Providers in the Kentico API are classes that allow the system to manipulate objects and perform various actions. The following list presents the providers that the system uses and their purpose:

- **Info providers** contain methods for working with specific types of objects stored in the database, such as users, contacts, custom table records, etc.
- **Data provider** handles low-level database operations. The Info providers are built on top of the Data provider and use it to store and fetch data.
- **Email provider** manages the email queue and sends emails.
- **Search provider** ensures the functionality of the SQL search.
- **File system providers** allow you to access various file systems. They extend the CMS.IO namespace. See Working with physical files using the API to learn how to write a custom file system provider.

By developing custom providers and using them instead of the standard ones, you can modify the behavior of the application (or a specific feature) according to your exact requirements.

Adding your custom code files

When developing custom functionality, you can place your code files in:

- A **separate project** (assembly) integrated into the Kentico solution
- The **App_Code** folder (or **Old_App_Code** on web application projects)

Placing your customizations into the App_Code folder ensures that the code is compiled dynamically when required and automatically referenced in all other parts of the application.

Writing the custom code

Every class used to customize the application must **inherit from the original class**. This allows you to implement your modifications or additions by overriding the members of the given class. When creating overrides for existing methods, it is recommended to call the original base method within your custom code.

Registering custom providers

After you write the code, you must register your custom classes to ensure that the system uses them instead of the default providers. You can choose between two options of registering custom provider classes:

- Using assembly attributes - the most straightforward way to register providers.
- Using the web.config file - allows you to switch between different providers (custom or default) without having to edit the application code.

Other customization options

In addition to providers, you can also customize the following helper and manager classes:

Helper class name	Namespace	Description
AutomationHelper	CMS.SalesForce.Automation	Provides marketing automation support for common actions related to Salesforce integration.
CacheHelper	CMS.Helpers	Handles operations with cache items.
ClassHelper	CMS.Base	Takes care of dynamically loaded classes and assemblies.
CookieHelper	CMS.Helpers	Contains methods for managing cookies.
DirectoryHelper	CMS.IO	Manages directories in the file system.
LeadReplicationHelper	CMS.SalesForce	Replicates contacts into SalesForce leads.

LocalizationHelper	CMS.Localization	Retrieves localized text from resource strings.
MediaHelper	CMS.Base.Web.UI	Provides methods for rendering media content.
MFAuthenticationHelper	CMS.Membership	Contains methods and properties that implement and configure multi-factor authentication.
OutputHelper	CMS.OutputFilter	Manages the HTML output and the output cache.
TriggerHelper	CMS.Automation	Manages marketing automation triggers.

Manager class name	Namespace	Description
AutomationManager	CMS.Automation	Handles the marketing automation process.
SyncManager	CMS. Synchronization	Synchronizes page and object data to other instances of the application when using Content staging.
VersionManager	CMS. DocumentEngine	Provides page versioning functionality.
WorkflowManager	CMS. DocumentEngine	Handles the workflow process.

You can also use global event handlers to customize the behavior of the system. Handlers allow you to execute custom code whenever a specific event occurs in the system, such as page or object changes, various parts of the user authentication process, etc.

Examples

- Custom Info provider example demonstrates how to customize standard Info providers.
- Custom Email provider example shows a sample customization of the email provider.
- Custom Data provider example describes the specific steps that you need to take to customize the Data provider.
- Custom SQL search provider example demonstrates how you can modify the SQL search provider.
- Customizing e-commerce data for Google Analytics demonstrates how to customize helper classes that the system uses to create product and order JSON data for the purposes of Google Analytics.