MPI Auto Deployment

Gateway System Documentation

|  |  |
| --- | --- |
| *Document content* | MPI Auto Deployment |
| *File name* | PSCAutoDeployDocumentation |
| *System name* | Platform – SLOT 3 |
| *Author* | Neil Street |
| *Current version* | 3.0 |
| *Last edit* | 10/02/2012 |
| *Status* | Release |

# Table of Contents

[1. Table of Contents 2](#_Toc299003730)

[2. Document control 3](#_Toc299003731)

[2.1 Review and signoff audience 3](#_Toc299003732)

[2.2 Document references 3](#_Toc299003733)

[2.3 Document revision history 3](#_Toc299003734)

[3. General 4](#_Toc299003735)

[3.1 About this document 4](#_Toc299003736)

[3.2 Features 4](#_Toc299003737)

[3.3 Assumptions 4](#_Toc299003738)

[3.4 Pre-requisites 6](#_Toc299003739)

[4. MPI\_AutoDeploy – Files & Directory Structure Explained 7](#_Toc299003740)

[5. Required Property Changes 8](#_Toc299003741)

[6. Running the Scripts 10](#_Toc299003742)

[6.1 Open a new command shell and configure the environment 10](#_Toc299003743)

[6.2 Package Preparation 10](#_Toc299003744)

[6.3 Manual preparation of the weblogic domain 11](#_Toc299003745)

[6.4 Configuring JMS: 11](#_Toc299003746)

[6.5 Configure JDBC resources 12](#_Toc299003747)

[6.6 Deploying PSC’s in GWA\_Applications domain: 12](#_Toc299003748)

[6.7 Deploying PSC’s in GWA\_ServiceBus domain: 12](#_Toc299003749)

[7. Post Deployment Process 13](#_Toc299003750)

[8. Known Issues 14](#_Toc299003751)

[9. Troubleshooting 15](#_Toc299003752)

# Document control

## Review and signoff audience

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Position** | **Sign-off/Review/Information** | **Date** |
| Darren Murphy | Infrastructure Architect | Review & Sign-off |  |
| Shibu Paul | Application Deployment Manager | Review |  |
| Lander Lopez | Integration Office Manager | Review |  |
| Hussain Anjum | Technical Support | Review |  |
|  |  |  |  |

## Document references

|  |  |  |
| --- | --- | --- |
| **Document Name** | **Owner** | **Version** |
|  |  |  |

## Document revision history

The document revision history covers changes to this document; changes to the system should be detailed in the *System revision history* section.

|  |  |  |  |
| --- | --- | --- | --- |
| **Description** | **Date** | **Name** | **Version** |
| 1st draft | 12/01/2011 | Surendar Patchimalla | 0.1 |
| 2nd draft | 01/02/2011 | Surendar Patchimalla | 0.2 |
| 3rd draft | 02/02/2011 | Neil Street | 0.3 |
| 4th draft | 15/02/2011 | Neil Street | 0.4 |
| 5th draft | 10/03/2011 | Surendar Patchimalla | 0.5 |
| Draft for 2.4 release | 19/07/2011 | Neil Street | 2.4 |
| Minor changes for 2.4.1 | 20/09/2011 | Neil Street | 2.4.1 |
| Changes to remove config registry | 10/02/2012 | Neil Street | 3.0 |

# General

## About this document

The purpose of this document is to provide the details steps in using MPI\_AutoDeploy scripts to deploy PSC’s into the environment. The scripts can be used to deploy PSC’s into GWA\_ServiceBus and GWA\_Applications Weblogic domains

## Features

* Deploys application and implementation code into GWA\_Applications domain targeted to cluster(s) or single/multiple managed servers
* Deploys /Imports OSB code
* Creates Data sources
* Creates JMS resources
* Can be used for PSC Patch deployment
* Can be used for PSC full deployment
* Checks the state of managed servers before deployment of web technology PSC’s
* Does support deployment against SSL domains of GWA\_Applications & OSB
* Does support custom roles for PSC’s in of GWA\_Applications domain

