

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 Rdb data BuildAndDeploy workflow job or command and is pertinent to users responsible for releasing Rdb data changes from the master database to target database schemas in other environments.

By running one of these workflows, the content of the master database will be exported to a file, packaged up and deployed to the node where the target database schema resides.

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

There are two interfaces available to execute the Rdb data BuildAndDeploy:

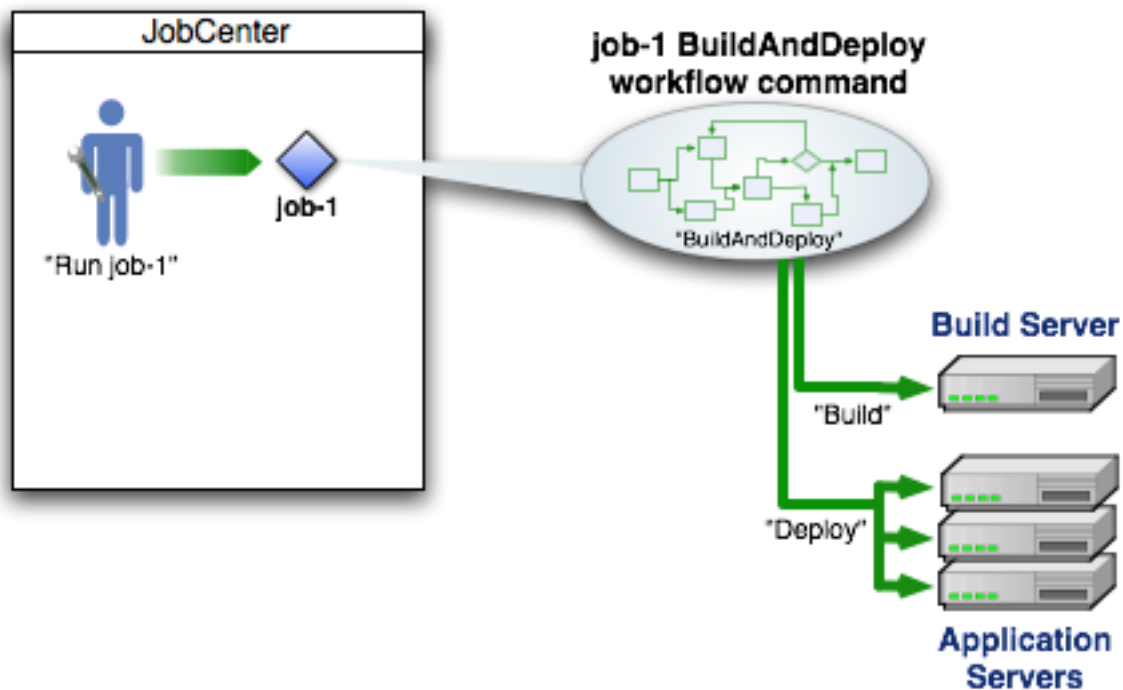
- Using JobCenter, the web-based graphical application,
- Using AntDepo's ad shell command

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 rdb 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 Updater -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 Updater -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 deploy workflow by running the appropriate command from one of the subordinate commands. For example, to run just the Deploy:

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

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

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

3. Rolling back

A nice feature of this library is its support for rollback. Users can choose from any RdbDataPackage in the repository and restore the schema data content to the desired version.

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 Updater -o name -c runChangdependencies -- \  
-buildstamp buildstamp
```

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

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

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

[Next: Reference #](#)