Deployer: Deploying a rule project to MWS

- Build a rule project with the Asset Build Environment (ABE)
- Create a repository
- Configure a Business Rules MWS server
- Create a Deployer project
- Define a Deployment Set
- Create a Deployment Map
- Add a Target Server
- Configure the Deployment Map "Merge project" and "Hot Deploy project""
- Create a Deployment Candidate
- Deploy the rule project to MWS

This page describes how to deploy a wM Business Rules project to a My webMethods Server, using wM Deployer.

Build a rule project with the Asset Build Environment (ABE)

- Refer to the "webMethods Deployer User's Guide", Chapter "Deploying Assets from a Source Asset Repository"
- Install the Asset Build Environment (ABE) using Installer
- Locate a directory containing rule project sources that were created with Designer Business Rules Development
 - This can either be in the Designer workspace
 - Or checked out from a Version Control System
 - For example, a "E:\ABE_INPUT\RULES" folder would contain rule project source files for individual rule projects:

```
E:\ABE_INPUT\RULES
RuleProjectA
        .classpath
        .project
        .version
        Actions
                .category
                someAction.action
                anotherAction action
        Data Models
                .category
                Customer.datamodel
                someActionInput.actiondatamodel
        Decison Tables
                .category
                CustomerRating.decisiontable
        Decision Trees
                .category
                CustomerRating.decisiontree
        Event Models
                PartInventoryLow.eventmodel
        Event Rules
                .category
                OrderChanged.eventrule
                InventoryAlert.externaleventrule
        Rule Sets
                .category
                RuleSetA
                        .ruleset
                RuleSetB
                        .ruleset
RuleProjectB
        Actions
        Data Models
        Decison Tables
        Decision Trees
        Event Models
        Event Rules
        Rule Sets
```

- Configure the *build.properties* file in "Software AG_directory \common\AssetBuildEnvironment\master_build"
 - Set the *build.src.dir* property to the directory containing rule project sources, for example "E:\ABE_INPUT"
 - build.source.dir=E:/ABE_INPUT

- The build.src.dir property may also contain directories for other asset types besides Rules, such as
 - Integration Server E:\ABE_INPUT\S
 - Universal Messaging E:\ABE_INPUT\UniversalMessaging
 - BPM E:\ABE_INPUT\BP
- Configure the build.output.dir property in the build.properties file
 - This defines the directory where the ABE build will place the created Rules assets, for example
 - build.output.dir=E:\ABE_OUTPUT
- Execute the ABE build by calling build.bat or build.sh
 - During the build you might see messages like

```
buildRULES:
...
[RulesArchiveAndAcldFileBuilder] E:\SL101\IS\MyCustomeISPackage is no rule project
...
```

- This is expected and indicates that the Rules ABE build encountered and skipped sources of a different asset type (for example, and IS package)
- The results of the ABE build will be available in the ABEs output directory under the RULES folder
 - for example:

E:\ABE_OUTPUT\RULES
RuleProjectA.jar
RuleProjectA.acdl
RuleProjectB.jar
RuleProjectB.acdl

Create a repository

Select the Asset Build Environment output directory as "File Directory", for example the "E:\(\text{ABE_OUTPUT}\)" directory used above.



MCJHENSCHEL02.eur.ad.sag :: webMethods Deployer

Close Window | About





WEBMETHODS Deployer

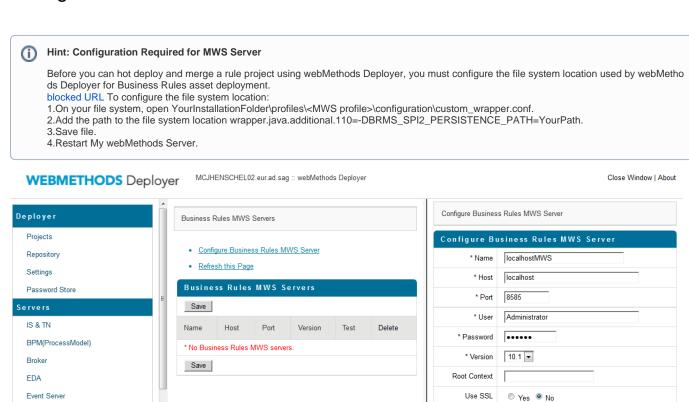
MCJHENSCHEL02.eur.ad.sag :: webMethods Deployer

