

# **Enterprise Manager for Oracle® Tuxedo**

User Guide

12c Release 1 (12.1.1.1)

February 2013

Copyright © 2013, 2013, 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 the Enterprise Manager for Oracle Tuxedo

Using Web-Based Console . . . . .	1-1
Logging into Enterprise Manager Cloud Control . . . . .	1-2
Viewing Tuxedo Summary . . . . .	1-2
Actions Bar . . . . .	1-3
Search Panel . . . . .	1-4
Target Summary . . . . .	1-4
Target Home Page Introduction . . . . .	1-4
Target Navigation Tree . . . . .	1-5
Target Menu . . . . .	1-6
Content Pane . . . . .	1-6
Right-Click Target Menu . . . . .	1-7
Monitoring Tuxedo Targets . . . . .	1-7
Viewing Configuration Topology . . . . .	1-10
TuxJES Monitoring . . . . .	1-10
Oracle Tuxedo Side Configuration . . . . .	1-11
Discovery . . . . .	1-11
Login to TuxJES Monitoring Homepage . . . . .	1-11
Monitoring Metrics . . . . .	1-12
Operations on UI Console . . . . .	1-12
GDG Management . . . . .	1-13
Prerequisites . . . . .	1-13

Operations on UI Console . . . . .	1-13
Tuxedo Event Monitoring . . . . .	1-14
Reporting for Tuxedo Targets . . . . .	1-20
Using Enterprise Manager SLA . . . . .	1-20
Creating Enterprise Manager Services . . . . .	1-20
Using Resource Broker . . . . .	1-23
About Resource Broker . . . . .	1-24
Application Package Organization and Content . . . . .	1-24
Deployed Application Package Structure On Machine . . . . .	1-25
Properties.xml . . . . .	1-26
Application Package Management . . . . .	1-34
Uploading a Package . . . . .	1-34
Removing a Package . . . . .	1-34
Domain UBBCONFIG Editor . . . . .	1-35
Editor Header . . . . .	1-35
Controls Panel . . . . .	1-35
Machine List Panel . . . . .	1-37
Package List Panel . . . . .	1-37
UBB Section Panels . . . . .	1-38
Deploying Application Packages to Tuxedo Domain . . . . .	1-42
Preparing Application Packages . . . . .	1-43
Discovering Tuxedo tlisten and Tuxedo Home . . . . .	1-45
Creating a Tuxedo Domain . . . . .	1-46
Deploying the Domain to Remote Machine . . . . .	1-49
Deploying an Application Package Dynamically . . . . .	1-51
Updating a Deployed Application Package . . . . .	1-51
Removing Package Associated Groups . . . . .	1-52
Dynamic Resource Broker . . . . .	1-52

Manually Dynamic Deployment . . . . .	1-52
Policy-Driven Dynamic Deployment. . . . .	1-53
Policies . . . . .	1-54
Controls on Tuxedo Domain Target . . . . .	1-57
Deploy . . . . .	1-57
Config. . . . .	1-57
Start Up. . . . .	1-57
Shut Down . . . . .	1-57
Undeploy . . . . .	1-58
Using Resource Broker for ART Batch. . . . .	1-58
Server Configuration . . . . .	1-58
JES Configuration . . . . .	1-59
Assembling and Deploying JES Domain. . . . .	1-66
Using Resource Broker for ART CICS . . . . .	1-68
Server Configuration . . . . .	1-68
CICS Application Package. . . . .	1-70
CICS Configuration Page. . . . .	1-72
Assembling and Deployment . . . . .	1-72
Integration with OVAB . . . . .	1-73
Supported Tuxedo Version . . . . .	1-73
Supported Platform . . . . .	1-73
Scaling Oracle Tuxedo Machines Number with OVAB Script . . . . .	1-74
Deploying Tuxedo Application . . . . .	1-74
Defining Enterprise Manager Policy . . . . .	1-77



# 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 covers the following:

- [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](#)

## 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

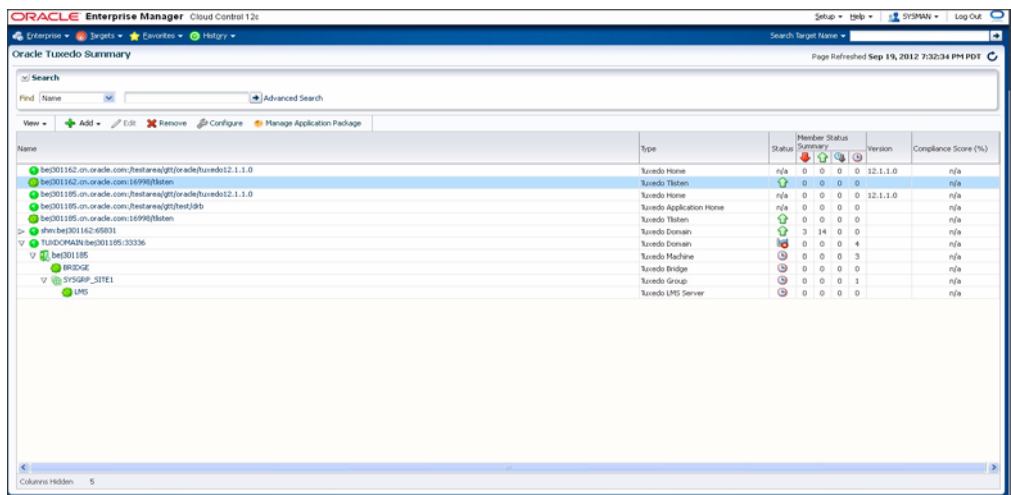
The Tuxedo Summary page filters all Tuxedo domain targets from a large number of Enterprise Manager monitoring targets, and allows you to centrally manage Tuxedo targets and easily navigate to each target home page.

