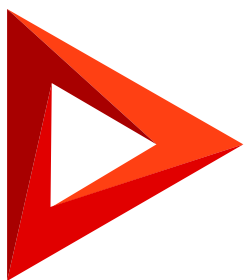


Existing additional feature

Manage an existing additional feature

Version 8.0



This documentation is provided under restrictions on use and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this documentation, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

Table of Contents

Manage an existing additional feature	4
Store the details of an additional feature	4
Open the Feature toggling page	5
Manage the status of an additional feature	7
FeatureUtilities class	7
Methods	8
FeatureState enumeration	11

Manage an existing additional feature

Advanced

This article covers the behavior of additional features relevant to Creatio 8.0.2 Atlas and later. Instructions for an earlier Creatio version: [Feature Toggle mechanism](#).

Feature toggle is a software development technique that lets you manage the status of additional features in a live application. Learn more: [Wikipedia](#). The [*Feature toggling*] Creatio page implements the feature toggle functionality.

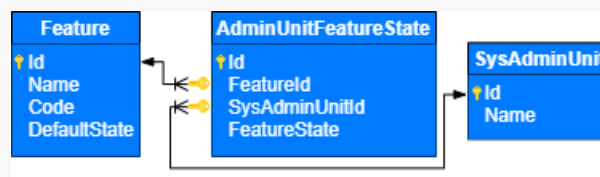
Feature toggle lets you use continuous integration while preserving the working capacity of the application and hiding features you are still developing. Instructions: [Implement a custom additional feature](#).

Store the details of an additional feature

Creatio stores the details of an additional feature in the following **database tables**:

- [Feature] . Stores the list of features you can toggle. Empty by default.
- [AdminUnitFeatureState] (the [FeatureState] column). Stores the status of an additional feature (turned on/off). The [AdminUnitFeatureState] table connects the [Feature] and [SysAdminUnit] database tables. [SysAdminUnit] is the status of an additional feature for Creatio users and user groups.

View the relationship chart between tables that store the details of an additional feature in the figure below.

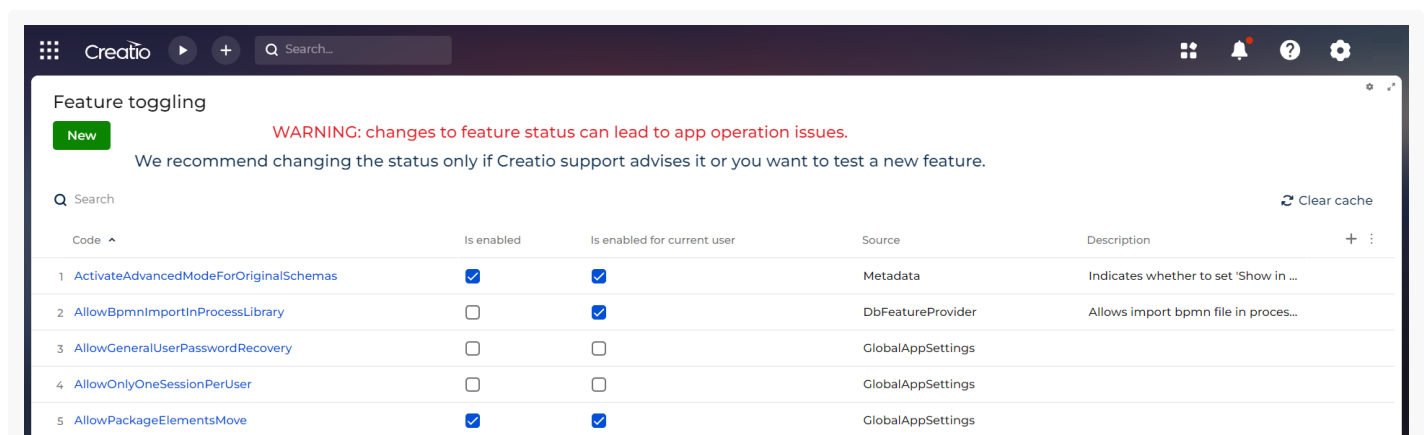


Primary columns of the database tables that store the details of an additional feature

