

BOM Verification User Guide

All rights reserved. This document contains proprietary information of Kronos, Inc. and is subject to a license agreement or nondisclosure agreement. No part of this document may be reproduced, transmitted, transcribed, stored in a retrieval system, or translated into another language, in any form or by any means, without the prior written consent of Kronos, Inc.

Contents

[BOM Verification User Guide 1](#_Toc289699284)

[1 Overview 3](#_Toc289699285)

[2 First Phase 3](#_Toc289699286)

[3 Requirements and Goals 4](#_Toc289699287)

[4 Implementation Details 5](#_Toc289699288)

[5 Bamboo Build Plans 6](#_Toc289699289)

[6 VM Configuration 10](#_Toc289699290)

[7 Creating a New Environment 12](#_Toc289699291)

[8 Process Ownership 14](#_Toc289699292)

[9 Enhancements 14](#_Toc289699293)

# Overview

In an effort to improve the speed and quality of 8x software releases, it has become a part of the HSG Release Engineering (RE) team’s charter to make sure development teams:

1. Spend less time troubleshooting builds and deployments
2. Help dev teams to verify their code changes on a real application server.

In an effort to improve the reliability and success of BOM (Bill of Materials) builds, RE will be working with IT to create several virtual machines (VMs) that will allow for rapid and continuous installations. Builds will be deployed and restored periodically in order to provide feedback to developers that the latest code changes do not disrupt QA and TestApp environments.

For more complete understanding of the BOM build process see: [All About The BOM](https://confluence.aspdeploy.com/display/aTAODev/All+About+the+BOM)

Also, for info on the initiation of this request see [JIRA 258](https://jira.aspdeploy.com/browse/RE-258)