## Assumptions

* If SSL has been configured for the GWA\_Applications & GWA\_ServiceBus domains the necessary Keystores should be available for scripts in the script root directory. The keystore files are found on either the managed server or the admin server at:

D:\Domains\<Domain Name>\servers\security\keystores

And are called MPITrustList.jks

The keystore from XXX\_Applications domain should be renamed AppsMPITrustList.jks and that from XXX\_ServiceBus domain should be renamed OSBMPITrustList.jks

* The instructions in this document will deploy against the Weblogic domains illustrated in the deployment topology below,

XXX\_ServiceBus domain



GWA\_Applications Domain



* Domain Administration servers ( ex: XXX-SR-ORADM-20) are accessible from the desktop running MPI\_Deploy Scripts. This may or may not be a XXX-DEPLY-nn deployment server.
* All the required properties are either copied into D:\PSCConfig on the respective Managed Servers or exists within the PSC’s ( i.e. WAR, JAR, EAR files). D:\PSCConfig is on the startup classpath of the managed servers and should exist when the server is started.
* PSC’s which needs deployment into XXX\_ServiceBus domain needs to have a customisation xml file with the right configuration details.

## Pre-requisites

The following are the pre-requisites to run MPI\_Autodeploy Scripts

1. Required Username & Passwords

* Domain Administrator Username name and Password for XXX\_ServiceBus domain
* Domain Administrator Username name and Password for XXX\_Applications domain

1. Windows domain admin user credentials to log on to XXX\_SR\_ORAPP-20 & XXX\_SR\_ORAPP-21
2. Trust Keystore for GWA\_ServiceBus domain
3. Trust Keystore for GWA\_Applications domain
4. Oracle Service Bus software v10.3 installed in the local machine where the scripts will be executed
5. Oracle Weblogic Software v10.3 or above installed in the local machine where the scripts will be executed
6. Java software installed in the local machine where the scripts will be executed
7. Scripts doesn’t copy property files required by PSC’s . These needs to be in place before the auto deployment.
8. The autodeploy scripts expect the deployment artefacts to be packaged either by using the configuration registry in a single zip file or to be represented by a manifest file which can be the product of the CMDB or a spreadsheet based on the deployables page of the configuration registry.

# MPI\_AutoDeploy – Files & Directory Structure Explained

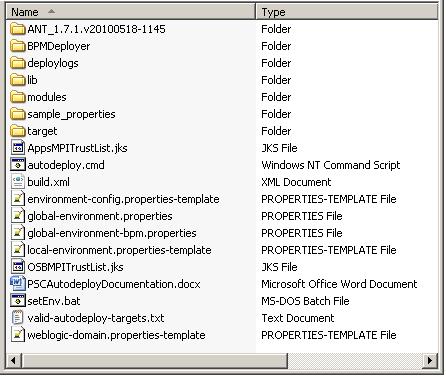
The scripts required for auto deploy should be acquired from the dsl and unzipped into a root directory where a file structure will be created as described below:

1. Java Keystore Files - OSBMPITrustList.jks and AppsMPITrustList.jks are the temporary files.

**Note: THIS APPLIES WHERE SSL IS CONFIGURED FOR THE DOMAIN**

**The MPITrust.jks trust files needs to be copied from respective domains and should be renamed as OSBMPITrustList.jks and AppsMPITrustList.jks**

1. Autodeploy.cmd the launcher for the ant environment – validates, checks and sets environment variables
2. build.xml: The Ant build file required for deployment.
3. modules: location of WLST resources
4. lib: The required library files for script execution
5. deploylogs: The log files are generated here. This directory will be created if it doesn’t exists
6. ANT\_1.7.1.v20100518-1145: Apache Ant software required for script execution.
7. Properties template files: Properties template files with changeable configuration to be edited locally to create properties files.
8. global-environment.properties: File containing common properties
9. setEnv.bat: Batch file initiated at the start of the deployment session to setup runtime environment for scripts



