Opswise Controller 5.2.0

Installation and Administration Guide

1. Installation, Upgrade, and Applying Maintenance . . . 3
- 1.1 Installation, Upgrade, and Applying Maintenance - Overview . . . 4
- 1.2 Opswise Controller Installation . . . 5
  - 1.2.1 Pre-Installation Procedure . . . 6
    - 1.2.1.1 Determining Space Requirements . . . 7
    - 1.2.1.2 Installing Opswise Controller Prerequisites . . . 8
      - 1.2.1.2.1 Downloading Java Runtime Environment . . . 9
      - 1.2.1.2.2 Installing Apache Tomcat . . . 10
      - 1.2.1.2.3 Installing a Database . . . 13
    - 1.2.1.3 Downloading Opswise Controller Software . . . 16
  - 1.2.2 Installing Opswise Controller . . . 18
  - 1.2.3 Installing Opswise Bundled Controller on AIX . . . 27
  - 1.2.4 Adding a Cluster Node . . . 34
- 1.3 Opswise Controller Upgrade and Maintenance . . . 39
  - 1.3.1 Upgrading Opswise Controller . . . 41
  - 1.3.2 Applying Maintenance to Opswise Controller . . . 49
- 1.4 Starting and Stopping Opswise Controller . . . 51
  - 1.4.1 Starting and Stopping Opswise Controller - UNIX and Windows . . . 52
  - 1.4.2 Starting and Stopping Opswise Bundled Controller on AIX . . . 53

2. Administration . . . 54
- 2.1 Administration Overview . . . 55
- 2.2 High Availability . . . 56
- 2.3 Maintenance Scripts . . . 63
- 2.4 Ports Configuration . . . 69
- 2.5 Opswise Controller Properties . . . 70

Installation, Upgrade, and Applying Maintenance

Overview

Installation, Upgrade, and Applying Maintenance - Overview

Installation Instructions

Overview

Pre-Installation

Installing Opswise Controller Prerequisites

Downloading Java Runtime Environment

Installing Apache Tomcat

Installing a Database

Installing Opswise Controller

Installing Opswise Bundled Controller on AIX

Adding a Cluster Node

Upgrade Instruction

Upgrading Opswise Cor

Applying Maintenan

Applying Maintenance t

Starting and Stoppin

Overview

Starting and Stopping tl

Starting and Stopping C

The information on these pages also is located in the Opswise Controller 5.2.0 Installation and Administration Guide.pdf.

Installation, Upgrade, and Applying Maintenance - Overview

- Installation, Upgrade, and Applying Maintenance
  - Installation
  - Upgrade
  - Applying Maintenance

Installation, Upgrade, and Applying Maintenance

There are separate procedures for installing, upgrading, and applying maintenance to Opswise Controller.

Installation

Installation refers to the installation of a Controller on a machine with any supported platform that does not already contain an installed Controller.

If you are installing Opswise Controller for the first time, see Opswise Controller Installation for instructions.

Upgrade

Upgrading refers to the increase of a currently installed Version, Release, or Modification level (see Versioning) of a Controller on a machine to a later Version, Release, or Modification level of that Controller (for example, upgrading Controller 1.7 or Controller 5.1.0.7 to Controller 5.2.0.0).

If you are upgrading from a previous version of Opswise Controller, see Upgrading Opswise Controller for instructions.

Applying Maintenance

Applying maintenance refers to the increase of a currently installed Maintenance level (see Versioning) of a Controller on a machine to a later Maintenance level of that Controller (for example, applying maintenance to a 5.2.0.0 Agent to increase its maintenance level to 5.2.0.2).

Exception for Opswise Controller 5.2.0 only

You can use the 5.2.0.x maintenance release of Opswise Controller to apply maintenance to any Opswise Controller 5.1.1.x release. You do not have to perform an upgrade.

If you are applying maintenance to your version of Opswise Controller, see Applying Maintenance to Opswise Controller.

Opswise Controller Installation

Opswise Controller Installation

The Opswise Controller is a Java web application running in a Tomcat web container. For this reason, the Opswise Controller software and the procedure for installing Opswise Controller on UNIX or Windows is basically the same.

However, the procedure for installing Opswise Bundled Controller on AIX, which includes installation of Apache Tomcat, is different.

⚠ **Note**

- If you are upgrading from a previous version of the Controller - for example, from Controller 5.1.0.7 to Controller 5.2.0.0 - see Upgrading Opswise Controller for instructions.
- If you are applying maintenance to the Controller - for example, applying maintenance to Controller 5.2.0.0 to increase its maintenance level to Controller 5.2.0.3 - see Applying Maintenance to Opswise Controller for instructions.

Pre-Installation Procedure

Pre-Installation Procedure

Before you install Opswise Controller or Opswise Bundled Controller for AIX, you must perform the following pre-installation procedure:

Step 1	Determine the space requirements for Opswise Controller software and the Opswise Controller database.
Step 2	Install all required Opswise Controller prerequisites.
Step 3	Download the platform-specific Opswise Controller distribution file from the Stonebranch website.

⚠ **Note**
You can install the Controller before, during, or after installation of Opswise Universal Agent.

Determining Space Requirements

- Overview
- Controller Space Requirements
- Database Space Requirements
    - Calculating Space Requirements
    - Output Retrieval

Overview

The following space requirements must be determined for the Controller and its database.

Controller Space Requirements

The Opswise Controller war file is approximately 60MB compressed and 200MB uncompressed, using a total of approximately 260MB of space when fully deployed.

However, the space requirements for the Controller are driven largely by logging. Logging requirements are based on the log levels selected in the Opswise Log Level and Platform Log Level Opswise Controller system properties.

A minimum 2GB of space is recommended for logging and other operations that require the Controller file system, such as bulk (and list) import/export.

The Log File Retention Period in Days Opswise Controller system property lets you specify the number of days that a Controller log file (and an Agent log file) is retained before it is purged. The default is 5 days.

Database Space Requirements

Each type of database software (MySQL, Microsoft SQL Server, Oracle) takes up different amounts of space. However, the space required for saved Controller data is the same; that is, for example, 1,000 tasks consume no more space in MySQL than they do in Oracle.

Calculating Space Requirements

Following the initialization of the Controller database, the initial table space size will be approximately 60MB.

Based on calculations using data from all task types, each Controller task instance consumes approximately 10KB of database space. You should estimate space requirements for your data based on your expected number of task executions per day and the duration for retaining history and activity data before purging.

Output Retrieval

An Agent always caches output. Output is stored in the database only if you do one or more of the following:

- Select Automatic Output Retrieval for a task.
- Create Email Notifications with output attachments for task.
- Retrieve output for a task instance.

A retrieved output file of 1K (for example) will require 2KB to 2.5KB of space in the database.

Installing Opswise Controller Prerequisites

Before installing Opswise Controller, you first must install the following prerequisites:

1. Java Runtime Environment
2. Apache Tomcat
3. Database

⚠ **Note**
If you will be installing the Opswise Bundled Controller on AIX, you do not need to install Apache Tomcat.

Downloading Java Runtime Environment

To download the Java Runtime Environment (JRE), access the Oracle site for Java JREs and download the appropriate package for your platform:

http://www.oracle.com/technetwork/java/javase/downloads/index.html

⚠ **Note**
JRE levels 6 and 7 are supported.

Installing Apache Tomcat

- Install Apache Tomcat
- Start and Validate Apache Tomcat

Note
Apache Tomcat versions 6.0.x and 7.0.x are supported.

Install Apache Tomcat

Perform the following steps to install Apache Tomcat (download and installation procedure for Apache Tomcat may vary a bit for each platform):

Step 1	Select an appropriate method of installation: **Windows** We recommend using the GUI installer to create the Apache Tomcat Service: 1. Download the "32-bit/64-bit Windows Service Installer" from Tomcat 6.0.xx or Tomcat 7.0.xx. 2. Follow the instructions to install the package. **Windows or Linux/Unix** Download a tar.gz or zip package that you unzip into a directory: 1. Download an appropriate package from Tomcat 6.0.xx or Tomcat 7.0.xx. 2. Follow the instructions to unzip the appropriate package (tar.gz or zip) into a directory on your file system. **Linux/Unix: Redhat and Centos distributions** Instead of downloading a tar.gz or zip package, you can use the **yum** installer.

Step 2	In order to accommodate large workloads, Opswise Controller requires that you update the JVM variables to the following minimum values:
	AIX
	`CATALINA_OPTS="-Xms512m -Xmx1024m"`
	z/Linux
	`CATALINA_OPTS="-Xms512m -Xmx1024m -Xjit:optLevel=noOpt"`
	All other platforms
	`CATALINA_OPTS="-Xms512m -Xmx1024m -XX:MaxPermSize=256m"`

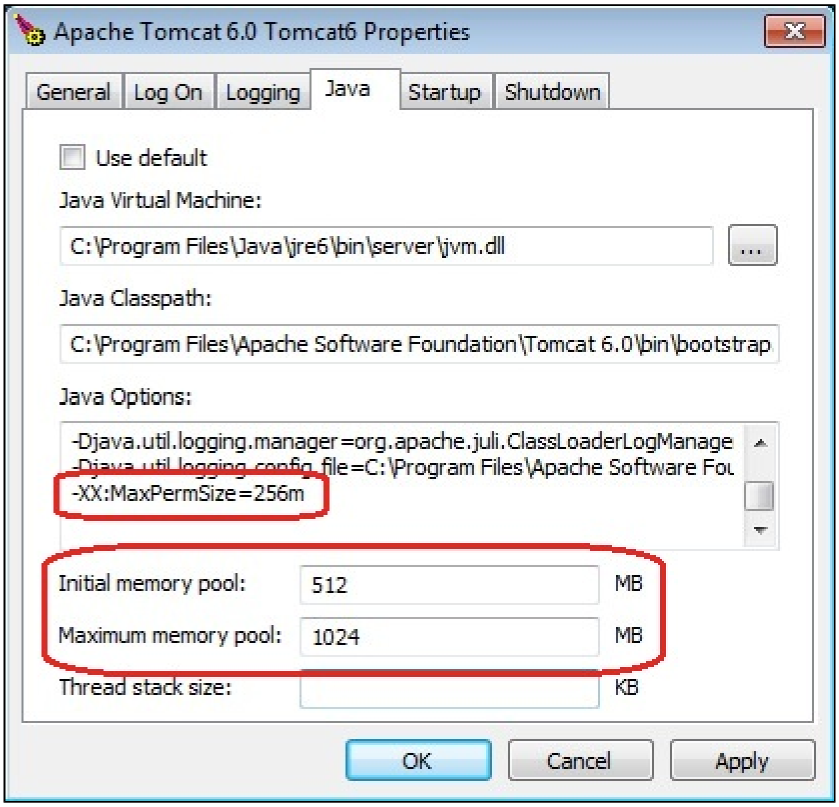
Select an appropriate method of updating the JVM variables:

All platforms
Add this value to `$TOMCAT_HOME/bin/catalina.bat` or `$TOMCAT_HOME/bin/catalina.sh` as the first line after the comment box.

All platforms
Add this parameter to the environment variables.

Windows
If you installed Tomcat as a Windows service, you can set values using the `$TOMCAT_HOME\bin\tomcatw.exe` GUI tool. Enter the parameters as follows:

- Enter the MaxPermSize parameter as a Java Option
- Initial memory pool = minimum heap size (Xms)
- Maximum memory pool = Maximum heap size (Xmx)



□ Use default

Java Virtual Machine:
C:\Program Files\Java\jre6\bin\server\jvm.dll

Java Classpath:
C:\Program Files\Apache Software Foundation\Tomcat 6.0\bin\bootstrap

Java Options:

```
-Djava.util.logging.manager=org.apache.juli.ClassLoaderLogManage
-Djava.util.logging.config.file=C:\Program Files\Apache Software Fou
-XX:MaxPermSize=256m
```

Initial memory pool: 512 MB

Maximum memory pool: 1024 MB

Thread stack size: KB

Note
Later, after you start Tomcat and log in to the Controller, you can validate these settings by running the `memory_usage.js` script, as follows:

1. From the navigation pane, select **Configuration > Maintenance Scripts**.
2. Run the `memory_usage.js` script. The min and max numbers on the top line (Heap) should be similar to the above settings.

Start and Validate Apache Tomcat

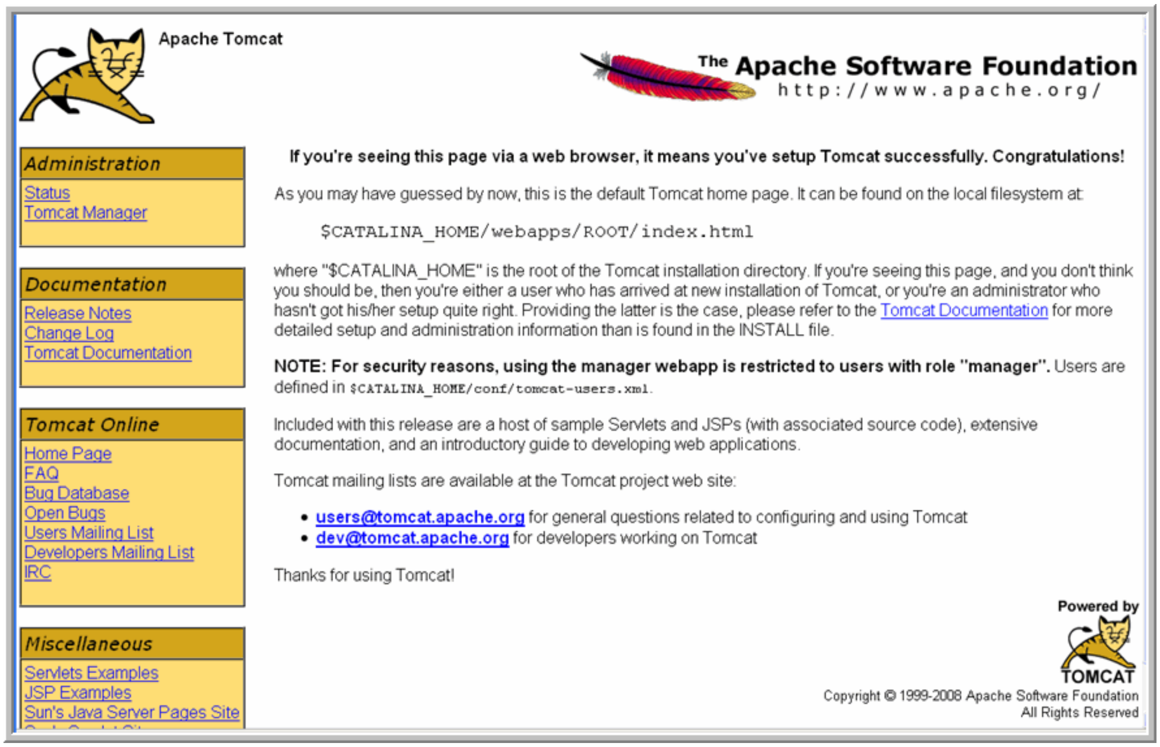
Perform the following steps to start and validate Apache Tomcat:

<table>
<tr><td><b>Step 1</b></td><td>Tomcat is normally run as a system service or daemon. You can start Tomcat using the standard method for your operating system or by using a script, as follows:

<b>Windows</b>
Use Windows Services to start Tomcat or start Tomcat from the command line as follows: <code>net start <name of Tomcat service&gt;</code>.

<b>Linux</b>
Start the Tomcat daemon using the script placed in the <code>/etc/init.d</code> directory for Tomcat: <code>service <name of Tomcat service&gt; start</code>.

<b>Windows or Linux</b>
Start the service using the <code>$TOMCAT_HOME/bin/startup.bat</code> or <code>$TOMCAT_HOME/bin/startup.sh</code> scripts.</td></tr>
<tr><td><b>Step 2</b></td><td>Open a browser and go to the following URL: http://localhost:8080.</td></tr>
<tr><td><b>Step 1</b></td><td>The following screen displays, verifying that you have successfully installed and started Tomcat:



Apache Tomcat
The Apache Software Foundation
http://www.apache.org/
Administration
Status
Tomcat Manager
Documentation
Release Notes
Change Log
Tomcat Documentation
Tomcat Online
Home Page
FAQ
Bug Database
Open Bugs
Users Mailing List
Developers Mailing List
IRC
Miscellaneous
Servlets Examples
JSP Examples
Sun's Java Server Pages Site
Powered by
TOMCAT

<b>If you're seeing this page via a web browser, it means you've setup Tomcat successfully. Congratulations!</b>

As you may have guessed by now, this is the default Tomcat home page. It can be found on the local filesystem at:

<code>$CATALINA_HOME/webapps/ROOT/index.html</code>

where "$CATALINA_HOME" is the root of the Tomcat installation directory. If you're seeing this page, and you don't think you should be, then you're either a user who has arrived at new installation of Tomcat, or you're an administrator who hasn't got his/her setup quite right. Providing the latter is the case, please refer to the Tomcat Documentation for more detailed setup and administration information than is found in the INSTALL file.

<b>NOTE: For security reasons, using the manager webapp is restricted to users with role "manager".</b> Users are defined in <code>$CATALINA_HOME/conf/tomcat-users.xml</code>.

Included with this release are a host of sample Servlets and JSPs (with associated source code), extensive documentation, and an introductory guide to developing web applications.

Tomcat mailing lists are available at the Tomcat project web site:

• <b>users@tomcat.apache.org</b> for general questions related to configuring and using Tomcat
• <b>dev@tomcat.apache.org</b> for developers working on Tomcat

Thanks for using Tomcat!

Copyright © 1999-2008 Apache Software Foundation
All Rights Reserved</td></tr>
</table>

Installing a Database

- Overview
- Database Management Systems
  - MySQL
  - Microsoft SQL Server
  - Oracle

Overview

Opswise Controller can use a database space of an existing database or you can install a database specifically for the Controller.

We recommend an initial size of 100MB.

Note
In a High Availability environment, each cluster node connects to the same database.

Database Management Systems

The following database management systems are supported:

- MySQL
- Microsoft SQL Server
- Oracle

MySQL

Note
MySQL versions 5.1.x and 5.5.x are supported.

Step 1	Download MySQL installation instructions.
Step 2	Download MySQL (Windows only). • For Windows, select Windows (x86, 32-bit), MSI Installer • For Unix and Linux, you can use a tar.gz download or select a systems package installer appropriate for your environment, such as Yum.
Step 3	Install MySQL as per the instructions.
Step 4	Make a note of the userid and password to be used later when installing the Controller.
Step 5	The database will be created automatically when you select MySQL during the Controller installation process.

MySQL Options

The following enhancements can be made to your MySQL database.

Speeding Up MySQL Performance

For Windows installations, you can speed up MySQL performance by adding the following parameter to the appropriate `MySQL.ini` file:

```
innodb_flush_log_at_trx_commit=0
```

For more information about this parameter, see the MySQL documentation:

http://dev.mysql.com/doc/refman/5.1/en/innodb-parameters.html#sysvar_innodb_flush_log_at_trx_commit

Setting the MySQL max_allowed_packet Configuration Variable

A communication packet is a single SQL statement sent to the MySQL server, a single row that is sent to the client, or a binary log event sent from a master replication server to a slave.

If you want the Controller to handle big packets, you must increase the MySQL **max_allowed_packet** configuration variable on the database server.

For detailed information about this variable, refer to the MySQL reference manual.

Microsoft SQL Server

> ⚠ **Note**
> Microsoft SQL Server versions 2005, 2008, and 2012 are supported.
>
> (If you are using Microsoft SQL Server 2005, when you create the database, set it to be CASE INSENSITIVE.)

Step 1	Download and install MS SQLServer as per the Microsoft documentation.
Step 2	Create the Controller database. You can use any legal name, but we recommend the name **opswise**.
Step 3	Make a note of the userid and password to be used later when installing the Controller.

The Unicode translation property can be changed to specify that prepared parameters for character data are sent as ASCII or Multi-byte Character Set (MBCS) instead of Unicode:

`sendStringParametersAsUnicode=false`

(The default value is **`true`**.)

Oracle

> ⚠ **Note**
> Oracle versions 10g and 11g are supported.

Step 1	Download and install Oracle as per the Oracle documentation.
Step 2	Create the Controller database. You can use any legal name, but we recommend the name **opswise**.
Step 3	Make a note of the userid and password to be used later when installing the Controller.

Oracle Options

The following enhancement can be made to your Oracle database.

Setting `open_cursors` Value for Large Imports

To facilitate large imports on Oracle, specify the maximum number of cursors that can be open by setting the **`open_cursors`** value to 1000.

(The cursors are used only during the import; they then are closed.)

Checking the Current Value of `open_cursors`

To check the current value for maximum open cursors, issue the following **sql*plus** utility command:

```
show parameter open_cursors
```

A listing similar to the following will display:

```
SQL> show parameter open_cursors;

NAME                                 TYPE        VALUE
------------------------------------ ----------- ------------------------------
open_cursors                         integer     1000
```

Setting a New Value for ***`open_cursors`***

You can temporarily set the **`open_cursors`** value with the following SQL:

```
alter system set open_cursors=1000
```

To make a permanent change, you must set the **`open_cursors`** value in the initialization parameters file.

Note
If you do not set **`open_cursors`** to 1000, you could receive the following error message during large imports:

```
ORA-01000: maximum open cursors exceeded
```

Downloading Opswise Controller Software

- Overview
  - Versioning
- Downloading Current Products Software
- Downloading Maintenance Software

Overview

This page tells you how to download the following from the Stonebranch website:

- Software for the current Opswise Controller 5.2.0 software.
- Latest version of Maintenance software, if any exists, that is appropriate for the current Opswise Controller 5.2.0 software on your platform.

Versioning

Opswise Automation Center software (Opswise Controller and Opswise Universal Agent) packages are labeled with four numeric identifiers: Version.Release.Modification.Maintenance.