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](#).

Figure 1 Tuxedo Summary Page



## Actions Bar

Table 1 lists the action bar functions.

Table 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 <a href="#">Domain UBBCONFIG Editor</a> .
Edit	This button is available only when highlighting a Tuxedo domain that is created by the Resource Broker domain editor. When you click this button, the selected domain is loaded in the <a href="#">Domain UBBCONFIG Editor</a> .
Remove	<p>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.</p> <p>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.</p>

**Table 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 2](#) lists descriptions to each summary column fields.

**Table 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 2](#) appears.

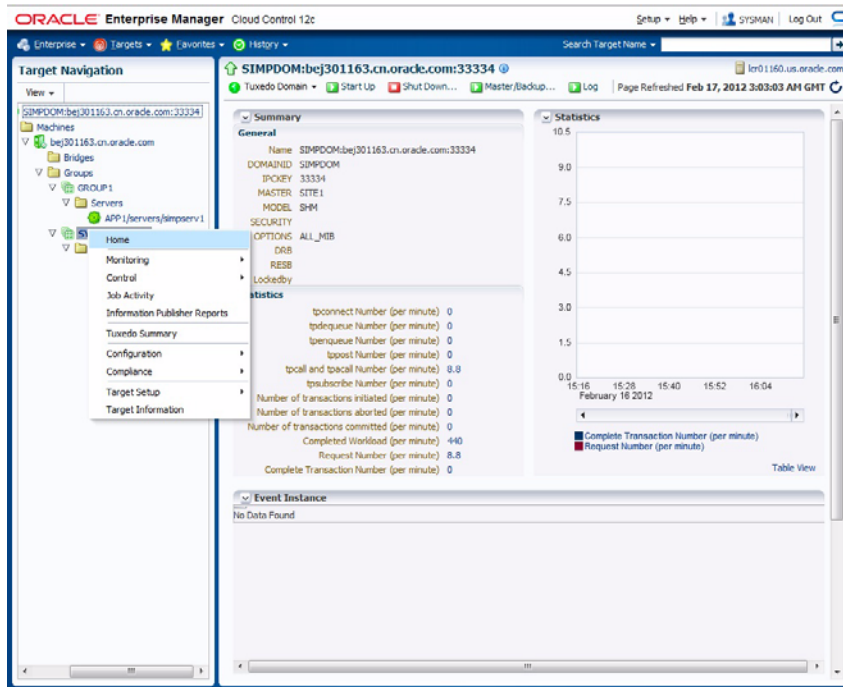
**Figure 2 Target Home Page**

Figure 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](#).

### Listing 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 3](#).

**Figure 3 Credentials**

The screenshot shows a dialog box titled "Credentials". At the top, it says "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". A message states "Select credential from one of the following options." There are three radio buttons: "Credential" (selected), "Preferred", and "Named". 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 field next to it containing the value "NC\_SHMSECBE\_2013-03-04-174828".

To configure credentials in [Figure 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 timezone:
 

```
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 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 3 Tuxedo Targets**

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

**Table 3 Tuxedo Targets**

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

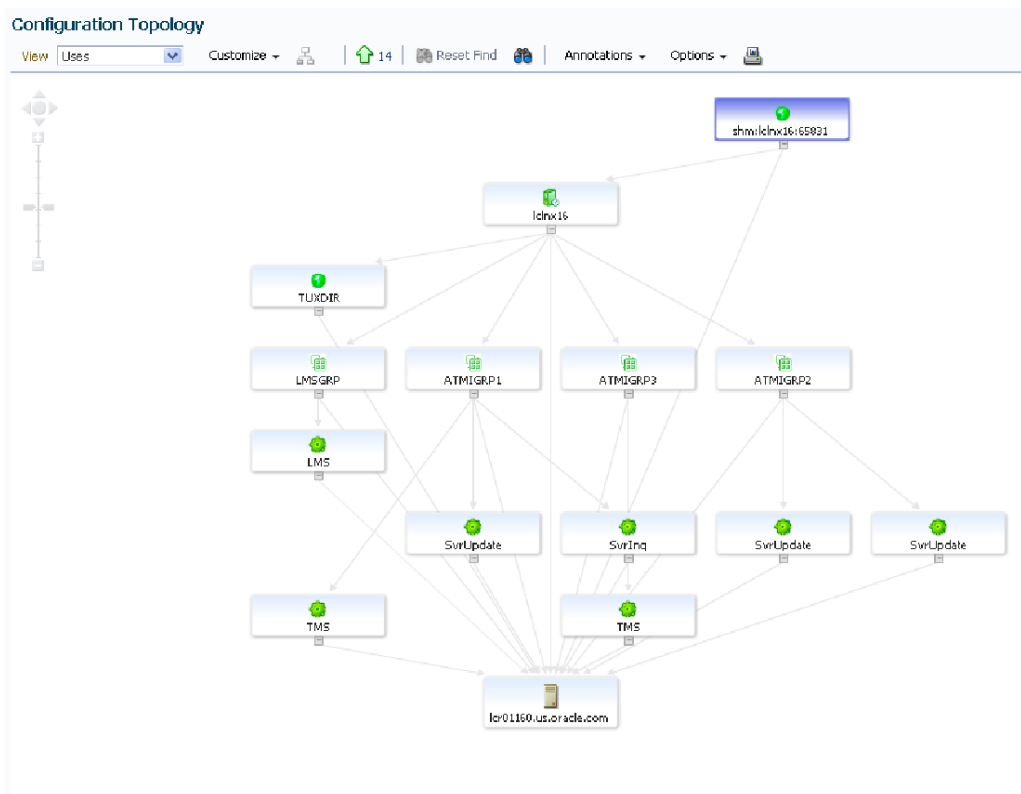
### Notes:

- When you change the collection interval of a Enterprise Manager console target, it takes effect on metrics data collection from the next interval.
- Since the domain, machine, and group targets are repository-side metrics that only process historical database content, they cannot be viewed in real time. You can only get correct real-time metrics data for target servers.

## 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 4](#).

**Figure 4 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.

Where the Trap host is the Enterprise Manager agent host, and the port is the Enterprise Manager agent listening port which configured as Batch System target monitoring property

Note the following example:

```
TRAP_HOST 10.182.54.215 1061 public
SNMP_ENABLE_AUTH_TRAP 1

TMAGENT tuxedo_agent_1 /home/tge/tuxedo12.1.1.0
/home/tge/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 `tux_snmpd`.

For more information, see [Oracle SNMP Agent Introduction](#).

## 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](#).

## 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:

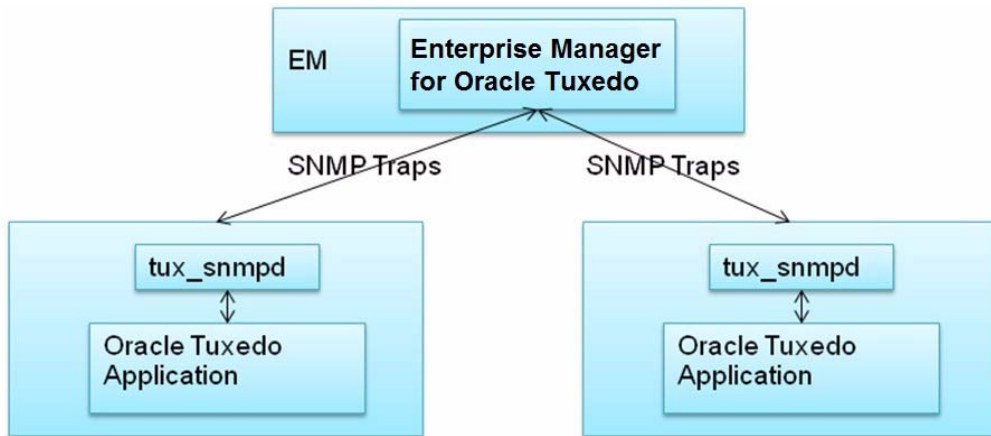
1. Make sure Enterprise Manager, ART Batch, and Oracle Tuxedo are installed.
2. Set up a Tuxedo domain (use the sample `$ART_HOME /Batch_RT/sample/simpjob` file in ART Batch). Set the environment variable `JESMONITOR=yes`.
3. Set the GDG management as DB-based in `$ART_HOME/Batch_RT/ejr/CONF/BatchRT.conf`. 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 5 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.

**Note:** For Tuxedo 11g Release (11.1.1.3.0) and Release (11.1.1.2.0), to use the User Event feature, you need to set the environment: `SNMP_USER_EVENT=y` before starting `tux_snmpd`.

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.

Note that the Trap host is the EM agent host and the port is EM agent listening port.

Following is an configuration example.

```
TRAP_HOST      10.182.54.215 1061 public
SNMP_ENABLE_AUTH_TRAP 1
TMAGENT tuxedo_agent_1 /home/tge/tuxedo12.1.1.0
/home/tge/test/jmx/fakemp/master/tuxconfig
```

3. Set the environment variable of BEA\_SM\_SNMP\_MIBFILE and BEA\_SM\_BEAMGR\_CONF.

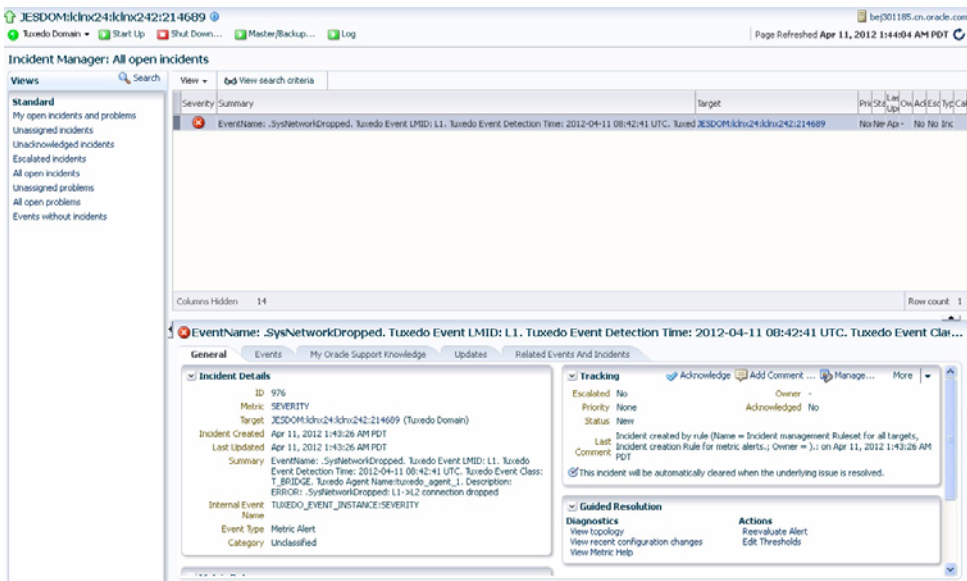
4. Start up Tuxedo SNMP agent.