# Required Property Changes

The following property values needs changing for the scripts to execute with right details These files does exists in MPI\_AutoDeploy directory

**Note: OSBMPITrustList.jks & AppsMPITrustList.jks provided in MPI\_AutoDeploy folder are samples and those Java Keystore cannot be used for deployment**

**OSBMPITrustList.jks** : Copy the MPITrustList.jks from Domain Administration server “ i.e. GWA-SR-ORADM-20” at D:\Domains\GWA\_ServiceBus\security\keystores and rename to “OSBMPITrustList.jks”

**AppsMPITrustList.jks** : Copy the MPITrustList.jks from Domain Administration server “ i.e. GWA-SR-ORADM-20” at D:\Domains\GWA\_Applications\security\keystores and rename to “AppsMPITrustList.jks”

**Note: All the files with .template extension must be renamed appropriately by deleting “.template” and chaging the values in the property file as specified below**

**weblogic-domain.properties-template**

These properties describe the weblogic domain properties for APP and OSB admin servers.

The following property values needs to be changed and the file saved as weblogic-domain.properties.

|  |  |  |
| --- | --- | --- |
| **Name** | **Value** | **Description** |
| weblogic.script.debugging | off | Used in troubleshooting to control output of WLST scripts |
| weblogic.application.deploy.without.version.identifier | false | Optional defaulting to false. Used to deploy without version – a workaround for a TempBooking issue. |
| domain.datacentre.code | GWA | Datacentre prefix – used in jms resource configuration – not fully implemented at this version as exact structure of naming schema still unknown |
| oradm.domain\_name | GWA\_Applications | GWA Application domain name |
| oradm.admin\_user | GWAApplicationsAdmin | Administration username for the GWA\_Applications domain |
| oradm.admin\_password | Or@c1e@pp@dm!n | Administration password for the GWA\_Applications domain |
| oradm.admin\_url | t3s://<Hostname/IP>:<PORT> | GWA\_Applications domain admin hostname and port |
| oradm.keystore.file | ./AppsMPITrustList.jks | Relative path to keystore file for GWA\_Applications domain |
| oradm.keystore.passphrase | password | Passphrase for keystore |
| gwapp.target | GWA\_User\_Applications | Cluster Name or coma separated Managed Server names |
| gwimp.target | GWA\_Service\_Implementations | Cluster Name or coma separated Managed Server names |
| Singletonapp | ORAPP20\_MS1 | Managed server for singleton applications |
| Singletonimp | ORAPP21\_MS2 | Managed server for singleton implementations |
| osbadm.adminUrl | t3://10.240.15.57:9003 | OSB Domain URL |
| osbadm.admin\_user | gwaosbdeployer | OSB Domain username |
| osbadm.admin\_password | G@teway80 | OSB Domain password |
| osbimp.target.name | GWA\_OSB | Name of the osb target used by config registry |
| osbimp.admin.server | osbadm | Prefix of properties containing osb admin credentials |
| gw.osb.keystores.identity.location | C:\\MPI\_AutoDeploy\\OSBMPITrustList.jks | OSB domain keystore |
| gw.osb.custom.keystore.passpharase | Password123!MP | Password for the above keystore |
| gw.keystores.identity.location | C:\\MPI\_AutoDeploy\\AppsMPITrustList.jks | User Applications domain keystore |
| gw.custom.keystore.passpharase | Password123!MP | User Applications domain keystore password |

**local-environment.properties-template**

These properties describe the local installation of required software in an ant-consumable format (unix-style paths look nicer than double-backslash).

