

# NGASI Shared-Runtime Manager Administration and User Guide

© 1999-2010 WebAppShowcase DBA NGASI



# Table of Contents

	0
<b>Part I Introduction</b>	<b>4</b>
1 Overview .....	4
2 Requirements.....	4
<b>Part II Administrator</b>	<b>6</b>
1 Login .....	6
2 Resellers/Webhosts .....	6
Create Reseller .....	6
Delete Reseller .....	7
3 Runtimes .....	7
Create Runtime .....	8
4 Security Manager.....	8
<b>Part III Resellers/Webhosts</b>	<b>11</b>
1 Login .....	11
2 Create User .....	11
3 Edit User .....	12
4 Delete User .....	12
<b>Part IV Users</b>	<b>14</b>
1 Login .....	14
2 Applications.....	15
Deploy New .....	15
Deploy From List .....	16
Virtual Hosting .....	16
Delete Deployed Application .....	17
File Privileges .....	17
Delete Application Archive .....	17
3 Services .....	17
Databases and DataSources .....	17
Mail .....	18
4 Configuration.....	18
5 Restart .....	18
6 Examples .....	18
7 Troubleshooting.....	19
File Access .....	19
<b>Index</b>	<b>0</b>



# 1 Introduction

NGASI Shared-Runtime Manager is an add-on module for NGASI AppServer Manager, that enables multiple user applications to run within a shared Runtime (JVM).

## 1.1 Overview

The first true Shared-runtime manager. Enabling web providers budget hosting and affordable reseller plans that provide capabilities for end users to easily deploy and manage their applications. NGASI Shared-Runtime Manager provides great flexibility by allowing Application Server choices as well as allocating the application server runtimes at the reseller level or globally.

Using NGASI Shared-Runtime Manager's custom Security Manager, users are able to deploy applications, such as those written with JAVA, in the same Virtual Machine, shared by other accounts. With the Security Manager, each application is insulated from the other within its own sandbox.

## 1.2 Requirements

NGASI Shared-Runtime Manager requires a valid installation of [NGASI AppServer Manager](#).

**If the NGASI AppServer Manager version is below 9.0** you must follow the steps below prior to proceeding:

1) Follow the steps to install the latest NGASI AppServer Manager update at the link below:

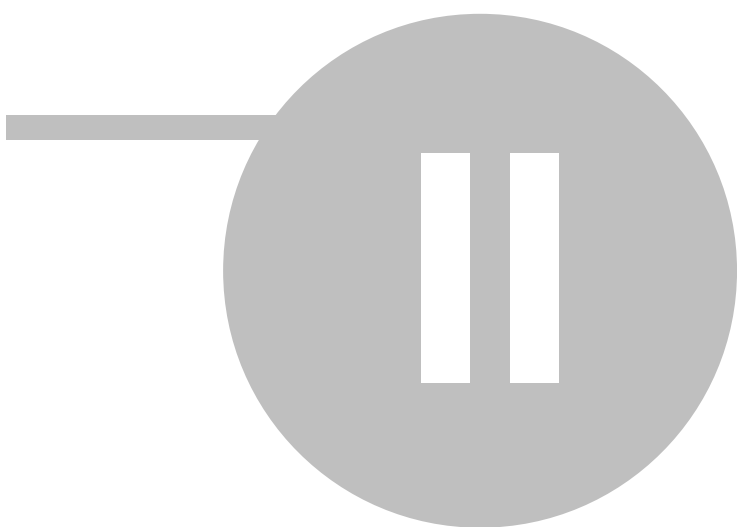
<http://www.ngasi.com/ngasihelp/ngasiadminguide/updates.htm>

2) Before starting up NGASI AppServer Manager, download the following file:

<http://downloads.ngasi.com/misc/web.xml>

and copy to the following directory:

/usr/ngasi/webapps/            /WEB-INF



## 2 Administrator

The Administrator "admin" is the NGASI Shared-Runtime Manager super user. It is used for managing all accounts, which includes Resellers and Users.

### 2.1 Login

The host name and port should be the same as NGASI AppServer Manager.

So assuming the default HTTPS port 8663 and host name localhost:

Login URL: <https://localhost:8663/modules/srm/admin/index.zul>

The login Administrator is "admin" and the default password is the same as the "admin" password for NGASI AppServer Manager. The default "admin" password is "coolgeek". To change the "admin" password requires login in to NGASI AppServer Manager (<https://localhost:8663/zp/appservermgr>)

### 2.2 Resellers/Webhosts



Use this feature to manage Resellers (Webhosts).

Resellers manages the end user accounts.

By default the "admin" account is also a built-in Reseller account that cannot be removed.


If you just want the "admin" account to function as a Reseller for all User Accounts, then skip to the [Resellers/Webhosts](#) section.

