The continuous integration solution allows you to exclude specific types of objects if you do not wish to synchronize all supported types. You can also exclude individual objects of a type that is otherwise included.

All related configuration is defined within the **repository.config** file in the root of the *CMS\App_Data\CIRepository* folder. Use the following process if you need to make any changes:

1. Edit the **repository.config** file.
2. Adjust the configuration according to your object filtering requirements. See the following sections for details:
   - Excluding object types
   - Excluding individual objects
3. Save the *repository.config* file.
4. Run complete serialization for all objects to bring your *CIRepository* folder into the required state:
   a. Disable running of scheduled tasks (using the **Settings -> System -> Scheduled tasks enabled** setting).
   b. Open the **Continuous integration** application in the Kentico administration interface.
   c. Click **Serialize all objects**.
   d. Wait until the serialization process finishes and then re-enable scheduled tasks.

Excluding objects in the *repository.config* file affects all processes related to continuous integration:

- Serializing of all objects onto the file system
- Automatic tracking of create, update and delete operations for the given objects and transferring of the changes to the serialized data
- Restoring of objects from the file system into the database

Important: To maintain consistency, always use the same *repository.config* settings across your entire development environment. Whenever you make any changes, use your source control system to synchronize the *config* file along with the other content of the *CIRepository* folder. When a developer loads a new version of *repository.config* from the source control, the new settings start applying after restoring objects into the database or a manual restart of the application.

Warning: Objects excluded in *repository.config* may still be deleted when restoring data from the file system because of dependencies. For example, even if you have setting keys excluded from continuous integration, the system still deletes setting keys that belong within a given setting category if you "restore" the deletion of the category.

Excluding object types

By editing your application's **repository.config** file, you can configure the system to ignore certain object types within continuous integration:

1. Add **<ObjectType>** elements into one of the following sections of the config file:
   - **<IncludedObjectTypes>** – defines a *whitelist* of object types. If you specify one or more object types, continuous integration only processes objects of the given type. No restrictions apply if the whitelist is empty.
   - **<ExcludedObjectTypes>** – defines a *blacklist* of object types. Continuous integration processes all included object types except for the listed types.
2. Set the values of individual **ObjectType** elements to the names of object types that support continuous integration.

   Tip: To find the *ObjectType* values, see Object types supported by continuous integration or check the names of the corresponding folders in the *CIRepository* folder.

The system uses the following rules to automatically exclude child and binding objects:

- Child object types follow the configuration of their parent object type. For example if you exclude page types, then queries, transformations and alternative forms of page types are also excluded.
- Binding object types are excluded if all object types within the given relationship are excluded.

The **<IncludedObjectTypes>** whitelist only allows you to specify the main object types (i.e. not child types or bindings).

In the **<ExcludedObjectTypes>** blacklist, you can exclude both the main object types and specific binding or child object types.

> By default, the *<ExcludedObjectTypes>* list contains the *cms.settingskey* object type. Settings may contain sensitive data – only remove the exclusion if you agree to make setting values available within the file system used by your application and any connected source control systems. You can remove the exclusion for settings in general, and add a *<ExcludedCodeNames ObjectType="cms.settingskey">* element to exclude individual setting keys that you consider sensitive.

Examples

```
<?xml version="1.0" encoding="utf-8"?>
<RepositoryConfiguration xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="
http://www.w3.org/2001/XMLSchema-instance">
        <!-- Continuous integration includes all supported objects except for email
templates. -->
        <IncludedObjectTypes>
        </IncludedObjectTypes>
        <ExcludedObjectTypes>
                <ObjectType>cms.emailtemplate</ObjectType>
        </ExcludedObjectTypes>
</RepositoryConfiguration>
```

```
<?xml version="1.0" encoding="utf-8"?>
<RepositoryConfiguration xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="
http://www.w3.org/2001/XMLSchema-instance">
        <!-- Continuous integration only includes web parts and web part categories.
Web part layouts are excluded. -->
        <IncludedObjectTypes>
                <ObjectType>cms.webpart</ObjectType>
                <ObjectType>cms.webpartcategory</ObjectType>
        </IncludedObjectTypes>
        <ExcludedObjectTypes>
                <ObjectType>cms.webpartlayout</ObjectType>
        </ExcludedObjectTypes>
</RepositoryConfiguration>
```

Excluding individual objects

By editing your application's **repository.config** file, you can exclude individual objects from continuous integration. The system combines the filtering rules for individual objects with any existing rules for entire object types.