For example, Opswise Controller 5.2.0.0:

- 5 = Version 5
- 2 = Release 2
- 0 = Modification Level 0
- 0 = Maintenance Level 0

Downloading Current Products Software

To download the Opswise Controller 5.2.0 software:

Step 1	Log in to the Stonebranch website. If you do not have a login, you can register for access on the login page.
Step 2	Access the Current Product Downloads page either of two ways: • Select **Support > Current Products**. • Select **Resources > Customer Support > Current Products**.
Step 3	Locate the Opswise Controller package Download File appropriate for your platform.
Step 4	Click the name of the file.
Step 5	In response to the prompt, click **Save File** and browse to your save location.

You can then install the software.

Downloading Maintenance Software

To download maintenance software:

Step 1	Log in to the Stonebranch website.
Step 2	Access the Maintenance page either of two ways: • Select **Support > Maintenance**. • Select **Resources > Customer Support > Maintenance**.
Step 3	Locate the Opswise Controller maintenance package Download File that is appropriate for your Opswise Controller software release.
Step 4	Click the name of the file.
Step 5	In response to the prompt, click **Save File** and browse to your save location.

Step 6	Extract the package and follow the instructions provided in the README file.

Installing Opswise Controller

- Overview
- Unpack the Opswise Controller Distribution File
- Install the Controller
  - Command Line Switches
  - Examples
- Deploy the Controller
- Verify the Installation
- Apply the License Key
  - License Information
- Configure System Notifications
  - System Notifications for License Violations and Expirations
  - System Notification for System Operations

Overview

This page tells you how to install Opswise Controller. It describes the following procedure:

1	Unpack the Downloaded Distribution File
2	Install the Controller
3	Deploy the Controller
4	Verify the Installation
5	Apply the License Key
6	Configure System Notifications

This procedure assumes you already have completed the following:

1	Installed prerequisite software.
2	Downloaded an Opswise Controller distribution file.

Unpack the Opswise Controller Distribution File

To unpack the Opswise Controller distribution file, use the following method appropriate for your platform:

Linux/Unix	`tar xvf`
	`opswise-controller-N.N.N.N.tar`

Windows	Use an appropriate archiving / unzipping product.

Install the Controller

To install the Controller, issue the following command that is appropriate for your platform:

Linux	`> sh install-controller.sh`
Windows	`> install-controller.bat`

You must include command line switches that specify information the Controller needs to access the Tomcat installation directory, the war file, and the database. You can include additional command line switches, but they are not required.

If a required command line switch is missing from the command line, an error message will identify it during the installation process.

The Controller installation process writes the values for some command line switches to the Opswise Controller start-up properties file, **`glide.properties`** (see the table, below). For any of those command line switches that are not required and, in fact, are not included on the command line, the Controller installation process writes their default value to **`glide.properties`**.

Command Line Switches

The following table describes the command line switches for the Controller installation process and identifies which are required.

For command line switches that have their value written to the Opswise Controller start-up properties file, **`glide.properties`**, the table also identifies the property in that file to which the value is written.

Note
All command line switches are case-sensitive.

<table>
<tr><th>Command Line Switch</th><th>Description</th><th>Default</th></tr>
<tr><td><code>--controller-file</code></td><td>Full path of the Opswise Controller war file from the downloaded Opswise Controller package.</td><td>none</td></tr>
<tr><td><code>--dbname</code></td><td>Opswise Controller database name.</td><td>opswise</td></tr>
<tr><td><code>--dbpass</code></td><td>Database user's password.</td><td>none</td></tr>
<tr><td><code>--dburl</code></td><td>JDBC connect URL.
Format: <b><code>jdbc:[database type]://localhost</code></b>
Examples (for MS SQLServer and Oracle, <b><code>opswise</code></b> is the database name):
<table>
<tr><th>MySQL</th><td><b><code>jdbc:mysql://localhost/</code></b></td></tr>
<tr><th>MS SQL Server</th><td><b><code>jdbc:sqlserver://localhost:1433;DatabaseName=opswise</code></b></td></tr>
<tr><th>Oracle</th><td><b><code>jdbc:oracle:thin:@//localhost:1521/opswise</code></b></td></tr>
</table></td><td>jdbc:mysql://localhost</td></tr>
<tr><td><code>--dbuser</code></td><td>Database user name.</td><td>none</td></tr>
<tr><td><code>--hubhost</code></td><td>Hostname or IP address of the Message Hub.
<b>Note</b>
This command line switch is needed only for backward-compatibility with pre-5.2.0 Agents.</td><td>none</td></tr>
</table>

`--hubport`	Port to use to communicate with the Message Hub. ⚠ **Note** This command line switch is needed only for backward-compatibility with pre-5.2.0 Agents.	none
`--rdbms`	Database type. Valid values are: • mysql • sqlserver • oracle * --rdbms ***is*** required if --dburl is used in the command.	mysql
`--tomcat-dir`	Path to the Tomcat installation directory (contains the directories:`/bin`, `/conf`, `/logs`, `webapps`).	none

Examples

Shown below are sample commands for installing the Controller on Linux and Windows platforms, using defaults for the database:

Linux	`sh install-controller.sh --tomcat-dir ~/tomcat --controller-file`
	`./opswise-controller-N.N.N.N-build.N.war --dbuser root --dbpass userpass`

Windows	`install-controller.bat --tomcat-dir "c:\Program Files\Apache Software Foundation\Tomcat`
	`7.0"`
	`--controller-file opswise-controller-N.N.N.N-build.N.war --dbuser root --dbpass userpass`

Deploy the Controller

In this procedure, you will start Tomcat, which starts the Controller and builds your database tables. This process takes several minutes. When it is complete, the Controller is started and ready to use.

If Tomcat already was running when you installed the Controller, you do not need to stop and restart it; this process will occur automatically after you start the installation.

Step 1 Start Tomcat as follows:

Linux
Start the Tomcat daemon using the script placed in the **`/etc/init.d`** directory for Tomcat.

```
service [name of Tomcat service] start
```

Windows
We recommend you use Windows Services to start Tomcat. Or, you can start Tomcat from the command line as follows:

```
net start [name of Tomcat service]
```

Linux or Windows
You can start the service using the **`$TOMCAT_HOME/bin/startup.bat`** or **`$TOMCAT_HOME/bin/startup.sh`** scripts.

Step 2

During this initial startup, the Controller builds the database tables, a process that takes several minutes. You can view details in the Tomcat window or monitor the Controller log, as described below:

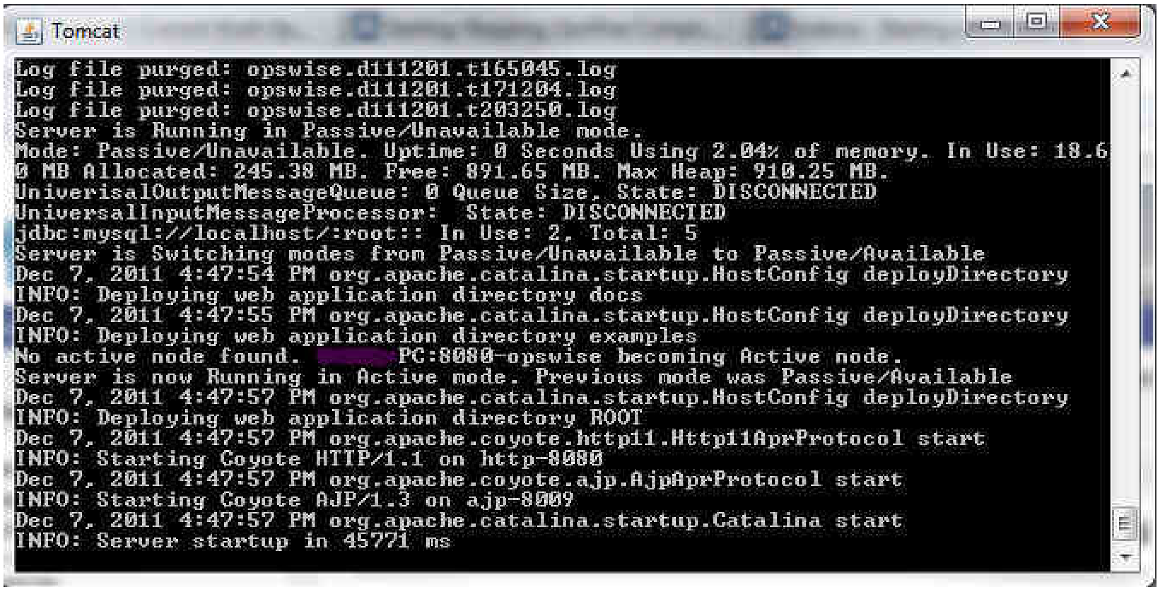
Linux/Unix
Users can tail the `opswise.log` to monitor the deployment process, as follows:

```
tail -f $TOMCAT_DIR/opswise_logs/opswise.log
```

Windows
Users can use a third-party tailing utility or open the log file using Notepad or other editor and scroll to the bottom to view the latest activity.

```
$TOMCAT_DIR/opswise_logs/opswise.log
```

Do not continue until you see output in the log similar to the following:



```
Log file purged: opswise.d111201.t165045.log
Log file purged: opswise.d111201.t171204.log
Log file purged: opswise.d111201.t203250.log
Server is Running in Passive/Unavailable mode.
Mode: Passive/Unavailable. Uptime: 0 Seconds Using 2.04% of memory. In Use: 18.6
0 MB Allocated: 245.38 MB. Free: 891.65 MB. Max Heap: 910.25 MB.
UniverisalOutputMessageQueue: 0 Queue Size. State: DISCONNECTED
UniversalInputMessageProcessor:  State: DISCONNECTED
jdbc:mysql://localhost/:root:: In Use: 2, Total: 5
Server is Switching modes from Passive/Unavailable to Passive/Available
Dec 7, 2011 4:47:54 PM org.apache.catalina.startup.HostConfig deployDirectory
INFO: Deploying web application directory docs
Dec 7, 2011 4:47:55 PM org.apache.catalina.startup.HostConfig deployDirectory
INFO: Deploying web application directory examples
No active node found. [redacted]PC:8080-opswise becoming Active node.
Server is now Running in Active mode. Previous mode was Passive/Available
Dec 7, 2011 4:47:57 PM org.apache.catalina.startup.HostConfig deployDirectory
INFO: Deploying web application directory ROOT
Dec 7, 2011 4:47:57 PM org.apache.coyote.http11.Http11AprProtocol start
INFO: Starting Coyote HTTP/1.1 on http-8080
Dec 7, 2011 4:47:57 PM org.apache.coyote.ajp.AjpAprProtocol start
INFO: Starting Coyote AJP/1.3 on ajp-8009
Dec 7, 2011 4:47:57 PM org.apache.catalina.startup.Catalina start
INFO: Server startup in 45771 ms
```

Step 3

When you see the following, the Controller is ready:

- **INFO [Ops.Cluster.Monitor.0] Server is now Running in Active mode. Previous mode was Passive/Unavailable**
- **INFO [Ops.Cluster.Monitor.0] Setting server to ACTIVE.**

Step 4

AIX and z/Linux only: Follow this procedure to change two default values in the Opswise Controller start-up properties file, **`glide.properties`**, which is read by the Controller.

(The **`glide.properties`** file resides in **`<tomcat directory>/webapps/opswise/WEB-INF/properties`**).

1. Change the following two properties from their default value to the AIX - z/Linux value:
    - **`opswise.trustmanager.algorithm=`** (Java trust manager algorithm)
        - Default value = SunX509
        - AIX - z/Linux value = IbmX509
    - **`opswise.trustmanager.provider=`** (Java trust manager provider)
        - Default value = SunJSSE
        - AIX - z/Linux value = IBMJSSE2
2. Restart Tomcat.

You now have completed the install process and the Controller is running.

Verify the Installation

To make sure the Controller is installed, running, and communication with Opswise Universal Agent and Opswise Message Service (OMS):

Step 1

From your browser, access the Opswise Controller user interface.

```
http://localhost:8080/opswise
```

`localhost` represents the machine name where you installed the server.

Step 2

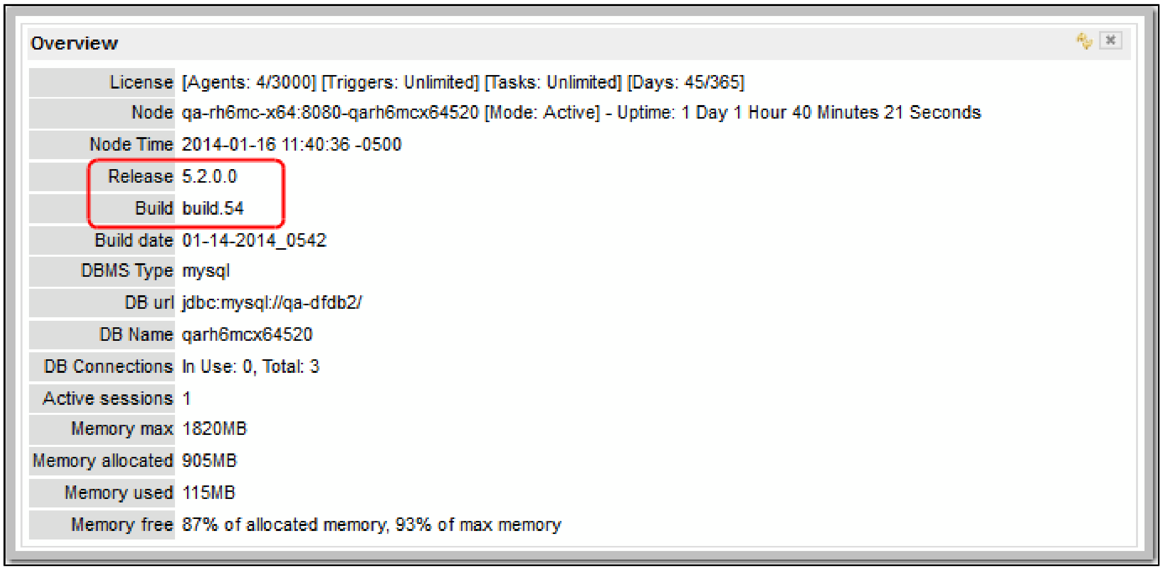
Log in with user **`ops.admin`** and no password.

Step 3

Create a password when prompted. The Opswise Automation Center home page displays.

Step 4

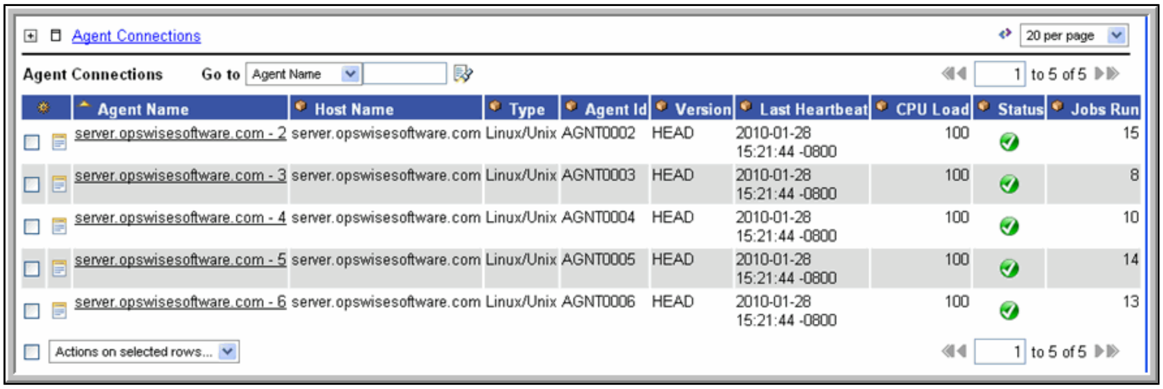
The home page includes a gadget called **Overview**, which provides current system information. Check the Release information to verify that the latest version number is displayed, as shown in the following example.



Overview

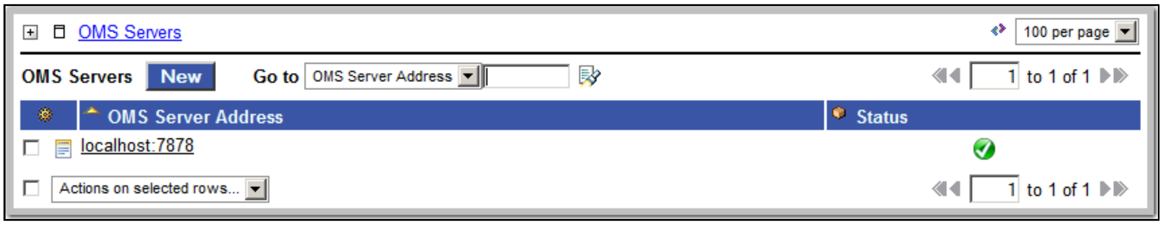
License	[Agents: 4/3000] [Triggers: Unlimited] [Tasks: Unlimited] [Days: 45/365]
Node	qa-rh6mc-x64:8080-qarh6mcx64520 [Mode: Active] - Uptime: 1 Day 1 Hour 40 Minutes 21 Seconds
Node Time	2014-01-16 11:40:36 -0500
Release	5.2.0.0
Build	build.54
Build date	01-14-2014_0542
DBMS Type	mysql
DB url	jdbc:mysql://qa-dfdb2/
DB Name	qarh6mcx64520
DB Connections	In Use: 0, Total: 3
Active sessions	1
Memory max	1820MB
Memory allocated	905MB
Memory used	115MB
Memory free	87% of allocated memory, 93% of max memory

Step 5 Select **Automation Center Resources > All Agents**. You will see a list similar to the following example. Make sure the Agent Status is Active (green circle with a check mark).



Agent Name	Host Name	Type	Agent Id	Version	Last Heartbeat	CPU Load	Status	Jobs Run
server.opswisesoftware.com - 2	server.opswisesoftware.com	Linux/Unix	AGNT0002	HEAD	2010-01-28 15:21:44 -0800	100	✓	15
server.opswisesoftware.com - 3	server.opswisesoftware.com	Linux/Unix	AGNT0003	HEAD	2010-01-28 15:21:44 -0800	100	✓	8
server.opswisesoftware.com - 4	server.opswisesoftware.com	Linux/Unix	AGNT0004	HEAD	2010-01-28 15:21:44 -0800	100	✓	10
server.opswisesoftware.com - 5	server.opswisesoftware.com	Linux/Unix	AGNT0005	HEAD	2010-01-28 15:21:44 -0800	100	✓	14
server.opswisesoftware.com - 6	server.opswisesoftware.com	Linux/Unix	AGNT0006	HEAD	2010-01-28 15:21:44 -0800	100	✓	13

Step 6 Select **Automation Center Resources > OMS Servers**. You will see a list similar to the following example. Make sure the OMS Statuses are Active (green circle with a check mark).



OMS Server Address	Status
localhost:7878	✓

Step 7 For more information about these components in the Opswise Controller user interface, see:

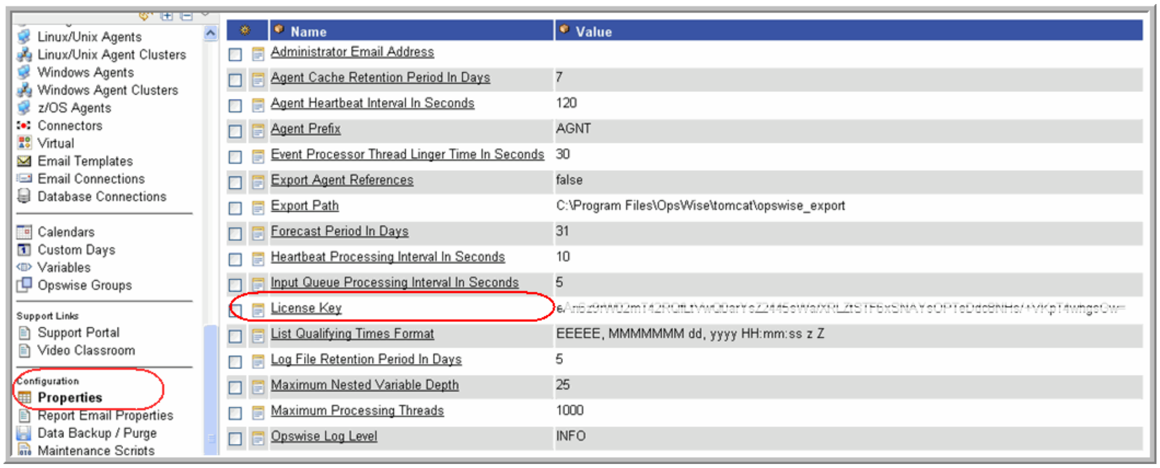
- Agents
- Opswise Message Service (OMS)

To get started using the Controller and become familiar with its features, we recommend you spend some time going through the Tutorials.

Apply the License Key

Although you do not normally need to enter a license key right away, at some point you will need to follow these steps to enter your key:

Step 1 From the navigation pane, select **Configuration > Properties > License Key**.



Name	Value
Administrator Email Address	
Agent Cache Retention Period In Days	7
Agent Heartbeat Interval In Seconds	120
Agent Prefix	AGNT
Event Processor Thread Linger Time In Seconds	30
Export Agent References	false
Export Path	C:\Program Files\OpsWise\tomcat\opswise_export
Forecast Period In Days	31
Heartbeat Processing Interval In Seconds	10
Input Queue Processing Interval In Seconds	5
License Key	[illegible]
List Qualifying Times Format	EEEEE, MMMMMMM dd, yyyy HH:mm:ss z Z
Log File Retention Period In Days	5
Maximum Nested Variable Depth	25
Maximum Processing Threads	1000
Opswise Log Level	INFO

Step 2 Enter your encrypted license key in the Value field and click **Update**.

Step 3 Return to the Automation Center home page (click the Home 🏠 icon) and review the License field on the Overview gadget to verify that the terms of your license are correct.

Step 4	Optionally, configure the Controller so that your system administrator receives notifications regarding license violations and expirations .

