

[<< Back to Dashboard](#)

# Exploring the BeanFactory Plugin

## Overview

I am a plugin that taps into WireBox or the ColdBox Compat BeanFactory.

**Important:** Please note that this plugin just proxies calls mostly to WireBox or our compatibility cf7 bean factory plugin (BeanFactoryCompat) that will be deprecated by 3.1. Use this plugin for populations or backwards compatibility, but we encourage you to access WireBox directly.

## autowire

Autowire an object using the ColdBox DSL

## Returns

- This function returns *void*

## Arguments

| Key                | Type | Required | Default       | Description  |
|--------------------|------|----------|---------------|--|
| target             | any  | Yes      | ---           | The object to autowire   |
| useSetterInjection | any  | No       | true          | Whether to use setter injection alongside the annotations property injection. cfproperty injection takes precedence.                               |
| annotationCheck    | any  | No       | false         | Boolean Value<br>false, it will only autowire if that metadata attribute is set to true. The default is false, which will autowire automatically.  |
| onDICCompleteUDF   | any  | No       | onDICComplete | Boolean Value<br>After Dependencies are injected, this method will look for this UDF and call it if it exists. The default value is onDICComplete. |

## Contents

- [Exploring the BeanFactory Plugin](#)
  - [Overview](#)
  - [autowire](#)
    - [Returns](#)
    - [Arguments](#)
    - [Examples](#)
  - [populateFromXML](#)
    - [Returns](#)
    - [Arguments](#)
    - [Examples](#)
  - [getBeanFactory](#)
    - [Returns](#)
    - [Examples](#)
  - [populateFromStruct](#)
    - [Returns](#)
    - [Arguments](#)
    - [Examples](#)
  - [removeExternalLocations](#)
    - [Returns](#)
    - [Arguments](#)
    - [Examples](#)
  - [addModelMapping](#)
    - [Returns](#)
    - [Arguments](#)
    - [Examples](#)
  - [populateModel](#)
    - [Returns](#)
    - [Arguments](#)
    - [Examples](#)
  - [configure](#)
    - [Returns](#)
    - [Examples](#)
  - [populateFromQueryWithPrefix](#)
    - [Returns](#)
    - [Arguments](#)
    - [Examples](#)
  - [getModel](#)
    - [Returns](#)
    - [Arguments](#)
    - [Examples](#)
  - [locateModel](#)
    - [Returns](#)
    - [Arguments](#)
    - [Examples](#)
  - [appendExternalLocation](#)
    - [Returns](#)
    - [Arguments](#)
    - [Examples](#)
  - [URF](#)
    - [Returns](#)
    - [Arguments](#)
    - [Examples](#)



## Returns

- This function returns *any*

## Examples

# populateFromStruct

Populate a named or instantiated bean from a structure

## Returns

- This function returns *any*

## Arguments

| Key           | Type    | Required | Default | Description   |
|---------------|---------|----------|---------|---|
| target        | any     | Yes      | ---     | This can be an instantiated bean object or a bean instantiation path as a string. If you pass an instantiation path and the bean has an 'init' method. It will be executed. This method follows the bean contract (set{property_name}). Example: setUsername(), setName() |
| memento       | struct  | Yes      | ---     | The structure to populate the object with.  |
| scope         | string  | No       | ---     | Use scope injection instead of setters population.  |
| trustedSetter | boolean | No       | false   | If set to true, the setter method will be called even if it does not exist in the bean  |
| include       | string  | No       |         | A list of keys to include in the population   |
| exclude       | string  | No       |         | A list of keys to exclude in the population   |

## Examples

# removeExternalLocations

Try to remove all the external locations passed in. @deprecated by 3.1

## Returns

- This function returns *void*

## Arguments

| Key       | Type | Required | Default | Description   |
|-----------|------|----------|---------|---|
| locations | any  | Yes      | ---     | Locations to remove from the lookup. Comma delimited allowed. |

## Examples

## addModelMapping

Add a new model mapping. Ex: `addModelMapping('myBean','security.test.FormBean')`. The alias can be a single item or a comma delimited list. @deprecated by 3.1

### Returns

- This function returns *void*

### Arguments

| Key   | Type | Required | Default | Description   |
|-------|------|----------|---------|---|
| alias | any  | No       | ---     | The model alias to use, this can also be a list of aliases. Ex: <code>SecurityService,Security</code> |
| path  | any  | Yes      | ---     | The model path (From the model conventions downward). Do not add full path, this is a convenience     |

## Examples

## populateModel

Populate a named or instantiated model (java/cfc) from the request collection items

### Returns

- This function returns *Any*

### Arguments

| Key           | Type    | Required | Default | Description   |
|---------------|---------|----------|---------|---|
| model         | any     | Yes      | ---     | The name of the model to get and populate or the actual model object. If you already have an instance of a model, then use the <code>populateBean()</code> method |
| scope         | string  | No       |         | Use scope injection instead of setters population. Ex: <code>scope=variables.instance</code> .  |
| trustedSetter | boolean | No       | false   | If set to true, the setter method will be called even if it does not exist in the bean  |

|         |        |    |   |
|---------|--------|----|---|
| include | string | No | A list of keys to include in the population |
| exclude | string | No | A list of keys to exclude in the population |

## Examples

## configure

Configure the bean factory for operation. @deprecated

### Returns

- This function returns *BeanFactory*

## Examples

## populateFromQueryWithPrefix

Populates an Object using only specific columns from a query. Useful for performing a query with joins that needs to populate multiple objects.