Close Window | About





Configure a Business Rules MWS server



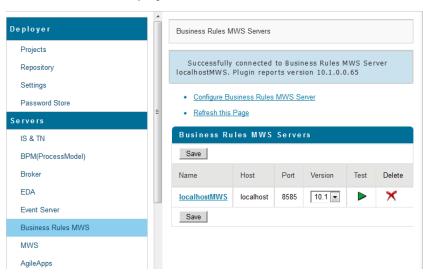
WEBMETHODS Deployer

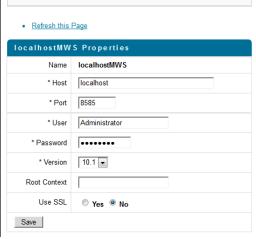
Business Rules MWS

MWS

MCJHENSCHEL02.eur.ad.sag :: webMethods Deployer

Close Window | About





Required fields are indicated by *

Configure

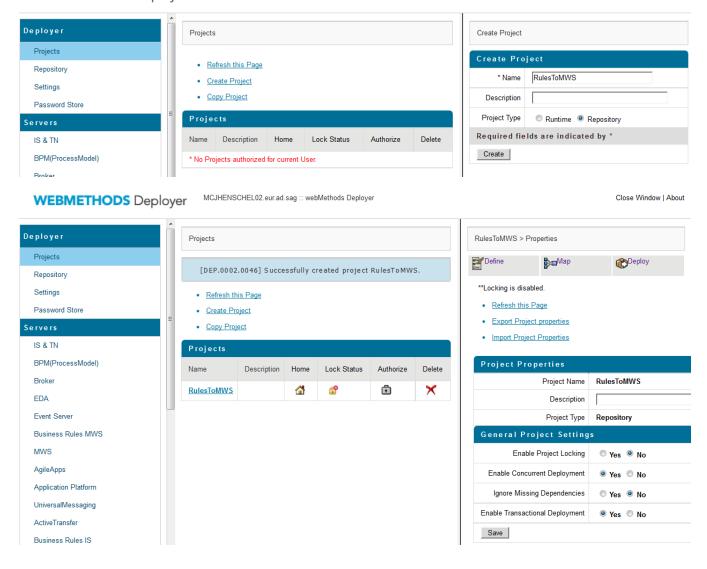
localhostMWS > Properties

Create a Deployer project

• Select "Project Type: Repository"

WEBMETHODS Deployer MCJHENSCHEL02.eur.ad.sag :: webMethods Deployer

Close Window | About

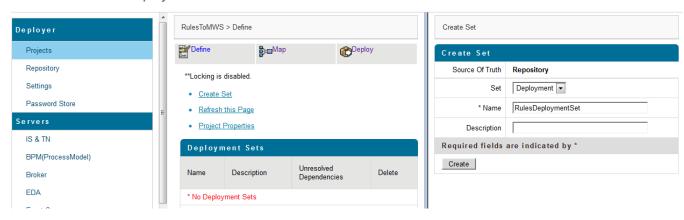


Define a Deployment Set

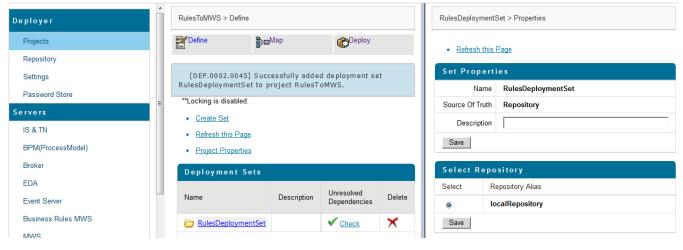
■ Select the Deployer project, click "Define", "Create Set", set "Set" to "Deployment"



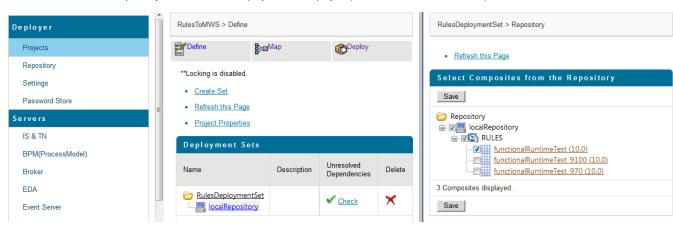
Close Window | About



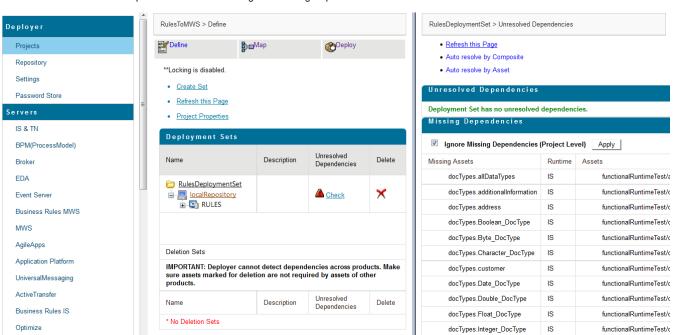
Select the previously created "localRepository"