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 6 shows an incident created by EM and caused by Tuxedo event trap.

**Figure 6 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 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 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
TA_EVENT_DESCRIPTION	tuxEventsDescription	Optional. String[1...1118]
TA_CLASS	tuxEventsClass	Optional. String[1...30]

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

**Listing 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 "Unix.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 )

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(Error32));
    (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 7.

**Figure 7 Services Page**

Services - Oracle Enterprise Manager - Mozilla Firefox

File Edit View History Bookmarks Tools Help

oracle.com https://bej301162.cn.oracle.com:7799/em/faces/core-groups-allSystemHome?result=8\_alfLoop=129830727949K

Services - Oracle Enterprise Manager New Tab

ORACLE Enterprise Manager

Grid Targets Favorites History

Search Target Name

Page Refreshed Oct 18, 2011 11:38:48 AM CST

Services Features

Services

Type ALL Name % Go Advanced Search

View Configure Remove Type Generic Service Add Detach

Name	Type	Status	Service Level Agreement Status	Incidents				System	Key Tests								
				Performance	Usage	Name	Status		Incidents	Status	Incidents	Monitoring Beacons					
EM Jobs Service	Generic Service	Up	Up	-	-	-	-	Management Services and Repository	Up	2	9	1	-	n/a	-	-	0
EM Monitoring Service	Generic Service	Up	Up	-	-	-	-	Management Services and Repository	Up	2	9	1	-	n/a	-	-	0
EM Console Service	Generic Service	Down	Up	-	-	-	-	Management Services and Repository	Up	2	9	1	-	1	-	-	1

2. Click **Add** to create Enterprise Manager services SLA, as shown in Figure 8 and Figure 9. For more information, see online Oracle Enterprise Manager Documentation.

Figure 8 Creating Enterprise Manager Services SLA

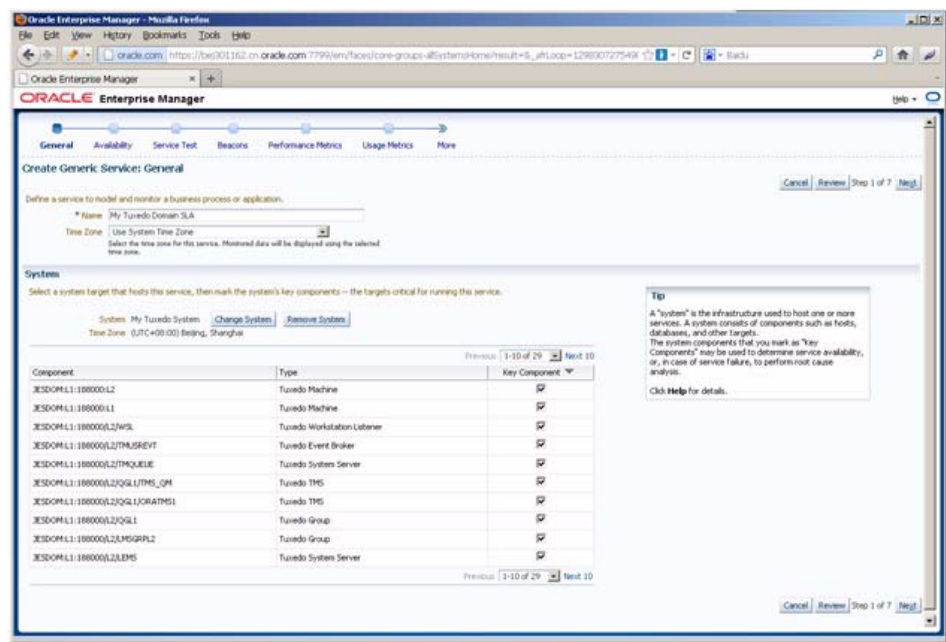
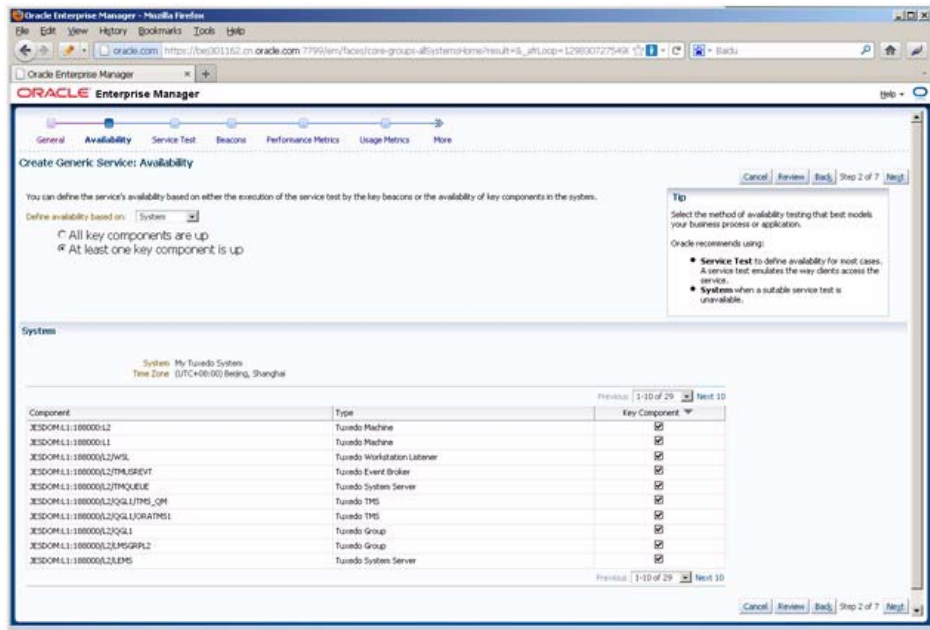