### Returns

- This function returns *any*

### Arguments

| Key           | Type    | Required | Default | Description   |
|---------------|---------|----------|---------|---|
| target        | any     | Yes      | ---     | This can be an instantiated bean object or a bean instantiation path as a string. If you pass an instantiation path and the bean has an 'init' method. It will be executed. This method follows the bean contract (set{property_name}). Example: setUsername(), setName() |
| qry           | query   | Yes      | ---     | The query to populate the bean object with  |
| RowNumber     | Numeric | No       | 1       | The query row number to use for population  |
| scope         | string  | No       |         | Use scope injection instead of setters population. Ex: scope=variables.instance.  |
| trustedSetter | boolean | No       | false   | If set to true, the setter method will be called even if it does not exist in the bean  |
| include       | string  | No       |         | A list of keys to include in the population   |
| exclude       | string  | No       |         | A list of keys to exclude in the population   |

|        |        |     |     |   |
|--------|--------|-----|-----|---|
| prefix | string | Yes | --- | The prefix used to filter, Example: 'user_' would apply to the following columns: 'user_id' and 'user_name' but not 'address_id'. |
|--------|--------|-----|-----|---|

## Examples

## getModel

Create or retrieve model objects by convention.

### Returns

- This function returns *any*

### Arguments

| Key                | Type | Required | Default | Description  |
|--------------------|------|----------|---------|--|
| name               | any  | No       |         | The name of the model to retrieve  |
| useSetterInjection | any  | No       | ---     | Whether to use setter injection alongside the annotations property injection. cfproperty injection takes precedence. Boolean     |
| onDICCompleteUDF   | any  | No       | ---     | After Dependencies are injected, this method will look for this UDF and call it if it exists. The default value is onDICComplete |
| stopRecursion      | any  | No       | ---     | A comma-delimited list of stoprecursion classpaths.  |
| dsl                | any  | No       | ---     | The dsl string to use to retrieve the domain object  |
| executeInit        | any  | No       | true    | Whether to execute the init() constructor or not. Defaults to execute, Boolean   |
| initArguments      | any  | No       | ---     | The constructor structure of arguments to passthrough when initializing the instance. Only available for WireBox integration     |

## Examples

## locateModel

Get the location instantiation path for a model object. If the model location is not found, this method returns an empty string. @deprecated by 3.1

## Returns

- This function returns *string*

## Arguments

| Key          | Type | Required | Default | Description           |
|--------------|------|----------|---------|-----------------------|
| name         | any  | Yes      | ---     | The model to locate   |
| resolveAlias | any  | No       | false   | Resolve model aliases |

## Examples

## appendExternalLocations

Try to append a new model external location. @deprecated by 3.1

## Returns

- This function returns *void*

## Arguments

| Key       | Type | Required | Default | Description   |
|-----------|------|----------|---------|---|
| locations | any  | Yes      | ---     | Locations to add to the lookup, will be added in passed order. Comma delimited allowed. |

## Examples

## populateBean

Populate a named or instantiated bean (java/cfc) from the request collection items

## Returns

- This function returns *Any*

## Arguments

| Key    | Type | Required | Default | Description   |
|--------|------|----------|---------|---|
| target | any  | Yes      | ---     | This can be an instantiated bean object or a bean instantiation path as a string. This method follows the bean contract (set{property_name}). Example: setUsername(), setName() |

|               |         |    |       |  |
|---------------|---------|----|-------|--|
| scope         | string  | No |       | Use scope injection instead of setters population. Ex: scope=variables.instance.       |
| trustedSetter | boolean | No | false | If set to true, the setter method will be called even if it does not exist in the bean |
| include       | string  | No |       | A list of keys to include in the population  |
| exclude       | string  | No |       | A list of keys to exclude in the population  |

## Examples

### populateFromQuery

Populate a named or instantiated bean from query

#### Returns

- This function returns *any*

#### Arguments

| Key           | Type    | Required | Default | Description   |
|---------------|---------|----------|---------|---|
| target        | any     | Yes      | ---     | This can be an instantiated bean object or a bean instantiation path as a string. If you pass an instantiation path and the bean has an 'init' method. It will be executed. This method follows the bean contract (set{property_name}). Example: setUsername(), setName() |
| qry           | query   | Yes      | ---     | The query to populate the bean object with  |
| RowNumber     | Numeric | No       | 1       | The query row number to use for population  |
| scope         | string  | No       |         | Use scope injection instead of setters population. Ex: scope=variables.instance.  |
| trustedSetter | boolean | No       | false   | If set to true, the setter method will be called even if it does not exist in the bean  |
| include       | string  | No       |         | A list of keys to include in the population   |
| exclude       | string  | No       |         | A list of keys to exclude in the population   |

## Examples

### populateFromJSON

Populate a named or instantiated bean from a json string

## Returns

- This function returns *any*

## Arguments

| Key           | Type    | Required | Default | Description   |
|---------------|---------|----------|---------|---|
| target        | any     | Yes      | ---     | This can be an instantiated bean object or a bean instantiation path as a string. If you pass an instantiation path and the bean has an 'init' method. It will be executed. This method follows the bean contract (set{property_name}). Example: setUsername(), setName() |
| JSONString    | string  | Yes      | ---     | The JSON string to populate the object with. It has to be valid JSON and also a structure with name-key value pairs.  |
| scope         | string  | No       |         | Use scope injection instead of setters population. Ex: scope=variables.instance.  |
| trustedSetter | boolean | No       | false   | If set to true, the setter method will be called even if it does not exist in the bean  |
| include       | string  | No       |         | A list of keys to include in the population   |
| exclude       | string  | No       |         | A list of keys to exclude in the population   |

## Examples