

MDK User's Guide

Welcome to the [MDK User's Guide](#) for Model Development Kit Version 2.4.0. Model Development Kit (MDK) is a MagicDraw plugin that syncs models to [View Editor](#) and other tools like Timeline Management System (TMS) using the [MMS](#) server. More information can be found here: [About MDK](#). This guide is designed to introduce users to the various features of MDK using a combination of tutorials, written explanations, and screencasts. Throughout this guide, various aspects of View Editor and MMS will be used.

*Check **your** specific project modeling practices with regard to tasks, sites, content control, etc.*

Table of Contents

1. [About MDK](#)
2. [Basic Set Up](#)
3. [Use MDK Project](#)
4. [Use MDK](#)
 1. [MDK Specific Features](#)
 2. [MMS Syncing](#)
 3. [Selected Modeling Tutorials](#)
5. [Use Teamwork](#)

Additional Resources

- Current deployment, resources, community, etc: [Model Development Kit \(MDK\)](#)
- Latest plugin release: [Release Companion](#)
- MagicDraw client: [MagicDraw](#) and [CAE MagicDraw FAQ](#)
- Latest MMS release: [Release Companion](#)
- View Editor: [View Editor](#) and [Release Companion](#)

For any further inquiries, please contact Ivan Gomes at Ivan.Gomes@jpl.nasa.gov or Marie Piette at Marie.Y.Piette@jpl.nasa.gov.

Table of Contents

1 About MDK	6
2 Basic Set Up	7
2.1 Download and Open MagicDraw	7
2.2 Start MagicDraw	8
2.3 Load MDK Plugin (Optional)	8
2.4 Use MDK Project	9
2.4.1 Create a Teamwork Project	9
2.4.2 Open a Teamwork Project	9
2.4.3 Create an MDK-Enabled Project	9
3 Use MDK	11
3.1 MMS Menu	11
3.2 MMS Syncing	12
3.2.1 Coordinated Syncing	12
3.2.2 Update and Commit	12
3.2.2.1 Update from MMS	13
3.2.2.2 Commit to MMS	14
3.2.2.3 Commit with Deletes to MMS	15
3.2.2.4 Generate All Documents and Commit to MMS	15
3.2.3 Manual Syncing	16
§ Manual Order of Operations	16
3.2.4 Resolving Conflicts and Errors	16
3.2.5 Dynamic Syncing	19
3.2.5.1 Dynamic Sync Elements	19
3.2.5.2 Dynamic Sync Views	20
3.3 MDK Specific Features	20
3.3.1 Documents and Views	20
3.3.2 Viewpoints and Viewpoint Methods	20
3.3.3 _MMSSync_ Package	21
3.3.4 holding_bin	21
3.4 Selected Modeling Tutorials	22
3.4.1 Create and Generate Documents	22
3.4.1.1 Create A Document With A View	22
3.4.1.1.1 Generate Views and Sync with MMS	23
§ Generate Views and Commit to MMS	23
§ Manually Generate Views and Validate with MMS	23
3.4.1.2 Insert Diagram as Image	24
3.4.1.3 Create and Generate a Rapid Table	24
3.4.2 Site Characterization	25
3.4.3 Create Enumerated Values	28
3.4.4 Create Toggable Boolean Values	29
3.4.5 Systems Reasoner	29
3.4.6 Create a Reusable Cover Page	30
4 Use Teamwork	31
4.1 Everyday Interaction	31
§ Teamwork and MMS	32
§ Handling Teamwork Crashes	33
4.2 Collaborate Menu Overview	33
4.3 Create Teamwork Branch	35
4.4 Open Teamwork Branch	35
4.5 Syncing with VE Task	35

List of Figures

List of Equations

List of Tables

1 About MDK

Model Development Kit (MDK) is a MagicDraw plugin that syncs models to [View Editor](#) and other tools like Timeline Management System (TMS) using the [MMS](#) server. This provides users the ability to work with real and useful data without having to open the actual [MagicDraw](#) model itself. This whole setup between applications and servers provides a reliable history of a system model as well as an environment where all types of users can contribute.