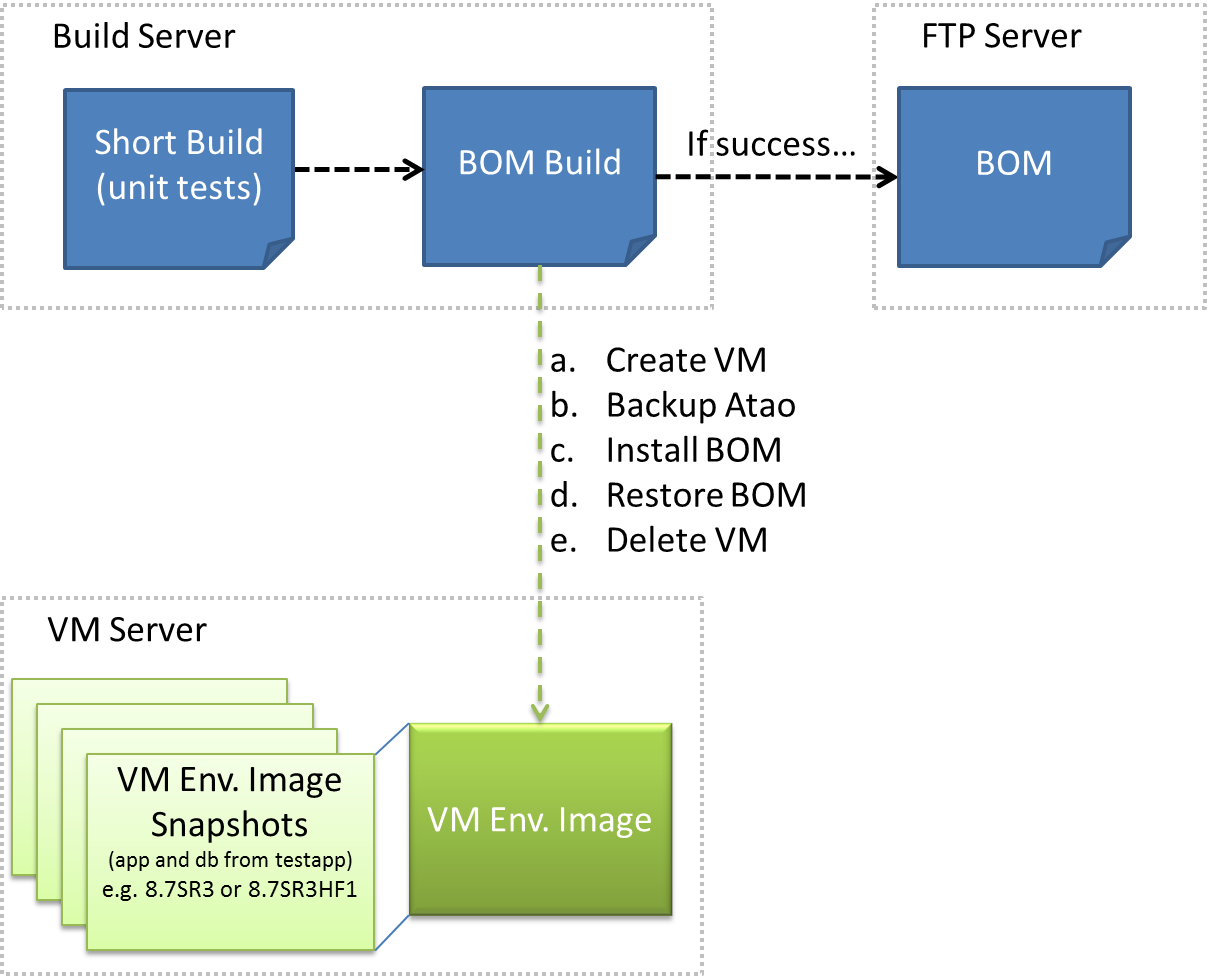
# First Phase

The first phase of this project will allow the Bamboo (automated build application) to follow these steps:

1. Create snapshot of current version of VM app and db server
2. Use BOM to install on target VM server
3. Backup, Install and Restore with BOM build commands via Ant
4. After restore or failure, delete VMs to be ready for next verify test

Whether the backup, installation or restore succeeds or fails, the original VM image(s) will be reloaded and a new build can be tested on the next round. Once a BOM is gold and goes live, a build will be run manually so that the VMs can be updated with the production version BOM and for the new VM baseline to be set.

Here is a diagram showing the basic project workflow for success and failure path:



# Requirements and Goals

There are two primary goals for this project:

1. Identify build and install failures prior to request for official build.
2. Allow the ability to maintain a useable and reliable environment.

The basic list of process requirements, with CE as the customer, is as follows:

1. Automate BOM creation (continuous)
2. Automate BOM installation
3. Automate BOM restore
4. Identify build and install failures (early and often)
5. Generate pristine install for QA
6. Access to VMware tools for manual and automatic restores

The basic list of hardware requirements

1. 2 -3 VM environments where one app and one db server make up and environment.
   1. SR environment
   2. HF environment
   3. Sandbox environment to troubleshoot failed installs
2. 500 GB of available storage
3. Average RAM and CPU configuration

# Implementation Details

This section is to help describe in more detail how this system will work. It will be primarily Ant scripts and a bit of VMware configuration.

Setting up initial images

A powerful benefit of working in a virtualized environment is how “cheap” it is to create and restore images at will. If configured correctly, snapshots can be created and restored quickly and maintained with a minimal amount of storage. Initially, the base snapshots will be from Continuous Engineering (CE) environment’s QA app and DB server. Those images will be the basis for installing and testing BOMs. They represent a “pristine” testable environment.

Automated build

A BOM is created for major and minor release as well as hotfixes. BOMs are created using a build plan managed by Bamboo build software. The goal will be to leverage the current build plan and deploy the build to the VM.

Automated install and restore

Since the Bamboo build plan triggers an Ant script that runs on some defined schedule, it can be enhanced to include a property to define which VM to install to. The current image will likely be named in accordance to the current version of the atao software. It could be something like: VMImage\_8.7SR\_SR10. As the version gets incremented, so does the VM name. At first, the property that defines the server name may have to be updated manually, but the goal will be to make it completely automated to save time and become more hands off.

Once the BOM has been installed successfully, a user could SSH into the system if needed and turn on Atao if so desired. The VM will be restored, which should leave the VM in the “pre-installed” state the next time the build runs. If the backup, install or restore fails, it means that the BOM has code in it that is not compatible with the current atao version and needs to be addressed by a developer. The Bamboo verification plan should email a subscriber list with the results and the user should run the “Restore VM” to set the VMs back into a known, working state.

