

[Back to Main Extension Provider Tutorial](#)

Writing an extension - Part 4 - The installer

After we have created and validated the configuration data, we should now finally install the software. The install method gives you the possibility to do nearly whatever you like within the Railo boundaries. In order to have some support methods available you can extend some existing helper CFC's. These helper CFC's provide some methods to help you create Mappings, Datasources etc. In my particular case I am using the `installFolderMapping.cfc` helper class. It has the following super methods:

- install
- update

`installFolderMapping.cfc` extends the CFC `InstallSupport.cfc` which comes with the following methods:

- updateMapping
- unzip
- removeMapping
- throwErrors

We will publish all helper components with all kinds of support methods so that you only have to extend the corresponding ones in order to have the necessary tools at hand. The install method of the CFC `install.cfc` gets invoked as soon as the last page of the `config.xml` form is validated properly. It takes three arguments:

- error
- path
- config

The important ones are path and config. The struct error can be filled in order to throw errors upon install. When you receive the config struct please note that it contains an array with all elements of the corresponding steps and a common struct called mixed that merges all the variables into one struct. If you have ambiguous variables (which you anyway should prevent) you can address them over the step array. Now let's look at the install method:

```
name="install" returnType="string" output="yes" hint="called from Railo to install application">
  name="error" type="struct">
  name="path" type="string">
  name="config" type="struct">
  var sReturn = "">
  <--- unzip the necessary files --->
  var message=super.install(argumentCollection=arguments)>
  sReturn = createDatabaseTables(argumentCollection=arguments)>
  len(trim(sReturn)) eq 0>
  variable="sReturn">
```

Done!

```
You can now start posting from your administration at: href=
"#config.mixed.blog_address#/admin/index.cfm?first=1"
]#contractPath(config.mixed.destination_path)#/admin/
```

```
Then you can view your blog at: href="#config.mixed.blog_address#" target=
"_blank">#contractPath(config.mixed.destination_path)#
```

```
sReturn>
```

The installer first calls the `super.install` method that will unzip the files into the destination path. Here's the `super.install` method:

```
name="install" returnType="string" output="no" hint="called from Railo to install application">
  name="error" type="struct">
  name="path" type="string">
  name="config" type="struct">
  <--- unzip to destination path --->
  action = "unzip" destination = "#config.mixed.destination_path#" file = "#path#content.zip" overwrite = "yes" recurse = "yes" storePath = "yes">
  "">
```

Nothing special here. The next thing that happens then is that the method `CreateDatabaseTables` is called.

At the end we get the result that Mango has been installed and that we could either follow the link to the administrator or the website itself. We have SUCCEEDED!!! This Method creates the

database tables and a datasource in Railo if necessary. So all is in your hand. You can use the tag in order to create necessary entries in the Railo config. The complete extension can be downloaded [here](#).

When the update and uninstall buttons are clicked the corresponding methods get invoked. They always get the complete configuration as an argument, as it was entered when the application was originally installed or updated. So the uninstall method looks like this:

```
name="uninstall" returntype="string" output="no" hint="called by Railo to uninstall the application">
  name="path" type="string">
  name="config" type="struct">
    arguments.config.path = arguments.config.mixed.destination_path>
  <--- remove the files --->
  deleteFiles(argumentCollection=arguments)>
  <--- remove the datasource & the tables --->
  dropDatabaseTables(argumentCollection=arguments)>
  "Mango blog has been successfully removed. Tables have beend dropped and the datasource and the corresponding files have been removed.">
```

Please note that you cannot delete the complete install path if the original install path was the webroot itself. There might be WEB-INF directories in there that either must not be deleted or are not deletable because they are protected (open for read). So mostly the installation is up to you so that you have full control over everything what happens.

The mango blog extension is available from the extension provider - <http://preview.railo.ch/ExtensionProvider.cfc> - if you like to test it. As soon as the Mango team approves the installer we will officially place it on www.getrailo.org.

[Back to Main Extension Provider Tutorial](#)