2 Basic Set Up

The following subviews are designed to guide a user through downloading, installing, and configuring MagicDraw (with MDK) locally so that the user can immediately begin working on a project.

2.1 Download and Open MagicDraw

In the following clips, the user demonstrates how to download and install MagicDraw (full installation) for daily use. A full installation means that the MDK has already been bundled within the MagicDraw installation/download. If the user is using the full installation as suggested, [Load MDK Plugin \(Optional\)](#) becomes optional.

Download the latest full installation build from the [MagicDraw](#) landing page.

If you have any trouble with installation or would like to install on a different operating system than shown below, please refer to the [MagicDraw Installation Instructions](#) in the [CAE MagicDraw FAQ](#) document or [Start MagicDraw](#).

1. Download MagicDraw:

- 1.
2. Link to video: https://jpltube.jpl.nasa.gov/NetworkFileStore/8706/media_720.mp4
3. Video Instructions:
4. Navigate to MagicDraw landing page.
 1. Click the URL link to download latest
 2. Login to Artifactory using your JPL username and password
 3. Click the zip file to download the installation file with MDK plugin
 4. Once downloaded, move the zipped file into whichever folder you wish. In this case, Applications folder.
 5. Unzip the file in new location

2. Run MagicDraw

- 1.
2. Link to video: https://jpltube.jpl.nasa.gov/NetworkFileStore/8687/media_720.mp4
3. Video Instructions:
 1. Find MagicDraw application and Double Click to open
 2. Click Open when it asks "Are you sure you want to open?"
 3. (If you see something along the lines of "MagicDraw cannot open because it is damaged", please refer to the MagicDraw FAQ)
 4. You are now ready to configure MagicDraw

3. Configure and Connect

- 1.
2. Link to video: https://jpltube.jpl.nasa.gov/NetworkFileStore/8707/media_720.mp4
3. Video Instructions:
 1. Click "OK" to connect to the FLEXnext License Server (cae-lic01.jpl.nasa.gov)
 2. Click "OK" to accept default Editon/Plugin options
 3. Choose the Full Featured - SSCAE perspective (default)
 4. (You may need to register for MagicDraw...fill in all the fields and Click register.)
 5. You are now ready to work with MagicDraw
 6. Notice the Quick Links provided in Manage Projects, No Magic News, and other tools

The following registration page may appear. The user should fill in all the fields and register.

Registration

Please identify yourself and register installation

Are these your details? If not, please, change:

First Name:

Last Name:

E-mail:

Please make sure your email address is correct.

Phone:

Role:

Industry:

Company:

Country:

Registration will enable us to provide you with professional services:

- Customer support for all problems, questions, and suggestions.
- Access to early releases and evaluations.
- New versions, patches, and updates.

Please share your experience about the tool (optional):

The user should now be fully ready to use MagicDraw with MDK.

For more specific user enabled settings, please refer to the [Configuration](#) in the FAQ.

2.2 Start MagicDraw

The assumption is that the user has already downloaded the full installation of MagicDraw and configured it appropriately upon installation. More information and instructions on how to do so can be found here: [Download and Open MagicDraw](#).

In this clip, the user demonstrates what a MagicDraw user will encounter upon every start.

<src="https://jpltube.jpl.nasa.gov/NetworkFileStore/7622/media_720.mp4">Your browser does not support embedded videos.

Link to video: https://jpltube.jpl.nasa.gov/NetworkFileStore/7622/media_720.mp4

Video Instructions:

1. Open Application from downloaded unzipped folder
2. Click "OK" to connect to the FLEXnext License Server (cae-lic01.jpl.nasa.gov)
3. Click "OK" to accept default Editon/Plugin options
4. Notice the quick links on the welcome page

2.3 Load MDK Plugin (Optional)

In this clip, the user demonstrates how to import the MDK plugin into an already existing MagicDraw installation. CAE highly encourages MDK users to download MagicDraw with the pre-bundled MDK but is showing this option for the convenience of the user. If the user has the pre-bundled application, this step is not necessary and should be skipped.

Download the latest MDK plugin from the [Model Development Kit \(MDK\)](#) landing page.