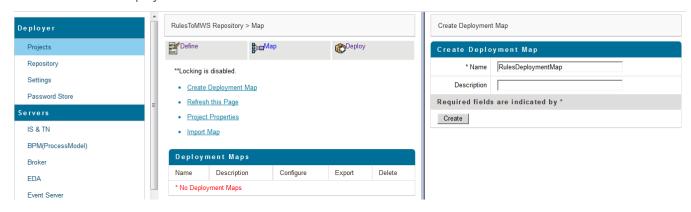
• Select the localRepository and select the rule project to be deployed (here, "functionalRuntimeTest")



■ Check"Unresolved Dependencies" and check "Ignore Missing Dependencies"

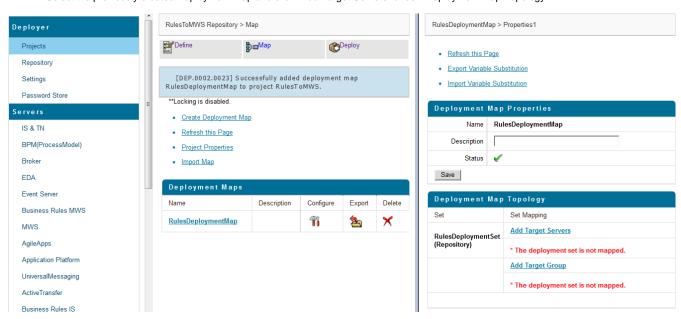


Create a Deployment Map

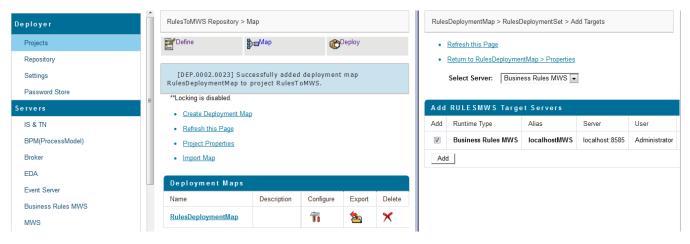


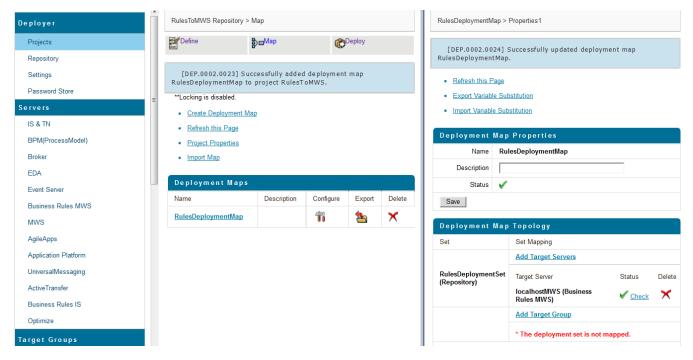
Add a Target Server

■ Select the previously created Deployment Map and click "Add Target Servers" under "Deployment Map Topology"



■ Select Server: "Business Rules MWS" and select the previously created "Business Rules MWS" server





Note: If your are deploying to a cluster of MWS servers, you MUST only configure one MWS server, NOT each cluster node. Rule projects are synchronized in the MWS cluster after deployment to this one MWS server. Deploying to multiple MWS cluster nodes in parallel will lead to synchronization errors.

Configure the Deployment Map - "Merge project" and "Hot Deploy project""

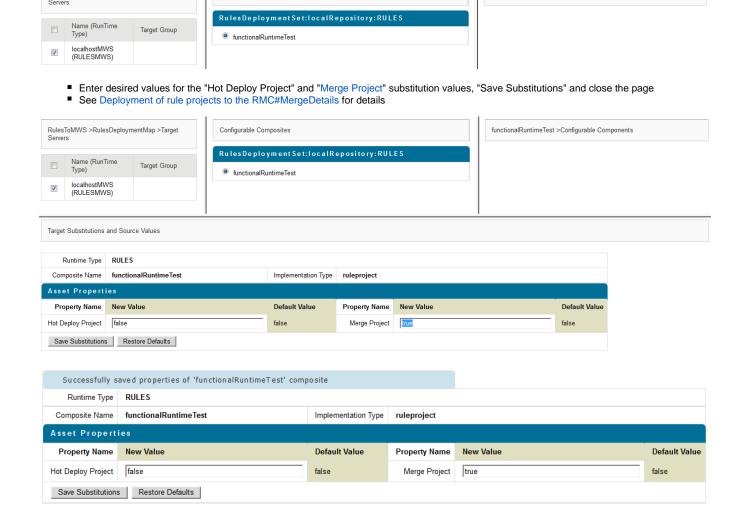
• Select the Deployment Project, click on "Map", click on "Configure" for the Deployment Map



