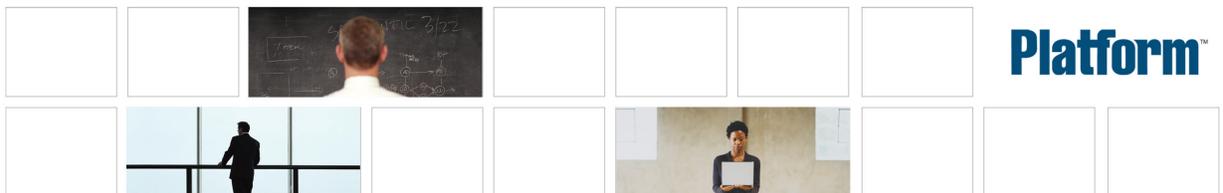


---

# EGO Tutorials

Platform EGO  
Version 1.2.3  
January 2008



Copyright

© 1994-2007 Platform Computing Corporation

All rights reserved.

Although the information in this document has been carefully reviewed, Platform Computing Corporation ("Platform") does not warrant it to be free of errors or omissions. Platform reserves the right to make corrections, updates, revisions or changes to the information in this document.

UNLESS OTHERWISE EXPRESSLY STATED BY PLATFORM, THE PROGRAM DESCRIBED IN THIS DOCUMENT IS PROVIDED "AS IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT WILL PLATFORM COMPUTING BE LIABLE TO ANYONE FOR SPECIAL, COLLATERAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, INCLUDING WITHOUT LIMITATION ANY LOST PROFITS, DATA, OR SAVINGS, ARISING OUT OF THE USE OF OR INABILITY TO USE THIS PROGRAM.

We'd like to hear from you

You can help us make this document better by telling us what you think of the content, organization, and usefulness of the information. If you find an error, or just want to make a suggestion for improving this document, please address your comments to [doc@platform.com](mailto:doc@platform.com).

Your comments should pertain only to Platform documentation. For product support, contact [support@platform.com](mailto:support@platform.com).

Document redistribution and translation

This document is protected by copyright and you may not redistribute or translate it into another language, in part or in whole.

Internal redistribution

You may only redistribute this document internally within your organization (for example, on an intranet) provided that you continue to check the Platform Web site for updates and update your version of the documentation. You may not make it available to your organization over the Internet.

Trademarks

®LSF is a registered trademark of Platform Computing Corporation in the United States and in other jurisdictions.

™ACCELERATING INTELLIGENCE, PLATFORM COMPUTING, PLATFORM SYMPHONY, PLATFORM JOBSCHEDULER, PLATFORM ENTERPRISE GRID ORCHESTRATOR, PLATFORM EGO, and the PLATFORM and PLATFORM LSF logos are trademarks of Platform Computing Corporation in the United States and in other jurisdictions.

®UNIX is a registered trademark of The Open Group in the United States and in other jurisdictions.

Linux® is the registered trademark of Linus Torvalds in the U.S. and other countries.

Microsoft is either a registered trademark or a trademark of Microsoft Corporation in the United States and/or other countries.

®Windows is a registered trademark of Microsoft Corporation in the United States and other countries.

Intel®, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Other products or services mentioned in this document are identified by the trademarks or service marks of their respective owners.

Third-party license agreements

<http://www.platform.com/Company/third.part.license.htm>

Third-party copyright notices

<http://www.platform.com/Company/Third.Party.Copyright.htm>

---

# Contents

1	EGO Tutorials .....	5
	Tutorial: Create a consumer tree .....	6
	Tutorial: Set up resource groups .....	14
	Tutorial: Modify the out-of-box resource plan .....	20



C H A P T E R

---

# 1

## EGO Tutorials

# Tutorial: Create a consumer tree

## Goal

The consumer tree organizes consumers into a structure that makes it easy to later apply resource plans. The tree defines organizational relationships among consumers and should mirror your business structure. In this tutorial we plan out a basic business structure and then create consumers to populate the consumer tree.

## At a glance

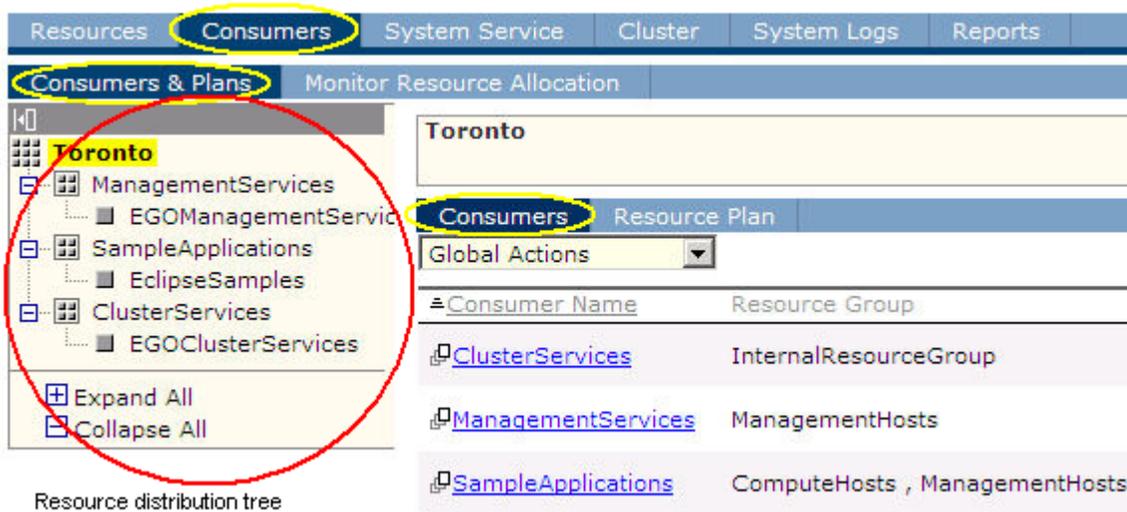
Contents include the following:

1. Recognize the default configurations
2. Learn about the consumer tree and its components
3. Plan your business structure
4. Create consumers

## 1. Recognize the default configurations

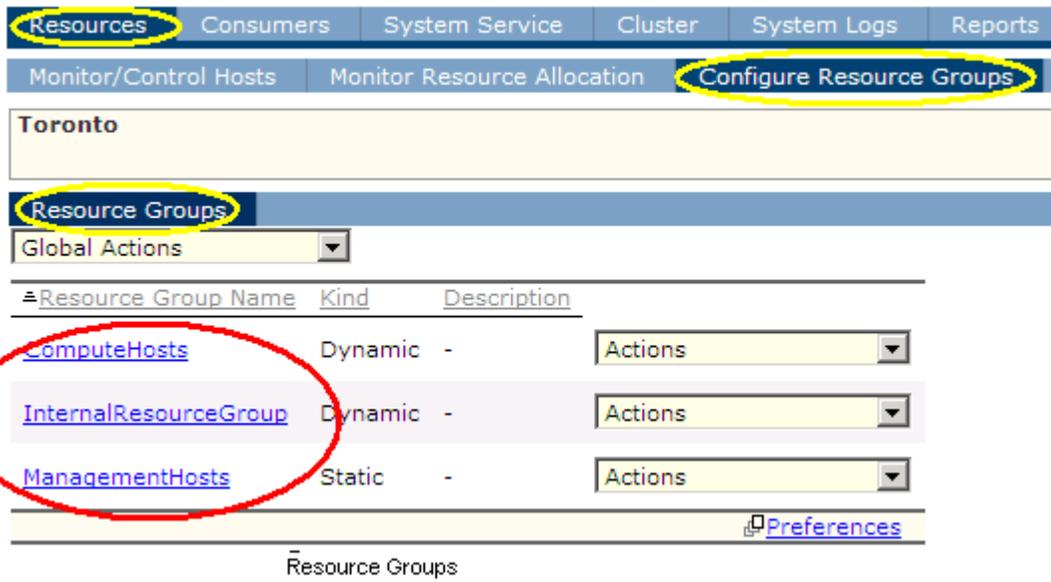
To help orient you, here is a list of the default resource components you see and work with in the Platform Management Console:

- Consumer tree: “ManagementServices” (with nested child consumer “EGOManagementServices”), “SampleApplications” (with nested child consumer “EclipseSamples”), and ClusterServices (with nested child consumer “EGOClusterServices”).