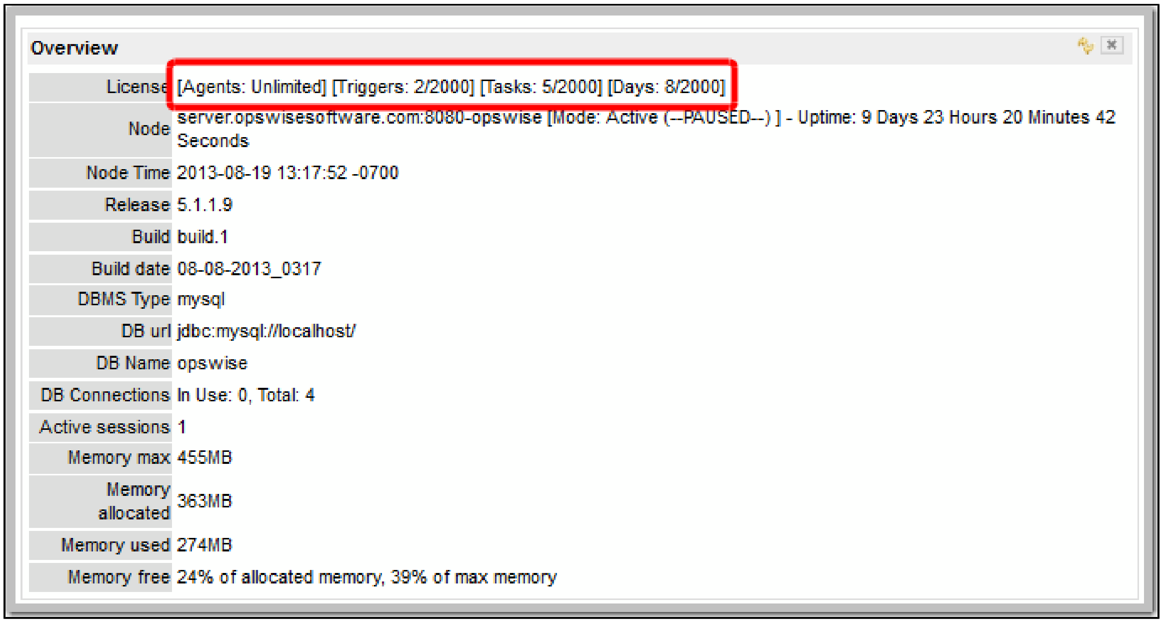
License Information

The License field in the Overview gadget on the Opswise Automation Center home page identifies license information for:

- Agents
- Triggers
- Tasks
- Days

The value for each field is either:

- Unlimited (unlimited number to the license)
- N/N (number remaining in license / total number in license)



Overview

License	[Agents: Unlimited] [Triggers: 2/2000] [Tasks: 5/2000] [Days: 8/2000]
Node	server.opswisesoftware.com:8080-opswise [Mode: Active (--PAUSED--)] - Uptime: 9 Days 23 Hours 20 Minutes 42 Seconds
Node Time	2013-08-19 13:17:52 -0700
Release	5.1.1.9
Build	build.1
Build date	08-08-2013_0317
DBMS Type	mysql
DB url	jdbc:mysql://localhost/
DB Name	opswise
DB Connections	In Use: 0, Total: 4
Active sessions	1
Memory max	455MB
Memory allocated	363MB
Memory used	274MB
Memory free	24% of allocated memory, 39% of max memory

Configure System Notifications

System Notifications are emails sent to one or more Opswise Controller system administrators based on either:

- Licensing issues (license violations, expired licenses, invalid licenses)
- Status of a system operation associated with a task instance.

> **Note**
> System Notifications are not the same as Email Notifications. Please refer to the following sections for explicitly defining Email Notifications.
>
> - Email Notifications for Agents
> - Email Notifications for OMS Servers
> - Email Notifications for Connectors
> - Email Notifications for Cluster Nodes
> - Email Notifications for Task Instance Events

In order for a system administrator to receive system notifications, you must configure the Controller for system notifications:

Step 1	Select an email connection on which the notifications will be sent and enable the Use for System Notifications field. ⚠ **Note** You can use only one Email Connection at any one time for sending system notifications.
Step 2	Identify the Opswise Controller Administrator(s) that will receive the system notifications by entering one or more valid email addresses for those administrators in the Administrator Email Address Opswise Controller system property.

System Notifications for License Violations and Expirations

When you have configured the Controller for system notification, notifications automatically are sent to the specified system administrator(s) for the following license issues:

- License violations
- Expired licenses
- Invalid licenses

License Violations

A system notification is sent for the following license violations:

- User attempts to create a task that exceeds the licensed maximum number of task definitions.
- User attempts to enable a trigger that exceeds the licensed maximum number of enabled triggers.
- Agent registration attempt exceeds the licensed maximum number of Agents.

The License field in the Overview gadget on the Opswise Automation Center home page identifies these maximum numbers (see License Information, above).

License Expiration

A system notification is sent at the following times if a license will expire in 7 days or sooner:

- Warning sent daily at midnight, processed same time as midnight log rollover, starting 7 days prior to license expiration.
- Warning sent on Controller start-up (or a cluster node becoming the Active cluster node) if license is within 7 days of expiring.
- Warning sent on License Key property change (if new license is still within 7 days of expiring).

A system notification is sent at the following times if a license has expired:

- Sent daily at midnight, processed same time as midnight log rollover.
- Sent on Controller start-up (or a cluster node becoming the Active cluster node).
- Send on License Key property change (if new license still expired).
- System paused on license expiration.

Invalid Licenses

A system notification is sent at the following times if a license is invalid:

- Sent on Controller start-up (or a cluster node becoming the Active cluster node).
- System paused on invalid license.

An invalid license will display in the Overview as [Agents: x/0] [Triggers: y/0] [Tasks: z/0] [Days: 1/0] where x, y, and z are the current number of agents, triggers, and tasks, respectively.

System Notification for System Operations

For any Controller task, you can select a system operation to be performed when any instance of that task reaches one or more specific statuses. You also can select whether or not to send system notifications based on the success and/or failure of that system operation.

For detailed information on how to set up these system notifications, see System Operation Actions.

Installing Opswise Bundled Controller on AIX

- Introduction
- Uncompress the Opswise Bundled Controller Distribution File
- Install the Bundled Controller
    - Silent Install
    - Interactive Install
    - Command Line Switches / Configuration File Options
    - Example
- Verify the Installation
- Apply the License Key
    - License Information
- Configure System Notifications
    - System Notifications for License Violations and Expirations
    - System Notification for System Operations

Introduction

This page tells you how to install Opswise Bundled Controller, which is the Opswise Controller bundled with Apache Tomcat.

> ⚠ **Note**
> Currently, the Bundled Controller is available only for the AIX operating system.

It describes the following procedure:

1	Uncompress the Downloaded Distribution File
2	Install the Bundled Controller
3	Verify the Installation
4	Apply the License Key
5	Configure System Notifications

The procedure assumes you already have completed the following:

1	Performed pre-installation procedures
2	Downloaded the Opswise Bundled Controller distribution file

Uncompress the Opswise Bundled Controller Distribution File

To uncompress the Opswise Bundled Controller distribution file:

```
tar xvfo opswise-controller-bundle-N.N.N.N.tar
```

Among the files contained in the tar file, please note the following:

File Name	Description
`install.sh`	Installation script
`config`	Configuration file with default values

Install the Bundled Controller

You can install the Bundled Controller silently or interactively.

Silent Install

A silent install of the Bundled Controller uses the default values contained in the Bundled Controller configuration file, **`config`**:

```
./install.sh -s
```

Interactive Install

An interactive install of the Bundled Controller prompts you for values for all command line switches:

```
./install.sh <options>
```

Command Line Switches / Configuration File Options

The following table describes the command line switches / configuration file options for the Bundled Controller installation process.

The installation process writes some of the command line switch / configuration file option values to the Opswise Controller start-up properties, **`glide.properties`**. The table identifies the properties in that file to which values are written.

Note
All command line switches are case-sensitive.

Command Line Switch	Configuration File Option	Description	Default	Controller Property
`--dbhost`	DBHOST	Database host name	localhost	n/a
`--dbname`	DBNAME	Database name	opswise	glide.db.name=
`--dbpass`	DBPASS	Database user password	(none)	glide.db.password=
`--dbport`	DBPORT	Database port number	3306	n/a
`--dbuser`	DBUSER	Database user name	opswise	glide.db.user=
`--group`	OPSWISE_GROUP	User group to be used for the Controller	opscntlr	n/a
`-h`	n/a	Usage screen	(none)	n/a

`--http-port`	PORT	HTTP server port number	8080	n/a
`--hubhost`	HUBHOST	Message Hub host name **Note** This command line switch is needed only for backward-compatibility with pre-5.2.0 Agents.	none	
`--hubport`	HUBPORT	Message Hub port number **Note** This command line switch is needed only for backward-compatibility with pre-5.2.0 Agents.	none	
`--install-dir`	INSTALL_DIR	Installation directory	/opt/opscntlr	n/a
`--java-home`	n/a	Path to the Java installation (JAVA_HOME)	(none)	n/a
`--rdbms`	RDBMS	Database type. Valid values are: • mysql • sqlserver • sqlserver-jtds • oracle	mysql	glide.db.rdbms=
`-s`	n/a	Silent (unattended) install. Default is interactive install.	(none)	n/a
`--shutdown-port`	SHUTDOWN_PORT	Server shutdown port	8005	n/a
`--user`	OPSWISE_USER	System account to be used for the Controller	opscntlr	n/a

Example

Shown below is a sample command for installing the Bundled Controller (default values are used for options not specified):

```
./install.sh -s --dbuser root --dbpass userpass
```

Verify the Installation

To make sure the Controller is installed, running, and communication with Opswise Universal Agent and Opswise Message Service (OMS):

Step 1 Start the Controller.

Step 2 From your browser, access the Opswise Controller user interface.

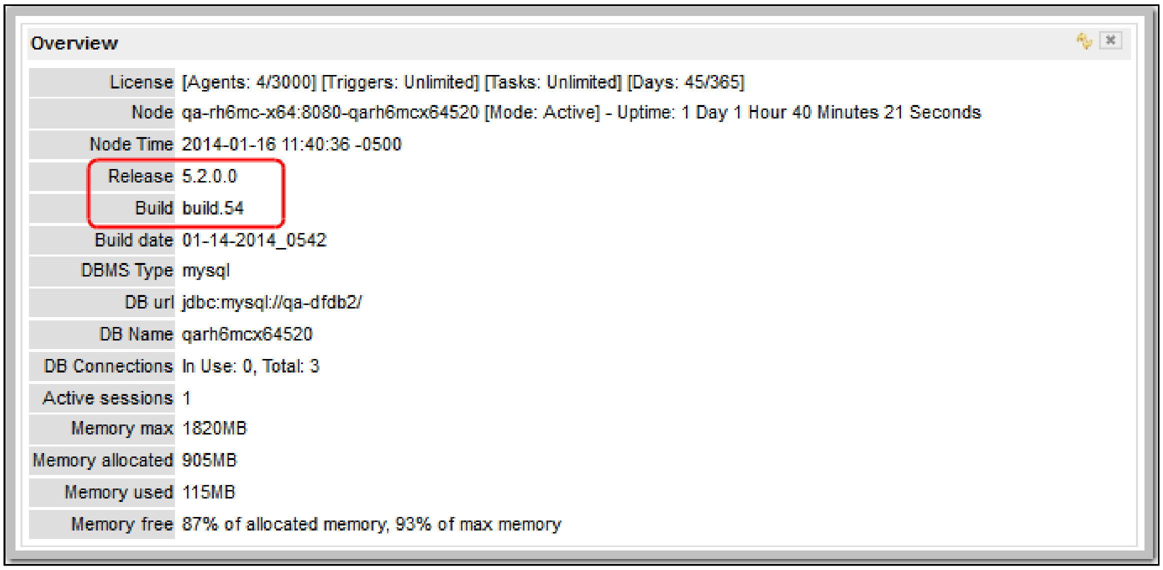
```
http://localhost:8080/opswise
```

`localhost` represents the machine name where you installed the server.

Step 3 Log in with user `ops.admin` and no password.

Step 4 Create a password when prompted. The Opswise Automation Center home page displays.

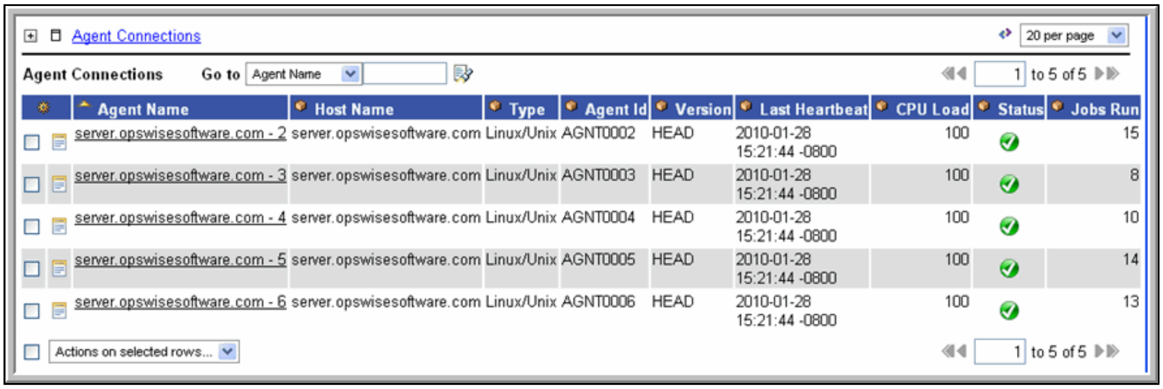
Step 5 The home page includes a gadget called Overview, which provides current system information. Check the Release information to verify that the latest version number is displayed, as shown in the following example.



Overview

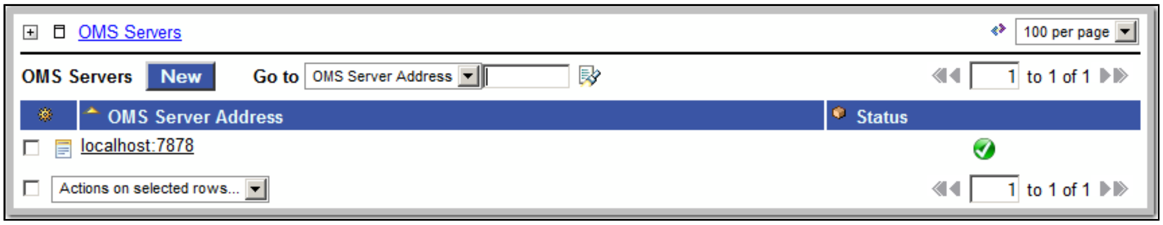
License	[Agents: 4/3000] [Triggers: Unlimited] [Tasks: Unlimited] [Days: 45/365]
Node	qa-rh6mc-x64:8080-qarh6mcx64520 [Mode: Active] - Uptime: 1 Day 1 Hour 40 Minutes 21 Seconds
Node Time	2014-01-16 11:40:36 -0500
Release	5.2.0.0
Build	build.54
Build date	01-14-2014_0542
DBMS Type	mysql
DB url	jdbc:mysql://qa-dfdb2/
DB Name	qarh6mcx64520
DB Connections	In Use: 0, Total: 3
Active sessions	1
Memory max	1820MB
Memory allocated	905MB
Memory used	115MB
Memory free	87% of allocated memory, 93% of max memory

Step 6 Select **Automation Center Resources > All Agents**. You will see a list similar to the following example. Make sure the Agent Status is Active (green circle with a check mark).



	Agent Name	Host Name	Type	Agent Id	Version	Last Heartbeat	CPU Load	Status	Jobs Run
□	server.opswisesoftware.com - 2	server.opswisesoftware.com	Linux/Unix	AGNT0002	HEAD	2010-01-28 15:21:44 -0800	100	✓	15
□	server.opswisesoftware.com - 3	server.opswisesoftware.com	Linux/Unix	AGNT0003	HEAD	2010-01-28 15:21:44 -0800	100	✓	8
□	server.opswisesoftware.com - 4	server.opswisesoftware.com	Linux/Unix	AGNT0004	HEAD	2010-01-28 15:21:44 -0800	100	✓	10
□	server.opswisesoftware.com - 5	server.opswisesoftware.com	Linux/Unix	AGNT0005	HEAD	2010-01-28 15:21:44 -0800	100	✓	14
□	server.opswisesoftware.com - 6	server.opswisesoftware.com	Linux/Unix	AGNT0006	HEAD	2010-01-28 15:21:44 -0800	100	✓	13

Step 7 Select **Automation Center Resources > OMS Servers**. You will see a list similar to the following example. Make sure the OMS Statuses are Active (green circle with a check mark).



OMS Servers — 100 per page

OMS Servers New Go to OMS Server Address — 1 to 1 of 1

	OMS Server Address	Status
□	localhost:7878	✓

□ Actions on selected rows... — 1 to 1 of 1

Step 8 For more information about these components in the Opswise Controller user interface, see:

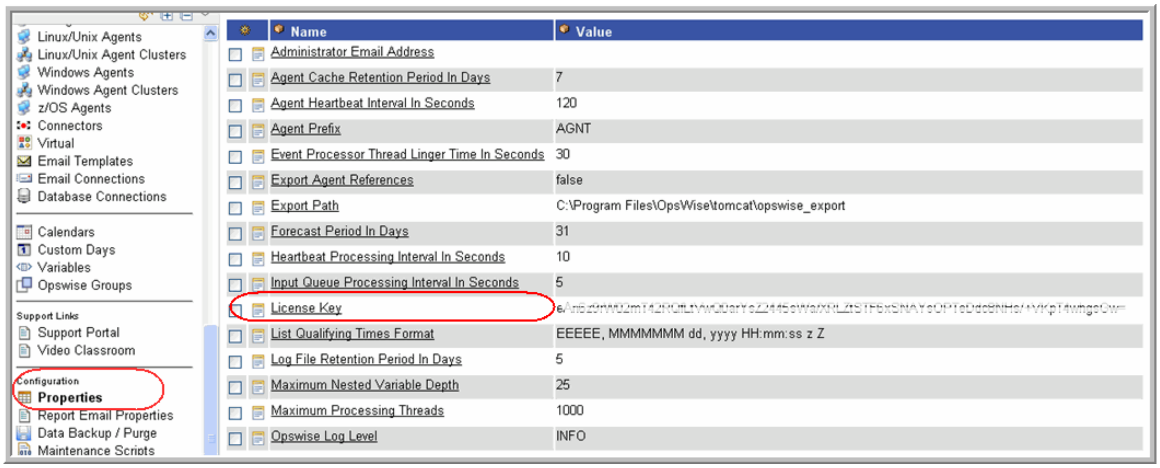
- Agents
- Opswise Message Service (OMS)

To get started using the Controller and become familiar with its features, we recommend you spend some time going through the Tutorials.

Apply the License Key

Although you do not normally need to enter a license key right away, at some point you will need to follow these steps to enter your key:

Step 1 From the navigation pane, select **Configuration > Properties > License Key**.



	Name	Value
□	Administrator Email Address	
□	Agent Cache Retention Period In Days	7
□	Agent Heartbeat Interval In Seconds	120
□	Agent Prefix	AGNT
□	Event Processor Thread Linger Time In Seconds	30
□	Export Agent References	false
□	Export Path	C:\Program Files\OpsWise\tomcat\opswise_export
□	Forecast Period In Days	31
□	Heartbeat Processing Interval In Seconds	10
□	Input Queue Processing Interval In Seconds	5
□	License Key	[redacted]
□	List Qualifying Times Format	EEEEE, MMMMMMM dd, yyyy HH:mm:ss z Z
□	Log File Retention Period In Days	5
□	Maximum Nested Variable Depth	25
□	Maximum Processing Threads	1000
□	Opswise Log Level	INFO

Step 2 Enter your encrypted license key in the Value field and click **Update**.

Step 3 Return to the Automation Center home page (click the Home icon) and review the License field on the Overview gadget to verify that the terms of your license are correct.

Step 4 Optionally, configure the Controller so that your system administrator receives notifications regarding license key violations and expirations.

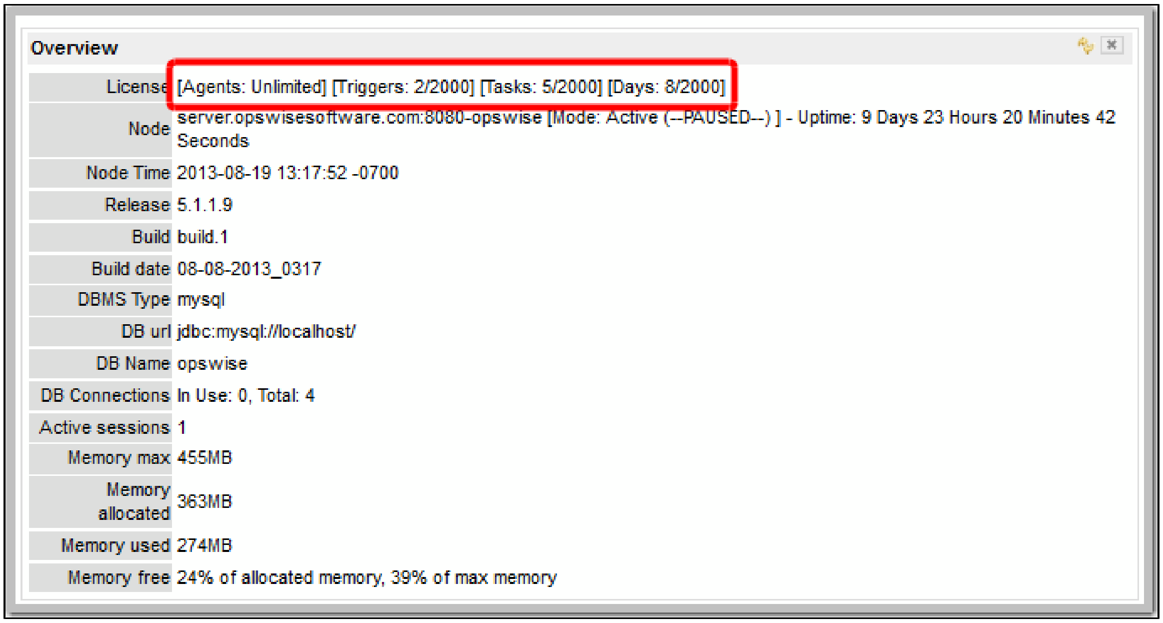
License Information

The License field in the Overview gadget on the Opswise Automation Center home page identifies license information for:

- Agents
- Triggers
- Tasks
- Days

The value for each field is either:

- Unlimited (unlimited number to the license)
- N/N (number remaining in license / total number in license)



Overview

License	[Agents: Unlimited] [Triggers: 2/2000] [Tasks: 5/2000] [Days: 8/2000]
Node	server.opswisesoftware.com:8080-opswise [Mode: Active (--PAUSED--)] - Uptime: 9 Days 23 Hours 20 Minutes 42 Seconds
Node Time	2013-08-19 13:17:52 -0700
Release	5.1.1.9
Build	build.1
Build date	08-08-2013_0317
DBMS Type	mysql
DB url	jdbc:mysql://localhost/
DB Name	opswise
DB Connections	In Use: 0, Total: 4
Active sessions	1
Memory max	455MB
Memory allocated	363MB
Memory used	274MB
Memory free	24% of allocated memory, 39% of max memory

Configure System Notifications

System Notifications are emails sent to one or more Opswise Controller system administrators based on either:

- Licensing issues (license violations, expired licenses, invalid licenses)
- Status of a system operation associated with a task instance.