<src="https://jpltube.jpl.nasa.gov/NetworkFileStore/7628/media_720.mp4">Your browser does not support embedded videos.

Link to video: https://jpltube.jpl.nasa.gov/NetworkFileStore/7628/media_720.mp4

Video Instructions:

1. Begin with MagicDraw opened and MDK Plugin downloaded
2. Navigate to the "Help" drop down list and select "Resource/Plugin Manager"
3. Click "Import"

4. Find and Select the ZIPPED plugin in the import finder, which user should have already downloaded. If plugin has not been downloaded, proceed to download it before continuing
5. Click "Open" to import plugin
 1. If a previous version of the MDK plugin was installed, a pop up message may appear asking to reinstall the resource - Click "Yes" if this occurs
6. A message will pop up indicating that the MDK plugin will be updated upon startup - Click "OK"
7. Close the Resource/Plugin Manager
8. Restart MagicDraw in order to model with the new plugin

2.4 Use MDK Project

The following subviews are designed to guide a user through creating/accessing a Teamwork project or creating an MDK-enabled project. An MDK-enabled project means that the entire project model will be able to sync to an MMS server and therefore sync with View Editor as well.

2.4.1 Create a Teamwork Project

To create a Teamwork project, a user must have an administrator account, given by the CAE Teamwork Admin. Please contact Ben Inada at Benjamin.Y.Inada@jpl.nasa.gov or Marie Piette at Marie.Y.Piette@jpl.nasa.gov for help with creating a teamwork project or any other admin privilege.

If a user is already an admin, refer to the official NoMagic documentation, [Adding Projects to Teamwork Server](#), for more help.

2.4.2 Open a Teamwork Project

In this screencast, the user demonstrates opening an existing, MDK-enabled project hosted on a Teamwork server. A MDK-enabled project means that the entire project model will be able to sync to an MMS server and therefore sync with the View Editor as well.

Other Resources:

- CAE basic teamwork information, such as server URL and version, can be found on the [Model Development Kit \(MDK\)](#) landing page.
- How to [Use Teamwork](#)

<src="https://jpltube.jpl.nasa.gov/NetworkFileStore/8685/media_720.mp4">Your browser does not support embedded videos.

Link to video: https://jpltube.jpl.nasa.gov/NetworkFileStore/8685/media_720.mp4

Video Instructions:

1. *Begin with MagicDraw opened
2. Navigate to the "Collaborate" drop down menu, click Login
3. Enter your jpl username and password to connect to the server
4. Make sure that "Use Secured Connection (SSL)" has been checked
5. *You may use the "Auto Login to Server" option if you wish
6. *Upon first connecting to a server, you will have to manually type in the server name
 1. *After connecting successfully the first time, MagicDraw will remember the last server you signed into and automatically select that server
 2. *A list of previous servers that have been successfully connected will appear from the dropdown menu next to "Server name"
7. Select desired server
8. Click "OK"
9. *Upon login, notice the other options that are now available
10. Select "Open Server Project"
 1. *The Open Server Project Model will display all the available projects on the current server.
11. *To view the current Teamwork server branches for a selected project, click the "..." button next to said project
12. *This will display all the current branches for that project.
13. *The "master" branch is named "Trunk" for teamwork
14. Select desired project and Click Open
15. *For the intents of this video, the open project process has been sped up
16. Upon opening, the project will automatically start Validating
17. You are now ready to begin on a Teamwork project

*Informational statement

2.4.3 Create an MDK-Enabled Project

In the following clips, the user sets up a brand new MagicDraw project to be MDK-enabled. A MDK-enabled project means that the entire project model will be able to sync to an MMS server and therefore sync with the View Editor as well.