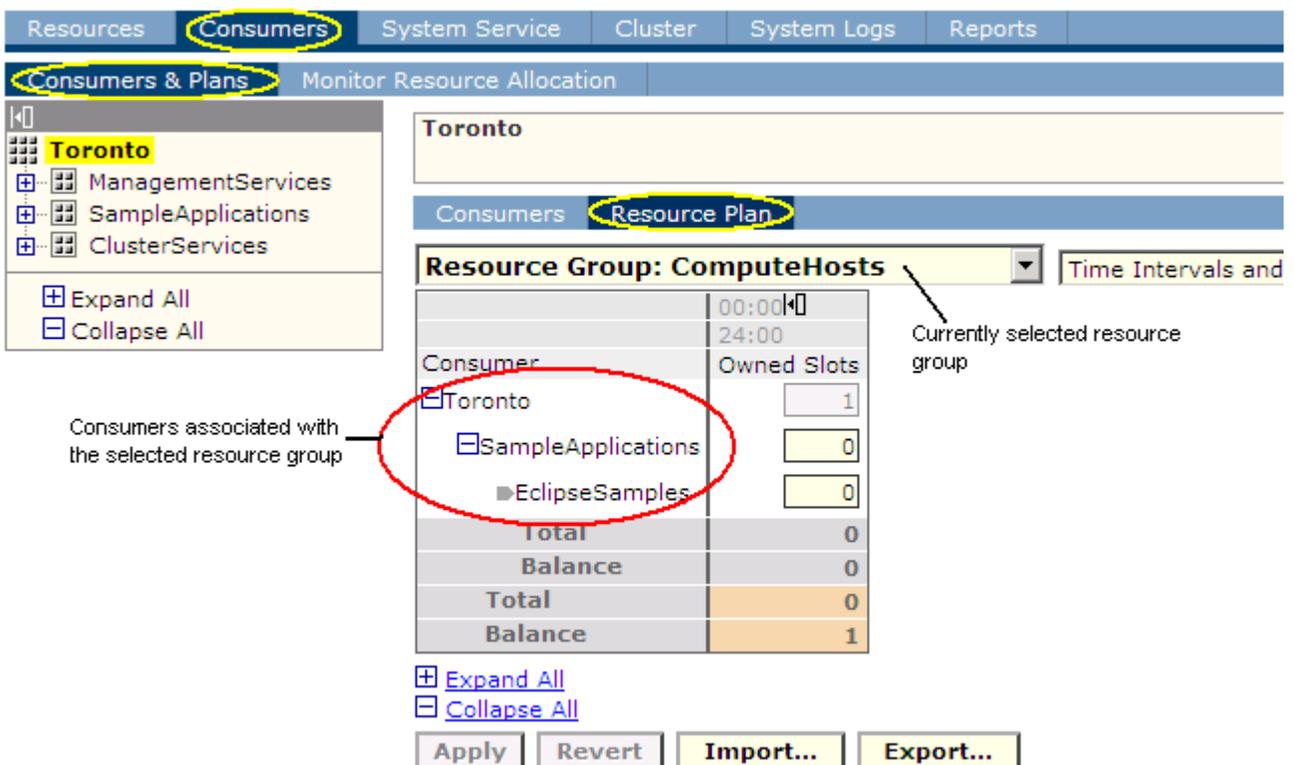
In this tutorial, we create a new branch with a new top-level consumer and add descendants (parent and child consumers) to the tree. Never remove the ManagementServices or ClusterServices branches, nor change their consumer names, when building or modifying your tree.

- Resource groups: “ComputeHosts” (executes workload units), “InternalResourceGroup” and “ManagementHosts” (run important EGO components and services).



In this tutorial, we work with the “ComputeHosts” resource group.

- Resource plan (default resource group upon opening page is “ComputeHosts”): Only consumers registered to a selected resource group show; select different resource groups to modify corresponding resource plans.



In this tutorial, we add a new parent consumer and a child consumer under the SampleApplications branch, and register them to an existing resource group.

## 2. Learn about the consumer tree and its components

The consumer tree identifies consumers of cluster resources, and organizes them into a manageable structure.

Component	Description
Tree	The resource distribution tree identifies consumers of the cluster resources, and organizes them into a manageable structure.
Plan (Resource plan)	The resource plan describes the relationship between the consumer tree and resource groups, and defines plans for how cluster resources are to be shared among consumers.
Consumers	<p>A consumer in the tree represents any entity that can demand resources from the cluster. A consumer might be a business service, a business process that is a complex collection of business services, an individual user, or an entire line of business.</p> <p>The consumers ManagementServices, SampleApplications, and ClusterServices, along with their sub-consumers, are installed by default.</p> <ul style="list-style-type: none"> <li>ManagementServices has one sub-consumer, EGOManagementServices, which runs important system services on management hosts in the cluster. Services include derbydb, plc, purger, RS, ServiceDirector, WEBGUI, and WebServiceGateway. ManagementServices is configured to use the ManagementHosts resource group. Do not modify or delete this consumer.</li> <li>SampleApplications has one sub-consumer, EclipseSamples.                     <p>The SampleApplications consumer and its sub-consumer can be modified or deleted (although you want to use the provided EclipseSamples to begin using EGO right away—this is a sample consumer with registered applications that are ready to run).</p> </li> <li>ClusterServices is configured to use the InternalResourceGroup resource group. It has one sub-consumer, EGOCustomerServices, which runs an essential RFA service (remote file access) on every host in the cluster. Do not modify or delete ClusterServices, or use it to run workload units.</li> </ul>
Multi-level tree	<p>Company projects are generally structured with multiple layers and components. For example, a project belongs to a department, a department to a business unit and so on. A multi-level consumer tree allows you to configure consumers in a hierarchical fashion to match your business structure.</p> <hr/> <p><b>Note:</b></p> <p>As a best practice, restrict the number of tree levels to four.</p>
Tree root	The root of the tree represents the entire cluster and all resources in it. Resources from the root are distributed through the tree to consumers.
Top-level consumers	The consumers attached directly to the root are called top-level consumers. The top-level consumer is the head of a consumer branch.
Leaf consumers	<p>If a consumer has no descendants, it is called a leaf consumer. Services and applications can only be associated with leaf consumers.</p> <p>Borrow and lend policies are set at this level.</p> <hr/> <p><b>Note:</b></p> <p>As a best practice, limit leaf consumers to fewer than 20.</p>

Component	Description
Branches, descendants	<p>If a consumer in the tree has other descendants, thereby creating a branch in a multi-level tree, it is called a branch consumer. Branch consumers exist to redistribute resources down the branch to their descendants.</p> <p>Descendants of a branch consumer may also have descendants, thereby becoming branch consumers themselves. Every branch in the tree ends with a leaf consumer.</p>
Parent	A consumer containing another consumer (a "child"). A parent can contain a child consumer, or be the child of another parent consumer.
Sibling	Two or more consumers sharing the same parent consumer.
Child, sub-consumer	A consumer nested within another consumer (a "parent"). A child (or sub-consumer) of one parent can be the parent to another nested child. A leaf consumer is always a child at the end of the branch.
Consumer administrators	For ease of management, you can create consumer administrators for top-level consumers in a multi-level tree. These users can change the plan for lower-level consumers on their branch (descendants), without requiring cluster administrator permissions. Only a cluster administrator can change the plan for top-level consumers.
ClusterServices consumer	The special ClusterServices consumer is a built-in top-level consumer that uses the built-in InternalResourceGroup resource group and runs necessary EGO components. It contains one descendant, EGOClusterServices, and is allocated a certain number of slots by default. Do not modify this consumer or use it to run workload units.
ManagementServices consumer	The special ManagementServices consumer is a built-in top-level consumer that uses the built-in ManagementHosts resource group and runs necessary EGO components. It contains one descendant, EGOManagementServices. Do not modify this consumer or use it to run workload units.
SampleApplications consumer	This is an out-of-box sample consumer with registered applications that are ready to run (if you choose to install and run the SDK samples). It contains one descendant, EclipseSamples.

### 3. Plan your business structure

The choice of consumers and the consumer hierarchy should reflect long-term business goals because it can be complicated to modify the tree. Before adding consumers to your tree, you need to map out the business structure you wish to reflect in your consumer tree.

For example, top-level business processes (or perhaps departments) become top-level consumers, while the lowest area of business becomes a leaf consumer. It is at the leaf level where you register such things as services or other application managers.