The following property values needs to be changed and the file saved as weblogic-domain.properties.

|  |  |  |
| --- | --- | --- |
| **Name** | **Value** | **Description** |
| osbimp.install.path | path | Path to your osb server home. This is validated by checking the existence of ${osbimp.install.path}\lib\alsb.jar |
| wl.home | path | Path to your weblogic server home. This directory is used to locate the environment setup batch file supplied by weblogic which is located at ${wl\_home}/ server/bin/setWLSEnv.cmd. This is also used to infer the BEA\_HOME location presumed to be the parent of this directory and this is validated by checking for the existence of ${wl\_home}\..\modules\com.bea.common.configfwk\_1.2.1.0.jar |
| mpi.autodeploy.home | path | Root of this directory (the directory created when the autodeploy distribution was unzipped) |
| jrockit.home | Path | Path to jrockit jre (located in a weblogic home directory or a java directory) |

**global-environment.properties**

These properties determine the behaviour of the deployment tasks. Do not change these unless directed to do so for debugging or similar purpose.

**environment-config.properties-template**

These properties describe the deployment session. The deployment session is the deployment of an environment-agnostic versioned zip (containing the deployable artefacts) together with a versioned configuration file containing environment-specific configuration.

This file should be changed for the deployment session and saved as environment-config.properties

|  |  |  |
| --- | --- | --- |
| **Name** | **Value** | **Description** |
| package\_zip | path | Either this property or manifest.file (and associated properties) must be present. Full path to the package zip |
| manifest.file | path | Either this property or package\_zip must be present. The path to the manifest file |
| manifest.source | cmdb, ci | If manifest.file present, this is required. The source of the manifest (describing its format) |
| manifest.metadata.source | cr, ci, cmdb | If manifest.file present, this is required. The source of the deployment metadata (describing its format) |
| manifest.metadata.file | Path | If manifest.file present, this is required. The path to the file containing the metadata |
| manifest.repository.protocol | file, ftp | If manifest.file present, this is required. How the deployable artefacts will be retrieved |
| manifest.repository.root | Path/hostname | If manifest.file present, this is required. Either the hostname or the path to the repository. The root from which the repo-path metadata element is relative. |
| config.file.path | path | Full path to the environment specific configuration file |
| config.file.version | version | The version of the environment-specific configuration file. The configuration file should contain a property – property.file.version=<version> and <version> should be the same as this value |

# Running the Scripts

This section explains how to run the scripts to deploy into respective weblogic domains. It is **important** that section 5 above is done completely before attempting to run the scripts. Also pay particular attention to section 3.3 Prerequisites.

The execution of a deployment will be referred to as a deployment session which consists of the following steps:

1. Ensure that the descriptive properties are defined. These are the property files described in section 5 which describe the weblogic domains, the local software installation and the deployment package and configuration.
2. Setenv.bat uses the properties to set the shell environment so that the deployment scripts can execute.
3. Package preparation. Depending on the type of deployment, either the environment-agnostic zip file is expanded into a temporary location and transformed for deployment into the target environment using the supplied environment-specific configuration properties file or the manifest is used together with a repository of artefacts to create the temporary environment-specific file system for deployment.
4. Manual preparation of the weblogic domains. This may include undeploying and deletion of staged files and will almost always include the copy of the PSCConfig folder from the prepared package to the root directory of the managed servers. Note: if using in a CI environment, exploded packaging cannot be deployed because manual intervention is required. Also, we do not yet have an automated way of dealing with configuration file deployment.
5. Execution of automated deployment steps in order:
6. Creation of JMS and/or datasource resources
7. Deployment of web technology PSC’s
8. Deployment of OSB technology PSC’s

The deployment session is assumed to start with a new command shell which is configured once for the execution of the deployment of the deployable package. The same shell is then used to perform the steps that comprise the deployment.

The scripts are run using ANT which is initiated via a batch file which tries to validate as much as possible. The autodeploy command file may display the current setting of the required environment variables. This is because it is vitally important that the environment is set correctly prior to script execution. Once you are satisfied that these are correct, using the –q commandline switch will switch off the prompts. If prompting is not switched off then the input of a change request number is enabled which can be set to the identifier that you wish to use to prefix the log files.