Column	Type	Description
[Feature] database table		
[Id]	uniqueidentifier	The unique ID of the record.
[Name]	nvarchar(250)	The name of the additional feature. By default, matches the [Code] column value. You can change the name of the additional feature manually.
[Code]	nvarchar(50)	The code of the additional feature.
[DefaultState]	int	The default status of the additional feature (1 : turned on, 0 : turned off).
[AdminUnitFeatureState] database table		
[Id]	uniqueidentifier	The unique ID of the record.
[FeatureId]	uniqueidentifier	The unique ID of the record from the [Feature] database table.
[SysAdminUnitId]	uniqueidentifier	The unique ID of the user record.
[FeatureState]	int	The status of the additional feature (1 : turned on, 0 : turned off).

Open the [Feature toggling] page

To open the [Feature toggling] page, use the [Creatio URL]/0/Shell#Section/AppFeature_ListPage **link**. For example, http://my.creatio.com/0/Shell#Section/AppFeature_ListPage. As a result, the browser will open the [Feature toggling] page.



The [*Feature toggling*] page displays the following data:

- **code of the additional feature** (the [*Code*] column)
- **status of the additional feature for all users** (the [*Is enabled*] column)
- **status of the additional feature for the current user** (the [*Is enabled for current user*] column)
- **source of the additional feature** (the [*Source*] column)
- **description of the additional feature** (the [*Description*] column)

The **source of the additional feature** (the [*Source*] column) can have the following types:

- `DbFeatureProvider` . Records of the [*Feature*] database table.
- `web.config` . Boolean key values from the `<appSettings>` section of the `web.config` file in Creatio root directory.
- `Metadata` . Additional features registered in the source code using the class that inherits from `FeatureMetadata` .
- `GlobalAppSettings` . Boolean properties of the `Terrasoft.Core.GlobalAppSettings` class. Learn more: [.NET class library](#).

The types of additional feature source are sorted by priority in descending order.

The **rules** that determine the status of the additional feature are as follows:

- If an additional feature is not registered in Creatio, it is disabled.
- If the status of an additional feature is specified in multiple sources, the resulting status is based on the source priority. Back-end changes to the status do not require you to refresh the browser page. Refresh the page to apply the front-end changes to the status.

View the examples of the additional feature status (`true` : turned on, `false` : turned off) determined based on multiple sources below.

Examples that define the status of the additional feature

Feature source			Resulting feature status
<code>DbFeatureProvider</code>	<code>web.config</code>	<code>Metadata</code> or <code>GlobalAppSettings</code>	
<code>true</code>	<code>false</code>	<code>false</code>	<code>true</code>
<code>false</code>	<code>true</code>	<code>false</code>	<code>false</code>
Not available	<code>true</code>	<code>false</code>	<code>true</code>
Not available	<code>false</code>	<code>true</code>	<code>false</code>

Manage the status of an additional feature

Attention. Changes to the status of an additional feature can affect Creatio operability. We recommend changing the feature status only if the support team approves it or you need to test a custom feature.

You can manage the status of an additional feature in multiple ways:

- **for a Creatio user group**
- **for all Creatio users**

Since version 8.0.2 Atlas, Creatio lets you manage the status of an additional feature using the **WorkspaceConsole utility**. Instructions: [Delivery in WorkspaceConsole](#).

Change the status of an additional feature for user groups

Since version 8.0.2 Atlas, Creatio lets you change the status of an additional feature for user groups.