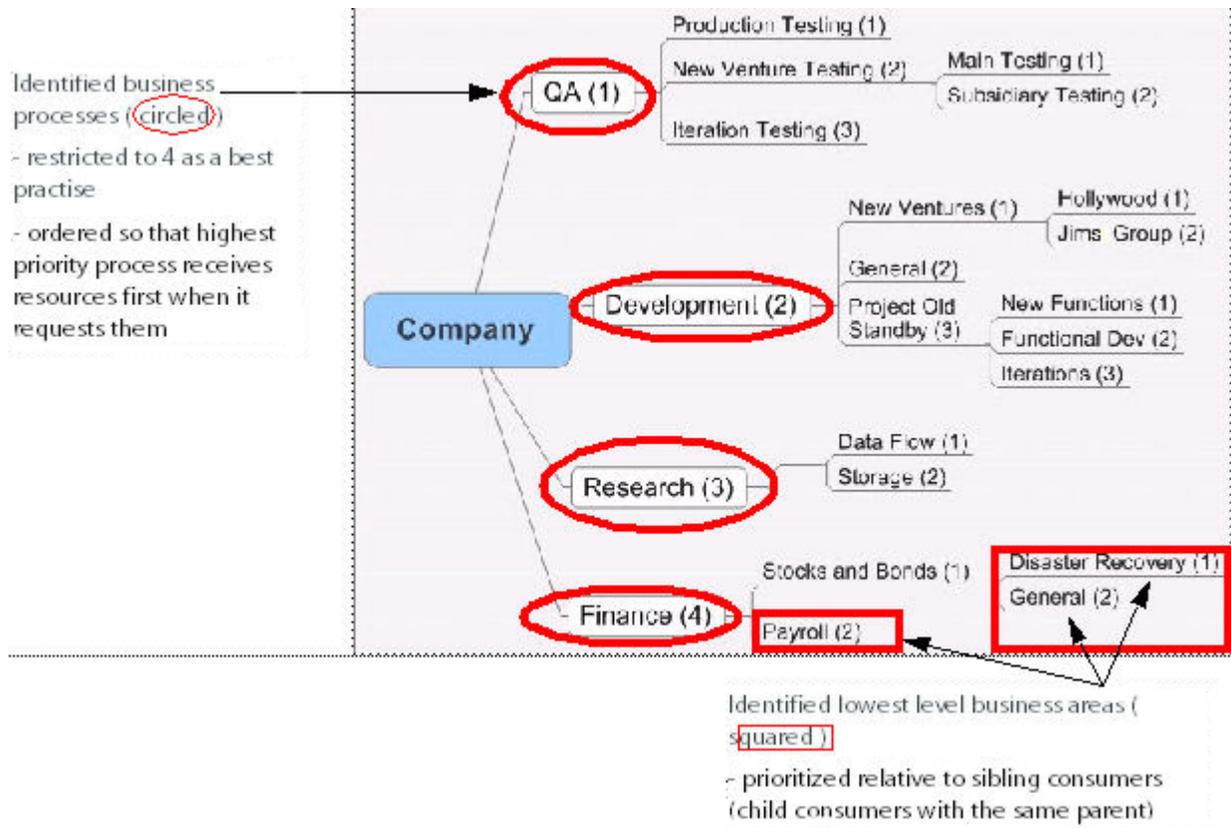
Out-of-the-box you have some built-in consumers that run necessary cluster services (ManagementServices and ClusterServices). You also have a SampleApplications consumer, used by the Sample client applications included in the EGO SDK.

General steps and considerations in mapping out a business structure include the following:

1. Map out your business structure by hand, restricting the number of levels to four.
2. Decide if any branches (top-level consumers) in your tree do not need to consume cluster resources, and then remove them from your diagram.
3. Prioritize your business processes, expecting that the highest priority process receive resources first when needed.

4. Ensure that your business structure makes a distinction that parallels how you want to manage and distribute your resources. You may want to break out special projects that need dedicated or specialized hosts.
5. Prioritize all your lower level business areas (leaf consumers) relative to other leaf consumers on the same branch.

Once mapped, a completed business structure might look like this:



## 4. Create consumers

You must be logged in as a cluster administrator and you should have already added most of your hosts to the cluster.

A consumer represents an entity that can demand resources from the cluster. A consumer might be a business service, a business process that is a complex collection of business services, an individual user, or an entire line of business.

1. Click Consumers > Consumers & Plans > Consumers.

The out-of-box consumers in your tree display (ManagementServices, SampleApplications, and ClusterServices).

2. If not already highlighted, select the tree root in the consumer tree.
3. From the Global Actions list, select Create a Consumer.
4. Fill in the consumer properties.

1. Specify a name for your new top-level consumer.
2. Choose one or more administrators for this consumer.

Specified administrators automatically become administrators for any other consumers created on this branch. You cannot add a cluster administrator user account to the consumer administrator list, as this is already configured by default.

3. Specify zero, one, or more users for this consumer.

4. Specify the workload execution user account (the OS account under which workload units run).

Windows accounts should include a domain name.

If you specify a Windows user account that has not already been configured, you have to log on to EGO as the cluster administrator and then run `egosh ego execpasswd` before the execution user can run an activity without exiting.

5. Specify one or more resource groups this consumer should have access to.

Only the resource groups specified by this consumer's parent are available for selection. If you have not modified your resource groups, simply keep the default resource groups that are already checked.

6. Leave the Reclaim behavior section blank.

Reclaim behavior is an advanced feature and must be coordinated with the resource distribution plan settings.

7. Check the box Rebalance when time intervals change.

This ensures that when your resource plan changes according to set time intervals, that originally configured share ratios, allocations, and lend/borrow policies are reapplied and enforced across all consumer branches in the consumer tree.

### Create a Consumer

**1) Identify your consumer (\*)**  
Name:

**2) Specify administrators for this consumer**  
Available User Accounts:   
Administrators for this Consumer:

**3) Specify users for this consumer**  
Available User Accounts:   
Users for this Consumer:

**4) Specify an OS user account**  
~~Domain\~~User Name: (Domain for Windows accounts only)

**5) Specify resource groups (\*)**  
 YourNewResourceGroup  
 InternalResourceGroup  
 ComputeHosts  
 ManagementHosts

**6) Reclaim behavior (when returning unowned resources)**  
Reclaim grace period:    
 Rebalance when time intervals change

\* Required fields

*Annotations:*  
Name of top level consumer; reflects a part of your planned business structure  
Add consumer administrators  
Add consumer users  
Specify the OS account under which workload runs  
Specify resource groups this consumer can access  
Check to rebalance resource distribution according to plan

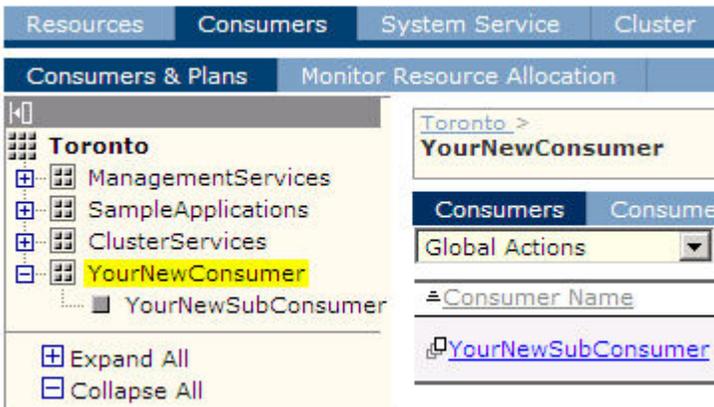
5. Click Create.

6. From the consumer tree, select the new consumer.



7. From the Global Actions list, select Create a Consumer.
8. Fill in the consumer properties for a new leaf consumer (child to the new top-level parent you just created), repeating step 4.

You have now created a new branch in your tree (a new top-level consumer), along with a sub-consumer (new leaf consumer).



If you have services or a Platform clustered application manager (such as Platform Symphony), you can now register them to the leaf consumer.

# Tutorial: Set up resource groups

## Goal

EGO comes with three default resource groups that have core services and sample applications registered to them. Out-of-the-box, there is also a basic consumer tree and resource plan. In this tutorial, you add a new resource group alongside the existing ones, but use the provided consumer tree and resource plan.