## Open a new command shell and configure the environment

Open a new command shell, change the current directory to the MPI\_Autodeploy location and run setenv.bat. This should result in 4 [INFO] messages indicating the environment variables set from the environment properties files, another [INFO] message indicating that the weblogic-supplied environment setup is being run, the echo of the (fairly lengthy) classpath and a message indicating that the environment is set (“Your environment has been set.”). If it does not, it is likely that one or more of the environment properties configured in section 5 are incorrect. Until this has run successfully, go back to section 5, configure the properties to reflect your local setup and do this again from a new command prompt. Once this has run successfully, use the same shell to perform all subsequent autodeploy activity.

## Package Preparation

To configure the packaged deployment for the target environment one of the following will occur:

Tthe packaged zip file is expanded into a temporary location and then prepared for the environment by translating tokenised configuration into environment-specific values using the environment-specific configuration properties file.

The manifest file and metadata files are processed to acquire the deployable artefacts and create the temporary file system. The configuration files have token substation performed on them. Note: right now, we do not have a way of dealing with the configuration files other than in a “release” deployment with either a zip package or a file system artefact repository.

The prerequisite for this step is the running of setenv.bat and this is the prerequisite for all subsequent activity. Package preparation results in the configured deployable artefacts being created in a single directory in the tmp folder. That folder location is set in the environment as PACKAGE\_HOME and the environment-config.properties file is updated with this location.

At the command prompt enter:

autodeploy package-prepare

The tmp folder now contains the expanded package prepared for the environment. Any manual changes that may have to be made as a result of compatibility issues should be made to the expanded package now. For example, we may wish to change a deployment target for a PSC from a cluster to a singleton. Check the release note that accompanies the package and refer to Known Issues below.

## Manual preparation of the weblogic domain

Because of the way in which some web technology PSC’s are deployed, when there is a subsequent deployment, there can be issues with staged files from the previous deployment. We do not yet have a satisfactory way of dealing with those automatically that will work in all environments. For this reason it is currently necessary to manually remove these files. It is also necessary to copy the configuration files from the prepared package to the root of D: on each of the managed servers. The steps to perform this preparation are as follows:

1. Log in to the admin console of the GW\_Applications domain. (url is the value of oradm.admin\_url property in section 5 replacing t3 with http and adding /console. User and password are the values of oradm.admin\_user and oradm.admin\_password properties)
2. In the deployments page stop and delete any of the applications in the deployment package that are already deployed.
3. Stop the managed servers
4. Access the D: drive of each of the managed servers in the appropriate manner for the environment (simply map a drive or use FTP) and:
   1. Copy the PSCConfig folderfrom the configuration folder of the package (specified in the release note) to the root of the D: drive
   2. For each <server-name> directory in D:\Domains\GW\_Applications\servers folder:
      1. Remove the contents of log and tmp folders
      2. Remove any staged files in stage folder associated with any of the applications in the deploy package
5. Restart the managed servers

## Configuring JMS:

1. In the command shell enter “autodeploy createjmsresources “ or “autodeploy createjmsresources –q”. This will configure the JMS specified in the released package. Check the Logs in deploylogs folders for errors
2. Without the –q option, the script will display and prompt for the required information set in the environment in the setenv and preparation phases above. Without it, the script will prompt for:
   1. PACKAGE\_HOME – this is the root of the prepared distribution set in the preparation phase
   2. Change Request Number – optional tag to prefix log files
   3. BEA\_HOME – as described in section 5 this is inferred from the value of wl\_home property
   4. OSB\_HOME – derived from the osb.install.path property defined in section 5 and set with setenv

If the –q switch is used, prompting is switched off so it is not possible to enter the change request number. If it is entered once or if the CHANGE\_REQUEST\_NUMBER environment variable is set manually in the shell, then the logs will be prefixed with that value. If the CHANGE\_REQUEST\_NUMBER value is never entered, the log prefix defaults to “autodeploy”