Note
System Notifications are not the same as Email Notifications. Please refer to the following sections for explicitly defining Email Notifications.

- Email Notifications for Agents
- Email Notifications for OMS
- Email Notifications for Connectors
- Email Notifications for Cluster Nodes
- Email Notifications for Task Instance Events

In order for a system administrator to receive system notifications, you must configure the Controller for system notifications:

Step 1	Select an email connection on which the notifications will be sent and enable the Use for System Notifications field. **Note** You can use only one Email Connection at any one time for sending system notifications.
Step 2	Identify the Controller Administrator(s) that will receive the system notifications by entering one or more valid email addresses for those administrators in the Administrator Email Address Opswise Controller system property.

System Notifications for License Violations and Expirations

When you have configured the Controller for system notification, notifications automatically are sent to the specified system administrator(s) for the following license issues:

- License violations
- Expired licenses
- Invalid licenses

License Violations

A system notification is sent for the following license violations:

- User attempts to create a task that exceeds the licensed maximum number of task definitions.
- User attempts to enable a trigger that exceeds the licensed maximum number of enabled triggers.
- Agent registration attempt exceeds the licensed maximum number of Agents.

The License field in the Overview gadget on the Opswise Automation Center home page identifies these maximum numbers (see License Information, above).

License Expiration

A system notification is sent at the following times if a license will expire in 7 days or sooner:

- Warning sent daily at midnight, processed same time as midnight log rollover, starting 7 days prior to license expiration.
- Warning sent on Controller start-up (or a cluster node becoming the Active cluster node) if license is within 7 days of expiring.
- Warning sent on License Key property change (if new license is still within 7 days of expiring).

A system notification is sent at the following times if a license has expired:

- Sent daily at midnight, processed same time as midnight log rollover.
- Sent on Controller start-up (or a cluster node becoming the Active cluster node).
- Send on License Key property change (if new license still expired).
- System paused on license expiration.

Invalid Licenses

A system notification is sent at the following times if a license is invalid:

- Sent on Controller start-up (or a cluster node becoming the Active cluster node).
- System paused on invalid license.

An invalid license will display in the Overview as [Agents: x/0] [Triggers: y/0] [Tasks: z/0] [Days: 1/0] where x, y, and z are the current number of agents, triggers, and tasks, respectively.

System Notification for System Operations

For any Controller task, you can select a system operation to be performed when any instance of that task reaches one or more specific statuses. You also can select whether or not to send system notifications based on the success and/or failure of that system operation.

For detailed information on how to set up these system notifications, see System Operation Actions.

Adding a Cluster Node

- Overview
    - Requirements for Adding a Cluster Node
    - Procedure for Adding a Cluster Node
- Copy and Unpack the Opswise Controller Distribution File
- Install the Controller
    - Command Line Switches
    - Examples
- Deploy the Controller
- Verify the Installation
- Adding an OMS Server
    - Add OMS Server to OMS Server Record
    - OMS Server Message Database

Overview

When you install Opswise Controller, you create a single instance (cluster node) of the Controller. To operate Opswise Automation Center in a High Availability (HA) environment, you must add one or more cluster nodes. Each cluster node should be installed on a separate machine.

This page tells you how to add one or more cluster nodes.

Requirements for Adding a Cluster Node

Each cluster node in an HA environment must connect to the same Opswise Controller database. If one of the cluster nodes stops processing, another cluster node continues processing with the same data.

Each cluster node in an HA environment must be the same version and build of the Controller. To ensure this, you can either:

- Install the downloaded version of the Controller on a second machine.
- Download a new version of the Controller software, update the current version, and then install the new version on a second machine.

It is strongly recommended that an HA environment has at least two OMS Servers, although you do not need an OMS Server for every cluster node if your HA environment contains three or more cluster nodes.

Procedure for Adding a Cluster Node

This page describes the following procedure:

1	Copy and Unpack the Downloaded Distribution File
2	Install the Controller
3	Deploy the Controller
4	Verify the Installation
5	Adding an OMS Server

This procedure assumes you already have performed any required pre-installation procedure steps for the cluster node being added.

Copy and Unpack the Opswise Controller Distribution File

Copy the downloaded distribution file, which was used to install the current, single instance of Opswise Controller, from its current location to the machine on which you want to install a new instance of the Controller.

To unpack the Opswise Controller distribution file, use the following method appropriate for your platform:

Linux/Unix	`tar xvf opswise-controller-N.N.N.N.tar`
Windows	Use an appropriate archiving / unzipping product.

Install the Controller

To install the Controller, issue the following command that is appropriate for your platform:

Linux	`> sh install-controller.sh`
Windows	`> install-controller.bat`

You must include command line switches that specify information the Controller needs to access the Tomcat installation directory, the war file, and the database. You can include additional command line switches, but they are not required.

If a required command line switch is missing from the command line, an error message will identify it during the installation process.

The Controller installation process writes the values for some command line switches to the Opswise Controller start-up properties file, **`glide.properties`** (see the table, below). For any of those command line switches that are not required and, in fact, are not included on the command line, the Controller installation process writes their default value to **`glide.properties`**.

Command Line Switches

The following table describes the command line switches for the Controller installation process and identifies which are required.

For command line switches that have their value written to the Opswise Controller start-up properties file, **`glide.properties`**, the table also identifies the property in that file to which the value is written.

Note
All command line switches are case-sensitive.

Command Line Switch	Description	Default
`--controller-file`	Full path of the Opswise Controller war file from the downloaded Opswise Controller package.	none
`--dbname`	Opswise Controller database name.	opswise
`--dbpass`	Database user's password.	none

<table>
<tr>
<td><code>--dburl</code></td>
<td>JDBC connect URL.
Format: <b><code>jdbc:[database type]://localhost</code></b>
Examples (for MS SQLServer and Oracle, <b><code>opswise</code></b> is the database name):
<table>
<tr><td><b>MySQL</b></td><td><b><code>jdbc:mysql://localhost/</code></b></td></tr>
<tr><td><b>MS SQL Server</b></td><td><b><code>jdbc:sqlserver://localhost:1433;DatabaseName=opswise</code></b></td></tr>
<tr><td><b>Oracle</b></td><td><b><code>jdbc:oracle:thin:@//localhost:1521/opswise</code></b></td></tr>
</table>
</td>
<td>jdbc:mysql://localhost</td>
</tr>
<tr>
<td><code>--dbuser</code></td>
<td>Database user name.</td>
<td>none</td>
</tr>
<tr>
<td><code>--hubhost</code></td>
<td>Hostname or IP address of the Message Hub.
⚠ <b>Note</b>
This command line switch is needed only for backward-compatibility with pre-5.2.0 Agents.</td>
<td>none</td>
</tr>
<tr>
<td><code>--hubport</code></td>
<td>Port to use to communicate with the Message Hub.
⚠ <b>Note</b>
This command line switch is needed only for backward-compatibility with pre-5.2.0 Agents.</td>
<td>none</td>
</tr>
<tr>
<td><code>--rdbms</code></td>
<td>Database type.
Valid values are:
• mysql
• sqlserver
• oracle
<b>*</b> --rdbms <b><i>is</i></b> required if --dburl is used in the command.</td>
<td>mysql</td>
</tr>
<tr>
<td><code>--tomcat-dir</code></td>
<td>Path to the Tomcat installation directory (contains the directories:<code>/bin</code>, <code>/conf</code>, <code>/logs</code>, <code>webapps</code>).</td>
<td>none</td>
</tr>
</table>

Examples

Shown below are sample commands for installing the Controller on Linux and Windows platforms, using defaults for the database:

Linux

```
sh install-controller.sh --tomcat-dir ~/tomcat --controller-file
./opswise-controller-N.N.N.N-build.N.war --dbuser root --dbpass userpass
```

Windows

```
install-controller.bat --tomcat-dir "c:\Program Files\Apache Software Foundation\Tomcat
7.0"
--controller-file opswise-controller-N.N.N.N-build.N.war --dbuser root --dbpass userpass
```

Deploy the Controller

In this procedure, you will start Tomcat, which starts the Controller and builds your database tables. This process takes several minutes. When it is complete, the Controller is started and ready to use.

If Tomcat already was running when you installed the Controller, you do not need to stop and restart it; this process will occur automatically after you start the installation.

Step 1

Start Tomcat as follows:

Linux
Start the Tomcat daemon using the script placed in the **`/etc/init.d`** directory for Tomcat.

```
service [name of Tomcat service] start
```

Windows
We recommend you use Windows Services to start Tomcat. Or, you can start Tomcat from the command line as follows:

```
net start [name of Tomcat service]
```

Linux or Windows
You can start the service using the **`$TOMCAT_HOME/bin/startup.bat`** or **`$TOMCAT_HOME/bin/startup.sh`** scripts.

Step 2

You can view details of the start-up in the Tomcat window or monitor the Controller log, as described below:

Linux/Unix
Users can tail the `opswise.log` to monitor the deployment process, as follows:

```
tail -f $TOMCAT_DIR/opswise_logs/opswise.log
```

Windows
Users can use a third-party tailing utility or open the log file using Notepad or other editor and scroll to the bottom to view the latest activity.

```
$TOMCAT_DIR/opswise_logs/opswise.log
```

Step 3

When you see the following, the Controller is ready:

- **INFO [Ops.Cluster.Monitor.0] Server is now Running in Passive/Unavailable mode.**
- **INFO [Ops.Cluster.Monitor.0] Setting server to PASSIVE/UNAVAILABLE.**

Step 4	**AIX and z/Linux only**: Follow this procedure to change two default values in the Opswise Controller start-up properties file, **`glide.properties`**, which is read by the Controller. (The **`glide.properties`** file resides in **`<tomcat directory>/webapps/opswise/WEB-INF/properties`**). 1. Change the following two properties from their default value to the AIX - z/Linux value: • **`opswise.trustmanager.algorithm=`** (Java trust manager algorithm) • Default value = SunX509 • AIX - z/Linux value = IbmX509 • **`opswise.trustmanager.provider=`** (Java trust manager provider) • Default value = SunJSSE • AIX - z/Linux value = IBMJSSE2 2. Restart Tomcat.

You now have completed the install process and the Controller is running.

Verify the Installation

To make sure the new cluster node is installed and running properly:

Step 1	Log in to the originally installed Controller.
Step 2	On the home page, verify that the **Cluster Node status** widget illustrates an Active and a Passive/Inactive cluster node.
Step 3	For detailed information on the new (and original) cluster nodes, select **Automation Center Resources > Cluster Nodes**.

> **Note**
> The license key for the installed Opswise Controller applies to all instances (cluster nodes) of that Controller; no additional licensing is required.
>
> System Notifications configured for the installed Opswise Controller apply to all instances (cluster nodes) of that Controller; no additional system notifications have to be configured.

Adding an OMS Server

To add a second OMS Server to an HA environment (which creates an OMS cluster), you must install Opswise Universal Agent on a machine where one of the additional cluster nodes has been added.

Add OMS Server to OMS Server Record

You must specify all members of an OMS cluster in your HA environment in the same OMS Server record.

The OMS Servers list screen will contain a single entry for all OMS cluster members defined in the record. (The OMS Servers list screen could have additional entries for an OMS Server or OMS cluster outside of your HA environment. For example, OMS Servers outside a firewall would connect to a different message database and serve different Agents, but would connect to to the same Controller.)

OMS Server Message Database

Members of an OMS cluster in an HA environment must use the same OMS Server message database.

The OMS SPOOL_DIRECTORY configuration option specifies the name of the directory where the OMS maintains its message database. For each OMS Server, you must set this option to a location shared by all of the OMS Servers in the HA environment.

Opswise Controller Upgrade and Maintenance

Opswise Controller Upgrade and Maintenance

Upgrading Opswise Controller refers to the increase from its currently installed Version, Release, or Modification level (see Versioning) to a later Version, Release, or Modification level. For example, upgrading Controller 1.7 or 5.1.0.7 to Controller 5.2.0.2.

Applying maintenance to Opswise Controller refers to the increase from its currently installed Maintenance level (see Versioning) to a later Maintenance level. For example, applying maintenance to Controller 5.2.0.0 to increase its maintenance level to Controller 5.2.0.2.

Exception for Opswise Controller 5.2.0 only

You can use the 5.2.0.x maintenance release of Opswise Controller to apply maintenance to any Opswise Controller 5.1.1.x release. You do not have to perform an upgrade.

The procedures for upgrading Opswise Controller differ from the procedures for applying maintenance to Opswise Controller.

Migrating from Transporter to OMS

Opswise Automation Center 5.2.0 replaces the Transporter and Message Hub with Opswise Message Service (OMS) as its network communication provider. OMS is message-oriented middleware that provides reliable network communications between Opswise Controller 5.2.0 and Opswise Universal Agent 5.2.0.

The Transporter and Message Hub are supported in the 5.2.0 release, but have been functionally stabilized, and support will be discontinued in a future release. It is highly recommended that all customers migrate from the Transporter and Message Hub to OMS as soon possible.

The effort to migrate from the Transporter and Message Hub to OMS for large Opswise Automation Center 5.2.0 deployments is significant. The transition has been made manageable with the following capabilities:

- Opswise Controller 5.2.0 supports both network communication providers concurrently. In addition to OMS, the Transporter and Message Hub continue to be supported as in prior releases in order to communicate with pre-5.2.0 Opswise Universal Agents. The ability to support both pre-5.2.0 and 5.2.0 Agents allows for the Agents to be migrated in manageable steps.
- Opswise Universal Agent 5.2.0 supports either of the two network communication providers. The 5.2.0 Agents can be configured to use either the Transporter and Message Hub, or OMS, but not both together. When an Agent configuration is changed to use a different network provider, it registers with the Controller with the same identity. Only the method used to communicate with the Controller changes.

The communication capabilities of Opswise Automation Center 5.2.0 allow for three migration strategies, which are described in the following sections:

1. Upgrading the Opswise Controller first, followed by the Opswise Agents.
2. Upgrading Opswise Universal Agents first, followed by the Opswise Controller.
3. Upgrade both the Opswise Controller and Opswise Agents at the same time.

Upgrading Opswise Controller First

In this upgrade scenario, the Opswise Controller is upgraded to 5.2.0 first. The Controller 5.2.0 continues to communicate with the pre-5.2.0 Opswise Universal Agents using the Transporter and Message Hub. The next step is to install an OMS Server and define it to the Controller. The final step is to upgrade the Opswise Universal Agents to 5.2.0 and configure them to use OMS. The Agents can be upgraded in manageable steps until all Agents have been upgraded.

In summary, the upgrade steps are:

1. Upgrade the Opswise Controller to 5.2.0 and continue using the 5.1.0 Transporter and Message Hub.
2. Install an OMS Server and define it to the 5.2.0 Controller.
3. Upgrade the Opswise Universal Agents to 5.2.0 and configure them to use the OMS Server.
4. Once all Agents are upgraded and using OMS, the Transporter and Message Hub should be uninstalled.

Upgrading Opswise Universal Agents First

In this upgrade scenario, the Opswise Universal Agents are upgraded to 5.2.0 first. The 5.2.0 Agents continue to use the 5.1.0 Transporter and Message Hub as their network provider. Once the Agents are upgraded, the Opswise Controller is upgraded to 5.2.0 and continues to use the Transporter and Message Hub. The next step is to install an OMS Server and define it to the Controller. The final step is to reconfigure the Opswise Universal Agents from the Transporter and Message Hub to OMS.

In summary, the upgrade steps are:

1. Upgrade the Opswise Universal Agents from 5.1.0 to 5.2.0, leaving them configured to use the Transporter as their network provider.
2. Upgrade the Opswise Controller to 5.2.0 and continue using the 5.1.0 Transporter and Message Hub.
3. Install an OMS Server and define it to the 5.2.0 Controller.
4. Reconfigure the Opswise Universal Agents to use the OMS Server as their network provider.
5. Once all Agents are upgraded and using OMS, the Transporter and Message Hub should be uninstalled.

Upgrading Opswise Controller and Opswise Universal Agent

In this scenario, the Opswise Controller and Opswise Universal Agents are upgraded to 5.2.0 at the same time. This is the most disruptive, as work cannot be executed until everything is upgraded. The Opswise Controller is upgraded to 5.2.0 without the Transporter and Message Hub. The OMS Server is installed, and an OMS Server definition is created in the Controller. The final step is to upgrade all Opswise Agents to 5.2.0 and configure them to use OMS.

In summary, the upgrade steps are:

1. Upgrade the Opswise Controller to 5.2.0. The upgrade does not require the Transporter and Message Hub.
2. Install an OMS Server and define it to the 5.2.0 Controller.
3. Upgrade all Opswise Agents to 5.2.0 and configure them for OMS.

Applying Maintenance

If you are applying maintenance to Opswise Controller, all Controller configurations are preserved.

If your Controller is configured to be backward-compatible with pre-5.2.0 Agents, it still will be configured that way after maintenance is applied.

Upgrading Opswise Controller

- Overview
  - Upgrading vs. Applying Maintenance
- Upgrade Procedures
  - Supported Upgrade Paths
- Make Sure No Records Are Being Processed
- Stop OMS
- Back Up Your Database
- Run an Export on the Active Controller
  - Export Scripts
  - Running the Export
- Stop Tomcat and Remove All Controllers
- Prepare Your Database
- Download the New Controller
- Install the Controller
- Verify the Active Controller Installation
- Run an Import on the Active Controller
- Check Your Data
- Verify the Passive Controller Installations
- Start OMS
- Verify the Upgrade

Overview

Upgrading Opswise Controller refers to the increase of a currently installed Version, Release, or Modification level of the Controller (see Versioning) to a later Version, Release, or Modification level (for example, upgrading Controller 1.7 or 5.1.0.7 to Controller 5.2.0.2).

Exception for Opswise Controller 5.2.0 only

You can use the 5.2.0.x maintenance release of Opswise Controller to apply maintenance to any Opswise Controller 5.1.1.x release. You do not have to perform an upgrade.

Controller 5.2.0 uses Opswise Message Service (OMS) 5.2.0 as the network communications provider between itself and Opswise Universal Agent 5.2.0. However, Controller 5.2.0 must use Opswise Automation Center Outboard (Message Hub and Transporter) 5.1.0 as the network communications provider between itself and pre-5.2.0 Agents. Therefore, if you want to use OMS as the network communications provider between Controller and Agents, you must upgrade the Agents to version 5.2.0. OMS is packaged as a component of Opswise Universal Agent 5.2.0 (for UNIX and Windows) and is installed as part of the Agent installation.

Likewise, if you are upgrading the Controller from a pre-5.1 release (1.5, 1.6, or 1.7) to version 5.2.0, and you want to continue running tasks on pre-5.2.0 Agents, you must upgrade the Outboard to a 5.1.x release *after* the Controller upgrade is complete.

Note
Upgraded 5.1.1.x Controllers will maintain their pre-existing Message Hub and Transporter configuration properties files. However, if you will be using the Message Hub and Transporter as the network provider between the Controller and any Agents (5.2.x and/or 5.1.x), you must enter the Hub Hostname and Hub Port of the Message Hub in the Cluster Node definition for the upgraded Controller.

Upgrading vs. Applying Maintenance

Applying maintenance to a Controller refers to the increase of its currently installed Maintenance level (see Versioning) to a later Maintenance level (for example, applying maintenance to Controller 5.2.0.0 to increase its maintenance level to Controller 5.2.0.2).

The procedures for applying maintenance differ from the procedures for upgrading (see Applying Maintenance to Opswise Controller).

Upgrade Procedures

These instructions comprise the following procedures:

!	Make Sure No Records Are Being Processed
2	Stop OMS
3	Back Up Your Database
4	Run an Export on the Active Controller
5	Stop Tomcat and Remove All Controllers
6	Prepare Your Database
7	Download the New Controller
8	Install the Controller
9	Verify the Active Controller Installation
10	Run an Import on the Active Controller
11	Check Your Data
12	Verify the Passive Controller Installations
13	Start OMS
14	Verify the Upgrade

The instructions assume that you are running a High Availability Opswise Controller system: a system configured with Active and Passive Controllers (cluster nodes). If you are running a single Controller, disregard the steps for the Passive Controller.

Supported Upgrade Paths

You can use these instructions for the supported upgrade paths shown in the following table. For any other upgrade path, consult your Stonebranch representative.

Upgrade Controller to...	1.6.0	1.7.0	5.1.0	5.2.0
From 1.5.0	✓	✓	✓	✓
From 1.6.0		✓	✓	✓
From 1.7.0			✓	✓
From 5.1.0				✓

If you are upgrading the Controller to 5.2.0, you also should update your Agents to 5.2.0. OMS cannot be used as the network communications provider between a 5.2.0 Controller and pre-5.2.0 Agents.

Make Sure No Records Are Being Processed

WARNING
If the Controller is processing task instances when you launch the export, the results are unpredictable.

Step 1	Log in with **ops.admin** or a user with administrator privileges.
Step 2	Disable all active triggers to make sure no tasks are being processed.
Step 3	Check the Activity screen to verify that there are no active task instances. If there are, wait until they complete before you start the export process. If necessary, you can force finish tasks.