## At a glance

Contents include the following:

1. Recognize the default configurations
2. Learn about resource groups
3. Create a new resource group by host names
4. Assign the new resource group to a consumer
5. Update the resource plan with new resource group

## 1. Recognize the default configurations

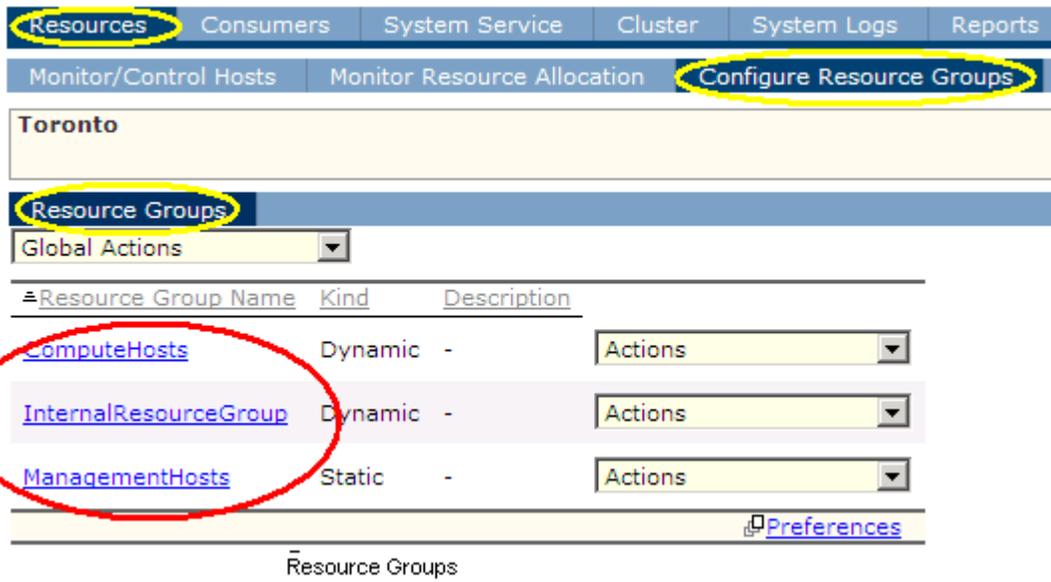
To help orient you, here is a list of the default consumer tree, resource groups, and resource plan components you see and work with in the Platform Management Console:

- Consumer tree: “ManagementServices” (with nested child consumer “EGOManagementServices”), “SampleApplications” (with nested child consumer “EclipseSamples”), and ClusterServices (with nested child consumer “EGOClusterServices”).



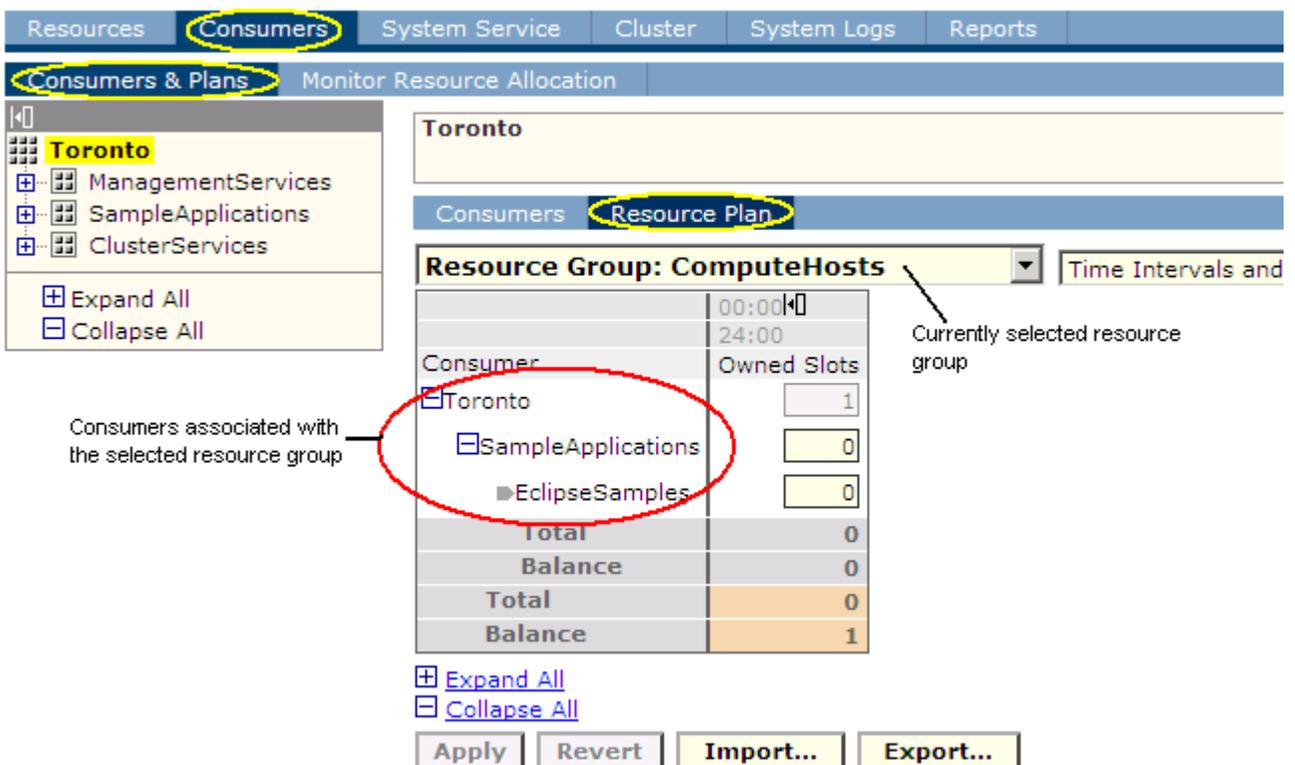
In this tutorial, we work with the “SampleApplications” top-level consumer and its “EclipseSamples” sub-consumer.

- Resource groups: “ComputeHosts” (executes workload units), “InternalResourceGroup” and “ManagementHosts” (run important EGO components and services).



In this tutorial, we work with the “ComputeHosts” resource group, and create a new resource group.

- Resource plan (default resource group upon opening page is “ComputeHosts”): Only consumers registered to a selected resource group show; select different resource groups to modify corresponding resource plans.



In this tutorial, we update the resource plan to include the new resource group you create.

## 2. Learn about resource groups

Resource groups generally fall into one of three categories:

- Resource groups that include compute hosts with certain identifiable attributes a consumer may require in a requested resource (for example, resources with large amounts of memory; considered “dynamic”—new hosts added to the cluster that meet the requirements are automatically added to the resource group)
- Resource groups that only include certain compute hosts (for example, so that specified resources are accessed by approved consumers; considered “static”—any new hosts added to the cluster have to be manually added to the resource group)
- Resource groups that encompass management hosts only (reserved for running services, not a distributed workload; for example, the out-of-the-box “ManagementHosts” group)

## 3. Create a new resource group by host names

You must be logged in as a cluster administrator and you should have already added most of your hosts to the cluster.

Create a new resource group from the Platform Management Console to ensure your consumers have the appropriate group of compute hosts available to them. Resource groups are often the easiest way to create a group of same-type hosts for a consumer. Resource groups are either defined by resource requirements (dynamic) or by host names (static).

1. In the Platform Management Console, click Resources > Configure Resource Groups > Resource Groups.

Out-of-the-box, existing resource groups include “ComputeHosts”, “InternalResourceGroup”, and “ManagementHosts”. DO NOT delete the “InternalResourceGroup” or “ManagementHosts” groups nor modify the existing EGO services registered to them.

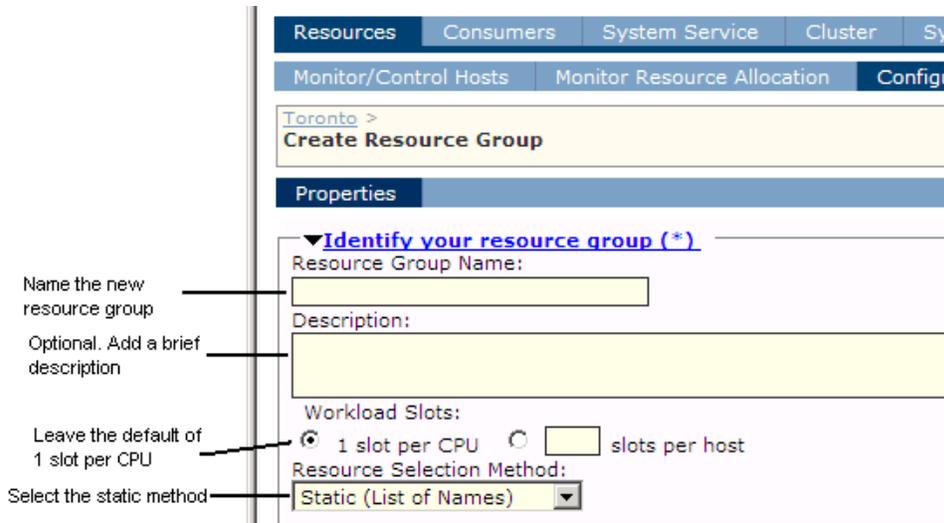
2. From the Global Actions drop-down list, select Create a Resource Group.
3. Identify the new resource group in the top section of the Properties page:

1. Specify a resource group name.

Resource group names must consist of letters and numbers only (no spaces or special characters) and must be 64 characters or less.

