

# Commandbox

- [Commandbox Install in Windows](#)
- [Commandbox Commands](#)
- [Commandbox Install in Ubuntu](#)

# Commandbox Install in Windows

1. Download CommandBox from <https://www.ortussolutions.com/products/commandbox>
2. Extract downloaded zip to directory of your choice (Ex: c:\commandbox)
3. Ensure you have Java installed (Amazon Coretto at <https://docs.aws.amazon.com/corretto/latest/corretto-8-ug/downloads-list.html> works well)
4. Right-click on Start Menu --> Click on "System", under "Find a settings" search box search for "Advanced System" and then click on "View Advanced System Settings" result. Click on the "Environment Variables" button, under "System Variables" locate and select "Path" and click on "Edit". Click "New" and enter an entry pointing to "c:\commandbox" or whatever directory you extracted commandbox from Step 2 above.
5. Open an Administrator command prompt and enter "box" to initialize CommandBox.
6. Next to navigate to c:\users\username\.CommandBox and create a commandbox.properties file with the following content (Assuming c:\commandbox is the directory you extracted CommandBox in Step 2 above. And YES the double slash in c:\\commandbox is NOT a typo):

```
commandbox_home=c: \\commandbox
```

7.

# Commandbox Commands

## Install Coldbox MVC and Deploy App

```
box mkdir /opt/hermes/webapps/Coldbox-Hermes
cd /opt/hermes/webapps/Coldbox-Hermes
box install coldbox
box coldbox create app HermesSEG
```

## Start Server Arguments:

```
Numeric port (port number)
String host (bind to a host/ip)
Boolean openbrowser (open a browser after starting)
String directory (web root for this server)
Numeric stopPort (stop socket listener port number)
Boolean force = "false" (force start if status is not stopped)
Boolean debug (Turn on debug output while starting and stream server output to console.)
String webConfigDir (custom location for web context configuration)
String serverConfigDir (custom location for server configuration)
String libDirs (comma-separated list of extra lib directories for the server to load)
String trayIconFile (path to .png file for tray icon)
String webXML (path to web.xml file used to configure the server)
Boolean HTTPEnable (enable HTTP)
Boolean SSLEnable (enable SSL)
Numeric SSLPort (SSL port number)
String SSLCertFile (Path to SSL certificate file)
String SSLKeyFile (Path to SSL key file (required if SSLCert specified))
String SSLKeyPass (SSL key passphrase)
Boolean rewritesEnable (enable URL rewriting (default false))
String rewritesConfig (optional URL rewriting config file path)
Numeric heapSize (The max heap size in megabytes you would like this server to start with, it defaults to 512mb)
Numeric minHeapSize (The min heap size in megabytes you would like this server to start with)
Boolean directoryBrowsing (Enables directory browsing (default false))
```

String JVMArgs (Additional JVM args to use when starting the server. Use "server status --verbose" to debug)

String runwarArgs (Additional Runwar options to use when starting the server. Use "server status --verbose" to debug)

Boolean saveSettings = "true" (Save start settings in server.json)

String cfengine (sets the cfml engine type)

String WARPath (sets the path to an existing war to use)

String serverConfigFile (The path to the server's JSON file. Created if it doesn't exist.)

Numeric startTimeout (The amount of time in seconds to wait for the server to start (in the background).)

Boolean console (Start this server in the foreground console process and wait until Ctrl-C is pressed to stop it.)

String welcomeFiles (A comma-delimited list of default files to load when visiting a directory (index.cfm, index.htm, etc))

String serverHomeDirectory (The folder where the CF engine WAR should be extracted)

String restMappings (A comma-delimited list of REST mappings in the form of /api/\*, /rest/\*. Empty string to disable.)