#### 2.2.1 Create Reseller



Click the "Create Reseller" to create Reseller Accounts.

Select a Reseller in the Select-box.

Then click the "Add" button .

The Resellers listed in the Select-box are derived from the installed Control Panel.

#### Plesk

A Plesk Client or Reseller Account is considered as a NGASI Shared-Runtime

Reseller Account.


### **cPanel**

A cPanel account with cPanel Reseller privileges is considered as a NGASI Shared-Runtime Reseller Account. By default, cPanel users are owned by "root".

### **DotNetPanel**

The default Reseller is the "admin" account.


Once the NGASI Shared-Runtime Reseller Account is Added, the Reseller should appear in the "Reseller List". The Reseller Login and management is detailed in the [Resellers/Webhosts](#) section.

As the Administrator, you can login directly to the Reseller Account by clicking on the corresponding  icon in the "Reseller List".

## **2.2.2 Delete Reseller**

NOTE: Deleting a Reseller will automatically Delete any User Accounts created by the Reseller. Also the embedded "admin" Reseller cannot be Deleted.

Select one or more Resellers in the "Reseller List".

Then Click the "Delete User" button to delete the selected Reseller Accounts. 

## **2.3 Runtimes**



Click the "Runtimes" button in the Left Menu

As NGASI Shared-Runtime deploys and manages applications in a shared Runtime environment, 1 or more Runtime needs to be made available to the application User Account.

A Runtime is a specific Application Server running on a specific JDK (Virtual machine). So it is essentially a combination of Application Server and JDK. For example:

Tomcat 6.0.18/JDK6.0\_10

Tomcat 5.5.27/JDK5.0\_11

2 types of Shared-Runtime that can be set are Pooled and Exclusive. Pooled Runtime is made available to all User Accounts across all Resellers.

This is ideal in situations where system resource is limited, i.e. in a VPS. Exclusive Runtimes are available only to User Accounts belonging to a specific Reseller. Exclusive Runtimes is ideal in environments with many resellers. Performance is also much better when Exclusive Runtimes are used as the environments are partitioned by Resellers.

NOTE. You can also have a mixed environment, where one Runtime may be Pooled while another is made Exclusive. For example since Tomcat 5.5.27/JDK5.0\_11 is an older platform, less users may be inclined to use it, so then you may consider making that Pooled. Then for the latest platform such as Tomcat 6.0.18/JDK6.0\_10, you make Exclusive since you would expect more demand for it.

If NO Runtime was created, by default, a Pooled Runtime with the latest Platform is automatically created.

### 2.3.1 Create Runtime



Click the "Create Runtime" button in the Top menu to create Runtimes. Select the Application Server and Memory; set whether Pooled or Exclusive. If "Exclusive" is checked, a Selection box of Resellers will appear. Select the desired Reseller.

Finally click the "Add" button. Once completed, the Runtime will appear in the "Runtime" List.

## 2.4 Security Manager

The custom Security Manager is what makes the Shared-Runtime possible. Each application is run in a sandbox insulated from other applications within the same Runtime machine and with very little system access privileges. Below are the specific access that each application has:

#### **Application system access:**

- 1)TCP connection to any host and port.
- 2)Read and Write access within the application directory.

NOTE: As applications are deployed in a shared environment, the Security Manager enforces a strict policy, thus limiting certain access. This means that some applications will not work under such an environment. In this case, the user should consider a Private Runtime environment as provided with NGASI AppServer Manager.

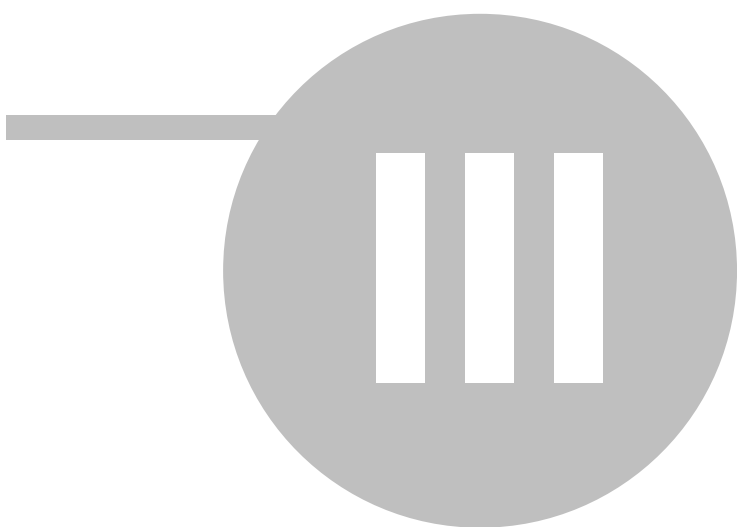
---

Below are some specific differences of Private Runtime (Private JVM) vs Shared Runtime (Shared JVM):

- TEMPORARY disruption due to Runtime restart.
- Limited system access
- Applications cannot execute another program
- shell access is not required
- read-write access limited to application directory
- All Applications share the same memory in the JVM.
- The client will have to configure their own logging. This is normally done with log4j.

