Tuxedo Bridge	Tuxedo Bridge	
	Tuxedo TMS	Tuxedo TMS	
	Tuxedo LMS	Tuxedo LMS	
	Tuxedo /T Domain Gateway	Tuxedo /T Domain Gateway	
	Tuxedo Web Service Gateway	Tuxedo Web Service Gateway	
	Tuxedo Workstation Listener	Tuxedo Workstation Listener	Viewing and managing handlers
	Tuxedo Jolt Listener	Tuxedo Jolt Listener	
	Tuxedo Event Broker	Tuxedo Event Broker	
	Tuxedo /Q Manager Server	Tuxedo /Q Manager Server	
OTMQ	Tuxedo OTMQ Manager Server	Tuxedo OTMQ Manager Server	Creating a Queue
ART	Tuxedo ART CICS Region	Tuxedo ART CICS Region	
	Tuxedo ART CICS Transaction	Tuxedo ART CICS Transaction	
	ART CICS TSQ Server	ART CICS TSQ Server	
	ART CICS TDQ Server	ART CICS TDQ Server	
	Tuxedo Batch System	Tuxedo Batch System	<ul style="list-style-type: none"> • JES Job Operations • GDG Management

Notes:

- When you change the collection interval of a Enterprise Manager console target, it takes effect on metrics data collection from the next interval.
- All metrics with repository-side type only process historical database context and cannot be viewed in real time; otherwise, an exception of `NoSuchMetricException` is printed in `oms log` (`$MW_HOME/gc_inst/em/EMGC_OMS1/sysman/log/emoms.log`).

Viewing TUXCONFIG / BDMCONFIG

Enterprise Manager ECM CCS (Customer Configuration Specification) is used to show Enterprise Manager Agent side configuration files, such as file name, full path name, content size, and content details.

Tuxedo Configuration files, `TUXCONFIG` and `BDMCONFIG`, are shown by CCS.

To show `TUXCONFIG` or `BDMCONFIG`, click **Tuxedo Domain Target Instance Home Menu**, and click **Configuration > Last Collected**. `TUXCONFIG.cfg` / `BDMCONFIG.cfg` is displayed under the Configuration Files tree node. If the `TUXCONFIG.cfg` or `BDMCONFIG.cfg` is missing under Configuration Files tree node after the Tuxedo Domain is discovered first time, click the tree root at the left panel. Click **Actions** at right panel, and click **Refresh**.

Click `TUXCONFIG.cfg` or `BDMCONFIG.cfg` under Configuration Files tree node to view each file details.

`BDMCONFIG.cfg` is shown only if it has been configured at Tuxedo side.

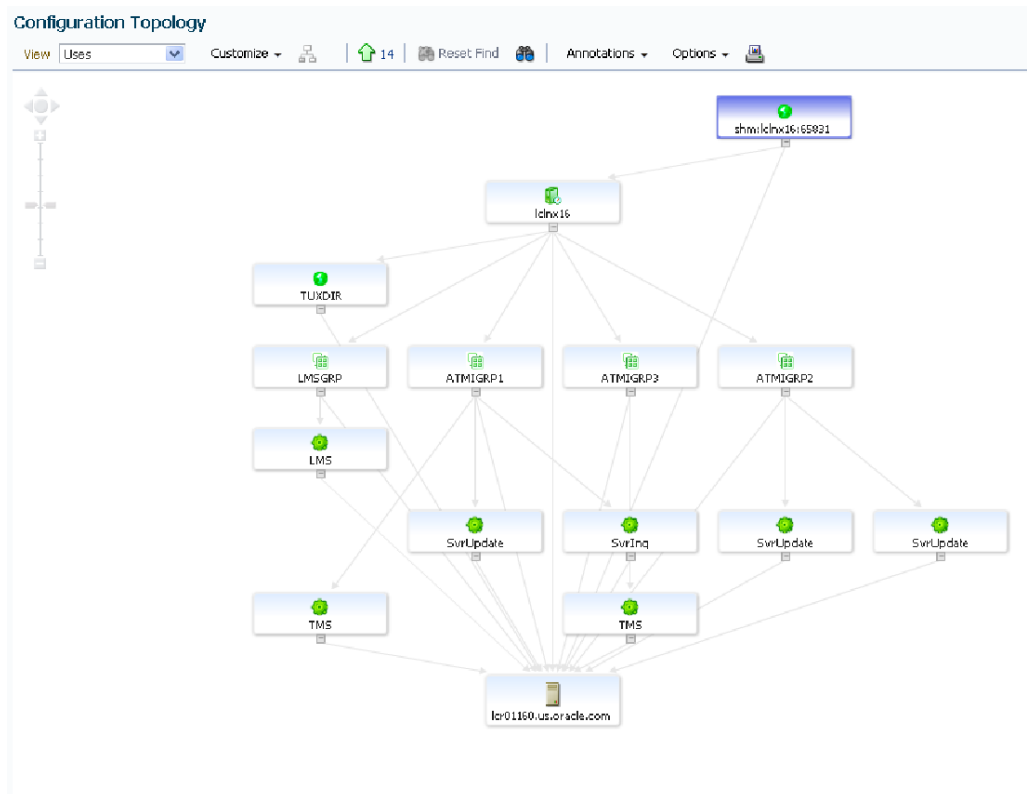
Notes:

- The details of `TUXCONFIG` and `BDMCONFIG` are gotten from decompilation by `tmunloadcf` or `dmunloadcf`. Actually there is no `TUXCONFIG.cfg` or `BDMCONFIG.cfg` disk file at Enterprise Manager agent side as normal CCS module. The file name, full name, and content size of `TUXCONFIG.cfg` or `BDMCONFIG.cfg` are virtual compared with other ECM CCS module.
- `*JDBCCONNPOOLS` is ignored after running `tmunloadcf` if the `*JDBCCONNPOOLS` section is empty; if the `*JDBCCONNPOOLS` section is not empty, `*JDBCCONNPOOLS` is shown as other sections.

Viewing Configuration Topology

For targets on a JMX agent, you can view the configuration topology by clicking **Members -> Topology** from the Tuxedo Domain context menu. Select **Uses** in the View drop-down list, as shown in [Figure 2](#).

Figure 2 Configuration Topology



TuxJES Monitoring

Enterprise Manager for Oracle Tuxedo enables you to monitor the Tuxedo Job Enqueueing Service (TuxJES) from Enterprise Manager Cloud Control.

Oracle Tuxedo-Side Configuration

Before the JES target instance can receive job statistics, do the following steps:

1. Set the environment variable `JESMONITOR=yes` before starting the Tuxedo domain.
2. Configure `UBBCONFIG` file `TMSYSEVT` and `TMUSREVT` parameters.
3. Configure `beamgr.conf` `TRAP_HOST` and `TMAGENT` parameters. For more information, refer to [Oracle SNMP Agent Configuration File: beamgr.conf](#).

For Enterprise monitoring, the Trap host is the Enterprise Manager agent host, and the port is the Enterprise Manager Agent SNMP listening port which will be configured at Enterprise Manager Agent side (see [Monitoring Metrics](#)).

Note the following example:

```
TRAP_HOST 10.182.54.215 1061 public
SNMP_ENABLE_AUTH_TRAP 1
TMAGENT tuxedo_agent_1 /home/oracle/tuxedo12c
/home/oracle/test/jmx/fakemp/master/tuxconfig
```

4. Set the `BEA_SM_SNMP_MIBFILE` and `BEA_SM_BEAMGR_CONF` environment variables.

Note the following example:

- `BEA_SM_SNMP_MIBFILE=${TUXDIR}/udataobj/snmp/etc/mib.txt`
- `BEA_SM_BEAMGR_CONF=<the directory for beamgr.conf>`

5. Start Oracle SNMP Agent. For more information, refer to [Starting Oracle SNMP Agent](#).

Following is an example:

```
tux_snmpd -l tuxedo_agent_1 -p 5400 -s
```

Note: If your Enterprise Manager agent located at the same machine as your Tuxedo domain, make sure the port number followed by "-p" option is not equal to the trap port configured in `beamgr.conf`.

Discovery

TuxJES monitoring discovery is done during the Tuxedo domain discovery process. After Tuxedo domain discovery is done, the JES target instance is created.

Login to TuxJES Monitoring Homepage

To log into TuxJES monitoring homepage, do the following steps:

1. From Enterprise Manager Cloud Control, click a Batch system target.
2. If the Batch system is on a Tuxedo domain where Privilege Mode parameter is not `NONE` (See [PRIVILEGE MODE](#)), you are required to log in with credentials. Click **Login**.
3. In the **Credentials** page, do one of the following:
 - Select **Preferred** or **Named** to load a previous-saved credential. Click **OK**.
 - Click **New** to create a new one. Set the following parameters:
 - **Tuxedo Username/Password/Application Password:** The login information used for joining the Tuxedo domain. The user name is used as the job owner when submitting a job, or the requestor when controlling jobs. The acceptable length is from 0 to 30 characters. If any of these three credential fields is input incorrectly or not specified, you cannot enter the Batch System target homepage.
 - **Connection String:** Optional. The acceptable length is from 0 to 2047 characters. You can type `NONE` in the field as an empty value.

Note: An Enterprise Manager user cannot change Batch System credentials once logged in until the credentials expire or you log out from the Enterprise Manager Console.

After you log in to the Batch system target home page, if the Tuxedo domain Privilege Mode parameter is `NONE` (See [PRIVILEGE MODE](#)), when you first click **Submit** on the page, a dialog box appears requiring a connection string. Once input, the connection string is saved in the current session until the session expires.

Monitoring Metrics

Tuxedo Batch System is a system target type used to monitor the metrics generated by JES jobs. For more information, see [ART Batch Targets](#).

Before the Tuxedo Batch System can receive JES job statistics, you need to configure the SNMP port. Do the following steps:

1. From the Tuxedo Batch System homepage, click the target menu. Select **Target Setup -> Monitoring Configuration**.
2. Set the SNMP port in the **Receive Port for SNMP Agent** field to the value configured in `beamgr.config`.

Operations on UI Console

You can view all available jobs and perform a set of Job operations by clicking the JES Jobs tag from the Tuxedo JES Admin target homepage. For more information, see [JES Job Operations](#).

Creating an Enterprise Manager Job for Tuxedo Batch Operation

From Enterprise Manager Cloud Control, you can create the following types of JES jobs:

- **Tuxedo Batch Process Control:** This job makes one of these operations: submit, purge, hold, release, or cancel a Batch job.
- **Tuxedo Batch Synchronize Job:** This job synchronizes all the JES Jobs related metrics data with the job statistics stored in JESROOT.
- **Purge Tuxedo Batch Jobs:** This job purges Batch Jobs at a scheduled time automatically.

To create an Enterprise Manager job, do the following steps:

1. From Enterprise Manager Cloud Control, click **Enterprise** -> **Job** -> **Activity**.
2. In the **Activity** page, select the job type in the **Create Job** list and click **Go**.
3. Type a name and description for the Enterprise Manager job.
4. Depending on the job type you selected, do one of the following:
 - For the Tuxedo Batch Process Control job, click **Add** to add a Tuxedo Batch System target for this job. Click the **Parameters** tab and select an operation from the **Operation** list and input the related parameters. Following operations can be made for the Batch job:
 - Submit a Batch job
 - Purge a Batch job
 - Hold a Batch job
 - Release a Batch job
 - Cancel a Batch job
 - For the Tuxedo Batch Synchronize Job, click the **Parameters** tab and select one or multiple Tuxedo domains in the **Domains** list. Only Batch domains are listed.

- For the Purge Tuxedo Batch Jobs, click the **Parameters** tab and set the day/week time in the **Older than** text box. The JES Jobs older than that day/week will be purged. Select one or multiple Tuxedo domains listed in the **Domain** list. Only Batch domains are listed.
5. Click the **Credentials** tab and specify the credential information by one of the following:
 - Select an existing credential in the Preferred or Named category
 - Create a new credential by filling in the required information

In the case that Tuxedo security is not enabled, you can create a Tuxedo Batch System Credential with any value for **Tuxedo Username**, **Tuxedo Password**, and **Tuxedo Application Password**.

In the case that Tuxedo security is enabled, the Tuxedo Batch System credentials must be provided to make the JES Job operations. The credential provided here may be the same with the one used on the JES Jobs query page.

6. Click the **Schedule** tab and specify the Batch Job schedule. The Batch Job can be submitted immediately or at the scheduled time.
7. Click **Submit**.

The Enterprise Manager job status is shown in the Enterprise Manager Job Activity page. You can view the Batch Job status from the Tuxedo Batch System target page.

You can also manually synchronize JES jobs by clicking the **Synchronize** button in the Tuxedo Batch System target home page.

GDG Management

Enterprise Manager for Oracle Tuxedo enables you to manage Generation Data Group (GDG) files from Enterprise Manager Cloud Control.

Prerequisites

Before the GDG can receive data, do the following steps:

1. Make sure Enterprise Manager, ART Batch, and Oracle Tuxedo are installed.
2. Set up a Tuxedo domain (use the sample `$ART_HOME /Batch_RT/sample/simpjob` file in ART Batch). Set the environment variable `JESMONITOR=yes`.

3. Set the GDG management as DB-based in

`$ART_HOME/Batch_RT/ejr/CONF/BatchRT.conf`. The values that need to be set in `BatchRT.conf` are:

- a. `MT_GENERATION=${MT_GENERATION:-GENERATION_FILE_DB}`
- b. `MT_DB=${MT_DB:-DB_ORACLE}`
- c. `MT_GDG_DB_ACCESS=${MT_GDG_DB_ACCESS:-"tsam/tsam@tsam"}`

4. Copy the job template

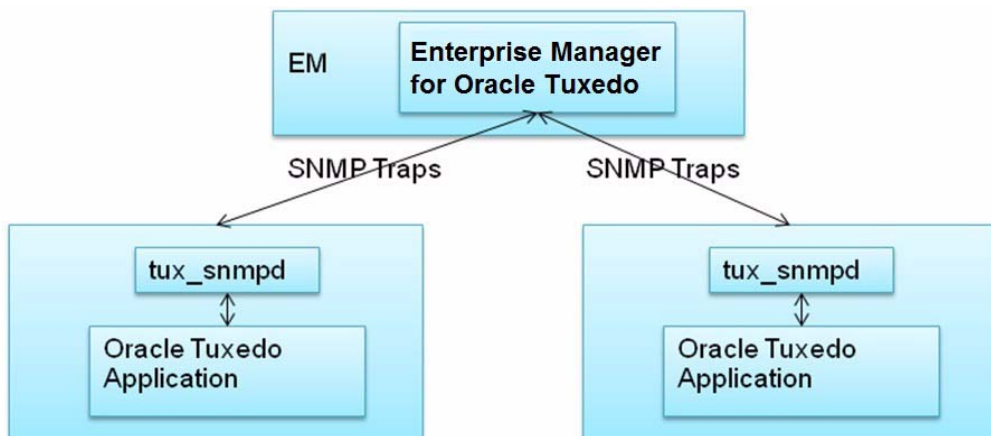
`$JESDIR/ejr_mf_ora/pp/macro/template/GDG_PREDEFINED_JOB` to the directory where TuxJES can be found. If `JOBREPOSITORY` is set in the `jesconfig` file, put it under `JOBREPOSITORY`; otherwise, put it under `APPDIR`.

Operations on UI Console

You can manage GDG files by clicking the GDG tag from the Tuxedo JES Admin target homepage. For more information, see [GDG Management](#).

Tuxedo Event Monitoring

Figure 2-1 Tuxedo Event Monitoring Structure



Every Tuxedo event trap on the domain target is received by the SNMP Receivlet in Enterprise Manager for Oracle Tuxedo. The Enterprise Manager for Oracle Tuxedo generates an Enterprise Manager metric alert with a severity of "Critical" or "WRAN" for the received Tuxedo event traps. "Critical" means the Tuxedo event trap severity is Error (1), while "WRAN" means the

Tuxedo event trap severity is Warn (2). The severity Info (3) is not supported. Every alert generated by Tuxedo event trap creates an EM incident automatically if the SNMP listening port is configured correctly.

The event message is in the following format:

```
"EventName: <tuxEventsName>. Tuxedo Event LMID: <tuxEventsLmid>. Tuxedo
Event Detection Time: <tuxEventsTime>. Tuxedo Event Class:
<tuxEventsClass>. Tuxedo Agent Name:<beaLogicalAgentName>. Description:
<tuxEventsDescription>"
```

For more information, see Oracle Tuxedo Documentation for Event Traps.

To enable Tuxedo events collection in EM, do the following steps:

1. Configure TMSYSEVT in UBBCONFIG(5) file.
2. Configure the TRAP_HOST and TMAGENT in beamgr.conf. For more information, refer to [Oracle SNMP Agent Configuration File: beamgr.conf](#).

Note that the Trap host is the EM agent host and the port is the Enterprise Manager Agent SNMP listening port which will be configured at Enterprise Manager Agent in step 5.

Following is an configuration example.

```
TRAP_HOST          10.182.54.215 1061 public
SNMP_ENABLE_AUTH_TRAP 1
TMAGENT tuxedo_agent_1 /home/oracle/tuxedo12c
/home/oracle/test/jmx/fakemp/master/tuxconfig
```

3. Set the environment variable of BEA_SM_SNMP_MIBFILE and BEA_SM_BEAMGR_CONF.
4. Start Oracle SNMP Agent. For more information, refer to [Starting Oracle SNMP Agent](#).

Following is an example:

```
tux_snmpd -l tuxedo_agent_1 -p 5400 -s
```

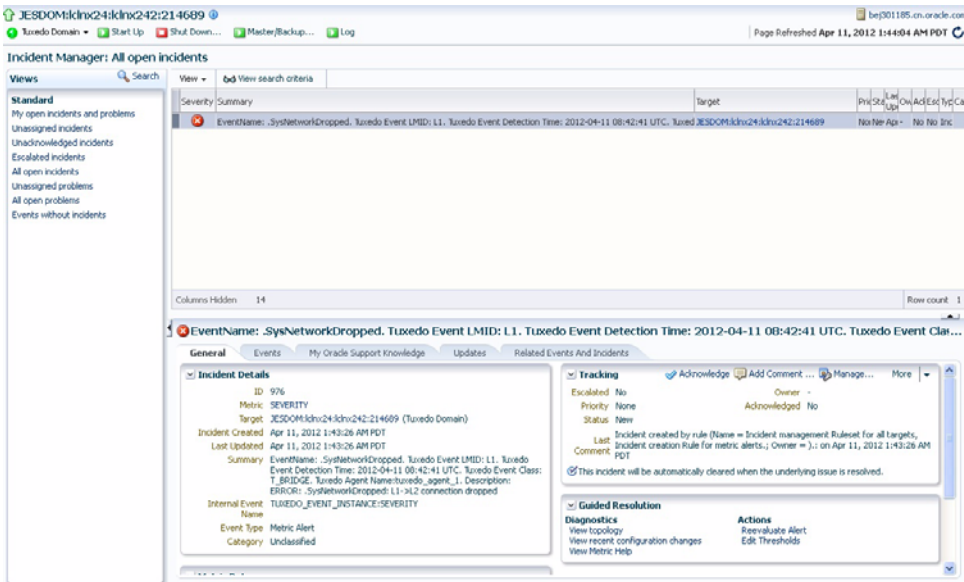
Note: If your Enterprise Manager agent located at the same machine as your Tuxedo domain, make sure the port number followed by "-p" option is not equal to the trap port configured in beamgr.conf.

5. Add SNMP listening port in the Tuxedo domain target configuration page.

Note: If one EM agent monitors multiple Tuxedo domains, each Tuxedo domain target SNMP listening port must be unique within one EM agent.

[Figure 2-2](#) shows an incident created by EM and caused by Tuxedo event trap.

Figure 2-2 Incident Example



Note: Tuxedo Event Monitoring can monitor user events when the post data is a 32-bit fielded buffer containing the fields listed in Table 2-1. The Tuxedo user events generated by TSAM Plus alert definitions can be trapped by the Tuxedo SNMP agent and subsequently monitored by Enterprise Manager for Oracle Tuxedo. Also, the TMUSREVT and TMSYSEVT must be configured for user event collection.

Table 2-1 FML Fields

FML Field Name	Name in Event Trap	Description
TA_EVENT_NAME	tuxEventsName	Required. String[1...31]
TA_EVENT_SEVERITY	tuxEventsSeverity	Required. In the format of "ERROR", "WARN", "INFO"
TA_EVENT_LMID	tuxEventsLmid	Optional. String[1...30]
TA_EVENT_TIME	tuxEventsTime	Optional. Time, in seconds, since 00:00:00 UTC, January 1, 1970

Table 2-1 FML Fields

FML Field Name	Name in Event Trap	Description
TA_EVENT_DESCRIPTOR	tuxEventsDescription	Optional. String[1...1118]
TA_CLASS	tuxEventsClass	Optional. String[1...30]

[Listing 2-1](#) shows an example of user event that is trapped by Enterprise Manager for Oracle Tuxedo.

Listing 2-1 Example of User Event Trapped by Enterprise Manager for Oracle Tuxedo

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <errno.h>
#include <time.h>
#ifdef WIN32
#include <sys/types.h>
#endif
#include "atmi.h"
#include "userlog.h"
#include "Uunix.h"
#include <evt_mib.h>

#define NFLOATS      50      /* Number of string fields */

#define LEN (NFLOATS*sizeof(float))
static long long_v;
#define LFADDR(x)    ( long_v=(x), (char*)&long_v )
```

Using the Enterprise Manager for Oracle Tuxedo

```
int
main(int argc, char *argv[])
{
    long len;
    int error;
    time_t currtime;
    FBFR32 * buf;
    char decTime[14];

    if (tpinit((TPINIT *)NULL) == -1) {
        (void)fprintf(stderr, "Failed to join application -- %s\n",
            tpstrerror(tperrno));
        (void)userlog("Clientpost failed to join application -- %s\n",
            tpstrerror(tperrno));
        (void)exit(1);
    }

    if ((buf = (FBFR32 *)tpalloc("FML32", NULL, Fneeded32(NFLOATS, LEN))) ==
    NULL) {
        (void)fprintf(stderr, "Failure to allocate FML32 buffer -- %s\n",
            tpstrerror(tperrno));
        (void)userlog("Clientpost failure to allocate FML32 buffer -- %s\n",
            tpstrerror(tperrno));
        (void)tpterm();
        (void)exit(1);
    }

    if (Fadd32(buf, TA_EVENT_NAME, "Test Event", 0) == -1) {
```

```

    (void)fprintf(stderr, "Failed to add TA_EVENT_NAME field -- %s\n",
Fstrerror32(Ferror32));

    (void)tpfree((char *)buf);

    (void)tpterm();

    (void)exit(1);
}

    if(Fadd32(buf, TA_EVENT_SEVERITY, "ERROR", 0) == -1){

        (void)fprintf(stderr, "Failed to add TA_EVENT_SEVERITY field -- %s\n",
Fstrerror32(Ferror32));

        (void)tpfree((char *)buf);

        (void)tpterm();

        (void)exit(1);

    }

    currtime = time(NULL);

    if(Fadd32(buf, TA_EVENT_TIME, LFADDR(currtime), 0) == -1){

        (void)fprintf(stderr, "Failed to add TA_EVENT_TIME field -- %s\n",
Fstrerror32(Ferror32));

        (void)tpfree((char *)buf);

        (void)tpterm();

        (void)exit(1);

    }

    if(tppost("Test Event", (char *)buf, 0, TPSIGRSTRT) == -1)
    {

        (void)fprintf(stderr, "Failure to post request -- %s\n",
                        tpstrerror(tperrno));

        (void)userlog("Clientpost failed to post request -- %s\n",

```

```
        tpstrerror(tperrno));  
  
    /* Free any allocated buffers, leave the application, and exit */  
    (void) tpfree((char *)buf);  
    (void) tpterm();  
    (void) exit(1);  
}  
  
(void) tpfree((char *)buf);  
(void) tpterm();  
  
exit(0);  
}
```

Reporting for Tuxedo Targets

Reporting for monitored Tuxedo targets adopts the Enterprise Manager GC solution. BI publisher is a high customizable product (and is one recommended reporting solution for Enterprise Manager GC).

Follow the *BI Publisher Installation Guide* to install BI on your existing Enterprise Manager. After installing BI, you can go to the BI Publisher Enterprise Reports page by clicking on **Enterprise -> Reports -> BI Publisher Enterprise Reports** from the Enterprise Manager home page.

The new Webapp opens. You can only use the Enterprise Manager administrator user **sysman** to log into the new Webapp.

For more information, see [Oracle Enterprise Manager 12c: Reporting with BI Publisher](#).

Using Enterprise Manager SLA

In order to use Enterprise Manager Service Level Agreement (SLA) functionality with Enterprise Manager for Oracle Tuxedo, you need to define Generic Service target first. In Enterprise

Manager for Oracle Tuxedo, Generic Service targets can only be defined upon the following Enterprise Manager System targets:

- Tuxedo Domain
- Tuxedo Machine
- Tuxedo Group
- Tuxedo ART CICS region
- Tuxedo ART Batch system

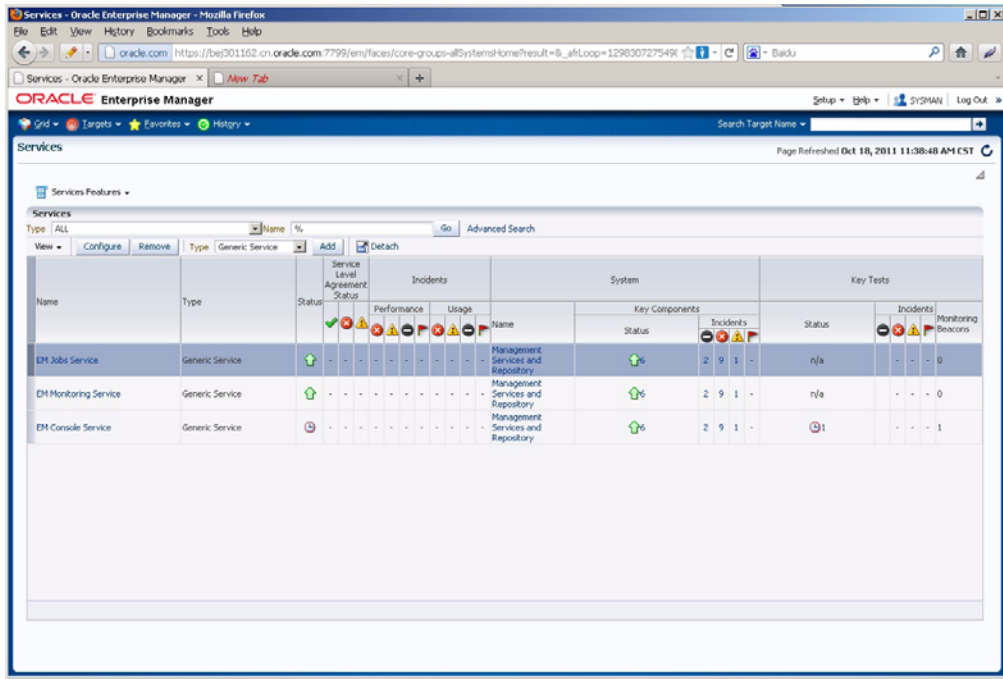
From the Enterprise Manager home page, click **Targets->Services** at the top actions bar to open the Services dashboard. On the Services page, click **Help** on the upper right corner for more information.

Creating Enterprise Manager Services

Tuxedo SLA is managed by Enterprise Manager Services SLA. To create an Enterprise Manager service, do the following steps:

1. From the Enterprise Manager home page, click **Targets->Services**. The Service page appears as shown in [Figure 2-3](#).

Figure 2-3 Services Page



2. Click **Add** to create Enterprise Manager services SLA, as shown in [Figure 2-4](#) and [Figure 2-5](#). For more information, see online Oracle Enterprise Manager Documentation.

Figure 2-4 Creating Enterprise Manager Services SLA

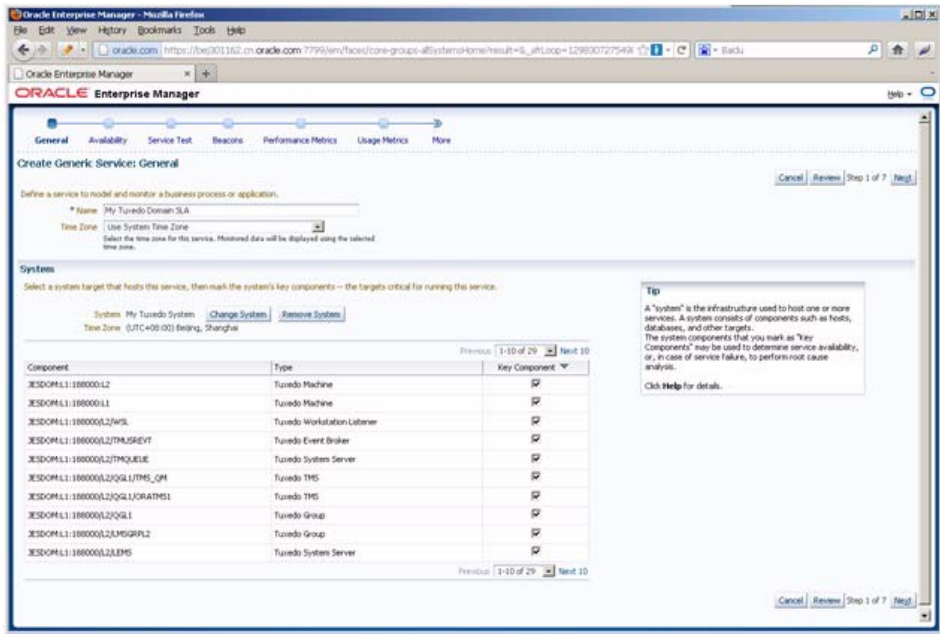
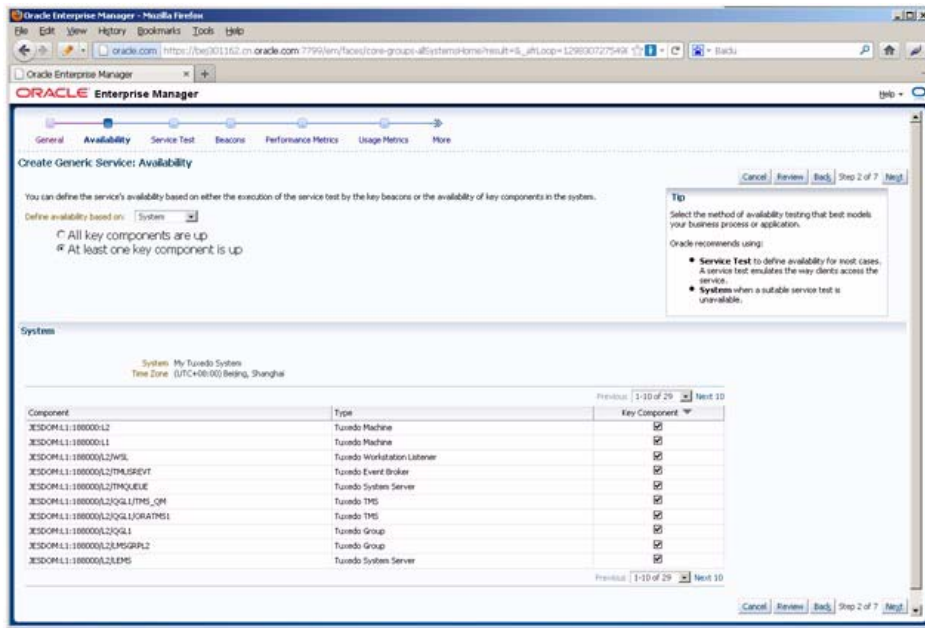


Figure 2-5 Creating Enterprise Manager services SLA (Continued)



Using Resource Broker

This section contains the following topics:

- [About Resource Broker](#)
- [Application Package Organization and Content](#)
- [Application Package Management](#)
- [Domain UBBCONFIG Editor](#)
- [Deploying Application Packages to Tuxedo Domain](#)
- [Deploying an Application Package Dynamically](#)
- [Updating a Deployed Application Package](#)
- [Dynamic Resource Broker](#)
- [Controls on Tuxedo Domain Target](#)

- [Using Resource Broker for ART Batch](#)
- [Using Resource Broker for ART CICS](#)

About Resource Broker

Resource Broker is a major Enterprise Manager for Oracle Tuxedo feature that coordinates and allocates resources elastically based on predefined Oracle Tuxedo application policies. It includes the following major functions:

- Manages Oracle Tuxedo application functions by organizing them to application packages.
- Creates a new Tuxedo domain based on application packages using Domain Editor.
- Edits both existing Tuxedo domains and new Tuxedo domains created by Resource Broker using Domain Editor.
- Dynamically applies application package content to a running Oracle Tuxedo domain. The deployment can be performed manually or automatically using predefined policies.
- Dynamically adds slave machines to a running Oracle Tuxedo domain. The deployment can be performed manually or automatically using predefined policies.
- Dynamically adjusts the number of certain Tuxedo servers according to the predefined deployment policies.

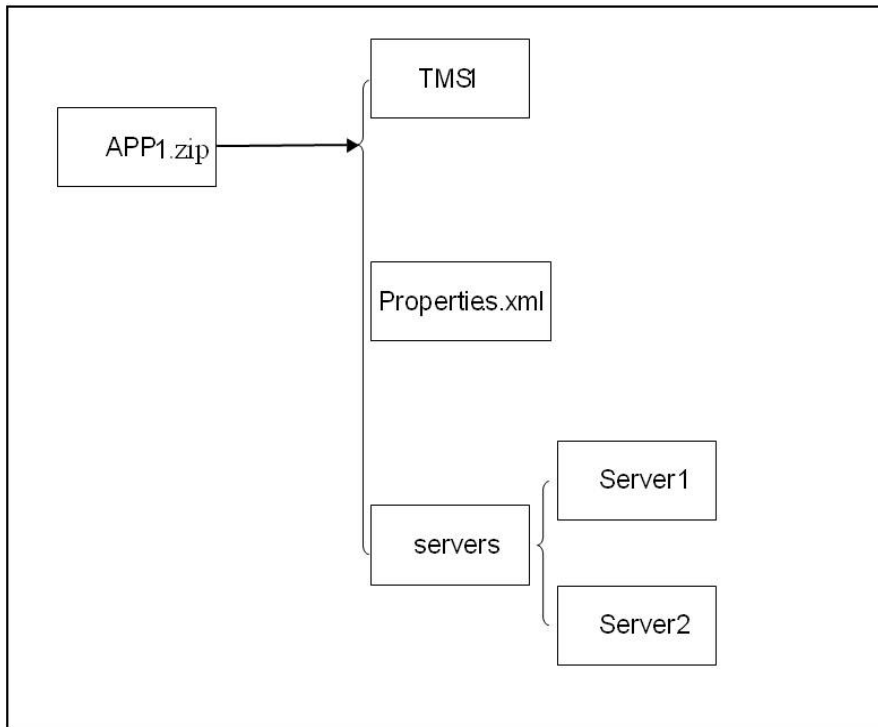
Application Package Organization and Content

A Tuxedo application (domain), as defined in a `TUXCONFIG` (`UBBCONFIG`) configuration file, is a set of machines, servers, and other resources. It can exist in a single machine or cross multiple network-connected machines. To deploy the entire Tuxedo applications automatically, first you need to upload your application packages from the Enterprise Manager Console. An application package (the basic unit of an application), may contain several Tuxedo application servers and other resources of a Tuxedo group. A Tuxedo application consists of one or more application packages.

One application package can be applied to different domains, one domain on different machines or one domain on the same machine repeatedly.

An application package is a zip file and contains all of the Tuxedo group consisting files that are needed by the Tuxedo system (such as application servers, TMS servers, `ENVFILES`, application level configuration files, etc.). The application package structure is shown in [Figure 2-6](#).

Figure 2-6 Application Package Structure



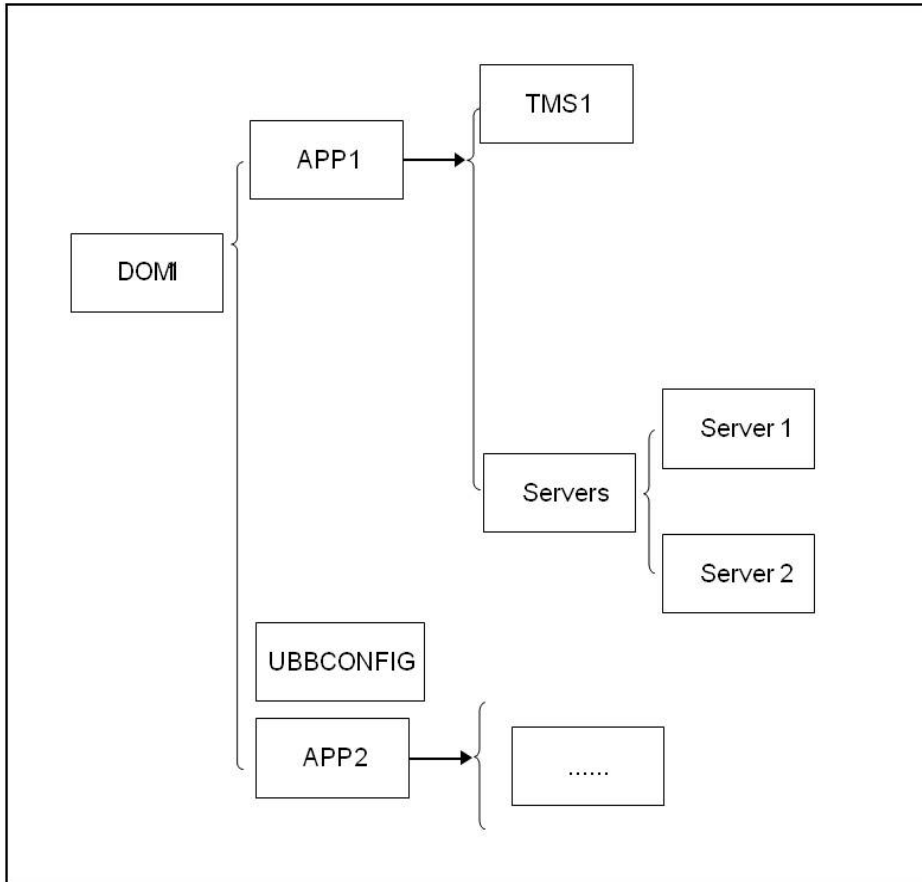
- The application package name must be unique within all the uploaded application packages.
- The application package name must consist of 64 characters or less, without containing asterisk (*), comma, or colon.

Under the first directory, you can also create sub-directories to organize your applications. As shown in above graphics, there are two application servers, Server1 and Server2, located in the sub-directory servers.

Deployed Application Package Structure On Machine

[Figure 2-7](#) shows the machine application structure after the package is deployed on the machine. The application structure under DOM1 remains the same as the original application packages.

Figure 2-7 Application Package Structure After Deployed on Machine



Properties.xml

The "Properties.xml" file is a group-level UBBCONFIG file which describes the relationship and parameters of all the servers within groups in the package. It contains properties in the *GROUPS, *RMS, *SERVERS, and *SERVICES sections of a complete UBBCONFIG file and can contain multiple groups.

The Properties.xml file is used to generate the final UBBCONFIG file when deploying the package to a machine. Its content can be modified at that time.

A simple Properties.xml file is shown in [Listing 2-2](#).

Listing 2-2 Adding Pathname to Server Names

