



# NGASI AppServer Manager

## White Paper

In the world of IT, SaaS, and Web Hosting, the management of JAVA application servers and their applications becomes increasingly complex as the number grows. Traditionally the solution has been to throw more consultants and hardware as a solution. Even though 99% of the tasks associated with provisioning were the same in the process. Thus in most cases it ends up being a costly consulting service. Won't it make sense then to automate those same repetitive tasks and functions to save time and money?

NGASI AppServer Manager automates multiple installations and configurations for Application Servers and Applications on a single server machine, across multiple virtual servers, and across multiple physical servers. NGASI AppServer Manager is designed to automate 99% of deployments, so consultants and IT staff can concentrate on the other 1% that really needs personal attention.

One of the powerful feature included is the ability to centrally manage end-to-end deployments across remote servers. This is in addition to central management on a single server or across multiple virtual servers. Plus the Application Factory enables 1 click installation of standard applications.

For IT Managers, NGASI AppServer Manager enables central management of application server assets and resources spread across the enterprise. And for Service Providers, it enables SaaS-Enabled JAVA Hosting.

NGASI AppServer Manager may be broken down into 3 distinct JAVA automation solutions – Private JVM JAVA Hosting, SaaS Enablement, IT and

Enterprise Management.

### **Private JVM JAVA Hosting**

NGASI AppServer Manager was designed with hosted environments in mind.

NGASI AppServer Manager enables Private JVM JAVA Hosting on a shared server for multiple accounts. Our architecture allows a level of security

beyond regular Shared JVM hosting. NGASI AppServer Manager

uses the system user, of a Web Account created by a Web Control Panel, or just a regular standalone system user, which is required to run

Web Applications. The web account user is therefore able to install individual application servers and JVMs via the NGASI AppServer Manager control panel.

NGASI accounts are easily created via the NGASI AppServer Manager Admin interface or via a script (ideal for account automation).

### **SaaS Enablement**

A major part of the service provided by SaaS infrastructure providers, is consulting.

Many of the functions provided by the consulting, are repetitive and mundane during

the deployment and provisioning stages. These tasks should be automated. Such

automation would lead to quicker turn around and lowered costs thus facilitating

wider adoption to the SaaS model.

An example is a software solution developed with the JAVA EE technology,

as standard JAVA applications tend to be more complex than

others. In this example a company has developed a sales and services software

solution for cellular retailers (essentially the local cellular stores you go to purchase

your wireless services). The software developer can A) package the software and license it to the cell retailer using the traditional licensing model. In this case, the cell provider will have to hire a consultant to deploy the application, plus pay a traditional hosting provider or host it in house on a server (so even more work/costs thus not a viable solution). B)The software developer can host the software for each customer. So that the only challenge to the customer will be the training aspects of the software. Under scenario B, the software developer can employ the services of a SaaS infrastructure provider to manage the hosting of the application themselves. So the provider is responsible for installing applications servers, the application, and configuring it for each customer. This is more efficient than A but there is much more efficiency to be add, especially as the customer scales up to many customers. What if there is a way to automate the installation and configuration process. So by packaging the application in the standard SWAR format, and utilizing a SaaS automation engine, such as NGASI AppServer Manager, the provider is able to cut down on time and costs. Some functions of the Automation tool, should also be made accessible to the software developer as well, thus giving the software developer more manageable control of the process.

## **IT and Enterprise Management**

For IT Managers, NGASI AppServer Manager enables central management of application server assets and resources spread across the enterprise.

NGASI AppServer Manager enables central management of application server installations spread across the enterprise. Traditionally, each department, requests server resources from IT in order to install their favored application server. Time passes, and people move on. Corporations are left with installs unaccounted for,

some with sensitive data. With NGASI AppServer Manager, IT Management can take control, and yet satisfy the needs of individual departments.

NGASI AppServer Manager key benefits include:

### **Multiple JDKs**

Accounts users have the ability to choose which JDK and JDK version to use to run their individual Application Server. So one user can be using JDK 5.x while at the same time another account might be using JDK 6.x.

### **Multiple Application Servers**

Similar to JDK choices, accounts users have the ability to choose which Application Server and Application Server version to run their individual applications. So one user can be using Tomcat while at the same time another account might be using JBoss. In addition an account may be using Tomcat 5.x while another account is able to use Tomcat 6.x at the same time all on the same server. This is because NGASI AppServer Manager sets up separate Application Server installations for each account. Thus not only is each account running a separate JVM and Application Server instance, but each account is running its own Application Server installation. This mean that each account can configure their Application Server, install their own applications and class versions – just as if on their own private server or machine.

### **Application Factory**

The **Application Factory** is our stack of common standards based JAVA applications. Applications are packaged in the open SWAR format to allow one click installation.

### **SWAR Engine (SaaS Web Application Archive Packaging format)**

SWAR is an open application packaging format that enables standard applications to be deployed with a descriptor that instructs a SWAR compliant engine, such as NGASI AppServer Manager, how to deploy it. Instructions include automated DataSource configuration, JDBC Realm configuration, default login user setup, and much more. Thus SWAR allows developers to just concentrate on their core competencies instead of having to deal with the mundane deployment tasks, that may include manually configuring the application.

## **Backups**

The Backup tool automates the backing up of account's application server files, such as:

webapps web application deployment directory

conf configuration directory

bin startup directory

private data directory

## **Central Management across multiple Virtual Servers (VPS)**

NGASI AppServer Manager supports OS based virtualization, such as SWSOFT Virtuozzo and openVZ. NGASI AppServer Manager support for these VPSs includes:

Central management of Application Server installations for cPanel/WHM, Plesk, Helm or standalone accounts within multiple VPSs. For Hypervisor based VPS such as VMWare and XEN, refer to the Remote Servers section.

## **Central Management across multiple remote servers**

One of the more powerful features of NGASI AppServer Manager, is its ability to allow management of remote servers as if on the same server as NGASI AppServer Manager is installed on.