2. Include a description (max. 200-characters) of the resource group.
3. Leave the default setting of 1 slot per CPU for Workload Slots (this defines how many slots per host you would like to have the system count; unless you are an advanced user, do not change this setting).
4. For Resource Selection Method, select Static (List of Names).

Static resource selection means that you are manually selecting specific hosts to belong to this resource group. Any new compute hosts that are later added to the cluster, and that you want to add to this resource group, must be manually added.



4. Under Hosts to Show in List, select All hosts.

A list of all hosts that belong to your cluster displays.

5. Review the hosts found in your cluster:

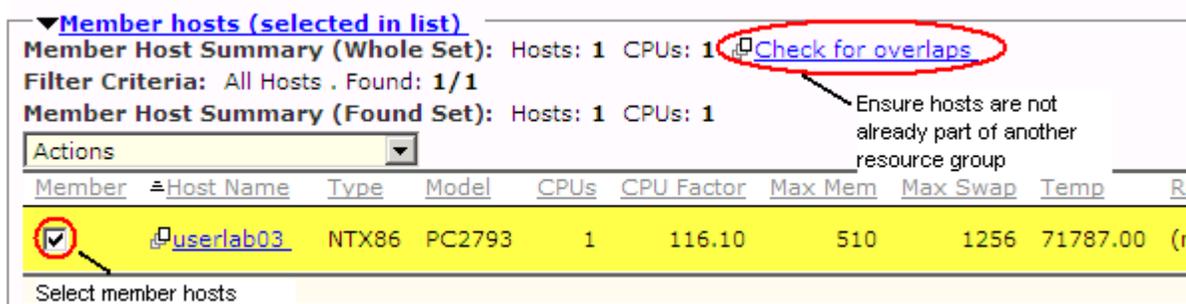
1. Click Member hosts to expand the section and review the hosts found in your cluster.
2. Review your member hosts and select the hosts you want using the check boxes.

If you select no member hosts, all hosts in your cluster are added to this resource group when you create it.

3. Click Check for overlaps.

If any hosts overlap, remove them from this resource group or remove them from the overlapping resource group.

No hosts should overlap between resource groups. Resource groups are used to plan resource distribution in your resource plan. Having overlaps causes the hosts to be double-counted (or more) in the resource plan, resulting in recurring under-allocation of some consumers. The exception is with hosts listed in InternalResourceGroup—although all hosts in the cluster are listed here they are not “double-counted” in the resource plan.



6. Click Create.

## 4. Assign the new resource group to a consumer

You need to assign new resource groups to consumers. Consumers must already exist and be a part of your consumer tree.

1. Click Consumers > Consumers & Plans > Consumers.

2. Select a consumer to assign the new resource group to.
  - If you have already created your consumer tree and modified the out-of-box structure, locate and click the consumer from tree to which you want to assign the new resource group.
  - If you have not modified the consumer tree, click SampleApplications from the consumer tree pane on the left to assign the new resource group to this consumer.
3. Click Consumer Properties.
4. Specify one or more resource groups that this consumer should have access to.

5. Click Apply.
 

The Consumer Properties page updates and your changes are saved.

## 5. Update the resource plan with new resource group

Any time you add or modify a resource group, you need to manage resource distribution for it using the resource distribution plan. If you have not yet created a new resource plan, or set up the provided out-of-box plan to meet your needs, do not complete this step.

1. Click Consumers> Consumers & Plans > Resource Plan.

- Use the drop-down menu to switch between resource groups, and then modify your resource plan details for each resource group.

**Note:**

Resources groups that do not yet have consumers assigned to them do not appear in the drop-down menu. Consumers must first be assigned from the Consumers & Plans page.

The screenshot shows the 'Monitor Resource Allocation' page for the 'Toronto' resource group. The 'Resource Plan' tab is selected. A drop-down menu is open, showing the following options:

- Resource Group: ComputeHosts
- Resource Group: InternalResourceGroup
- Resource Group: ManagementHosts
- Resource Group: YourNewResourceGroup

The 'YourNewResourceGroup' option is highlighted. Below the menu is a table with resource usage data:

SampleApplications	0
EclipseSamples	0
<b>Total</b>	<b>0</b>
<b>Balance</b>	<b>0</b>
<b>Total</b>	<b>0</b>
<b>Balance</b>	<b>1</b>

Below the table are links for 'Expand All' and 'Collapse All', and buttons for 'Apply', 'Revert', 'Import...', and 'Export...'.

Select your new resource group from the drop-down list

# Tutorial: Modify the out-of-box resource plan

## Goal

The resource distribution plan is closely related to the consumer tree. The plan cannot be defined without the tree. EGO comes with supplied resource groups and a basic out-of-the-box consumer tree to begin working with. In this tutorial you learn about resource plans by modifying the supplied version.

## At a glance

Contents include the following:

1. Recognize the default configurations
2. Learn about resource plans
3. Create a time interval
4. Set owned slots
5. Set lend policy
6. Set borrowing policy
7. Understand share ratios
8. Set share ratios
9. Understand consumer rank
10. Set consumer rank
11. Export or import resource plan

## 1. Recognize the default configurations

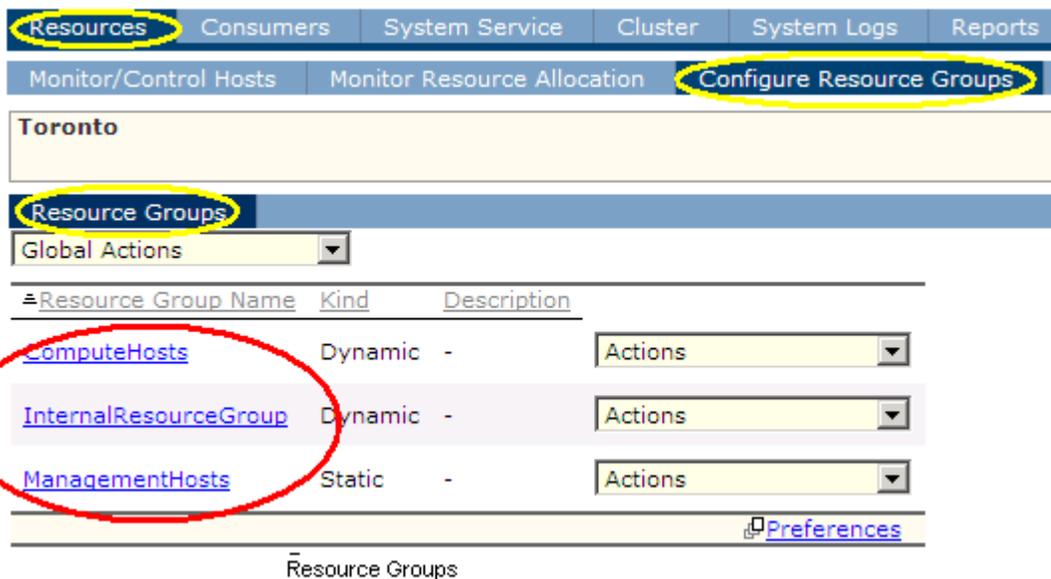
To help orient you, here is a list of the default consumer tree, resource groups, and resource plan components you see and work with in the Platform Management Console:

- Consumer tree: “ManagementServices” (with nested child consumer “EGOManagementServices”), “SampleApplications” (with nested child consumer “EclipseSamples”), and ClusterServices (with nested child consumer “EGOClusterServices”).



In this tutorial, we work with the “SampleApplications” top-level consumer and its “EclipseSamples” sub-consumer.