NGASI Shared-Runtime Manager is intended for simple database driven applications not requiring much system resources, in terms of RAM/CPU usage.

Even though Applications are deployed in a Shared Runtime environment, NGASI Shared-Runtime Manager, allows per application configuration of DataSource, MailSession, and Realms; thus enabling support for many standard based application frameworks like Spring and Struts.



## 3 Resellers/Webhosts



Use this feature to manage User Accounts.  
User Accounts are the end users that deploys and manages the applications.

### 3.1 Login

Login Info:

The host name and port should be the same as  
NGASI AppServer Manager.

So assuming the default HTTPS port 8663 and host name localhost:

Login URL: <https://localhost:8663/modules/srm/host/index.zul>

The user login information would be the same as for your Control Panel  
(Plesk or cPanel/WHM).

### 3.2 Create User



Click the "Create User" to create User Accounts.

Select a User in the Select-box.

Then click the "Add" button .

The Users listed in the Select-box are derived from the installed Control Panel.

#### **Plesk**

A Plesk Domain user is considered as a NGASI User Account.

#### **cPanel**


A regular cPanel account with Domain privileges is considered as a  
NGASI User Account.

#### **DotNetPanel**

A DotNetPanel Customer User Account with FTP Management Privilege  
is considered as a NGASI User Account.

**NOTE:** The NGASI DotNetPanel user is created with the default password  
"coolgeek".

The user should change the default password after login.

Once the User Account is Added, the User should appear in the "User List". The User Login and management is detailed in the [Users](#) section. As the Reseller, you can login directly to the User Account by clicking on the corresponding  icon in the "User List".

### 3.3 Edit User

Select an Accounts in the "User List".

Then Click the "Edit User" button to edit the selected User Account.



After making the appropriate changes, click the "Save" button.



### 3.4 Delete User

NOTE: Deleting a User will automatically Delete any applications deployed by the User.

Select one or more Accounts in the "User List".

Then Click the "Delete User" button to delete the selected User Accounts.





## 4 Users

The User accounts is the end user that deploys and manages the applications.

NOTE: As applications are deployed in a shared environment, the Security Manager enforces a strict policy, thus limiting certain access. This means that some applications will not work under such an environment. In this case, the user should consider a Private Runtime environment as provided with NGASI AppServer Manager.

Below are some specific differences of Private Runtime (Private JVM) vs Shared Runtime (Shared JVM):

- TEMPORARY disruption due to Runtime restart.
- Limited system access
- Applications cannot execute another program
- shell access is not required
- read-write access limited to application directory
- The client will have to configure their own logging. This is normally done with log4j.

NGASI Shared-Runtime Manager is intended for simple database driven applications not requiring much system resources, in terms of RAM/CPU usage.

Even though Applications are deployed in a Shared Runtime environment, NGASI Shared-Runtime Manager, allows per application configuration of DataSource, MailSession, and Realms; thus enabling support for many standard based application frameworks like Spring and Struts.

### 4.1 Login

Login Info:

The host name and port should be the same as NGASI AppServer Manager.

So assuming the default HTTPS port 8663 and host name localhost:

Login URL: <https://localhost:8663/modules/srm/user/index.zul>

The user login information would be the same as for your Control Panel (Plesk, cPanel/WHM, or DotNetPanel)

**NOTE:** Your Administrator is able to configure your Control Panel to enable direct access from the Control Panel to NGASI Shared-Runtime Manager. ([http://www.ngasi.com/ngasihelp/ngasiadminguide/control\\_panel\\_button.htm](http://www.ngasi.com/ngasihelp/ngasiadminguide/control_panel_button.htm))

**NOTE:** The NGASI DotNetPanel user is created with the default password "coolgeek".

The user should change the default password after login.

## 4.2 Applications



This section covers topics about application management.

By default, applications are deployed under an Application Server running as ngasirun system user.

To enable the application to be deployed under the web account user's HTTP directory and to also enable FTP access to the application directory, then the root system user (system under Windows) would need to be used to run the applications. If the Application Server system user is other than root (or system on Windows), then FTP access is disabled.

All applications with FTP access are deployed under the HTTP document directory.

### **Plesk:**

For a Plesk user with Domain mydomain.com, an application, myapp, will be deployed as follows:

`/var/www/vhosts/mydomain.com/httpdocs/myapp`

or

`C:\inetpub\vhosts\mydomain.com\httpdocs\myapp`

### **cPanel:**

For a cPanel user named user1, an application, myapp, will be deployed as follows:

`/home/user1/public_html/myapp`


### **DotNetPanel:**

For a DotNetPanel user named user1, an application, myapp, will be deployed as follows:

`C:\HostingSpaces\user1\webapps\myapp`

For applications with FTP disabled, the application is deployed in the webapps directory of the Application Server.