Figure 9 Creating Enterprise Manager services SLA (Continued)



## 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 Oracle Tuxedo plug-in feature that coordinates and allocates resources elastically based on predefined Oracle Tuxedo application policies. It includes the following major functions:

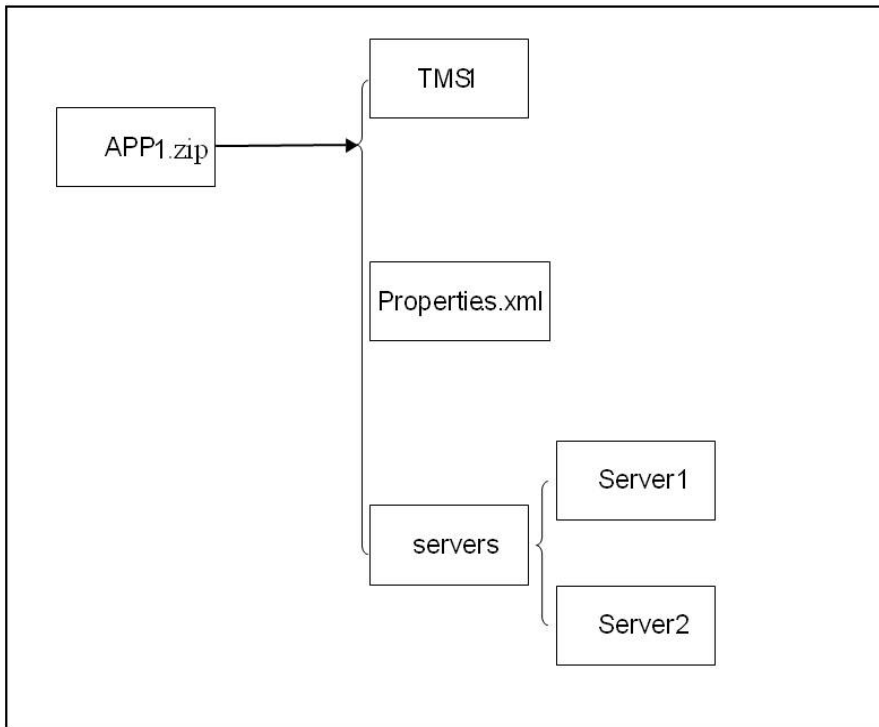
- Manages Oracle Tuxedo application functions by organizing them to application packages.
- Creates a new Tuxedo domain based on application packages using Domain Editor.
- Dynamically applies application package content to a running Oracle Tuxedo domain. The deployment can be performed manually or automatically 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 10](#).

**Figure 10 Application Package Structure**

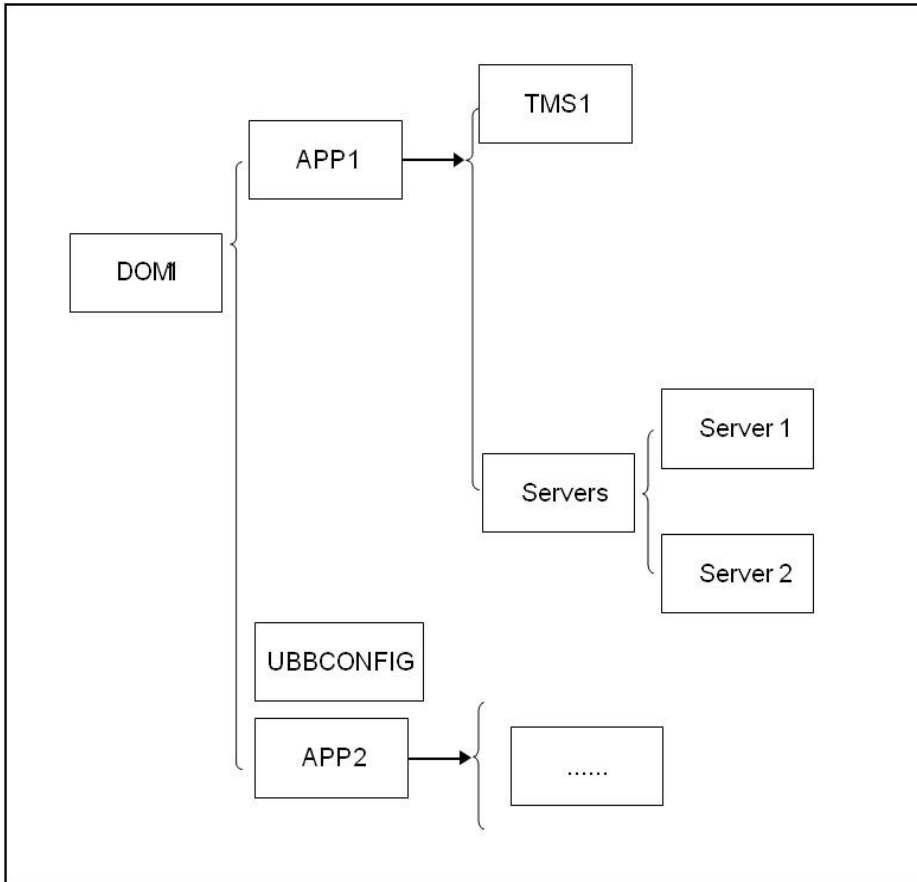
- The application package name must be unique within all the uploaded application packages.
- The application package name must consist of 64 characters or less, without containing asterisk (\*), comma, or colon.