1. Add **<ExcludedCodeNames>** elements into the **<ObjectFilters>** section.
2. Specify the type of the objects that you wish to exclude for each *<ExcludedCodeNames>* element:
   - If you do not set the **ObjectType** attribute for the element, it applies to objects of ALL types.
   - To apply the filtering rule to a specific object type, set the element's **ObjectType** attribute to the given object type name.
     - You can use both main object types and child types (that are included in your continuous integration configuration).
     - Binding types typically do not have a code name field, so cannot be used.
     - Rules for specific object types override rules for all object types. You can disable rules for all object types by adding empty *<ExcludedCodeNames>* elements for individual object types.

    - Do not add multiple *<ExcludedCodeNames>* elements for the same object type. Instead, use multiple values separated by semicolons and/or wildcards.

3. Within the content of each *<ExcludedCodeNames>* element, list the **code names** of objects of the given type that you wish to exclude.
    - To cover multiple objects with a shared code name prefix or suffix, add the % wildcard to the **end or start** of the code name entry. Using only the % wildcard as the value excludes all objects of the given type.
    - You can add multiple code name entries for a single object type, separated by semicolons.
    - To exclude pages (the *cms.document* object type), use **alias path** values instead of code names. This identifies pages based on their location in the content tree.
    - The code name values are case insensitive.

When you exclude an object, the system automatically excludes dependent objects according to the following rules:

- **Child objects** – all child objects are excluded with their parent. For example, excluding a country object also excludes all states under the given country.
- **Binding objects** – binding objects are automatically excluded if at least one of the related objects is excluded. For example, excluding a user automatically excludes all binding records that involve the given user, such as user-role relationships, bindings that identify the user as a moderator of forums, etc.
- **Site objects** – excluding a site automatically excludes all site-specific objects belonging to the given site.
    - Does not apply to objects that are only connected to sites through bindings. For example, users can be assigned to sites through bindings, but are still global objects (not excluded along with sites).
- **Community group objects** – excluding a group automatically excludes all objects that belong to the group (group members, forums, media libraries, etc.).
    - Does not apply to pages assigned to groups through the *Owned by group* property.
- **Category hierarchies** – for objects that are organized within categories in a tree hierarchy in the administration interface, excluding a category also excludes all objects in the sub-tree. For example, excluding a web part category also excludes all web parts and sub-categories within the given category.
    - UI elements are an exception. Continuous integration does not automatically exclude the entire sub-tree under excluded elements. However, excluding a module also excludes all UI elements that belong to the given module.

Notes

- It is not possible to exclude objects without a code name field (except for pages).
- The object exclusion rules apply globally for all sites, even for pages. For example, adding the */Testing/%* alias path for pages excludes all pages under the */Testing* section on any site in the system.
- When excluding objects that have a defined and significant order (personalization variants, device profiles, etc.), you must always keep excluded objects as last in the order, after the remaining objects. Otherwise you may encounter order inconsistencies when transferring objects between different instances. You may need to manually move excluded objects to the end of the order after restoring new objects from the *CIRepository* folder.

Examples

```
<?xml version="1.0" encoding="utf-8"?>
<RepositoryConfiguration xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="
http://www.w3.org/2001/XMLSchema-instance">
        <!-- Continuous integration includes all supported object types -->
        <IncludedObjectTypes>
        </IncludedObjectTypes>
        <ExcludedObjectTypes>
        </ExcludedObjectTypes>
        <ObjectFilters>
                <!-- Excludes the "SMTP server password" and "Staging service
password" setting keys -->
                <ExcludedCodeNames ObjectType="cms.settingskey">CMSSMTPServerPassword;
CMSStagingServicePassword</ExcludedCodeNames>
```

```
                <!-- Excludes all web parts whose code name starts with the "test"
prefix -->
                <ExcludedCodeNames ObjectType="cms.webpart">test%</ExcludedCodeNames>
                <!-- Excludes pages under the "/Testing" section of the content tree --
>
                <ExcludedCodeNames ObjectType="cms.document">/Testing/%<
/ExcludedCodeNames>
        </ObjectFilters>
</RepositoryConfiguration>
```

```
<?xml version="1.0" encoding="utf-8"?>
<RepositoryConfiguration xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="
http://www.w3.org/2001/XMLSchema-instance">
        <!-- Continuous integration includes all supported object types -->
        <IncludedObjectTypes>
        </IncludedObjectTypes>
        <ExcludedObjectTypes>
        </ExcludedObjectTypes>
        <ObjectFilters>
                <!-- Excludes all objects of any type whose code name starts with the
"test" prefix, except for web parts -->
                <ExcludedCodeNames>test%</ExcludedCodeNames>
                <ExcludedCodeNames ObjectType="cms.webpart"></ExcludedCodeNames>
                <!-- Excludes pages under the "/test" section of the content tree -->
                <ExcludedCodeNames ObjectType="cms.document">/test/%<
/ExcludedCodeNames>
        </ObjectFilters>
</RepositoryConfiguration>
```