```
<ApplicationProperties>
  <PackageName>APP1.zip</PackageName>
  <TuxedoVersion>12.1.3.0.0</TuxedoVersion>
  <SupportedOS>Linux</SupportedOS>

  <TuxedoWordSize>64</TuxedoWordSize>
  <MachineArch>x86_64</MachineArch>
  <LibPath>/path/to/libs</LibPath>

  <Groups>
    <GroupSection GROUPNAME="G1" GRPNO="29999">
      <ServerSections>
        <ServerSection AOUT="servers/simpserv1" SRVID="20">
          </ServerSection>
        </ServerSections>
      </GroupSection>
    </Groups>
  </ApplicationProperties>
```

Besides the Tuxedo group related information in the `Groups` element, there are some package global attributes defined at the beginning of the `Properties.xml` file as shown in [Table 1-2](#).

Table 2-2 Description Information of the Package

Item	Description
PackageName	The global unique application package name, for example, APP1.zip.
TuxedoVersion	The Tuxedo version this package is built on. This item is checked when assembling Tuxedo domain to decide if the package is suitable for certain Tuxedo installation.
SupportedOS	The Operation System this package can be deployed to. This information is compared with the corresponding item in Machine list entry when deploying the package. The possible values are Linux, SunOS, HP-UX, and AIX.
TuxedoWordSize	The Tuxedo word size this package is built on. The value can be 32 or 64 (bit). This information is compared with the corresponding item in Machine list entry when deploying the package.
MachineArch	The machine architecture this package can apply to. The possible values are: x86_64, SUNW, powerpc, IA64, and etc. Check the Platform property (Target Setup -> Properties) of a Host target to decide the appropriate value of the MachineArch here. Only when the MachineArch property value is a substring of the host Platform property, the package can be deployed on the target machine.
LibPath	The path where the library locates in the package, if there is a library in it.

All parameters in the UBBCONFIG file *GROUPS, *RMS, *SERVERS, and *SERVICES sections are divided into three categories, which determines how they are defined in `Propertise.xml`.

- Forbidden

Indicates this parameter cannot be defined in the application packages `Properties.xml` file. It can only be filled in when assembling the UBBCONFIG in Resource Broker domain editor.

- Optional

Indicates this parameter can be filled in the `Properties.xml` if needed, and it can also be modified when assembling the UBBCONFIG.

- Mandatory

Indicates this parameter must be filled in the Properties.xml, and it can also be modified when assembling the UBBCONFIG.

Table 2-3 lists the category of each parameter in the *GROUPS, *RMS, *SERVERS, and *SERVICES sections.

Table 2-3 UBBCONFIG Parameter Categories

Section	Parameter	Category
GROUPS	GROUPNAME	Mandatory
	GRPNO = number	Mandatory
	LMID = string_value1 [,string_value2]	Forbidden
	ENVFILE = string_value	Optional
	TMSNAME = string_value	Optional
	MRM = {Y N}	Optional
	SIGNATURE_REQUIRED = {Y N}	Optional
	ENCRYPTION_REQUIRED = {Y N}	Optional
	OPENINFO = string_value	Optional
	CLOSEINFO = string_value	Optional
	TMSCOUNT = number	Optional
	SEC_PRINCIPAL_NAME = string_value	Forbidden
	SEC_PRINCIPAL_LOCATI ON = string_value	Forbidden
	SEC_PRINCIPAL_PASSVA R = string_value	Forbidden

Table 2-3 UBBCONFIG Parameter Categories

Section	Parameter	Category
RMS	RMSNAME	Mandatory
	RMID = number	Mandatory
	TMSNAME = string_value	Optional
	OPENINFO = string_value	Optional
	CLOSEINFO = string_value	Optional
	TMSCOUNT = number	Optional
	AUTO = {Y N}	Optional

Table 2-3 UBBCONFIG Parameter Categories

Section	Parameter	Category
SERVERS	SRVID = number	Mandatory
	AOUT	Mandatory
	ENVFILE = string_value	Optional
	RCMD = string_value	Optional
	CONV = {Y N}	Optional
	CLOPT = string_value	Optional
	SEQUENCE = number	Optional
	RQADDR = string_value	Optional
	MIN = number	Optional
	MAX = number	Optional
	RQPERM = number	Optional
	REPLYQ = {Y N}	Optional
	RPPERM = number	Optional
	MAXGEN = number	Optional
	GRACE = number	Optional
RESTART = {Y N}	Optional	

Table 2-3 UBBCONFIG Parameter Categories

Section	Parameter	Category
SERVERS	SYSTEM_ACCESS = identifier[, identifier]	Optional
	MAXDISPATCHTHREADS = number	Optional
	MINDISPATCHTHREADS = number	Optional
	THREADSTACKSIZE = number	Optional
	SEC_PRINCIPAL_NAME = string_value	Forbidden
	SEC_PRINCIPAL_LOCATION = string_value	Forbidden
	SEC_PRINCIPAL_PASSWORD = string_value	Forbidden
	SICACHEENTRIESMAX = string_value	Optional
	CONCURR_STRATEGY=PER _REQUEST	Forbidden
CONCURR_STRATEGY = PER_OBJECT	Forbidden	

Table 2-3 UBBCONFIG Parameter Categories

Section	Parameter	Category
SERVICES	SRVGRP = string_value	Forbidden
	SVCNM	Mandatory
	BUFTYPE = "type1[:subtype1[,su btype2 . . .]][:type2[:subtype3[, . . .]]] . . .	Optional
	SIGNATURE_REQUIRED = {Y N}	Optional
	ENCRYPTION_REQUIRED = {Y N}	Optional
	LOAD = number	Optional
	PRIO = number	Optional
	BUFTYPECONV = {XML2FML XML2FML32}	Optional
	BLOCKTIME numeric_value	Optional
	SVCTIMEOUT = number	Optional
	SESSIONROLE	Optional
	AFFINITYSCOPE	Optional
	AFFINITYSTRICT	Optional
	AUTOTRAN = {Y N}	Optional
	ROUTING = string_value	Forbidden
TRANTIME = number	Optional	

For more information, see [Properties.xml Schema](#) in [Enterprise Manager for Oracle Tuxedo Reference](#).

To make the TMS executable file (generated by command `buildtms`), accessible for Tuxedo application runtime when running `tmboot`, the TMS executable file should be located in sub-directory 'bin' of the application package. The `<package_name>/bin` folder is appended to the environment variable `PATH` in the generated `setenv` file. The `setenv` file is sourced before booting up the Tuxedo application.

For the `ENVFILE` mentioned previously, `RCMD` files, and `control_file` specified by "`TMSYSEVT -f`", this feature does not support deploying them to directories other than `APPDIR`. For example, if `APPDIR` is `/nfs/lcfilec/vol1/APPDIR`, you cannot specify `ENVFILE=/nfs/lcfilec/vol2/envfile` in `UBBCONFIG` because `ENVFILE` is under `/nfs/lcfilec/vol1/APPDIR/APP1/...`

Note: All the Tuxedo servers dependent third party libraries (such as Database client libraries), are not in the deployment scope of this feature.

Application Package Management

After you have prepared the application packages, you need to upload them using the Enterprise Manager Console. This can be done on the Tuxedo Summary page. For more information, see [Viewing Tuxedo Summary](#).

Uploading a Package

To upload a package, do the following steps:

1. From the Tuxedo Summary page, click **Manage Application Package**.
2. Click **Browse**. Select your package in the File Upload dialog box.

Note: The package names must be unique among all the uploaded packages, otherwise the upload page reports an error.

Removing a Package

To delete an existing application package, do the following steps:

1. From the Tuxedo Summary page, click **Manage Application Package**.
2. Select one or more packages in the Application Package Name drop-down list. Click **Delete**.

Domain UBBCONFIG Editor

Domain UBBCONFIG editor is used for creating an Oracle Tuxedo domain or editing a discovered Oracle Tuxedo domain. To enter the Domain editor page, click **Add->Create Tuxedo Domain** on the Tuxedo Summary page; enter a domain name. Click **OK**. The domain editor initial page appears.

Note: The domain name must be global unique within current EM and consist of 0 to 256 characters.

Editor Header

The editor header on top of domain editor displays the following information:

- Name of the domain being created.
- Domain target status

It is shown by icons, indicating three status of Status Pending, Up, and Metrics Collection Error. The target status is refreshed every five minutes automatically.

- Domain deployment status

The same as the status shown on Oracle Tuxedo Summary page.

A refresh button is shown on the top right corner of the page. Clicking on the button refreshes the domain status, as well as, machine/package list.

Controls Panel

[Table 2-4](#) shows the Controls panel control buttons.

Table 2-4 Control Buttons

Control Button	Description
Save	<p>Saves the current domain configuration. You can load the saved domain by clicking Edit on the Tuxedo Summary page.</p> <p>When saving the domain, the corresponding UBBCONFIG file is generated according to the parameters defined in each section and verified thoroughly. Once the domain is saved it is ready to be deployed.</p>
Save & Deploy	<p>Deploys the current domain directly in the domain editor page. Save action is performed automatically for each deployment.</p> <p>There are two kinds of deployment: full (static) deployment and dynamic deployment. The deploy type of each deployment depends on the domain status.</p>
Policy Management	<p>Enters the Policy Management console where you define and manage policies centrally.</p> <p>Policies define the conditions that trigger a dynamic deployment/undeployment of application packages and machines and automatic server spawn/decay.</p> <p>For more information, see Policy Management or click Help on the upper right of the page.</p>
Security Management	<p>Enters the Security Management page. If you enable Tuxedo SECURITY while assembling a Tuxedo application, you need to configure certain authentication and authorization related information in the Security Management page. For more information, see Security Management or click Help on the upper right of the page.</p>
DMConfig	<p>Lets you configure each section parameters of DMCONFIG. For more information, see Creating Domain Config or click Help on the upper right of the page.</p>
JESConfig	<p>Lets you configure the Tuxedo system servers that are specific to ART Batch in the Tuxedo domain. For more information, see JES Configuration or click Help on the upper right of the page. For the existing domain that is not ART Batch domain, this button is disabled.</p>

Table 2-4 Control Buttons

Control Button	Description
CICSConfig	Lets you configure the Tuxedo system servers that are specific to ART CICS in the Tuxedo domain. For more information, see Using Resource Broker for ART CICS . For the existing domain that is not ART CICS domain, this button is disabled.
Packages	Brings up the application package management dialog (the same as the one on the Tuxedo Summary page).
Procedures	Opens the Procedure Activity page in a new tab.
Save & Close	Saves current domain configuration and returns to the Tuxedo Summary page.
Cancel	Cancels current domain configuration and returns to the Tuxedo Summary page.

Machine List Panel

The Machine List panel lists all machines managed by EM that are available for deploying Tuxedo domains. Machines used for Tuxedo domain deployment should meet the following two prerequisites:

- The host is being monitored by Enterprise Manager.
That is, there is one Enterprise Manager agent deployed on that host, and the corresponding agent and host targets have been shown on the All Targets page. The Tuxedo domain can only be deployed to a host with Enterprise Manager agent running locally.
- `tlisten` and Tuxedo Home targets have been discovered and monitored by the Enterprise Manager agent on that host.

Notes:

- It is strongly suggested that you use the full machine name when adding Enterprise Manager agent and starting the Tuxedo `tlisten` process. Resource broker may not work properly with short machine names.
- The standalone targets added by **Add Non-Host Targets by Specifying Target Monitoring Properties** in the Discovery page are not supported by Resource Broker.

Once the machine is added to the current domain by clicking '+' button for one machine entry, machine information appears as tooltips when pausing the mouse cursor over the machine entry.

Package List Panel

The package list panel (lower right) lists all the uploaded packages t. Also, pausing the mouse cursor over the package name displays detailed package information.

UBB Section Panels

On the left side, are the UBB section panels. Each UBBCONFIG file section corresponds to a panelbox. Only the valid parameters of each section can be edited in this area (that is, only the parameters belonging to the master machine version can be defined).

Initially, when you create an Oracle Tuxedo domain, the Domain UBBCONFIG editor provides a limited set of parameters in the *RESOURCES section that do not vary among different Tuxedo versions.

The **More ...** in the *RESOURCES section and **Add ...** in other sections are disabled. These parameters are displayed until the master machine is added to the domain. Once the master machine is added to the domain, according to the master machine Tuxedo version, all the invalid parameters for each section are filtered out and turned to invisible.

On first entering of the domain editor, the indicator `EXT_MON` of `OPTIONS` parameter is set by default for metrics collection purposes.

*RESOURCES Section

This panel provides a `UBB_Resource` template that you can choose to generate the UBBCONFIG file *RESOURCES section when creating a domain. The items in the template are listed in [Table 2-5](#).

Table 2-5 UBB_Resource Template Items

Item	Value
IPCKEY	33333
MASTER	NULL When you specify the master machine, this item is generated automatically.
MODE	You can specify SHM or MP.

All items can be modified except for `MASTER`. The `MASTER` item is generated as a Locked Variable when you specify the master and backup machines. Besides these parameters, you can add other parameters as well.

You can also create your own `UBB_Resource` templates and save them in the Software Library. In all templates, `MASTER` is filled in by the system automatically following above rule.

*MACHINES Section

This panel provides an `UBB_Machine` template that you can choose to generate the `UBBCONFIG` file `*MACHINES` section when creating a domain and specifying machine. The items in the template are listed in [Table 2-6](#).

Table 2-6 UBB_Machine Template Items

Item	Value
ADDRESS	This item is generated by the system according to the machine logical name. It cannot be modified by users.
LMID	This item is generated automatically by the system. The naming rule is <code>SITE1</code> , <code>SITE2</code> , and so on. It cannot be modified by users.
APPDIR	This item is generated by the system according to the machine <code>APPDIR</code> and the domain information when you add machines to one domain.
TUXCONFIG	This item is in <code>\$APPDIR/tuxconfig</code> and cannot be changed by users.
TUXDIR	This item is generated automatically by the system using the <code>TUXDIR</code> specified by the machine list. It cannot be modified by users.

You can add other `UBBCONFIG` file `*MACHINES` section parameters to the as well. You can also specify the `TLOG` to raw disk. The system deletes it when undeploying this domain.

You can create your own `UBB_Machine` templates and save them in the Software Library. The parameters replacement follows the rule described in [Table 2-6](#).

*GROUPS Section

When you add an application package to a domain, the system replaces some `*GROUPS` section parameters in `Properties.xml` as shown in [Table 2-7](#).

Table 2-7 *GROUPS Section Parameters

Item	Value
GROUPNAME	This item is generated automatically by the system. The naming rule is GROUP1, GROUP2, and so on. It cannot be modified by users.
LMID	This item is generated by the system according to the machine to which the package is deployed.
GRPNO	This item is generated automatically by the system. It cannot be modified by users.

For more information, see [Table 2-3](#). You can add other parameters to the `UBBCONFIG` file as well.

*RMS Section

When you add an application package to one domain, the system replaces some *RMS section parameters `Properties.xml` as shown in [Table 2-8](#).

Table 2-8 *RMS Section Parameters

Item	Value
RMSNAME	This item is generated automatically by the system. The naming rule is RMS1, RMS2, and so on. It cannot be modified by users.
SRVGRP	This item will be replaced with the new group name generated by the system before. It cannot be modified by users.
RMID	This item is generated automatically by the system. The naming rule is 1, 2, and so on. It cannot be modified by users.

For more information, see [Table 2-3](#). You can add other parameters to the `UBBCONFIG` file as well.

*NETGROUPS Section

All the parameters in this section should be entered manually by users if needed.

*NETWORK Section

This section is also included in the `UBB_Machine` template. If the domain is in MP mode, the system automatically adds this section to the `UBBCONFIG` file as shown in [Table 2-9](#).

Table 2-9 *NETWORK Section Parameters

Item	Value
LMID	This is generated automatically by the system. You cannot modify it.
NADDR	The system will generate <code>//hostname:port_number</code> , from which you can modify the <code>port_number</code> .
NLSADDR	The system will generate <code>//hostname:port_number</code> , from which you can modify the <code>port_number</code> .

You can add other parameters to the `UBBCONFIG` file as well.

*SERVERS Section

When adding an application package to a domain, system replaces some `*SERVERS` section parameters in `Properties.xml` as shown in [Table 2-10](#).

Table 2-10 *SERVERS Section Parameters

Item	Value
SVRGRP	This item is replaced with the new group name generated by the system before. It cannot be modified by users.
SRVID	This item is generated automatically by the system. The naming rule is 1, 2, and so on. It cannot be modified by users.

For more information, see [Table 2-3](#). You can add other parameters to the `UBBCONFIG` file as well.

*SERVICES Section

When adding an application package to a domain, system replaces some `*SERVICES` section parameters in `Properties.xml` as shown in [Table 2-11](#).

Table 2-11 *SERVICES Section Parameters

Item	Value
SRVGRP	If <code>Properties.xml</code> contains this parameter, the system replaces it with the corresponding group generated before. Although this parameter is not mandatory, it is recommended that you add it. When there are same services in different application packages, they can have different operations.
ROUTING	If <code>Properties.xml</code> contains this parameter, system replaces it with the corresponding routing name generate by the system.

For more information, see [Table 2-3](#). You can add other parameters to the `UBBCONFIG` file as well.

*ROUTING Section

When adding an application package to a domain, system replaces some *ROUTING section parameters in `Properties.xml`.

For other `Properties.xml` file *ROUTING section parameters, the system keeps their values and

Table 2-12 *ROUTING Section Parameters

Item	Value
ROUTING_CRITERIA_NAME	This item is generated automatically by the system. The naming rule is <code>ROUTING1</code> , <code>ROUTING2</code> , and so on. It cannot be modified by users.
RANGES	The group of this parameter will be replaced by the system.

allows you to modify manually. You can add other parameters to the `UBBCONFIGfile` as well.

Deploying Application Packages to Tuxedo Domain

Typically, you need to perform the following procedures to deploy application packages to a Tuxedo domain:

- [Preparing Application Packages](#)
- [Discovering Tuxedo tlisten and Tuxedo Home](#)

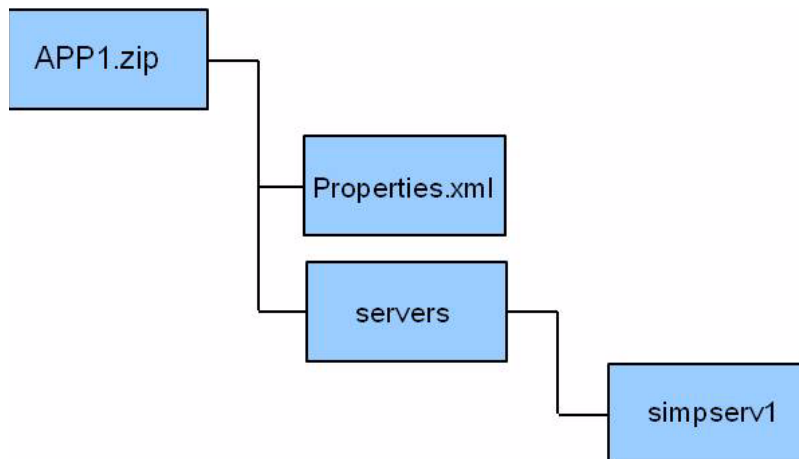
- [Preparing a Tuxedo Domain or Loading an Existing Domain](#)
- [Deploying Application Packages to Domain](#)

Preparing Application Packages

Each machine needs one application package. Refer to the descriptions in [Application Package Organization and Content](#) to prepare two application packages: APP1.zip and APP2.zip.

APP1.zip content structure is shown in [Figure 2-8](#):

Figure 2-8 APP1.zip Content Structure



APP1.zip Properties.xml is shown in [Listing 2-3](#):

Listing 2-3 APP1.zip Properties.xml

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<ApplicationProperties>
  <PackageName>APP1.zip</PackageName>
  <TuxedoVersion>12.1.3.0.0</TuxedoVersion>
  <SupportedOS>Linux</SupportedOS>
  <TuxedoWordSize>64</TuxedoWordSize>
```



```

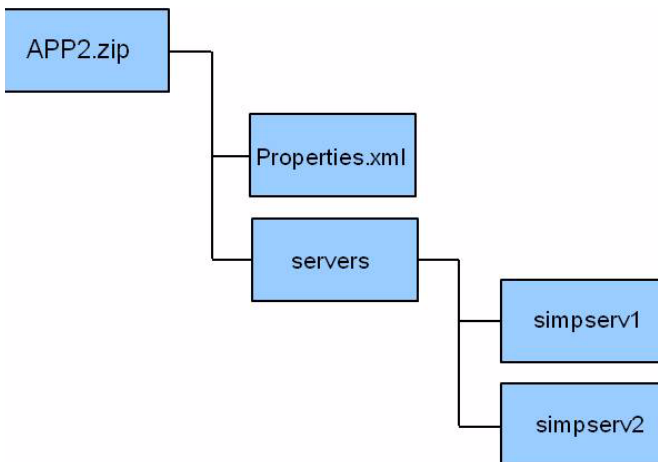
<MachineArch>x86_64</MachineArch>

<GroupSection GROUPNAME="GROUP1" GRPNO="29999">
  <MRM>N</MRM>
  <ServerSections>
    <ServerSection AOUT="servers/simpserv1" SRVID="30000">
      <CONV>N</CONV>
    </ServerSection>
  </ServerSections>
</GroupSection>
</ApplicationProperties>

```

APP2.zip content structure is shown in [Figure 2-9](#):

Figure 2-9 APP2.zip Content Structure



APP2.zip Properties.xml is shown in [Listing 2-4](#):

Listing 2-4 APP2.zip Properties.xml

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<ApplicationProperties>
  <PackageName>APP2.zip</PackageName>
  <TuxedoVersion>12.1.3.0.0</TuxedoVersion>
  <SupportedOS>Linux</SupportedOS>
  <TuxedoWordSize>64</TuxedoWordSize>
  <MachineArch>x86_64</MachineArch>

  <GroupSection GROUPNAME="GROUP1" GRPNO="29999">
    <ServerSections>
      <ServerSection AOUT="servers/simpserv2" SRVID="20000" />
      <ServerSection AOUT="servers/simpserv3" SRVID="20010" />
    </ServerSections>
  </GroupSection>
</ApplicationProperties>
```

After the application packages are prepared, refer to [Uploading a Package](#) to upload them to EM through Manage Application Package on the Tuxedo Summary page.

Discovering Tuxedo tlisten and Tuxedo Home

Before a host can be used for deploying Tuxedo domains, Tuxedo `tlisten` and Tuxedo home targets must be discovered by EM agent on the host.

To discover Tuxedo `tlisten` and Tuxedo home targets, do the following steps:

1. From the Oracle Tuxedo Summary page, click **Add > Oracle Tuxedo Domain Discovery**.
2. Fill in the `tlisten` host name and RMI port. Select a EM agent you want to use for discovery.
3. Uncheck **Find Oracle Tuxedo Domains**.

4. Click **Discover Now**.

Note: The two targets must be discovered by the preceding discovery process. Adding the two targets as standalone target separately does not work for Resource Broker.

Preparing a Tuxedo Domain

To prepare a Tuxedo Domain to be deployed, you can edit an existing domain discovered by Enterprise Manager or create a new domain. You can create two types of Tuxedo domains: SHM domain and MP domain.

Creating an SHM Domain

To create an **SHM** domain, do the following:

1. Refer to [Domain UBBCONFIG Editor](#) to enter the domain editor page.
2. Fill out the *RESOURCES section. Note the following:
 - **IPCKEY:** This parameter is initialized to 33333 automatically for each new created domain.
 - **DOMAINID:** This parameter takes the last specified domain name by default.
 - **MASTER:** You can leave it blank since the value is initialized automatically when you add the first machine to the domain.
 - **MODEL:** Keep the default value, **MP**, to make dynamic machine deployment possible even if you are creating an SHM domain.

Click **More...** to show more options.

3. From the Machine List panel, add a machine to domain by clicking + beside the machine name.

A popup dialog is displayed to collect machine related information.

If it is the first time the machine is being added, the **Application Home** option must be specified. The Application Home is the root directory where all domains are deployed.

The “Environment Path Variable” can also be specified when adding a machine. It is a colon (UNIX/Linux) or semi-colon (Windows) separated string that contains paths where the environment specific commands are searched for. For example, the compiler or Oracle DB client that could be used by user programs.

4. Click **OK**.

Another popup dialog which contains all `UBBCONFIG` file `*MACHINES` section parameters is displayed.

5. Fill in any parameter specific to this Tuxedo machine. Click **OK**.

After a domain is created, the following changes appear on the page:

- All the required information for this machine is collected and one `UBBCONFIG` file entry is added to the `*MACHINES` section on the left side.
- Several `UBBCONFIG` file entries are generated automatically in other sections as well.
- An LMS server (used for monitoring data collection), is added to the `*SERVERS` section. It is a mandatory server to enable the machine to be monitored by EM.
- A group named `SYSGRP_<LMID>` is added to include LMS server.
- The `*NETWORK` section entry is also added automatically.

Creating an MP domain

The procedures of creating an MP domain are similar to creating an SHM domain. An additional task is required; adding another available machine for deployment.

To deploy EM agent to a new host, do the following:

1. From the Enterprise Manager home page, click **Setup->Add Target->Add Targets Manually**.
2. In the Add Targets Manually page, select **Add Host Targets**. Click **Add Host ...**
3. Enter the information required to deploy a new agent step-by-step to finish the deployment.

For more information, click **Help** in the upper right of the page.

After the deployment is successfully completed, you can find two new targets in the All Targets page. This is a prerequisite for discovery on the new host.

To make the newly-added machine available for Tuxedo domain deployment, the `tlisten` process must be started and discovered on the 'Tuxedo Domain Discovery' page by the EM agent. Refer to [Discovering Tuxedo tlisten and Tuxedo Home](#) for instructions. Once the `tlisten` and Tuxedo home targets are discovered, verify the status on the [Domain UBBCONFIG Editor](#) page; the newly-added machine appears in the machine list.

Edit a Discovered Domain

To edit a discovered domain, do the following:

1. From the Oracle Tuxedo Summary page, select the domain you want to edit.
2. Click **Edit**. The domain is loaded in [Domain UBBCONFIG Editor](#).
3. Refer to preceding sections to edit the domain.

Note:

- Only when tlistens running on all machines deployed in a domain are discovered by their local Enterprise Manager agents, the domain can be edited.
- Some files will be overwritten after deployment of the existing domain, such as `setenv.sh` `java.env`.
- Full deployment with security is not supported for the existing domain.
- For the ART CICS domain configured with MP mode, the `KIXDIR` and `COB installation DIR` must be set the same for machines.

Loading an Existing Domain

Resource Broker works for existing domains either. Before an existing domain can be deployed with application packages, you need to make sure the following:

- The domain is running and has been discovered on Enterprise Manager.
- All machines in the domain have Enterprise Manager agent installed and running.
- The domain is discovered using the Enterprise Manager agent of master machine, while the `tlisten` and `tuxedo home` targets on other slave machines are discovered using the corresponding Enterprise Manager agent of each machine.

Once you confirm the above prerequisites are met, load the existing domain parameters to Domain Editor by the following steps:

1. Refer to [Viewing Tuxedo Summary](#) to enter the Tuxedo Summary page.
2. Highlight the domain you want to deploy, click **Edit**.

The UBBCONFIG and DMCONFIG parameters are retrieved from Tuxedo domain backend, and then are parsed and filled in the corresponding section in the Domain Editor. Exceptionally, the default value of `IPCKEY`, `DOMAINID`, and `MASTER` in the `RESOURCES` section of UBBCONFIG editor cannot be changed.

On the right side, all the machines available for adding to the domain are listed in the Machine List panel. If a machine belongs to the existing domain, the '+' button beside its name is disabled.

The Package List panel box contains all the Tuxedo application packages. If a package is available to be added to the machines in the domain, its '+' button is enabled.

ART Batch and ART CICS configuration is not supported on the existing domain. If the original domain has ART Batch or ART CICS system configured, Resource Broker keeps the ART related resources untouched during a full deployment or a dynamic deployment. In that event, you need to provide correct environment variables so as to booting up the system.

Controls

A few controls in the Controls panel box have special behaviors than the ones of a Resource Broker created domain.

- Save

At the first time saving of the loaded existing Tuxedo domain, the application home targets are created automatically. The application home path is determined by the corresponding APPDIR on each machine. It is the parent directory of APPDIR.

A window for collecting the credential information of each machine in the Tuxedo domain is displayed after the validation phase. The credential information is used by the subsequent deployment actions, especially for the dynamic deployment.

Once the domain is saved, it is turned to a Resource Broker managed domain, which means the domain configuration status will be maintained on OMS side and all the changes made directly to the Tuxedo domain backend will not be detected by Resource Broker anymore.

- Deploy

During a full deployment, the APPDIR path is kept unchanged. This requires that the provided credential must have write permission to the application home (the parent directory of APPDIR), otherwise the deployment will fail.

In boot phase, you need to customize the boot script to provide any required environment variables.

Deploying Application Packages to Domain

To add an application package to a domain, do the following steps:

1. Click + beside the package entry you want to add to the Package List.

The Add Application Package popup dialog appears. In the dialog, the Machines drop-down list shows all machines added to this domain and are applicable for this package.

Note: The + button is inactive if the package is not suitable for the machine.

2. Check the machine entry tooltip information and package entry respectively to make sure the OS type, machine architecture, and Tuxedo version all match.
3. Click **OK**.

All entries described in `Properties.xml` of the package are added to the corresponding UBBCONFIG file sections as shown in [Figure 2-10](#)

The (*) signs before the `GROUPS` entry and `SERVERS` entry indicate new-added items to this domain that have not been deployed.

Figure 2-10 New Added Entries

The screenshot displays a configuration window with several sections. The 'GROUPS Section' is expanded, showing two entries. The first entry is for 'SYSGRP_SITE1' with LMID 'SITE1' and GRPNO '29001'. The second entry, marked with a star, is for 'GROUP1' with LMID 'SITE1' and GRPNO '10'. Below this are sections for 'RMS Section', 'NETGROUPS Section', and 'NETWORK Section'. The 'SERVERS Section' is also expanded, showing two entries. The first entry is for 'LMS' with SRVGRP 'SYSGRP_SITE1' and SRVID '10'. The second entry, marked with a star, is for 'APP1/servers/simpserv1' with SRVGRP 'GROUP1' and SRVID '30000'.

Section	Package/Asset	GROUPNAME	LMID	GRPNO	SRVID	SRVGRP	AOUT	Buttons
GROUPS Section		SYSGRP_SITE1	SITE1	29001				Edit, Duplicate, Remove
	(APP1.zip)	GROUP1	SITE1	10				Edit, Remove
RMS Section								
NETGROUPS Section								
NETWORK Section								
SERVERS Section					10	SYSGRP_SITE1	LMS	Edit, Duplicate, Remove
	(APP1.zip)				30000	GROUP1	APP1/servers/simpserv1	Edit

The `GROUPNAME` and `GRPNO` parameters are generated automatically when adding the package. All groups and their assets defined in `Properties.xml` are added to the domain. Unlike machine entries, one application package can be added more than one time.

The Tuxedo domain is ready to be deployed and monitored.

Deploying the Domain to Remote Machine

After completing all necessary configuration for the Tuxedo domain, do the following steps to deploy the domain to a remote machine:

1. Click **Save&Deploy** in the Controls panel to deploy and boot the domain.

The new Tuxedo Domain target is created (in the following format) <DomainId>:<PMID of master machine>:<domain IPCKEY>.

Note: During every deployment process, all machine user credentials included in this domain are required. If the user credentials are inputted once, the retained ones are displayed. You must be a valid OS user on the target machine and have write privileges to the Application Home directory. The user must be in the same user group as the one who installed the Management Agent.

2. From the checkbox that indicates whether or not to boot the deployed domain, you can choose to customize the config and boot scripts for special requirements (for example, create TLOG, QUE, or ENVFILE).

Figure 2-11 shows the domain deployment options screen.

Figure 2-11 Domain Deployment Options

Domain Information

Host Credentials
Provide credential information for each machine:

Machine Name	Username	Password
bej301163.cn.oracle.com	cguo	••••••

Domain Deployment Options
Boot Oracle Tuxedo domain after deployment

You can customize the config and boot scripts.:

Domain Boot Script:

```

./setenv.sh
tmboot -y
if [ $? -ne 0 ]; then
    echo "Error booting Tuxedo domain."
    exit 1
fi

```

Domain Config Script:

Machine Name	Config Script
bej301163.cn.oracle.com	<pre> #!/bin/ksh set -e ./setenv.sh rm -f tuxconfig rm -f tuxconfig tmloadcf -y UBBCONFIG if [\$? -ne 0]; then echo "Error loading UBBCONFIG" exit 1 fi </pre>

Note: Because certain information may not be collected from an existing domain, when deploying new machines or special servers to the existing domain, you need to manually add extra script before deployment.

3. Enter the usernames and passwords, and click **OK** on the upper right corner to proceed to the deployment tracking page.
4. Since the deployment is an asynchronous procedure, you need to manually refresh in the page periodically, or set the auto refresh rate at the upper right corner of the page.

When all the steps are successful, the domain has been deployed to the Application Home on the target machine, and booted. The last step of deployment is **CheckDeployStatus**, where the deployment status is checked and corresponding entries status in the domain editor is changed accordingly. If the deployment is successful, an automatic domain discovery is performed. The new discovered targets belonging to the domain can then be found in the All Targets and Tuxedo Summary page.

Deploying an Application Package Dynamically

To deploy an application package dynamically, do the following steps:

1. Select the SHM domain target created before, for instance, in the Tuxedo Summary page and then click **Edit**.

The domain content is reloaded into the [Domain UBBCONFIG Editor](#) page.

2. Click + beside package APP2.zip to add it to the domain.

The two servers `simpser2` and `simpser3` in the group described in `APP2.zip` `Properties.xml` are added to the `*SERVERS` section.

3. Click **Save & Deploy** to activate the newly-added group and servers.

Note:

- The domain must be in **Up** status when deploying the application package.
- After deploying, only the newly-added packages are delivered to the existing domain `APPDIR`, and the included new group/servers are activated by Tuxedo MIB calls.
- This feature is not supported for the existing ART CICS domain.

Once the deployment is complete, the new directory, `APP2`, can be found in domain `APPDIR` and the new targets corresponding to `GROUP2` and `simpser2/simpser3` are created.

Updating a Deployed Application Package

Sometimes you may need to update the application packages that have been uploaded. Currently, you can manually update it following these steps:

1. Remove the old package and then upload the updated one in the Manage Application Package dialog box in Tuxedo Summary page. See [Uploading a Package](#) and [Removing a Package](#).
2. Shut down the Tuxedo domain which needs to be updated.

3. After the domain target status changes to down status, right click the domain target in the Tuxedo Summary page, and then select **Control->Deploy** to perform a full deployment of the domain.

Removing Package-Associated Groups

Once the package associated groups are added to a machine, the 'X' button beside the package name is available to remove the package from the machine.

To remove the package associated groups, do the following steps:

1. Click the **X** button beside the package you want to remove.
2. In the window that appears, choose the target groups in the package removing popup dialog. According to the selected groups, do one of the following:
 - For package-associated groups that have not been deployed and activated on a Tuxedo machine (indicated by *), click **OK** to remove the selected groups/members.
 - For deployed and activated package-associated groups, a **Remove from management server only** checkbox is displayed.

If you leave it unchecked and click **OK**, an instant undeployment action is triggered completely removing the selected groups from Tuxedo application runtime. If you check it and click **OK**, the selected groups are removed from the UI only, not impacting underlying Tuxedo application.

Once such action is performed, no further dynamic deployment can be performed until full deployment is carried out for the domain.

Dynamic Resource Broker

The Dynamic Resource Broker enables Tuxedo server manual and policy-driven deployment, groups, and machines on a running Oracle Tuxedo domain for flexible resource allocation of a certain Enterprise Manager target.

There are three levels of dynamic deployment:

- Package-level dynamic deployment and undeployment
- Machine-level dynamic deployment and undeployment
- Policy-driven server spawn and decay

Manual Dynamic Deployment

Both Tuxedo application packages and Tuxedo machines can be added to an active domain manually.

To deploy a package manually, click **Save & Deploy** on the **Controls** panel when there are new added packages to the domain (identified by an asterisk '*').

Once the **Save & Deploy** button is clicked, the current domain editing status is saved and a full validation is started on the domain `UBBCONFIG`. If the validation fails, the deployment process is stopped and an error message appears.

Note: The `UBBCONFIG` entries added manually by clicking **Add...** cannot be deployed and activated dynamically.

Dynamic deployment only works on `UBBCONFIG` entries imported from application packages. To make the manually added `UBBCONFIG` entries active, a full deployment of the domain must be performed.

To deploy a machine manually, do the following steps:

1. Click the plus '+' in the **Machine List** panel to add a machine.

The newly-added machine is identified with asterisk (*) at the beginning of each imported `UBBCONFIG` entry.

2. Configure the new machine in the `UBBCONFIG` section editor, if needed.
3. Click **Save & Deploy** in the **Controls** panel.

The deployment takes place immediately using the collected machine credentials and adds the new machine to the domain as a slave machine.

Note: The machine-level dynamic deployment supports domains in MP mode only.

Policy-Driven Dynamic Deployment

The following three levels of dynamic deployment are driven by user-defined policies:

- Package-level

Dynamic deployment or undeployment of one or more application packages on an active Tuxedo machine is triggered under certain conditions (for example, CPU load is low and does not exceed 60%). The final machine where the packages are deployed is selected among the candidate machines according to the predefined filtering policy.

Before you create a package-level policy, packages should be added from the Package List as candidates (“**As Candidate**” checkbox is checked). You can modify the imported `UBBCONFIG` entries detailed configuration by clicking **Edit** for each entry and then create deployment policies for the candidate packages in the Policy Management console.

Note: Before dynamic deployment of a package, any newly introduced TMS server must be put under the directory referred by the `PATH` environment variable on Tuxedo boot-up, otherwise, the TMS server will not be able to be found by Tuxedo application runtime, even if it is located in the `bin` sub-directory of the package.

- Machine-level

Dynamic deployment or undeployment of machines is triggered when some conditions (incident rules), defined by policy are met.

You can define policies on the new added machine entries to convert them to candidate machines. The final machine to be deployed is selected among the candidate machines according to the predefined filtering policy.

Machine-level dynamic deployment is often used together with package-level deployment. When a machine is dynamically deployed, the packages added to the machine are deployed at the same time.

- Server-level

The number of deployed running Tuxedo servers can be increased or decreased automatically when some conditions (incident rules), defined by policy are met. The minimum and maximum server number depends on the `MIN` and `MAX` parameter set in the `UBBCONFIG`.

For more information, see [Policies](#).

Policies

The Resource Broker deployment policy (which works in association with Enterprise Manager incident rules), defines under what conditions the Resource Broker deployment policy evaluation is triggered.

Creating an Incident Rule

An incident rule specifies condition details that trigger a resource broker policy. The condition can be complex and made up of criteria combination that represents the event characters (for example, the event type, severity, category, and etc.). By defining incident rules, you can specify when one of the following conditions is met, the deployment policies are evaluated:

- The CPU utilization of a host reaches 85%
- The average execution time of certain Tuxedo service exceeds 500 microseconds
- Some Tuxedo machines are down, etc.

An incident rule can contain any metrics provided by any monitored target. To allow the incident rule to be evaluated on specific targets, create the Enterprise Manager Group on the targets and refer to this group in the rule set.

The following steps are given as an example to introduce how to define a CPU metric even type. Before creating a policy, you need to create incident rules.

1. From the Enterprise Manager console home page, click **Setup-> Incidents -> Incident Rules**.
2. On the Incident Rules page, click **Create Rule Set...**
3. Enter a rule name, and choose **Specific targets** on the Targets tab at the bottom of the page.
4. Add your predefined Enterprise Manager Group(s) which contain(s) the target events under consideration.
Note: Selecting specific groups in this step is important, otherwise all the events of same type on any target implements the incident rule.
5. Choose the **Rules** tab. Click **Create** to create a incident rule.
6. Click **Continue** in the pop-up window to define an event-type incident rule.
7. In the page that appears, select **Type**. Choose **Metric Alert** in the list.
8. Choose **Specific events of type Metric Alert**.
A Metric Alert panel appears.
9. Click **Add** to designate a specific Metric Alert event.
10. In the Select Specific Metric Alert page, define an incident rule carried out when the CPU utilization of one host reaches the Warning threshold, as shown in [Figure 2-12](#):

Figure 2-12 Define a CPU Metric Event- Example

The screenshot shows the 'Select Specific Metric Alert' configuration page. The search bar is set to Target Type: Host and Metric Name: CPU Utilization. The table below shows the selected metric group and its details.

Select	Metric Group	Metric	Objects	Excluded Obj
<input checked="" type="checkbox"/>	Load	CPU Utilization (%)	Not Applicable	Not Applicable
<input type="checkbox"/>	CPU Usage	CPU Utilization (%) of a cpu	<input checked="" type="radio"/> All objects (CPU Number) <input type="radio"/> Select []	
<input type="checkbox"/>	Top Processes	CPU Utilization for Top Process	<input checked="" type="radio"/> All objects (Process ID) <input type="radio"/> Select []	
<input type="checkbox"/>	Program Resource L	Program's Max CPU Utilization	<input checked="" type="radio"/> All objects (Program Name;Owner) <input type="radio"/> Select []	
<input type="checkbox"/>	Program Resource L	Program's Total CPU Utilization	<input checked="" type="radio"/> All objects (Program Name;Owner) <input type="radio"/> Select []	

The 'Severity' dropdown is set to 'Warning'.

11. Click **OK**.

The newly-created event appears in the Metric Alert panel.

12. Click **Next** to proceed to the Add Actions step page.

13. In the Advanced Notifications panel, check the predefined notification named **Tuxedo Event Connector (PL/SQL Procedure)**.

This means when the incident is produced, Tuxedo Event Connection is notified.

14. Click **Continue** in the upper right corner to complete the definition. Click **Next**.

15. Enter a name in the Specify Name and Description page. Click **Next**, and then **Continue** to finish the configuration.

16. Return to the Edit Rule Set page. Click **Save** to activate the defined incident rule.

On the All Enterprise Rules page, you can find the incident rule is added under the Tuxedo Incident Rules rule set with “active” status.

Creating a Policy

The policies are defined and managed centrally in the **Policy Management** console. To enter the **Policy Management** console, click **Policies** in the domain editor Controls panel.

Once the incident rule is defined, you can create a policy by clicking **Add** in the **Policy Management** console and specifying the options in the **Policy Properties** page. For more information, see [Policy Management](#). or click **Help** in the upper right of the page.

Controls on Tuxedo Domain Target

The Tuxedo Summary page provides a set of controls for the Tuxedo Domain target created by the Domain Editor. Right-clicking the domain target you can find actions specific to Resource Broker are included in the Control menu item.

Deploy

This action is exactly the same as the one on the Domain Editor page. When the domain is down or has not been deployed to the target machines, a full deployment is carried out. When the domain is in active status, the newly-added packages are deployed and activated dynamically.

Config

A domain should be configured so it can be booted up after it is deployed. This action can only be applied to a domain which is **not** in active status.

The default config action is to do `tmloadcf` of the `UBBCONFIG` on the master machine. If there are any specific configurations required before the domain boots up (for example, creating TLOG device, or QUEUE space), you can customize the script.

Start Up

This action can be applied to a domain which is **not** in active status. The default Start Up action is to execute `tmboot -y` on the master machine. You can also customize the boot script.

Shut Down

This action can only be applied to a domain which is in **active** status. The default action of Shut Down is to execute `tmshutdown -y` on the master machine. You can also customize the shutdown script.

Undeploy

This action can only be applied to a domain which is **not** in active status. The default Undeploy action is to remove the entire `APPDIR` on target machines. You can customize the `unconfig` script to do some cleanup before `APPDIR` is removed.

Using Resource Broker for ART Batch

To use resource broker for ART Batch in a Tuxedo domain, you need to configure both server and JES parameters. For existing domains, these information are introspected from runtime environment. All the configurations can be adjusted again just like the domain is created by resource broker.

Server Configuration

Several Tuxedo system servers specific to ART Batch are required to be configured in the Tuxedo domain to make ART Batch work properly. For example, the servers `ARTJESADM`, `ARTJESCONV`, `ARTJESINITIATOR`, and `ARTJESPURGE` are necessary for a functional ART JES system.

Since ART Batch uses the Oracle Tuxedo Event component, an Oracle Tuxedo user event server, `TMUSREVT`, is required in the `UBBCONFIG` file. Additionally, a `TMQUEUE` server is also mandatory to store the JES job information.

There are three ways for adding these Tuxedo system servers.

Adding Tuxedo System Servers Manually

You can add the servers manually by clicking **Add** in the Servers section of [Domain UBBCONFIG Editor](#). Make sure you provide correct parameters (`CLOPT`), to the servers. Once you save configurations in the domain, the Domain Editor performs `CLOPT` validation on the system servers.

Adding Tuxedo System Servers Using Application Package

System server-related information can be described in `Properties.xml`, and then packed into an application package. By adding the application packages to a Tuxedo domain, the corresponding servers are added accordingly.

This is the only way you can use policy-driven dynamic resource broker features.

The ART Batch specific system servers are:

- `ARTJESADM`

- ARTJESCONV
- ARTJESINITIATOR
- ARTJESPURGE

There should be no corresponding binaries for the above servers packed in the application packages. These system server binaries are included in the ART Batch installation directory.

The system servers are referred to in `Properties.xml` using a single server name.

Adding Tuxedo System Servers Using Server Template

You can add ART Batch related servers using a server template. Do the following steps:

1. From the [Domain UBBCONFIG Editor](#), click **JESConfig**.
2. Select one or more machines from the Machines list. Click **Apply Server Template**.
Following servers are added to the machine:

- JESQUEGRP
- TMQUEUE
- ARTJESGRP
- TMUSREVT
- ARTJESADM
- ARTJESCONV
- ARTJESINITIATOR
- ARTJESPURGE

Since the `TMS_QM` used by server `TMQUEUE` requires a `TLOG` device, after applying the server template, `TLOGDEVICE` and `TLOGSIZE` parameters of the corresponding machine are configured to `$APPDIR/TLOG` if no path is specified.

JES Configuration

- You can configure JES parameters by clicking **JESConfig** from the [Domain UBBCONFIG Editor](#). JES Configuration consists of two parts:
 - [JES Configuration File](#)
 - [Misc Configuration](#)

JES Configuration File

A JES configuration file is used by the TuxJES administration server `ARTJESADM`. You can configure the following JES configuration file parameters:

- JES Base Directory

The base JESROOT directory. If it is not set, the base directory is `$APPDIR`.

- JESROOT

The root repository to store job information. This directory is created under the path set in **JES Base Directory**. For example, if `JESROOT` is set to `jesroot` and **JES Base Directory** is set to `$APPDIR`, the full path of the `JESROOT` directory is `{ $APPDIR } / jesroot`.

Note: If ART Batch related servers are configured on two Tuxedo machines, they must share the same `JESROOT` repository; only an absolute `JESROOT` path is acceptable and the directory should exist in the Network File System.

- DEFAULTJOBCLASS

Optional. The default job class if the job class has not been set in `JCL`. The default job class is `A` if this option is not set.

- DEFAULTJOBPRIORITY

Optional. The default job priority if the job priority has not been set in `JCL`. The default job priority is `0` if this option is not set.

- DUPL_JOB

If `NODELAY` is not checked, only one job can be in execution status per job name. When it is checked, `NODELAY` removes the dependency check.

- EVENTPOST

Specifies events posted for a job at particular stages.

- S: Job submission event.
- C: Job conversion complete event.
- E: Job execution complete event.
- P: Job purge event.
- A: All supported events

If `EVENTPOST` is not specified, no event is posted. The data buffer with event post is `FML32` type. The fields are defined in `tuxjes/include/jesflds.h`.

- JOBREPOSITORY

The path of the stored job repository. The job submitting script file path may be a relative path in `JOBREPOSITORY` if it is set.

- **PRIVILEGE MODE**

Specifies whether and how to enable the user substitution (See [TuxJES User Substitution](#)).

- **NONE**: Default value. Indicates jobs are executed by the OS user who starts the JES system. This is compatible with all previous JES system implementations.
- **USER_IDENTICAL**: Indicates jobs are executed by the Oracle Tuxedo user where the JES client joins JES system. Make sure that each Oracle Tuxedo user corresponds to an existing OS user before you choose this option.
- **USER_MAPPING**: The JES system looks up the TuxJES user mapping file and finds the OS user corresponding to the Oracle Tuxedo user where JES client joins JES system, and then appoints this OS user as the job executor.

- **USER MAPPING FILE**

This option is enabled when you choose **USER_MAPPING** as Privilege Mode. Specify the user mapping file name. User mapping file is under the `$APPDIR` directory by default.

The owner of the User Mapping File created by Resource Broker is root, and its file permission is "`-rw-----`".

Note: The User Mapping File name cannot be:

- one of the following: `Logs`, `QUE`, `TLOG`, `UBBCONFIG`, `acc`, `crlog`, `data`, `data_source`, `jesconfig`, `jesqinit`, `tools`, `tpacl`, `tpgrp`, `tpusr`, and `tuxconfig`.
- the same as `JESROOT` if `JESROOT` is under the `$APPDIR` directory.

For the existing domain, since security is not supported, the user mapping file cannot be introspected and parsed.

- **USER MAPPING Table**

The user mapping table displays the mapping relationship between Oracle Tuxedo users and OS users. When Privilege Mode is set to **USER_MAPPING**, the user mapping table is enabled allowing you to add, edit, or delete a user mapping entry. Every line in the mapping table is in the following format:

```
tuxedousername OSusername
```

Note:

- Duplicated user pairs are not allowed.

- One Oracle Tuxedo user can only be mapped to one OS user.

Permission Settings When Enabling User Substitution

When you set the privilege mode to either **USER_IDENTICAL** or `USER_MAPPING`, the following permission settings are made automatically by Resource Broker:

- `UBBCONFIG` permission settings
 - The `PERM` value in the `RESOURCES` section is set to `0666`. The `RQPERM` and `RPPERM` value in all JES servers entries in the `*SERVERS` section is set to `0666`. Therefore, all job users have full permissions to JES system IPC resources.
 - The `UID` and `GID` parameters of the machine where JES servers are located is set to root user `UID` and `GID`.

[Listing 2-5](#) shows a `UBBCONFIG` file example.

Listing 2-5 UBBCONFIG Example

```
*RESOURCES
IPCKEY      <IPCKEY> # for example 132770
DOMAINIDjessample
MASTER     SITE1
MODEL      SHM
MAXACCESSERS  200
MAXSERVERS  50
NOTIFY     SIGNAL
PERM       0666 #Adding "PERM=0666" in RESOURCES section
SECURITY   USER_AUTH
AUTHSVC    "AUTHSVC"

*MACHINES
#
<uname -n>
```

Using the Enterprise Manager for Oracle Tuxedo

```
LMID = SITE1
TUXDIR = "<full path of TUXEDO software>"
TUXCONFIG = "<full path of APPDIR>/tuxconfig"
TLOGDEVICE = "<full path of APPDIR>/TLOG"
TLOGSIZE=10
APPDIR = "<full path of APPDIR>"
ULOGPFX = "<full path of APPDIR>/ULOG"

*GROUPS

ARTGRP
    LMID = SITE1  GRPNO = 1

QUEGRP
    LMID = SITE1  GRPNO = 2
    TMSNAME = TMS_QM TMSCOUNT = 2
    OPENINFO = "TUXEDO/QM:<full path of APPDIR>/QUE:JES2QSPACE"

EVTGRP
    LMID= SITE1 GRPNO=3

#
*SERVERS
# Adding RQPERM=0666 RPPERM=0666 in all JES servers entry in SERVERS section
DEFAULT:CLOPT="-A"

TMUSREVT          SRVGRP=EVTGRP SRVID=1 CLOPT="-A"
                  RQPERM=0666 RPPERM=0666
```

TMQUEUE

```

SRVGRP = QUEGRP  SRVID = 1
GRACE = 0  RESTART = Y  CONV = N  MAXGEN=10
CLOPT = "-s JES2QSPACE:TMQUEUE -- -t 5 "
RQPERM=0666  RPPERM=0666

```

```

ARTJESADM          SRVGRP =ARTGRP  SRVID = 1  MIN=1  MAX=1
CLOPT = "-A -- -i jesconfig"
RQPERM=0666  RPPERM=0666

```

```

ARTJESCONV        SRVGRP =ARTGRP  SRVID = 20  MIN=1  MAX=1
CLOPT = "-A -- "
RQPERM=0666  RPPERM=0666

```

```

ARTJESINITIATOR   SRVGRP =ARTGRP  SRVID =30
CLOPT = "-A -- -n 20 -d"
RQPERM=0666  RPPERM=0666

```

```

ARTJESPURGE       SRVGRP =ARTGRP  SRVID = 100
CLOPT = "-A -- "

```

```

AUTHSVR  SRVGRP=ARTGRP  SRVID=104  CLOPT="-A"
RQPERM=0666  RPPERM=0666

```

```

*SERVICES

```

- Permission settings for file system
 - File permission for the following folders is set to 777 under `$JES_BASE_DIR`:
 - `$DATA_SOURCE`
 - `$DATA`
 - `$MT_LOG`
 - `$SPOOL`
 - `$MT_TMP`
 - `$JESROOT`
 - The following empty files are created under `$JES_BASE_DIR` with file permission 666:
 - `$MT_ACC_FILEPATH/AccWait`
 - `$MT_ACC_FILEPATH/AccLock`
 - File permission for `TUXCONFIG` is set to 644 under `$APPDIR`:

Permission Requirements When Enabling User Substitution

When you set privilege mode to either **USER_IDENTICAL** or `USER_MAPPING`, the following permission requirements must be met.

- Permission requirements for `UBBCONFIG`
 - The `TUXCONFIG` file `*RESOURCE` section `SECURITY` parameter must be `USER_AUTH`, `ACL`, or `MANDATORY_ACL`.
 - `*RESOURCES` section `AUTHSVC` and `SERVERS` section `AUTHSVR` parameters must be set properly in accordance with `USER_AUTH`, `ACL`, or `MANDATORY_ACL` settings.
- Permission requirements for file system
 - The home directory access permission where JES is deployed should be granted to the user who initiates the deployment, and JES executor before the deployment.
 - App home may be created by Resource Broker during deployment if it does not exist. In this case, you need to configure App home directory with access permission for other group.
 - The `$JES_BASE_DIR` directory should be created before deployment and be readable/writable/executable for JES domain creator and all JES job executors.
- Permission Requirements for Creating, Configuring, and Booting JES Domain
 - Resource broker needs "root" permission to create, configure, or boot JES domain.

- For the SHM domain, JES domain should be created, configured, or booted by root user.
 - For the MP domain, if JES servers are on a machine, the machine should be created, configured, or booted by root user.
 - If a machine is created, configured, or booted up by root user, `tlisten` on that machine must be booted by root user as well before deploying the Oracle Tuxedo domain using resource broker.
- SUDO configurations must be made for performing activities with root user privileges.
- From Oracle Enterprise Manager Cloud Control, click **Setup > Security > Privilege Delegation**. For more information, see [Configuring Privilege Delegation Settings](#) in Oracle Enterprise Manager Lifecycle Management Administrator's Guide.

Misc Configuration

You can configure the following Misc Configuration field options to provide ART Batch related information:

- JES queue init script

The JES queue initializing script. The one provided in the JES simpjob example is used as the default template. You can customize text field, if you do so, the queue file name must be "QUE".

- JES install dir

Mandatory. The ART Batch installation directory must be specified on the page.

Note: When configuring an MP domain, the ART Batch installation directory path on all the machines must be the same.

- PDKSH

Optional. The `PDKSH` executable file which is used for running ART Batch. If not specified, the system default `ks`h command is used at runtime.

Assembling and Deploying JES Domain

There are two types of deployment:

- Static Deployment (or Full Deployment)

The domain does not exist at the outset. You need to create and assemble the JES enabled domain in the Domain Editor from scratch, and then deploy it onto the target machines.

- **Dynamic Deployment**

The JES domain has already been deployed and booted up. Under certain circumstances, some new servers are added into the domain and activated dynamically.

Notes:

- When redeploying an existing domain, some files will be overridden (Ex `setenv.sh`, `config.sh`, etc).
- All the runtime environment variables introspected from running domain will be exported to `setenv.sh` when redeploying.

Static Deployment

When using static deployment, all ART Batch related system servers are assembled in the `UBBCONFIG` file as other servers. The saved JES Configuration file is packed in the distribution package for deployment. The deployed JES configuration file name is the one pointed by the `-i` option of `ARTJESADM CLOPT`. The `JESROOT` directory is not created at this stage; it is created automatically by ART Batch at runtime.

The following environment variable settings are added to the `setenv.sh` file if JES is configured:

```
JESDIR=<full path of ART JES software>
MT_KSH=<full path of pdksh>
MT_ROOT=$JESDIR/ejr
PATH=$JESDIR/bin:$PATH
QMCONFIG=$APPDIR/QUE
DATA_SOURCE=$APPDIR/data_source
DATA=$APPDIR/data
TMP=$APPDIR/tmp
MT_TMP=$APPDIR/tmp
MT_ACC_FILEPATH=$APPDIR/acc
MT_LOG=$APPDIR/Logs/log
SPOOL=$APPDIR/Logs/sysout
```

During configuration, the `JES_QUEUE` system is initialized by the default `jesqinit` script or user-defined script.

Dynamic Deployment

Package-level dynamic deployment is handled by the existing Resource Broker framework.

The JES-related Tuxedo system servers are described in the `Properties.xml` which is included in various application packages. You can add these packages into an existing domain as candidates and define their policies. When the policy is evaluated to be fulfilled, the packages are then deployed and the servers described in it are activated.

Note: Dynamic deployment is only on the package (server) level. `JESCONFIG` does not support dynamic deployment (for example, running Tuxedo domain without ART Batch configured), the `JESCONFIG` cannot be enforced dynamically.

Using Resource Broker for ART CICS

Server Configuration

When using resource broker for ART CICS, you must configure several Tuxedo provided system servers in the `UBBCONFIG` file to enable resource broker for ART CICS Runtime. For existing domain, such information is introspected automatically. Note that ART CICS and COBOL compiler must be installed on the same location for each machine configured in domain.

There are three ways to add these Tuxedo system servers.

- [Adding Tuxedo System Servers Manually](#)
- [Adding Tuxedo System Servers Using Application Package](#)
- [Adding Tuxedo System Servers Using Server Template](#)

Adding Tuxedo System Servers Manually

For information, see [Adding Tuxedo System Servers Manually](#).

Adding Tuxedo System Servers Using Application Package

The ART CICS specific system servers can be divided into two categories:

- The ART CICS provided system servers:
 - ARTADM
 - ARTCNX

- ARTMON
 - ARTTCPH
 - ARTTCPL
 - ARTTDQ
- The system servers that can be built by users:
 - ARTSTR1
 - ARTSTRN
 - ARTATR1
 - ARTATRn
 - ARTCTR1
 - ARTCTRn
 - ARTDPL
 - ARTTSQ
 - ARTCKTI

For the ART CICS provided system servers, only the server name should be referred to `inProperties.xml`.

For the system servers that can be built by users, the server name you define in the `Properties.xml` must be prefixed by the listed server names.

Adding Tuxedo System Servers Using Server Template

You can add the ART CICS related servers using the server template. Do the following steps:

1. From the [Domain UBBCONFIG Editor](#), click **CICSConfig**.
2. Select one or more machines from the Machines list, and choose one or more of the following templates:
 - 3270 Terminal Initiated CICS Transaction, adds these servers to the machine:
 - ARTTCPL
 - ARTADM
 - ARTCNX
 - ARTSTRN
 - ARTATRn
 - Non-3270 Clients Initiated CICS Transaction, adds these servers to the machine:

- ARTWTRN
 - ARTADM
 - Distributed Program Link, adds these servers to the machine:
 - ARTADM
 - ARTDPL
 - All, applies all the three template types together.
3. Click **Apply Server Template**.

CICS Application Package

The CICS application package is introduced specifically to handle various CICS resources.

Package Content

Similar to the Tuxedo application package, the CICS application package is a .zip file that packs assets for running CICS Runtime. It includes components described in the following.

Properties.xml

The `Properties.xml` (located in the root of the zip file), is used to describe the general properties of the whole package. These general Tuxedo application package parameters are also applicable to CICS packages:

- `PackageName`
- `TuxedoVersion`
- `CICSVersion`
- `SupportedOS`
- `TuxedoWordSize`
- `MachineArch`

All the elements are of the same meaning as the ones in a Tuxedo application package except for `SupportedOS`. There is no ART CICS for Windows platform. Only Linux, SunOS, and AIX are supported.

A new element, `CICSVersion`, is added to `Properties.xml`. It indicates which ART CICS version such package is built. This field is used for extension in the future. It can be configured to any value that matches the regex rule `1[1-9](\.[0-9]){3,4}`. For the 12cR2 release, the suggested value is 12.1.3.0.0.

To distinguish from Tuxedo application packages, a type attribute is introduced for the `ApplicationProperties` root element. To mark the package a CICS application package, set the type attribute to `CICS`. If the type is not set or set to `TUX`, the package is regarded as a Tuxedo application package.

Resource Configuration Files

Following CICS resource configuration files are supported by CICS Runtime:

- Groups Configuration (`list_of_groups.desc`)

- Tranclasses (`transclasses.desc` file)

It is not allowed to define two same tranclasses.

- Transactions (`transactions.desc` file)

It is not allowed to define two same transactions, even if they are in different CICS groups.

- Programs (`programs.desc` file)

It is not allowed to define two same programs in one CICS group. It is allowed to define two same programs in different groups.

- TS Queue Model (`tsqmodel.desc` file)

It is not allowed to define two same TSQ models, even if they are in different CICS groups.

- Mapsets (`mapsets.desc` file)

It is allowed to define two same mapsets in the file, but only the first one is used.

- Typeterms (`typeterms.desc` file)

It is allowed to define two same typeterms in the file, but only the first one is used.

- Enqmodel (`enqmodel.desc`)

It is not allowed to define two same "enqmodel" in the file, even if they are in different CICS groups.

- Extra TDQUEUE (`tdqextra.desc`)

It is not allowed to define two same `tdqueue`, even if they are in different CICS groups.

- Intra TDQUEUE (`tdqintra.desc`)

It is not allowed to define two same `tdqueue`, even if they are in different CICS groups.

Each CICS application package can contain a set of the previously described `.desc` files, and each `.desc` can only occur at most once in one package. All the `.desc` files should be located in the package `DESC` sub-directory.

COBOL Programs

All COBOL `.gnt` files are located in the `COBLIB` sub-directory of the package. These COBOL programs are referenced by the `programs.desc` file at runtime.

Mapdefs

All the `.mpdef` files are located in the `MAP` sub-directory of the package. They are used by the `mapsets.desc` file.

Shell Scripts

The script used to create `/Q` for `ARTTDQ` and `TSQ DB` may also be packed in the package. They should be located in the `SCRIPT` sub-directory and are called when the package is deployed. All the shell scripts are supposed to be independent of each other, and the calling sequence does not impact the script running result.

Package Management

The CICS application packages are managed in the Tuxedo Summary page (same as Tuxedo application packages). For more information, see [Application Package Management](#).

When uploading, a validation is performed for `Properties.xml` file and `.desc` resource configuration files in each package. The `Properties.xml` file is validated against the `.xsd` XML schema, while the resource configuration files are validated for the correct file format and content (for example, unique key field constraint, column number, and required columns etc.).

If any validation fails, the uploading fails with error message appearing. All errors in the package are reported in the pop-up dialog.

For existing domains, one default package is generated and applied to current ART CICS package list. This special package only contains resource files used in runtime environment.

To dynamically deploy a new added machine, configuration shell script including creating `TLOG/Q`, special environment variable export, and so on must be added to CICS package.

CICS Configuration Page

To enter the CICS Configuration page, do the following:

1. From [Domain UBBCONFIG Editor](#), click **CICSConfig**.
2. Fill in CICS and COBOL installation directories Click **Edit Configurations**.
3. In the Confirmation message that appears, click **Yes**.

For more information, see [Configuring CICS Configuration](#) or click **Help** on the upper right.

Assembling and Deployment

During deployment, all saved CICS resource files and added CICS application packages are retrieved from the software library and then packed into the distribution package together with other Tuxedo domain assets.

In the deployed `APPDIR`, each CICS application package corresponds to a sub-directory named according to the package name. All the merged `.desc` resource files are located in the `APPDIR/resources` directory, which is referenced by the `KIXCONFIG` environment variable.

Note: Package level dynamic deployment is not supported for existing ART CICS domains.

The following environment variable settings are added to the `setenv.sh` file:

```
KIXDIR=<full path of ART CICS software>
KIXCONFIG=<full path of ART CICS resources configuration files, fixed as
$APPDIR/resources>
PATH=$KIXDIR/bin:$PATH
COBPATH=APP1/COBLIB:APP2/COBLIB:APP3/COBLIB: ... :$COBPATH
COB_LIBRARY_PATH=$COBPATH:$COB_LIBRARY_PATH
KIX_MAP_PATH=APP1/MAP:APP2/MAP:APP3/MAP: ... :$KIX_MAP_PATH
```

Notes:

- If any environment variable not listed above is used, you must specify it manually in the `confide` script or `boot` script.
- For existing domains, all the environment variables are collected from master machine, so these variables are distributed to the target machine when re-deploying. For slave machine, if the same environment variable is defined before `tlisten` is started, the value from master machine is replaced. `KIXCONFIG` must be set before `tlisten` is started on master machine.

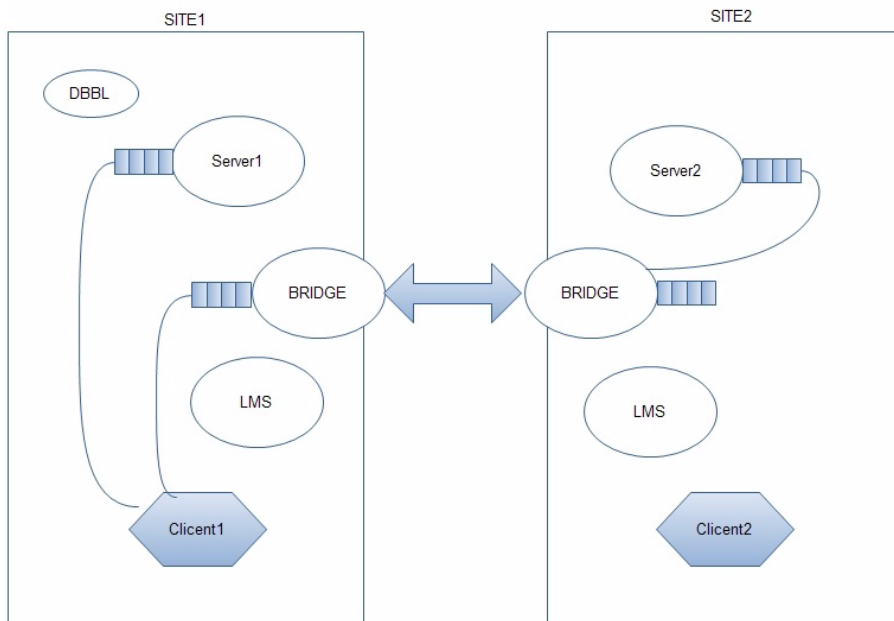
Dynamic Request Broker

Dynamic Request Broker is a Tuxedo load balancing functionality enhancement that minimizes response time for request calls and provides more reasonable workload distribution across multiple servers (especially in a multiprocessor (MP) domain). More specifically, Dynamic Request Broker is a mechanism that introduces new request routing metrics (for example, service execution time, network time, etc.), as a replacement for the static load values set in `UBBCONFIG` file and reflects service and network load level dynamically. Furthermore, it syncs metrics across all distributed machines in an MP domain to achieve optimal resource utilization.

Typically, Dynamic Request Broker can be useful in following cases:

- In an MP domain, one of the servers in the candidate list is handling the requests from different machines. As shown in [Figure 2-13](#), `server2` serves both `client1` and `client2` that are located on different machines.

Figure 2-13 Dynamic Request Broker in MP Domain



- The service execution time is not constant; however, the load value specified in the `UBBCONFIG` file is static, which may not be applicable for the service as the situation changes.

- The server may be added or removed during run-time; however, the load value specified in the `UBBCONFIG` file is static, which may not be applicable for the service as the situation changes.

Note: Dynamic Request Broker only supports Oracle Tuxedo 12c and later releases.

Enabling/Disabling Dynamic Request Broker

You can enable or disable dynamic request broker functionality by clicking **Control** -> **Enable/Disable Request Broker** from the Tuxedo Domain menu on Enterprise Manager Console.

Viewing Dynamic Load Information

After the dynamic request broker function is enabled, you can view dynamic load information using following command:

```
$ tadmin
> psc -m all -v
```

Integration with OVAB

Oracle Virtual Assembly Builder (OVAB), enables administrators to quickly configure and provision Oracle Tuxedo applications to cloud environments. Using Enterprise Manager for Oracle Tuxedo you can take advantage of Enterprise Manager policy mechanisms to run an OVAB script that deploys Tuxedo applications under specific conditions.

Supported Tuxedo Version

- Tuxedo 11gR1 (11.1.1.2.0) and later
- Tuxedo 12cR1

Supported Platform

- Linux platform

Scaling Oracle Tuxedo Machines Number with OVAB Script

The following provides an integration example which triggers Enterprise Manager to call OVAB script to scale the number of Oracle Tuxedo slave machine.

Deploying Tuxedo Application

Suppose there is a Golden system consisting of three MP machines: master, backup, and slave, and we want to create a deployment plan which scales the number of slave machine between 1 to 10, follow these steps to deploy the Tuxedo application:

1. Login to the machine on where the OVAB is installed.
2. Under the OVAB bin directory, create the Connection and Target using the following commands:

```
./abctl createDeployerConnection -name WLSADMIN -url
http://10.182.73.21:7001 -username ovabAdmin
```

```
./abctl createTarget -name myTarget -type ovm -connectionName WLSADMIN
-properties ovm.user=admin ovm.pwd=Tuxqa123
ovm.url=http://10.182.73.11:7001 ovm.vmOperationTimeout=3600
ovm.vmmversion=3.0 ovm.poolName=MyServerPool -default
```

You can check Connection and Target availability using the following commands:

```
./abctl describeDeployerConnections
./abctl describeTargets -connectionName WLSADMIN
```

3. Introspect Tuxedo applications, as follows:

```
./abctl introspectTuxedo -TUXDIR /testarea/tuxreg/tuxedo/tuxedo12c
-TUXCONFIG /testarea/tuxreg/workspace/OVAB/tux/work/domain/tuxconfig
-environmentScript
/testarea/tuxreg/workspace/OVAB/tux/work/domain/setenv_ovab.sh
-remoteHost bej301175.cn.oracle.com -remoteUser tuxreg -name master
-force
```

```
./abctl introspectTuxedo -TUXDIR /testarea/tuxreg/tuxedo/tuxedo12c
-TUXCONFIG /testarea/tuxreg/workspace/OVAB/tux/work/domain/tuxconfig
-remoteHost bej301165.cn.oracle.com -remoteUser tuxreg -name slave
-force
```

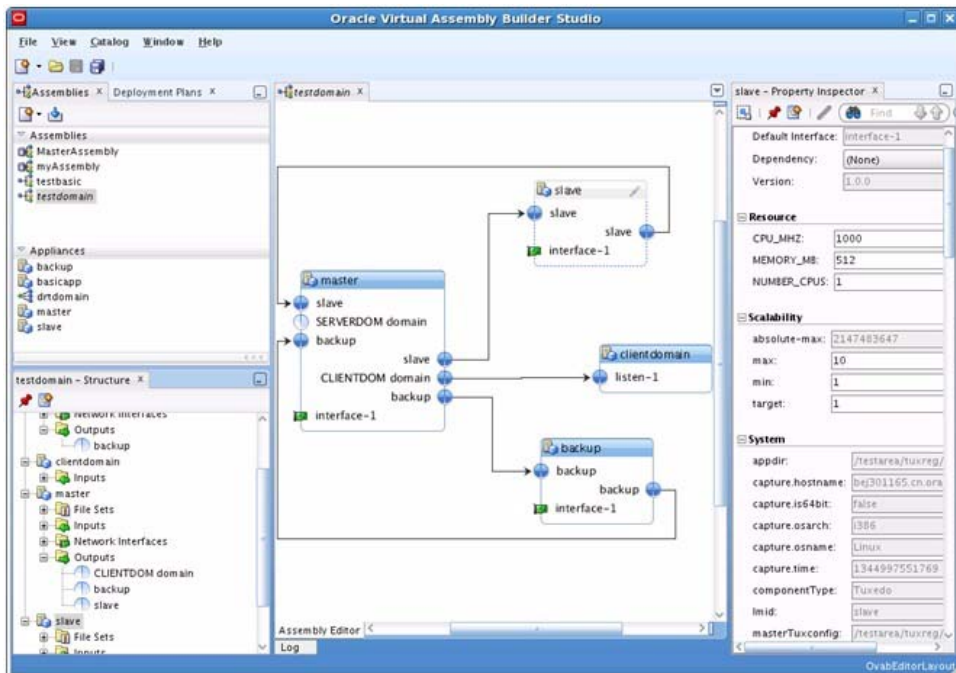
Using the Enterprise Manager for Oracle Tuxedo

```
./abctl introspectTuxedo -TUXDIR /testarea/tuxreg/tuxedo/tuxedo12c  
-TUXCONFIG /testarea/tuxreg/workspace/OVAB/tux/work/domain/tuxconfig  
-remoteHost bej301168.cn.oracle.com -remoteUser tuxreg -name backup  
-force
```

```
./abctl createAssembly -name testdomain  
./abctl addToAssembly -name master -into testdomain  
./abctl addToAssembly -name slave -into testdomain  
./abctl addToAssembly -name backup -into testdomain
```

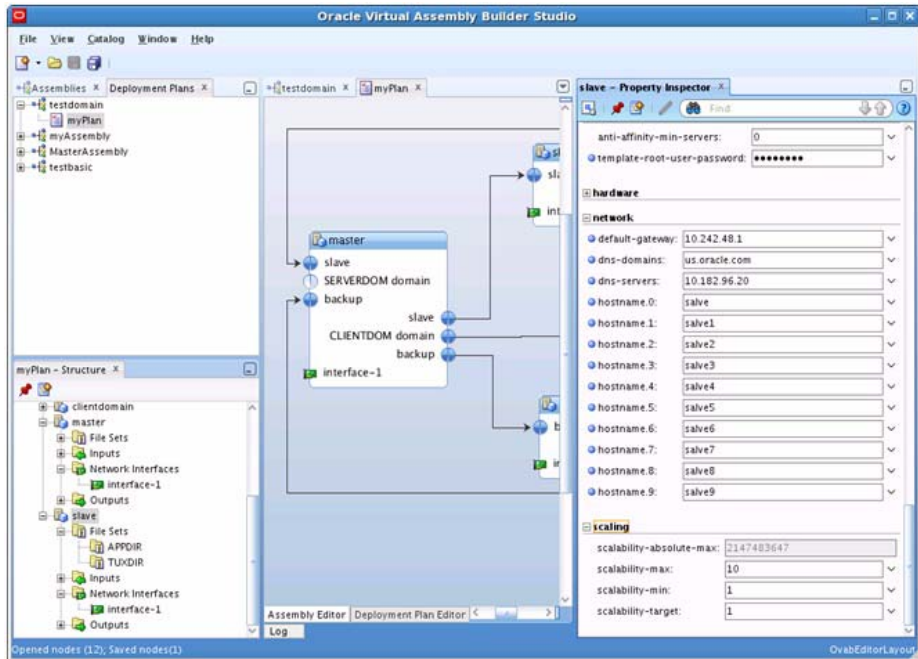
4. From OVAB GUI, link resources as shown in [Figure 2-14](#)

Figure 2-14 Link Resources



5. From slave - Property Inspector, set the max, min, and target value scaling field as 10, 1, and 1, respectively as shown in [Figure 2-15](#)

Figure 2-15 Set Scaling Parameters

**Notes:**

- The Hostname entered in a deployment plan must be the full name (for example, slce04vm010.us.oracle.com, the short name slce04vm010 will not work).
- All virtual machine hostnames must be configured in the DNS server.

6. Create and register Archive.

```
./abctl createAssemblyArchive -name testdomain -platform OVM -force

./abctl uploadAssemblyArchive -fileName
/u01/slce04cn06/general/testarea/tuxreg/OVABR2/oracle.ovab/ab_instance/
archives/testdomain.ova -name testdomain -connectionName WLSADMIN

./abctl downloadAssemblyMetadata -connectionName WLSADMIN -name
testdomain -generateplan -version 1
```

```
./abctl registerAssemblyArchive -name testdomain -connectionName  
WLSADMIN -waitForComplete -pollTime 300 -version 1 -target myTarget
```

7. Create Instance and deploy Instance:

```
./abctl createAssemblyInstance -deploymentPlan  
/u01/s1ce04cn06/general/testarea/tuxreg/OVABR2/oracle.ovab/ab_instance/  
catalog/metadata/testdomain/deploymentPlans/myPlan/deploymentPlan.xml  
-name testdomain -version 1 -target myTarget -connectionName WLSADMIN
```

```
./abctl deployAssemblyInstance -assemblyInstanceId  
OV0G_5Yn8_testdomain8_1 -connectionName WLSADMIN -waitForComplete  
-pollTime 3600
```

Defining Enterprise Manager Policy

Creating an OVAB Script

Create an OVAB script `deploy_ovab.sh` as follows:

```
./abctl ./scale -scalingGroupId MpNZ-qhmZ_testdomain_1:testdomain/slave  
-target 2 -connectionName WLSADMIN
```

You can find the `scalinggroupID` using the following command:

```
./abctl describeScalingGroups -connectionName MasterDeployer
```

Creating a Notification Method

To create an Enterprise Manager notification method associated with the OVAB script created in the previous step, do the following steps:

1. From the Enterprise Manager Cloud Control, click **Setup > Notifications > Notification Methods** at the upper right of the page.
2. On the **Notification Methods** page, choose adding **OS Command** in the **Scripts and SNMP Traps** field. Click **Go**.
3. Follow the instructions on screen to give a notification method name and enter the OVAB script created beforehand in **OS Command**, as shown in [Figure 2-16](#).

Figure 2-16 Add OS Command



Setup

Notification Methods > Add OS Command

Add OS Command

Define a new Notification Method using an Operating System command or script that will be called in Incident Rules.

* Name:

Description:

* OS Command:

Enter a fully qualified command or script (example: /bin/perl /u1/bin/myscript.pl). This OS command needs to be located on all OMS hosts.

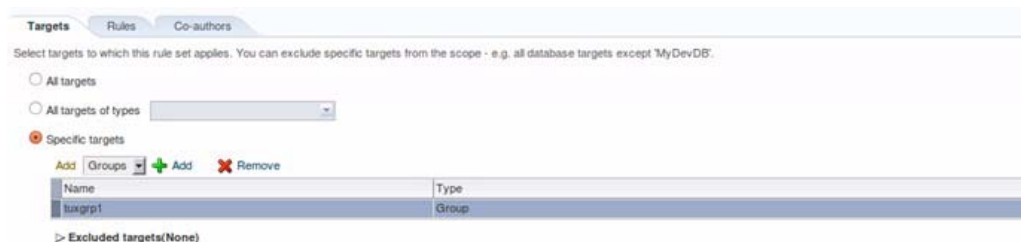
4. Click **OK**.

Creating an Incident Rule

To create an incident rule that triggers the notification method defined above, do the following steps:

1. From the Enterprise Manager Cloud Control, click **Setup > Incidents > Incident Rules** on the upper right of the page.
2. On the page that appears, click **Create Rule Set**.
3. Define a rule name and specify the target group, as shown in [Figure 2-17](#).

Figure 2-17 Specify Target Group



Targets Rules Co-authors

Select targets to which this rule set applies. You can exclude specific targets from the scope - e.g. all database targets except 'MyDevDB'.

All targets

All targets of types

Specific targets

Add

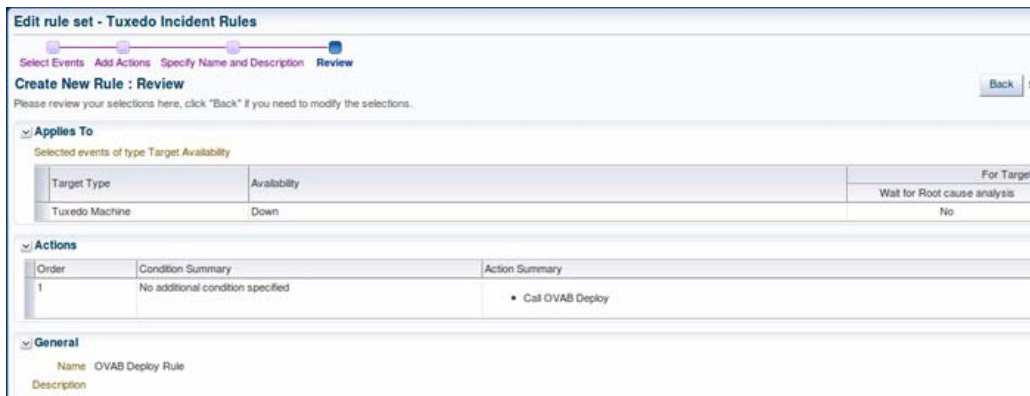
Name	Type
luxgrp1	Group

> Excluded targets(None)

4. Click **Rules** tag, and then click **Create**.
5. In the dialog box that appears, choose **Incoming events and updates to events**. Click **Continue**.
6. On the **Create New Rule: Select Events** page, select **Type**, **Target Availability**, and then **Specific events of type Target Availability**.

7. Click **Add**, and choose **Tuxedo Machine** on the **Target Type** list.
8. In the Availability States field that appears, select **Down** and click **OK**.
9. On the **Create New Rule: Select Events** page, click **Next**.
10. On the **Create New Rule: Add Actions** page, add an action by clicking **Add**.
11. Choose the OVAB Deploy under the **Advanced Notifications** section,. Click **Continue**.
12. Follow the instructions on screen and click **Next** until you enter the **Create New Rule: Review** page.
13. Check your selections and click **Continue** to proceed to the final step.
14. You are prompted that the rule has been successfully added. Click the **Save** button on the page to save the newly added rule.

Figure 2-18 Newly Added Tuxedo Incident Rule



Edit rule set - Tuxedo Incident Rules

Select Events Add Actions Specify Name and Description **Review**

Create New Rule : Review Back

Please review your selections here, click "Back" if you need to modify the selections.

Applies To

Selected events of type Target Availability

Target Type	Availability	For Target
Tuxedo Machine	Down	Wait for Root cause analysis No

Actions

Order	Condition Summary	Action Summary
1	No additional condition specified	• Call OVAB Deploy

General

Name OVAB Deploy Rule
Description

DM_MIB MBean Interface

TSAM Plus wraps configurations `DM_MIB` classes to MBean operations.

Creating the TuxedoDomainGatewayTargetMBean Object

`JMXConnectorFactory.connect` (`JMXServiceURL` `serviceURL`, `Map<String,?>` `environment`) is used to create a connection to the tlisten JMX server. The "serviceURL" format is: "service:jmx:rmi:///jndi/rmi://%s:%s/server", `host`, `port`, where

"host" is the host name tlisten runs, and "port" is the "tlisten -j" port value. The environment key is "jmx.remote.credentials". The environment value is a String array.

Table 2-13 lists its elements.

Table 2-13 Environment Value Elements

Index	Value	Description
0	Tuxedo user name	Its value may be null when the Tuxedo resource "SECURITY" value is not USER_AUTH, ACL, or MANDATORY_ACL.
1	Tuxedo user password	Its value may be null when the Tuxedo resource "SECURITY" value is not USER_AUTH, ACL, or MANDATORY_ACL.
2	Tuxedo application password	Its value may be null when the Tuxedo resource "SECURITY" value is not APP_PW.
3	Tuxedo DOMAINID	
4	Tuxedo IPCKEY	

ObjectName domainId + "_" + ipckey +
":type=tuxedo_domain_gateway,group_name=" + groupName + ",SRVID=" + SVRID is used to create the TuxedoDomainGatewayTargetMBean object, where domainId is the Tuxedo domain DOMAINID, ipckey is the Tuxedo domain IPCKEY, groupName is the server GWTDOMAIN SRVGRP , and SVRID is the server GWTDOMAIN SVRID.

get Operations

get operations are used to get the existing TM MIB entities.

Name

get+MIB class name (in camel naming format)+s (if the return type is TabularData)

Parameters

All parameters are input ones. The format is [keyType keyValue]*, CompositeData filter).

- keyValue: the key fields that are required when calling the DM_MIB get interface

- filter: includes all optional fields.

Return Type

The return type is `CompositeData` if only one record is returned; otherwise the return type is `TabularData`.

Exception

It is always the `MBeanException` type.

add Operations

`add` operations are used to add a new TM MIB entity.

Name

`add+MIB class name (in camel naming format)`

Parameters

All parameters are input ones. The format is `CompositeData parameters, where parameters` includes all fields.

Return Type

`void`

Exception

It is always the `MBeanException` type.

delete Operations

`delete` operations are used to delete an existing TM MIB entity.

Name

`delete+MIB class name (in camel naming format)`

Parameters

All parameters are input ones. The format is `[keyType keyValue]+)`, where `keyValue` is the key fields that are required when calling the `DM_MIB set` interface.

Return Type

void

Exception

It is always the `MBeanException` type.

update Operations

`update` operations are used to update an existing TM MIB entity.

Name

`update+MIB class name (in camel naming format)`

Parameters

All parameters are input ones. The format is `[keyType keyValue]+, CompositeData parameters)`

- **keyValue:** the key fields that are required when calling the `DM_MIB set` interface.
- **parameters:** includes all fields to be updated.

Note: Key fields must not be in the "CompositeData parameters", otherwise an exception will be thrown.

Return Type

void

Exception

It is always the `MBeanException` type.

Application Programming Interfaces

All MBean `DM_MIB` operations are in the class `"oracle.tuxedo.jmx.tux.emmodel.TuxedoDomainGatewayTarget"`, which extends `"javax.management.modelmbean.RequiredModelMBean"`.

Here is the list of them:

Table 2-14 Tdomain and Command Interfaces

MBeanOperation Name	Class Name	TA_OPERATION	Description
TabularData getACLs (Composite Data filter)	T_DM_ACL	get	Get an ACL list
void addACL (Composite Data parameters)	T_DM_ACL	set	Add a new ACL
void deleteACL (String ACLName)	T_DM_ACL	set	Delete an ACL
void updateACL (String ACLName, CompositeData parameters)	T_DM_ACL	set	Update an existing ACL
TabularData getEventIns (Compo siteData filter)	T_DM_EVT_IN	get	Get events, which can be received from remote domains
void addEventIn (Compos iteData parameters)	T_DM_EVT_IN	set	Add an event, which can be received from remote domains
void deleteEventIn (Str ing eventName, String lAccessPoint)	T_DM_EVT_IN	set	Delete an event, which can be received from remote domains
TabularData getEventOuts (Comp ositeData filter)	T_DM_EVT_OUT	get	Get events, which can be sent to remote domains
void addEventOut (Compo siteData parameters)	T_DM_EVT_OUT	set	Add an event, which can be sent to remote domains

Table 2-14 Tdomain and Command Interfaces

MBeanOperation Name	Class Name	TA_OPERATION	Description
void deleteEventOut (String eventName, String lAccessPoint, String rAccessPointList)	T_DM_EVT_OUT	set	Delete an event, which can be sent to remote domains
TabularData getExports (CompositeData filter)	T_DM_EXPORT	get	Get local resources that are exported to one or more remote domains through a local access point
void addExport (CompositeData parameters)	T_DM_EXPORT	set	Add a local resource that is exported to one or more remote domains through a local access point
void deleteExport (String resourceName, String lAccessPoint)	T_DM_EXPORT	set	Delete a local resource that is exported to one or more remote domains through a local access point
void updateExport (String resourceName, String lAccessPoint, CompositeData parameters)	T_DM_EXPORT	set	Update a local resource that is exported to one or more remote domains through a local access point
TabularData getImports (CompositeData filter)	T_DM_IMPORT	get	Get remote resources that are imported through one or more remote domain access points
void addImport (CompositeData parameters)	T_DM_IMPORT	set	Add a remote resource that is imported through one or more remote domain access points

Table 2-14 Tdomain and Command Interfaces

MBeanOperation Name	Class Name	TA_OPERATION	Description
void deleteImport (String resourceName, String lAccessPoint, String rAccessPointList)	T_DM_IMPORT	set	Delete a remote resource that is imported through one or more remote domain access points
void updateImport (String resourceName, String lAccessPoint, String rAccessPointList, CompositeData parameters)	T_DM_IMPORT	set	Update a remote resource that is imported through one or more remote domain access points
TabularData getLocals (CompositeData filter)	T_DM_LOCAL	get	Get local domain access points
void addLocal (CompositeData parameters)	T_DM_LOCAL	set	Add a local domain access point
void deleteLocal (String accessPoint)	T_DM_LOCAL	set	Delete a local domain access point
void updateLocal (String accessPoint, CompositeData parameters)	T_DM_LOCAL	set	Update a local domain access point
TabularData getPasswords (CompositeData filter)	T_DM_PASSWORD	get	Get passwords for interdomain authentication through access points of type TDOMAIN

Table 2-14 Tdomain and Command Interfaces

MBeanOperation Name	Class Name	TA_OPERATION	Description
oid addPassword(Compo siteData parameters)	T_DM_PASSWORD	set	Add a password for interdomain authentication through access points of type TDOMAIN
void deletePassword(St ring lAccessPoint, String rAccessPoint)	T_DM_PASSWORD	set	Delete a password for interdomain authentication through access points of type TDOMAIN
void reCryptPasswords(String lAccessPoint, Stri ng rAccessPoint)	T_DM_PASSWORD	set	Re-encrypt all passwords using a new encryption key. Applies to all password instances in the T_DM_PASSWORD classes
TabularData getRemotes(Compos iteData filter)	T_DM_REMOTE	get	Get remote domain access points
void addRemote(Composi teData parameters)	T_DM_REMOTE	set	Add a remote domain access point
void deleteRemote(Stri ng accessPoint, String type)	T_DM_REMOTE	set	Delete a remote domain access point
void updateRemote(Stri ng accessPoint, String type, CompositeData parameters)	T_DM_REMOTE	set	Update a remote domain access point
TabularData getResources(Comp ositeData filter)	T_DM_RESOURCES	get	Get resources

Table 2-14 Tdomain and Command Interfaces

MBeanOperation Name	Class Name	TA_OPERATION	Description
void setResources (CompositeData parameters)	T_DM_RESOURCES	set	Set resources
TabularData getRoutings (CompositeData filter)	T_DM_ROUTING	get	Get all routing list to a domain
void addRouting (CompositeData parameters)	T_DM_ROUTING	set	Add a routing to a domain through a remote domain access point
void deleteRouting (String routingName, String bufType)	T_DM_ROUTING	set	Delete a routing to a domain through a remote domain access point
void updateRouting (String routingName, String bufType, CompositeData parameters)	T_DM_ROUTING	set	Update a routing to a domain through a remote domain access point
TabularData getTDomain (CompositeData filter)	T_DM_TDOMAIN	get	Get the TDomain specific configuration for a local or remote domain access point
void addTDomain (CompositeData parameters)	T_DM_TDOMAIN	set	Add the TDomain specific configuration for a local or remote domain access point

Table 2-14 Tdomain and Command Interfaces

MBeanOperation Name	Class Name	TA_OPERATION	Description
void deleteTDomain(String accessPoint, String nwaddr, String lAccessPoint)	T_DM_TDOMAIN	set	Delete the TDomain specific configuration for a local or remote domain access point
void setTDomain(String accessPoint, String nwaddr, String lAccessPoint, CompositeData parameters)	T_DM_TDOMAIN	set	Update the TDomain specific configuration for a local or remote domain access point

Security Requirement

The security level is controlled by Oracle Tuxedo. The corresponding credentials are required when the configuration item `SECURITY` value in the Tuxedo `UBBCONFIG` is not `NONE`.

`$TUXDIR/jmx/tmjmx_tux.jar` is required to add the `JAVA` classpath, in which `oracle.tuxedo.jmx.tux.utility.Encryption.getInstance().encrypt()` is used to encrypt the `APP` password and user passwords.

When the security level "USER_AUTH" or above is enabled, the client of the user used to login is required to be set to `tpsysadm` when calling `tpusradd`.

User Cases

User Case 1 - Specify the network address associated with a access point

1. Intall TSAM Plus Agent in Oracle Tuxedo 12c
2. Start `tlisten` with the "-j" option using the following command:

```
tlisten -j rmi://<host>:<rmiport> -l //<host>:<tlistenport>
```
3. Boot up the tuxedo domain.

4. Compile and run the following source file:

Listing 2-6 User Case 1 - Specify the network address associated with a access point

```
import java.io.Closeable;
import java.io.IOException;
import java.net.MalformedURLException;
import java.util.HashMap;
import java.util.Map;

import javax.management.MBeanServerConnection;
import javax.management.ObjectName;
import javax.management.openmbean.CompositeData;
import javax.management.openmbean.CompositeDataSupport;
import javax.management.openmbean.CompositeType;
import javax.management.openmbean.SimpleType;
import javax.management.openmbean.TabularData;
import javax.management.remote.JMXConnector;
import javax.management.remote.JMXConnectorFactory;
import javax.management.remote.JMXServiceURL;

public class TuxedoDomainGatewayTargetTestCustom implements Closeable {

    private JMXConnector jmxcr = null;
    private MBeanServerConnection mbsc = null;
    private ObjectName tuxedoDomainGatewayTargetMBean = null;

    private String host = "rno05038.us.oracle.com";
```

```
private String port = "5039";

private String userName = "";

private String password = "";

private String appPassword = "";

private String domainId = "DOMAIN1";

private String ipckey = "123468";

private String groupName = "SYSGRP1";

private String SVRID = "3";

TuxedoDomainGatewayTargetTestCustom() throws Exception {

    String[] credentials = new String[] { userName, password,
appPassword,
                                domainId, ipckey};

    jmxrc = createJMXConnector(credentials);
    mbsc = jmxrc.getMBeanServerConnection();
    tuxedoDomainGatewayTargetMBean = new ObjectName(domainId +
"_" + ipckey
```

Using the Enterprise Manager for Oracle Tuxedo

```

        + ":type=tuxedo_domain_gateway,group_name=" +
groupName
        + ",SRVID=" + SVRID);
    }

    @Override
    public void close() throws IOException {
        jmx.c.close();
    }

    /*get all Tdomain items with the filter
    "TA_DMACCESSPOINT=accessPoint"*/
    public void getTDomain(String accessPoint) throws Exception {
        String[] itemNames = { "TA_DMACCESSPOINT" };
        Object[] itemValues = new Object[] { accessPoint };
        javax.management.openmbean.OpenType<?>[] itemTypes = new
javax.management.openmbean.OpenType<?>[] { SimpleType.STRING };

        CompositeType compositeType = new CompositeType(
            "tuxedo_domain_gateway",
            "tuxedo_domain_gateway", itemNames,
            itemNames, itemTypes);
        CompositeDataSupport filter = new
CompositeDataSupport(compositeType,
            itemNames, itemValues);

        TabularData result = (TabularData) mbsc.invoke(
            tuxedoDomainGatewayTargetMBean, "getTDomain",
            new Object[] { filter },

```

```

        new String[] { CompositeData.class.getName()
    });

    if (result != null)
        System.out.println(result.toString());
}

/*add a TDomain item*/
public void addTDomain(String accessPoint, String nwaddr,
    String lAccessPoint) throws Exception {
    String[] itemNames = { "TA_DMACCESSPOINT", "TA_DMNWADDR",
        "TA_DMLACCESSPOINT" };
    Object[] itemValues = new Object[] { accessPoint, nwaddr,
lAccessPoint };
    javax.management.openmbean.OpenType<?>[] itemTypes = new
javax.management.openmbean.OpenType<?>[] {
        SimpleType.STRING, SimpleType.STRING,
SimpleType.STRING };

    CompositeType compositeType = new CompositeType(
        "tuxedo_domain_gateway",
"tuxedo_domain_gateway", itemNames,
        itemNames, itemTypes);
    CompositeDataSupport compositeDataSupport = new
CompositeDataSupport(
        compositeType, itemNames, itemValues);

    mbsc.invoke(tuxedoDomainGatewayTargetMBean, "addTDomain",
        new Object[] { compositeDataSupport },

```

Using the Enterprise Manager for Oracle Tuxedo

```
        new String[] { CompositeData.class.getName()
    });

    }

    /*delete a TDomain item*/
    public void deleteTDomain(String accessPoint, String nwaddr,
        String lAccessPoint) throws Exception {
        mbsc.invoke(tuxedoDomainGatewayTargetMBean, "deleteTDomain",
            new Object[] { accessPoint, nwaddr,
lAccessPoint },
            new String[] { String.class.getName(),
String.class.getName(),
                String.class.getName() });
    }

    private JMXConnector createJMXConnector(String[] credentials)
        throws MalformedURLException, IOException {
        Map<String, Object> env = new HashMap<String, Object>();
        env.put("jmx.remote.credentials", credentials);

        // Create an RMI connector client and
        // connect it to the RMI connector server

        JMXServiceURL url = new JMXServiceURL(String.format(
            "service:jmx:rmi:///jndi/rmi://%s:%s/server",
host, port));
    }
}
```

```

        System.out.print("try to connect " + url + " with the
credentials="
                        + env + "...");

        JMXConnector jmxcr = JMXConnectorFactory.connect(url, env);
        System.out.println("OK");

        return jmxcr;
    }

    /*change the nwaddr of "DOMAIN2_3" from "//localhost:11003"
    * to "//localhost:11004"*/
    public static void main(String[] args) throws Exception {
        String accessPoint = "DOMAIN2_3";
        String oldNwaddr = "//localhost:11003";
        String newNwaddr = "//localhost:11004";
        String lAccessPoint = "DOMAIN1_1";

        TuxedoDomainGatewayTargetTestCustom
tuxedoDomainGatewayTargetTestCustom = new
TuxedoDomainGatewayTargetTestCustom();

        tuxedoDomainGatewayTargetTestCustom.deleteTDomain(accessPoint,
                                                            oldNwaddr, lAccessPoint);

        tuxedoDomainGatewayTargetTestCustom.addTDomain(accessPoint,
newNwaddr,
                                                            lAccessPoint);
    }

```

Note: \$TUXDIR/udataobj/jmx/tmjmx_exceptions.jar should be added to the classpath when running it.

User Case 2 - When Tuxedo Security Level is "USER_AUTH"

1. Install TSAM Plus Agent in Oracle Tuxedo
2. Add the Tuxedo user "user1" with the group name "group1" and password "user123456":

```
tpusradd -g group1 -c tpsysadm user1
Enter password for user1:user123456
Re-enter password for user1:user123456
```

Note: The client name must be "tpsysadm".

3. Specify the app password "app123456" when running tmloadcf:

```
tpusradd -g group1 -c tpsysadm user1
Enter password for user1:app123456
Re-enter password for user1:app123456
```

4. Start tlisten with the "-j" option using the following command:

```
tlisten -j rmi://<host>:<rmiport> -l //<host>:<tlistenport>
```

5. Boot up the tuxedo domain.
6. Compile and run the following source file:

Listing 2-7 User Case 2 - When Tuxedo Security Level is "USER_AUTH"

```
import java.io.Closeable;
import java.io.IOException;
import java.net.MalformedURLException;
import java.util.HashMap;
import java.util.Map;

import javax.management.MBeanServerConnection;
import javax.management.ObjectName;
```



```
import javax.management.openmbean.CompositeData;
import javax.management.openmbean.CompositeDataSupport;
import javax.management.openmbean.CompositeType;
import javax.management.openmbean.SimpleType;
import javax.management.openmbean.TabularData;
import javax.management.remote.JMXConnector;
import javax.management.remote.JMXConnectorFactory;
import javax.management.remote.JMXServiceURL;

public class TuxedoDomainGatewayTargetTestCustom implements Closeable {

    private JMXConnector jmxc = null;
    private MBeanServerConnection mbsc = null;
    private ObjectName tuxedoDomainGatewayTargetMBean = null;

    private String host = "rno05038.us.oracle.com";

    private String port = "5039";

    private String userName = "user1";

    private String password = oracle.tuxedo.jmx.tux.utility.Encryption
        .getInstance().encrypt("user123456");

    private String appPassword =
oracle.tuxedo.jmx.tux.utility.Encryption
        .getInstance().encrypt("app123456");
```

Using the Enterprise Manager for Oracle Tuxedo

```
private String domainId = "DOMAIN1";

private String ipckey = "123468";

private String groupName = "SYSGRP1";

private String SVRID = "3";

TuxedoDomainGatewayTargetTestCustom() throws Exception {

    String[] credentials = new String[] { userName, password,
appPassword,
                                domainId, ipckey};

    jmxrc = createJMXConnector(credentials);
    mbsc = jmxrc.getMBeanServerConnection();
    tuxedoDomainGatewayTargetMBean = new ObjectName(domainId +
"_" + ipckey
                                + ":type=tuxedo_domain_gateway,group_name=" +
groupName
                                + ",SRVID=" + SVRID);
}