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 11](#) 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 11 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](#).

**Listing 2 Adding Pathname to Server Names**

---

```

<ApplicationProperties>
    <PackageName>APP1.zip</PackageName>
    <TuxedoVersion>12.1.1.0 </TuxedoVersion>
    <SupportedOS>Linux</SupportedOS>

    <TuxedoWordSize>64</TuxedoWordSize>
    <MachineArch>x86_64</MachineArch>
    <LibPath> 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 2](#).

**Table 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 3](#) lists the category of each parameter in the \*GROUPS, \*RMS, \*SERVERS, and \*SERVICES sections.

**Table 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_LOCATION = string_value	Forbidden
	SEC_PRINCIPAL_PASSWORD = string_value	Forbidden

**Table 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 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 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 3 UBBCONFIG Parameter Categories**

Section	Parameter	Category
SERVICES	SRVGRP = string_value	Forbidden
	SVCNM	Mandatory
	BUFTYPE = "type1[:subtype1[, subtype2 . . . ]][;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 Guide](#).

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/lcfileirc/vol1/APPDIR`, you cannot specify `ENVFILE=/nfs/lcfileirc/vol2/envfile` in `UBBCONFIG` because `ENVFILE` is under `/nfs/lcfileirc/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. 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 4](#) shows the Controls panel control buttons.

**Table 4 Control Buttons**

Control Button	Description
Save	<p>Saves the current domain configuration. You can load the saved domain by clicking <b>Edit</b> 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 <a href="#">Policy Management</a> or click <b>Help</b> 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 <a href="#">Security Management</a> or click <b>Help</b> on the upper right of the page.</p>
DMConfig	<p>Lets you configure each section parameters of DMCONFIG. For more information, see <a href="#">Creating Domain Config</a> or click <b>Help</b> 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 <a href="#">JES Configuration</a> or click <b>Help</b> on the upper right of the page.</p>
CICSConfig	<p>Lets you configure the Tuxedo system servers that are specific to ART CICS in the Tuxedo domain. For more information, see <a href="#">Using Resource Broker for ART CICS</a>.</p>

**Table 4 Control Buttons**

Control Button	Description
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 5](#).

**Table 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 6](#).

**Table 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 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 7](#).

**Table 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 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 8](#).

**Table 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 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 9](#).

**Table 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 10](#).

**Table 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 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 11](#).

**Table 11 \*SERVICES Section Parameters**

Item	Value
SRVGRP	<p>If Properties.xml contains this parameter, the system replaces it with the corresponding group generated before.</p> <p>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.</p>
ROUTING	<p>If Properties.xml contains this parameter, system replaces it with the corresponding routing name generate by the system.</p>

For more information, see [Table 3](#). You can add other parameters to the UBBCONFIGfile as well.

### \*ROUTING Section

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

For otherProperties.xml file \*ROUTING section parameters, the system keeps their values and

**Table 12 \*ROUTING Section Parameters**

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

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](#)

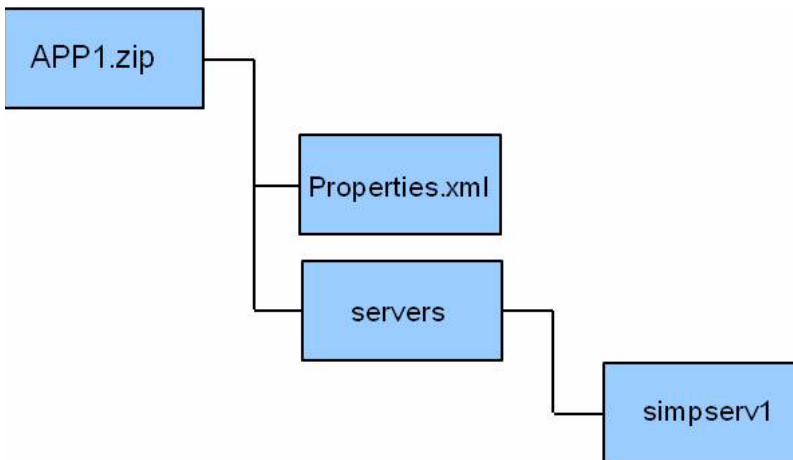
- [Creating a Tuxedo Domain](#)
- [Deploying Application Packages to Domain](#)

## 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 12](#):

**Figure 12 APP1.zip Content Structure**



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

**Listing 3 APP1.zip Properties.xml**

---

