Presto User Guide

Last Updated: 3-Mar-2012

[Overview 3](#_Toc318574895)

[What is Presto? 3](#_Toc318574896)

[The Deployment Process 3](#_Toc318574897)

[What are the benefits of Presto? 3](#_Toc318574898)

[The Pieces 3](#_Toc318574899)

[What Presto Does 4](#_Toc318574900)

[What Presto Doesn’t Do 4](#_Toc318574901)

[Sample Setup and Deployment – Screenshots 4](#_Toc318574902)

[Installation 9](#_Toc318574903)

[RavenDB 9](#_Toc318574904)

[Presto Task Runner 9](#_Toc318574905)

[Presto Dashboard 10](#_Toc318574906)

[Setting up Applications 10](#_Toc318574907)

[Setting up Servers 10](#_Toc318574908)

[Creating and Using Custom Variables 10](#_Toc318574909)

[Deployments 10](#_Toc318574910)

[Initiating Deployments 10](#_Toc318574911)

[Monitoring Deployments 10](#_Toc318574912)

# Overview

## What is Presto?

Presto is an automated deployment tool. It allows users to create deployment steps and then have servers update themselves when new versions of an app are ready. This allows for automated deployments to development, QA, and production environments.

The main reason to use Presto is when you have applications that need to be installed on many servers, in various environments, repeatedly, and each server requires a slightly different version of the app. That is Presto’s sweet spot.

## The Deployment Process

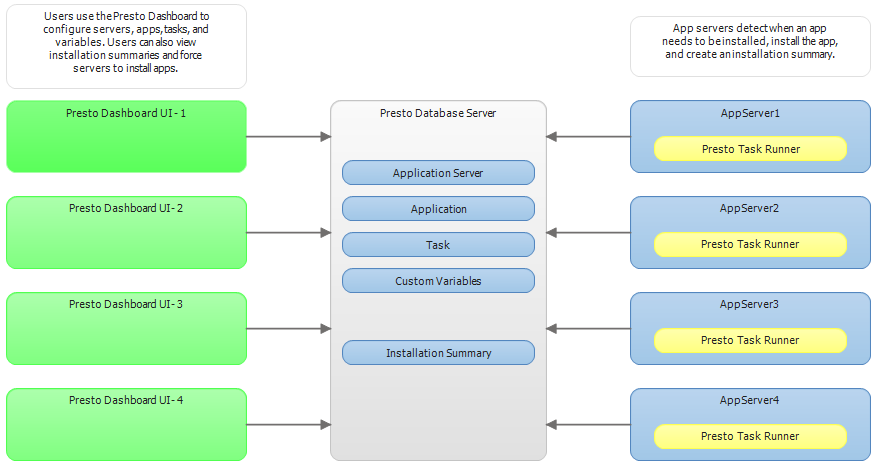
1. Within the Presto Dashboard, set up applications, servers, and variables.
2. When a new version of an application is ready to be deployed, copy the binaries to a common location.
3. Within the Presto Dashboard, initiate a deployment.
4. To deploy new versions of an application, go to step two.

## What are the benefits of Presto?

* Deployment is automated; manual error is significantly reduced.
* Logging
  + When users add an application to a server
  + When users initiate a deployment
  + Each deployment: server, application, start time, end time, result/status
  + Installation details of each task when Presto installs an app
* Flexibility: Variables are created and used by applications and servers
* Ease of use
  + Deploy an application (or many apps) to a specific server
  + Deploy an application to all servers within an environment (i.e.: QA)
  + Presto will update itself on all servers when a new version is available

## The Pieces

* **Presto Dashboard**: This is the application with which the user administers Presto. The user will create applications (and their tasks), set up servers, and create/use custom variables.
* **Presto Task Runner**: This is the actual deployment engine, running as a service on all servers. It polls the database to detect when an installation needs to occur.
* **Database**: RavenDB



## What Presto Does

* Modify XML documents; usually configuration files
* Copy files
* Command prompt tasks. Examples:
  + Create a Windows service
  + Configure and start a windows service
  + Stop a clustered resource
  + Execute a batch file
  + Anything else that can be done at a command prompt

In a nutshell, Presto performs the three main tasks listed above. The power is in the third feature: command prompt tasks. Basically, anything that can be done at a command prompt can be automated with Presto.