Update snapshot

Once the gold (the last good install after code is frozen) build has been deployed and installed successfully to the VM, that version will become the new VM snapshot image that will be used to test the next round of installs for the new atao version.

# Bamboo Build Plans

Bamboo build plans control the ability to execute the VM management as well as control the BOM installation steps. The only manual intervention required in these plans is changing the SR or HF numbers so that the build pulls down the correct BOM to test with.

Routine BOM testing

Manage Current SR VM

* VM support plan : Restore VM
* Bom Install plan : Repeat BOM Install

The Bamboo plan is setup to run manually or periodically in order to exercise the latest BOM version against the current BOM version. An example would be applying build 65123 of BOM SR 12 on top of current 64234 of BOM SR 11.1. If there is a failure someone will have to manually restore the VM to the BOM state.

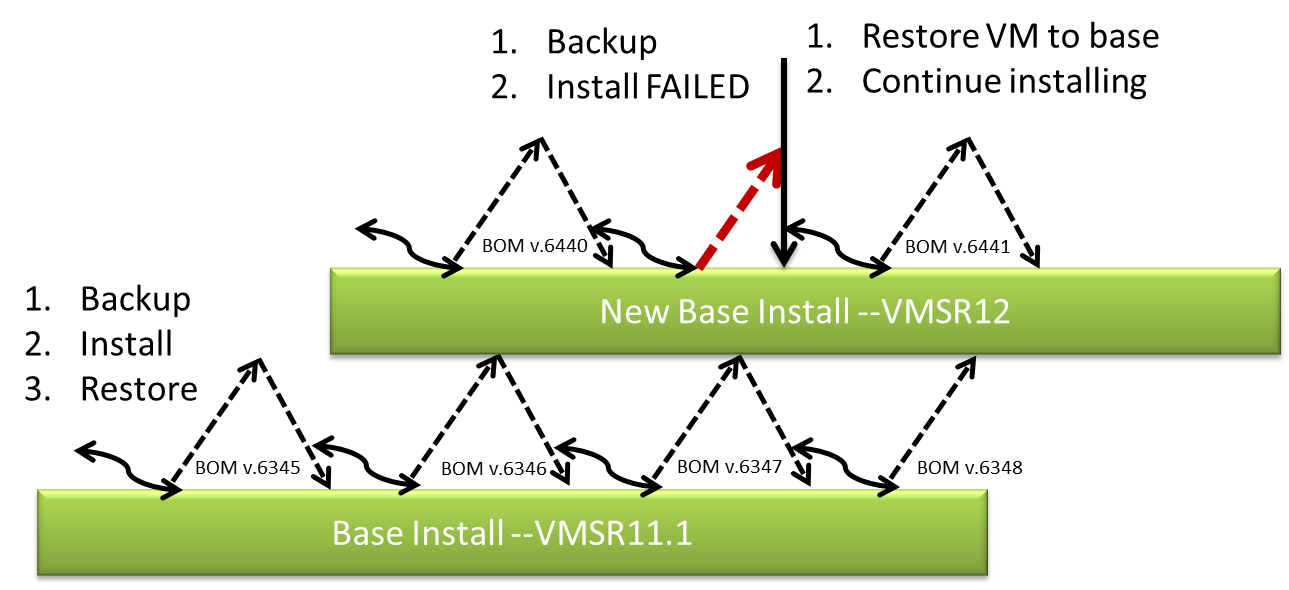
Move to New Version

Upgrade Current SR VM

* VM plan : Save VM Version
* Dependent plan : Update Install Version

This plan will perform the upgrade to the new SR or HF version and leave it in a state to run future build plans on this new baseline VM.

Here is a diagram of how this process works visually, per SR and HF.



