

Deployit Cookbook - Adding a check to the Deployment Checklist

Version 3.7.4

Table of Contents

Table of Contents	2
Adding a check to the Deployment Checklist	3
Adding the check to synthetic.xml	3
Configuring the Environment	3
Setting permissions	4

Adding a check to the Deployment Checklist

This Cookbook entry describes how to add a single check to the Deployment Checklist in the Release Dashboard. We will take an existing pipeline that has Dev, Test, Acceptance and Production environments. We will show how to add a single condition 'Signed off by Release Manager' that will be put on the Production environment.

See the **Release Dashboard Manual** for more extensive coverage on the subject.

Adding the check to synthetic.xml

First, we will have to tell Deployit that the new condition exists. We do this by adding properties to existing Deployit types. Deployment conditions in Deployit are defined by pairs of properties, one on the Environment and one on a DeploymentPackage (a version of an application).

To introduce the property into the Deployit's type system, we have to edit the `synthetic.xml` file on the Deployit Server. The `synthetic.xml` file can be located in `$DEPLOYIT_HOME/conf`. For more information about the Deployit type system, please refer to the **Reference Manual** and the **Customization Manual**

First we add the property to the Environment type. Locate the definition of `udm.Environment` and add the property:

```
<type-modification type="udm.Environment">
  ...
  <property name="requiresSignedOffByReleaseManager" kind="boolean"
    required="false"
    category="Deployment Checklist" />
  ...
</type-modification>
```

This means that for *any* Environment there will be an additional boolean property. The property has to start with the prefix `requires` in order to be picked up by the Release Dashboard.

The next thing we have to do in the `synthetic.xml` is to tell Deployit that deployment packages will have a property that can satisfy the check specified above. Note that each version of an Application will have different sets of properties checked during its voyage through the deployment pipeline. We therefore set the property on the `udm.Version` type.

Locate the type `udm.Version` and add the property. The property name is based on the property set on the Environment, but with the `requires` prefix replaced by `satisfies`.

```
<type-modification type="udm.Version">
  ...
  <property name="satisfiesSignedOffByReleaseManager" kind="boolean"
    required="false"
    category="Deployment Checklist" label="Signed off by Release Manager" />
  ...
</type-modification>
```

What we just did: for each individual deployment package we add the possibility to *satisfy* the constraints imposed by the *requires* property of the Environment.

After saving the file, restart the Deployit Server for the changes to take effect.

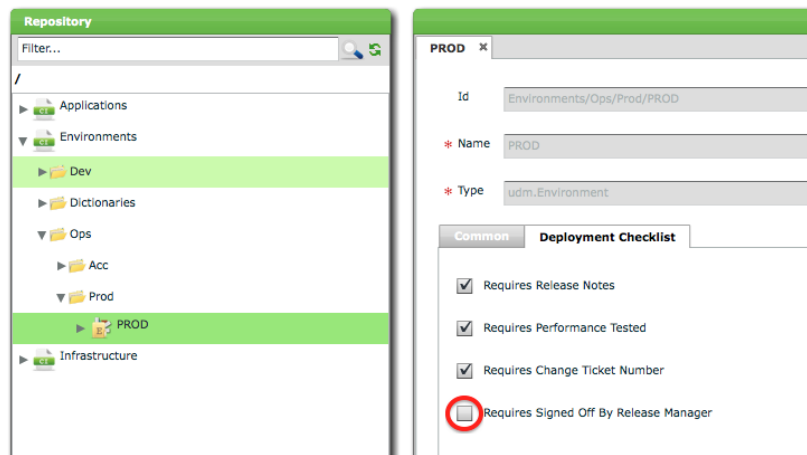
Configuring the Environment

Now that we have added the check, we can enable it for an environment. In this case, we will enable it only for the Production Environment, but of course you could enable it for any environment.

Once a check has been enabled on an environment, only deployment packages that satisfy the requirements can be deployed to this environment. Deployments that do not satisfy the check are not allowed.

We can easily configure the Environment from the UI. After restarting the Deployit Server, reload the UI in the browser. Got to the Repository tab, and find the Production Environment in the Environments tree. Double-click on it to open its editor and click on the "Deployment Checklist" tab. The new property will not be enabled, meaning that this check is not active for

the Production Environment. Tick the box to enable it and save.



Now go to the Release Dashboard and you will see that the condition has been added



Setting permissions

Here's a summary of the relevant security settings for the Release Dashboard:

- The `repo#edit` permission on the `DeploymentPackage` is used to check if a user is allowed to set or unset checks in the Deployment Checklist.
- When viewing the deployment pipeline, only the environments are shown if the user has `read` permission on it. For example, if user 'developer' has access to the Development and Testing environments, she will only see those environments in the Release Dashboard and not the others.
- Regular deployment permissions (`deploy#initial`, `deploy#upgrade`) apply when a deployment is initiated from the Release Dashboard.