```

<?xml version="1.0" encoding="ISO-8859-1" ?>
<ApplicationProperties>
    <PackageName>APP1.zip</PackageName>
    <TuxedoVersion>12.1.1.0</TuxedoVersion>
    <SupportedOS>Linux</SupportedOS>
    <TuxedoWordSize>64</TuxedoWordSize>
  
```

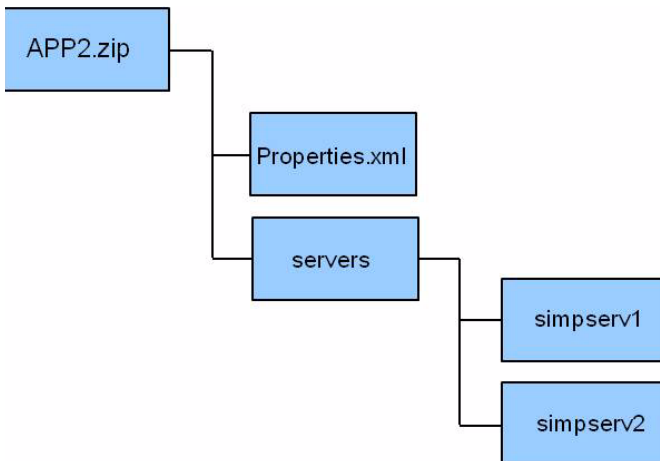
```
<MachineArch>x86_64</MachineArch>

<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 13](#):

**Figure 13 APP2.zip Content Structure**



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

**Listing 4 APP2.zip Properties.xml**

---

```

<?xml version="1.0" encoding="ISO-8859-1" ?>
<ApplicationProperties>
    <PackageName>APP2.zip</PackageName>
    <TuxedoVersion>12.1.1.0</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 **With Oracle Tuxedo Authentication**.

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.

## Creating a Tuxedo 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 `UBBCONFIGfile *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 `UBBCONFIGfile` entry is added to the `*MACHINES` section on the left side.
- Several `UBBCONFIGfile` 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.

## 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 14](#)

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

**Figure 14** New Added Entries



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 new Tuxedo domain is ready to be deployed and monitored.

## Deploying the Domain to Remote Machine

After completing all necessary configuration for the newly created Tuxedo domain, do the following steps to deploy the new 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 15 shows the domain deployment options screen.

**Figure 15 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>

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 `simpserve2` and `simpserve3` 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.

Once the deployment is complete, the new directory, `APP2`, can be found in domain `APPDIR` and the new targets corresponding to `GROUP2` and `simpserve2/simpserve3` 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 16](#):

**Figure 16 Define a CPU Metric Event- Example**

**Select Specific Metric Alert**

To select metrics and optional key values, select a target type first and click "Search".

**Search**

\* Target Type: Host Metric Name: CPU Utilization Search

Select	Metric Group	Metric	Objects	Excluded Objects
<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	+

**Severity and Corrective Action Status**

Severity: Warning

Corrective action status: ☐ Success for critical metric alert

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.

### 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 `$APPPDIR/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 \$APPPDIR 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 \$APPPDIR directory.

- **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 5 shows a UBBCONFIG file example.

### Listing 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>

    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 "
```

## Using the Enterprise Manager for Oracle Tuxedo

```
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 `ksh` 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.

## 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:$PATH
QMCONFIG=$APPPDIR/QUE
DATA_SOURCE=$APPPDIR/data_source
DATA=$APPPDIR/data
TMP=$APPPDIR/tmp
MT_TMP=$APPPDIR/tmp
MT_ACC_FILEPATH=$APPPDIR/acc
MT_LOG=$APPPDIR/Logs/log
SPOOL=$APPPDIR/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 batch, you must configure several Tuxedo provided system servers in the UBBCONFIG file to enable resource broker for ART CICS Runtime.

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 in `Properties.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 12cR1 release, the suggested value is 12.1.1.0.

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 transclasses.

- 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.

## 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.

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
$APPPDIR/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
```

**Note:** If any environment variable not listed above is used, you must specify it manually in the `confide` script or `boot` script.

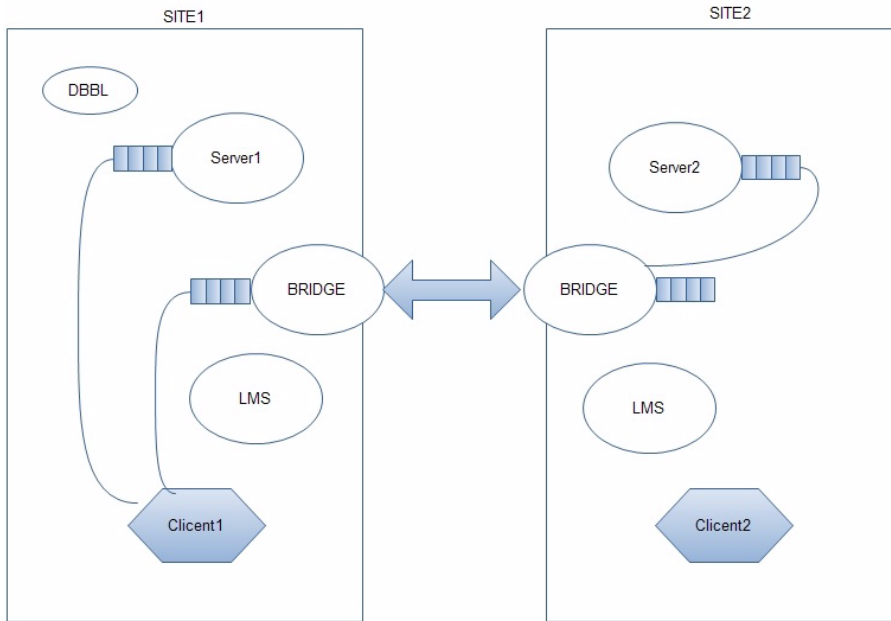
## Dynamic Request Broker

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 17](#), `server2` serves both `client1` and `client2` that are located on different machines.