1. Create New Project

- 1.
2. Link to video: https://jpltube.jpl.nasa.gov/NetworkFileStore/8675/media_720.mp4
3. Video Instructions:
 1. File>> New Project
 2. Select "SysML Project"
 3. Name the Project
 4. Choose project save location
 5. OK
 6. See message about System Engineering perspective - choose either (more information in description)

2. Import MDK Module to project

- 1.
2. Link to video: https://jpltube.jpl.nasa.gov/NetworkFileStore/8679/media_720.mp4
3. Video Instructions:
 1. Go to Options>>Modules
 2. Use Module
 3. Find the MDK folder (under MD's profiles)
 4. Select "SysML Extensions.mdxml"
 5. Finish
 6. See message about showing Auxiliary Resources - choose either (more information in description)

3. Add Model Management System stereotype

- 1.
2. Link to video: https://jpltube.jpl.nasa.gov/NetworkFileStore/8674/media_720.mp4
3. Video Instructions:
 1. Select "Data", Right Click, Select Stereotype
 2. If greyed out, means that Data needs to be locked for teamwork before editing
 3. Search for "MMS"
 4. Select "Model Management System" stereotype
 5. Apply

4. Assign Project to MMS server

- 1.
2. Link to video: https://jpltube.jpl.nasa.gov/NetworkFileStore/8676/media_720.mp4
3. Video instructions:
 1. Open "Data" specification (double click or use right click menu)
 2. Scroll down to MMS section
 3. Enter MMS URL aka server URL
 4. Enter desired MMS site (see description for how to create new site)
 5. Close

5. Initialize Project

- 1.
2. Link to video: https://jpltube.jpl.nasa.gov/NetworkFileStore/8677/media_720.mp4
3. Video Instructions:
 1. Login to MMS
 2. Right click "Data">>MMS>> Validate Models
 3. Wait for validation window to see initialization error
 4. Right Click>>Initialize Project
 5. Choose run on server (to get confirmation of successful project)

6. Sync new project

1. MDK 2.4+ Directions

1. Save your project. All elements will be exported via CSync at that time.

2. MDK 2.3 Directions

- 1.
2. Link to video: https://jpltube.jpl.nasa.gov/NetworkFileStore/8673/media_720.mp4
3. More information on syncing can be found here: [Commit to MMS](#)
4. More information about the new package can be found here: [_MMSSync_ Package](#)
5. Video Instructions:
 1. Run "Commit to MMS"
 2. Notice new "_MMSSync_" package

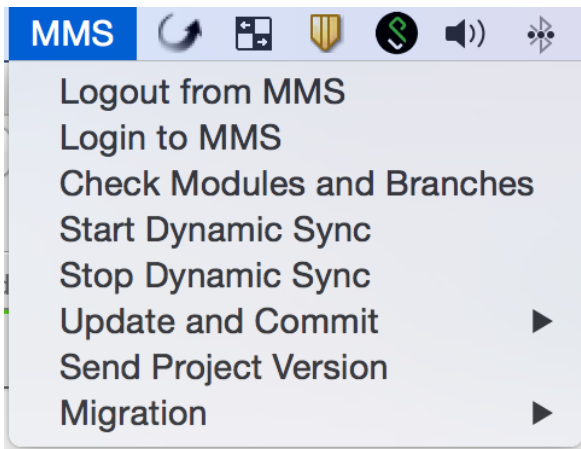
3 Use MDK

The following subviews are designed to guide the user through using all aspects of the MDK.

Table of Contents (Summarized Highlights)

1. [MDK Specific Features](#) - Explanation of some specific MDK features, such as Viewpoint Methods and View Instances, as well as what they mean to the daily user
2. [MMS Menu](#) - Brief overview about what can be found in the MMS menu from the MagicDraw toolbar menu
3. [MMS Syncing](#) - Information provided by CAE about the different aspects of syncing an MDK-enabled model with the MMS, including "one-click" syncing and manual syncing
 1. [Coordinated Syncing](#) - The new option provided in MDK 2.4.0+ to make syncing with the MMS occur automatically on model save / commit
 2. [Update and Commit](#) - The different options that are provided in MDK to make syncing with the MMS less tedious for the user
 3. [Manual Syncing](#) - Information about manual syncing to the MMS/View Editor, including order of operations, canceling requests, and screencasts
 4. [Resolving Conflicts and Errors](#) - A compilation of common conflicts/errors that appear during the syncing process and instructions on how to resolve them
4. [Selected Modeling Tutorials](#) - Collection of instructions on useful features provided by MDK to enhance the interaction between View Editor content and the MD model, including creating/syncing new documents both Manually and through "one-click" syncing

3.1 MMS Menu



The following provides an overview of the different operations that are offered through the top MMS menu. For information about the MMS menu that appears when selecting an element, such as a document, refer to [Manually Generate Views and Validate with MMS](#).