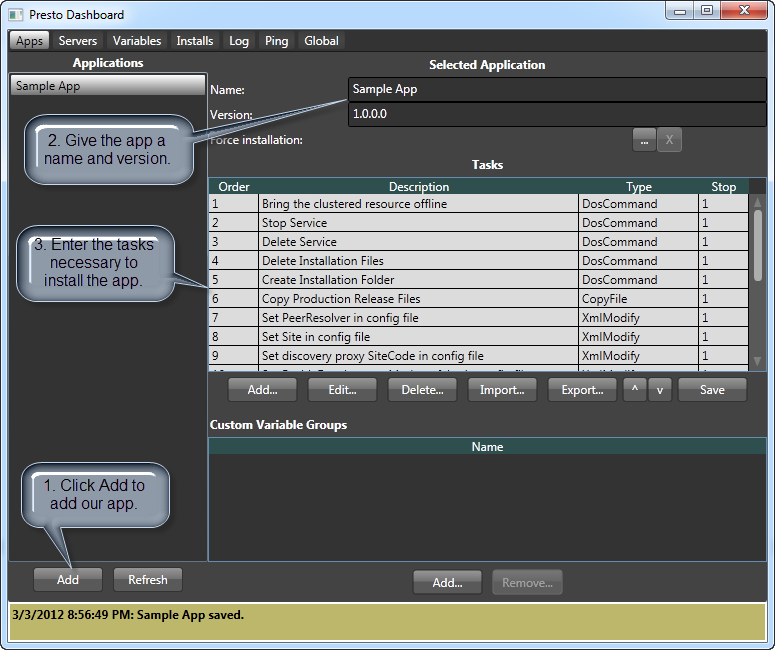
## What Presto Doesn’t Do

* Deploy ClickOnce apps
* Deploy database schema and scripts

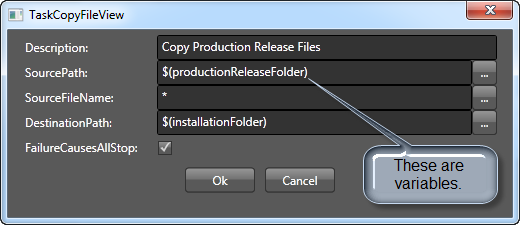
## Sample Setup and Deployment – Screenshots

Note: These screenshots are here for overview reasons, to see how things come together. For a comprehensive explanation of each of the steps, see the remainder of this guide.

Adding the application in the Presto Dashboard:

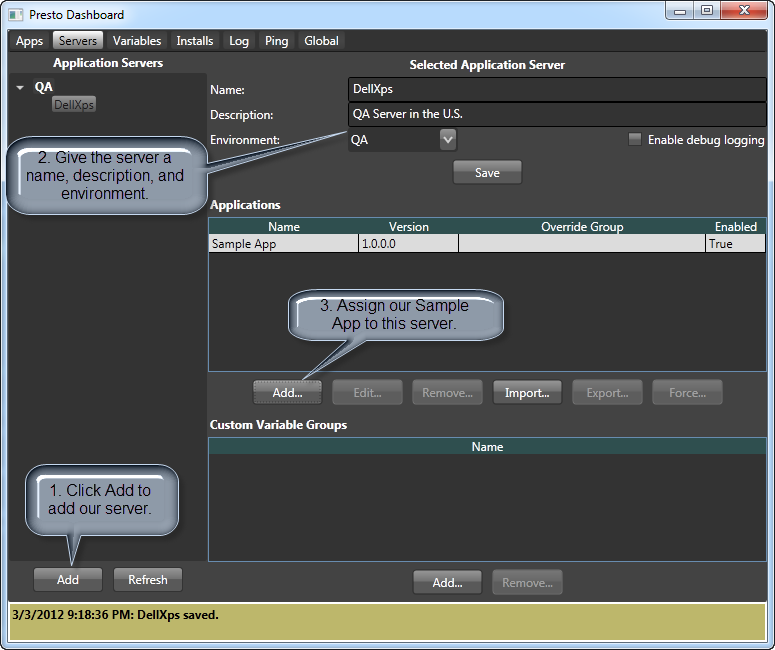


This is an example of one of the tasks (task 6 above, Copy Production Release Files). Here, we’re simply copying the app’s binaries from our production release folder to the installation folder:

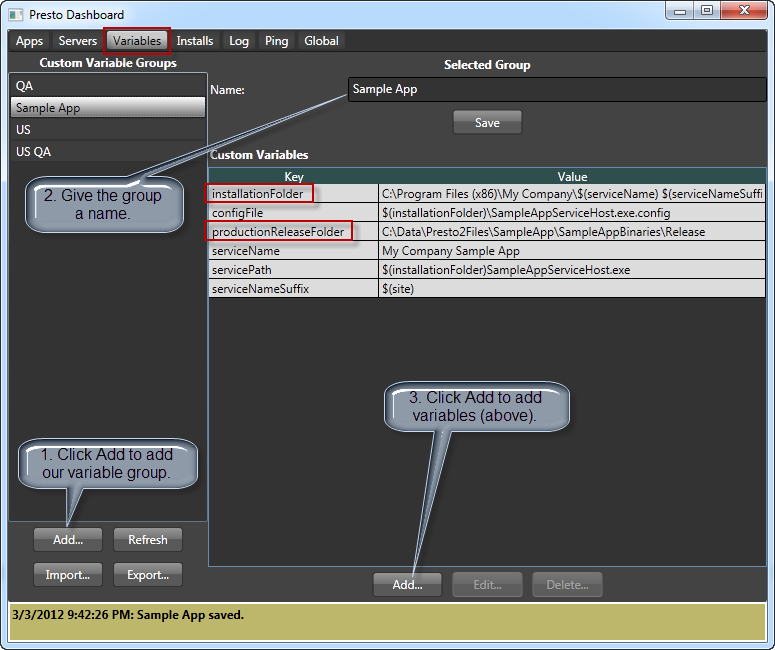


Note: Don’t worry about the variables. We’ll get to those below. That variable will resolve to whatever is set as the production release folder; something like: [\\SomeServer\ProductionBuilds\SampleApp\Release\](file:///\\SomeServer\ProductionBuilds\SampleApp\Release\).