Updating Build Properties

Can find the buildTime.properties file at *[SVN : svn+ssh://username@svn.deploy.com/svnroot/repository6/trunk/ataoTools/verification]*

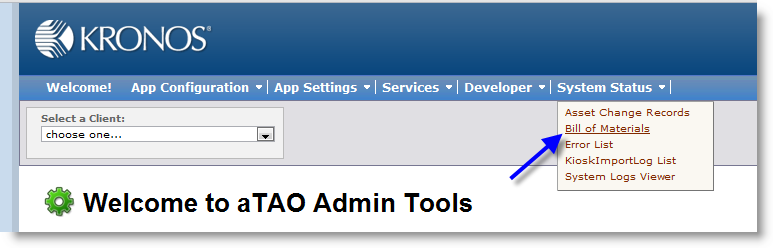
Change these values when incrementing build versions.

|  |  |  |  |
| --- | --- | --- | --- |
| Purpose | Ant Property | Current Value | New Value |
| New VM Snapshot name | new.sr.number | SR11.1 | SR12 |
| Previous Installed SR\HF Version | old.sr.number | 11 | 11.1 |
| Major Release Version | major.release.number | 8.8 | 8.8 |
| Complete Version number moving to | current.release.number | 8.8.0.1 | 8.0.0.0 |
| Major Release Version | base.vm.name | 8.8 | 8.8 |
| Allows to bypass, dev ftp site | is.dev.ftp | True\false | False |
| Set it to a network BOM repo | Manual.BOM.path | Ex. /exports/rebuild/bamboo2/xml-data/build-dir/V88SR1-BOM-JOB1/checkout/atao/relWorkArea |  |
| For manually choosing the bom version number | Is.manual.BOM | True\false | false |
| Specific version | Manual.BOM.version | Ex. 70309.final8.8.0.0 |  |
| Specific zip name if different than bom version folder name | Manual.BOM.zip.version | 70309 |  |
|  |  |  |  |

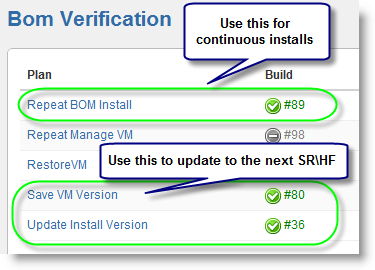
Upgrade Build Example With Step By Step Instructions

NOTE: Although the build configurations can be overwritten in Bamboo build plans, the preferred method is by updating the buildTime.properties file.

The following will give an example of upgrading a VM version. In this example, the current state of the VMs is at SR11.1 build number 69295 and this environment will be updated to SR12 build number 69735. *NOTE: The current version should be verified on the environment by starting the application on the server and checking out the current installed version which can be found on the admin tools page.*



1. Go to KAP-US-BAMBOO2 [http://kap-us-bamboo2:8085] Browse to the BOM Verification build project.



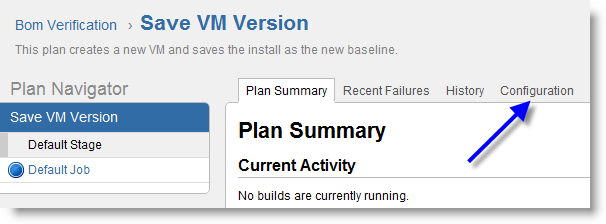
* 1. Update Install Version will upgrade the VM snapshot to the latest baseline
  2. Save VM version will be called automatically when the installation finishes successfully.
  3. Repeat BOM install will be configured to run twice a day to automatically apply and remove the latest BOM version.

1. Since a new baseline is being created (going from 11.1 to 12), the **Save VM Version** and **Update Install Version** plans need to be updated. The values that need to be changed can be found by clicking on the build plan.
   1. Save VM Version creates a new VM snapshot that will be used as the new point in time for the VM.
   2. Update Install Version will actually get the latest BOM for the request version and install it.

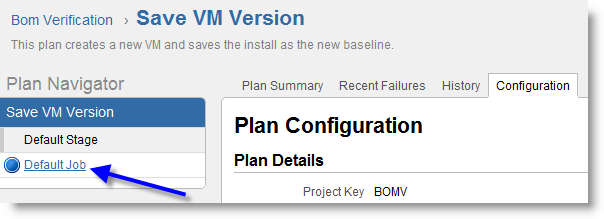
There are several values that need to be changed in order to have the server be updated.