Implications:

MDK 2.4 and MDK 2.3:

- Logout from MMS
- Login from MMS - login to the MMS server, using a jpl username and password, that is associated with the project, which is the same as the server the project is synced to. To check, navigate to the "Data" folder at the top in the containment tree, go to its specification, and see MMS URL.
 - If you see this error, try logging in again
[2016.01.26::13:05:11] You are not authorized or don't have permission, (you can login and try again).
 - [2016.01.26::13:05:42] [ERROR] Getting changes from mms failed: Could not connect to broker URL: tcp://cae-ems.jpl.nasa.gov:61616. Reason: java.net.SocketTimeoutException: connect timed out
- Migration - The options to migrate projects from Project Migrations from 2.1 to 2.2 and 2.2 to 2.3 are provided to help the user sync their outdated model to the updated server

MDK 2.4:

- Validate - Contains options for validating Modules and Branches of the project.
- Generate All Documents - Generates views and view instances for all documents in the project.

MDK 2.3:

- Check Modules and Branches
- Start Dynamic Sync - refer to [Dynamic Syncing](#) ; Dynamic Sync is used for real time syncing, withough manual validations/committing to server
- Stop Dynamic Sync - refer to [Dynamic Syncing](#)
- Send Project Version - refer to [Dynamic Syncing](#)
- Update and Commit - refer to [Update and Commit](#)

3.2 MMS Syncing

The key to having MagicDraw/MDK interact with View Editor is through syncing the project model with the MMS server and in turn, View Editor. The following subviews elaborate on the different operations that MMS provides in order to properly sync the model with View Editor.

3.2.1 Coordinated Syncing

This documentation only applies to MDK 2.4.

Coordinated Syncing ties the MMS synchronization functionality to MagicDraw's model save or Teamwork model commit actions. When a save or commit is selected in MagicDraw, a similar commit also happens to the MMS. This minimizes the user interactions required to keep the model up to date on the MMS and ensures that parity is maintained between the local/Teamwork model and the MMS model. This also vastly reduces the need for alternative sync methods such as [Manual Syncing](#) .

Description:

While you are editing the model, element changes are collected in memory to be processed on your next save / commit. MMS changes by other users, in MagicDraw of View Editor, are also stored in memory but not processed. When a coordinated sync occurs, these two change lists are processed. MMS element changes are updated into the MagicDraw model as possible. Coordinated sync will not lock any elements, so in Teamwork projects it can only add new elements or update elements that are already locked by the user. After this update, MagicDraw model changes are committed to the MMS in the background. Finally, the save / commit occurs.

Any conflicts caused by changes to elements in both MagicDraw and MMS will be presented to the user in the validation window. Similarly, any elements with MMS changes that could not be updated (generally because they were not locked) will be presented to the user in the validation window. These conflicts will be presented to the user on every save/commit until resolved. The presence of conflicts does not prevent completion of the save/commit operation.

Usage:

MDK 2.4

- On Model Open
 - Coordinated Sync will prompt the user to log in to MMS. Logging in allows it to listen to messages track changes in real time, and be ready to gather and apply them at the appropriate time.
 - Cancelling this login, or submitting invalid credentials, will not prevent you from working on the model. It will only prevent you from receiving messages about elements changed in the MMS and prevent you from committing to the MMS until after you log in manually. Changes made to the model will be persisted locally for later commit.
- On Model Save / Teamwork Commit
 - Coordinated Sync will automatically perform model synchronization with MMS, committing MagicDraw elements to MMS or deleting elements no longer present in MagicDraw.
 - Coordinated Sync will update elements in MagicDraw based on MMS changes if the element can be edited and there is no version conflict between the MagicDraw and MMS versions of the elements.
 - Elements that can not be edited for the update will be logged, and another attempt to update the elements will be made during the next Coordinated Sync.
 - Coordinated Sync will present validation messages to the current user for any element with a MagicDraw / MMS conflict. so long as the element could be edited in MagicDraw.
 - Elements with a MagicDraw / MMS conflict that can not be edited at the time of the Coordinated Sync will be logged and presented to the current user for resolution after the next coordinated sync in which the element can be edited.