### 4.2.1 Deploy New

Click the "Deploy New"  button to Upload a new Application Archive for deployment.

Fill out any necessary information then click "Continue"  to proceed.

Once successfully deployed, an Icon symbolizing the deployed application will be added to the Canvas display.



#### 4.2.2 Deploy From List

NGASI Shared-Runtime Manager enables you to redeploy an Application that has already been uploaded, so you do not have to re-upload the same application each and every time.

Click the "Deploy from list" button.



Drag-and-drop desired Application from the Application Factory list to the Canvas.

Click the "YES" button to proceed.

#### 4.2.3 Virtual Hosting

Application Virtual Hosting (not to be confused with Web Server Virtual Hosting) maps an application to the root of one or more Domain hostnames.

For example an application, myapp, deployed on [www.mydomain.com](http://www.mydomain.com):


<http://www.mydomain.com/myapp>

could be accessed instead as

<http://www.mydomain.com>

with Virtual Hosting enabled.

Click on the icon, in the Canvas, of desired Application. Then click the

"Virtual Hosting"  button

(you can also drag-and-drop the icon to the button as well).

Select one or more Domain hostnames in the Selection Box, then


Click "Add"  to proceed.

NOTE: To remove any Virtual Hosting that was set, you would

click the "Delete"  button instead.

#### 4.2.4 Delete Deployed Application

To delete a deployed application, click on the icon, in the Canvas, of desired Application.

Then click the "Delete"  button  
(you can also drag-and-drop the icon to the button as well).

#### 4.2.5 File Privileges



For FTP access to the deployed Application, file privileges may need to be reset.

Click on the icon, in the Canvas, of desired Application. Then click the

"File Privileges" button   
(you can also drag-and-drop the icon to the button as well).

#### 4.2.6 Delete Application Archive

Use this option to remove an Uploaded Application Archive from the Application Factory.

Click the "Deploy from list" button.   
Drag-and-drop desired Application from the Application Factory list to the  
"Delete"  button.

Click "Continue"  to proceed.

### 4.3 Services

Even though Applications are deployed in a Shared Runtime environment, NGASI Shared-Runtime Manager, allows per application configuration of DataSource, MailSession, and Realms; thus enabling support for many standard based application frameworks like Spring and Struts.

#### 4.3.1 Databases and DataSources

During the process of adding and application to the repository, a DataSource, for the application may be defined. And at deployment, NGASI is able to automatically configure the DataSource for the application.


### 4.3.2 Mail

During the process of adding and application to the repository, a Mail Session Object, for the application may be defined. And at deployment, NGASI is able to automatically configure the Mail Session Object for the application.

## 4.4 Configuration

If more than 1 Application Server is available to your account, you may change the Application Server or other aspects of your setup by following the steps below.

Click the "Configuration"  button in the Left Menu.

Make appropriate changes. Then click the "Update"  button to proceed.

## 4.5 Restart

Sometimes you may have a need to Restart the Application Server. Click the "Restart" button in the Left Menu to do so.

## 4.6 Examples

Below are descriptions and links to some sample Applications:

[javawebexamples.war](#) 

This is just the classic Java Examples that accompanies most Java Application Servers. Due to the Security Manager, some of the examples will not work.

[sqltestapp.war](#) 

This is a simple database driven counter application. The value of the counter is stored in a Database. The Database is accessed by a DataSource object, named "GenericDataSource". In addition, the application requires an SQL script be run to create the necessary tables. NGASI Shared-Runtime Manager is able to do so by uploading the SQL script.

The required SQL script is [sqltestapp.sql](#).

[struts-examples.war](#) 

The most popular JAVA Web Application Framework. Due to the Security Manager, some of the examples will not work.

More information on Struts can be found at the following URL:

<http://struts.apache.org>

[zkdemo.war](#) 

ZK is a popular JAVA based AJAX Framework.

Due to the Security Manager, some of the examples will not work.  
More information on ZK can be found at the following URL:  
<http://www.zkoss.org>

## 4.7 Troubleshooting

As mentioned before, your Application is run under a Security Manager, which limits system and resource access. This may result in your application failing to startup with a SEVERE error. In such a case the browser usually reports a 404 result, with a detailed message as

The requested resource () is not available

Below are some tips to follow to help make sure your application runs with a Security Manager.

### 4.7.1 File Access

All references to Files and file paths should include the full path name. Also by default, system read/write access is limited to resources within the application directory.

So if referring to log,log4j, and configuration files, etc., the above guidelines should be followed.

Endnotes 2... (after index)

Back Cover