Updating “Save VM Version”

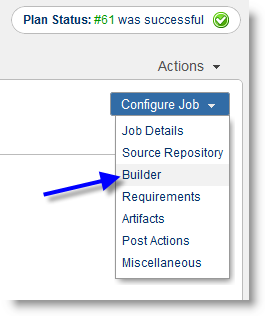
1. Click on **Save VM Version**
2. Click on “Configuration” Tab



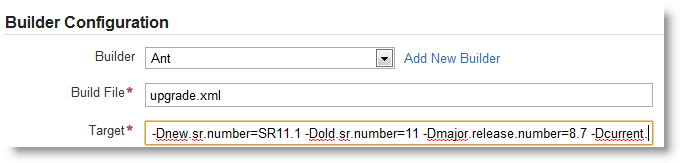
1. Click on “Default Job”



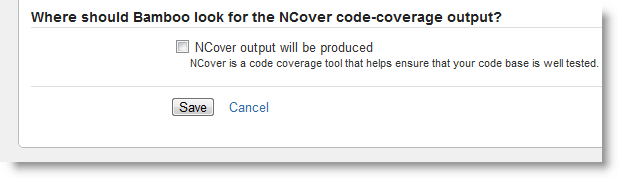
1. Under Actions on the far right, under “Configure Job” drop down, select “Builder”



1. Update the build properties in the “Target” config string



1. Make sure to **Save** after changing the version numbers in “Target”



Properties that need to be changed are…

|  |  |  |  |
| --- | --- | --- | --- |
| **Purpose** | **Ant Property** | **Current Value** | **New Value** |
| New VM Snapshot name | -Dnew.sr.number | SR11.1 | SR12 |
| Previous Installed SR\HF Version | -Dold.sr.number | 11 | 11.1 |
| Major Release Version | -Dmajor.release.number | 8.7 | 8.7 |
| Complete Version number moving to | -Dcurrent.release.number | 8.7.11.1 | 8.7.12.0 |

Updating “Update Install Version”

This process is just like the Save VM update, only this time click on **Update Install Version**. Here are values in the Target configuration that need to be changed…

|  |  |  |  |
| --- | --- | --- | --- |
| **Purpose** | **Ant Property** | **Current Value** | **New Value** |
| New VM Snapshot name | -Dnew.sr.number | SR11.1 | SR12 |
| Previous Installed SR\HF Version | -Dold.sr.number | 11 | 11.1 |
| Major Release Version | -Dmajor.release.number | 8.7 | 8.7 |
| Complete Version number moving to | -Dcurrent.release.number | 8.7.11.1 | 8.7.12.0 |

Failure Condition

In the case where there is a failure, the bamboo logs will need to be reviewed and the VM version reset. It can be reset using the Utility Plan “DeleteVM”.

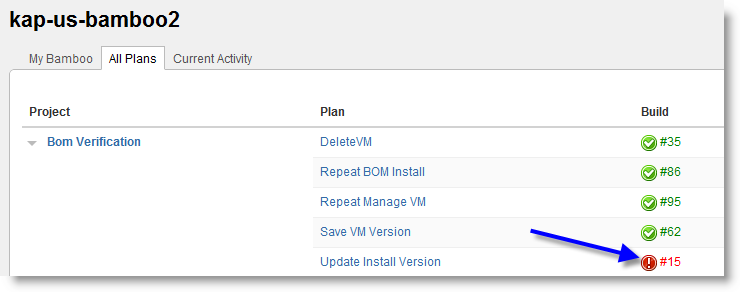
Reset the current snapshot to earlier version. Be sure to set the correct values in the “Target” config box. This plan needs to be run when the Backup, Install or Restore fails. Here is what should be set when rolling back to a different version.

Example: Want to Restore the VM to SR10 from SR11 (value for the “New” snapshot indicates what snapshot will be deleted and the “Old” snapshot indicates which snapshot the VM will be restored to)

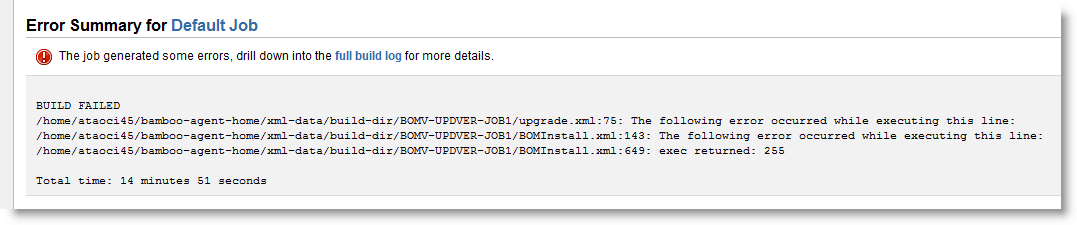
|  |  |  |
| --- | --- | --- |
| **Purpose** | **Ant Property** | **New Value** |
| New VM Snapshot name to delete | -Dnew.sr.number | SR11 |
| Previous Installed SR\HF snapshot version to restore to | -Dold.sr.number | SR10 |