## Configure JDBC resources

1. In the command shell enter “autodeploy createdatasources “. This will configure the JMS specified in the released package
2. The script will prompt for the required information as in 6.4 and can be switched off by using the –q switch
3. Logs in deploylogs folders for errors
4. Check DEPLOY\_STATUS<Timestamp>.log for the Success /Failure of Deployment

## Deploying PSC’s in GWA\_Applications domain:

1. In the command shell enter “autodeploy gwappdeploy “ or “autodeploy gwappdeploy –q”. The –q switch switches off prompting as described in 6.4
2. Check the Logs in deploylogs folders for errors
3. Check DEPLOY\_STATUS<Timestamp>.log for the Success /Failure of PSC Deployment

## Deploying PSC’s in GWA\_ServiceBus domain:

1. In the command shell enter “autodeploy osbdeploy “. This will deploy all the PSC’s specified in serviceslist.properties of the prepared package.
2. The script will prompt for the required information as in 6.4 and can be switched off by using the –q switch
3. Check the Logs in deploylogs folders for errors
4. Check DEPLOY\_STATUS<Timestamp>.log for the Success /Failure of PSC Deployment

# Post Deployment Process

# Known Issues

The following are the know issues which needs to be addressed in the future releases of the MPI\_AutoScript

1. Error handling in the scripts can be misleading. Most importantly, the scripts in their current form require accurate input. The configuration of the 3 configurable properties files in section 5 is error-prone so extreme care should be exercised to ensure that the values are correct. The autodeploy wrapper attempts some validation but some scenarios may not be catered for.
2. This version of the tooling expects a single zip packaged by the configuration registry using the generic token configuration. This version is not backward compatible.
3. The environment-specific configuration file used to map the package configuration to the environment should have a property:

property.file.version=<version>

where <version> exactly equals the value set in environment-config.properties for the property config.file.version. The tool validates this. The configuration registry is currently used to create this configuration file (it is our intention to phase this out and give ownership of the configuration map to the DC) so, currently, the presence of this property in the configuration properties file should be checked and, if it is not present, it should be added manually.

1. This version of the tool supports deployment of web technology PSC’s to one or more targets which are specified in weblogic-domain.properties. Up until now, the tool could only deploy to one or other of gwapp or gwimp. This change makes it impossible to maintain backwards compatibility in this regard. For the foreseeable future, the target for singleton deployment will have to be changed manually from the cluster to a single managed server. This is achieved by changing the value of the <psc-name>.target property in psc\_details.properties in the prepared expanded package in the tmp folder from gwimp to singletonimp and from gwapp to singletonapp.
2. The –q (quiet setting) is “sticky”. That is to say, once used, it is set by default. This is brought about because the ant batch file that is used to execute most of the deploy goals exits the command shell without returning to the calling command file. To unset the quite flag set the environment variable QUIET to something other than TRUE. This will be fixed in a future version.

# Troubleshooting

Check the logs for any errors if the build fails. The log is created in deploylogs directory.

The following are few exceptions and details on how to fix those.

1)

**ERROR**:

[java] Loading customization File C:\\SVN\\OIReq117\\OSB\\ApplicationExceptionService-1.0.0-build.3.jar.xml

[java] Unexpected error: org.apache.xmlbeans.XmlException

**FIX: Check for the xml file (Customisation File) is in the right format.**

2)

**Error:**

[java] File "C:\MPI\_AutoDeploy\import.py", line 196, in ?

[java] File "C:\MPI\_AutoDeploy\import.py", line 59, in importIntoDomain

[java] TypeError: \_\_add\_\_ nor \_\_radd\_\_ defined for these operands

**Fix: Check psc\_details.properties, if the entries are in right format.**

3)

java.io.FileNotFoundException: class path resource [application\_management\_config.properties] cannot be opened because it does not exist

**Fix: All classpath resources need to be present before deployment**