@Override
public void close() throws IOException {
    jmxrc.close();
}
```

```

        /*get all TDomain items with the filter
        "TA_DMACCESSPOINT=accessPoint"*/

        public void getTDomain(String accessPoint) throws Exception {

            String[] itemNames = { "TA_DMACCESSPOINT" };

            Object[] itemValues = new Object[] { accessPoint };

            javax.management.openmbean.OpenType<?>[] itemTypes = new
            javax.management.openmbean.OpenType<?>[] { SimpleType.STRING };

            CompositeType compositeType = new CompositeType(
                "tuxedo_domain_gateway",
                "tuxedo_domain_gateway", itemNames,
                itemNames, itemTypes);

            CompositeDataSupport filter = new
            CompositeDataSupport(compositeType,
                itemNames, itemValues);

            TabularData result = (TabularData) mbsc.invoke(
                tuxedoDomainGatewayTargetMBean, "getTDomain",
                new Object[] { filter },
                new String[] { CompositeData.class.getName()
            });

            if (result != null)
                System.out.println(result.toString());
        }

        /*add a TDomain item*/
        public void addTDomain(String accessPoint, String nwaddr,

```

Using the Enterprise Manager for Oracle Tuxedo

```
        String lAccessPoint) throws Exception {
    String[] itemNames = { "TA_DMACCESSPOINT", "TA_DMNWADDR",
        "TA_DMLACCESSPOINT" };
    Object[] itemValues = new Object[] { accessPoint, nwaddr,
lAccessPoint };
    javax.management.openmbean.OpenType<?>[] itemTypes = new
javax.management.openmbean.OpenType<?>[] {
        SimpleType.STRING, SimpleType.STRING,
SimpleType.STRING };

    CompositeType compositeType = new CompositeType(
        "tuxedo_domain_gateway",
"tuxedo_domain_gateway", itemNames,
        itemNames, itemTypes);
    CompositeDataSupport compositeDataSupport = new
CompositeDataSupport(
        compositeType, itemNames, itemValues);

    mbsc.invoke(tuxedoDomainGatewayTargetMBean, "addTDomain",
        new Object[] { compositeDataSupport },
        new String[] { CompositeData.class.getName()
});

}

/*delete a TDomain item*/
public void deleteTDomain(String accessPoint, String nwaddr,
        String lAccessPoint) throws Exception {
    mbsc.invoke(tuxedoDomainGatewayTargetMBean, "deleteTDomain",
```

```

        new Object[] { accessPoint, nwaddr,
lAccessPoint },
        new String[] { String.class.getName(),
String.class.getName(),
String.class.getName() });
    }

    private JMXConnector createJMXConnector(String[] credentials)
        throws MalformedURLException, IOException {
        Map<String, Object> env = new HashMap<String, Object>();
        env.put("jmx.remote.credentials", credentials);

        // Create an RMI connector client and
        // connect it to the RMI connector server

        JMXServiceURL url = new JMXServiceURL(String.format(
            "service:jmx:rmi:///jndi/rmi://%s:%s/server",
host, port));

        System.out.print("try to connect " + url + " with the
credentials="
            + env + "...");
        JMXConnector jmxcr = JMXConnectorFactory.connect(url, env);
        System.out.println("OK");

        return jmxcr;
    }

```

```
/*change the nwaddr of "DOMAIN2_3" from "//localhost:11003"
 * to "//localhost:11004"*/
public static void main(String[] args) throws Exception {
    String accessPoint = "DOMAIN2_3";
    String oldNwaddr = "//localhost:11003";
    String newNwaddr = "//localhost:11004";
    String lAccessPoint = "DOMAIN1_1";

    TuxedoDomainGatewayTargetTestCustom
tuxedoDomainGatewayTargetTestCustom = new
TuxedoDomainGatewayTargetTestCustom();

tuxedoDomainGatewayTargetTestCustom.deleteTDomain(accessPoint,
                                                    oldNwaddr, lAccessPoint);

    tuxedoDomainGatewayTargetTestCustom.addTDomain(accessPoint,
newNwaddr,
                                                    lAccessPoint);
}
```

Note: \$TUXDIR/udataobj/jmx/tmjmx_tux.jar should be added to the classpath when compiling and running it.

Integration with Exalogic Elastic Cloud

Viewing Tuxedo Domains in Exalogic Elastic Cloud Targets


From Exalogic Elastic Cloud target home page, you can view details about the Oracle Tuxedo domains deployed on Exalogic Elastic Cloud hosts.

To go to the Exalogic Elastic Cloud target home page, follow these steps:

1. In Enterprise Manager, click **Targets > Systems** to go to the Systems page.
2. Filter the entries in the Search field by choosing Exalogic Elastic Cloud from the drop-down list and click the search button.
3. In the Search Results table, choose the Exalogic Elastic Cloud you want to view.

Enterprise Manager displays the Exalogic Elastic Cloud Home page where you can monitor the status of the Exalogic target and its components.

By default, Exalogic Elastic Cloud Home page Software tab does not display Tuxedo Domains. Follow these steps to add Tuxedo Domains manually:

1. In the Exalogic Elastic Cloud Home page, click **Personalize Page** () at the upper right.
2. Click **Add Content**.
3. Click **Add Tuxedo Domains in Exalogic System**.
4. Close the page to complete editing.

Exalogic Elastic Cloud Home page displays the Tuxedo Domain Summary panel.

On the Summary panel you can view a chart that shows the status of the Tuxedo Domains and displays the percentage of domains that are up and down. You can also view the status information along with alerts and policy violation and metric data for each Domain, as follows:

- Name: Tuxedo Domain target full name. Click the hyperlink to go to related Tuxedo Domain home page.
- Status: Tuxedo Domain target status.
- Member Status Summary: Total number of member status summary.
- Incidents: All incidents occurred in this Tuxedo Domain.
- Machines: All machines (hosts), on which Tuxedo domain is deployed in UBBCONFIG. The drop down list lists all machines belonging to the Tuxedo domain; If the machine is an exalogic compute node and deployed with Enterprise Manager agent, a hyperlink is available for going to the host home page.
- Tuxedo domain metric data:
 - Request Per Minute
 - Clients

- tpcall and tpacall Per Minute
- Native Clients
- Workstation Clients
- Jolt Clients



Columns of tpcall and tpacall Per Minute/ Native Clients/ Workstation Clients/ Jolt Clients are hidden by default.

Easy Configuration for Multiple Tuxedo Domains

You can easily configure multiple Oracle Tuxedo domains in Oracle Tuxedo Application Runtime environment by clicking the **Config** button on **Tuxedo Domains** panel.

You can dynamically change the parameters in `UBBCONFIG` for multiple Oracle Tuxedo domains by doing the following steps:

1. From the **Tuxedo Domains** panel, click **Config** -> **UBB Configure**.
2. In the Domain Parameter Configuration panel, specify the desired parameters and the domain to be applied.

There is a  button beside each parameter field. Click that button and then  to select the domain you want to apply the parameter, and click **Apply**.

You can set the following parameters in `UBBCONFIG`:

- RESOURCES section in `UBBCONFIG`
 - AUTOTRAN
 - TRANTIME
 - LDBAL
 - OPTIONS (only `NO_AA`, `EXT_AA`, `EXT_MON`, `ECID_CREATE`, and `ECID_USERLOG` can be set)
 - SCANUNIT
 - SANITYSCAN
 - DBBLWAIT
 - BBLQUERY
 - BLOCKTIME
 - DBBLFAILOVER

- SGRPFAILOVER

3. Click **Next**. The current domain parameter configuration is displayed.
4. Click **Submit**.

You can also dynamically change the following parameters in `DMCONFIG` for multiple Oracle Tuxedo domains by clicking **Config -> DM Configure** in the **Tuxedo Domains** panel:

- `DM_LOCAL` section in `DUBB`
 - `BLOCKTIME`
 - `CONNECTION_POLICY`
 - `RETRY_INTERVAL`
 - `DYNAMIC_RAP`
- `DM_TDOMAIN` section in `DUBB`
 - `CONNECTION_POLICY`
 - `RETRY_INTERVAL`
 - `TCPKEEPALIVE`
 - `DMKEEPALIVE`
 - `DMKEEPALIVEWAIT`

You can also reboot one or multiple Tuxedo domains running on one Exalogic RAC.

Easy Applying Oracle Tuxedo Patches

You can centrally manage the Oracle Tuxedo patch and easily apply an Oracle Tuxedo patch to existing Oracle Tuxedo systems for Batch by clicking **Tuxedo Patch** on the **Tuxedo Domain** panel.

You can centrally manage Oracle tuxedo patches for Linux `x86_64` and Solaris `SPARC_64` platforms:

Uploading a Tuxedo Patch

To upload a Tuxedo Patches, do the following steps:

1. From the **Tuxedo Domains** panel, click **Tuxedo Patch -> Manage Patch**.
2. In the Upload Patch area, specify the following:
 - The tuxedo version that patch can be applied. Currently only 12.1.1.0 is supported

- The tuxedo platform that patch can be applied. Currently only Linux x86_64 and Solaris SPARC 64 are supported
3. Select the Tuxedo patch on local file disk and upload to OMS soft library. The patch must meet the conditions that Tuxedo Version and Tuxedo Platform required.

When the patch is uploaded to the OMS soft library, it is shown in the table. You can select one patch and delete it from OMS soft library by clicking **Delete Patch**.

Distributing a Tuxedo Patch

To Distribute a Tuxedo Patch, do the following steps:

1. From the **Tuxedo Domains** panel, click **Tuxedo Patch -> Distribute Patch**.

The Distribute Patch page is displayed with all the uploaded Tuxedo patches and existing domains loaded in the page. Before applying the Tuxedo patch, you can remove the domains listed in the table by clicking **Remove** or reload the Tuxedo patches from OMS soft library and all the existing domains by clicking **Reset**.

2. From the **Select One Tuxedo Patch** list, select the Tuxedo patch to be applied to the existing tuxedo systems.
3. Click OK.

A new job is created to distribute and apply the selected package.

Notes:

- If none of the domain is valid and can be applied, the **OK** button is hidden.
- The host preferred credential must be set first before applying the patch.

Oracle Tuxedo Exalogic Compliance Standard

Oracle Tuxedo Exalogic Standard is integrated into Enterprise Manager Compliance Library to evaluate the compliance of Tuxedo Domains under Exalogic.

Oracle Tuxedo Exalogic Standard includes following rules:

- Oracle Tuxedo Exalogic Check
- Oracle Tuxedo Exalogic Direct Cross Node Communication Leveraging RDMA Check
- Oracle Tuxedo Exalogic Self-Tuning Lock Mechanism Check
- Oracle Tuxedo Exalogic SDP Support for MP Check

- Oracle Tuxedo Exalogic SDP Support for GWTDOMAIN Check
- Oracle Tuxedo Exalogic SDP Support for WSL Check
- Oracle Tuxedo Exalogic SDP Support for JSL Check
- Oracle Tuxedo Exalogic Shared Memory for Inter Process Communication Check
- Oracle Tuxedo Exalogic Read-only Optimization for XA Check

To view details of Oracle Tuxedo Exalogic Standard rules, follow these steps:

1. From the Enterprise menu, select **Compliance-> Library**.
2. Click the **Compliance Standards** tab.
3. Click **Oracle Tuxedo Exalogic Standard** in the Compliance Standard column.

The View Compliance Standard page appears. All the Oracle Tuxedo Exalogic Standard rules are listed on the right. Click each rule item to view the related properties.

Associating the Oracle Tuxedo Exalogic Standard with Targets

Before you can use the Oracle Tuxedo Exalogic Standard to check the Tuxedo domain compliance, you need to establish association between them using one of the following methods:

- Associate the Oracle Tuxedo Exalogic Standard with various Tuxedo domains:
 - a. From the Enterprise menu, select **Compliance -> Library**.
 - b. Click the **Compliance Standards** tab.
 - c. Highlight **Oracle Tuxedo Exalogic Standard**. Click the **Associate Target** button.
 - d. Select the targets you want to associate with this compliance standard. Click **OK**.
- Associate the individual Tuxedo domain with the Oracle Tuxedo Exalogic Standard:
 - a. Go to the home page of the Tuxedo domain you want to associate the compliance standard.
 - b. From the target menu, click **Compliance -> Standard Associations**.
 - c. Click **Edit Association Settings**.
 - d. Click **Add**.
 - e. Select **Oracle Tuxedo Exalogic Standard** and click **OK**.

Viewing Tuxedo Domain Compliance Evaluation Results

You can view the Tuxedo domain compliance evaluation results by way of the Cloud Control home page and individual target home pages.

To view results using Cloud Control home page, follow these steps:

1. From the Enterprise menu, select **Compliance -> Results**.
2. Highlight **Oracle Tuxedo Exalogic Standard**. Click **Show Details**.

To view compliance evaluation results from a target's home page, follow these steps:

1. Go to the home page of the Tuxedo domain you want to view compliance result.
2. From the target menu, click **Compliance -> Results**.
3. Highlight **Oracle Tuxedo Exalogic Standard**. Click **Show Details**.

Use the result page to get a comprehensive view about a target in regards to compliance over a period of time. Using the tables and graphs, you can easily watch for trends in progress and changes.

Business Transaction Management (BTM) Integration

This chapter contains the following topics:

- [Overview](#)
- [Prerequisite](#)
- [Enable BTM Monitoring from Tuxedo](#)
- [BTM Discovery](#)
- [Tuxedo Container in BTM](#)
- [Dependency](#)
- [BTM Transaction for Tuxedo Operation](#)
- [Call Path in TSAM Console](#)
- [Tuxedo Service in BTM](#)

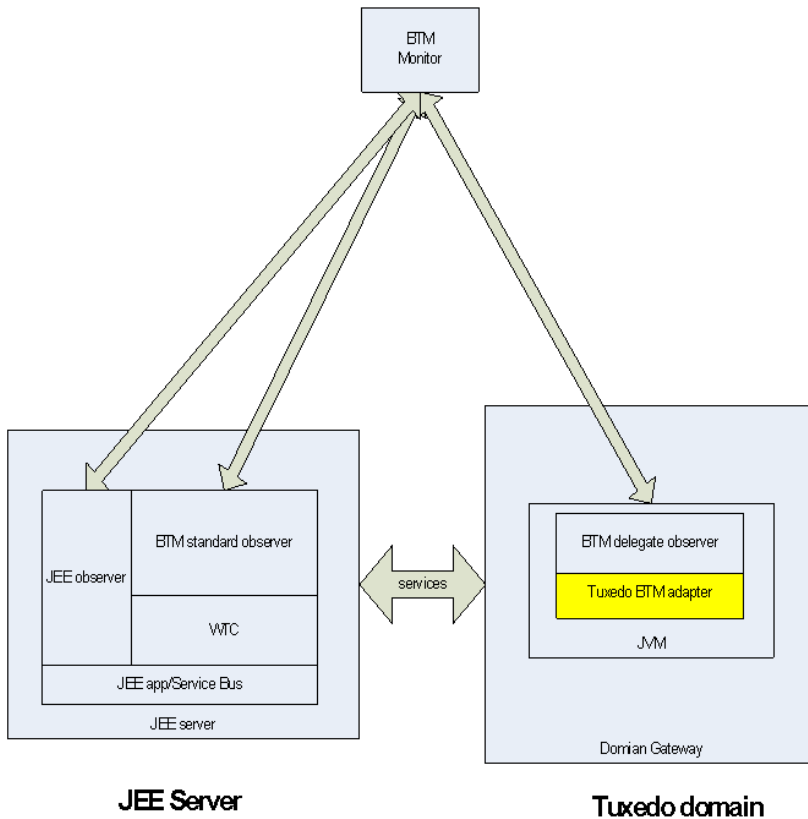
Overview

By integrating TSAM Plus and BTM, users are able to:

- Discover and track transactions across Tuxedo and more products monitored by BTM.
- Drill down into TSAM Plus call path detailed information for a transaction coming from WTC side using the transaction ECID from BTM console.

Figure 3-1 describes a typical scenario of a BTM transaction across the JEE server and Tuxedo domains.

Figure 3-1 Delegate Container



In this scenario, a Tuxedo BTM delegate observer runs in a JVM started in the /T domain gateway by JNI technology. When the /T domain gateway receives inbound/outbound messages, it sends BTM inbound/outbound requests to BTM delegate observer.

Prerequisite

The following prerequisites must be met before monitoring Tuxedo by BTM.

- BTM 12.1.0.6 is installed. For more information, see [Oracle® Business Transaction Management Installation Guide](#).
- The WTC and BTM JavaEE observer (12.1.0.6 or higher) are configured in Weblogic 12.1.3.0 or later.
- TSAM 12.1.3.0.0 or later is required for BTM monitoring.
- TSAM 12.1.3.0.0 agent is installed by enabling TSAM Plus during Tuxedo 12.1.3.0.0 installation.

Enable BTM Monitoring from Tuxedo

BTM Discovery

BTM discovery is entirely traffic-based. If messages are not flowing through the observed endpoints, Business Transaction Management cannot discover any application components nor can it discover the dependencies between these components.

Before discovery is performed, following steps must be done.

1. Install the observers in the environment you want to observe.

A delegate observer is included in `$TUXDIR/udataobj` after TSAM Plus installation. You do not need to install any observer for Tuxedo observe. For the WTC monitoring, you must install the JavaEE observer in Weblogic side. For more information, see [Installing Observers Overview](#).

2. From the **Observer Communication Policy** page in BTM Console, check the `DELEGATE` probe is enabled.
3. From the **Observer Communication Policy** page in BTM Console, choose **Use fully qualified name (FQN) for Observer Behavior -> Mapping Algorithm**.

Discovery involves the following steps:

1. Send traffic through your system.
2. Check the containers being observed. (Restarting the containers after installing Business Transaction Management is not enough to make them visible, you must send traffic to have them be seen.)
3. View services to see that all the services you are interested in observing have been discovered.

4. Check that measurements are being taken by looking at the **Summary** or **Analysis** tab for a service. Throughput, traffic, maximum response time, and average response time should be available for all the services involved in message traffic. If measurements are inaccurate or missing, it's possible that not enough time has elapsed for Business Transaction Management to calculate traffic measurements or perhaps monitoring was disabled.

Note: Make sure you have created "BTM transaction for Tuxedo operation" before viewing measurements.

5. Look at dependencies in the Service Map to see how traffic is flowing in your system.
6. Adjust as necessary. Generally speaking if the discovered picture does not meet your expectations, the first thing to do is run more traffic to make sure Business Transaction Management has had time to see all pieces of your system. If that does not resolve your problem, you might need to do one or more of the following:
 - Check the `-m` option for `GWTDOMAIN` is configured correctly, and "The BTM monitoring is enabled." is printed in `ULOG`.
 - Check the "DELEGATE" probe for the "Observer Communication Policy" is enabled.
 - Check if the Weblogic and Tuxedo version is supported by BTM Tuxedo monitoring.
 - Resolve discovery problems due to versioning policy or replication problem.

Tuxedo Container in BTM

When BTM monitoring is enabled, `GWTDOMIAN` starts an embedded JVM and runs a BTM delegate observer to monitor bidirectional calls between WTC and itself. The Type for this container is "DELEGATE" and the address for this container is always "delegate://<Tuxedo host address>:11/"

Figure 3-2 Delegate Container

Name	Address	Type	OS
bej301493:7001	http://bej301493.cn.oracle.com:7001/	WebLogic Server/12.1.2.0.0	Linux
slc05are:11	delegate://slc05are.us.oracle.com:11/	DELEGATE	Linux
slc05are:5139	tuxedo://slc05are.us.oracle.com:5139/	TUXEDO/12.1.3.0.0	Linux

Container: bej301493:7001

Profile Services

Container Details

Notes

Base Address http://bej301493.cn.oracle.com:7001/

All Tuxedo endpoints with the same base address belong to a same Tuxedo container. The address for this container is the endpoint's base address and the Type for this container is "TUXEDO/<Tuxedo version number>".

Figure 3-3 Tuxedo Container

Name	Address	Type	OS
bej301493:7001	http://bej301493.cn.oracle.com:7001/	WebLogic Server/12.1.2.0.0	Linux
slc05are:11	delegate://slc05are.us.oracle.com:11/	DELEGATE	Linux
slc05are:5139	tuxedo://slc05are.us.oracle.com:5139/	TUXEDO/12.1.3.0.0	Linux
TolowerPort	tuxedo://slc05are.us.oracle.com:5139/Tolower	TUXEDO	Linux
TOUPPERPort	tuxedo://slc05are.us.oracle.com:5139/TOUPPER	TUXEDO	Linux

Container: slc05are:5139

Profile Services

Container Details

Notes

Base Address tuxedo://slc05are.us.oracle.com:5139/

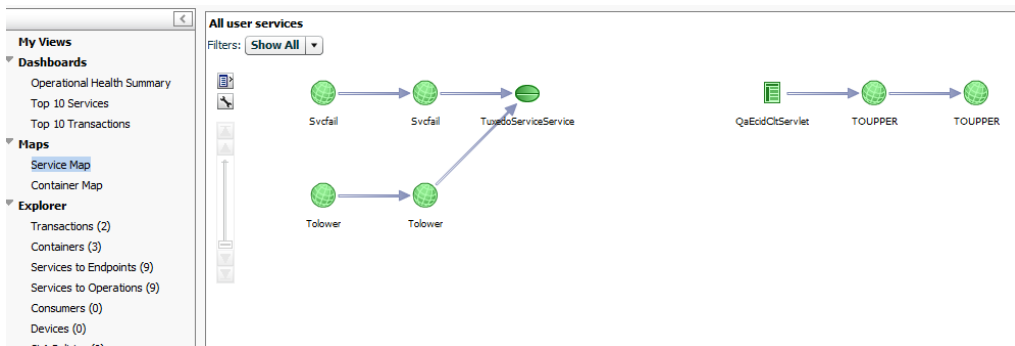
Aliases -

Dependency

For a service call through WTC and Tuxedo, no matter it is inbound or outbound call from Tuxedo, two services are defined in a BTM transaction to represent Tuxedo and WTC respectively.

The dependency between Tuxedo service and WTC service can be viewed in BTM console.

Figure 3-4 Service Map



BTM Transaction for Tuxedo Operation

By use of the ECID (Execution Context ID) propagation feature introduced in Weblogic and Tuxedo, you can create the transaction in BTM console using ECID as Instance's message key. With the ECID, you can query the call path inner Tuxedo domain in the TSAM management console when the policy is configured.

All measurements based on the transaction can be shown in BTM console.

Note: All BTM observers or BTM delegate observers (including the one embedded in GWTDOMAIN) must be located at machines with the same system time.

Following is the steps for creating a BTM Transaction based on Tuxedo operation.

1. From BTM console, click **Create -> Transaction** in the up-left corner.
2. In the popup window, choose the operations invoked in the transaction.
3. Click OK.

If the dependency is already discovered by BTM, the related operations are automatically added in the transaction. Otherwise, you need to manually add the operations you want to monitor in this transaction.

4. Select **Start message** and **End message**.
5. Click the **Logging** tab. Enable the instance and message logging. This step is required only if you want record every transaction instance.

Figure 3-5 Enable the Instance and Message Logging

<input checked="" type="checkbox"/>	Service	Operation
<input checked="" type="checkbox"/>	QaEcidCitServlet	-
<input checked="" type="checkbox"/>	TOUPPER	callTuxedo
<input checked="" type="checkbox"/>	TOUPPER	processWTCCall
<input type="checkbox"/>		
<input type="checkbox"/>		

6. Click the **Message Keys** tab.
7. Select requestECID as **Start key** and **End key** for all operations.
8. Type a transaction name.
9. Click **OK**.

Note: For the transaction initiated by Tuxedo, you must configure ECID_CREATE as OPTIONS in UBBCONFIG.

Call Path in TSAM Console

When TSAM Plus is registered correctly and LMS server is configured in the Tuxedo domain, each call from the remote WTC monitored by BTM automatically initializes a TSAM call path from GWTDOMAIN. You can get the call path information from TSAM Plus console.

Tuxedo Service in BTM

Table 3-1 lists the BTM property definitions for the Tuxedo service.

Table 3-1 BTM Property Definitions for Tuxedo Service

BTM Property	Tuxedo Property	
	Remote Service	Local Service
Service Namespace	http://oracle.tuxedo	
ServiceName	The Remote Name of WTC exported service	The Remote Name of WTC imported service
OperationName	processWTCCall	callWTC
Request Url	tuxedo://<remote NWADDR>/ServiceName	tuxedo://<The network address for WTC remote access point>/ServiceName
EndpointType	urn:com.amberpoint.servicegenres:TUXEDO	
ServiceType	urn:com.amberpoint.servicegenres:TUXEDO_SERVICE	
Protocol	ATMI	
ApplicationProtocol	SOAP	

[Table 3-2](#) lists the BTM property definitions for the WTC services.

Table 3-2 BTM Property Definitions for WTC Service

BTM Property	WTC Property	
	Exported Service	Imported Service
Service Namespace	http://oracle.weblogic.wtc	
ServiceName	Remote Name in WTC Exported Services	Remote Name in WTC Imported Services
OperationName	processTuxedoCall	callTuxedo

Table 3-2 BTM Property Definitions for WTC Service

Request Url	apwtc://weblogic address: LocalAccessPointPort/WTC_EXPO RT/ServiceName", for example, "apwtc://localhost:10086/WTC_EXP ORT/TOLOWER	apwtc://weblogic address:port/WTC_IMPORT/LocalAc cessPointId_RemoteAccessPointId/Ser viceName
EndpointType	urn:com.amberpoint.servicegenres: WTC_EXPORT	urn:com.amberpoint.servicegenres:WT C_IMPORT
ServiceType	urn:com.amberpoint.servicegenres:WTC_SERVICE	
Protocol	ATMI	AmberPointObservedMessage
ApplicationProtocol	SOAP	

Business Transaction Management (BTM) Integration

Oracle Tuxedo Scripting Tool

This chapter describes the Oracle Tuxedo Scripting Tool (TXST). It explains how you use the TXST command-line scripting interface to configure, manage, and persist change to Tuxedo running instances and monitor and manage server run-time events.

This chapter is organized as follows:

- [Using the Tuxedo Scripting Tool](#) describes how the scripting tool works, its modes of operation, and the basic steps for invoking it.
- [Navigating MBeans](#) describes how to retrieve Tuxedo domain configuration and run-time information, and edit configuration or custom MBeans.
- [Managing the Server Life Cycle](#) describes using TXST to start and stop Tuxedo Server instances and to monitor and manage the server life cycle.

Using the Tuxedo Scripting Tool

What is the Tuxedo Scripting Tool?

The Tuxedo Scripting Tool (TXST) is a command-line scripting environment that you can use to manage, and monitor Tuxedo domains. The TXST scripting environment is based on the Java scripting interpreter, Jython. In addition to supporting standard Jython features such as local variables, conditional variables, and flow control statements, TXST provides a set of scripting functions (commands) that are specific to Tuxedo. See <http://www.jython.org>.

How Does TXST Work?

You can use the scripting tool online (connected to a running Tuxedo domain or a JMX agent embedded in the tlisten process). For information on TXST commands, see [TXST Command and Variable Reference](#).

Online, TXST provides simplified access to Managed Beans (MBeans), Java objects that provide a management interface for an underlying resource that you can manage through JMX.

When TXST is connected to a Tuxedo domain, the scripting tool lets you navigate and interrogate MBeans, and supply configuration data to the instance. When TXST is connected to a JMX agent, its functionality is limited to commands which MBeans support.

Modes of Operation

TXST is a command-line interpreter that interprets commands either interactively, supplied one-at-a-time from a command prompt, or in batches, supplied in a file (script), or embedded in your Java code. The modes of operation represent methods for issuing TXST commands:

- Interactively, on the command line—Interactive Mode
- In a text file—Script Mode
- Embedded in Java code—Embedded Mode

Interactive Mode

Interactive mode, in which you enter a command and view the response at a command-line prompt, is useful for learning the tool, prototyping command syntax, and verifying configuration options before building a script. Using TXST interactively is particularly useful for getting immediate feedback after making a critical configuration change. The TXST scripting shell maintains a persistent connection with an instance of Tuxedo domain. Because a persistent connection is maintained throughout the user session, you can capture multiple steps that are performed against the instance. For more information, see [Recording User Interactions](#).

Script Mode

Scripts invoke a sequence of TXST commands without requiring your input, much like a shell script. Scripts contain TXST commands in a text file with a .py file extension, for example, filename.py. You use script files with three modes (Invoking TXST with the script, using Jython commands for running scripts and invoking the script with Ant build file). See [Running Scripts](#).

Embedded Mode

In embedded mode, you instantiate an instance of the TXST interpreter in your Java code and use it to run TXST commands and scripts. Except for `startRecording()` and `stopRecording()`, all TXST commands and variables that you use in interactive and script mode can be run in embedded mode.

Listing 1 illustrates how to instantiate an instance of the TXST interpreter and use it to connect to a running domain and create a server.

Listing 4-1 Running TXST From a Java Class

```
import java.util.*;
import oracle.tuxedo.jmx.tux.jyscript.TXSTInterpreter;
import org.python.util.InteractiveInterpreter;

public class EmbeddedTXST {
    static InteractiveInterpreter interp = null;
    EmbeddedTXST() {
        interp = new TXSTInterpreter();
    }

    private static void connect() {
        interp.exec("connect('///localhost:5037', 'simpapp', '38075')");
    }

    private static void create_server() {
        StringBuffer buffer = new StringBuffer();
        buffer.append("cd('simple/GROUP1')\n");
        buffer.append("srv1= create('simplserv', 'SERVER', srvID=40)\n");
        buffer.append("print 'Script ran successfully...'\n");
    }
}
```

```
        interp.exec(buffer.toString());
    }

    public static void main(String[] args){
        new EmbeddedTXST();

        connect();
        create_server();
    }
}
```

Main Steps for Using TXST

The following sections summarize the steps for setting up and using TXST:

- [Setting Up Your Environment](#)
- [Invoking TXST](#)
- [Requirements for Entering TXST Commands](#)
- [Running Scripts](#)
- [Exiting TXST](#)
- [Getting Help](#)
- [Recording User Interactions](#)
- [Redirecting TXST Output to a File](#)

Setting Up Your Environment

To set up your environment for TXST:

1. Install and configure the TSAM agent.

2. Set TUXDIR environment variable as Tuxedo install directory and \$TUXDIR/bin to the PATH environment variable.
3. Include
\$TUXDIR/jmx/jython-standalone.jar:\$TUXDIR/jmx/tmjmx_tux.jar:\$TUXDIR/jmx/tmjmx_metadata.jar in the CLASSPATH. Or you can invoke TXST just using txst.sh or txst.cmd which we provided.

Invoking TXST

To invoke TXST, enter the following command: **txst.sh**

A welcome message and the TXST prompt appears:

```
txst: /offline>
```

txst.sh is placed in \$TUXDIR/bin. If you invoke TXST with txst.sh, you do not need to set up the CLASSPATH.

If you added the related jar files into CLASSPATH as we mentioned before, you can enter the command: java oracle.tuxedo.TXST. For instance,

```
#export
CLASSPATH=$CLASSPATH:$TUXDIR/jmx/jython-standalone.jar:$TUXDIR/jmx/tmjmx_tux.jar:$TUXDIR/jmx/tmjmx_metadata.jar
# java oracle.tuxedo.TXST

Initializing Tuxedo Scripting Tool (TXST) ...
```

Note: If using dmib related commands, you should use the package
\$TUXDIR/udataobj/tuxj/com.bea.core.jatmi_2.0.0.0.jar

Jython scans all the jar files it can find at first startup. Depending on the system, this process may take a few minutes to complete, and TXST may not return a prompt right away.