Here is an example of a failure…

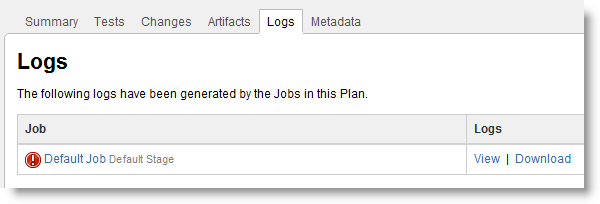
Click on the exclamation point.



View the Error Summary, or click on full build log for the full details of the failure.



Alternatively, click on the Logs tab



# VM Configuration

The two VM servers that are currently providing support for the BOM Verification are

* KWS-US-BOMVSR (Application server)
* KDB-US-BOMVSR (Database serve)

There is a service account set up to perform VM tasks via the VMWare API to perform tasks such as:

* Turning the VMs ON and OFF
* Creating new snapshots
* Restoring to previous snapshots
* Deleting snapshots

The VSphere client installation provides the necessary libraries in order to access the API as well as sample files that can be leveraged to perform these basic types of functions. The VMware SDK needs to be downloaded and installed. Because some of the scripts used to connect to VMware needed to be compatible for testing on Windows and Linux, the scripting technology used was Perl. <http://www.vmware.com/support/developer/viperltoolkit/>

How to Add VMWare task to verification

The primary Ant script used to prepare the environment for installation is upgrade.xml. Various VM tasks are contained in this script like the create VM for example:

<target name="run.primary.vm.create">

<echo message="===> Creating new app SR VM snapshot : ${new.vm.name}." />

<exec executable="perl">

<arg value="snapshotmanager.pl"/>

<arg value="--server"/>

<arg value="${vm.server.ip}"/>

<arg value="--vmname"/>

<arg value="${vm.app.server.name}"/>

<arg value="--snapshotname"/>

<arg value="${new.vm.name}"/>

<arg value="-username"/>

<arg value="${vm.username}"/>

<arg value="--password"/>

<arg value="${vm.password}"/>

<arg value="--operation"/>

<arg value="create"/>

</exec>

<echo message="===> New db SR VM ${new.vm.name} has been created." />

<echo message="===> Creating new app SR VM snapshot : ${new.vm.name}." />

<exec executable="perl">

<arg value="snapshotmanager.pl"/>

<arg value="--server"/>

<arg value="${vm.server.ip}"/>

<arg value="--vmname"/>

<arg value="${vm.db.server.name}"/>

<arg value="--snapshotname"/>

<arg value="${new.vm.name}"/>

<arg value="-username"/>

<arg value="${vm.username}"/>

<arg value="--password"/>

<arg value="${vm.password}"/>

<arg value="--operation"/>

<arg value="create"/>

</exec>

<echo message="===> New db SR VM ${new.vm.name} has been created." />

</target>

The Ant task is leveraging a VM support file called *snapshotmanager.pl* and is passing in the required arguments to create a new snapshot of the App and Db server VM task files include:

* Snapshotmanager.pl
* Vmcontrol.pl
* AppUtil (Contains libraries required for Perl to use to access VSphere client)

These support files have been included in the verification checkout so that that they are local to the build. *[SVN : svn+ssh://username@svn.deploy.com/svnroot/repository6/trunk/ataoTools/verification]*

# Creating a New Environment

This is a bit complicated to cover all of the changes required when cloning an existing environment and renaming it and changing the correct config files in order to the get the environment working. The BOMVSR servers were imaged from the QA environment: QA-Test-1. See <https://confluence.aspdeploy.com/display/ConEng/Qa-test-1+Environment+Setup+Plan>

When cloning an environment, the only thing that changes by IT is the server DNS name and the IP address. All other files in the environment need to be updated so that Atao and the database can operate correctly on the newly named machines.

