

Using Teamwork Cloud

Introduction

Teamwork Cloud (TWCloud) is our next-generation repository for collaborative development and versioned storage of models. TWCloud is intended and developed to improve the areas of model governance, model analysis, and integration with third-party tools. The benefits to enterprise users include:

- Role-Based Access Control is a major feature that enables efficient management of users' access rights across the enterprise. Additionally, as distributed systems can be challenging to administer, we have built a convenient [web-based administrative interface to TWCloud](#), which provides a single way to manage user accounts, licensing, access control, LDAP integration, and SSL setup.
- TWCloud is designed from the ground up to support working with large models. In particular, the efficient delta-based protocol used when users commit and retrieve updates to a model will save time for those users who access the repository through poor network connections. Likewise, the amount of data that is transmitted depends only on the size of the changes being made to the model.
- A database-driven model storage paradigm in TWCloud unveils powerful client-side features for model changes and history tracking at the element level.
- **Project Usages location:** rather than mounting everything under the main model root element, the Model Browser has a dedicated root for Project Usages in Teamwork Cloud.
- **Shared Packages:** there are no shared packages in the TWCloud Project Usages; all content is available for reference. However, the TWCloud project can be stripped using the [Hiding packages](#) feature.
- The mounting of Used Projects in TWCloud is achieved via a model-based Mount relationship. In order to ensure the same Containment tree structure when adding/migrating projects to TWCloud, original mounting is re-created by adding the Mount relationship automatically.
- Teamwork Cloud architecture supports the sticky versions (used project versions that remain used until they are changed in the main project) approach of used projects. However, you can easily update the used project to the latest version by setting up a notification that will inform you about the appearance of a new used project version. Alternatively, you can use the [Teamwork Cloud Used Project Updater plugin](#) that allows the automatic update of used projects.
- In Teamwork Cloud, every new version of the project gets a unique consecutive number, even between different branches, which makes it easier to keep track of the commit history and identify the specific project versions.
- Teamwork Cloud used [projects are read-only](#). This helps to prevent cyclic dependencies. We recommend that you open a used project as a stand-alone and edit it there; however, you can move elements to other projects using the [cross-project refactoring functionality](#) in Teamwork Cloud as well. This feature allows transferring data between projects in just a few steps.

You can find more information about TWCloud's key benefits, required resources, and the latest news in its [official product page](#).

Related pages

- [Adding projects to Teamwork Cloud](#)
- [Managing server projects](#)
- [Opening projects from Teamwork Cloud](#)
- [Collaboration between disconnected teams](#)
- [Offline modeling](#)
- [Package permissions](#)
- [Locking model for edit](#)
- [Using Lock-Free Editing mode](#)
- [Unlocking model](#)
- [Committing changes to Teamwork Cloud](#)
- [Updating changes from Teamwork Cloud](#)
- [Reviewing model history](#)
- [Managing project usages](#)
- [Moving elements to other projects](#)
- [Hiding packages](#)
- [Global element usage search](#)
- [Managing Teamwork Cloud users](#)
- [Representation of Teamwork Cloud users](#)
- [Used Project Switch Impact Analysis](#)
- [Used Projects Auto Update Plugin](#)
- [Model merge in Teamwork Cloud](#)
- [Configuring Resource Usage Map URL](#)
- [Troubleshooting](#)