Cloning allows you to create exact copies of objects in the system. All objects support cloning, from the most basic records to complex data structures such as page types or user accounts.

Manually cloning objects

To clone an object in the Kentico administration interface:

1. Find the object on the listing page in the appropriate section of the UI.
2. Click **...** next to the object and select **Clone**.
   - For objects displayed in a tree menu instead of a standard list, use the clone button ( ) in the panel above the tree.
3. Configure the clone settings and click **Clone**.

You can clone object by calling the appropriate API in your custom code, for example in:

- Global event handlers
- User controls or web parts

The following sample code demonstrates how to clone objects using the API. The **CloneCountry** method in the example:

- Creates a clone of the USA country object, including all child states.
- Manually sets the country code values of the new object.

```
using CMS.Globalization;
using CMS.DataEngine;

...

public void CloneCountry()
{
    // Gets the country to be cloned.
    CountryInfo country = CountryInfoProvider.GetCountryInfo("USA");

    if (country != null)
    {
        // Prepares the settings used for the cloning process.
        CloneSettings settings = new CloneSettings()
        {
            // Sets the clone names.
            CodeName = "MyClonedCountry",
            DisplayName = "My Cloned Country",

            // Ensures that the state objects under the country are also cloned.
            IncludeChildren = true,

            // Registers a callback method that performs additional actions before the
clone is added.
            BeforeCloneInsertCallback = ChangeCountryCodes
        };

        // Clones the country according to the defined settings.
        country.Generalized.InsertAsClone(settings);
    }
}

private void ChangeCountryCodes(CloneSettings settings, BaseInfo cloneToBeInserted)
{
    // Changes the values of additional fields before the clone is inserted to the DB.
    CountryInfo country = cloneToBeInserted as CountryInfo;

    if (country != null)
    {
        country.CountryThreeLetterCode = "MCC";
        country.CountryTwoLetterCode = "MC";
    }
}
```

To clone an object, you need to:

1. Prepare an instance of the **CloneSettings** class (requires a reference to the **CMS.DataEngine** namespace).
2. Configure the cloning process by assigning values to the properties of the CloneSettings object.
3. Call the **InsertAsClone** method for the original object (converted to a generalized object type), with the prepared *CloneSettings* specified through the parameter.

CloneSettings properties

In addition to setting the code name and display name of the cloned object, the properties of the **CloneSettings** class determine what operations the system performs during the cloning process. In the example, the **IncludeChildren** flag is set to true, so the code also clones all state objects under the given country and assigns them to the new country.

The **BeforeCloneInsertCallback** property registers a custom handler method that the system executes just before the clone is inserted into the database. Such methods allow you to implement any functionality required to correctly clone the object. The object that is being cloned and the corresponding *CloneSettings* object are passed as parameters, so you can dynamically set the values based on the currently assigned clone settings.

Other possible callback options are:

- **AfterCloneInsertCallback** - executed after the cloned object itself is created and inserted, but before the system starts cloning any associated child objects or bindings.
- **AfterCloneStructureInsertCallback** - called once all objects included in the cloning process are created.