

Run

Table of contents

1 Execute via JobCenter.....	2
2 Execute via 'ad'.....	3
3 Rolling back.....	4

Overview

This document describes how to run a content BuildAndDeploy workflow job or command and is pertinent to users responsible for releasing content changes from the source code repository to targeted deployment environments.

By running one of these workflows, pending changes in the SCM repository will be packaged up and deployed to the configured target node and directory.

For a new job to be established, the steps covered in the [Configure](#) section will already have been completed yielding a new content BuildAndDeploy workflow.

There are two interfaces available to execute the content BuildAndDeploy:

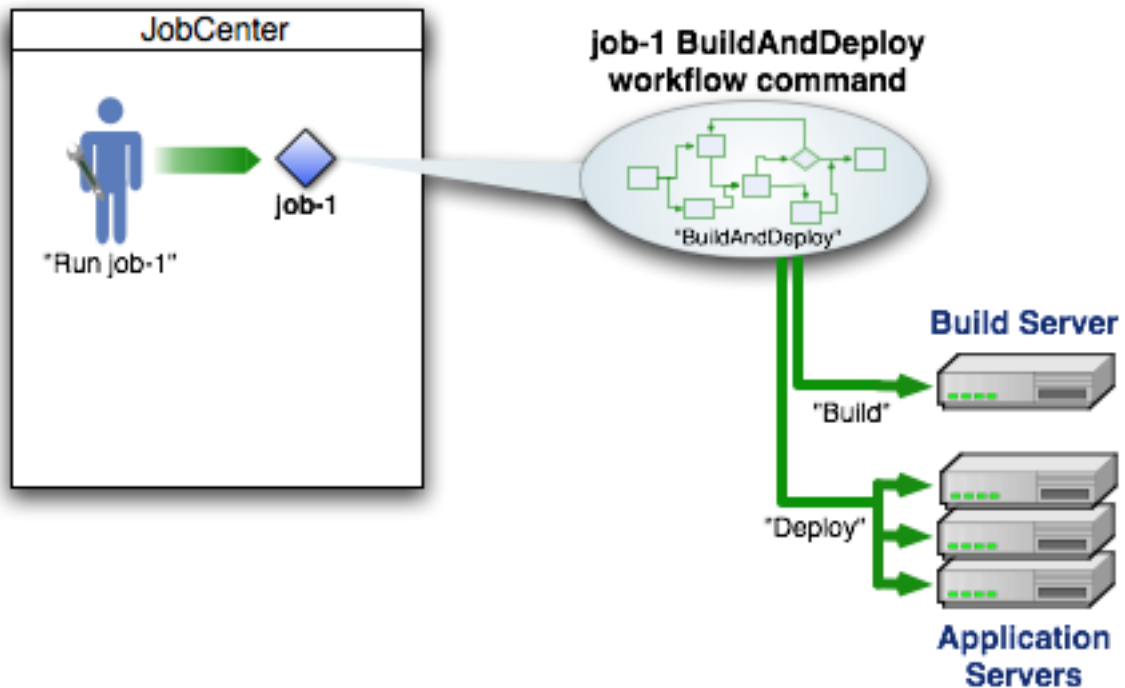
- Using JobCenter, the web-based graphical application,
- Using AntDepo's ad shell command

Instructions for running the BuildAndDeploy command using either interface are explained below.

1. Execute via JobCenter

In [Configure Step #3](#), a new job will have been defined for the BuildAndDeploy process. After logging into JobCenter a list of the defined jobs will be displayed. Choosing and running a job will execute the underlying BuildAndDeploy command.

Run



The general steps to using JobCenter to operate content BuildAndDeploy are listed:

1. Login to JobCenter
2. Identify the desired job
3. Run the job
4. Customize a report

2. Execute via 'ad'

An alternative to executing the BuildAndDeploy command via JobCenter is to execute it directly via the `ad` shell command. The general usage is shown below:

```
ad -p project -t ContentUpdater -o name -c BuildAndDeploy -- \  
-buildstamp buildstamp
```

A typical convention is to use the date and time as the `-buildstamp` argument. For example:

```
ad -p project -t ContentUpdater -o name -c BuildAndDeploy -- \  
-buildstamp 200711071500
```

If you are an experienced AntDepo user, you may also know how to run individual parts of the build and update workflow by running the appropriate command from one of the subordinate commands. For example, to run just the Deploy:

```
adminhost$ ad -p project -t ContentUpdater -o name -c Deploy
```

Or to run just the build, you can execute the Builder directly:

```
buildhost$ ad -p project -t ContentBuilder -o name -c Build -- \  
-buildstamp buildstamp
```

3. Rolling back

A nice feature of the Content library is its support for rollback. Before a ContentPackage is deployed, the library first checks if existing local files will be overwritten by new ones in the package. If any are detected, a "rollback" package is generated and stored on the repository. This package can be used to later restore the original files.

General Usage

Rollback is accomplished by running two commands on the admin host:

1. runChangeDependencies: This configures the deployments to depend on the rollback package:

```
ad -p project -t ContentUpdater -o name -c runChangependencies -- \  
-buildstamp rollback-buildstamp
```

2. Deploy: This deploys the rollback package to the ContentDeployments:

```
ad -p project -t ContentUpdater -o name -c Deploy
```

In depth descriptions of the Content library commands can be found in the Reference section of the documentation.

[Next: Reference #](#)