

[<< Back to Dashboard](#) | [<< Projects Viewer](#)

# Lookup Manager

The ColdBox lookup manager is a handy utility used to manage a set of tables and their relationships. This in turn is a dynamic scaffolding tool based on ColdBox and Transfer ORM. More ORMs will be supported in the future.

## Requirements

- ColdBox 2.6.4 or greater
- Transfer 1.1 or greater

## Installation

Just Install the parts in their appropriate locations of your application or just copy the entire application and use it to begin a new project.

```

/Config
- coldbox.xml (Use this base coldbox.xml or use the settings in the file:StandAloneSettings.xml)
- StandAloneSettings.xml ( Copy the settings here to your own coldbox.xml )
- modelMappings (By default it uses model integration, so just copy the alias)
/handlers
- lookups.cfc
/includes
/lookups/*
/layouts
-Layout.Lookups.cfm
/model
/lookups/*
/views
/lookups/*

```

That's it. Once you have dropped the appropriate files in your application you are ready to start configuring the lookup manager for operation.

## Generating the code

If you are using Railo 3.1 or ColdFusion 8 or greater you can generate the appropriate lookup information using our handy wizard. Just go to the lookups main event and you will be presented with the main lookup screen. From here just click on the button.

## Decorator Methods

You can create some decorator methods in order to interact with the lookup manager:

- **get/set TableConfig**: To get and set the table of configuration data, saved as *instance.tableConfig*
- **validate():Array**: Method to know if the object' has been populated accordingly.
- **configure**: To configure the table config.

## Table Configuration Data

The table config structure contains some extra metadata on how to render the table in the lookup manager.

Key	Required	Description
SortBY	TRUE	The column to sort on by default
{column}.html	FALSE	The html element to use. For boolean types use: <b>radio or select (default)</b> . For other types (string,date,UUID,GUID) use: <b>text (default), password, textarea, richtext (if implemented)</b>
{column}.display	FALSE (Default:true)	Whether to display the column or not, use: <b>true or false</b>
{column}.validate	FALSE	The type of validation to do according to the validation JS library. Default is no validation.
{column}.maxlength	FALSE	The max length of the field in the database.
{column}.helptext	FALSE	The help text to show on the form
{m2o relation name}.displayColumn	TRUE	If the object has a m2o relation, then choose the column to use in the select box

## Contents

- [Lookup Manager](#)
- [Requirements](#)
- [Installation](#)
- [Generating the code](#)
- [Decorator Methods](#)
  - [Table Configuration Data](#)
  - [Validations](#)

## **Validations**

The valid masks are

- email
- domain
- phone
- zip
- numeric
- image (gif,jpg,bmp,png,jpeg)
- media (avi,mov,mpeg,wmv)
- pdf