Oracle connection from app server

Locate the *tnsnames.ora* file to change the database server name accessed. [*/home/oracle/10.2.0/db\_1/network/admin/tnsnames.ora*]

Ex. Before

ORCLQA2 =

(DESCRIPTION =

(ADDRESS = (PROTOCOL = TCP)(HOST = kdb-us-v8qa1db.us.kronos.com)(PORT = 1521))

(CONNECT\_DATA =

(SERVER = DEDICATED)

(SERVICE\_NAME = orclqa2)

)

)

Ex. After

ORCLQA2 =

(DESCRIPTION =

(ADDRESS = (PROTOCOL = TCP)(HOST = kdb-us-bomvsr.us.kronos.com)(PORT = 1521))

(CONNECT\_DATA =

(SERVER = DEDICATED)

(SERVICE\_NAME = orclqa2)

)

)

Atao jboss config files

There are 4 jboss configuration files that need to be updated with the appropriate server information or Atao will not start up correctly. (See [qa-test-1](https://confluence.aspdeploy.com/display/ConEng/Qa-test-1+Environment+Setup+Plan) environment config for more help)

1. Atao-ds.xml
2. Atao.hibernate.cfg.xml
3. Custom.properties
4. Log4j.properties

Here is the Jboss start and stop commands on this server.

nohup run.sh -b 0.0.0.0 -g kws-us-bomvsr -u 230.0.0.227 1> /home/ataoci45/jboss/jboss-as/jboss.out 2> /home/ataoci45/jboss/jboss-as/jboss.err &

shutdown.sh -S -u admin -p admin --server=kws-us-bomvsr:1099

Bamboo agent auto install

Since the webserver will be restarting frequently, it will be important for the bamboo agent to restart automatically on startup so that the build plan can start running as soon the OS starts. This was accomplished by calling the bamboo startup script by putting a bash script in init.d.

The bash script was created at */etc/init.d/bamboo.* Here is the script (NOTE: ataoci45 is the user on kws-us-bomvsr)

#!/bin/sh

# chkconfig: 345 99 10

# description: bamboo agent startup script

#

# start the bamboo agent

#

BAMBOO\_HOME=/home/ataoci45/bamboo-agent1/

BAMBOO\_USER=ataoci45

if [ ! -f $BAMBOO\_HOME/bin/bamboo-agent.sh ]

then

echo "Bamboo Agent startup: cannot start"

exit 1

fi

case "$1" in

start)

# Starting Bamboo Agent

echo -n "Starting Bamboo agent: "

su - $BAMBOO\_USER -c "$BAMBOO\_HOME/bin/bamboo-agent.sh start"

echo "OK"

;;

stop)

# Stopping Bamboo Agent

echo -n "Shutdown Bamboo Agent: "

su - $BAMBOO\_USER -c "${BAMBOO\_HOME}/bin/bamboo-agent.sh stop"

echo "OK"

;;

reload|restart)

$0 stop

$0 start

;;

status)

# Status of Bamboo Agent

echo -n "Starting Bamboo agent: "

su - $BAMBOO\_USER -c "$BAMBOO\_HOME/bin/bamboo-agent.sh status"

echo "OK"

;;

\*)

echo "Usage: $0 start|stop|restart|reload|status"

exit 1

esac

exit 0

Bamboo agent license

It is important to note that each server has to have a paid bamboo license.

Database sqlplus auto install

Similar to the auto start of the bamboo agent, the database server needs to be set to startup up the sqlplus service on startup or the BOM installation will fail because it won’t be able to access to the database (remember the installation is running from the app server).