Stop OMS

Stop Opswise Message Service (OMS).

The start/stop procedure for Opswise Universal Agent components (such as OMS) may differ depending on your platform. For instructions, see Starting and Stopping Agent Components.

Back Up Your Database

> **Important**
> Before upgrading your Controllers, back up your database. The database backup is a fail-safe measure; you will be using the Controller export/import utility described below to migrate your data.

Run an Export on the Active Controller

In this procedure, you are performing a bulk export of data that you will import to your upgraded system in a later procedure using the bulk import.

Export Scripts

Export scripts copy and save the specified records to one or more XML files. The files can then be imported into the upgraded system.

The list of export scripts changed from Opswise Automation Center 1.7 to Opswise 5.1. Depending on what version you are upgrading from, select the appropriate upgrade script from the following descriptions.

Exporting Records from Opswise Automation Center 1.7 or Earlier

The following export scripts are available:

- **opswise_bulk_exec_export.js** - Not recommended for migration. Exports all unfinished activity (task instances in the Activity display).
- **opswise_bulk_export.js** - Exports all record definitions and task instance history, which includes all task instances in an end state (cancelled, failed, skipped, finished, success).

Exporting Records from Opswise Controller 5.1 or Later

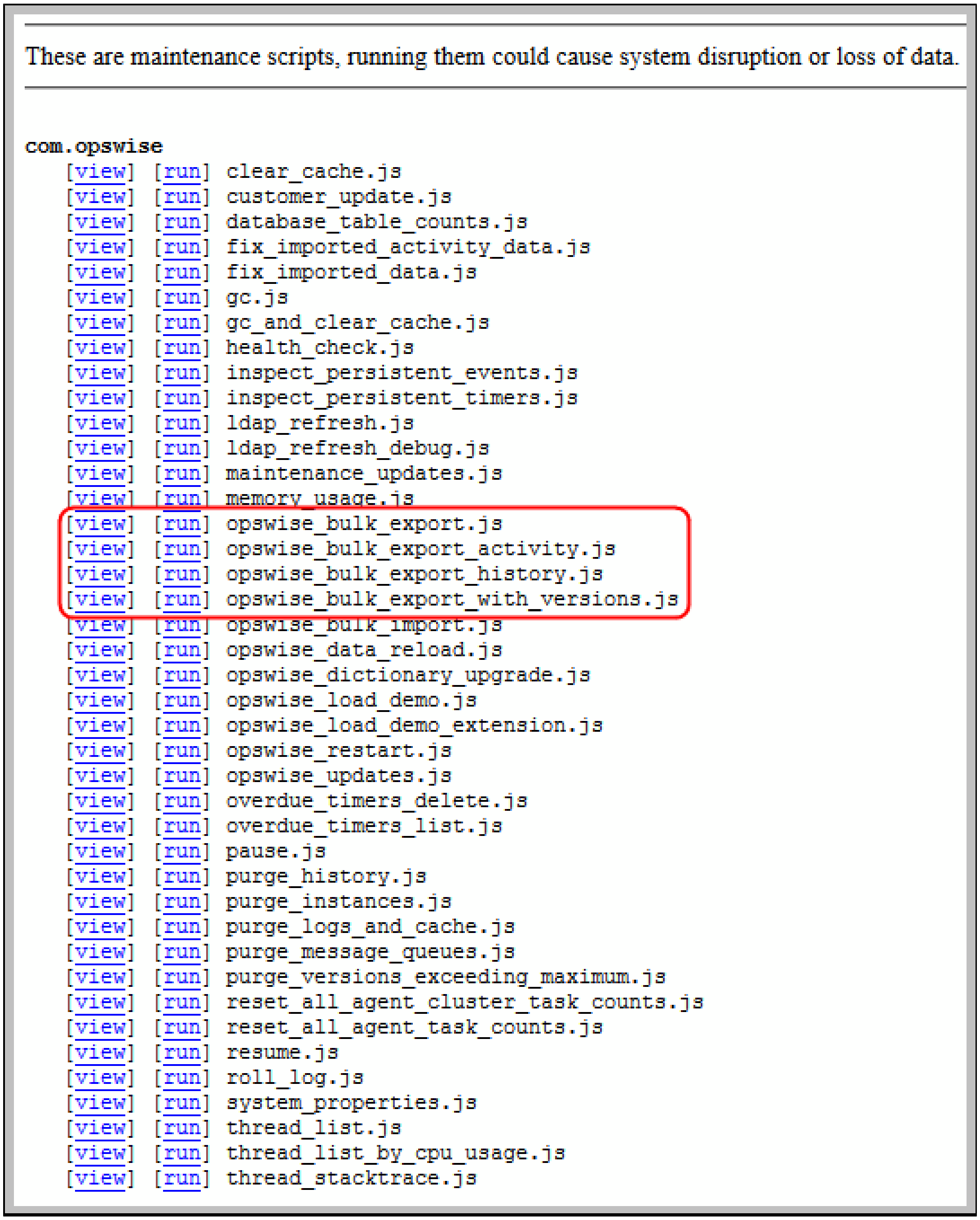
The following export scripts are available:

- **opswise_bulk_export_activity.js** - Not recommended for migration. Exports all unfinished activity (task instances in the Activity display).
- **opswise_bulk_export.js** - Exports all current record definitions, without versions.
- **opswise_bulk_export_with_versions.js** - Exports all current records along with older (non-current) versions of record definitions.
- **opswise_bulk_export_history.js** - Exports task instance history, which includes all task instances in an end state (cancelled, failed, skipped, finished, success).

Running the Export

Perform the following steps to run a bulk export:

Step 1 From the navigation pane, select **Automation Center Administration > Configuration > Maintenance Scripts**. The image below shows export script options for Controller 5.2.0.

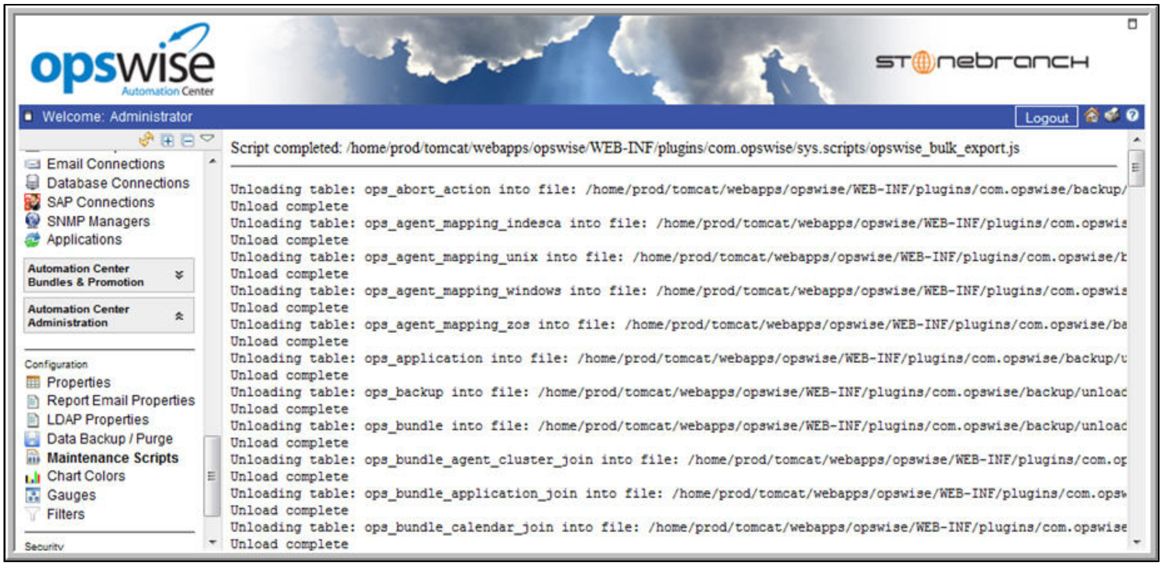


These are maintenance scripts, running them could cause system disruption or loss of data.

```
com.opswise
    [view] [run] clear_cache.js
    [view] [run] customer_update.js
    [view] [run] database_table_counts.js
    [view] [run] fix_imported_activity_data.js
    [view] [run] fix_imported_data.js
    [view] [run] gc.js
    [view] [run] gc_and_clear_cache.js
    [view] [run] health_check.js
    [view] [run] inspect_persistent_events.js
    [view] [run] inspect_persistent_timers.js
    [view] [run] ldap_refresh.js
    [view] [run] ldap_refresh_debug.js
    [view] [run] maintenance_updates.js
    [view] [run] memory_usage.js
    [view] [run] opswise_bulk_export.js
    [view] [run] opswise_bulk_export_activity.js
    [view] [run] opswise_bulk_export_history.js
    [view] [run] opswise_bulk_export_with_versions.js
    [view] [run] opswise_bulk_import.js
    [view] [run] opswise_data_reload.js
    [view] [run] opswise_dictionary_upgrade.js
    [view] [run] opswise_load_demo.js
    [view] [run] opswise_load_demo_extension.js
    [view] [run] opswise_restart.js
    [view] [run] opswise_updates.js
    [view] [run] overdue_timers_delete.js
    [view] [run] overdue_timers_list.js
    [view] [run] pause.js
    [view] [run] purge_history.js
    [view] [run] purge_instances.js
    [view] [run] purge_logs_and_cache.js
    [view] [run] purge_message_queues.js
    [view] [run] purge_versions_exceeding_maximum.js
    [view] [run] reset_all_agent_cluster_task_counts.js
    [view] [run] reset_all_agent_task_counts.js
    [view] [run] resume.js
    [view] [run] roll_log.js
    [view] [run] system_properties.js
    [view] [run] thread_list.js
    [view] [run] thread_list_by_cpu_usage.js
    [view] [run] thread_stacktrace.js
```

Step 2 Select the script you want to run and click **Run**.

Step 3 The utility prompts for a confirmation. Click **Yes**. As your data is exported, the output from the script is written to the screen, as shown.



Script completed: /home/prod/tomcat/webapps/opswise/WEB-INF/plugins/com.opswise/sys.scripts/opswise_bulk_export.js

```
Unloading table: ops_abort_action into file: /home/prod/tomcat/webapps/opswise/WEB-INF/plugins/com.opswise/backup/
Unload complete
Unloading table: ops_agent_mapping_indesca into file: /home/prod/tomcat/webapps/opswise/WEB-INF/plugins/com.opswis
Unload complete
Unloading table: ops_agent_mapping_unix into file: /home/prod/tomcat/webapps/opswise/WEB-INF/plugins/com.opswise/b
Unload complete
Unloading table: ops_agent_mapping_windows into file: /home/prod/tomcat/webapps/opswise/WEB-INF/plugins/com.opswis
Unload complete
Unloading table: ops_agent_mapping_zos into file: /home/prod/tomcat/webapps/opswise/WEB-INF/plugins/com.opswise/ba
Unload complete
Unloading table: ops_application into file: /home/prod/tomcat/webapps/opswise/WEB-INF/plugins/com.opswise/backup/u
Unload complete
Unloading table: ops_backup into file: /home/prod/tomcat/webapps/opswise/WEB-INF/plugins/com.opswise/backup/unloac
Unload complete
Unloading table: ops_bundle into file: /home/prod/tomcat/webapps/opswise/WEB-INF/plugins/com.opswise/backup/unloac
Unload complete
Unloading table: ops_bundle_agent_cluster_join into file: /home/prod/tomcat/webapps/opswise/WEB-INF/plugins/com.op
Unload complete
Unloading table: ops_bundle_application_join into file: /home/prod/tomcat/webapps/opswise/WEB-INF/plugins/com.opsw
Unload complete
Unloading table: ops_bundle_calendar_join into file: /home/prod/tomcat/webapps/opswise/WEB-INF/plugins/com.opswise
Unload complete
```

Step 4 Look over the output for any error messages. If you see any, copy the output to a file and send it to support@stonebranch.com.

Step 5 Zip or tar the contents of:

```
[tomcat directory]/webapps/opswise/WEB-INF/plugins/com.opswise/backup/unload/
```

Step 6 Copy the zip/tar file to a safe place for use after the upgrade process.

Step 7 Copy your **`glide.properties`** file to a safe place. You may need to consult this file later. The file is located here:

```
[tomcat directory]/webapps/opswise/WEB-INF/properties
```

Step 8 Copy your license key from the Opswise Properties screen and store it in a safe place.

Step 9 Copy the LDAP mapping file to a safe place for use after the upgrade process.

```
[tomcat directory]/webapps/opswise/WEB-INF/properties/users/ldapmap.xml
```

Stop Tomcat and Remove All Controllers

Important
Make sure you have copied to a safe location all of the exported files from the bulk export before continuing here, where you will stop Tomcat and remove the Controller.

Step 1 Stop the Tomcat containers in which all Passive Controllers are deployed:

Windows
Use the services application to stop Tomcat. You also can issue the stop command on a command line:

```
net stop [name of Tomcat service]
```

UNIX
Stop the daemon using the script found in the `/etc/init.d` directory for Tomcat.

```
service [name of Tomcat service] stop
```

Windows or UNIX
Stop the service using the **`$TOMCAT_HOME/bin/shutdown.bat`** or **`$TOMCAT_HOME/bin/shutdown.sh`** scripts:

- **Windows**

```
cd $CATALINA_HOME\bin
shutdown
```

- **Linux/Unix**

```
cd $CATALINA_HOME/bin
./shutdown
```

Step 2 Use the Windows Task Manager to confirm that the Tomcat processes where the Passive Controllers are deployed are not running.

Step 3 Back up the Passive Controller deployment directories in any folder other than one under the Tomcat installation.

For example, if you deployed the Opswise Controller **war** as **`opswise.war`**, the following would be your deployment directory:

```
[tomcat-install]\webapps\opswise
```

Step 5 Repeat steps 1 through 3 for the Active Controller.

Step 4 Delete the deployment directory and **`opswise.war`** for all Controllers.

For example, if you deployed the Controller **war** as **`opswise.war`**, the following would be your deployment directory and **`opswise.war`**:

```
[tomcat-install]\webapps\opswise
[tomcat-install]\webapps\opswise.war
```

Prepare Your Database

Delete or drop your database using the appropriate database admin tool.

Important
Before dropping your existing database, make sure you have created a backup, as mentioned at the beginning of this procedure.

Download the New Controller

From the Stonebranch website, download an Opswise Controller package from the Current Product Downloads page (for instructions, see Downloading Opswise Controller Software).

Install the Controller

Install the Opswise Controller for the Active and all Passive deployments.

Note
If you are upgrading the Controller from a 5.1.0 or earlier release to version 5.2.0, but you want to continue running tasks on pre-5.2.0 Agents, you must include the --hubhost and --hubport command line switches in the install to continue using the 5.1.0 Message Hub and Transporter as the network communications provider.

The Opswise Controller is a Java application running within Apache Tomcat. For this reason, the Controller software and installation procedure is basically the same for all platforms.

Note
If you have deployed any JDBC driver jar files (or in the case of DB2, a JDBC driver license jar file) to the `$TOMCAT_HOME/webapps/opswise/WEB-INF/lib` directory, you must recopy these files to this directory and restart tomcat after your initial validation.

Verify the Active Controller Installation

Step 1 Start Tomcat where the Active Controller is deployed.

When the database initialization is complete and the Controller is running, you will see the following (for example) in the log:

```
2012-09-12-12:53:07:339   INFO [Ops.Cluster.Monitor.0] Server is now Running in Active mode.
Previous mode was Passive/Unavailable.
2012-09-12-12:53:07:339   INFO [Ops.Cluster.Monitor.0] Setting server to ACTIVE.
```

Step 2 Log in to the Active Controller with **`ops.admin`** (password is not set). On the Opswise Automation Center Home page, verify that the Overview specifies the correct release.

Run an Import on the Active Controller

In this procedure, you are performing a bulk import of the data that you exported earlier using a bulk export.

Step 1 Unzip/untar the backup file that you created earlier using the export.

Step 2 Copy the XML files to the following directory:

```
 $CATALINA_HOME/webapps/opswise/WEB-INF/plugins/com.opswise/backup/unload
```

Step 3	From the navigation pane, select **Automation Center Administration > Configuration > Maintenance Scripts**.
Step 4	Locate and run the script: `opswise_bulk_import.js`
Step 5	The utility prompts for a confirmation. Click **Yes**.
Step 6	As your data is imported, the output from the script is written to the screen. Look over the output for any error messages. If you see any, copy the output to a file and send it to support@stonebranch.com.
Step 7	Run the following script: `clear_cache.js`
Step 8	If you are importing data from a 5.1.0.x Controller, and the data you exported using the bulk export included Activity data (**opswise_bulk_export_activity.js** in version 5.1.0.x or **opswise_bulk_exec_export.js** in version 1.7 or earlier), then you also must run the **fix_imported_activity_data.js** script.

Check Your Data

At this point, your previous definitions, users and passwords have all been restored. Log out and in again, and review your records to make sure all your previous definitions, users, and passwords have been restored successfully.

Verify the Passive Controller Installations

Step 1	Start Tomcat where each Passive Controller is deployed.
Step 2	Log in to the Passive Controller with **`ops.admin`** or a user with equivalent authorization. On the Opswise Automation Center Home page, verify that the Overview specifies the correct release.

Start OMS

Start Opswise Message Service (OMS).

The start/stop procedure for Opswise Automation Center components may differ depending on your platform. For instructions, see Starting and Stopping Opswise Controller and select your platform.

Verify the Upgrade

Verify that the Controller is installed and running properly (see Verifying a Controller Installation).

Verify that your Agent components are communicating with the Active Controller (see Verifying Opswise Universal Agent Installation).

Applying Maintenance to Opswise Controller

- Overview
  - Applying Maintenance vs. Upgrading
- Opswise Controller Maintenance
  - Paused Mode

Overview

Applying maintenance to Opswise Controller 5.2.0 refers to the increase from the currently installed Maintenance level (see Versioning) to a later Maintenance level (for example, applying maintenance to Controller 5.2.0.0 to increase its maintenance level to Controller 5.2.0.2).

Exception for Opswise Controller 5.2.0 only

You can use the 5.2.0.x maintenance release of Opswise Controller to apply maintenance to any Opswise Controller 5.1.1.x release. You do not have to perform an upgrade.

Note
If you are using an Opswise Automation Center Outboard (Message Hub and Transporter) as the network communications provider between the Controller and Agents, and you want to apply maintenance to the Outboard, see Applying Opswise Outboard Maintenance in the Opswise 5.1.1 Documentation.

Since Opswise Message Service (OMS), which replaces the Outboard in Opswise Automation Center 5.2.0, is a component of Opswise Universal Agent, any maintenance for OMS is implemented by applying maintenance to the Agent.

The 5.2.0 Controller and Agent, and 5.1.0 Outboard, can be at different Release, Modification, and Maintenance levels of the same Version (for example, Controller 5.2.0.0, Agent 5.1.0.10, and Outboard 5.1.0.12).

Applying Maintenance vs. Upgrading

Upgrading a Controller refers to the increase of its currently installed Version, Release, or Modification level (see Versioning) to a later Version, Release, or Modification level (for example, upgrading Controller 1.7 or 5.1.0.x to Controller 5.2.0.2).

The procedures for upgrading differ from the procedures for applying maintenance (see Upgrading Opswise Controller).

Opswise Controller Maintenance

To apply maintenance to the currently installed release of Opswise Controller:

Step 1	From the Stonebranch website, download the Opswise Controller 5.2.0.x package from the Maintenance page (for instructions, see Downloading Opswise Controller Software).
Step 2	Follow the instructions in the README file for your platform; README files for Windows and UNIX are included in the package.
Step 3	Verify that your Agents and OMS are communicating with the Controller server. See Verifying an Installation for instructions.

Paused Mode

After you apply maintenance and restart the Controller instance, it could - depending on the Controller release - enter into Paused mode if you did not execute the maintenance_updates.js script during the maintenance procedure, as specified in the README file.

If the Controller has entered Paused mode, the Overview widget on the Opswise Automation Center Home Page will display the following information in the **Node** field:

```
hostname:8080-opswise [Mode: (--PAUSED FOR MAINTENANCE--)] - Uptime: mm Minutes ss Seconds
```

The Controller instance will remained paused until execution of maintenance_updates.js has completed.

Starting and Stopping Opswise Controller

These pages provide platform-specific instructions for starting and stopping Opswise Controller 5.2.0:

- Starting and Stopping Opswise Controller - UNIX and Windows
- Starting and Stopping Opswise Bundled Controller on AIX

Starting and Stopping Opswise Controller - UNIX and Windows

- Starting and Stopping the Controller on UNIX
- Starting and Stopping the Controller on Windows

Starting and Stopping the Controller on UNIX

Note
These procedures are appropriate for all [supported systems] of UNIX:

Linux	To start or stop the Controller (all versions), issue the following commands:
	`/$TOMCAT_HOME/bin/startup.sh`
	`/$TOMCAT_HOME/bin/shutdown.sh`
	or
	`service tomcat start`
	`service tomcat stop`
	If you have configured your system with **init.d**, you also can use the following commands:
	`/etc/init.d/tomcat start`
	`/etc/init.d/tomcat stop`

AIX	The procedures for starting and stopping the Controller are dependent on how Tomcat was configured when the Controller installed.

Starting and Stopping the Controller on Windows

To start or stop the Controller (all versions) from the DOS prompt, use the following commands:

```
c:\$CATALINA_HOME\bin\startup.bat
c:\$CATALINA_HOME\bin\shutdown.bat
```

Starting and Stopping Opswise Bundled Controller on AIX

Starting the Opswise Bundled Controller

To start the Opswise Bundled Controller, execute the following script:

```
/etc/rc.d/rc2.d/Sopswise start
```

Stopping the Opswise Bundled Controller

To stop the Opswise Bundled Controller, execute the following script:

```
/etc/rc.d/rc2.d/Kopswise stop
```

Administration

Overview

Administration of Opswise Controller

High Availability

Introduction

High Availability System

How High Availability Works

What To Do If a Failover Occurs

High Availability Configuration

High Availability Components

Maintenance Scripts

Maintenance Scripts

Ports Configuration

Ports Configuration

Opswise Controller I

Overview

Opswise Controller Star

Opswise Controller Sys

LDAP Properties

User Interface Propertie

Report Email Properties

Command Line Interface

The information on these pages also is located in the Opswise Controller 5.2.0 Installation and Administration Guide.pdf.