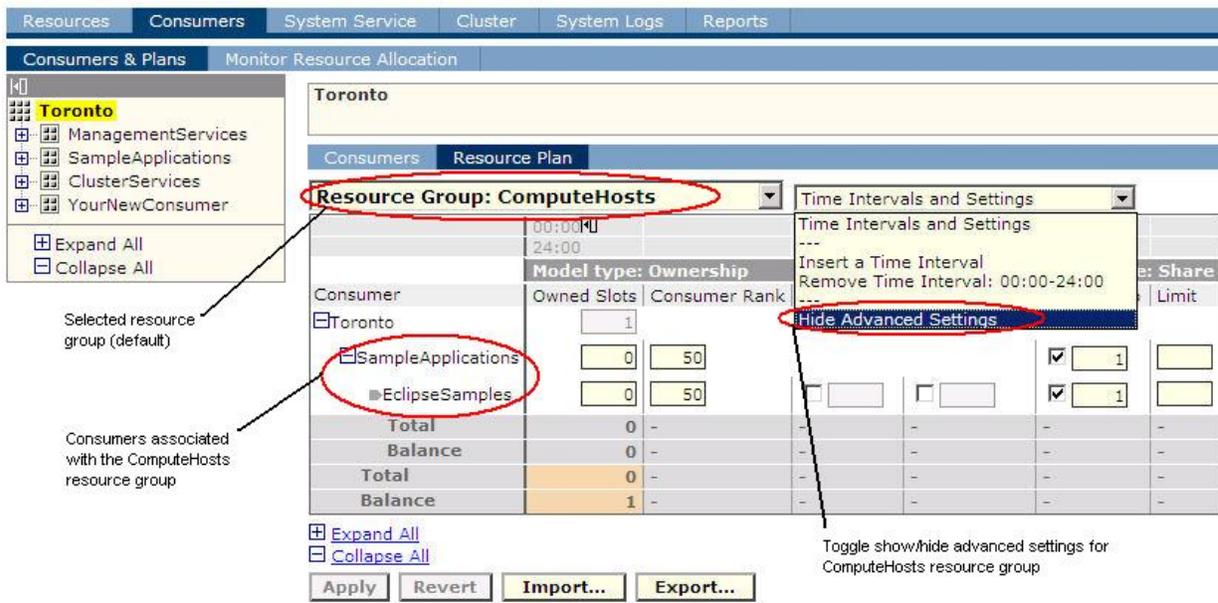
- Resource groups: “ComputeHosts” (executes workload units), “InternalResourceGroup” and “ManagementHosts” (run important EGO components and services).



In this tutorial, we work with the “ComputeHosts” resource group.

- Resource plan (default resource group upon opening page is “ComputeHosts”): Only consumers registered to a selected resource group show; select different resource groups to modify corresponding resource plans. Notice the following:
  - The resource group “ComputeHosts” is associated with one consumer (“SampleApplications”).

- There is only one defined time interval in the resource plan for the resource group view “ComputeHosts” (00:00 - 24:00).
- The consumer “SampleApplications” owns zero slots.
- Lending and borrowing are disabled for all consumers.
- Sharing is enabled for all consumers, with a default share ratio of one.
- Consumers associated with this resource group have a consumer rank set at 50 (where a higher number = a lower ranking)



In this tutorial, we change the default time intervals, slot ownerships, lending and borrowing policies, share ratio, and consumer rank.

## 2. Learn about resource plans

The resource plan defines how cluster resources are allocated among consumers. The plan takes into account the differences between consumers and their needs, resource properties, and various other policies concerning consumer rank and the allocation of resources.

A resource plan supports a number of rules used to set policies for how, when, and where resources get allocated. This includes policies concerning resource allocation and ownership, borrowing and lending, sharing, and the reclaiming of borrowed resources. In all cases, resource plans must be set at the leaf level (consumers with no descendents).

A resource plan allows for sharing, lending, and borrowing between specified consumers. A good resource plan anticipates time periods of peak activity and sets policies to ensure high priority consumers can access (borrow) required resources during these times.

EGO systematically allocates resources to consumers experiencing by demand by this ordered process:

1. Allocate idle resources owned by consumer
2. Allocate idle, owned resources from share pool
3. Allocate idle resources borrowed from other consumers
4. Allocate resources reclaimed by consumer
5. Allocate resources reclaimed by share pool

### 3. Create a time interval

Make sure you are at the top of your tree (at the cluster level).

Time-based configuration in the resource plan allows the resource distribution for a consumer to change according to the time of day.

Time intervals exist across all resource groups. No time gaps are allowed in the resource distribution plan.

Time interval boundaries are at the hour point, so the minimum time interval is 60 minutes. The daily pattern repeats itself every 24 hours.

1. Click Consumers > Consumers & Plans.

A list of consumers at that level in your tree displays.

2. Click Resource Plan.

The resource distribution plan displays.

3. Select the resource group you want to add the time interval for.
4. Select Time Intervals and Settings > Insert a Time Interval.

An insert element displays just above the resource distribution plan.

5. Specify a start time and an end time.

By default, settings for a new time interval are inherited from an existing one.

6. Click Insert.
7. Click OK to confirm that you want to insert the time interval.

The new time interval displays in your resource distribution plan for the selected resource group.

8. Add the time interval for any other resource groups.

### 4. Set owned slots

Before beginning, ensure you are still on the Consumers & Plans page, in the Resource Plan. The selected resource group should be "ComputeHosts".

When consumers "own" resources, they are guaranteed of a minimum allocation of resources, regardless of competition from other consumers. Ownership is expressed as a numeric quantity.

The number of owned slots available to each resource group and branch in the consumer tree is set elsewhere (see the tutorial for creating resource groups). For this tutorial, we'll use the default settings. Out-of-the-box, there is only 1 slot available for allocation within the "ComputeHosts" resource group per time interval (the default setting shows a balance of "1" at the bottom of the dialog box).

---

#### Tip:

Upon completion of your resource plan, each balance must show "0" except the cluster balance, which is allowed to retain unallocated resources (and therefore a balance greater than 0).

---

1. Click Expand All to expand the entire list of consumers associated with the ComputeHosts resource group.

- For each time interval, set the owned slots for each consumer and leaf consumer. Be sure to distribute all available resources as far down a consumer branch as possible to its leaf consumers. For example, assign 1 slot to SampleApplications, and then do the same for EclipseSamples.

The screenshot shows the 'Resource Plan' interface for the 'ComputeHosts' resource group. The table below shows the 'Owned Slots' for various consumers. Annotations explain the values:

- The 'Owned Slots' for the root consumer 'Toronto' is 1, indicating that one owned slot is available for allocation within this consumer branch.
- The 'Owned Slots' for the sub-consumer 'SampleApplications' is 1, and for its sub-consumer 'EclipseSamples' is also 1. This indicates that the available slot is assigned to the consumer SampleApplications and its sub-consumer (a leaf) EclipseSamples.
- The 'Balance' for the root consumer 'Toronto' is 0, indicating that all available slots from the ComputeHosts resource group are assigned for this branch and this time interval. Balance is now zero.

Consumer	Owned Slots	Consumer Rank	Lend	Limit	Borrow	Limit	Share Ratio	Limit
Toronto	1							
SampleApplications	1	50					<input checked="" type="checkbox"/> 1	
EclipseSamples	1	50	<input type="checkbox"/>		<input type="checkbox"/>		<input checked="" type="checkbox"/> 1	
<b>Total</b>	<b>1</b>	-	-	-	-	-	-	-
<b>Balance</b>	<b>0</b>	-	-	-	-	-	-	-
<b>Total</b>	<b>1</b>	-	-	-	-	-	-	-
<b>Balance</b>	<b>0</b>	-	-	-	-	-	-	-

**Tip:**

In EGO, the bottom sub-consumer is called a “leaf” consumer; a top-level consumer, such as SampleApplications, is considered a “branch” in the consumer “tree”.

When all available resources are allocated, the cluster balance at the bottom of the Owned Slots section changes to 0.

- Click Apply to set the new values and immediately apply the changes to the running cluster. If you simply want to save the changes you made to the plan, click Export and you can save the changes to a file that you can later import and apply to activate.

## 5. Set lend policy

Before beginning, ensure you are still on the Consumers & Plans page, on the Resource Plan page. The selected resource group should be “ComputeHosts”.

Lending is optional. You can enable lending only for leaf consumers who own resources (there are no lend setting available for non-leaf consumers in the resource plan). With lending, a consumer's unused resources become available

for other consumers to borrow. This kind of resource sharing improves the efficiency of the cluster. Without lending, owned resources cannot be shared with other consumers and idle resources are wasted.

1. Select Time Intervals and Settings > Show Advanced Settings to display advanced settings and policy configuration options for the resource group.
2. For each time interval, you can choose to Lend slots allocated to leaf consumers during periods of non-use.
  1. To lend a leaf consumer’s slots, check the box under Lend | Limit in the Model Type: Ownership section of the plan.
  2. Beside the check box, click Details. In the Lend Details dialog box that opens, specify the maximum number of slots you want to lend from this branch.