The bash scripts were created at */etc/init.d/oracleListener.* Here is the script (NOTE: oracle is the user on kdb-us-bomvsr)…

#!/bin/bash

#

# start the bamboo agent

#

su oracle

echo lsnrctl start

echo dbstart

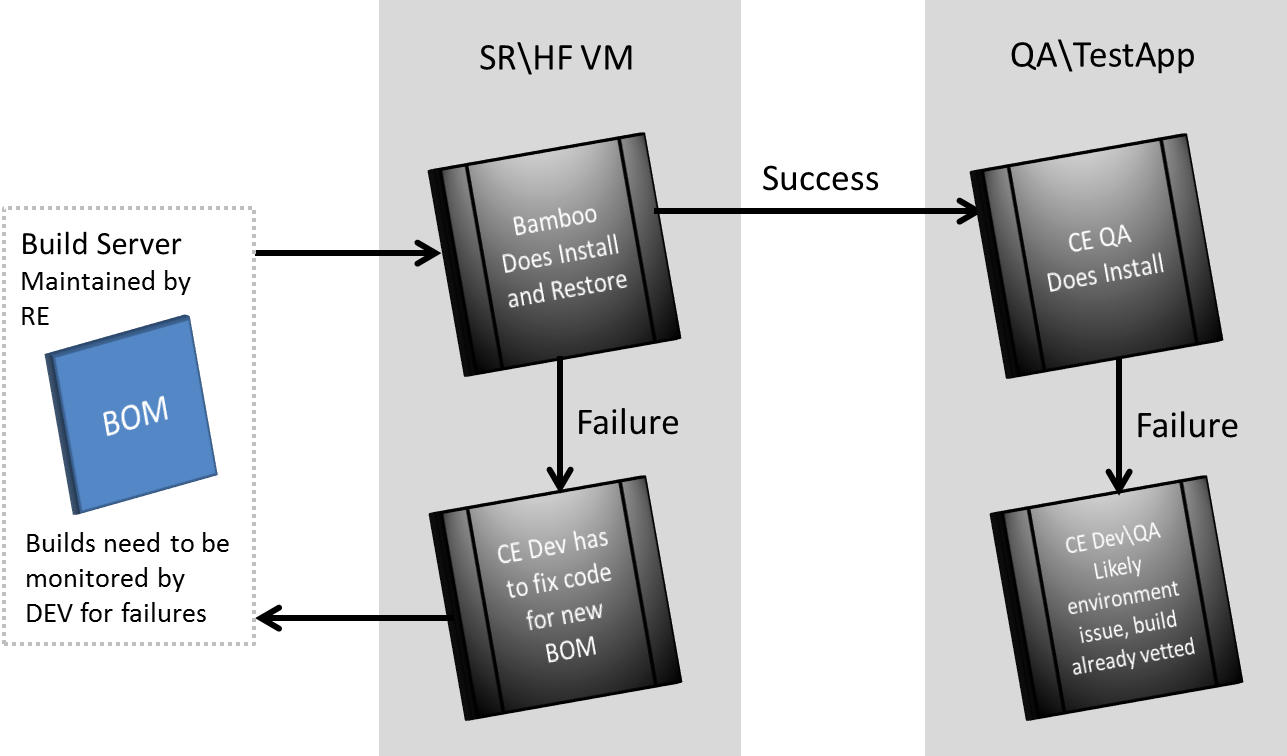
Database listener

The listener needs to be updated on the database server as well or the database will be inaccessible. The change is made in the *tnsname.ora* file like on the app server. Here is the location on the database server : */home/oracle/oracle/product/10.2.0/db\_1/network/admin/tnsnames.ora*

# Process Ownership

RE will be responsible to support the build servers and monitor the VMs to ensure installs can be made continuously and VMs restored as necessary. If the BOM fails to build, it will likely be the result of a check-in of invalid code, but that will have to be determined by watching the build logs. This is the same process as today, as the build could also fail because of an invalid setting in the build configuration.

Bamboo Ant scripts will automate the installation and if there is a failure, it needs to be addressed by DEV as it is likely a code issue. VMs will be restored automatically via perl scripts using the VMWare API.



Bamboo

Does Install and Restore

SR\HF VM

QA\TestApp

Failure

Build Server

Maintained by RE

CE QA

Does Install

Success

BOM

Failure

Bamboo

Does Install and Restore

SR\HF VM

QA\TestApp

Failure

Build Server

Maintained by RE

CE QA

Does Install

Success

BOM

Failure

Builds need to be monitored by DEV for failures

# Enhancements

In this section the goal is collect atomic enhancements that may or may not be implemented in the first release or future releases.

1. Allow BOM build to be kicked off on code check-in.
2. Add 2 additional VMs so a developer could load a snapshot of a failed install or install a failed BOM to troubleshoot install or restore errors.