Administration Overview

Opswise Controller Administration

Administration of Opswise Controller includes:

Opswise Controller Start-Up Properties	These properties are required for Controller start-up, initialization, and operation. They are contained in the **`glide.properties`** file and have their values set during installation. To reset the values, you must stop the Controller, edit **`glide.properties`**, and restart the Controller.
Opswise Controller System Properties	These properties define Controller system information and performance. They have their values set during installation. They are available, and can be reset, only via the user interface.
Additional Opswise Controller Properties	These properties are Controller feature-specific properties. • LDAP Properties • User Interface Properties • Report Email Properties They have their values set only via the user interface; they are not set at installation.
Opswise Command Line Interface (CLI) Properties	CLI provides a sample configuration file, **`cmdtools.props`**, that you can use to pass CLI Global parameters to a CLI command. The file is created during installation of Opswise Universal Agent if Opswise Command Line Interface has been selected to be installed.
Maintenance Scripts	Opswise Controller maintenance scripts help you maintain and administer your Controller installation. Many of these scripts should be run only by Technical Support or upon request by Technical Support.

⚠ **Note**
If you are using an Opswise Automation Center 5.1.0 Outboard (Message Hub and Transporter) as the network communications provider between a 5.2.0 Controller and pre-5.2.0 Agents, see Message Hub Properties and Transporter Properties in Opswise Automation Center 5.1.1.

High Availability

- Introduction
- Opswise Controller Changes with Opswise Message Service
- High Availability System
- High Availability Components
  - Cluster Nodes
  - Agent
  - Opswise Message Service (OMS)
- How High Availability Works
  - Cluster Node Status
  - High Availability Start-Up
  - Determining Mode of a Cluster Node at Start-up
  - Checking the Active Cluster Node During Operations
- What To Do If a Failover Occurs
  - Viewing Cluster Node Status
- High Availability Configuration
  - Configuring Cluster Nodes
  - Configuring OMS
  - Configuring Agents
  - Configuring Notifications Based on Component Status
- Load Balancer

Introduction

High Availability (HA) of Opswise Automation Center means that it has been set up to be a redundant system; in addition to the components that are processing work, there are back-up components available to continue processing through hardware or software failure.

This page describes a High Availability environment, how High Availability components recover in the event of such a failure, and what actions, if any, the user must take.

Opswise Controller Changes with Opswise Message Service

The Opswise Message Service (OMS) network communication component was introduced in the 5.2.0 release. OMS improves the network communication reliability and availability.

Its introduction also introduces the following changes to the Opswise Controller:

- New deployments of the 5.2.0.x Controller are not pre-configured to use a Message Hub.
- Upgraded Controller instances (from 5.1.1.x) will maintain their pre-existing Message Hub configuration.
- The Controller can operate with multiple network providers (one Message Hub and multiple OMS servers) in order to communicate with 5.1.0 and 5.2.0 Agents.
- Any 5.2.x.x Controller (cluster node) with a configured Message Hub will not change with respect to HA behavior.
- Any 5.2.x.x Controller (cluster node) without a configured Message Hub will change with respect to HA behavior.
- Cluster node becomes Passive/Available immediately and is immediately eligible to become the Active cluster node.
- Disconnection from an OMS Server or servers has no impact on a cluster node’s fail-over decision.
- To change the Controller to use an OMS Server-only configuration, the administrator can clear the Message Hub Hostname and Hub Port from all defined cluster nodes.

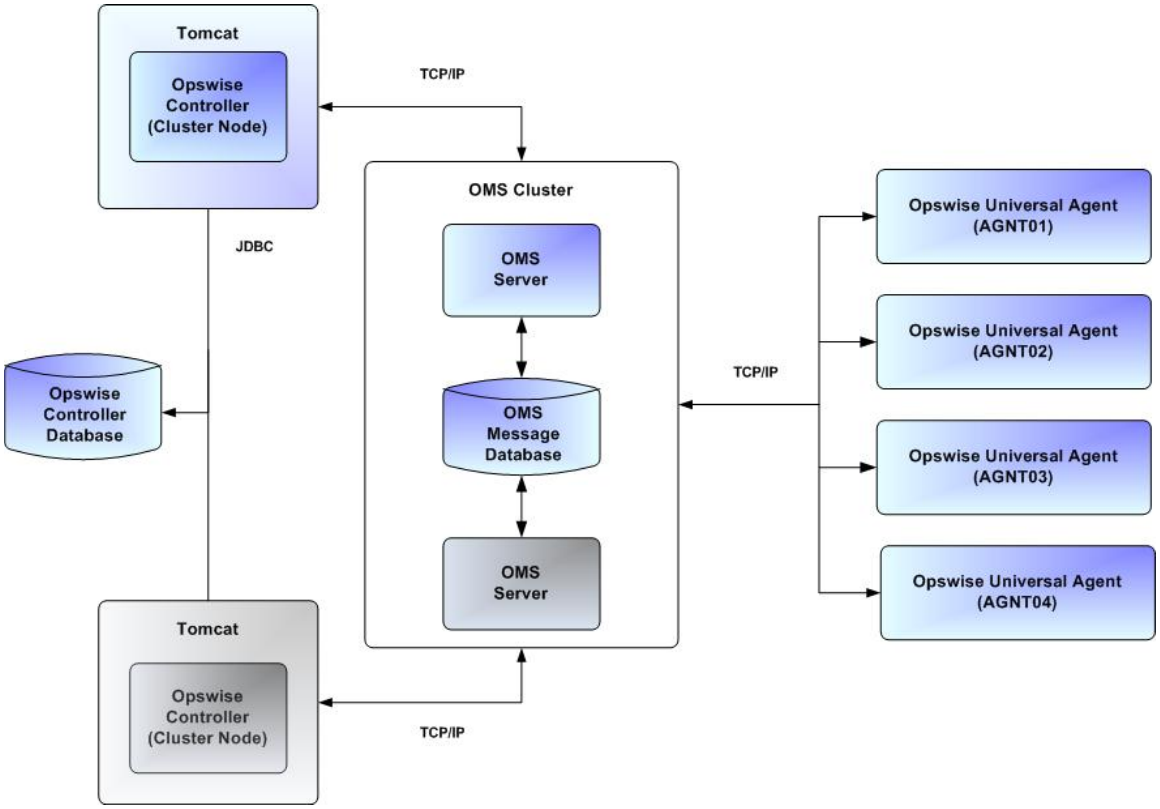
High Availability System

The following illustration is a typical, although simplified, Opswise Automation Center system in a High Availability environment.

In this environment, there are:

- Two Opswise Controller instances (Cluster Nodes)
- Two Opswise Message Service (OMS) network communications providers in an OMS cluster
- Four Opswise Universal Agent (Agent) machines

The components in blue are active and operating. The components in gray are available for operations but currently are inactive (passive).


Tomcat
Opswise Controller (Cluster Node)
TCP/IP
JDBC
OMS Cluster
OMS Server
Opswise Controller Database
OMS Message Database
OMS Server
TCP/IP
Opswise Universal Agent (AGNT01)
Opswise Universal Agent (AGNT02)
Opswise Universal Agent (AGNT03)
Opswise Universal Agent (AGNT04)
Tomcat
Opswise Controller (Cluster Node)
TCP/IP

See High Availability Components for a detailed description of how each component type functions in a High Availability environment.

Note
An Opswise Automation Center system environment must use the Opswise Automation Center 5.1.0 Outboard (Message Hub and Transporter) as the network communications provider between a 5.2.0 Controller and 5.1.x Agents.

For information on High Availability configuration with a 5.1.0 Outboard, see High Availability in the Opswise 5.1.1 Documentation.

High Availability Components

This section provides detailed information on the cluster nodes and Agents in a High Availability environment.

Cluster Nodes

Each Opswise Automation Center installation consists of one or more instances of Opswise Controller; each instance is a cluster node. Only one node is required in an Opswise Automation Center system; however, in order to run a High Availability configuration, you must run at least two nodes.

At any given time under High Availability, one node operates in Active mode and the remaining nodes operate in Passive/Available and/or Passive/Unavailable mode.

An Active node performs all system processing functions; Passive nodes can perform limited processing functions.

Passive Cluster Node Limitations

Passive cluster nodes cannot execute any automated or scheduled work.

Also, from a Passive node you cannot:

- Perform a workflow instance insert task operation.
- Update a task instance.
- Update an enabled trigger.

- Update an enabled data backup/purge.

However, Passive nodes do let you perform a limited number of processing functions, such as:

- Launch tasks.
- Monitor and display data.
- Access the database.
- Generate reports.

Agent

The Agent runs as a Windows service or Linux/Unix daemon. A cluster node sends a request to the Agent to perform a function. The Agent processes the request, gathers data about the operation of the client machine, and sends status and results back to the node. It performs these functions by exchanging messages with the node.

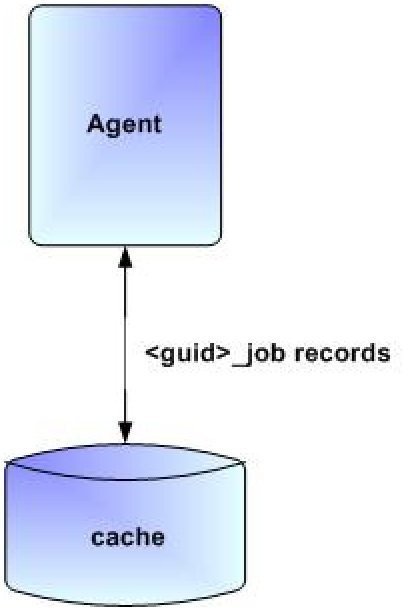
Once an Agent has registered with a node, you can view it by selecting that Agent type from the **Automation Center Resources** section of the Opswise Controller user interface navigation pane. A list displays showing all the registered Agents of that type. See Agents for more information.

If an Agent fails, Universal Broker restarts it. The Agent then attempts to determine what tasks or functions were in process at the time of failure.

In order to support such a determination, Agent task processing includes the following steps:

Step 1	Each time the Agent receives a task, it writes to cache a record called **`[guid]_job`**, where **`[guid]`** is a unique tracking number assigned to the task instance.
Step 2	As the task runs, the Agent updates the **`[guid]_job`** record with status information.
Step 3	When the task run completes, the Agent deletes the **`[guid]_job`** record.
Step 4	If an Agent is restarted, it looks in the cache for **`[guid]_job`** records. If any are found, the Agent looks at the status. If the record indicates that the job is supposed to be running, the Agent searches the system to locate it. If the Agent is able to locate the task and resume tracking, it continues and marks the task resumed. If the Agent is not able to resume tracking a task, it returns a message to the Cluster Node, setting the status of the task instance to **IN-DOUBT**. This then requires manual follow-up to determine the state of the process.

As illustrated below, the Agent reads/writes a record to its agent/cache directory for each task instance that it manages.



Opswise Message Service (OMS)

Opswise Message Service (OMS) sends and receives messages between the cluster nodes and Agents.

OMS consists of an OMS Server and an OMS Administration Utility. The OMS clients - cluster nodes and Agents - establish persistent TCP/IP socket connections with the OMS Server.

OMS provides for reliable message communication by persisting all OMS queued messages to persistent storage. The OMS Server maintains

OMS queues in an OMS message database that resides on persistent storage.

See Opswise Message Service (OMS) for detailed information on OMS.

How High Availability Works

In a High Availability environment, passive Cluster Nodes play the role of standby servers to the active (primary) Cluster Node server. All running nodes issue heartbeats and check the status of other running nodes, both when they start up and continuously during operations. If a node that currently is processing work can no longer do so, one of the other nodes will take over and continue processing.

Each Cluster Node connects to the Opswise Controller database and to an OMS HA cluster, and each Agent connects to the same OMS HA cluster. An Opswise Controller HA configuration can use a single OMS server, that is not an HA cluster, with the understanding that a single OMS server would introduce a single point of failure. Using an OMS HA cluster is recommended.

See High Availability Configuration for information on how these connections are made.

Cluster Node Status

The status (mode) of a Cluster Node indicates whether or not it is the node processing work:

Active	Cluster node currently is performing all system processing functions.
Passive/Available	Cluster Node is connected to OMS and is available to perform all system processing functions.
Passive/Unavailable	Cluster Node is not connected to OMS but is available to perform all system processing functions, except that it would not be able to exchange data with an Agent.
Offline	Cluster node is not running or is inoperable and needs to be restarted.

⚠ **Note**
Cluster nodes in Passive/Available and Passive/Unavailable status can perform limited system processing functions.

High Availability Start-Up

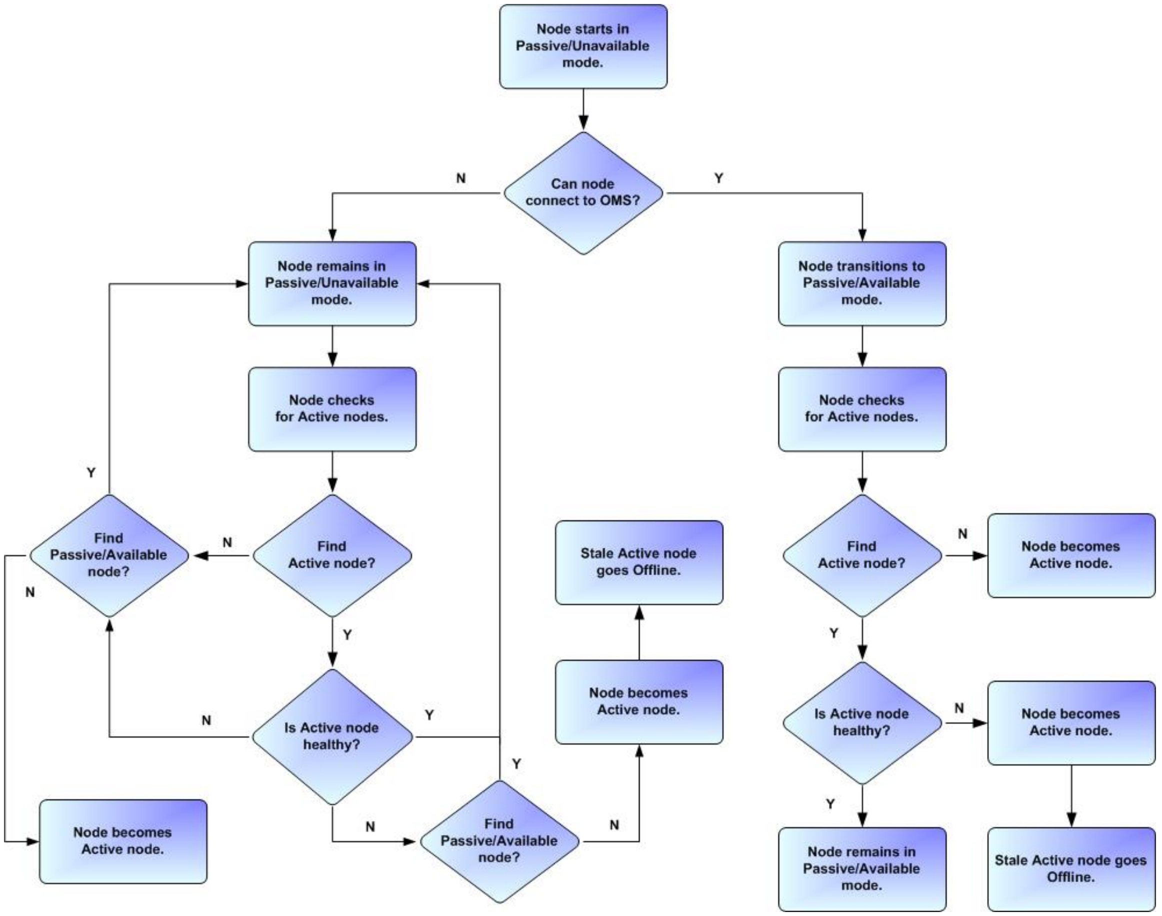
The following steps describe how a High Availability environment starts up:

Step 1	User starts the Cluster Nodes.
Step 2	Each Cluster Node reads its `glide.properties` file.
Step 3	Each Cluster Node locates and connects to the database and retrieves information about the Opswise Automation Center environment.
Step 4	Each Cluster Node connects to an OMS server.
Step 5	Each Agent connects to an OMS server.

Determining Mode of a Cluster Node at Start-up

A Cluster Node starts in Passive/Unavailable mode. It then determines if it should remain in Passive/Unavailable mode or switch to either Passive/Available or Active mode.

The following flow chart describes how a Cluster Node determines its mode at start-up:


Node starts in Passive/Unavailable mode.
Can node connect to OMS?
N
Y
Node remains in Passive/Unavailable mode.
Node transitions to Passive/Available mode.
Node checks for Active nodes.
Node checks for Active nodes.
Y
Find Passive/Available node?
N
Find Active node?
Stale Active node goes Offline.
Find Active node?
N
Node becomes Active node.
N
Y
Y
Node becomes Active node.
N
Is Active node healthy?
Y
Is Active node healthy?
N
Node becomes Active node.
Y
Y
Node becomes Active node.
N
Find Passive/Available node?
N
Node remains in Passive/Available mode.
Stale Active node goes Offline.

> **Note**
> A cluster node is considered "healthy" or "stale" based on its heartbeat timestamp.

Checking the Active Cluster Node During Operations

When all Cluster Nodes have started, each one continuously monitors the heartbeats of the other running nodes.

If a Passive node determines that the Active node is no longer running, the Passive node automatically takes over as the Active node based upon the same criteria described above:

- If the node is in Passive/Available mode, it will take over as the Active node.
- If the node is in Passive/Unavailable mode, it will take over as the Active node only if no other node is in Passive/Available mode.

This determination is made as follows:

Step 1	The Active node sends a heartbeat by updating a timestamp in the database. The heartbeat interval is 10 (seconds).
Step 2	All Passive nodes (Available and Unavailable) check the Active node's timestamp to determine if it is current. (This check runs every 60 seconds.)
Step 3	If a Passive node determines that the Active node's timestamp is stale, failover occurs: the Passive node changes the status of the Active node to Offline and takes over as the Active node. If more than one node is operating in Passive mode, the first node eligible to become Active that determines that the Active node is not running becomes the Active node. A stale node is a node whose timestamp is older than 5 minutes.

Active Cluster Node without an OMS Connection

An Active Cluster Node that loses the connection to OMS checks regularly to see if there is a Passive/Available node eligible to become Active; that is, it has a connected OMS. If it finds an eligible node, the Active node without a connected OMS automatically will stop and restart. This allows the Passive/Available node to take over as the Active node.

This stop and restart is done within the Tomcat process. That is, the Tomcat process remains running while the Cluster Node just shuts down its internal processes (threads) and then starts them back up. Tomcat itself is not stopped and started.

What To Do If a Failover Occurs

A Passive node taking over as an Active node is referred to as failover. If failover occurs, the event is invisible unless you are using the Active node in a browser.

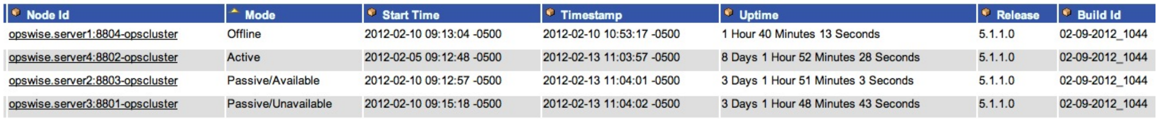
If you are using the Active node in a browser and the node fails, you will receive a browser error. In this case, take the following steps to continue working:

Step 1	Access the new Active node in your browser. To determine which node is now Active, from the navigation pane select **Automation Center Resources > Cluster Nodes** and check the **Mode** column on the Cluster Nodes list screen (see Viewing Node Status, below).
Step 2	If you were adding, deleting, or updating records at the time of the failure, check the record you were working on. Any data you had not yet saved will be lost.

Viewing Cluster Node Status

To view a list of all Cluster Nodes, select **Automation Center Resources > Cluster Nodes**. The Cluster Node list screen displays a list of all registered Cluster Nodes, as shown in the following example.

Note
A Cluster Node becomes registered the first time it starts. From then on, it always appears in the Cluster Nodes list, regardless of its current mode or status.



Node Id	Mode	Start Time	Timestamp	Uptime	Release	Build Id
opswise.server1:8804-opscluster	Offline	2012-02-10 09:13:04 -0500	2012-02-10 10:53:17 -0500	1 Hour 40 Minutes 13 Seconds	5.1.1.0	02-09-2012_1044
opswise.server4:8802-opscluster	Active	2012-02-05 09:12:48 -0500	2012-02-13 11:03:57 -0500	8 Days 1 Hour 52 Minutes 28 Seconds	5.1.1.0	02-09-2012_1044
opswise.server2:8803-opscluster	Passive/Available	2012-02-10 09:12:57 -0500	2012-02-13 11:04:01 -0500	3 Days 1 Hour 51 Minutes 3 Seconds	5.1.1.0	02-09-2012_1044
opswise.server3:8801-opscluster	Passive/Unavailable	2012-02-10 09:15:18 -0500	2012-02-13 11:04:02 -0500	3 Days 1 Hour 48 Minutes 43 Seconds	5.1.1.0	02-09-2012_1044

To see detailed information about a Cluster Node on the list, click the **Node ID** for that Cluster Node. The Cluster Node definition screen for that Cluster Node displays.

See Cluster Nodes for a description of the fields displayed on these screens.

High Availability Configuration

To achieve High Availability for your Opswise Automation Center system, you must configure the Cluster Nodes, OMS, and Agents.

Configuring Cluster Nodes

All cluster nodes in a High Availability environment must point to the same database by making sure the following entries in their `glide.properties` files are the same.

For example:

```
glide.db.name=opswise
glide.db.rdbms=mysql
glide.db.url=jdbc:mysql://10.10.1.1/
```

Configuring OMS

OMS HA cluster configuration is described in the OMS Reference Guide.