**Figure 17 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/tuxedo12.1.1.0  
-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
```

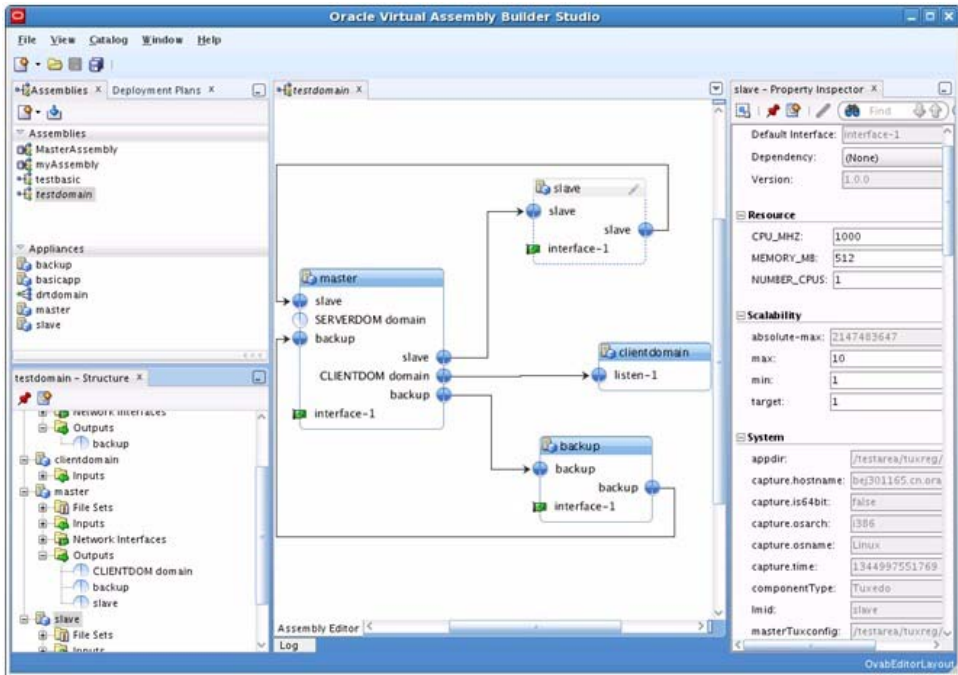
```
./abctl introspectTuxedo -TUXDIR /testarea/tuxreg/tuxedo/tuxedo12.1.1.0  
-TUXCONFIG /testarea/tuxreg/workspace/OVAB/tux/work/domain/tuxconfig  
-remoteHost bej301165.cn.oracle.com -remoteUser tuxreg -name slave  
-force
```

```
./abctl introspectTuxedo -TUXDIR /testarea/tuxreg/tuxedo/tuxedo12.1.1.0  
-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
```

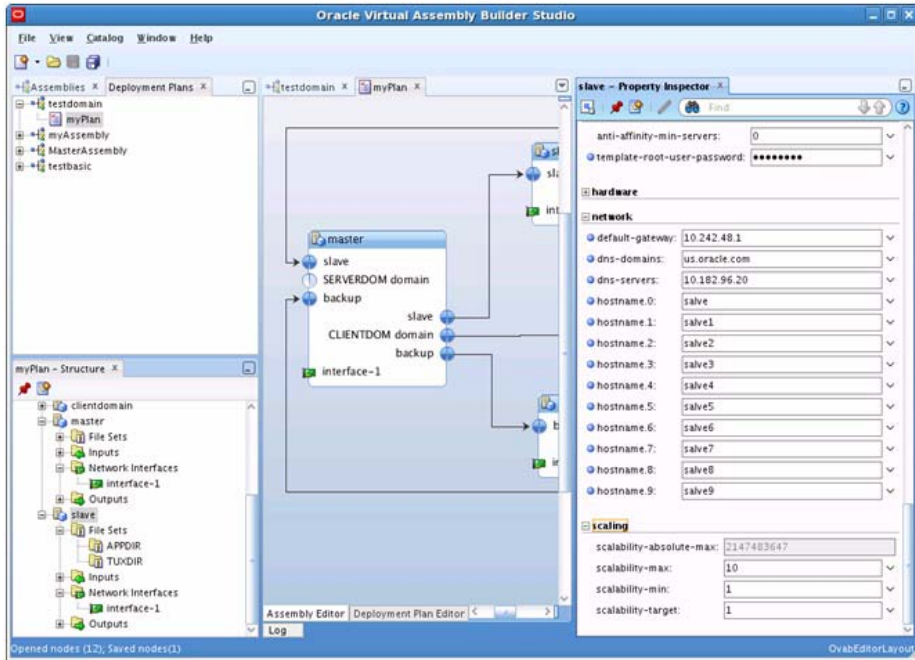
4. From OVAB GUI, link resources as shown in [Figure 18](#)

Figure 18 Link Resources



- From slave - Property Inspector, set the max, min, and target value scaling field as 10, 1, and 1, respectively as shown in [Figure 19](#)

Figure 19 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 20](#).

**Figure 20 Add OS Command**

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 21](#).

**Figure 21 Specify Target Group**

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 22 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