Boolean trace (Enable trace level logging)

String javaHomeDirectory (Path to the JRE home directory containing ./bin/java)

Boolean AJPEnable (Enable AJP)

Numeric AJPPort (AJP Port number)

String javaVersion (Any endpoint ID, such as "java:openjdk11" from the Java endpoint)

String startScript (If you want to generate a native script to directly start the server process pass bash, cmd, or pwsh)

String startScriptFile (Optional override for the name and location of the start script. This is ignored if no startScript param is specified)

Boolean dryRun (Abort actually starting the server process, but all installation and downloading will still be performed to "warm up" the engine e installation.)

Boolean verbose (Activate extra server start information without enabling the debug mode in the actual server (which you wouldn't want in production))

Boolean trayEnable (Enable the system tray icon/menu)

String profile (Controls default server settings. Profiles: production, development, none)

String blockCFAdmin (Block access to Lucee or ACF admin. Valid values are true, false, external)

## Start Server

```
box server start host=0.0.0.0 port=8282 --rewritesEnable
```

## Start Server with Lucee Admin Enabled

```
box server start host=0.0.0.0 port=8282 BlockCFAdmin=false --rewritesEnable
```

## Start Server with Lucee Admin Enabled & Custom Directories

```
box server start host=0.0.0.0 port=8282 BlockCFAdmin=false  
webConfigDir=/opt/hermes/webapps/Lucee/web serverConfigDir=/opt/hermes/webapps/Lucee/server  
serverHomeDirectory=/opt/hermes/webapps/Lucee CommandBox_home=/opt/hermes/webapps/Commandbox  
--rewritesEnable
```

## Start Server with Lucee Admin Enabled, Custom Directories and Admin Password Set

```
box server start host=0.0.0.0 port=8282 BlockCFAdmin=false  
webConfigDir=/opt/hermes/webapps/Lucee/web serverConfigDir=/opt/hermes/webapps/Lucee/server  
serverHomeDirectory=/opt/hermes/webapps/Lucee adminPassword=password --rewritesEnable
```

## Stop Server

```
box server stop
```

## Deploy Coldbox App

```
box coldbox create app myApp
```

# Create handler

```
CommandBox> coldbox create handler helloWorld index, add, edit, list
```

# Create Virtual Event

A virtual event is a view that can be executed with no event handler controller which is a good way to incorporate non-mvc files into ColdBox.

```
coldbox create view name="virtual/hello"
```

# Re-initialize Framework

```
http://IP_ADDRESS:8282/?fwreinit=1
```

# Install Packages

Full list of packages/modules available on forgebox: <https://forgebox.io/type/modules>

Below is a list of helpful examples:

- **BCrypt** -- Industry-standard password hashing
- **cbdebugger** -- For debugging Coldbox apps
- **cbjava-loader** - For interacting with Java classes and libraries
- **cbMarkdown** - For writing in markdown
- **cbMessagebox** -- Display nice error/success messages
- **cborm** -- Awesome ORM Services
- **cb18n** -- For multilingual sites
- **cbt** - ColdBox templating language
- **cbValidation** - Back-end validation framework
- **qb** - Fluent query builder and schema builder
- **route-visualizer** - For visualizing your application routes

# Install package from Commandbox prompt

```
install cbmessagebox
```

# Install package from root prompt

```
box install cbmessagebox
```

```
Commandbox> install cbmessagebox
```

```
root@localhost: box install cbmessagebox
```

# Commandbox Install in Ubuntu

Run the following command in CLI prompt to install Commandbox:

```
/usr/bin/curl -fsSl https://downloads.ortussolutions.com/debs/gpg | sudo apt-key add - && echo  
"deb https://downloads.ortussolutions.com/debs/noarch /" | sudo tee -a  
/etc/apt/sources.list.d/commandbox.list && /usr/bin/apt update && /usr/bin/apt install apt-  
transport-https commandbox -y
```

Run the following command in CLI prompt to initialize Commandbox for the first time to install Commandbox in the default user's home directory:

```
/usr/local/bin/box exit
```

Run the following command in CLI prompt to initialize Commandbox for the first time to install Commandbox in a specified directory (/opt/hermes/webapps/Commandbox):

```
/usr/local/bin/box -commandbox_home=/opt/hermes/webapps/Commandbox exit
```