

Models (Platform)

Models are predefined applications, consisting of files/reports, OLAP databases, Integrator jobs, scheduled tasks, and settings. Not every model must contain all objects mentioned above. Models are created by a model designer, then shipped in a so-called package to the end-user, who can configure and customize them. The shipping can happen via the new Marketplace (a panel in Jedox Web, which shows content from a centralized server), or a model can be installed directly from a file. The user can see and change installed models in the new “My Models” panel in Jedox Web.

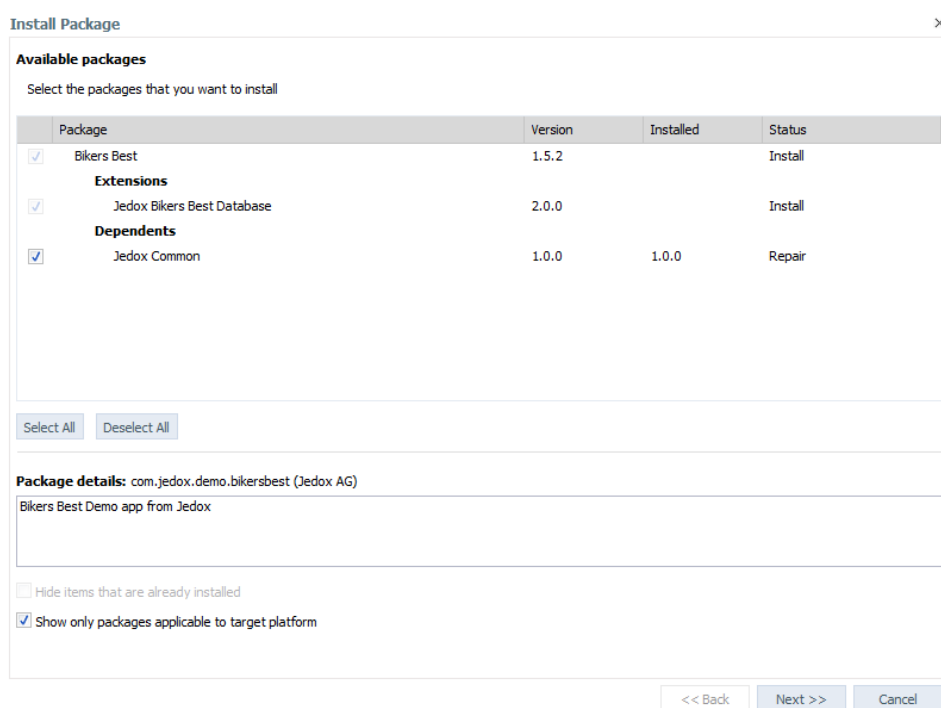
1. Marketplace and Model Installation

The “Marketplace” panel shows all [published models](#). The content here comes from a centralized server which is accessed over the Internet. The server has to be accessible (that is, connection to the Internet has to be possible) for both the server where Jedox is installed, and the client machine where a web browser is accessing Jedox. If the latter machine has no Internet access (for example if it is restricted to a LAN network), the Marketplace will fail to load; if the former (Jedox server) has no access to the Internet, the Marketplace may be shown on the client’s web browser, but the actual installation of models will not be possible.

Furthermore, the Jedox server needs a valid [certificate for the communication with the Marketplace](#).

When clicking on a specific model in Marketplace, the user will be shown Details for this model, such as a description, screenshots, and a changelog of previous model versions.

To install a specific model, click the “Install” button of that model in Marketplace. This will prompt an Installation Wizard.



Install wizard

Models can be installed multiple times in parallel, using so-called namespaces. Namespaces behave similar to “installation directories”. When installing a model several times in various namespaces, most contents of the model will be unique and can be modified independently; you can later also install updates for the model separately. The exception is the database created during the installation (if there is a database being created. By default, the model

will always use the same name for the database it creates. However, if the installation allows to set a custom database name, you can choose a different name in each separate model installation / namespace. By default, the installation dialog suggests to install the model in the “global” namespace.

The install dialog then shows a detailed overview of a model’s contents, such as model extensions. Model extensions usually contain additional content, like database sample data, which is not essential for usage of the model. If the model designer made installation of extensions optional, the user can uncheck the extensions he doesn’t wish to install here.

If the model installation requires the user to accept Terms & Conditions for usage of the model, the next page of the installation wizard will show the Terms and Conditions (opt-in). After accepting the Terms & Conditions and clicking “Next”, the wizard will show a page detailing the following installation steps. Another click on “Next” starts the installation.

If the model includes database contents, the user will usually be prompted for entering a name for the database that will be created, or modified by the model.

For offline installation from file, the Installation Wizard process is identical.

When the model installation is executed, Jedox Web generates the required content locations e.g. in “Reports” and “Reports Designer”; model contents there are always located within the “models” group. If the model includes database contents, the installer will execute one or

several Jedox Database Scripts to set these contents up.