■ Select the "RULESMWS" target server



Select the rule project to configure (here "functionalRuntimeTest")



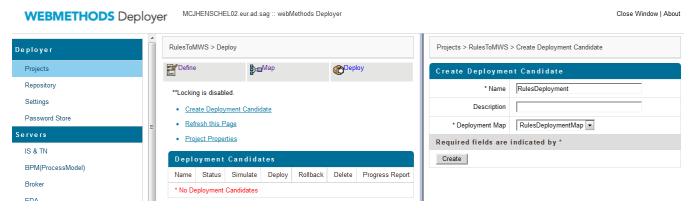
functionalRuntimeTest >Configurable Components

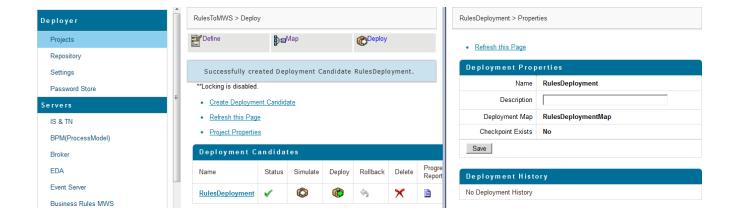
Create a Deployment Candidate

RulesToMWS >RulesDeploymentMap >Target

Configurable Composites

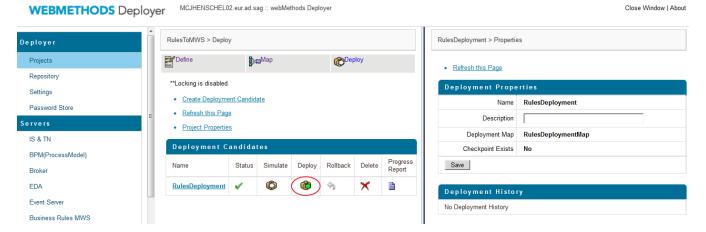
■ Select the Deployment Project, click "Deploy", "Create Deployment Candidate", select the previously created Deployment Map



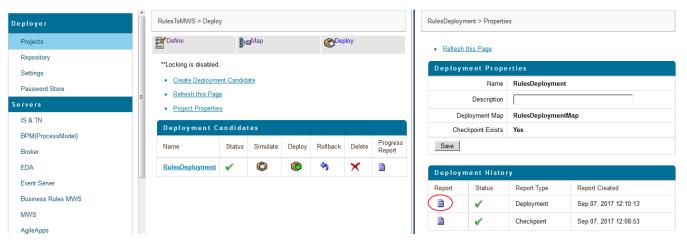


Deploy the rule project to MWS

• Select the Deployment Project, click "Deploy" and the "Deploy" icon



- Wait a bit. The rule project will now be deployed and merged, according to the previously set substitution values. This may take a while.
- Check the Deployment report



Target Server: localhostMWS				
TimeStamp	Message Type	Asset Type	Asset Name	Message
2017-09-07 12:08:53 CEST	MESSAGE			[DEP.0005.1168] Deleted previously delivered artifacts successfully for RULESMWS.
2017-09-07 12:08:53 CEST	MESSAGE			[DEP.0005.1164] Artifacts delivered successfully to RULESMWS.
2017-09-07 12:08:53 CEST	MESSAGE	ruleproject	functionalRuntimeTest	Merge is activated.
2017-09-07 12:10:13 CEST	MESSAGE	ruleproject	functionalRuntimeTest	Merge finished successfully.
2017-09-07 12:10:13 CEST	MESSAGE	ruleproject	functionalRuntimeTest	Hot deployment is not activated.
2017-09-07 12:10:13 CEST	SUCCESS	ruleproject	functionalRuntimeTest	Successfully activated project [functionalRuntimeTest]

[•] Voila! The rule project is now deployed to MWS and has been merged with the previously deployed version.