MDK 2.3

- Coordinated Sync is not available in MDK 2.3. It is recommended that you use [Update and Commit](#) instead.

3.2.2 Update and Commit

Update and Commit has been removed from MDK 2.4, and is no longer available for use. It has been replaced with [Coordinated Syncing](#) .

This documentation only applies to MDK 2.3.

The following subviews focus on the MMS capabilities that enable a user to interact with View Editor. All of the following operations are officially promoted by CAE as opposed to [Manual Syncing](#).

Motivation:

The motivation behind MMS Update and Commit capabilities was to provide users not only a more user friendly way of syncing a model, but also allow the MDK to operate the strenuous validations. With this "one-click" approach, a user will no longer have to constantly validate packages and judge every validated element to see if it should be updated from MMS. A simple decision to "Accept" instead of "Commit" or vice versa could cause damages to the model as well as overwrite once wanted data. This method was created to ensure protection against human error and allow the user to focus on the actual System Engineering instead of how to sync the model.

The "one-click" approach is so named because upon clicking one of the options below, the MDK performs many automated tasks and validations to provide the user with a comprehensive list of changed elements. If there are conflicts, errors, hierarchy changes, this "one-click" may turn into a couple or more; however, compared to the manual sync that has often been used, this is a streamlined solution. These conflicts, errors, etc. are handled by the [_MMSync_Package](#) and Teamwork.

IMPORTANT: It is **highly** suggested, though not absolutely required, that a MagicDraw project be fully synced before using these options for the first time. "Fully synced" means that the model has undergone a full [Manual Syncing](#).

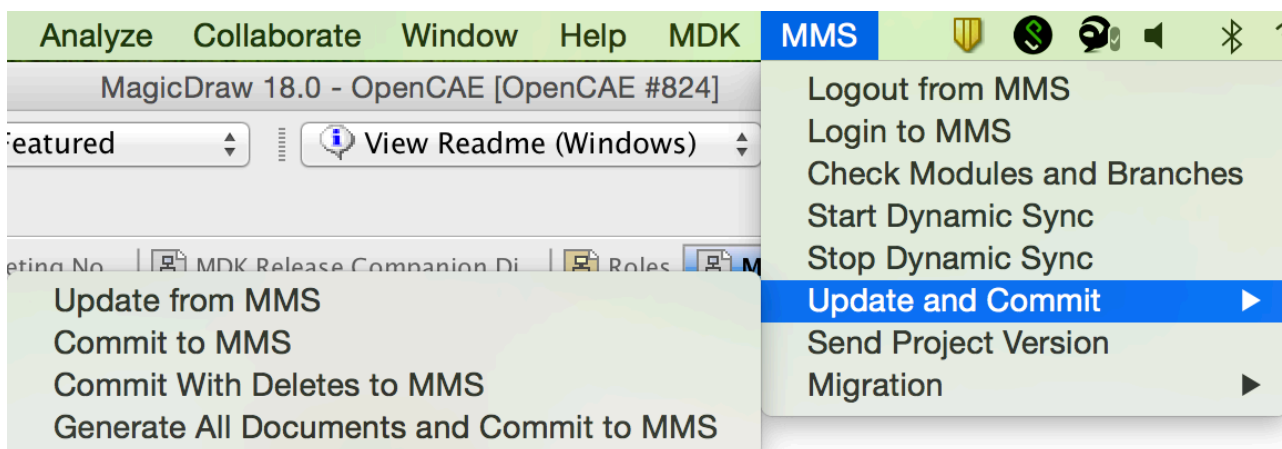
The idea is that once fully synced, the Update and Commit will continuously keep the model in sync through automation (and without much effort on the user's side). Although the user could *technically* decide to not use a synced model, this decision could cause major conflicts and the user would have to use it at his/her own risk. For example: if there are some elements that don't exist on the server and the MDK tries to sync those elements' children, the server will put them in its own Holding_bin. Also, if there are elements on the server that try to get synced back, but there are missing references because the model is not in sync, the update will fail.