The Opswise Controller OMS Server definitions specify an OMS HA cluster as an ordered, comma-separated list of OMS Server addresses, one for each member of the OMS HA cluster.

OMS configuration
Do not define multiple OMS Server records for individual OMS HA cluster members. An OMS HA cluster must be defined as a single OMS Server record with an OMS address list containing each OMS HA cluster member.

As an example, if an OMS HA cluster contains three OMS Servers, `oms1.acme.com`, `oms2.acme.com`, and `oms3.acme.com`, the Opswise Controller OMS Server definition would be defined with an OMS Server address value of `oms1.acme.com,oms2.acme.com,oms3.acme.com`.

Configuring Agents

If you want to configure an Agent to be able to access an OMS HA cluster, you must configure the Universal Automation Center Agent (UAG) OMS_SERVERS configuration option.

Configuring Notifications Based on Component Status

You can configure the Controller to generate Email Notifications or SNMP Notifications based on the status of your Cluster Nodes, OMS Servers, and Agents.

Load Balancer

If you are using a load balancer in your High Availability environment, it can utilize the following HTTP requests:

http://serverhost:8080/opswise/is_active_node.do	If a cluster node is active, this URL returns the status **200** (OK) and a simple one word content of **ACTIVE**. If a cluster node is not active, this URL returns the status **403** (cluster node is not active) and lists the actual mode of the cluster node: **PASSIVE/AVAILABLE**, **PASSIVE/UNAVAILABLE**, or **OFFLINE**.
http://serverhost:8080/opswise/ops_node_info.do	This URL returns information about a cluster node: • Node: serverhost.com:8080-opswise • Release: 5.2.0.6* • Build Id: 03-26-2013_1129 • Mode: Active • Host Name: serverhost.com • Host IP: 192.168.50.50 • Uptime: 7 Days 3 Hours 22 Minutes 37 Seconds • Hub Hostname: localhost • Hub Port: 6776

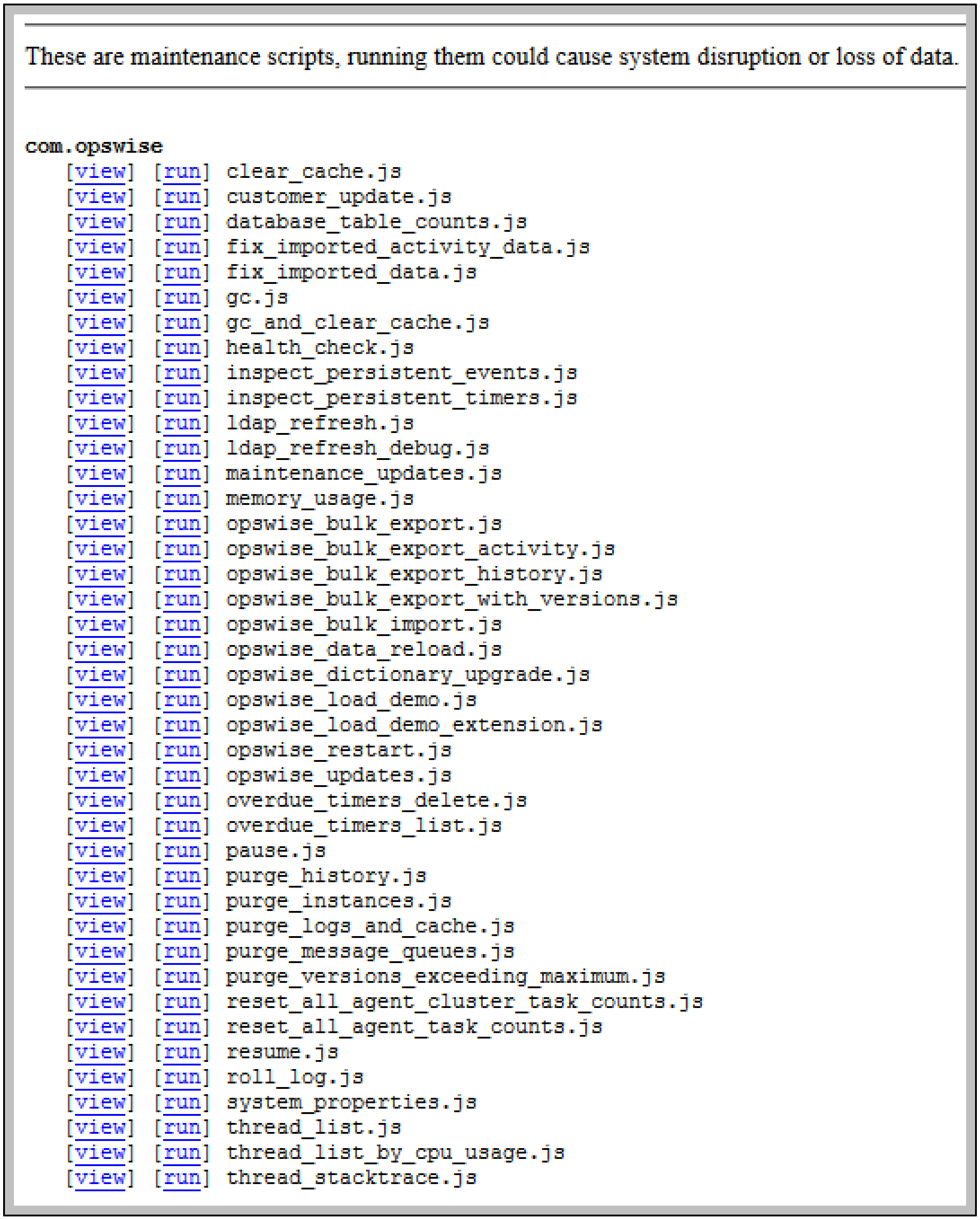
Maintenance Scripts

- Overview
- Maintenance Script Completion
- Maintenance Script Descriptions
- Maintenance Scripts for Windows

Overview

Opswise Controller provides a set of maintenance scripts that help you maintain and administer your Controller installation. Many of the scripts, as noted, should be run only by Technical Support or upon request by Technical Support.

To access the maintenance scripts, select **Automation Center Administration > Configuration > Maintenance Scripts** from the navigation pane.



These are maintenance scripts, running them could cause system disruption or loss of data.

```
com.opswise
    [view]  [run]  clear_cache.js
    [view]  [run]  customer_update.js
    [view]  [run]  database_table_counts.js
    [view]  [run]  fix_imported_activity_data.js
    [view]  [run]  fix_imported_data.js
    [view]  [run]  gc.js
    [view]  [run]  gc_and_clear_cache.js
    [view]  [run]  health_check.js
    [view]  [run]  inspect_persistent_events.js
    [view]  [run]  inspect_persistent_timers.js
    [view]  [run]  ldap_refresh.js
    [view]  [run]  ldap_refresh_debug.js
    [view]  [run]  maintenance_updates.js
    [view]  [run]  memory_usage.js
    [view]  [run]  opswise_bulk_export.js
    [view]  [run]  opswise_bulk_export_activity.js
    [view]  [run]  opswise_bulk_export_history.js
    [view]  [run]  opswise_bulk_export_with_versions.js
    [view]  [run]  opswise_bulk_import.js
    [view]  [run]  opswise_data_reload.js
    [view]  [run]  opswise_dictionary_upgrade.js
    [view]  [run]  opswise_load_demo.js
    [view]  [run]  opswise_load_demo_extension.js
    [view]  [run]  opswise_restart.js
    [view]  [run]  opswise_updates.js
    [view]  [run]  overdue_timers_delete.js
    [view]  [run]  overdue_timers_list.js
    [view]  [run]  pause.js
    [view]  [run]  purge_history.js
    [view]  [run]  purge_instances.js
    [view]  [run]  purge_logs_and_cache.js
    [view]  [run]  purge_message_queues.js
    [view]  [run]  purge_versions_exceeding_maximum.js
    [view]  [run]  reset_all_agent_cluster_task_counts.js
    [view]  [run]  reset_all_agent_task_counts.js
    [view]  [run]  resume.js
    [view]  [run]  roll_log.js
    [view]  [run]  system_properties.js
    [view]  [run]  thread_list.js
    [view]  [run]  thread_list_by_cpu_usage.js
    [view]  [run]  thread_stacktrace.js
```

Maintenance Script Completion

When a maintenance script has been run and completed, the Controller issues an INFO-level log message.

For example:

```
2012-10-18-16:29:22:679   INFO [http-8080-3] Running System Script: script_name.js
```

```
2012-10-18-16:29:23:254   INFO [http-8080-3] System Script completed: script_name.js in N Seconds
```

Maintenance Script Descriptions

The following table describes and, where appropriate, provides links for each maintenance script.

Script	Description and Links
clear_cache.js	Clears the internal server cache. You can use this script if you are experiencing unexpected behaviors with the Controller system. For example, Technical Support may ask you to first run this script to clear the server cache, then clear your browser cache.
customer_update.js	This script is used when applying patches to the Controller system. For use only by Technical Support personnel or when you are requested to run it by Technical Support.
database_table_counts.js	Number of rows in each table.
fix_imported_activity_data.js	This script must be executed if bulk importing Activity data from a 5.1.0.x or earlier version of the Controller.
fix_imported_data.js	Only at the request of Technical Support, this script may need to be executed when importing data from a 5.1.0.x or earlier version of the Controller.
gc_and_clear_cache.js	Runs the clear_cache.js maintenance script, then clears the Controller instance LRU caches, and then runs the gc.js maintenance script.
gc.js	Runs the "garbage collector." The gc method suggests that the Java Virtual Machine expend effort toward recycling unused objects in order to make the memory they currently occupy available for quick reuse. When control returns from the method call, the Java Virtual Machine has made a best effort to reclaim space from all discarded objects.

health_check.js	Displays information about the current instance of the Controller. Example:
	`*** Script:`
	`---------------------------------------------------------------`
	`Using 13.75% of memory. In Use: 125.13 MB Allocated: 510.63 MB.`
	`Free: 785.12 MB. Max Heap: 910.25 MB.`
	`UniverisalOutputMessageQueue: 4 Queue Size, State: DISCONNECTED`
	`UniversalInputMessageProcessor:  State: UNKNOWN`
	`jdbc:mysql://localhost/:root:: In Use: 0, Total: 4`

inspect_persistent_events.js	For use only by Technical Support personnel or when you are requested to run it by Technical Support.
inspect_persistent_timers.js	For use only by Technical Support personnel or when you are requested to run it by Technical Support.
ldap_refresh.js	If LDAP is configured for this installation, it refreshes every 24 hours. This script forces a refresh. The refresh is performed in the background and sends log entries to the Controller log.
ldap_refresh_debug.js	If LDAP is configured for this installation, it refreshes every 24 hours. This script forces a refresh. The refresh writes all log entries to the user interface as well as to the log, and prevents all other user activity while the process is running. If you estimate the refresh could take a considerable amount of time, we recommend you use **ldap_refresh.js**.
maintenance_updates.js	For use only when it is requested to run as part of a maintenance upgrade.

<table>
<tr>
<td>memory_usage.js</td>
<td>

Provides a summary of Controller memory usage. Example:

```
*** Script:
-------------------------------------------------------------
Heap: init = 512MB (524288K) used = 213MB (218961K)
committed = 510MB (523008K) max = 910MB (932096K)
-------------------------------------------------------------
Non Heap: init = 18MB (18688K) used = 60MB (62325K)
committed = 61MB (62560K) max = 560MB (573440K)
-------------------------------------------------------------
Code Cache
Type Non-heap memory
Usage init = 2MB (2304K) used = 9MB (9250K)
committed = 9MB (9312K) max = 48MB (49152K)
Peak Usage init = 2MB (2304K) used = 9MB (9258K)
committed = 9MB (9312K) max = 48MB (49152K)
Collection Usage
-------------------------------------------------------------
PS Eden Space
Type Heap memory
Usage init = 128MB (131072K) used = 88MB (90788K)
committed = 168MB (172288K) max = 339MB (347200K)
Peak Usage init = 128MB (131072K) used = 260MB (267008K)
committed = 260MB (267008K) max = 340MB (348736K)
Collection Usage init = 128MB (131072K) used = 0MB (0K)
committed = 168MB (172288K) max = 339MB (347200K)
-------------------------------------------------------------
PS Survivor Space
Type Heap memory
Usage init = 21MB (21824K) used = 0MB (0K)
committed = 1MB (1152K) max = 1MB (1152K)
Peak Usage init = 21MB (21824K) used = 31MB (32315K)
committed = 40MB (41728K) max = 40MB (41728K)
Collection Usage init = 21MB (21824K) used = 0MB (0K)
committed = 1MB (1152K) max = 1MB (1152K)
-------------------------------------------------------------
PS Old Gen
Type Heap memory
Usage init = 341MB (349568K) used = 125MB (128172K)
committed = 341MB (349568K) max = 682MB (699072K)
Peak Usage init = 341MB (349568K) used = 131MB (134334K)
committed = 341MB (349568K) max = 682MB (699072K)
Collection Usage init = 341MB (349568K) used = 125MB (128172K)
committed = 341MB (349568K) max = 682MB (699072K)
-------------------------------------------------------------
PS Perm Gen
Type Non-heap memory
Usage init = 16MB (16384K) used = 51MB (53098K)
committed = 52MB (53248K) max = 512MB (524288K)
Peak Usage init = 16MB (16384K) used = 51MB (53098K)
committed = 78MB (80000K) max = 512MB (524288K)
Collection Usage init = 16MB (16384K) used = 51MB (53057K)
committed = 52MB (53248K) max = 512MB (524288K)
```

</td>
</tr>
<tr>
<td>opswise_bulk_export.js</td>
<td>Exports all current record definitions, without versions, when migrating data to updated software (see Upgrading Opswise Controller).</td>
</tr>
<tr>
<td>opswise_bulk_export_activity.js</td>
<td>Exports all task instance activity (task instances in the Activity screen) when migrating data to updated software (see Upgrading Opswise Controller).</td>
</tr>
<tr>
<td>opswise_bulk_export_history.js</td>
<td>Exports task instance history (task instances in the History screen), which includes a history of task instance end statuses, including Cancelled, Failed, Skipped, Finished, or Success, when migrating data to updated software. For z/OS, a history of Step Restarted statuses also are included on the History screen (see Upgrading Opswise Controller).</td>
</tr>
<tr>
<td>opswise_bulk_export_with_versions.js</td>
<td>Exports all current records along with older (non-current) versions of record definitions when migrating data to updated software (see Upgrading Opswise Controller).</td>
</tr>
<tr>
<td>opswise_bulk_import.js</td>
<td>Imports all data from an exported file when migrating data to updated software (see Upgrading Opswise Controller).</td>
</tr>
<tr>
<td>opswise_data_reload.js</td>
<td>For use only by Technical Support personnel.</td>
</tr>
<tr>
<td>opswise_dictionary_upgrade.js</td>
<td>For use only by Technical Support personnel.</td>
</tr>
<tr>
<td>opswise_load_demo.js</td>
<td>Loads base demo data into your database, including example workflows, tasks, and so on.</td>
</tr>
</table>

opswise_load_demo_extension.js	Loads additional demo data.
opswise_restart.js	Stops and restarts the Controller within the running Tomcat server. The Controller is effectively shut down and started up without stopping and starting Tomcat.
opswise_updates.js	For use only by Technical Support personnel.
overdue_timers_delete.js	Deletes any overdue timers, as listed by the overdue_timers_list.js script.
overdue_timers_list.js	Lists any timers that are overdue, which normally should be none.
pause.js	Pauses the Controller, which prevents it from processing tasks and events. If an event or task is already running when you run this script, the event or task will complete. The Controller remains in a paused state until you run the resume.js script.
purge_history.js	Purges all records from the History table (ops_history). Any time a task instance goes into an end state (cancelled, failed, skipped, finished, success), a copy of it is written to the History table. You can view your History table by selecting **Task Instances > History** from the navigation pane. If you do not need to keep or back up your task instance history, we recommend that you run this script periodically; otherwise, make sure you schedule a routine Data Backup/Purge for history using Automation Center **Administration-> Data Backup / Purge** from the navigation pane. Executing a Data Backup/Purge of an extremely large history table can seriously degrade Controller system performance.
purge_instances.js	Purges everything in the Activity table (ops_exec), which contains all system activity, including task instances in any status (including end states). Records in the Activity table remain there until they are purged. **Warning** Running this script will purge any live data; that is, task instances that have not completed.
purge_logs_and_cache.js	Sends a request to all active Agents to purge their logs and cache.
purge_message _queues.js	(Deprecated.)
purge_versions_exceeding_maximum.js	Purges versions of records in excess of the maximum specified by the System Default Maximum Versions Opswise Controller system property.
reset_all_agent_cluster_task_counts.js	Resets the current task count for all Agent Clusters to 0.
reset_all_agent_task_counts.js	Resets the current task count for all Agents to 0.
resume.js	Resumes the Controller after it has been paused using the pause.js script.
roll_log.js	Renames the existing log to a timestamped log and opens a new log file.
system_properties.js	Displays all properties of the operating system on which the Controller is running. You may be asked to run this script by Technical Support.
thread_list.js	Captures information about internal Controller system processes. You may be asked to run this script by Technical Support.
thread_list_by_cpu_usage.js	Captures information about internal Controller system processes. You may be asked to run this script by Technical Support.
thread_stacktrace.js	Captures information about internal Controller system processes. You may be asked to run this script by Technical Support.

Maintenance Scripts for Windows

By default, Windows uses a GUI-based VBScript interpreter (`wscript.exe`). However, we recommend you use the console version of the VBScript interpreter (`cscript.exe`). To do so, specify `cscript.exe` before the script name in a task definition, as shown in the following example:

```
"cscript.exe C:\Work\script.vbs".
```

With `wscript.exe`, if your script tries to display an error message that requires a user-response (for example, **Click OK**), you will never see the dialog box. The script therefore gets stuck in the "Running" state.

Ports Configuration

Ports Configuration

Ports configured for Opswise Controller 5.2.0 components and prerequisites cannot be blocked by a firewall.

The following table identifies the default ports, which you can change during installation or configuration:

Component or Prerequisite	Default Port
MySQL	3306
Microsoft SQL Server	1433
Oracle	1521
Opswise Controller (Tomcat)	8080
Opswise Message Service (OMS)	7878

Opswise Controller Properties

- Overview
- Opswise Controller Start-up Properties (glide.properties)
- Opswise Controller System Properties
    - Overriding Opswise Controller System Properties
- Additional Opswise Controller Properties
- Command Line Interface (CLI) Properties

Overview

Opswise Controller contains four types of configurable properties:

Opswise Controller Start-up Properties (glide.properties)	Opswise Controller start-up properties are the default properties contained in the glide.properties file when the Controller is installed. These properties are required for Controller start-up and operation. The values for these properties are set during the installation process. Some of the values are based on information that you provide during the installation. You can reset these properties by stopping the Controller, editing glide.properties, and restarting the Controller. The changes will take effect after the restart (see Starting and Stopping Opswise Controller).
Opswise Controller System Properties	Opswise Controller system properties define Controller system information and performance. They have their values set during installation. You can reset these properties at any time, without having to stop the Controller, via the user interface.
Additional Opswise Controller Properties	Additional Controller properties are feature-specific properties. They have their values set only via the user interface.
Opswise Command Line Interface (CLI) Properties	CLI provides a sample configuration file, **`cmdtools.props`**, that you can use to pass CLI Global parameters to a CLI command. The file is created during installation of Opswise Universal Agent if Opswise Command Line Interface has been selected to be installed.

⚠ **Note**
Properties for Opswise Message Service (OMS) are installed as configuration file options when OMS is installed as a component of Opswise Universal Agent. The values for these options are set during the installation. There are several configuration methods for changing these values.

If you are using an Opswise Automation Center 5.1.0 Outboard (Message Hub and Transporter) as the network communications provider between a 5.2.0 Controller and pre-5.2.0 Agents, see Message Hub Properties and Transporter Properties in Opswise Automation Center 5.1.1.

Opswise Controller Start-up Properties (glide.properties)

The **`glide.properties`** file is read by the Controller, which is started by Tomcat.

The **`glide.properties`** file resides here:

```
[tomcat directory]\webapps\opswise\WEB-INF\properties
```

Property Name	Description	Default
(For MySQL)		
`glide.db.rdbms=mysql`	Database type. Specify this property if you are using a MySQL database.	
`glide.db.url=jdbc:mysql://localhost/`	JDBC connect URL. Specify this property if you are using a MySQL database.	
(For SQLServer)		
`glide.db.rdbms=sqlserver`	Database type. Specify this property if you are using a SQLServer database.	
`glide.db.url=jdbc:sqlserver://localhost:1433;DatabaseName=opswise`	JDBC connect URL. Specify this property if you are using a SQLServer database.	
(For LDAP)		
`glide.ldap.groups.filter_indirect=`	When this property is set to true, any Groups synchronized indirectly (that is, through a User's **memberOf** attribute) will honor the Group search filter and Group OU filters under the LDAP Advanced Settings section. ⚠ **Note** The code default for this property, which is used if this property is not set, is false.	true
`glide.ldap.groups.single_parent_per_child=`	⛔ **IMPORTANT** This property should only be set to true if your Groups being synchronized from AD have at most one parent Group. When synchronizing Groups, the default behavior in the Controller is to copy the members of a Sub Group into the Parent Group. When this property is set to true, the Controller assumes that each Group has, at most, a single Parent Group and will use the Parent field on the Group definition to maintain the hierarchy instead of copying members.	false
(For all databases)		
`glide.db.user=`	Login ID that the Controller will use to log in to your database.	root

`glide.db.password.encrypted=`	Encrypted version of **`glide.db.password`** that will replace **`glide.db.password`** in the **`glide.properties`** file upon start-up. To change this password: 1. Delete (or comment out) **`glide.db.password.encrypted`** in the **`glide.properties`** file. 2. Manually reenter **`glide.db.password`**, including a new password, in the **`glide.properties`** file. 3. Stop and restart the Controller to replace **`glide.db.password`** with **`glide.db.password.encrypted`** in the **`glide.properties`** file.	(none)
`glide.db.name=`	Name for the Controller database.	opswise
`glide.sys.boot_script=`	Script used to initialize the database. **Internal use only; do not change.**	zboot_opswise.js
`glide.db.pooler.connections=`	Minimum number of connections that can remain idle in the pool without extra connections being created, or zero to create none.	25
`glide.db.pooler.connections.max=`	Maximum number of active connections that can be allocated from this pool at the same time, or negative for no limit.	100
`glide.servlet.port=`	Port number used by Tomcat.	8080
`glide.ui.session_timeout=`	Default browser session timeout, in minutes. To use the Tomcat session configuration (default 30 minutes), set this property to 0.	60
`glide.xmlhttp.excessive=`	Default maximum number of items included for selection in the left side of the slushbucket (for example).	200
`opswise.hub.host=`	Hostname or IP address of the machine where the Message Hub resides. **Note** This property (and `opswise.hub.port=`) appears in the **`glide.properties`** file only if the --hubhost command line switch was specified at Controller installation for backward-compatibility with pre-5.2.0 Agents.	