```
Welcome to Tuxedo Administration Scripting Tool
```

```
Type help() for help on available commands
```

```
txst:/offline>
```

Requirements for Entering TXST Commands

Follow these rules when entering TXST commands. Also see [TXST Command and Variable Reference](#).

- Command names and arguments are case sensitive.

- Enclose arguments in single or double quotes. For example, 'newServer' or "newServer".
- If you specify a backslash character (\) in a string, for example when specifying a file pathname, you should precede the quoted string by r to instruct TXST to interpret the string in its raw form and ensure that the backslash is taken literally. This format represents standard Python syntax.
- Display help information for TXST commands by printing the command's documentation string. See Getting Help.

Running Scripts

TXST incorporates three ways for running scripts. To run the script examples in this guide, copy and save the commands in a text file with a .py file extension, for example, filename.py. Use the text file with the commands for running scripts that are listed below. There are sample scripts that you can use as a template when creating a script.py file from scratch. If the script will connect TXST to a running Tuxedo domain, start Tuxedo domain before running the script.

You can run the script in three modes:

- Invoke TXST and run a script:

```
java oracle.tuxedo.TXST myscript.py
```

- run a script from TXST:

```
execfile('myscript.py')
```

Listing 4-2 Running script myscript.py

```
connect("//localhost:5037", "simpapp", "38075");  
cd('simple/GROUP1')  
ls('c')  
disconnect()  
exit()
```

- Invoke the TXST script from an Ant build file

Scripts invoke a sequence of TXST commands without requiring your input, much like a shell script. Scripts contain TXST commands in a text file with a .py file extension.

TXST provides a custom ant task to invoke the TXST script from an Ant build file. If you want to use the task with your own Ant installation, add the following task definition in your build file:

```
<taskdef name="txst"
classname="oracle.tuxedo.jmx.tux.jyscript.TXSTTask" />
```

To run TXST from an Ant script:

- a. Set your environment in a command shell, which include TUXDIR, CLASSPATH, JAVA_HOME, ANT_HOME and so on.
- b. Add a call to the txst Ant task to execute TXST commands. For example:

```
<target name="GetParameters">
    <txst debug="false" failOnError="false"
        executeScriptBeforeFile="true" fileName="./test.py">
        <script>
            connect ("//slce04cn03:5067", "simpapp", "38175")
            ls()
            print cmo.GetParameters()
        </script>
    </txst>
</target>
```

Using the txst Ant task, you can specify TXST commands within a TXST script(.py), using the fileName attribute, or embed TXST script commands inside the build file, within the <script> tags.

- c. Execute the Ant task or tasks specified in the build.xml file by typing ant in the staging directory, optionally passing the command a target argument:

```
bash-3.2$ ant getParameters
```

the syntax of the txst Ant task is as follows:

```
<txst properties="propsFile" filename="filename" arguments="arglist"
    failOnError="value" executeScriptBeforeFile="value" debug="value" >
<script>
    txst-commands
```

</script>

The following table defines the Ant task attributes that you can specify as part of the txst Ant task.

Attribute	Definition
properties="propsFile"	Optional. Name and location of the properties file that contains name-value pairs that you can reference in your script.
fileName=" fileName "	Optional. Name and location of the TXST script file that you would like to execute. If the specified TXST script file does not exist, the txst Ant task fails.
arguments="arglist "	Optional. List of arguments to pass to the script. These arguments are accessible using the sys.argv variable.
failOnError="value"	Optional. Boolean value specifying whether the Ant build script will fail if the txst Ant task fails. This attribute defaults to true.
executeScriptBeforeFile="value"	Optional. Boolean value specifying whether the embedded script is executed before the specified TXST script file. This attribute defaults to true, specifying that the embedded script is executed first.
debug="value"	Optional. Boolean value specifying whether debug statements should be output when the TXST Ant Task is executed. This attribute defaults to false.

In the following example, test.py is the file in which there are TXST commands. You also can add TXST command in the “script” node:

```
<project default="all" basedir=".">
    <taskdef name ="txst"
        classname="oracle.tuxedo.jmx.tux.jyscript.TXSTTask" />
    <target name="getParameters">
        <txst debug="false" failOnError="false"
            executeScriptBeforeFile="true" fileName="./test.py">
            <script>
                connect ("//slce04cn03:5067", "simpapp", "38175")
                cmo.GetParameters ()
            </script>
        </txst>
    </target>
</project>
```

```

        </script>
    </txst>
</target>
</project>

```

TXST does not support Maven in this version.

Exiting TXST

To exit TXST:

```
txst:simpapp_38075:/>exit()
```

```
c:\jython\bin>
```

Getting Help

To display information about TXST commands, print the command's documentation string.

For example, to display information about the disconnect command, enter the following command:

```
txst:/simpapp>help('disconnect')
```

The command returns the following:

```
    This function is used to disconnect from tuxedo server instance.
```

```
    Arguments list:
```

```
    ()
```

```
txst: /simpapp > disconnect()
```

Recording User Interactions

To start and stop the recording of all TXST command input, enter:

```
startRecording(outputFile)
```

```
stopRecording()
```

You must specify the file pathname for storing TXST commands when you enter the startRecording command.

For example, to record TXST commands in the record.py file, enter the following command:

```
txst:/simpapp> startRecording('record.py')
```

For more information, see `startRecording` and `stopRecording`.

Redirecting TXST Output to a File

To start and stop redirecting TXST output to a file, enter:

```
redirect(outputFile, [toStdOut])
```

```
stopRedirect()
```

You must specify the pathname of the file to which you want to redirect TXST output. You can also optionally specify whether you want TXST output to be sent to stdout; the `toStdOut` argument defaults to `True`.

For example, to redirect TXST output to the `logs/txst.log` file in the current directory and disable output from being sent to stdout, enter the following command:

```
txst:/simpapp> redirect('./logs/txst.log', False)
```

For more information, see `redirect` and `stopRedirect`.

Navigating MBeans

Navigating and Interrogating Mbeans

TXST online provides simplified access to MBeans. While JMX APIs require you to use JMX object names to interrogate MBeans, TXST enables you to navigate a hierarchy of MBeans in a similar fashion to navigating a hierarchy of files in a file system.

Tuxedo organizes its MBeans in a hierarchical data model. In the TXST file system, MBean hierarchies correspond to drives; MBean instances are directories; MBean attributes and operations are files. TXST traverses the hierarchical structure of MBeans using commands such as `cd`, `ls`, and `pwd` in a similar way that you would navigate a file system in a UNIX or Windows command shell. After navigating to an MBean instance, you interact with the MBean using TXST commands.

In the hierarchy, the root directory is `DomainMBean`; each instance of the MBean type is a subdirectory; and MBean attributes and parameters are nodes (like files) under the MBean instance directory.

Domain MBean

 Machine MBean

 Bridge MBean

Group MBean

Server MBean

ARTBatchSystem Mbean

Changing the Current Management Object

When TXST first connects to an instance of Tuxedo domain, the variable, `cmo` (Current Management Object), is initialized to the root of all configuration management objects, `DomainMBean`. When you navigate to an MBean type, the value of `cmo` reflects the parent MBean. When you navigate to an MBean instance, TXST changes the value of `cmo` to be the current MBean instance, as shown in Listing 3.

For more information on TXST variables, see [TXST Command and Variable Reference](#).

Listing 4-3 Changing the Current Management Object

```
txst:/offline>connect("//slce04cn03:5035", "simpapp", "58103")
txst:simpapp_58103:/>cmo.getMbeanName()
simpapp_58103:type=tuxedo_domain
txst:simpapp_58103:/>cd('L1')
txst:simpapp_58103:/L1>cmo.getMbeanName()
simpapp_58103:type=tuxedo_machine,LMID=L1
```

Navigating and Displaying Configuration MBeans Example

The commands in Listing 4 instruct TXST to connect to a Tuxedo domain instance, navigate, and display MBeans in `DomainMBean`. If no argument is specified, the `ls` command lists all the child MBeans and attributes.

Listing 4-4 Navigating and Displaying Configuration MBeans

```
txst:/offline>connect("//slce04cn03:5035", "simpapp", "58103")
txst:simpapp_58103:/>ls()
```

Oracle Tuxedo Scripting Tool

attributes:

State ACTIVE

parameters:

TA_AUTHSVC	
TA_AUTOTRAN	N
TA_BBLQUERY	30
TA_BLOCKTIME	6
TA_CLASS	T_DOMAIN
TA_CMTRET	COMPLETE
TA_COMPONENTS	
TA_CURDRT	0
TA_CURGROUPS	1
TA_CURINTERFACES	0
TA_CURMACHINES	0
TA_CURQUEUES	2
TA_CURRFT	0
TA_CURRTDATA	0
TA_CURSERVERS	2
TA_CURSERVICES	9
TA_CURSTYPE	1
TA_CURTYPE	12
TA_DBBLFAILOVER	0
TA_DBBLWAIT	2
TA_DOMAINID	simpapp
TA_ENCRYPTION_REQUIRED	N
TA_ERROR	0
TA_GID	8500

TA_HWDRT	0
TA_HWGROUPS	1
TA_HWINTERFACES	0
TA_HWMACHINES	1
TA_HWQUEUES	2
TA_HWRFT	0
TA_HWRTDATA	0
TA_HWSERVERS	2
TA_HWSERVICES	9
TA_INSTANCE_AFFINITY	NONE
TA_IPCKEY	58103
TA_LDBAL	N
TA_LICEXPIRE	
TA_LICMAXUSERS	10000000
TA_LICSERIAL	
TA_MASTER	L1
TA_MAXACCESSERS	15
TA_MAXACLGROUPTS	16384
TA_MAXBUFSTYPE	32
TA_MAXBUFTYPE	16
TA_MAXCONV	1
TA_MAXDRT	0
TA_MAXGROUPS	100
TA_MAXGTT	100
TA_MAXINTERFACES	150
TA_MAXMACHINES	256
TA_MAXNETGROUPS	8

Oracle Tuxedo Scripting Tool

TA_MAXOBJECTS	1000
TA_MAXQUEUES	20
TA_MAXRFT	0
TA_MAXRTDATA	8
TA_MAXSERVERS	20
TA_MAXSERVICES	50
TA_MAXSPDATA	18504
TA_MAXTRANTIME	0
TA_MIBMASK	0
TA_MODEL	SHM
TA_MORE	0
TA_NOTIFY	SIGNAL
TA_OCCURS	1
TA_OPTIONS	
TA_PERM	384
TA_PREFERENCES	
TA_SANITYSCAN	12
TA_SCANUNIT	10
TA_SCANUNIT_EXT	10
TA_SECURITY	NONE
TA_SEC_PRINCIPAL_LOCATION	
TA_SEC_PRINCIPAL_NAME	
TA_SEC_PRINCIPAL_PASSVAR	
TA_SGRPFALLOVER	0
TA_SIGNATURE_AHEAD	3600
TA_SIGNATURE_BEHIND	604800
TA_SIGNATURE_REQUIRED	N

```

    TA_SSL_RENEGOTIATION          0
    TA_STATE                      ACTIVE
    TA_SYSTEM_ACCESS             FASTPATH
    TA_TRANTIME                  30
    TA_UID                       532
    TA_USIGNAL                   SIGUSR2

childBean:
    tuxedo_machine                L1

additional info:

```

Browsing Runtime MBeans

Lists all the children beans and/or attributes for the current bean.

You can optionally control the output by specifying an argument. If no argument is specified, the command lists all children beans and attributes in the domain. The output is returned as a string.

Finding MBeans

To locate a particular MBean, you can use the find command. TXST returns the pathname to the specified MBean. You can use the getMBean command to return the MBean specified by the path. For more information, see find and getMBean.

For example:

```

txst:simpapp_58103: />find('ser*')

/L1/GROUP1/server_1

txst:simpapp_58103: />bean = getMBean('/L1/GROUP1/server_1')

txst:simpapp_58103: />path = getPath(bean)

txst:simpapp_58103: />print path

/L1/GROUP1/server_1

```

Managing the Server Life Cycle

During its lifetime, a running instance can transition through a number of operational states, such as shutdown, starting, suspended, resuming, and running. TXST commands such as start, suspend, resume, and shutdown cause specific changes to the state of a server instance.

Using TXST, you can:

- Start and shutdown a Tuxedo domain, machine, group or server. See Starting and Stopping.
- Migrate the servers in the specified scope to an alternate location.
- Suspend and resume a specified target. See Suspending and Resuming.

Starting and Stopping

TXST provides several ways to start and stop a specified object. The method that you choose depends on the target object.

Domain

To start a Tuxedo domain, you should invoke `startDomain()` command. This command must use while TXST is connecting to JMX agent embedded in the `tlisten` process. If invoking is success, you can connect to Tuxedo domain. See `startDomain`.

To stop a Tuxedo domain, you should invoke `shutdownDomain()` command. If domain is shutdown successfully, TXST will disconnect from domain and go back to connect JMX agent. See `shutdownDomain`.

Machine, Group, and Server

For machine, group and server, you can invoke `start()` or `shutdown()` directly. If your target is server, there are two ways to operate these command. See `start` and `shutdown`.

Suspending and Resuming

The following MBeans are supported by suspending and resuming: Machine, Server, Bridge, WSH, JSH. However, the target object is different from the Mbeans. For more details about targets, refer to following table.

Mbean	Target
Machine	Client
Server	Service
Bridge	Remote bridge
WSH	WSH
JSH	JSH

TSAM Plus Accounting and Chargeback

This chapter contains the following topics:

- [Accounting Information Output](#)
- [Accounting Information Analyzing](#)
- [Known Issues](#)

Accounting Information Output

Prerequisite

Before you can use this feature, make sure TSAM Plus agent is installed and TSAM Plus plug-in is enabled.

Configuration

A new option, `-a`, is added to [servopts \(5\)](#) for providing more various accounting information output fields:

```
-a FILENAME [ [$FIELD] ... ]
```

If `-a` option is set, the Tuxedo server prints accounting information into the file specified by `FILENAME`.

The accounting file is generated daily with a file name in the format of `[FILENAME].mmddyy`.

The fields that can be specified for `-a` option are as follows:

- **DID:** Indicates the Tuxedo Domain ID of the server process.
- **LMID:** Indicates the LMID of the server process.
- **GRPID:** Indicates the Tuxedo Group ID of the server process.
- **GRPNAME:** Indicates the Tuxedo Group name of the server process.
- **SID:** Indicates the Tuxedo Server ID of the server process.
- **PID:** Indicates the process ID of the Tuxedo server process.
- **PNAME:** Indicates the process name of the Tuxedo server process.
- **SVC:** Indicates the name of the service being invoked. It always appears in the output fields when “-a” is specified.
- **GID:** Indicates the Tuxedo user group invoking the service.
- **UID:** Indicates the Tuxedo user ID invoking the service.
- **CLIENT:** Indicates the Tuxedo client invoking the service.
- **STIME:** Indicates the time entering the service routing. The output of this field is in the format of “tv_sec/tv_usec”, where two values are expressed as seconds and microseconds since 00:00 Coordinated Universal Time (UTC), January 1, 1970 respectively. It always appears in the output fields when “-a” is specified.
- **ETIME:** Indicates the time leaving the service routing. The output of this field is in the format of “tv_sec/tv_usec”, where two values are expressed as seconds and microseconds since 00:00 Coordinated Universal Time (UTC), January 1, 1970 respectively. It always appears in the output fields when “-a” is specified.
- **TIME:** Indicates the total CPU time (in microsecond) consumed by the service call.
- **SYSTIME:** Indicates the System CPU time (in microsecond) consumed by the service call.
- **USRTIME:** Indicates the User CPU time (in microsecond) consumed by the service call.
- **ERRNO:** Indicates the ERRNO returned to the service invoker.
- **URCODE:** Indicates the URCODE returned to the service invoker.
- **SIZE:** Indicates the size of the request message (in byte).
- **WTIME:** Indicates the waiting time of the request message in a queue (in microsecond).

- ALL: Indicates all fields above.

If no field is specified for “-a” option, these fields are included in sequence by default: UID, SVC, STIME, ETIME, SYSTIME, USRTIME.

If a duplicate field name is specified, it is ignored and a warning message is printed in ULOG: “ERROR: duplicate field xxx for -a option”.

If an invalid field is specified, it is ignored and an error message is printed in ULOG “ERROR: xxx is not a valid field for -a option”.

The fields separated by TAB character are listed in the first line of a new accounting file. Following is a sample:

```
#UID      SVC      STIME      ETIME      SYSTIME  USRTIME
```

Subsequently a line is printed for every service call in the same sequence of the field names and values are separated by a TAB character as well. If a field value is not available, it will be left empty.

Note:

- A Log file cannot be re-created if it has been deleted when Tuxedo is running.
- If multiple server instances output accounting information to the same file, the same fields in the same sequence must be specified for all relevant servers, otherwise *accrpt* cannot work.
- The services in system server, such as BBL, GWTDOMAIN, TMS, etc. or the system services with the name starting with '.' cannot generate accounting records. The following Oracle Tuxedo Application Runtime for CICS servers are supported:
 - ARTATR1
 - ARTATRN
 - ARTCNX
 - ARTDPL
 - ARTSTR1
 - ARTSTRN
 - ARTTDQ
 - ARTTSQ
- A multithread server (built with “-t” option) reports no TIME, SYSTIME or USRTIME.

- A server with the “NONE” or “APP_PW” value of the SECURITY parameter in TUXCONFIG file reports no GID, UID or CLIENT.

Samples

```
CLOPT="-A -a account_file$SVC$GID$UID$TIME$SIZE --"
```

```
CLOPT="-A -a account_file$ALL --"
```

```
CLOPT="-A -a account_file --"
```

Accounting Information Analyzing

You can use a new tool `accrpt` to analyze and aggregate the content of accounting file.

If an accounting file is specified, `accrpt` analyzes and aggregates the file content and print the aggregated information into `STDOUT`.

```
accrpt [-k FIELD[,FIELD]...] [-t  
{MINUTE|5MINUTES|10MINUTES|15MINUTES|HOURL|DAY}] [-o FIELD[,FIELD]...] [-i]  
[FILE(s)]
```

Configuration

-k option

It indicates the key fields separated by ‘,’. The following fields can be specified:

- DID: Indicates the Tuxedo Domain ID of the server process.
- LMID: Indicates the LMID of the server process.
- GRPID: Indicates the Tuxedo Group ID of the server process.
- GRPNAME: Indicates the Tuxedo Group name of the server process.
- SID: Indicates the Tuxedo Server ID of the server process.
- PID: Indicates the process ID of the Tuxedo server process.
- PNAME: Indicates the process name of the Tuxedo server process.
- SVC: Indicates the name of the service being invoked.
- GID: Indicates the Tuxedo user group invoking the service.

- **UID:** Indicates the ID of the Tuxedo user invoking the service.
- **CLIENT:** Indicates the Tuxedo client invoking the service.

If this option is not specified, the `SVC` field is specified by default. If a field is specified but its value is not available, the field will be left empty in the output result.

-t option

It indicates the time window to aggregate the accounting information. If it is not specified, the accounting information is aggregated in hour by default.

-o option

It indicates the measurement fields that are outputted for accounting and chargeback purposes. Fields are separated by ‘,’. The following fields can be specified in the list:

- **TIMES:** Indicates the total invoke times.
- **SUCC:** Indicates the invoke times for returning success.
- **APPFALL:** Indicates the invoke times for returning application failure.
- **SYSFALL:** Indicates the invoke times for returning system failure.
- **ELPTIME:** Indicates the total elapse time (in millisecond).
- **TIME:** Indicates the total CPU time (in millisecond).
- **SYSTIME:** Indicates the total System CPU time (in millisecond).
- **USRTIME:** Indicates the total User CPU time (in millisecond).
- **SIZE:** Indicates the total message size (in kilobytes) of the request message.
- **WTIME:** Indicates the average waiting time of the request message in a queue (in microsecond).

By default, fields `TIMES`, `ELPTIME`, `SYSTIME`, and `USRTIME` are selected as measurement fields.

-i option

If this option is specified, `acctpt` ignores the line of accounting information where any empty key field is included.

File(s)

Specifies accounting files for analyzing. If no file is specified, `accrpt` reads input from `STDIN`.

Output

`accrpt` analyzes raw accounting information and outputs the aggregated accounting information into `STDOUT`. Three groups of fields are outputted in sequence:

- **Implicit fields:** include `STIME` and `ETIME` of the time window. They are always printed as the first and second fields.
- **Key field:** All key fields specified by `-k` option.
- **Measurement fields:** all fields specified by `-o` option.

`accrpt` prints all fields in one line prefixed `#` character firstly and all accounting information subsequently, sorted by all Key fields, `STIME`, and `ETIME` in ascending order. In a line, all fields are separated by TAB character. If a value is not available, it is left empty.

Samples

Sample 1 - Default Usage

For `SimpleServer`, here we set its `CLOPT` parameter as “`-A -a simpserv_acct --`”

Every day this server outputs its accounting information into a new file named `simpserv_acct.mmddyy`.

```
>cat simpserv_acct.022102
```

```
# SVC  UID STIME      ETIME  SYSTEME      USRTIME
TOUPPER      John  1789249093/34452  1789249093/65389 100    50
.....
```

```
>accrpt  account3. 022102
```

```
#STIME      ETIME  SVC    TIMES  ELPTIME      SYSTEME      USRTIME
00:00:00    01:00:00  TOLOWER  10    100    20    40
01:00:00    02:00:00  TOLOWER  15    200    30    50
00:00:00    01:00:00  TOUPPER  10    100    20    40
01:00:00    02:00:00  TOUPPER  10    100    20    40
```

.....

Sample 2

```
>accrpt -t DAY account.022012
```

#STIME	ETIME	SVC	TIMES	ELPTIME	SYSTIME	USRTIME
00:00:00	24:00:00		TOLOWER	150	2000 300	500
00:00:00	24:00:00		TOUPPER	100	1000 200	400

Sample 3

```
>accrpt -k SVC-o TIMES,ELPTIME,TIME,SIZE account1.022012 account3.022012
```

#STIME	ETIME	SVC	TIMES	ELPTIME	TIME	SIZE
00:00:00	01:00:00		TOLOWER	10	100	20 100
01:00:00	02:00:00		TOLOWER	15	200	30 500
00:00:00	01:00:00		TOUPPER	10	100	20 100
01:00:00	02:00:00		TOUPPER	15	200	30 500

.....

Known Issues

- The user ID of an ART service "connect" call is the domain ID. This works by design.
- The ART service "disconnect" cannot be accounted.

TSAM Plus Accounting and Chargeback

User Payload Collection

This chapter contains the following topics:

- [Overview](#)
- [Supported User Payload Buffer Types](#)
- [Determine User Payload Storage](#)
- [Configuring User Payload Collection](#)
- [Exporting Collected User Payload Data](#)

Overview

Oracle Tuxedo applications send and receive their data in typed buffers. The input message and output message of a Tuxedo service are described in Tuxedo buffer types. TSAM Plus can collect the user payload encapsulated in the Tuxedo service input and output buffer, store the collected payload in Oracle Database, Hadoop or local file, and provide a series of APIs and tools to export the payload for further aggregation and analysis.

Supported User Payload Buffer Types

[Table 6-1](#) shows the supported Oracle Tuxedo buffer types for user payload.

Table 6-1 supported Oracle Tuxedo Buffer Types in User Payload

Type Name	Description
CARRAY	Character array (possibly containing NULL characters) that is neither encoded nor decoded during transmission
X_OCTET	Equivalent to CARRAY; provided for XATMI compatibility
STRING	NULL-terminated character array
FML	FML fielded buffer
VIEW	C structure or FML view
X_C_TYPE	Equivalent to VIEW; provided for XATMI compatibility
X_COMMON	Equivalent to VIEW; provided for XATMI compatibility
FML32	FML32 fielded buffer, using 32-bit identifiers and offsets
VIEW32	C structure or FML32 view, using 32-bit identifiers, counter variables, and size variables
XML	Buffer for XML documents

Determine User Payload Storage

TSAM Plus supports different kinds of storage for the collected user payload, which include Oracle Database, Hadoop, and local file.

User Payload Record in Oracle Database

User payload data is stored in the table `MON_PAYLOAD` in TSAM plus Database, as listed in [Table 6-2](#).

Table 6-2 User Payload Data in MON_PAYLOAD

COLUMN	TYPE	Nullable	Constraints	Since Version	Description
ID	NUMBER	NO		12.2.2	auto-generated ID
DOMAINID	VARCHAR2 (30)	YES		12.2.2	Domain ID, it is null when no DOMAINID is set in UBBCONFIG
LMID	VARCHAR2 (30)	YES		12.2.2	Machine LMID
GROUPID	SMALLINT	YES		12.2.2	Tuxedo group number
SERVERID	SMALLINT	YES		12.2.2	Server ID
SVCNAME	VARCHAR2 (127)	YES		12.2.2	Service name
CORRELATIONID	VARCHAR2 (255)	YES		12.2.2	Call path CORRELATIONID
ECID	VARCHAR2 (1024)	YES		12.2.2	Tuxedo call ECID
TPERRNO	SMALLINT	YES		12.2.2	tperrno

Table 6-2 User Payload Data in MON_PAYLOAD

COLUMN	TYPE	Nullable	Constraints	Since Version	Description
VALUETYPEINDEX	SMALLINT	YES		12.2.2	Tuxedo buffer type index, defined in <code>weblogic.wtc.jatmi.StandardTypes</code> CARRAY : 16 STRING : 17 FML : 18 VIEW : 19 X_OCTET : 20 X_C_TYPE : 21 X_COMMON : 22 FML32 : 23 VIEW32 : 24 XML : 25
VALUESUBTYPE	VARCHAR2 (127)	YES		12.2.2	Tuxedo sub buffer type. It is the view or view32 name
PAYLOADTYPE	SMALLINT	YES		12.2.2	1 means request; 2 means reply
VALUE	BLOB	YES		12.2.2	User payload value
LOGTIME	TIMESTAMP	YES		12.2.2	The timestamp when this record is reported from TSAM Plus Agent.

User Payload Record in Hadoop

User payload data can be stored in HDFS. It requires JRE 1.7 or above.

To store the payload into the Hadoop file system, you must set the environments `JAVA_HOME` and library path before starting LMS. Additionally, the `LMS -T` option must not be set or set to 1.

For Linux systems, the library path `LD_LIBRARY_PATH` must include the `libjvm.so`

Following is an example for Linux:

```
export JAVA_HOME=/home/user1/jdk1.7.0_21
export HADOOP_HOME=/home/user1/hadoop-2.6.0
export LD_LIBRARY_PATH=$JAVA_HOME/jre/lib/amd64/server/:$LD_LIBRARY_PATH
```

For AIX systems, set `LIBPATH` instead of `LD_LIBRARY_PATH`.

For example:

```
export
LIBPATH=/usr/$JAVA_HOME/jre/lib/ppc64/classic:$JAVA_HOME/jre/libppc64:$JAV
A_HOME/jre/lib/ppc64/j9vm
```

File Naming

If you choose to output the payload data to Hadoop file system, TSAM Plus generates a payload file to store the payload key and data. Each machine writes a distinct file every day.

The format is `<domain>_<machine>_<yyyymmdd>.payload`. The new hadoop file is produced every day and there is no file size limit.

Content Format

TSAM Plus uses the Hadoop SequenceFile format (Hadoop 2.6 ver=6) to store the payload in HDFS. Each payload record includes the key and value data. The record (`org.apache.hadoop.io.BytesWritable`) can be parsed using the Java class `com.oracle.tsam.payload.PayloadRecord`.

The record key class is `org.apache.hadoop.io.LongWritable` and the value class is `org.apache.hadoop.io.BytesWritable`. Both are stored as the record.

Following is the payload SequenceFile format:

- Header
 - version - 3 bytes of the magic header SEQ, followed by 1 byte of actual version number (e.g. SEQ4 or SEQ6)
 - keyClassName - Key class
 - valueClassName - Value class

- `compression` - A boolean that specifies if compression is turned on for keys/values in this file. 0 is used for payload file.
- `blockCompression` - A boolean that specifies if block compression is turned on for keys/values in this file. 0 is used for payload file.
- `metadata` - `SequenceFile.Metadata` for this file. 0x0, 0x0, 0x0, 0x0
- `sync` - A sync marker that denotes the end of the header.
- **Record**
 - Record length
 - Key length
 - Key
 - Value length
 - Value - Payload record that includes the payload key and value
- A sync-marker.

User Payload Record in Local File

The naming and content format of the user payload record in local file are same as rules on Hadoop.

You can specify the local file directory where the payload data will be stored using the LMS server `CLOPT` parameter in `UBBCONFIG`. For example, to store the payload data in the file at `/u01/common/patches/oracle/payload`, configure the following in the `UBBCONFIG` file:

```
LMS    SRVGRP=LMSGRP SRVID=2 CLOPT="-A -- -l <tsam_manager_host>:7001/tsam  
-p /u01/common/patches/oracle/payload"
```

For more information, refer to [Local Monitor Server](#) in Oracle TSAM Plus Reference Guide.

Specifying User Payload Storage from TSAM Plus Console

To specify the storage method for the user payload data, do the following:

1. From the TSAM Plus console top menu bar, click **Management -> Global Parameters**.
2. In the TSAM Plus Global Properties tab, select one of the storage: Database, Hadoop, or local file.
3. If Hadoop is selected, continue specifying the following fields:

- Hadoop URL:

Specify the Hadoop URL where you want to store the user payload data. The Hadoop URL must include the hostname, port, and payload file directory. For example:

```
hdfs://localhost:8080/tmp/tuxPayload/
```

4. Click **Modify**.

Configuring User Payload Collection

You can define user payload data collection policies for services or call path. Take call path for example, to configure collection policies, do these steps:

1. From the TSAM Plus console top menu bar, click **Policy**-> **Tuxedo Monitoring Policy**.
2. In the Monitoring Policy List page, click **Add** to enter the **Edit Policy** page.
3. In the left Tuxedo Component filter area, select the Tuxedo domain to monitor.
4. In the Call path tab, select **Enable** to enable the call path metrics collection.
5. Click the **Define Payload Collection** button to configure payload collection rules.
6. In the Define Payload Collection page, you can enable request and/or response payload collection by selecting the corresponding checkbox. If request payload collection is enabled, you can decide if only collecting the payload for initial called service by selecting **Initial Called Service Only** or not.
7. Click OK. Then enable the policy.

Note: The call path user payload data collection policies cannot be propagated to other domains.

Now you have completed the definition of a most general user payload collection policy. All the payload belonging to the defined scope will be collected and saved to the designated payload storage.

You can also narrow down the payload data collection scope by defining collection data criteria or result filter. For more information, see [Defining Payload Collection Data Criteria](#) and [Defining Payload Collection Result Filter](#).

Defining Payload Collection Data Criteria

When a data criteria is defined, TSAM Plus collects the payload only if the typed buffer value fulfills the criteria. For example, when a STRING typed buffer starts with the word Hello, or a

FML32 typed buffer whose ACCOUNT_ID field has a value larger than 1000, the system collects the payload.

To define the data criteria, do the following steps:

1. Go to the **Data Criteria** tab in the Define Payload Collection dialog. All the supported buffer types are listed in the drop down list.
2. Select the buffer type you want to define a criteria against and then click **Add**.

A table with the title Add one or multiple data criteria entries is displayed.

Table 6-3 Data Criteria Table

Column Name	Visible for Buffer Type	Description
Field Name	FML/FM32 VIEW/VIEW32 X_COMMON/X_C_TYPE	A single selection drop down list. For FML/FML32, it is the field name in the FML field table definition file. The supported field types are: short, long, float, double, char, and string. Fields with other types are hidden from the drop down list. For all VIEW liked buffer types, it is the VIEW name defined after the VIEW keyword in the VIEW table definition file.
Sub Field Name	VIEW/VIEW32 X_COMMON/X_C_TYPE	A single selection drop down list. The value is the VIEW field name (CNAME) in the VIEW table definition file. The supported field types are same with the FML fields.
Occurrence	FML/FML32	The specific occurrence of a FML field.
Operator	All buffer types	A single selection drop down list. For numeric fields, the options are: >, >=, <, <=, ==, != For alphabetic fields, the options are: ==, !=, %%, !%

Table 6-3 Data Criteria Table

Column Name	Visible for Buffer Type	Description
Value	All buffer types except XML	<p>A text input box.</p> <p>It is the value part (after the operator) of a FML Boolean Expression. No operators are allowed here.</p> <p>It must fit all the syntax defined by the FML Boolean Expression. For more information about FML Boolean Expression, refer to the online Oracle Tuxedo Documentation.</p> <p>Examples of the value are:</p> <ul style="list-style-type: none"> • 'Hello.*' (Note the single quote) • 1000
Expression	XML	<p>A text input box.</p> <p>A valid XPath expression is expected here.</p>
Direction	All buffer types	<p>This area is made up of three mutually exclusive radio buttons.</p> <p>If both request and response payload collection are enabled in the General tab, all the three radio buttons - Inout, In, and Out are enabled. Otherwise only the corresponding radio button is enabled.</p> <p>If Inout is selected, current criteria expression takes effect for both request and response buffers.</p> <p>If In or Out is selected, only request or response buffer is evaluated against current criteria.</p>
Actions	All buffer types	<p>This column contains one add button and one remove button.</p> <p>Multiple data criteria expression entries can be added to the table. On saving of the policy, each entry forms a FML Boolean expression and different entries in the same table are of the logical AND relationship.</p>

3. Once all the criteria expression entries are defined, click **Submit** to save your configurations.

After returning to the data criteria listing page, you can find the new-created data criteria in the listing table.

Notes:

- You can add multiple data criteria for the same buffer type. They are of logical OR relationship. It means that any of the data criteria is satisfied when the payload is collected.
- For the buffer type `FML32/FML/VIEW32/VIEW/X_C_TYPE/X_COMMON`, the field table definition files must be uploaded in the **Data Management -> Field Tables Management** tab page before defining the data criteria.

Defining Payload Collection Result Filter

The result filter defines for each buffer type, for example, which fields or the char sequence ranges are collected, regardless of the runtime payload data value. Compared to data criteria, it is a relative static definition to filter unnecessary fields out.

Defining the result filter are very similar to the operations of defining data criteria. The differences are:

In the result filter listing page, each buffer type can only be defined with one result filter.

In the result filter entries definition page, different filter entries are of logical OR relationship. This means in the runtime, TSAM Plus tries to collect all the defined entries of a result filter.

Note: If both the result filters for call path and services are enabled on a Tuxedo service call, the filters on the call path take effect.

Exporting Collected User Payload Data

The user payload data stored in Oracle TSAM database, local file, or Hadoop is in binary format. For ease of data analysis, you can export the data to text format or XML format using the export APIs or payload dump tool.

APIs for Data Export

[Table 6-4](#) lists the APIs for exporting the user payload data.

Table 6-4 APIs for Data Export

Category	API	Description
Interface	PayloadRepository	Payload metrics storage interface
Class	FilePayloadIterator	Payload record iterator of file storage
	FilePayloadRepository	File storage of payload metrics
	HadoopPayloadIterator	Payload record iterator of Hadoop storage
	HadoopPayloadRepository	Hadoop storage of payload metrics
	payloaddump	A command line tool dumping payload metrics
	PayloadRecord	Payload record in the storage
	PayloadValue	Payload metrics.
	TSAMDBPayloadIterator	Payload record iterator of TSAM database storage
	TSAMDBPayloadRepository	TSAM Plus database storage of payload metrics
Enum	Enum PayloadType	Payload metrics type

When using API for data export, you must include the following jar in CLASSPATH under

```
$TSAMDIR/lib :
```

```
CLASSPATH=payloadapi.jar:com.bea.core.jatmi_2.0.0.0.jar:com.oracle.tuxedo.tjatmi_12.1.3.0.jar:commons-cli-1.2.jar
```

If you want to export the payload from Hadoop dfs, set the HADOOP_HOME:

```
HADOOP_HOME=/home/user/hadoop-2.6.0/
```

If you want to export the payload from Oracle Database, include `ojdbc6.jar` in CLASSPATH.

PayloadRepository

All Superinterfaces

```
java.lang.AutoCloseable, java.io.Closeable,  
java.lang.Iterable<PayloadRecord>
```

All Known Implementing Classes

```
FilePayloadRepository, HadoopPayloadRepository, TSAMDBPayloadRepository
```

Description

```
public interface PayloadRepository  
extends java.lang.Iterable<PayloadRecord>, java.io.Closeable  
  
Payload metrics storage interface
```

FilePayloadIterator

```
java.lang.Object  
com.oracle.tsam.payload.FilePayloadIterator
```

All Implemented Interfaces

```
java.util.Iterator<PayloadRecord>
```

Description

```
public class FilePayloadIterator  
extends java.lang.Object  
implements java.util.Iterator<PayloadRecord>  
  
Payload record iterator of file storage.
```

Methods inherited from class java.lang.Object

```
equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait
```

Constructor Detail

FilePayloadIterator

```
public FilePayloadIterator(org.apache.hadoop.io.SequenceFile.Reader  
reader)
```

Constructs a FilePayloadIterator with a reader

Method Detail

hasNext

```
public boolean hasNext()
```

Specified by:

```
hasNext in interface java.util.Iterator<PayloadRecord>
```

next

```
public PayloadRecord next()
```

Specified by:

```
next in interface java.util.Iterator<PayloadRecord>
```

remove

```
public void remove()
```

Specified by:

```
remove in interface java.util.Iterator<PayloadRecord>
```

FilePayloadRepository

```
java.lang.Object
com.oracle.tsam.payload.FilePayloadRepository
```

All Implemented Interfaces

```
PayloadRepository, java.io.Closeable, java.lang.AutoCloseable,
java.lang.Iterable<PayloadRecord>
```

Description

```
public class FilePayloadRepository
extends java.lang.Object
implements PayloadRepository
```

File storage of payload metrics.

Methods inherited from class java.lang.Object

```
equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait
```

Constructor Detail

FilePayloadRepository

```
public FilePayloadRepository(java.lang.String path)
```

User Payload Collection

throws `java.io.IOException`

Constructs a `FilePayloadRepository` with a path

Throws:

`java.io.IOException`

Method Detail

iterator

```
public java.util.Iterator<PayloadRecord> iterator()
```

Specified by:

`iterator` in interface `java.lang.Iterable<PayloadRecord>`

close

```
public void close()
    throws java.io.IOException
```

Specified by:

`close` in interface `java.io.Closeable`

Specified by:

`close` in interface `java.lang.AutoCloseable`

Throws:

`java.io.IOException`

getFile

```
public java.io.File getFile()
```

Returns the File object

HadoopPayloadIterator

```
java.lang.Object
com.oracle.tsam.payload.HadoopPayloadIterator
```

All Implemented Interfaces

```
java.util.Iterator<PayloadRecord>
```

Description

```
public class HadoopPayloadIterator
    extends java.lang.Object
```

implements `java.util.Iterator<PayloadRecord>`

Payload record iterator of Hadoop storage.

Methods inherited from class `java.lang.Object`

`equals`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

Constructor Detail

HadoopPayloadIterator

```
public HadoopPayloadIterator(java.lang.String HadoopURL,
                             java.lang.String[] MatchFiles)
    throws java.io.IOException
```

Throws:

`java.io.IOException`

Method Detail

hasNext