As the out-of-the-box configuration only has one slot available to the “ComputeHosts” resource group, you are not able to lend more than one slot at this point.

Resource Group: ComputeHosts		Time Intervals and Settings	
		00:00	24:00
Model type: Ownership			
Consumer	Owned Slots	Consumer Rank	Lend   Limit
<input type="checkbox"/> Toronto	1		
<input type="checkbox"/> SampleApplications	1	50	
<input checked="" type="checkbox"/> EclipseSamples	1	50	<input checked="" type="checkbox"/> 1 <a href="#">Details</a>
<b>Total</b>	1	-	-
<b>Balance</b>	0	-	-
<b>Total</b>	1	-	-
<b>Balance</b>	0	-	-

Expand All    Check to lend slots allocated to this leaf consumer to other consumers during periods of non-use.

Collapse All

Click Details to limit the number of slots this leaf consumer is willing to lend out.  
 Note:  
 Enter 0 to lend no slots.  
 Leave it blank to lend unlimited slots.

3. In future, when you have more than one consumer branch in the “ComputeHosts” resource group, you want to specify particular leaf consumers to lend to as part of a strategic lending/borrowing policy.
  1. Click Details beside the consumer’s Lend | Limit box.
  2. Check the boxes beside those leaf consumers from other branches you lend to.
 

(Out-of-the-box, there are no other leaf consumers in the “ComputeHosts” resource group, so there are none to check. Skip this step. If you want to look at a configuration with multiple leaf consumers, change the resource plan view to “Resource Group: ManagementHosts”, and then click Details.)
  3. Click Apply and then Close.

## 6. Set borrowing policy

Before beginning, ensure you are still on the Consumers & Plans page, in the Resource Plan. The selected resource group should be “ComputeHosts”.

Borrowing is optional. If borrowing is disabled, the allocation to a leaf consumer never exceeds the configured ownership. Therefore, if borrowing is disabled for all consumers, any unused resources (owned by other consumers) are wasted.

You can enable borrowing only for leaf consumers (consumers with no descendents); there are no borrow settings available for non-leaf consumers in the resource plan.

If you enable borrowing, distribution to a consumer can exceed its actual ownership during peak periods of activity and times of increased need. Enable borrowing to ensure unowned resources are not wasted/left unused.

1. If not already selected, click Time Intervals and Settings > Show Advanced Settings to display advanced settings and policy configuration options for the resource group.
2. In a similar fashion to setting the lending policy, you can choose to borrow slots allocated to other leaf consumers during periods of increased activity.
  1. To borrow slots, check the box under Borrow | Limit in the Model Type: Ownership section of the plan.
  2. Beside the check box, click Details. In the Borrow Details dialog box that opens, specify the maximum number of slots you want to lend from this branch.

As the out-of-the-box configuration does not have any other consumers from which to borrow in the "ComputeHosts" resource group, you can leave this field blank.

Consumers		Resource Plan			
Resource Group: ComputeHosts		Time Intervals and Settings			
	00:00				
	24:00				
Model type: Ownership					
Consumer	Owned Slots	Consumer Rank	Lend   Limit	Borrow   Limit	
<input type="checkbox"/> Toronto	<input type="text" value="1"/>				
<input type="checkbox"/> SampleApplications	<input type="text" value="1"/>	<input type="text" value="50"/>			
<input checked="" type="checkbox"/> EclipseSamples	<input type="text" value="1"/>	<input type="text" value="50"/>	<input checked="" type="checkbox"/> <input type="text" value="1"/> <a href="#">Details</a>	<input checked="" type="checkbox"/> <input type="text" value="1"/> <a href="#">Details</a>	
<b>Total</b>	<b>1</b>	-	-	-	-
<b>Balance</b>	<b>0</b>	-	-	-	-
<b>Total</b>	<b>1</b>	-	-	-	-
<b>Balance</b>	<b>0</b>	-	-	-	-

Check to enable borrowing slots from other leaf consumers during periods of increased activity

Click Details to enter the maximum number of slots to borrow from other lending consumers  
 Note:  
 Enter 0 to borrow no slots.  
 Leave the field blank to borrow unlimited slots.

3. In future, when you have multiple lending consumers in a resource group, you want to specify particular leaf consumers to borrow from as part of a strategic lending/borrowing policy.
  1. For each leaf consumer that has borrowing enabled, click Details beside the consumer's Borrow | Limit box.
  2. Check the boxes beside those leaf consumers from which you want to borrow from.

The Borrow Details dialog box opens for this consumer.

(Out-of-the-box, there are no other leaf consumers in the “ComputeHosts” resource group, so there are none to check. Skip this step. If you want to see a configuration with multiple leaf consumers, change the resource plan view to “Resource Group: ManagementHosts”, and then click Details.)

3. Under the Borrow/ Preference Order column, specify the order in which you want to borrow from specific consumers.

The number 1 indicates that this consumer is your first choice to borrow from. Do not repeat any numbers.

(Out-of-the-box, there are no other leaf consumers to borrow from, so you do not see this field. Skip this step. If you want to see a configuration with multiple leaf consumers, change the resource plan view to “Resource Group: ManagementHosts”, and then click Details.)

4. Click Apply and then Close.
5. Run `ego restart` to have the changes take effect.

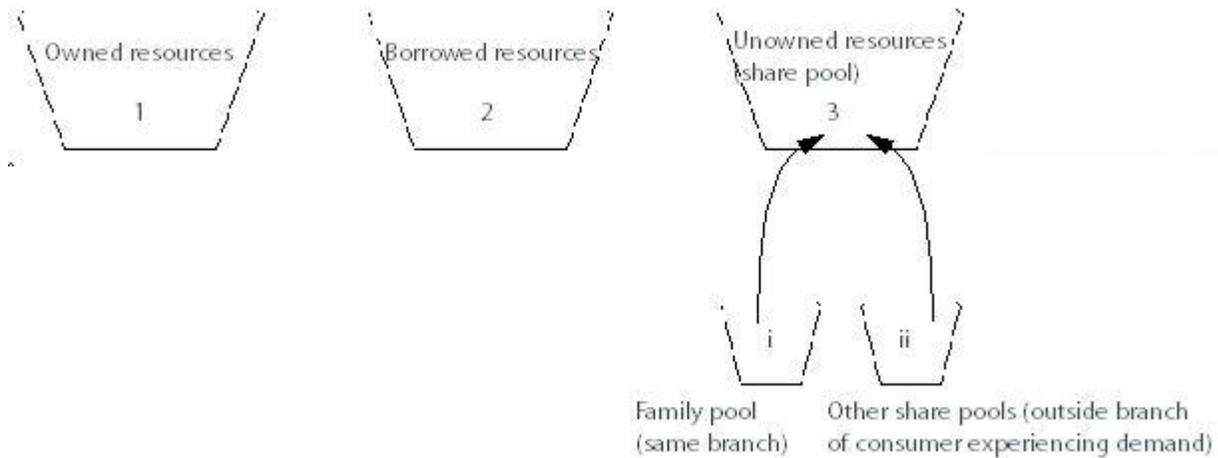
## 7. Understand share ratios

Sometimes there may be competition between leaf consumers from the same branch for additional resources. You must configure your resource plan to indicate how to appropriately satisfy the demands of these competing leaf consumers. This is done by specifying a share ratio.

When all the owned resources have been allocated according to the resource plan, and all resources that can be borrowed from other consumers have been borrowed, then EGO begins allocating unowned resources from a share pool. To plan how to allocate share pool resources, you need to specify a ratio value. Leaf consumers with a higher share ratio receives more resources during periods of competition.

To understand share ratios, it is important to first understand how resources generally are allocated. One way to do this is to picture three “buckets” containing owned resources, borrowed resources, and shared resources. Each bucket contains resources that get allocated to consumers according to the resource plan and to consumer demands. Resources are distributed from the buckets in this order:

1. EGO always allocates owned resources first to consumers according to the resource plan.
2. Then, when there are no owned resources left to borrow from consumers who are willing to lend them, EGO allocates unowned resources from the share pool to consumers with unsatisfied demand.
  - Share pool resources first come from the “family” pool (any unowned resources within a particular branch in the consumer tree).
  - Once the family pool is exhausted, then EGO distributes resources from other branches in the consumer tree, eventually moving up the tree to distribute any unowned resources from the cluster level.
  - Share pool resources are distributed to competing leaf consumers according to their allowed share ratio. If the share ratio for two competing consumers is 1:1, resources are allocated evenly. If the share ratio is 1:2, then the second competing consumer gets twice as many available resources from the share pool.
3. Finally, when there is demand, consumers lend and borrow resources from each other.