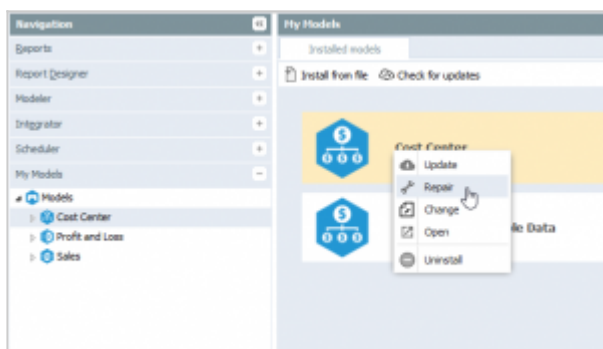
2. Panel “My Models”

The “My Models” panel provides an overview for all models that were created, or installed, on the local Jedox Web server. Starting with Jedox 7.0, the “My Models” panel will most likely be empty. When models are installed, they are listed in a hierarchical structure on the left side, with a detailed list in the main Window. The user can navigate to the reports of a specific model from “My Models” by clicking the arrow icon.

Reports are however of course also accessible by navigating directly to the Reports section of Jedox Web.

The list in “My Models” will also show if an [Update](#) is available in the Marketplace for one or more of the installed models.

For all installed models, the user has the option to “Repair”, “Change”, or “Uninstall” the model by right clicking the model and choosing the action from the context menu.



“Repair” of a model executes the same procedure as during the initial installation again. This can be used if for example model contents have been accidentally changed or deleted. The model contents will be recreated in the same state as after the installation. With “Change”, the user can optionally install or uninstall optional parts of the model. For example, if a model extension (such as sample data) was not included by the user in the initial installation, he can later add the extension via the “Change” option.

“Uninstall” will remove a local installation of a model. This will remove all model contents, such as reports, Integrator projects, settings etc. The exception is database content. Whether or not database content will be deleted is up to the model author. In the case of the Jedox models provided via Marketplace, database content is not deleted when uninstalling a model.

3. User Rights for Models

Access to the new “My Models” and “Marketplace” panels is handled like other panels, via OLAP right objects per role, and can be defined in the section “Roles” in the Administration panel. Note that this only refers to the panels and action in those panels, but not access to model contents themselves, such as reports, databases, or Integrator projects. For “My Models”, the relevant right object is “ste_packages”. Users (resp. roles) with no access (“N”) are not able to see the “My Models” panel. User with Read” access (“R”) are able to see the panel and the list of installed models. They are able to check for updates, but they are not able to install, uninstall or modify models. Users with “Write” access (“W”) are additionally able to install, update and

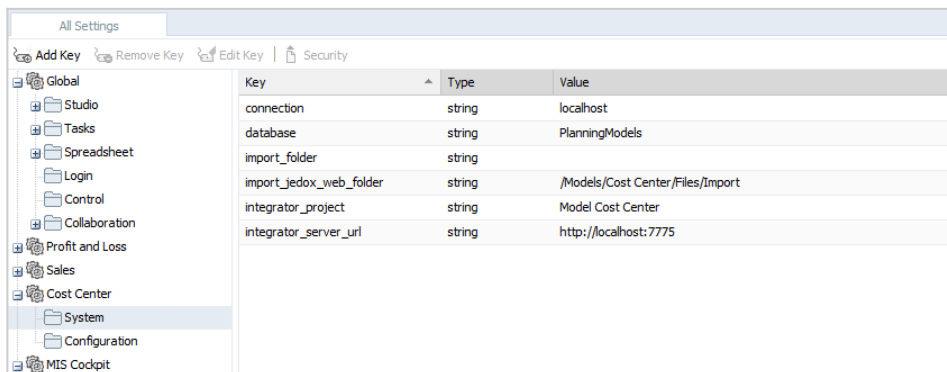
uninstall models.

The Marketplace panel is not accessible to users whose User Role has no access ("N") on the new "ste_repository" right object. Users with "Read" access ("R") are able to browse the Marketplace panel but they cannot install any of the available models. Users with "Write" or "Full" access are able to install models from the Marketplace.

Note that if a model executes database scripts during installation, the user running the installation also must have all OLAP rights required for the commands in the scripts. This usually means that rights for creating database, dimensions, cubes, elements, rules etc. will be required.

4. Model Settings

In many cases, models will require some specific settings, such as a database name or connection name. In Settings Manager, when creating a new settings key, the user can now select whether the key is "Global", or scoped within some (existing, i.e. created or installed) model. Model-scoped settings are displayed in separate nodes in the tree in Settings Manager.



Key	Type	Value
connection	string	localhost
database	string	PlanningModels
import_folder	string	
import_jedox_web_folder	string	/Models/Cost Center/Files/Import
integrator_project	string	Model Cost Center
integrator_server_url	string	http://localhost:7775

In spreadsheets, model-scoped settings can be retrieved with the already existing CONFIG.GET() function. It has been extended with an optional second parameter. In this parameter, the user can specify a unique name of a model; the function will then try to retrieve the setting key specified in first parameter within the scope of the specified model. To retrieve the unique name of the model to which the current spreadsheet belongs, the new function MODEL.ID() can be used (the alias PKG.ID() also exists). For example, assume that your model includes a setting "logo_path" (containing an URL to some image). You could retrieve this setting value with:

```
=CONFIG.GET("logo_path",MODEL.ID())
```