```
public boolean hasNext()
```

Specified by:

`hasNext` in interface `java.util.Iterator<PayloadRecord>`

next

```
public PayloadRecord next()
```

Specified by:

`next` in interface `java.util.Iterator<PayloadRecord>`

close

```
public void close()
    throws java.io.IOException
```

Throws:

`java.io.IOException`

remove

```
public void remove()
```

Specified by:

`remove` in interface `java.util.Iterator<PayloadRecord>`

HadoopPayloadRepository

```
java.lang.Object
com.oracle.tsam.payload.HadoopPayloadRepository
```

All Implemented Interfaces

```
PayloadRepository, java.io.Closeable, java.lang.AutoCloseable,
java.lang.Iterable<PayloadRecord>
```

Description

```
public class HadoopPayloadRepository
extends java.lang.Object
implements PayloadRepository
```

Hadoop storage of payload metrics.

Methods inherited from class java.lang.Object

```
equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait
```

HadoopPayloadRepository

HadoopPayloadIterator

```
public HadoopPayloadRepository(java.lang.String HadoopURL,
                               java.lang.String Domain,
                               java.lang.String Machine,
                               java.util.Date StartDate,
                               java.util.Date EndDate)
    throws java.io.IOException
```

Constructs a HadoopPayloadRepository with a Hadoop URL and filters of domain name, machine name, start date, and end date

Throws:

```
java.io.IOException
```

Method Detail

iterator

```
public java.util.Iterator<PayloadRecord> iterator()
```

Specified by:

```
iterator in interface java.lang.Iterable<PayloadRecord>
```


close

```
public void close()
    throws java.io.IOException
```

Specified by:

close in interface java.io.Closeable

Specified by:

close in interface java.lang.AutoCloseable

Throws:

java.io.IOException

getHadoopURL

```
public java.lang.String getHadoopURL()
Returns the Hadoop URL
```

payloaddump

```
java.lang.Object
    com.oracle.tsam.payload.payloaddump
```

Description

```
public class payloaddump
    extends java.lang.Object
```

A command line tool dumping payload metrics.

Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail**payloaddump**

```
public payloaddump()
```

Method Detail**main**

```
public static void main(java.lang.String[] args)
    throws org.apache.commons.cli.ParseException,
           java.lang.ClassNotFoundException,
           java.sql.SQLException,
           java.io.IOException,
```

```
javax.xml.parsers.ParserConfigurationException,  
java.text.ParseException,  
java.lang.Exception
```

Parameters:

args

-url

Mandatory parameter. Payload repository URL. It can be a database URL, Hadoop URL, or a file path

-username

Database user name

-password

Database password

-payloadtype

Mandatory parameter. Request or reply

-wherecondition

SQL query condition. It is available only when the URL is specified to a Database URL.

-domainid

The domain ID used to filter the results

-servicename

The service name used to filter the results

-outputtype

XML or text. The default output type is xml.

-fieldnames

Expected field names when the output type is text. For the Tuxedo type CARRAY, X_OCTET, and STRING, the fieldnames is the aliasNames specified in the result filter of the user payload policy. If url is specified to a Database URL and servicename is specified, the default fieldnames is from the service contract information. If url is specified to a Database URL and policynname and servicename are both specified, the default fieldnames is the join result of the service contract information and the result filter of the policy.

-o

Output file name. The default output is standard output.

Throws:

```

org.apache.commons.cli.ParseException
java.lang.ClassNotFoundException
java.sql.SQLException
java.io.IOException
javax.xml.parsers.ParserConfigurationException
java.text.ParseException
java.lang.Exception

```

PayloadRecord

```

java.lang.Object
com.oracle.tsam.payload.PayloadRecord

```

Description

```

public class PayloadRecord
extends java.lang.Object

```

Payload record in the storage.

Methods inherited from class java.lang.Object

```

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

```

Constructor Detail**PayloadRecord**

```

public PayloadRecord(java.lang.String dDOMAINID,
                    java.lang.String LMID,
                    short groupID,
                    short serverID,
                    java.lang.String serviceName,
                    java.lang.String correlationID,
                    java.lang.String ECID,
                    java.lang.Short errno,
                    PayloadType payloadType,
                    java.util.Date logtime,
                    PayloadValue payloadValue)

```

Method Detail**getdOMAINID**

```

public java.lang.String getdDOMAINID()

```

setdOMAINID

```
public void setdOMAINID(java.lang.String domainID)
```

getLMID

```
public java.lang.String getLMID()
```

setLMID

```
public void setLMID(java.lang.String lmid)
```

getGroupID

```
public short getGroupID()
```

setGroupID

```
public void setGroupID(short groupID)
```

getServerID

```
public short getServerID()
```

setServerID

```
public void setServerID(short serverID)
```

getServiceName

```
public java.lang.String getServiceName()
```

setServiceName

```
public void setServiceName(java.lang.String serviceName)
```

getCorrelationID

```
public java.lang.String getCorrelationID()
```

setCorrelationID

```
public void setCorrelationID(java.lang.String correlationID)
```

getECID

```
public java.lang.String getECID()
```

setECID

```
public void setECID(java.lang.String eCID)
```

getTperrno

```
public PayloadType getPayloadType()
```

setTperrno

```
public void setTperrno(java.lang.Short tperrno)
```

getPayloadType

```
public PayloadType getPayloadType()
```

setPayloadType

```
public void setPayloadType(PayloadType payloadType)
```

getPayloadValue

```
public PayloadValue getPayloadValue()
```

setPayloadValue

```
public void setPayloadValue(PayloadValue payloadValue)
```

getLogtime

```
public java.util.Date getLogtime()
```

setLogtime

```
public void setLogtime(java.util.Date logtime)
```

PayloadValue

```
java.lang.Object
com.oracle.tsam.payload.PayloadValue
```

Description

```
public class PayloadValue
extends java.lang.Object
```

Payload metrics. It has two kinds of output formats:

- **Text:** It is same to the Database export format in "text" mode so that it can be imported into a Database table as the BI data source.

Note: You can define the sequence of the output data using String [] fieldNames. If no fieldnames are provided, the fields are outputted in the default order. None of the embedded buffer is outputted.

- **XML:** The formatter can convert the typed buffer CARRAY, X_OCTET, STRING, FML, FML32, VIEW, VIEW32, X_C_TYPE, X_COMMON, and XML to XML format. If the buffers are nested, all the nested buffers are also converted.

For CARRAY/X_OCTET/STRING, the payload is converted to an element in XML format. CARRAY is encoded to a base64 string. The element name is the value specified to the parameter elemName.

For FML/FML32/VIEW/VIEW32/X_C_TYPE/X_COMMON, the formatter converts all the fields in the typed buffer. If these kinds of typed buffers are nested, all the nested buffers are also converted.

For XML, the formatter makes no change..

Methods inherited from class java.lang.Object

`equals`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

Constructor Detail

PayloadValue

```
public PayloadValue(short valueTypeIndex,  
                    byte[] payloadValue)
```

Method Detail

getFieldNames

```
public java.lang.String[] getFieldNames()
```

Returns:

The field names in this object

toXMLString

```
public java.lang.String toXMLString(java.lang.String encoding)  
                                throws java.io.IOException,  
                                javax.xml.parsers.ParserConfigurationException,  
                                org.xml.sax.SAXException
```

convert payload value to a XML string.

Parameters:

encoding

Throws:

java.io.IOException
javax.xml.parsers.ParserConfigurationException
org.xml.sax.SAXException

toText

```
public java.lang.String toText(char delimiter,  
                               char enclosure)  
                             throws java.io.IOException,
```

```
                             javax.xml.parsers.ParserConfigurationException,  
                             org.xml.sax.SAXException
```

convert payload value to text, which can be imported into a Database table. the embedded fields are not in the returned value. characters equaling to the enclosure are duplicated

Parameters:

delimiter - delimiter character

enclosure - enclosure character

Returns:

text

Throws:

java.io.IOException

javax.xml.parsers.ParserConfigurationException

org.xml.sax.SAXException

toText

```
public java.lang.String toText()
    throws java.io.IOException,
```

```
    javax.xml.parsers.ParserConfigurationException,
    org.xml.sax.SAXException
```

convert payload value to text, which can be imported into a Database table, which the default setting of delimiter=tab and enclosure="". the embedded fields are not in the returned value.

Returns:

text

Throws:

java.io.IOException

javax.xml.parsers.ParserConfigurationException

org.xml.sax.SAXException

toText

```
public java.lang.String toText(char delimiter,
    char enclosure,
    java.lang.String[] fieldNames)
    throws java.io.IOException,
```

```
    javax.xml.parsers.ParserConfigurationException,
    org.xml.sax.SAXException
```

convert the specified fields in payload value to text, which can be imported into a Database table. the embedded fields are not in the returned value. characters equaling to the enclosure are duplicated.

Parameters:

delimiter -

enclosure -

User Payload Collection

fieldNames -

Returns:

text

Throws:

java.io.IOException

javax.xml.parsers.ParserConfigurationException

toText

```
public java.lang.String toText(java.lang.String[] fieldNames)
                           throws java.io.IOException,
```

```
                           javax.xml.parsers.ParserConfigurationException,
                           org.xml.sax.SAXException
```

convert the specified fields of the payload value to text, which can be imported into a Database table, which the default setting of delimiter=tab and enclosure="". the embedded fields are not in the returned value.

Returns:

text

Throws:

java.io.IOException

javax.xml.parsers.ParserConfigurationException

TSAMDBPayloadIterator

```
java.lang.Object
com.oracle.tsam.payload.TSAMDBPayloadIterator
```

All Implemented Interfaces

```
java.util.Iterator<PayloadRecord>
```

Description

```
public class TSAMDBPayloadIterator
extends java.lang.Object
implements java.util.Iterator<PayloadRecord>
```

Payload record iterator of TSAM database storage.

Methods inherited from class java.lang.Object

`equals`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

Constructor Detail

TSAMDBPayloadIterator

```
public TSAMDBPayloadIterator(java.lang.String
    whereConditionalExpression,
                             java.sql.Connection con)
    throws java.sql.SQLException
```

Throws:

`java.sql.SQLException`

Method Detail

hasNext

```
public boolean hasNext()
```

Specified by:

`hasNext` in interface `java.util.Iterator<PayloadRecord>`

next

```
public PayloadRecord next()
```

Specified by:

`next` in interface `java.util.Iterator<PayloadRecord>`

remove

```
public void remove()
```

Specified by:

`remove` in interface `java.util.Iterator<PayloadRecord>`

TSAMDBPayloadRepository

```
java.lang.Object
com.oracle.tsam.payload.TSAMDBPayloadRepository
```

All Implemented Interfaces

`PayloadRepository`, `java.io.Closeable`, `java.lang.AutoCloseable`,
`java.lang.Iterable<PayloadRecord>`

Description

```
public class TSAMDBPayloadRepository
extends java.lang.Object
implements PayloadRepository
```

TSAM Plus database storage of payload metrics.

Methods inherited from class java.lang.Object

```
equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait
```

Constructor Detail

TSAMDBPayloadRepository

```
public TSAMDBPayloadRepository(java.lang.String TSAMDBURL)
                                throws java.sql.SQLException,
                                        java.lang.ClassNotFoundException
```

Constructs a TSAMDBPayloadRepository with a TSAM Plus database URL

Throws:

```
java.sql.SQLException
java.lang.ClassNotFoundException
```

TSAMDBPayloadRepository

```
public TSAMDBPayloadRepository(java.sql.Connection connection)
```

Constructs a TSAMDBPayloadRepository with a TSAM database connection

Method Detail

iterator

```
public java.util.Iterator<PayloadRecord> iterator()
```

Specified by:

```
iterator in interface java.lang.Iterable<PayloadRecord>
```

iterator

```
public java.util.Iterator<PayloadRecord> iterator(java.lang.String
whereConditionalExpression)
                                                throws java.sql.SQLException
```

Gets an iterator from the inputted SQL query condition

Parameters:

whereConditionalExpression - SQL query condition, for an example, "where DOMAINID='DOMAIN1' and LMID='L1' and GROUPID=1 and SERVERID=1 and SVCNAME='toupper' and PAYLOADTYPE=1"

Note: for the field PAYLOADTYPE, 1 means request, 2 means reply

Returns:

An iterator

Throws:

java.sql.SQLException

close

```
public void close()
    throws java.io.IOException
```

Specified by:

close in interface java.io.Closeable

Specified by:

close in interface java.lang.AutoCloseable

Throws:

java.io.IOException

getConnection

```
public java.sql.Connection getConnection()
```

Returns:

TSAM database connection

getFieldNames

```
public java.lang.String[] getFieldNames(java.lang.String domainID,
    java.lang.String serviceName,
    PayloadType payloadType)
    throws java.lang.Exception
```

Parameters:

domainID - domain ID
 serviceName - service name
 payloadType - payload type

Returns:

Field names

Throws:

```
java.lang.Exception
```

Enum PayloadType

```
java.lang.Object  
java.lang.Enum<PayloadType>  
com.oracle.tsam.payload.PayloadType
```

All Implemented Interfaces

```
java.io.Serializable, java.lang.Comparable<PayloadType>
```

Description

```
public enum PayloadType  
extends java.lang.Enum<PayloadType>
```

Payload metrics type.

Methods inherited from class java.lang.Enum

```
compareTo, equals, getDeclaringClass, hashCode, name, ordinal, toString,  
valueOf
```

Methods inherited from class java.lang.Object

```
getClass, notify, notifyAll, wait, wait, wait
```

Enum Constant Detail

request

```
public static final PayloadType request  
Request of a Tuxedo call
```

Method Detail

values

```
public static PayloadType[] values()  
Returns an array containing the constants of this enum type, in the order they are declared.  
This method may be used to iterate over the constants as follows:  
for (PayloadType c : PayloadType.values())  
    System.out.println(c);
```

Returns:

an array containing the constants of this enum type, in the order they are declared

valueOf

```
public static PayloadType valueOf(java.lang.String name)
```

Returns the enum constant of this type with the specified name. The string must match exactly an identifier used to declare an enum constant in this type. (Extraneous whitespace characters are not permitted.)

Parameters:

name - the name of the enum constant to be returned.

Returns:

The enum constant with the specified name

Throws:

java.lang.IllegalArgumentException - if this enum type has no constant with the specified name
java.lang.NullPointerException - if the argument is null

Payload API Samples

[Listing 6-1](#) shows an example that reads payload metrics from TSAM Plus database, and exports the data to a text file.

Listing 6-1 Payload API Sample 1

```
public class DBPayloadSample {

    public static void main(String[] args) throws IOException {

        TSAMDBPayloadRepository tsamDBPayloadRecordRepository = new
        TSAMDBPayloadRepository(

            "jdbc:oracle:thin:@localhost:1521:orcl",

            "tsam", "tsam");

        Iterator<PayloadRecord> iterator =
        tsamDBPayloadRecordRepository

            .iterator("where DOMAINID='DOMAIN1' and
            svcname='toupper' and PAYLOADTYPE=1");
```

User Payload Collection

```
String[] fieldNames =
tSAMDBPayloadRecordRepository.getFieldNames(
    "DOMAIN1", "toupper", PayloadType.request);

Writer writer = new FileWriter("payload.txt");

for (int i = 0; i < fieldNames.length - 1; i++) {
    writer.append "\"" + fieldNames[i] + "\"\t");
}
writer.append "\"" + fieldNames[fieldNames.length - 1] +
"\"");

while (iterator.hasNext()) {
    {
        PayloadValue payloadValue =
iterator.next().getPayloadValue();

writer.append(payloadValue.toText(fieldNames));
        writer.append('\n');
    }
}

writer.close();

tSAMDBPayloadRecordRepository.close();
```

```

    }
}

```

The output format of the file "payload.txt" is

```

"STRINGFIELD1" "STRINGFIELD2" "INTFIELD1" "SHORTFIELD1"
"abc" "def" "10000" ""
"hij" "" "1" "2"

```

[Listing 6-2](#) shows an example that reads payload metrics from Hadoop, and exports the data to an XML file.

Listing 6-2 Payload API Sample 2

```

public class HadoopPayloadSample {

    public static void main(String[] args) throws IOException {

        HadoopPayloadRepository hadoopPayloadRecordRepository = new
HadoopPayloadRepository(

            "hdfs://localhost:8080/tmp/tuxPayload/");

        Iterator<PayloadRecord> iterator =
hadoopPayloadRecordRepository

            .iterator();

        Writer writer = new FileWriter("payload.txt");

        while (iterator.hasNext()) {

            PayloadRecord payloadRecord = iterator.next();

```

User Payload Collection

```
        if ("DOMAIN1".equals(payloadRecord.getDOMAINID())
            &&
            "toupper".equals(payloadRecord.getServiceName())
            && (payloadRecord.getPayloadType() ==
PayloadType.request)) {

                PayloadValue payloadValue =
iterator.next().getPayloadValue();

                writer.append(payloadValue.toXml());
                writer.append('\n');

            }
        }

        writer.close();

        hadoopPayloadRecordRepository.close();
    }
}
```

Payload Dump Tool

Before running the payload dump tool, you must configure the following:

- Set the `CLASS PATH` as follows:

```
CLASSPATH=payloadapi.jar:com.bea.core.jatmi_2.0.0.0.jar:com.oracle.tuxe
do.tjatmi_12.1.3.0.jar:commons-cli-1.2.jar
```


- If you want to export the payload from Hadoop dfs, set the `HADOOP_HOME`:

```
export HADOOP_HOME=/home/user/hadoop-2.6.0/
```
- If you want to export the payload from Oracle Database, include `ojdbc6.jar` in `CLASSPATH`.

The payload dump tool command is in the following format:

```
payloaddump -url <URL> [-username <username> -password <password>]
-payloadtype request|reply [-wherecondition <wherecondition>] [-domainid
<domainid>] [-servicename <servicename>] [-outputtype xml|text]
[-fieldnames <fieldnames>] [-o outputFile]
```

For each parameter description, refer to [payloaddump](#).

For example, to export the payload from Hadoop, run the following command:

```
java com.oracle.tsam.payload.payloaddump -url
"hdfs://localhost:9000/data/" -payloadtype reply -outputtype xml
```

To export the payload from TSAM Plus Database, run the following command:

```
java com.oracle.tsam.payload.payloaddump -url
"jdbc:oracle:thin:@localhost:1521:tsam" -username tsam -password tsam
-payloadtype reply -outputtype xml
```

User Payload Collection

Using Test Load Generator

This chapter contains the following topics:

- [Overview](#)
- [Creating Call Path Policy for a Replay](#)
- [Creating a Replay Definition](#)
- [Running a Replay](#)
- [Running a Replay in Auto-Created Test Domain](#)

Overview

Test load generator is a Tuxedo applications load test tool that facilitates effective and efficient load testing. This feature works in conjunction with the payload collection feature to capture the call path and payload metrics coming from a domain gateway, Web service client or application server. One can create replay definition from call paths and execute the replay based on policies for standard and stress test modes.

Creating Call Path Policy for a Replay

A replay can be constructed from one or more call paths.

In order to collect call path and payload metrics, you need to create a call path policy on the target service with payload collection enabled by doing these steps:

1. From the TSAM Plus console top menu bar, click **Policy-> Tuxedo Monitoring Policy**.
2. In the Monitoring Policy List page, click **Add** to enter the **Create Policy** page.
3. In the Call path tab, select **Enable** to enable the call path metrics collection.
4. Select the domain in which payload data will be collected in the left **Domain** list, the **Define Payload Collection** button is activated. Click the button.
5. In the Define Payload Collection page, enable request and response payload collection by selecting the corresponding checkbox.

It is recommended you select **Initial Called Service Only** to avoid collecting unnecessary metrics. The payload collection must not include any result filter.

Creating a Replay Definition

After the call path policy for the replay is created, do one or more Tuxedo calls to collect the source metrics from call paths. Then you can create a replay definition from call paths by doing these steps:

1. From the TSAM Plus console top menu bar, click **Load Generator-> Replay Definition**.
2. Click **Add**. The Call Path Query and Call Path Result Panels are displayed.
3. Set the query filter options and click **Query**. The desired call paths are displayed in the result panel.
4. Select the call path from the list and click **Create Replay Definition**.

The new-created replay definition is displayed in the Replay Definition list.

Setting Replay Definition Attributes

You can edit the replay definition attributes.

[Table 7-1](#) lists replay definition attributes.

Table 7-1 Replay Definition Attributes

Attribute	Description	Default Value	Editable
name	Replay definition name		yes
Tuxedo user name	The Tuxedo user name used to do <code>tpinit()</code>		yes
Tuxedo client name	The Tuxedo client name used to do <code>tpinit()</code>		yes
Tuxedo group name	The Tuxedo client name used to do <code>tpinit()</code>		yes
transaction calls	Call <code>tpbegin()</code> when starting the call(s), and call <code>tpcommit()</code> when ending the calls	From source call path metrics	yes
transaction parameters	Call <code>tpbegin()</code> when starting the call(s), and call <code>tpcommit()</code> when ending the calls	From source XA metrics	no
break when any call fails	Whether to break when any call fails. Its value is yes or no.		yes
create time	The time when the definition was created		no
modify time	The time when the definition was modified		no

The default service call attributes are from the first called service (not the "Initial Called Service" in the call path when the initiator is a server) when the replay definition is constructed from call path metrics.

[Table 7-2](#) lists service calls attributes of replay definition.

Table 7-2 Replay Definition Service Calls Attributes

Attribute	Default Value	Editable
service name	From source metrics	No
source correlation ID	From source metrics	No

Table 7-2 Replay Definition Service Calls Attributes

Attribute	Default Value	Editable
call flag "TPNOTRAN"	From the source metrics	Yes
call flag "TPNOCHANGE"	From the source metrics	Yes
call flag "TPNOCOPY"	From the source metrics	Yes
call flag "TPNOBLOCK"	From the source metrics	Yes
call flag "TPNOTIME"	From the source metrics	Yes
call flag "TPSIGRSTRT"	From the source metrics	Yes

Running a Replay

To run a replay, execute a replay agent in the command line with the exported replay definition file. TSAM Plus replay agents are under `$TUXDIR/bin/`. There are two kinds of agents:

- `tsamreplayagent`

It supports `tpcall()/tpacall()/tpgetrply()` calls.

- `tsamreplaywagent`

It supports `tpcall()/tpacall()/tpgetrply()` calls in the workstation mode.

Before running a replay, make sure the target Tuxedo domain is started up. If TSAM Plus metrics are expected, you must create corresponding policies in TSAM Plus.

Note: The evaluation unit of throughput, latency, success/fail number is the whole replay definition, not one call in it.

[Table 7-3](#) lists the replay agent parameters.

Table 7-3 Replay Agent Parameters

Parameter	Description
-p	Invokes the execution plan file when running a replay. Refer to Execution Plan File .
-f	The path of the security profile to be generated. Refer to Replay Security Profile .

Execution Plan File

An execution plan file stores a replay execution plan attributes. The attribute format is "NAME=VALUE". The attribute name is case-insensitive.

There are two types of execution plan modes, normal mode and stress test mode. [Table 7-4](#) lists the general attribute.

Table 7-4 General Attribute

Attribute	Description	Default Value
definitionFile	Definition file name	

Normal Mode Attributes

The normal mode is the default mode. It is normally used to do unit test or diagnostic a service call. It can also be used to do a stress test by setting the parameters `concurrency number` and `repeat times` to huge values.

Table 7-5 Normal Mode Attributes

Attribute	Description	Default Value
concurrencyNumber	Concurrency number	1
repeatTime	Repeat time	1
sleepTime	Sleep time when repeating in seconds	0
compareReply	Compares the reply with the one in the replay definition	False

Following is an execution plan file example:

```
definitionFile=def1.xml
concurrencyNumber=10
repeattime=100
sleepTime=0
```

```
compareReply=true
```

Stress Test Mode Attributes

There are three test mode types:

- Increase the load till the throughput does not increase any longer.

Following is an execution plan file example:

```
definitionFile=def1.xml
timeWindow=20
minCurrency=10
maxCurrency=1000
iniCurrency=10
totalRunTime=3600
```

- Increase the load till the max/average latency reaches the specified value. It is enabled when the parameter `maxLatency` or `avgLatency` is specified.

Note: If both `maxLatency` and `avgLatency` are specified, `maxLatency` is ignored.

Following is an execution plan file example:

```
definitionFile=def1.xml
timeWindow=20
minCurrency=10
maxCurrency=1000
iniCurrency=10
maxLatency=1000
avgLatency=500
totalRunTime=3600
```

- Run with the specified throughput. It is enabled when the parameter `throughput` is specified.

Following is an execution plan file example:

```
definitionFile=def1.xml
timeWindow=20
throughput=1000
```



```
totalRunTime=3600
```

Table 7-6 Stress Test Mode Attributes

Attribute	Description	Default Value
timeWindow	Load increasing/decreasing time window in seconds	30 seconds
minCurrency	Minimum concurrency number limitation. The concurrency number is always not less than it.	1
maxCurrency	Maximum concurrency number limitation. The concurrency number is always not greater than it.	10240
iniCurrency	The initial concurrency number when the replay starts.	1
maxLatency	Maximum latency in microseconds	
avgLatency	Average latency in microseconds	
throughput	Run with the specified throughput per second	
totalRunTime	Total running time in seconds	No limitation

Replay Security Profile

`genreplayprofile` is under `$TUXDIR/bin`. Its parameter is the path of the security profile to be generated. When `genreplayprofile` is launched, you are prompted to enter the Oracle Tuxedo application password, user name and user password. It generates the replay security profile with the inputted content. The inputted content is used by the replay agent to do `tpinit()`. If different user names are specified in the replay security profile and the replay definition, the one in the replay security profile takes effect.

Replay Execution Result

Replay execution result is outputted to the standard output after the replay is finished or cancelled.

The normal schedule mode result contains the following contents:

- Submit Time
- End Time
- Success number
- Failure number
- Throughput per second
- Average latency in microseconds
- Identical replies
- Different replies

The stress schedule mode result contains the following contents:

- Detailed information in every 30-second execution time window, including:
 - Current time Success number
 - Failed number
 - Throughput
 - Average latency
 - Concurrency number
- Replay summary
 - Submit Time
 - End Time
 - Success number
 - Failure number
 - Max Throughput per second when the test is ending
 - Throughput per second when the test is ending
 - Minimum latency in microseconds when the test is ending
 - Maximum latency in microseconds when the test is ending
 - Average latency in microseconds when the test is ending
 - Concurrency number when the test is ending

Running a Replay in Auto-Created Test Domain

You can use the tool `tlgdomgen.py` to create a test domain to call the service in the target domain. The tool is under `$TUXDIR/bin` and can be run using either python or Tuxedo Script Tool (TXST).

Configuration File

Before running `tlgdomgen.py`, you need to create the domain configuration file.

[Table 7-7](#) lists the configuration file parameters.

Table 7-7 Configuration File Parameters

Parameter	Description
LDOMPORT	Mandatory. Local access point port of the GWTDOMAIN in the test domain
TUXDIR	Mandatory. TUXDIR of the test domain
APPDIR	Mandatory. APPDIR of the test domain
RDOMID	Mandatory. Access point ID of the GWTDOMAIN in the target domain
RDOMPORT	Mandatory. Access point port of the GWTDOMAIN in the target domain
RDOMHOST	Mandatory. Access point hostname of the GWTDOMAIN in the target domain
IMPORTSERVICES	List of service names to be imported to the test domain, delimited by space
TSAMMGRHOST	Hostname of the TSAM Plus Manager to be connected by the test domain. LMS is not configured if this item is not specified
LDOMID	Access point ID of the GWTDOMAIN in the test domain
IPCKEY	IPCKEY of the test domain
TSAMMGRPORT	Port of the TSAM Manager to be connected by the test domain. LMS is not configured if this item is not specified

Following is a configuration file example:

```
[Input]
LDMPORT = 8888
TUXDIR = /home/beadev/bea_linux64.13c/tuxedo12.2.2.0.0
APPDIR = /home/beadev/applicationgrid/domains/domain4
RDOMID = DOMAIN2
RDMPORT = 6666
RDMHOST = localhost
```

Executing tlgdomgen.py

To run `tlgdomgen.py` using `TXST`, do these steps:

1. set Tuxedo environment:

```
$TUXDIR/tux.env
```

2. set `TXST` environment:

```
export
CLASSPATH=$CLASSPATH:$TUXDIR/jmx/jython-standalone.jar:$TUXDIR/jmx/tmjm
x_tux.jar:$TUXDIR/jmx/tmjmx_metadata.jar:$TUXDIR/udataobj/tuxj/com.bea.
core.jatmi_2.0.0.0.jar
```

3. run `tlgdomgen.py`:

```
java oracle.tuxedo.TXST $TUXDIR/bin/tlgdomgen.py <configuration file
name>
```

The following files are generated:

- `ubbconfig.test`: `UBBCONFIG` of the test domain.
 - `addrom.mib`: MIB scripts to add `GWTDOMAIN` access point to the target domain.
 - `setenv.test`: `setenv` of the test domain.
 - `run.sh`: execution scripts to set up and run the test domain.
 - `dmconfig.test`: `DMCONFIG` of the test domain.
4. In the target domain, run the following commands to add the test domain to the target domain:

```
. $TUXDIR/tux.env
export FLDTBLDIR32=$TUXDIR/udataobj
export FIELDTBLS32="tpadm,Usysf132"
```

```
ud32 -C tpsysadm < addrom.mib
```

5. Boot the test domain using the file `run.sh`, and then run the replay.

Using Test Load Generator

Setting Up Oracle Tuxedo as a Cloud Service

This chapter contains the following sections:

- [Overview](#)
- [Setting Up the Tuxedo Self Service Portal](#)
- [Using the Tuxedo Self Service Portal](#)
- [Introspecting an Existing Domain](#)

Overview

Enterprise Manager for Oracle Tuxedo supports modeling Oracle Tuxedo system and applications as a cloud service (TXaaS) and supplies a set of implementation (UI, Profiles, Jobs etc.) to be managed by Enterprise Manager. TXaaS enables customers to have a unified private middleware PaaS solution across different versions of Oracle Tuxedo. Self service application administrator (SSA admin) can manage Oracle Tuxedo cloud and corresponding Oracle Tuxedo services based on requirements. Self service application users (SSA user) select proper service for provisioning and then deploy their business applications on the service instance. From cloud administration point of view, this feature supplies the cloud management functions like resource provider management, quota management, and service template definition. From end user point of view, this feature provides operations like, service instance creation, application management, and deployment, automatic scaling based on scheduled plan or metrics.

Setting Up the Tuxedo Self Service Portal

Note: Before you set up a Tuxedo service family, the common setup tasks must have been performed. See Common Setup Tasks in [Enterprise Manager Cloud Administration Guide](#) for details.

To set up the Oracle Tuxedo as a Service, go to the Tuxedo Service Setup page by following these steps:

1. Log in to Enterprise Manager as a user with the `EM_SSA_ADMINISTRATOR` role.
2. From the Enterprise menu, select **Cloud** -> **Cloud Home**. The Cloud Home page appears.
3. From the Oracle Cloud menu, select **Setup** -> Getting Started. Click the **Setup** link next to **Tuxedo** in the left panel.

From the **Getting Started: Tuxedo** page, you need to complete the the following required setup steps. Click on a tab in the left to show the related setup page in the right side detail pane.

1. [Creating a Tuxedo PaaS Infrastructure Zone](#)
2. [Creating a Tuxedo Pool](#)
3. [Configuring Tuxedo Profiles](#)
4. [Setting Up Quotas](#)
5. [Creating a Service Template](#)
6. [Configuring Request Settings](#)
7. [Configuring Chargeback](#)

Creating a Tuxedo PaaS Infrastructure Zone

A PaaS Infrastructure Zone can contain a group of hosts. Each zone can contain homogeneous resources of only one type. Each resource in a zone represents a location at which a service instance is to be deployed.

The Tuxedo PaaS zone is an out-of-box component and can share with other cloud services. The Tuxedo PaaS zone can be deleted only when none of service instance is running on any of its hosts.

To create a Tuxedo PaaS zone, follow these steps:

1. In the **Getting Started: Tuxedo** page left panel, click **PaaS Infrastructure Zone**. Then click **Create**.
2. In the **Create PaaS Infrastructure Zone: General** page, enter the target name, name, and description for the zone. Click **Next**.
3. Click **Add** to add one or more host members to the zone. The hosts that you select cannot be present in more than one PaaS Infrastructure Zone. Click **Next**.
4. In the **Create PaaS Infrastructure Zone: Credentials** page, specify the host credentials that are to be used for provisioning all targets in this zone. The credentials provided should be of global credential type (valid across the members in the zone) and owned by the current user. Click **Next**.

Note: To create new named credentials, from the Setup menu, select Security, then select Named Credentials. You can create user or password named credentials here. Other types of credentials such as SSH must be created from the Security menu and can be used here. The named credential must be the one configured as "Global preferred credential" for the host target type.

5. Specify appropriate values for the placement policy constraints based on the resources available to the host members.

The placement policy constraints are used to set maximum ceilings for any host in the zone. Placement policy values are used during deployment to filter out hosts that are overburdened. You can define the following placement policy constraints for each host:

- **Maximum CPU Utilization:** The average CPU utilization percentage for the past 7 days.
- **Maximum Memory Allocation:** The average memory allocation for the past 7 days.

Hosts exceeding these thresholds will not be considered for provisioning operations.

6. Click **Next**. In the **Create PaaS Infrastructure Zone: Characteristics** page, specify the target properties such as cost center, lifecycle status, location, and so on.
7. Click **Next**. The **Create PaaS Infrastructure Zone: Roles** page appears. A PaaS Infrastructure Zone can be made available to a restricted set of self service users with the `EM_SSA_USER` role. Click **Add** to add and select the roles that can access this PaaS Infrastructure Zone.
8. Click **Next**. The **Create PaaS Infrastructure Zone: Review** page appears. Review the details of the zone being created and click **Submit** to create the zone.

Creating a Tuxedo Pool

A Tuxedo Pool contains a set of resources that can be used to provision a Tuxedo service instance within a PaaS Infrastructure Zone. You must create separate Tuxedo pools in the PaaS Infrastructure Zone for different type of hosts.

To create a Tuxedo Pool, follow these steps:

1. In the **Getting Started: Tuxedo** page left panel, click **Tuxedo Pool**.
2. Click **Create**. The **Create Tuxedo Pool: General** page appears. Enter a name and description for the Tuxedo Pool. Select the PaaS Infrastructure Zone in which the Tuxedo pool is to be created. Click **Next**.
3. In the **Members** page, select the platform of the hosts you are going to add. The Host table filters all the hosts running on the selected platform.
4. Click **Add** to add one or more host members to the pool. Click **Next**.
5. In the Create Tuxedo Pool: Placement Constraints page, enter the maximum number of tuxedo domain instances that can be deployed to a host target. Click **Next**.
6. In the Characteristics region, specify directory under which the Tuxedo products and related applications are to be installed and a NFS location used to deploy the JES domain that requires NFS directory for JESROOT configuration.
Note: For the existing domain, if `IsHostExclusive` of the domain package was configured to `true`, you can choose to keep all primary directory structure and deploy to the same directories on the target host.
7. In the **Target Properties** region, specify the characteristics that will be used to tag and identify the target properties. Click **Next**.
8. Review the information and click Submit to create the Tuxedo pool. The newly created pool can now be used to create Tuxedo Service Templates.

Configuring Tuxedo Profiles

Provision profiles are designed to provide the Tuxedo domain deployment package for provisioning service instance and creating one kind of Tuxedo installation. By use of the provision profile, user can collect necessary software installation into an image and distribute it to other hosts avoiding install and configure all these software from the beginning.

TSAM Plus supports packaging Tuxedo products, customer domain deployment packages, and Oracle DB client libraries in a profile. All the softwares required by running the domain must be included in the domain package.

For products, the Tuxedo installation media must be uploaded from local machine.

For customer applications, the whole Tuxedo domain should be introspected from target hosts. The same directory structure is kept during Tuxedo domain redeployment.

Creating a Tuxedo Profile

The Tuxedo profile uses the Tuxedo domain package to provision service instances. Before creating a Tuxedo profile, you must have created the desired domain package. You can create it from scratch or from an existing domain using the introspection feature in the Enterprise Manager Console. For more information, refer to [Introspecting an Existing Domain](#).

To create a Tuxedo profile, follow these steps:

1. In the **Getting Started: Tuxedo** page left panel, click **Tuxedo Profile**.
2. In the Tuxedo: Tuxedo Profile page, click **Create**
3. Enter a name and description for the profile, and select the platform the product installation media supports. Click **Next**.
4. Click **Add** and select the product you want to add.

You can add more than one products to a profile for provisioning. Only one tuxedo installation media is needed.

Oracle Tuxedo, Oracle Tuxedo Application Runtimes, Oracle Tuxedo Patch, and Other Products can be configured in the profile. Default installation scripts (Perl script) is provided for installing these products. You can specify the default scripts for special installation purpose. When configuring other products, installation location and environment variables are required to tell provisioning procedure where the product to be installed and what runtime environment variables should be exported to applications. PROD_DIR,ORA_HOME, and BASE_DIR can be used as the marco in runtime variables. The macro format is `${macro name}`.

For example, for ART CICS application, COBOL-IT is mandatory and the third party software must be installed before Oracle Tuxedo Application Runtime installation. Following parameters must be specified when configuring COBOL-IT installation.

Name: COBOL-IT_Package

Display Name: COBOL-IT Package

Setting Up Oracle Tuxedo as a Cloud Service

Description:

Install Location:cobol

Environment Variables:

```
export COBOLITDIR=${PROD_DIR}
export LD_LIBRARY_PATH=${COBOLITDIR}/lib:${LD_LIBRARY_PATH}
export PATH=${COBOLITDIR}/bin:$PATH
export COB_ENABLE_XA=1
```

Note: \${ORA_HOME} is used as the default home name in Oracle inventory. it must not be used before.

You need to configure the installation scripts if the default scripts cannot meet your installation requirements.

Other products installation media must be in the .zip format without containing a root folder.

For example:

```
COBOL-IT.zip
.
|-sbin
|-samples
|---test
|---sort
|---screens
|---customer
|-docs
|-doc
|-share
|---icucit
|-----51.1
|-----config
|---locale
```

If two products depend on each other, you need to customize the scripts to install them properly.

Click **Save** when you complete editing of the scripts. Click **Next**.

5. (Optional) If you want to deploy the Tuxedo domain, add a domain package to the profile and click **Next**. Currently, only one domain package can be configured.
6. Review the information and click **Submit** to create the Tuxedo profile.

You will find the profile status shows Ready if the Tuxedo profile has been created successfully.

Setting Up Quotas

Quota refers to the total machines of a Tuxedo domain that can be allocated to each self service user belonging to a certain role. This quota applies only to the service instances provisioned through the Tuxedo Self Service Portal. Quota is enforced at the time of self service request.

If the current quota is reduced, this change in quota does not affect existing service instances. The new quota applies only to new self service requests.

To define a quota, follow these steps:

1. In the **Getting Started: Tuxedo** page left panel, click **Quotas**.
2. In the Create Quota page, select the Role Name for which the quota is to be granted.
3. Specify the maximum number of the Tuxedo machines that can be configured for the user with this role. Click **OK**.

Creating a Service Template

A service template can be used to provision Tuxedo service instances into one or more Tuxedo Pool. When creating a service template, you need to define a provision profile to provide product installation media, default configurations and corresponding Tuxedo domain deployment packages. So a Tuxedo provision profile must have been created beforehand.

To create a service template, follow these steps:

1. In the **Getting Started: Tuxedo** page left panel, click **Service Templates**.
2. Click **Create**. Enter a name and description for the service template.

Note: If you want to create a new template with similar configuration settings of an existing service template, you can get a quick start by selecting that service template in the list and click **Create Link** to launch the creating wizard with all configured parameters loaded.

3. In the Provisioning Profile field, click the search icon and select a Tuxedo provisioning profile.

The Tuxedo provisioning profile you select can contain either of the following:

- Only the Tuxedo products. This type of profile just packages the Tuxedo installation media.
- Tuxedo application binaries besides Tuxedo installation. In this case, entire Tuxedo configurations, scripts, and binaries are included in the profile in the format of Tuxedo domain package.

The details of the selected profile are displayed when a profile is selected.

4. Click **Next**. In the Characteristics page, specify the characteristics that will be used to tag and identify the target properties. Service instances created from the template will have the characteristics.
5. Click **Next**. In the Resource Providers page, Click **Add** to select the Tuxedo Pool into which the service instances are to be provisioned.
6. Click **Next**. In the Configuration page, specify the parameters that will be used to provision service instances according to onscreen descriptions.

The Tuxedo domain configuration options displayed here vary depending on the placeholders configured in the domain package definition.

Before creating the service instance, these parameters can be overridden by the SSA user if the parameter is not locked or hidden by the SSA administrator.

To hide or lock the parameter to the SSA user, select the Hide checkbox or click the Lock icon.

Note: If no domain package is defined in the provisioning profile or just a number of hosts with necessary products installed are provided, you can set the "Only Products" option to true and specify the host number. In such case, only the specified number of hosts are reserved and installed when requesting a service instance.

7. Click **Next**. In the Roles page, click **Add** to select the self service user roles to which this service template will be available. All users belonging to the selected role can use this service template.

Note: : If no role is added here, the SSA user will fail to request a service due to lack of quota.

8. Click **Next**. Review the information entered so far and click Submit.

You will find the newly created service template is displayed in the Service Templates page. You can click on the Service Template Name link to view additional details.

Configuring Request Settings

You can configure the request settings by specifying when a request can be made, its duration, and so on. To configure the request settings, follow these steps:

1. In the **Getting Started: Tuxedo** page left panel, click **Request Settings**.
2. Specify the following parameters to set the global settings or settings for a specific role:
 - Future Reservation: How far in advance can a request be made.
 - Request Duration: Maximum duration for which a service can exist.
 - Archive Policy: Maximum duration after which archived requests are automatically deleted.
 - Service Instance Approval: Select this checkbox if all requests need to be approved by the self service administrator.
 - Service Instance Share Settings: A service instance can be shared with another user or a role. You can select either or both the checkboxes:
 - Allow SSA User to share service instances with other Users.
 - Allow SSA User to share service instances with Roles.
3. Click **Apply**.

Configuring Chargeback

Note: Before using Chargeback, it is assumed that you have read Chargeback Administration in [Enterprise Manager Cloud Administration Guide](#). Make sure that you have deployed Oracle Consolidation Planning and Chargeback plugin. If Oracle Consolidation Planning and Chargeback is absent, deploy it following Managing Plug-Ins in [Enterprise Manager Cloud Administration Guide](#).

There is only one Chargeback Entity Type for Tuxedo, Oracle Tuxedo Machine. Therefore, Tuxedo Chargeback is based on Tuxedo Machine target instance number.

To create a Chargeback for the Tuxedo Machine, follow these steps:

1. Click Configure Charge Plan link to go to Charge Plans page to create a Chargeback for Tuxedo Machine.

2. Click Create drop down list, select **Plan** to create a new Tuxedo chargeback plan.
3. Click **Add** under Entity Types. In the **Add Entity Types** window, make sure Oracle Tuxedo Machine is selected.
4. In the Oracle Tuxedo Machine page, click **Add** to add Chargeback items.
5. Click **Configure Cost Center** to configure cost center. For more information, see Chargeback Administration in Enterprise Manager Cloud Administration Guide.
6. Click **Configure Targets** to assign the Chargeback plan to an existing Tuxedo Machine.

You can choose the target instances of Tuxedo Cloud Service, Tuxedo Domain, or Tuxedo Machine. If you choose target instances of either Tuxedo Cloud Service or Tuxedo Domain, only the Tuxedo Machines inside the Tuxedo Cloud Service or Tuxedo Domain are used to calculate total costs.

Using the Tuxedo Self Service Portal

Self Service Application users can use the Tuxedo Self Service Portal to create and control Tuxedo service instances, and monitor Tuxedo services.

Note: Before using the Tuxedo self service portal, the `EM_SSA_USER` role must have been configured in the Paas Infrastructure Zone, Quotas, and Service Templates definition.

To view the Tuxedo Self Service Portal, log in to Enterprise Manager as a user with `EM_SSA_USER` role. Select **Enterprise -> Cloud -> Self Service Portal**, and click **Tuxedo Cloud Services**.

In the Tuxedo Cloud Services page, the top region shows the following icons:

- **Instances:** This icon shows the number of service instances that have been created by the current SSA user and their status.
- **Expiry:** The number of instances due to expire today, this week, and in the future.
- **Usage:** Shows the service instance number in normal status.

From the tabs in the left of the page, you can also enter the following pages:

- **Requests:** Click on this tab to view a list of all requests. For each request, the status, type, start date, and submission date of the request is displayed. Click on the status icon to view the execution details of the request. These details are helpful in troubleshooting a failed request

Select columns from the View menu to view more details about the request. You can select a scheduled request and click **Reschedule** to reschedule the request. To delete a request, select the request and click Delete.

- **Chargeback:** Click the Chargeback tab to view the charge summary information. For more information, see [Chargeback](#).
- **Policies:** Click the Policies tab to view the list of policies that you have defined. You can create a new Performance or Schedule Based Policy or a Policy Group for one or more Tuxedo services and define Scale Up or Scale Down on these policies. For more information, see [Defining Policies for Tuxedo Services](#).

Requesting a Tuxedo Service

To request a Tuxedo service, follow these steps:

1. Click **Create Instance** in the Home page of the Tuxedo Cloud Self Service Portal.
2. The Service Catalog - Create Instance window appears. A list of service templates available to the self service user is displayed. Choose a service template and click **Select** to launch the Create Instance wizard.
3. Enter a name for the instance and select the Self Service Zone in which the new service is to be deployed.
4. In the **Instance Duration** region, specify the schedule for the instance.
 - **Creation Date:** You can choose Immediately (request is executed immediately) or Later. If you select the Later option, specify the date from which the service is to be available.
 - **End Date:** Select Indefinitely to make the service available for a indefinite period of time. If you select Until, specify the date on which it will be deleted and no longer available.
5. Click **Next**. The Characteristics page appears.
6. In the Target Properties region, specify the characteristics that will be used to tag and identify the target properties. Service instances created from the template will have the characteristics.
7. Click **Next**. The Configuration page appears.
8. Specify the Tuxedo domain externalizable variables, service provisioning mode, and port range according to the description of each option in the page. Click **Next**.

Note:

This step settings overwrites the parameters configured in the Service Template.

You can configure a Tlisten password which is used for security connections between tlisten processes in a MP mode domain.

9. Review the information entered so far and click **Submit** to create the service request.

This request will now appear in the Request page in the Tuxedo Self Service Portal. You can monitor the status of this request in this table. When the request has been completed, you can refresh the page to view the new service in the Instances table.

Notes:

- The user with the EM_SSA_ADMIN role cannot create service instances.
- The SSA users having no quota cannot make a service request.

Controlling the Tuxedo Service

In the Instances page, the Instances table shows the list of Tuxedo services owned by the self service user and any other services for which the user has view privileges. You can perform the following actions when selecting a service from the table:

- **Delete:** Delete the service instance. The entire domains associated to this service instance will be stopped and all applications will be deleted from each host that has been deployed before. This action does not uninstall the Tuxedo installation media.
- **Extend:** Extend the life of the service instance. You are prompted for the End Date. Enter a date later than the current one and click OK to extend the service instance.
- **Start:** Start the service and boot the corresponding Tuxedo domain.
- **Stop:** Stop the service and shut down the corresponding Tuxedo domain.
- **Grant:** You can grant view, manage, or full privileges over the service instance to other users and roles.
- **Revoke:** You can revoke privileges that you have granted to other users and roles.

Viewing the Tuxedo Service Home Page

The Tuxedo Service Home page provides detailed information about the services you have requested. To view the Tuxedo Service Home page, click the Name link of the Tuxedo service instance in the service instance table. This page contains the following regions:

- Top action menu: it provides the following actions:
 - Set End Date: Specify an end date for the Tuxedo service.
 - Start and Stop: Start or stop the service.
 - Scale Up and Scale Down: Increase or reduce the number of Tuxedo machines.
- Summary: Shows the availability status of the service, status summary of the Tuxedo Servers.
- Tuxedo Domain List: Lists Tuxedo domains.
- Requests: Lists the requests against this Tuxedo service instance in 24 hours.

Chargeback

The Chargeback page displays charge information for Tuxedo machines in login SSA user. Only the Tuxedo Machine target instances associated with predefined Tuxedo Charge Plan are monitored here. For more information, see [Configuring Chargeback](#).

Top region shows total charge amount. From the bottom region you can view the Charge Trend in graph and details in table.

Defining Policies for Tuxedo Services

To define a policy, click the Policies tab. From the Create menu, you can select either of the following:

- Schedule Based Policy
- Performance Based Policy

Creating a TXaaS Schedule Policy

For instance, if you want to create a schedule based policy that states that a tuxedo service must be scaled up every Monday, follow these steps:

1. Click the **Policies** tab in the Tuxedo Self Service Portal.
2. Click **Create** and select **Schedule Policy**.
3. Enter a name and description for the policy.
4. Select the Policy Action. There are four actions:

- Scale Down Tuxedo Service Instance
 - Scale Up Tuxedo Service Instance
 - Start Tuxedo Service Instance
 - Stop Tuxedo Service Instance
5. In the **Start** field, specify the date from which this policy must be effective.
 6. In the **Repeat** field, specify **Every N** minutes/hours/days/weeks and then specify **N**.
 7. Click **OK** to create the policy. You will return to the Policies page where the policy you have defined will appear in the Policies table. To apply these policy for a target, click **Add** in the **Associations** region and select a target with which this policy must be associated.

Note: TSAM Plus 12.2.2 does not support scaling up the Jolt domain.

Defining a TXaaS Performance Policy

Performance policies are formed based on the performance metrics of a specific target type and its associated targets. You can create conditions and rules for the policy based on the metrics. To define a performance policy, follow these steps:

1. Click the **Policies** tab in the Tuxedo Self Service Portal.
2. Click **Create** and select the **Performance Policy** type.
3. In the General Information page, enter the name and description of the policy. Click **Next**.
4. In the Create Performance Policy: Rule page, select a metric from the right panel and click **Add to Definition**. Specify the condition, value, and number of occurrences. When these conditions are met, the policy is executed. Click **Next**.
5. Select one of the following actions from the **Action** drop down list. Click **Next**.
 - Scale Down Tuxedo Service Instance
 - Scale Up Tuxedo Service Instance
 - Start Tuxedo Service Instance
 - Stop Tuxedo Service Instance
6. Review the policy details and click **Save** and return to the Policies page.
7. In the Associations region, click **Add** and select one or more targets to be associated with this policy. The associated targets are listed in the **Associations** region.

Click on the target link to view a list of policies associated with the target. Once the targets have been associated, the policy is activated. Click **View History** to view the policy action history.

Introspecting an Existing Domain

The Tuxedo profile uses the Tuxedo domain package to provision service instances. Before creating a Tuxedo profile, you must have created the desired domain package. You can create it from scratch or from an existing domain using the introspection feature in the Enterprise Manager Console or introspecting command.

This section contains the following topics:

- [Introspecting an Existing Domain From Console](#)
- [Introspecting an Existing Domain Using Command Line](#)

Introspecting an Existing Domain From Console

To introspect an existing domain into a domain package from Enterprise Manager Console, follow these steps:

1. Create a new introspection project:
 - a. From the Enterprise Manager Console menu, click **Targets > Middleware** at the top.
 - b. In the Middleware page, click **Middleware Features > Tuxedo Summary**.
 - c. In the Oracle Tuxedo Summary page, click **Introspect**.
 - d. Click **New** to create a new introspection project and select one of the domains monitored by Enterprise Manager from the list.

The master machine information is listed at the bottom. If the host on which the master machine deployed are configured with the preferred credential, the host credential will be filled automatically.

Alternatively, if you want to introspect a tuxedo domain not monitored by Enterprise Manager, fill the necessary machine configuration parameters in the bottom panel.

Note that any host can be the candidate to retrieve the UBB.

- e. Click **OK** to retrieve the tuxedo domain UBB configurations.

When the tuxedo domain UBB configuration is successfully retrieved from target host/machine, each tab page shows the machine configuration of this domain. You can

configure “Third Party Software” which is required by the domain but outside of APPDIR. Such binary/configuration file must be added in the right table list one by one. When packaging the machine binary package, these file/folder will be collected together with all contents under APPDIR.

2. Click **Package >Introspect** to invoke the introspecting process. When the process completes successfully, the Confirmation window with the result and domain package file name is displayed.
3. Click **Package >Download** to download the desired domain package to the local machine.
4. (Optional) Click **Save** to export the newly created introspection project parameters to the local computer. The file is exported as XML format without a user password. The file can be used to introspect the domain under the command line mode as well.
5. (Optional) Click **Open** to upload a local project file saved before.

After the project file is successfully uploaded and parsed, all the introspection parameters are shown.

Introspecting an Existing Domain Using Command Line

Introspection Command

Name

```
java -jar TxdoDomainPackager.jar
```

Synopsis

```
java -jar TxdoDomainPackager.jar -prop property file [-template]
```

Options

prop

Required. Full path to the property file which contains introspection parameters and corresponding values.

template

Optional. Creates a property file template to let you focus on specifying necessary values.

Examples

```
java -jar TxdoDomainPackager.jar -template
```

A property file template `InputParamsTemplate.xml` is created under the location where `java` command invoked.

Edit the template `InputParamsTemplate.xml` to suit your environment and save it as `params.xml`.

```
java -jar TxdoDomainPackager.jar -prop params.xml
```

A binary package zip file is created under the location where `java` command invoked.

Property File Template Format

Listing 8-1 Property File Template

```
<?xml version="1.0" encoding="UTF-8"?>
<CmdOptions>
<General>
<!--if these options not configured in general section, then configure these
items should be configured to each host section
, especially for those mandatory items, i.e. AppDir,TuxDir,
TuxConf,User,Pwd.
Default value of "IncludeLocal" is false.
-->
    <AppDir>[application directory]</AppDir>
    <TuxDir>[Description of this package]</ TuxDir >
<JavaHome>[full path of java home]</ JavaHome>
<TuxConf>[tuxedo configuration full path]</ TuxConf >
<EnvFile>[Set runtime environment file]</EnvFile>
<DmConf>[tuxedo configuration full path]</ DmConf >
<SaltConf>[tuxedo configuration full path]</SaltConf >
<TnsNamesLocation>[tuxedo configuration full path]</ TnsNamesLocation>
<OraDbClientDir> [Oracle DB client Directory] </OraDbClientDir>
<IncludeLocal>[true or false]</ IncludeLocal>
```

Setting Up Oracle Tuxedo as a Cloud Service

```
<User>[user which can be shared with all hosts]</User>
<Pwd>[password which can be shared with all hosts]</Pwd>
</General>
<Hosts>
  <Host>
    <NetAddress></NetAddress>
    <Softwares>
      <Software> <!--can be multiple -->
        <Name>[Name of this software]</Name>
        <Location>[full path where is this software]</Location>
      </Software>
    </Softwar>
```

[Following options can be specified to override the same one configured in general section]

```
  <User></User>
  <Pwd></Pwd>
  <AppDir/>
  <TuxDir/>
  <JavaHome/>
  <WorkingDir />
</TuxConf/>
<DmConf/>
<SaltConf/>
<TnsNamesLocation/>
<IncludeLocal/>
<JavaHome/>
  </Host>
```

Following is a property file example:

Listing 8-2 Property File Example

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<CmdOptions>
  <General>
    <TuxDir>/home/yfli/oracle/tux_oh/tuxedo12.1.3.0.0</TuxDir>
    <JavaHome>/home/yfli/tools/jdk1.7.0_55</JavaHome>
    <User>yfli</User>
    <Pwd></Pwd>
  </General>
  <Hosts>
    <Host>
      <NetAddress>localhost</NetAddress>
    </Host>
  </Hosts>
  <TnsNameLocation>/home/yfli/tools/ora_client/12.1.0/client_1/tnsnames.ora<
  /TnsNameLocation>
  <OraDbClientDir>/home/yfli/tools/ora_client/12.1.0/client_1</OraDbClientDir>
  <AppDir>/home/yfli/apps/tux1213/domapp/domain1</AppDir>
  <TuxConf>/home/yfli/apps/tux1213/domapp/domain1/tuxconfig</TuxConf>
  <DmConf>/home/yfli/apps/tux1213/domapp/domain1/bdmconfig</DmConf>
  <SaltConf>/home/yfli/apps/tux1213/domsalt/saltconfig</SaltConf>
  <EnvFile>/home/yfli/apps/tux1213/domsalt/setenv.sh</EnvFile>
  <Softwares>
    <Software>
      <Name>test1</Name>
    </Software>
  </Softwares>
</CmdOptions>
```

```
        <Location>/home/yfli/apps/tux1213/cics</Location>
    </Software>
<Softwares>          </Host>
<Host>
    <NetAddress>... .cn.oracle.com</NetAddress>
    <JavaHome>/home/yfli/tools/jdk1.7.0_55</JavaHome>
    <AppDir>/home/yfli/apps/tux12130/mp</AppDir>
    <TuxDir>/home/yfli/oh/tux12130</TuxDir>
    <TuxConf>/home/yfli/apps/tux12130/mp/tuxconfig</TuxConf>
    <User>yfli</User>
    <Pwd> </Pwd>
    <Softwares>
<Software>
        <Name>cmds</Name>
        <Location>/home/yfli/cmds</Location>
    </Software>
</Softwares>          </Host>
</Hosts>
</CmdOptions>
```

Application/Library Package

A binary package zip file is created during introspection process for each of machine configured in the Tuxedo domain. This package includes all the applications, libraries, and all third party products that the domain needs. The file is named [lmid].zip and is configured in the Tuxedo domain package configuration file. For example, when introspecting the machine "LMID1" on a host, its binary package in the configuration file may look like [Listing 8-3](#):

Listing 8-3 Binary Package in Configuration File

```

<?xml version="1.0" encoding="UTF-8"?>
<DomainPackageConfig>
...
<PackagesUsed>
    <BinaryPackage>
        <ArchiveFormat>ZIP</ArchiveFormat>
        <SubFolders>
            <FolderInfo>
                <Name>APPDIR</Name>
                <Destination/>
                <IsShared>>false</IsShared>
            </FolderInfo>
        </SubFolders>
        <Name>SITE1.zip</Name>
        <Description/>
        <Location/>
        <Destination/>
    </BinaryPackage>
    ...
</PackagesUsed>
...
</DomainPackageConfig>

```

The zip file includes following contents which are all copied from APPDIR:

```
`-- APPDIR
```

```
|-- java.env
|-- setenv.sh
|-- simpcl
|-- simpserv1
|-- starttlisten
|-- tuxconfig
|-- tuxconfig.decompiled
|-- ubb
`-- ubb.test
```

Note that the configuration files in this folder will be overwritten by the ones in config_LMID1 folder if those files destination location is APPDIR.

Limitation

When introspecting an existing domain running on the different machine with OMS, the machines configured in the domain can not post message in the `.bashrc` configuration file. This causes SFTP connection failure.

Placeholders File Format and Usage

Placeholders are used to replace old values in configuration files. These placeholders can be treated as "Variables" which will be filled with the real value at deployment time. All the placeholders are defined in the package configuration file. [Listing 8-4](#) shows the definitions.

Listing 8-4 Placeholders Definition

```
<Name></Name> // this is the variable name used
in the configuration file in the format of ${name}

<Type> </Type> // Indicates that the macro is
well-known or not

<Label> </Label> //Label name that will displayed
on GUI

<Value> </Value> //Value of this
placeholder delegate
```

```

    <Description/>                                //Hint messages that will
displayed on GUI

    <IsSecurity></IsSecurity>                       //The value should be inputted
by a security text box

    <OriginalValue></OriginalValue>                //Original value in the configuration
file

```

Placeholders can be used in any configuration and script files which will be handled during provisioning procedure. The placeholder in these files is in format of `${placeholder}`.

Listing 8-5 Placeholders Example

```

<?xml version="1.0" encoding="UTF-8"?>
<DomainPackageConfig>
...
<Placeholders>
<Placeholder>
    <Name>SITE1-tsam-manager-host</Name>
    <Type>UBB_TSAM_ADDR</Type>
    <Label>TSAM Manager Address</Label>
    <Description/>
    <IsSecurity>>false</IsSecurity>
    <OriginalValue>10.182.55.120</OriginalValue>
</Placeholder>
...
</Placeholders>
...
</DomainPackageConfig>

```

UBB, DM, SALT, and Oracle TNS configuration files are parsed during introspection, and replaced with placeholders at deployment time automatically. [Listing 8-6](#) lists the well-known placeholder types:

Listing 8-6 Well-known Placeholder Types

```
IPC_LOCAL           //IPC key which will be filled value automatically
NET_PORT           //Net port which will be filled value
automatically
    LOCAL_NET_ADDR //Local net address which will be filled value
automatically
UBB_MACHINE_NAME   //real host name of machine
    UBB_NET_ADDR   // net address
    UBB-NLS_PORT   //NLS net port
    UBB_NA_PORT    // NA net port
JMX_PORT           //JMX net port
JMX_ADDRESS        //JMX net address
DM_LOCAL_ADDRESS   // Domain gateway local net address;
DM_LOCAL_PORT     // Domain gateway local net port;
DM_REMTOTE_PORT   // Domain gateway remote net port;
DM_REMOTE_ADDRESS // Domain gateway remote host address;
UBB_DOMAIN_IPCKEY //Domain IPC Key
UBB_DOMAIN_ID     //Domain ID
DOMAIN_PWD        //Domain Application password
DOMAIN_USER       //User name if domain configured
with ACL/Mandatory ACL
DOMAIN_USER_PWD   //User password
GLOBAL_VAR        //Global variable
CUSTOMIZED        //User defined placeholder
```

The placeholders assigned with the type of UBB_MACHINE_NAME, UBB_NET_ADDR, UBB-NLS_PORT, UBB_NA_PORT, JMX_PORT, JMX_ADDRESS, DM_LOCAL_ADDRESS, DM_LOCAL_PORT are handled automatically. That is, their values are filled by provisioning procedure. For those placeholders assigned with the type of "GLOBAL_VAR", the value can be embedded with other placeholders. See the example shown in [Listing 8-7](#).

Listing 8-7 Placeholders with GLOBAL_VAR

```
<?xml version="1.0" encoding="UTF-8"?>
<DomainPackageConfig>
...
<Placeholders>
<Placeholder>
    <Name>SITE1-G_VAR</Name>
    <Type> GLOBAL_VAR </Type>
    <Label>TSAM Manager Address</Label>
    <Value>${SITE1-ADDRESS}:${SITE1-PORT}</Value>
    <Description/>
    <IsSecurity>>false</IsSecurity>
    <OriginalValue>10.182.55.120</OriginalValue>
</Placeholder>
...
</Placeholders>
...
</DomainPackageConfig>
```

When manually making a domain package instead of using introspecting tool, you can use the following placeholders in the configuration and scripts files directly:

- `lmid-APPDIR`
- `lmid-TUXDIR`
- `lmid-ORAHOME`

`lmid` is the logical name of the machine configured in UBB.

For those automatically handled placeholders, for example, `${SITE1-machine-name}` is filled with a real host name at provisioning time without any input from end users. But in the case that there is no proper value to replace the placeholder, you need to input the values. For example, `${GWDOMAIN3-remote-port}` in DM CONF will be filled with proper value to the best extent possible. If no proper value can be determined automatically, you need to input the value. Here is an example that shows the process of how to handle such placeholders.

Suppose two domains are connected with each other through domain gateway and both are included in a provisioning profile. Domain 1 connects to WTC which will be viewed as domain 3. The DM configuration will look like this:

Listing 8-8 Domain 1 DM configuration-dm1.conf

```
*DM_RESOURCES

*DM_LOCAL

"GWDOMAIN1 "      GWGRP="SYSGRP"

                  ACCESSPOINTID="GWDOMAIN1 "

                  ...

*DM_REMOTE

"GWDOMAIN2 "      ACCESSPOINTID="GWDOMAIN2 "

                  CREDENTIAL_POLICY="LOCAL "

"GWDOMAIN3 "      ACCESSPOINTID="GWDOMAIN3 "

                  CREDENTIAL_POLICY="LOCAL "

*DM_TDOMAIN

"GWDOMAIN1 "      NWADDR="//${GWDOMAIN1-local-addr}:${GWDOMAIN1-local-port}"
```



```

NWPROTOCOL=LLE

"GWDOMAIN2 "
NWADDR="//${GWDOMAIN2-remote-addr}:${GWDOMAIN2-remote-port}"

"GWDOMAIN3 "
NWADDR="//${GWDOMAIN3-remote-addr}:${GWDOMAIN3-remote-port}"

```

Listing 8-9 Domain 2 DM configuration-dm2.conf

```

*DM_RESOURCES

*DM_LOCAL

"GWDOMAIN2 "      GWGRP="SYSGRP"

                ACCESSPOINTID="GWDOMAIN2 "

                ...

*DM_REMOTE

"GWDOMAIN1 "      ACCESSPOINTID="GWDOMAIN1 "

                CREDENTIAL_POLICY="LOCAL"

*DM_TDOMAIN

"GWDOMAIN1 "

NWADDR="//${GWDOMAIN1-remote-addr}:${GWDOMAIN1-remote-port}"

                NWPROTOCOL=LLE

"GWDOMAIN2 "      NWADDR="//${GWDOMAIN2-local-addr}:${GWDOMAIN2-local-port}"

```

When handling dm1.conf at provisioning time, the assembly procedure attempts to find the host in Domain 2 UBB configuration (find it by access point) by comparing original net address value. If the host is successfully found, the new address will be filled `${GWDOMAIN2-remote-addr}`, but if domain 2 is not included in the profile or such host can not be found in domain 2 UBB, then you need to provide this value before submitting the provisioning procedure, for example, provide the value for `${GWDOMAIN3-remote-addr}`.

Following configuration files are introspected and handled automatically:

Setting Up Oracle Tuxedo as a Cloud Service

- UBB CONF
- DM CONF
- SALTCONFIG
- Oracle TMS configuration file

UBB sample assembled with placeholders:

Listing 8-10

```
...
*MACHINES
"${SITE1-machine-name}" LMID="SITE1"
    TUXCONFIG="${SITE1-APPDIR}/tuxconfig"
    TUXDIR="${SITE1-TUXDIR}"
    APPDIR="${SITE1-APPDIR}"
    TLOGNAME="TLOG"
    TLOGSIZE=100
    MAXWSCLIENTS=20
"${SITE2-machine-name}" LMID="SITE2"
    TUXCONFIG="${SITE2-APPDIR}/tuxconfig"
    TUXDIR="${SITE2-TUXDIR}"
    APPDIR="${SITE2-APPDIR}"
    TLOGNAME="TLOG"
    TLOGSIZE=100
...
*SERVERS
LMS"    SRVGRP="GROUP1"
        SRVID=100
```

```

CLOPT="-A -- -1
${SITE1-tsam-manager-host}:${SITE1-tsam-manager-port}/tsam"

RQPERM=0666

REPLYQ=N

RPPERM=0666

MIN=1

MAX=1

...

```

Introspecting ART Applications

When introspecting ART applications, following parameters and files are collected and zipped into a domain package.

ART CICS Introspection

Description	Source	Type	Placeholder	Add to Package
ARTTCPL -n netaddr -S ssladdr -L pnetaddr	CLOPT	Network address	Y	n/a
ARTTCPL [-p profile-name]	CLOPT	File location	Y	N
ARTCSKL -p port	CLOPT	Network address	Y	n/a
DD_VSAMFILE	Env Variable	File location	Y	Y
KIXCONFIG	Env Variable	File location	Y	Y
KIXDIR	Env Variable	File location	N (auto set to ARTCICS install dir)	N

Description	Source	Type	Placeholder	Add to Package
KIX_CWA_IPCKE Y	Env Variable	IPC Key	Y	n/a
KIX_DUMP_FILE	Env Variable	File location	Y	N
KIX_JOURNAL_D IR	Env Variable	File location	Y	N
KIX_MAP_PATH	Env Variable	File location	Y	Y
KIX_QSPACE_IP CKEY	Env Variable	IPC Key	Y	n/a
KIX_SHR_IPCKE Y	Env Variable	IPC Key	Y	n/a
KIX_SPOOL_OUT PUT_DIR	Env Variable	File location	Y	N
KIX_TD_DIR	Env Variable	File location	Y	N
KIX_TD_QSPACE _IPCKEY	Env Variable	IPC Key	Y	n/a
KIX_TECH_DIR	Env Variable	File location	Y	N
KIX_TRACE_PAT H	Env Variable	File location	Y	N

ART Batch Introspection

Description	Source	Type	Placeholder	Add to Package
JESDIR	Env Variable	File location	N (Batch RT install dir)	N
\$JESDIR/ejr/ CONF/*				Y
PROCLIB	Env Variable	File location	Y	N

Description	Source	Type	Placeholder	Add to Package
MT_ACC_FILEPATH	Env Variable	File location	Y	N
MT_CTL_FILES	Env Variable	File location	Y	N
MT_KSH	Env Variable	File location	N (pdksh dir)	N
MT_DB_LOGIN	Env Variable	User/passwd	Y	n/a
MT_DB_LOGIN2	Env Variable	User/passwd	Y	n/a
MT_GDG_DB_ACCESS	Env Variable	User/passwd	Y	n/a
MT_LOG	Env Variable	File location	Y	N
MT_ROOT	Env Variable	File location		
MT_SYSOUT	Env Variable	File location	Y	N
MT_TMP	Env Variable	File location	Y	N
DATA	Env Variable	File location	Y	N
TMP	Env Variable	File location	Y	N
SYSIN	Env Variable	File location	Y	N
MT_CONFIG_FILE	Env Variable	File location	Y	Y
ARTJESADM -i JESCONFIG	CLOPT	File location	Y	Y
JESCONFIG:JESROOT	Config option	File location	Y	N
JESCONFIG:JES_ACL_FILE	Config option	File location	Y	Y
JESCONFIG:JOBREPOSITORY	Config option	File location	Y	Y
JES_TRACE_PATH	Env Variable	File location	Y	N

ART IMS Introspection

Description	Source	Type	Placeholder	Add to Package
IMSDIR	Env Variable	File location	N (IMS RT install dir)	N
ART_IMS_CONF IG	Env Variable	File location	Y	Y
ART_IMS_DB	Env Variable	File location	Y	Y
ART_IMS_FMT	Env Variable	File location	Y	Y