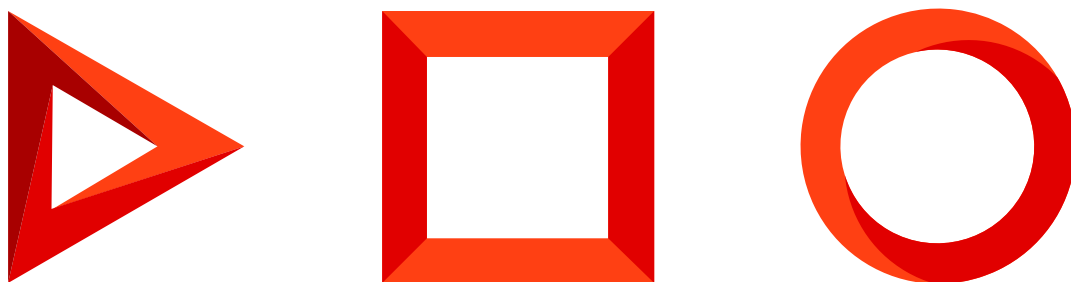


Marketplace app compatibility

Recommendations for developing compatible Marketplace apps

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

Recommendations for developing compatible Marketplace apps	4
Recommendations for developing a reversibly installable Marketplace app	4
Recommendations for developing a Marketplace app compatible with Creatio in multi-tenant mode	5
Recommendations for developing a Marketplace app compatible with Creatio that has external file storage connected	5

Recommendations for developing compatible Marketplace apps



Upcoming Creatio releases change the way apps are installed into the environment and app compatibility requirements. Configure the Marketplace app to ensure the app is compatible with Creatio.

Follow these **recommendations** to ensure the Marketplace app compatibility:

- Recommendations for developing a reversibly installable Marketplace app
- Recommendations for developing a Marketplace app compatible with Creatio in multi-tenant mode
- Recommendations for developing a Marketplace app compatible with Creatio that has external file storage connected

Recommendations for developing a reversibly installable Marketplace app

In version 8.0 Atlas, we changed the rollback mechanism that the users can initiate if the package installation ends with an error. The **key changes** to the rollback mechanism are as follows:

- You can roll back changes executed by data bindings.
- You can flag custom SQL scripts as backward compatible. Learn more in a separate article: [Backward compatible SQL scripts](#).
- You can use install scripts to execute CRUD operations with data as part of package installation. Learn more in a separate article: [Customize delivery process](#).

These changes ensure the restoring the configuration without using database backup as long as the functionality of the installed package executes reversible database changes. To **develop a reversibly installable Marketplace app**, follow our recommendations and requirements when implementing the functionality.

To make the recovery process easier, the upcoming releases will include an **automatic app analyzer** that detects irreversible changes. Before and while installing the package, the analyzer will check the package functionality for irreversible changes and notify the user on detected irreversible changes. The WorkspaceConsole utility already includes partial functionality of the analyzer. Upcoming Creatio releases will add the analyzer to the mechanism that installs packages via UI.

To ensure the functionality of the Marketplace app is installable reversibly, follow these **recommendations**:

- Use **package install scripts** instead of SQL scripts to execute CRUD operations with data as part of the package installation process.

If the installation of a package that contains Marketplace app functionality ends with an error, you can roll the configuration back completely as long as configuration elements make reversible changes to the database as part of the package installation. However, you cannot roll the configuration back completely if the configuration elements of the [*SQL script*] type make irreversible changes to the database as part of the package installation.

Install scripts execute operations with data using the `Entity` class. Creatio saves changes made by install scripts as part of the package installation to the package backup. You can roll back to the previous configuration state completely in case of an error. Learn more in a separate article: [Customize delivery process](#).

- Use **backward compatible SQL scripts** to create database objects. Learn more in a separate article: [Backward compatible SQL scripts](#).

Recommendations for developing a Marketplace app compatible with Creatio in multi-tenant mode

Multi-tenant mode is a Creatio deployment mode that enables multiple organizations to work independently on a single Creatio instance. Multi-tenant mode in Creatio grants the organization users access to the organization sections, apps, and functionality. Users, data, and functionality of different organizations do not overlap. As such, Creatio in multi-tenant mode requires organization data to be isolated.

To ensure the Marketplace app is compatible with Creatio in multi-tenant mode, use **package install scripts** instead of SQL scripts to execute CRUD operations with data as part of package installation. Creatio forbids executing SQL scripts on Creatio instances in multi-tenant mode to isolate organizations and ensure data security. Learn more about install scripts in a separate article: [Customize delivery process](#).

Recommendations for developing a Marketplace app compatible with Creatio that has external file storage connected

Users can set up integration with S3 (since version 7.18.3) and Azure Blob (since version 8.0.2) external file storages. Learn more in the user documentation: [File storage](#).

Users of Creatio in the cloud can store files in the S3 file storage on demand. We will configure integration with the Azure Blob file storage for users of Creatio in the cloud in the upcoming Creatio releases. As such, Marketplace apps that use direct SQL queries to access files will be incompatible with Creatio that has S3 or Azure Blob file storages connected.

To ensure the Marketplace app is compatible with Creatio that has external file storage connected, use file management API in your Marketplace app instead of direct SQL queries. The API lets you access files stored in the database and external storage. Learn more about the API in a separate article: [API for file management](#).