Now that we have our application, let’s create a server on which to install it.

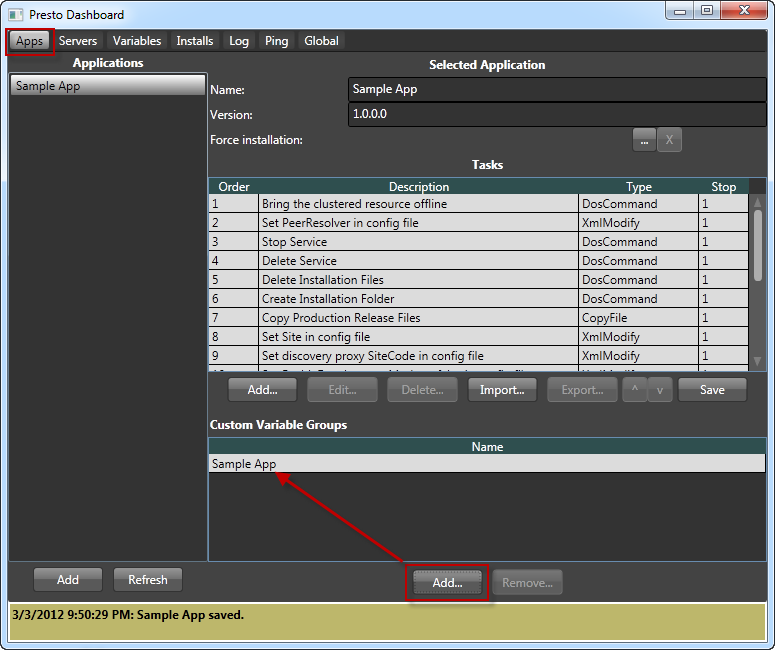


Before being able to deploy our app, we need to add our custom variables.

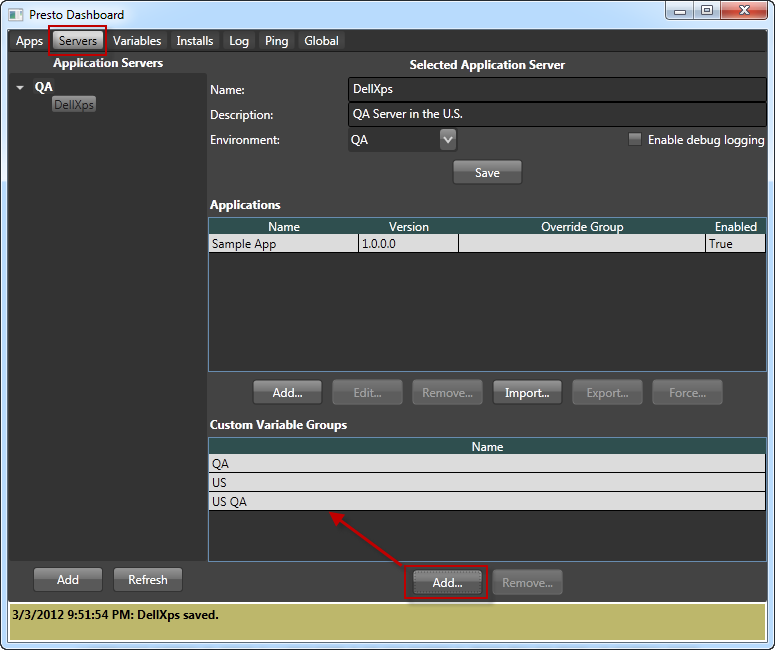


Remember our productionReleaseFolder and installationFolder variables from our task that we created a few minutes ago? This is where they’re defined.

Now we just need to assign our custom variables to the app and server. Here’s the app:



And here’s the server:



# Installation

## RavenDB

Presto currently uses build 573 of RavenDB, found here:

<http://builds.hibernatingrhinos.com/builds/RavenDB>

Download RavenDB-Build-573 and save it on the server’s (what will be your DB server) hard disk.

Unzip...

## Presto Task Runner

The Presto Task Runner (PTR) needs to be installed on every server that participates in automated deployment.

## Presto Dashboard

Dashboard…

# Setting up Applications

# Setting up Servers

# Creating and Using Custom Variables

# Deployments

## Initiating Deployments

## Monitoring Deployments