To change the status of an additional feature for user groups:

1. **Open the [Feature toggling] page.** Instructions: [Open the \[Feature toggling \] page](#).
2. **Open the page of the additional feature** whose status to change for the user group.
3. If needed, **add the user group** to the [Feature status for the system administration object] expanded list.
 - a. Click the **+** button on the [Feature status for the system administration object] expanded list.
 - b. Select the corresponding user group in the [Admin unit] field.
4. **Open the user group page.**
5. **Select or clear the [Feature state] checkbox.**
6. **Save the changes.**
7. **Refresh the page of the additional feature** to apply the changes.

Attention. If the additional feature is turned on for all users and turned off for a user group, Creatio turns it off only for the user group.

Change the status of an additional feature for all users

1. **Open the [Feature toggling] page.** Instructions: [Open the \[Feature toggling \] page](#).
2. **Open the page of the additional feature** whose status to change for all users.
3. **Select or clear the [Is enabled] checkbox.**
4. **Save the changes.**

FeatureUtilities class C#



The `Terrasoft.Configuration` namespace.

The `Terrasoft.Configuration.FeatureUtilities` class provides a set of extension methods to the `UserConnection` class. These methods let you use the `Feature Toggle` functionality in the source code schemas of Creatio back-end. The `FeatureUtilities` class also declares the enumeration of feature statuses (the `[FeatureState]` column of the `[AdminUnitFeatureState]` database table).

Note. Use the [.NET class reference](#) to access the full list of the constructors, methods and properties, parent classes, and implemented interfaces of the `UserConnection` class.

Methods

```
static int GetFeatureState(this UserConnection source, string code)
```

Returns the feature status.

Parameters

source	The user connection.
code	The feature code.

```
static int GetFeatureState(this UserConnection source, string code, Guid sysAdminUnitId)
```

Returns the feature status.

Parameters

source	The user connection.
code	The code of the feature.
sysAdminUnitId	The unique record ID.

```
static bool GetIsFeatureEnabledForAnyUser(this UserConnection source, string code)
```

Returns the feature status for any user.

Parameters

source	The user connection.
code	The code of the feature.

```
static bool GetIsFeatureEnabledForAllUsers(this UserConnection source, string code)
```

Returns the feature status for all users.

Parameters

source	The user connection.
code	The code of the feature.

```
static Dictionary<string, int> GetFeatureStates(this UserConnection source)
```

Returns all feature statuses.

Parameters

source	The user connection.
--------	----------------------

```
static List<FeatureInfo> GetFeaturesInfo(this UserConnection source)
```

Returns data about all features and their statuses.

Parameters

source	The user connection.
--------	----------------------

```
static void SetFeatureState(this UserConnection source, string code, int state, bool forAllUsers)
```

Sets the feature status.

Parameters

source	The user connection.
code	The code of the feature.
state	The new status of the feature.
forAllUsers	Sets the status of the feature for all users.

```
static void SafeSetFeatureState(this UserConnection source, string code, int state, bool forAll)
```

Sets the feature status or creates the feature if it does not exist.

Parameters

source	The user connection.
code	The code of the feature.
state	The new status of the feature.
forAllUsers	Sets the status of the feature for all users.

```
static void CreateFeature(this UserConnection source, string code, string name, string description)
```

Creates the feature.

Parameters

source	The user connection.
code	The code of the feature.
name	The name of the feature.
description	The description of the feature.

```
static bool GetIsFeatureEnabled(this UserConnection source, string code)
```

Checks if the feature is turned on.

Parameters

source	The user connection.
code	The code of the feature.

```
static bool GetIsFeatureEnabled(this UserConnection source, string code, Guid sysAdminUnitId)
```

Checks if the feature is turned off.

Parameters

source	The user connection.
code	The code of the feature.
sysAdminUnitId	The unique record ID.

FeatureState enumeration

The `FeatureState` enumeration defines the feature statuses (the `[FeatureState]` column of the `[AdminUnitFeatureState]` database table).

Disabled	0	The feature is turned off.
Enabled	1	The feature is turned on.
Established	2	The feature is established.