Why a ratio instead of an absolute number? Because only those consumers experiencing a current demand factor into the distribution calculation—because current demand fluctuates, making it difficult to predict who will have demand at any given time, a ratio is the best way to indicate a relative quantity of resources a particular consumer should receive.

## 8. Set share ratios

Before beginning, ensure you are still on the Consumers & Plans page, in the Resource Plan. The selected resource group should be “ComputeHosts”. Under Time Intervals and Settings, you must select Show Advanced Settings.

1. Under the Model type: Sharing section of the plan, check the Share Ratio box for the SampleApplications consumers and EclipseSamples leaf consumer, and enter the share ratio 1 for each. (Out-of-the-box there are no other leaf consumers competing for available resources, so the share ratio has no effect at this point.)

If there were other EclipseSamples leaf consumers, you might enter a share ratio of 1 for each of them. This would ensure that all leaf consumers share available surplus resources equally. If you wanted one leaf consumer to have twice as many resource made available during times of competition for resources, you would enter a share ratio of 2 for it, and 1 for the others.

2. Indicate a sharing limit for leaf consumers.

By default, a leaf consumer who is first in line to borrow (either because it is the first one to express a demand or because of its higher consumer rank), can borrow all the resources available for sharing if it has enough demand. If you want to limit the amount of resources a consumer receives through borrowing, you can indicate a limit in the Sharing Limit field. The borrowing limit is expressed as a numeric quantity.

Consumers Resource Plan

Resource Group: ComputeHosts Time Intervals and Settings

Consumer	Model type: Ownership			Model type: Share		
	Owned Slots	Consumer Rank	Lend   Limit	Borrow   Limit	Share Ratio	Limit
<input type="checkbox"/> Toronto	<input type="text" value="1"/>					
<input type="checkbox"/> SampleApplications	<input type="text" value="1"/>	<input type="text" value="50"/>			<input checked="" type="checkbox"/>	<input type="text" value="1"/>
<input type="checkbox"/> EclipseSamples	<input type="text" value="1"/>	<input type="text" value="50"/>	<input checked="" type="checkbox"/> <input type="text" value="1"/> Details	<input checked="" type="checkbox"/> <input type="text" value="1"/> Details	<input checked="" type="checkbox"/>	<input type="text" value=""/>
<b>Total</b>	1	-	-	-	-	-
<b>Balance</b>	0	-	-	-	-	-
<b>Total</b>	1	-	-	-	-	-
<b>Balance</b>	0	-	-	-	-	-

Check to apply a share ratio to this consumer  
 Enter a share ratio, applicable between consumers at the same level in the same branch  
 Enter a maximum number of resources available for sharing that this consumer can borrow  
 Note: Leave the field blank to indicate unlimited borrowing of resources available for sharing

## 9. Understand consumer rank

Resources are borrowed between siblings according to their consumer ranking: sibling consumers (consumers at the same level of the same branch) with the highest priority setting have the first opportunity to borrow if they are simultaneously competing for resources.

When carrying out the resource distribution plan, EGO considers consumer rank in deciding which sibling consumer to apply the plan to first. If parts of the plan compete with each other, the plan is first applied to sibling consumers with a higher consumer ranking.

An assigned consumer rank is only considered when lending unused or surplus resources to consumers with a resource need; those borrowing consumers with the highest set priority are lent resources first. Note that a child's rank has nothing to do with the rank of its parent.

## 10. Set consumer rank

Before beginning, ensure you are still on the Consumers & Plans page, in the Resource Plan. The selected resource group should be "ComputeHosts". Under Time Intervals and Settings, you must select Show Advanced Settings.

- Under the Model Type: Ownership section of the plan, rank consumers within the SampleApplications branch. (Out-of-the-box, there is only 1 top-level consumer and 1 leaf consumer, so the ranking has no effect at this point.)

Specify any positive whole number in the Consumer Rank field, where 1 is the highest priority. Priority settings are relative to one another within the resource group. If you leave the priority blank, that consumer has no priority over any other consumer (it does not form part of any consumer ordering/sequencing). If all consumers have a high ranking, any advantage one may have over another is nullified. Be selective in assigning a high consumer rank to a consumer.

Consumers		Resource Plan	
<b>Resource Group: ComputeHosts</b>			
		00:00	
		24:00	
<b>Model type: Ownership</b>			
Consumer	Owned Slots	Consumer Rank	
<input type="checkbox"/> Toronto	<input type="text" value="1"/>		
<input type="checkbox"/> SampleApplications	<input type="text" value="1"/>	<input type="text" value="50"/>	
<input checked="" type="checkbox"/> EclipseSamples	<input type="text" value="1"/>	<input type="text" value="50"/>	
<b>Total</b>	<b>1</b>	<b>-</b>	
<b>Balance</b>	<b>0</b>	<b>-</b>	
<b>Total</b>	<b>1</b>	<b>-</b>	
<b>Balance</b>	<b>0</b>	<b>-</b>	

Assign a rank to each sibling consumer, where 1 is the highest priority.  
 Here, EclipseSamples does not have a sibling.  
 Note: Leave the field blank if this consumer has no priority over any other sibling consumer.

Note:  
 The consumer rank settings for SampleApplications and EclipseSamples do not affect each other because they are at different levels of the tree. A consumer's rank is only in effect for consumers with the same "parent" (who are siblings).

- In future, repeat step 1 for other consumer branches in the consumer tree.

For low priority consumer branches, you can choose to leave the Consumer Rank field blank.

## 11. Export or import resource plan

Any changes you apply to the resource plan in the Platform Management Console are implemented immediately (if you click Apply). We recommend exporting the plan instead, saving it under a different name for future import. That way, the original default resource plan remains available to you for future re-use. This procedure walks you through how to export a completed plan, and then how to import it at a later time.

- Export the plan.

If you do not want to immediately apply the resource plan you have created, export it for later use.

- At the bottom of the Resource Plan page, click Export.  
 A dialog box opens, asking if you want to save or open the file. Click Save.
- Navigate to a local directory location to save your resource plan.

You may want to give the XML file a unique name including date or plan specifics so that you can import the plan of your choice quickly and easily at a later time.

Changing your consumer tree structure after exporting a plan may introduce errors to the plan when it is later imported. Check your plan carefully when you later import it if the consumer tree structure has changed, and make any required modifications to it.

## 2. Import plan at a later time.

To import a saved resource plan at a later time, follow these steps.

1. Ensure that a resource plan exists (and was exported).
2. Ensure that the resource plan you import has an identical consumer tree structure to the active resource plan. If not, check your plan carefully and make any required modifications. (For example, if the current tree has additional leaf consumers, the imported plan may need to distribute resources to the added leaf consumers and apply lend/borrow policies.)

It is ok if the time intervals are different.

3. Click on your cluster name in the consumer tree.
4. Click Consumer > Consumers & Plans > Resource Plan.

If you do not see Resource Plan, make sure you are at the top of your consumer tree.

5. At the bottom of the page, click Import.

The Import a resource plan (XML) dialog displays.

6. Browse for the location of the XML resource plan you wish to import.
7. Click Import.

Importing a resource plan makes it the active plan. You can export this plan and import it again as needed if you want it to apply it only to specific days of the week. You can create as many resource distribution plans as you want and import them as needed for a quick change in your distribution.



---

# Index

## B

branches  
about 9

## C

child  
consumer 9  
cluster  
distribute resources 23  
ClusterServices  
about 8  
ClusterServices consumer  
about 9  
consumer administrators  
about 9  
consumers  
about 8, 10  
branch 9  
child 9  
default 8  
leaf 8  
parent 9  
sibling 9  
top-level 8

## D

descendants  
about 9

## L

leaf consumers  
about 8  
lending  
about 24

## M

ManagementServices  
about 8

ManagementServices consumer  
about 9

## O

overlaps  
between resource groups 17  
ownership  
about 23

## P

parent  
consumer 9

## R

resource groups  
creating by  
hosts 16  
static (list of names) 16  
resource plan  
about 8  
overview 22  
resource planning  
by time 23  
resources  
lending 24  
owning 23  
root  
consumer tree  
about 8

## S

SampleApplications  
about 8  
SampleApplications consumer  
about 9  
sibling

consumer 9  
static (list of names)  
creating a resource group 16  
T  
tree

about 8