`opswise.hub.port=`	Port used for Message Hub communications. ⚠ **Note** This property (and `opswise.hub.host=`) appears in the **`glide.properties`** file only if the --hubport command line switch was specified at Controller installation for backward-compatibility with pre-5.2.0 Agents.	
`opswise.trustmanager.algorithm=`	Java trust manager algorithm. For IBM AIX, value must be IbmX509. The default works for all other platforms.	SunX509
`opswise.trustmanager.provider=`	Java trust manager provider. For IBM AIX, value must be IBMJSSE2. The default works for all other platforms.	SunJSSE
`opswise.trustmanager.truststore=`	Location of the keystore which holds certificates and keys.	
`opswise.trustmanager.truststore.password=`	Password (if required) for the keystore.	

Sample glide.properties File

```
# DB
glide.db.rdbms=mysql
glide.db.url=jdbc:mysql://localhost/
# MYSQL
# glide.db.rdbms=mysql
# glide.db.url=jdbc:mysql://localhost/
# MS SQLSERVER
# glide.db.rdbms=sqlserver
# glide.db.url=jdbc:sqlserver://localhost:1433;DatabaseName=opswise
# ORACLE
# glide.db.rdbms=oracle
# glide.db.url=jdbc:oracle:thin:@//localhost:1521/@oracle.db.name@
#
# COMMON
#
# trust manager algorithm & provider
#opswise.trustmanager.algorithm=SunX509
#opswise.trustmanager.provider=SunJSSE
#
glide.db.user=root
glide.db.password=pswd
glide.db.name=opswise
glide.sys.boot_script=zboot_opswise.js
glide.db.pooler.connections=2
glide.db.pooler.connections.max=40
glide.servlet.port=8080
glide.ui.session_timeout=60
glide.xmlhttp.excessive=200
```

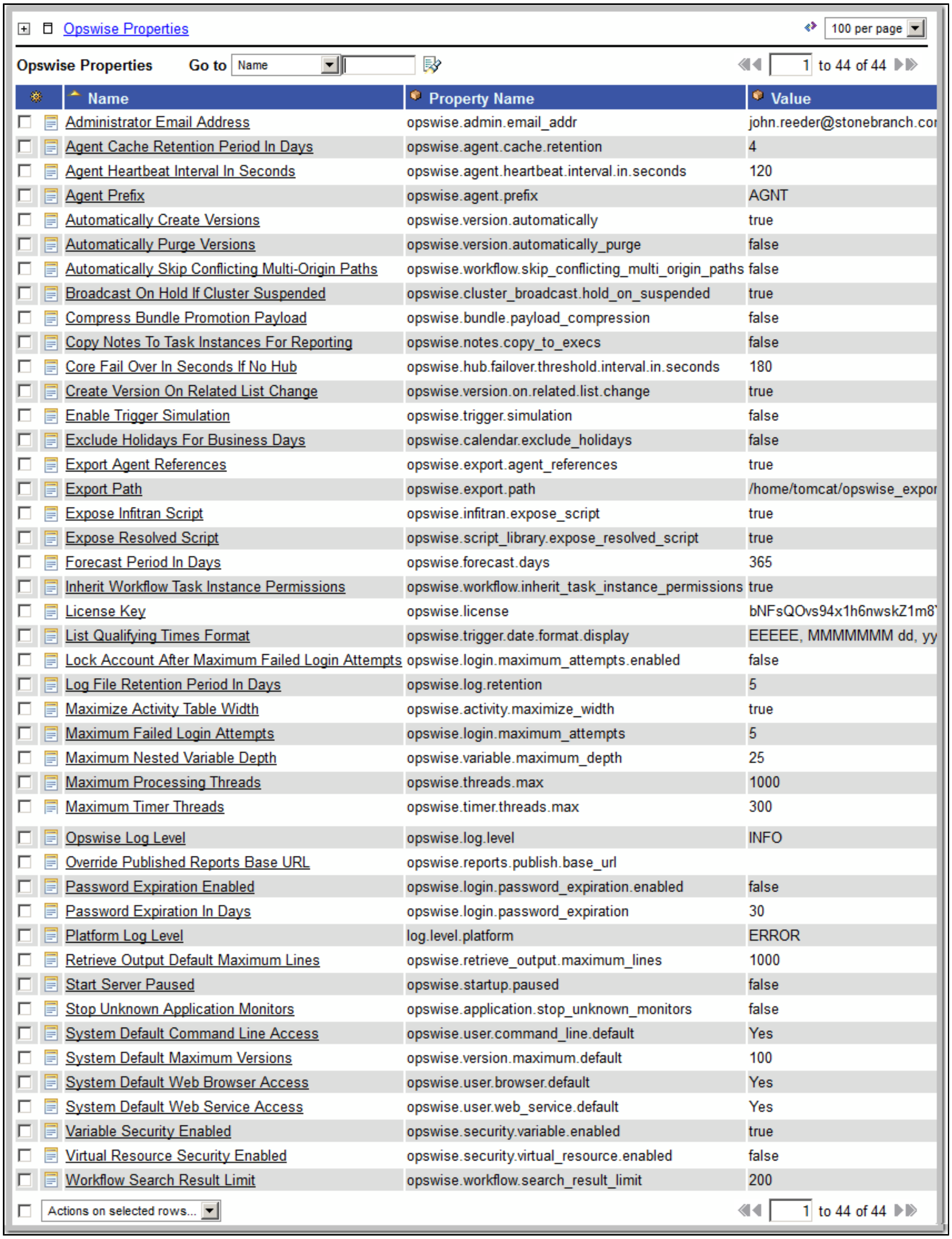
Opswise Controller System Properties

Properties for your Opswise Controller system are set (in the Controller database) during Controller installation. These properties let you define Controller system information and performance.

Opswise Controller system properties do not reside in a properties file; they are available only via the user interface. You can reset these properties any time after the Controller is in operation without having to stop and restart the Controller.

⚠ **Note**
You must be assigned the ops_admin role in order to reset these properties.

Step 1 Select **Automation Center Administration > Configuration > Properties** from the navigation pane. The Opswise Properties screen displays.



Name	Property Name	Value
Administrator Email Address	opswise.admin.email_addr	john.reeder@stonebranch.cor
Agent Cache Retention Period In Days	opswise.agent.cache.retention	4
Agent Heartbeat Interval In Seconds	opswise.agent.heartbeat.interval.in.seconds	120
Agent Prefix	opswise.agent.prefix	AGNT
Automatically Create Versions	opswise.version.automatically	true
Automatically Purge Versions	opswise.version.automatically_purge	false
Automatically Skip Conflicting Multi-Origin Paths	opswise.workflow.skip_conflicting_multi_origin_paths	false
Broadcast On Hold If Cluster Suspended	opswise.cluster_broadcast.hold_on_suspended	true
Compress Bundle Promotion Payload	opswise.bundle.payload_compression	false
Copy Notes To Task Instances For Reporting	opswise.notes.copy_to_execs	false
Core Fail Over In Seconds If No Hub	opswise.hub.failover.threshold.interval.in.seconds	180
Create Version On Related List Change	opswise.version.on.related.list.change	true
Enable Trigger Simulation	opswise.trigger.simulation	false
Exclude Holidays For Business Days	opswise.calendar.exclude_holidays	false
Export Agent References	opswise.export.agent_references	true
Export Path	opswise.export.path	/home/tomcat/opswise_expor
Expose Infitran Script	opswise.infitran.expose_script	true
Expose Resolved Script	opswise.script_library.expose_resolved_script	true
Forecast Period In Days	opswise.forecast.days	365
Inherit Workflow Task Instance Permissions	opswise.workflow.inherit_task_instance_permissions	true
License Key	opswise.license	bNFsQOvs94x1h6nwskZ1m8
List Qualifying Times Format	opswise.trigger.date.format.display	EEEEE, MMMMMMM dd, yy
Lock Account After Maximum Failed Login Attempts	opswise.login.maximum_attempts.enabled	false
Log File Retention Period In Days	opswise.log.retention	5
Maximize Activity Table Width	opswise.activity.maximize_width	true
Maximum Failed Login Attempts	opswise.login.maximum_attempts	5
Maximum Nested Variable Depth	opswise.variable.maximum_depth	25
Maximum Processing Threads	opswise.threads.max	1000
Maximum Timer Threads	opswise.timer.threads.max	300
Opswise Log Level	opswise.log.level	INFO
Override Published Reports Base URL	opswise.reports.publish.base_url	
Password Expiration Enabled	opswise.login.password_expiration.enabled	false
Password Expiration In Days	opswise.login.password_expiration	30
Platform Log Level	log.level.platform	ERROR
Retrieve Output Default Maximum Lines	opswise.retrieve_output.maximum_lines	1000
Start Server Paused	opswise.startup.paused	false
Stop Unknown Application Monitors	opswise.application.stop_unknown_monitors	false
System Default Command Line Access	opswise.user.command_line.default	Yes
System Default Maximum Versions	opswise.version.maximum.default	100
System Default Web Browser Access	opswise.user.browser.default	Yes
System Default Web Service Access	opswise.user.web_service.default	Yes
Variable Security Enabled	opswise.security.variable.enabled	true
Virtual Resource Security Enabled	opswise.security.virtual_resource.enabled	false
Workflow Search Result Limit	opswise.workflow.search_result_limit	200

Step 2 Click a property name to open the record for the property you want to change.

The following table describes the Opswise Controller system properties:

Name (Property Name)	Description	Default

Administrator Email Address (opswise.admin.email_addr)	System administrator email address(es) specified as the recipient(s) for System Notifications. Addresses for multiple administrators should be specified in a comma-separated list.	(none)
Agent Cache Retention Period in Days (opswise.agent.cache.retention)	Number of days that cache files (stdout, stderr) are retained by the system.	7
Agent Heartbeat Interval in Seconds (opswise.agent.heartbeat.interval.in.seconds)	Number of seconds between each heartbeat message sent by the agent to the Controller.	120
Agent Prefix (opswise.agent.prefix)	Prefix appended to the Queue name for newly registered agents. A 4-digit number is appended to this prefix.	AGNT
Automatically Create Versions (opswise.version.automatically)	Specification (true or false) for whether or not the Controller will retain copies of previous versions. Affects system behavior when you make updates to records in your Controller database, such as changing a task definition.	true
Automatically Purge Versions (opswise.version.automatically_purge)	Specification (true or false) for whether or not the Controller will automatically purge versions of records that exceed the maximum number of version records to maintain per definition as defined by the System Default Maximum Versions Controller system property. Changing the value from **false** (the default) to **true** will automatically invoke a system-wide purge of versions for definitions that exceed the maximum number of versions specified by System Default Maximum Versions. Also if the value is **true**: • Any modification of a definition that generates a new version will automatically invoke a purge of the oldest version of that definition if the number of versions exceeds the configured maximum. • Any modification of System Default Maximum Versions that decreases the maximum number of versions to maintain will invoke a system-wide purge of versions for definitions that exceed the new maximum number of versions.	false
Automatically Skip Conflicting Multi-Origin Paths (opswise.workflow.skip_conflicting_multi_origin_paths)	Specification (true or false) for whether or not the Controller will automatically skip a task (within a workflow) that is connected to multiple upstream tasks, where one or more of the upstream tasks would cause the task to be run and one or more would cause the task to be skipped.	false
Broadcast On Hold If Cluster Suspended (opswise.cluster_broadcast.hold_on_suspended)	Specification (true or false) for whether or not cluster broadcast tasks will be run if the agent cluster selected for the broadcast has been suspended.	true
Compress Bundle Promotion Payload (opswise.bundle.payload_compression)	Specification (true or false) for whether or not the Controller will compress record bundles during a promotion.	false

Copy Notes to Task Instances for Reporting (opswise.notes.copy_to_execs)	Specification (true or false) for whether or not the Controller will copy task notes to task instances so that task notes can be included in activity reports or gauges. For example, if true is specified, you can create a gauge that lists task notes for failed task instances. This property should be enabled only as needed.	false
Create Version On Related List Change (opswise.version.on.related.list.change)	Specification (true or false) for whether or not a record version will be created if the user changes a record associated with the current record. For example, if true, the system will create a version of the task when the user changes a task variable.	true
Enable Trigger Simulation (opswise.trigger.simulation)	Specification (true or false) for whether or not to simulate the launching of tasks when triggers are eligible to fire. If simulation is enabled, only the scheduled launch of the task by the trigger is inhibited. All other aspects of the trigger execution, including generation of forecast data, are enabled. You can still force a trigger by using the Trigger Now command or launch a task by using the Launch command.	false
Exclude Holidays for Business Days (opswise.calendar.exclude_holidays)	Specification (true or false) for whether or not the Controller will consider a Business Day on which a holiday falls as a non-Business Day. • If true, holidays that fall on Business Days are considered non-Business Days. • If false (the default), holidays that fall on Business Days are considered Business Days. For example, if the default value (false) is used, and a job is defined to run on Business Days, the job will run on Christmas Day, even though it is a holiday. This behavior applies to Triggers, Task Run Criteria, and JavaScript functions that operate on Business Days, and provides a means to avoid having to specify a restriction or skip criteria for holidays.	false
Export Agent References (opswise.export.agent_references)	Specification (true or false) for whether or not the Controller will export referenced Agents when exporting definition XMLs with the Export References feature.	false

Export Path (opswise.export.path)	Pathname where exported xml files are written.
	This applies only when using the Export ->XML feature, not the bulk export.
		`<tomcat-home>/opswise_export`
		or
		`<tomcat-home>\opswise_export`
		(`<tomcat-home>` is the base tomcat installation directory.)

Expose Infitran Script (opswise.infitran.expose_script)	***For debugging use only.*** Specification (true or false) for whether or not the Controller prepares a script when it launches a file transfer on an Infitran installation. If troubleshooting is necessary, enabling this property allows you to view the script in the Output tab on the task instance.	false

Expose Resolved Script (opswise.script_library.expose_resolved_script)	Specification (true or false) for whether or not to generate a SCRIPT output type capturing the resolved contents of the Script Library script for each task instance run attempt that utilizes a script from the Script Library. This property only applies to Script Library scripts defined with the **Resolve Opswise Variables** option checked. Any user with the task instance Read permission for a specific task instance will be able to view the SCRIPT output type content for that specific instance. To avoid generating unnecessary output, we recommend enabling this property only for debugging purposes. The unresolved script content can always be viewed from the Script Library.	false
Forecast Period in Days (opswise.forecast.days)	Number of days to be included in a trigger forecast. See Displaying Trigger Forecast Information.	31
Heartbeat Processing Interval in Seconds (opswise.hb.processing.interval.in.seconds)	***Deprecated.*** Interval (in seconds) at which the various watchdogs inspect the heartbeats of attached components.	10
Inherit Workflow Task Instance Permissions (opswise.workflow.inherit_task_instance_permissions)	Specification (true or false) for whether or not a user that has access to a workflow task instance will automatically have access to the tasks belonging to that workflow task instance. If this property is set to **false**, workflow task instance permissions will be inherited only when running commands. For example, if you can Force Finish a workflow task instance, then you still will be able to Force Finish task instances within that workflow task instance. However, you will not be able to access task instances within a workflow task instance unless you have explicit permission to do so.	true
Input Queue Processing Interval in Seconds (opswise.iq.processing.interval.in.seconds)	***Deprecated.*** Number of seconds that are used to poll the incoming message queues.	5
License Key (opswise.license)	License key for your installation; provided to you by your Opswise Controller representative.	(none)
List Qualifying Times Format (opswise.trigger.date.format.display)	Format that you want the Controller to use when listing qualifying times for Time and Cron Triggers. See List Qualifying Times.	EEEEE, MMMMMMM dd, yyyy HH:mm:ss z Z
Lock Account After Maximum Failed Login Attempts (opswise.login.maximum_attempts.enabled)	Specification (true or false) for whether or not to lock a user account if the user has reached the maximum number of successive failed login attempts that is allowed, as specified by the Maximum Failed Login Attempts property. Whenever this property is enabled (value is changed from **false** to **true**), the current number of failed login attempts for all users is reset to 0.	false
Log File Retention Period in Days (opswise.log.retention)	Number of days that the Controller retains its log files.	5

Maximize Activity Table Width (opswise.activity.maximize_width)	Specification (true or false) for whether or not the Controller will maximize the width of the Activity table. ⚠ **Note** The default value (true) applies only to new deployments of the Controller. Applying maintenance to the Controller (5.1.1.10 or later) will not change the current value.	true
Maximum Failed Login Attempts (opswise.login.maximum_attempts)	Maximum number of successive failed login attempts that a user can make before the user's account is locked if the Lock Account After Maximum Failed Login Attempts property is set to **true**.	5
Maximum Nested Variable Depth (opswise.variable.maximum_depth)	Maximum number of nested variables allowed.	25
Maximum Processing Threads (opswise.threads.max)	Maximum number of processing threads used.	1000
Maximum Timer Threads (opswise.timer.threads.max)	Maximum number of timer threads used.	300
Opswise Log Level (opswise.log.level)	Level of logging for the Controller: • ALL • TRACE • DEBUG • INFO • WARN • ERROR • SEVERE • OFF	INFO
Override Published Reports Base URL (opswise.reports.publish.base_url)	Allows full control over the Base URL. For example, you can set this property to a value of the format: `http[s]://hostname[:port]/`	(none)
Password Expiration Enabled (opswise.login.password_expiration.enabled)	Specification (true or false) for whether or not user passwords will expire after the maximum number of days that a user password can remain unchanged before expiring, as specified by the Password Expiration in Days property. ⚠ **Note** Password expiration is not applicable to LDAP authenticated users.	false
Password Expiration in Days (opswise.login.password_expiration)	Maximum number of days that a user password can remain unchanged before expiring, if the Password Expiration Enabled property value is **true**.	30

Platform Log Level (log.level.platform)	Level of logging for the platform where the Controller operates: • ALL • TRACE • DEBUG • INFO • WARN • ERROR • OFF	WARN
Retrieve Output Default Maximum Lines (opswise.retrieve_output.maximum_lines)	Limit for the number of lines retrieved when Automatic Output Retrieval is enabled on a task.	100
Server Fail Over in Seconds If No Message Hub (opswise.hub.failover.threshold.interval.in.seconds)	In a high availability configuration, if the communications between the Controller and the Message Hub (MsgHub) are disrupted, the number of seconds that Controller waits before bringing itself down and allowing the backup Controller server to take over.	120
Start Server Paused (opswise.startup.paused)	Specification (true or false) for whether or not the start server process brings up the server in paused mode.	false
Stop Unknown Application Monitors (opswise.application.stop_unknown_monitors)	Specification (true or false) for whether or not to stop any application monitors currently running on an Agent if the Controller is no longer managing those monitors (Windows and Linux/Unix only).	false
System Default Command Line Access (opswise.user.command_line.default)	Specification (Yes or No), for all users whose **Command Line access** field on their User Definition screen is set to **-- System Default --**, for whether or not to control a user's ability to access the Controller through the Command Line Interface (CLI).	Yes
System Default Maximum Versions (opswise.version.maximum.default)	Maximum number of version records (1 to 255) to maintain per definition.	100
System Default Web Browser Access (opswise.user.browser.default)	Specification (Yes or No), for all users whose **Web Browser access** field on their User Definition screen is set to **-- System Default --**, for whether or not to control a user's ability to access the Controller through the user interface.	Yes
System Default Web Service Access (opswise.user.web_service.default)	Specification (Yes or No), for all users whose **Web Service access** field on their User Definition screen is set to **-- System Default --**, for whether or not to control a user's ability to access the Controller through the RESTful Web Services API.	Yes
Variable Security Enabled (opswise.security.variable.enabled)	Specification (true or false) for enabling enhanced Global Variable security.	false
Virtual Resource Security Enabled (opswise.security.virtual_resource.enabled)	Specification (true or false) for enabling enhanced Virtual Resource security.	false
Workflow Search Result Limit (opswise.workflow.search_result_limit)	Results limit when querying for task records from the Task Find or Open Workflow pop-up.	200

Overriding Opswise Controller System Properties

You can override any Opswise Controller system property by adding it to the Opswise Controller Start-up Properties (glide.properties) file and restarting the Controller.

Any Opswise Controller system property added to **`glide.properties`** must be in the same format as the **`glide.properties`** properties: **`<Property Name>=<value>`**. For example: **`opswise.startup.paused=true`**

When the restarted Controller reads the **`glide.properties`** file, it updates the database with the value of any Opswise Controller system property included in the file. It then removes that property from the file.

Additional Opswise Controller Properties

The following additional Controller properties are available through the user interface:

LDAP Properties	Enable the LDAP bridge for both UNIX and Windows operating systems.
User Interface Properties	Select the number of items that appear on any list from the drop-down menu that displays in the top right corner of the page.
Report Email Properties	Set up the email server that you want to use for scheduling automatic report distribution.

Command Line Interface (CLI) Properties

A sample Command Line Interface (CLI) configuration file, **`cmdtools.props`**, created during the installation of the CLI (an optionally installed component of Opswise Universal Agent), is provided for your use. However, you can create a configuration file with any name; it must exist in the directory from where you are issuing the commands. (See Command Line Interface (CLI)).