Working with a Teamwork Project: When working with a Teamwork Project, users will encounter several extra factors when syncing. For more information, see [Teamwork and MMS](#).

The following operations can be found on the MMS menu:

- [Update from MMS](#)
- [Commit to MMS](#)
- [Commit with Deletes to MMS](#)
- [Generate All Documents and Commit to MMS](#)

MMS Menu:



3.2.2.1 Update from MMS

Update and Commit has been removed from MDK 2.4, and is no longer available for use. It has been replaced with [Coordinated Syncing](#).

This documentation only applies to MDK 2.3.

IMPORTANT: It is **highly** suggested that a MagicDraw project be fully synced before using [Update from MMS](#) for the first time. "Fully synced" means that the model has undergone a full [Manual Syncing](#).

Background:

The "Update from MMS" function triggers the MDK to run a full validation on the entire project; once the validation is complete and the changes are collected from MMS, the MDK begins to automatically update the model to reflect the changes made on View Editor. Anything that exists solely on the View Editor will either be placed in the appropriate [holding_bin](#), document, or containment package. When the Update from MMS function runs, several issues could arise including, conflicts, errors, non-editable elements, etc. When these do occur, the user is prompted to either resolve them (and rerun the function) or they are saved into the model in the [_MMSSync_Package](#). See [Resolving Conflicts and Errors](#) for some examples of conflicts.

Screencast:

In the clip below, the user runs "Update from MMS" on a teamwork project (refer to [Teamwork and MMS](#) for more information, including the pop up message and updating from Teamwork). This process automatically validates the entire model and accepts any changes from View Editor, as can be seen in the "Delta Sync" window. Had there been conflicts (such as documentation change on both sides), the user would have to resolve them individually and rerun the function.

<src="https://jpltube.jpl.nasa.gov/NetworkFileStore/7616/media_720.mp4">Your browser does not support embedded videos.

Link to Video: https://jpltube.jpl.nasa.gov/NetworkFileStore/7616/media_720.mp4

Video Annotations:

1. Begin with a project opened
2. MMS>>Update and Commit>>Update from MMS
3. *For teamwork projects, a message dialog will appear recommending to update the project from teamwork first and commit immediately after
 1. Choose, "Cancel, I will update from teamwork first"
4. Update from Teamwork
 1. Collaborate>>Update Project
 2. Wait until project is updated
5. Run update again
 1. MMS>>Update and Commit>>Update from MMS
 2. This time, choose "Continue with MMS"
6. Wait for project to update from MMS
7. Resolve any conflicts
 1. Once conflicts are resolved, rerun "Update from MMS"
8. Commit to teamwork
 1. Uncheck "Keep Locks"

3.2.2.2 Commit to MMS

Update and Commit has been removed from MDK 2.4, and is no longer available for use. It has been replaced with [Coordinated Syncing](#).

This documentation only applies to MDK 2.3.

IMPORTANT: It is **highly** suggested that a MagicDraw project be fully synced before using [Commit to MMS](#) for the first time. "Fully synced" means that the model has undergone a full [Manual Syncing](#).

Background:

The "Commit to MMS" function triggers the MDK to first run the [Update from MMS](#) function and then commit all changes to the MMS (except for DELETES, see [Commit with Deletes to MMS](#)). This was developed to be similar to the practices of Teamwork and Git, where commits cannot happen until the project is up to date. This is done to try to prevent overwrite when syncing. Once the model is updated, the MDK then tries to commit all the changed model elements to the MMS. The changes include those changes made in the Model as well as the newly changed elements from the Update.

During the Update, conflicts and errors may arise and will appear in the "Delta Sync" window. For more information about common conflicts and specific errors, refer to [Resolving Conflicts and Errors](#).

Screencast:

In this clip, the user demonstrates the events that occur when running the "Commit to MMS" function on a teamwork project (see [Teamwork and MMS](#) for more information about the message). This function automatically runs through the necessary steps and at the end, the changes to the MMS are added to the queue to be sent.

Note: although not shown, a user should Commit to Teamwork so that all the changes that have been made in the model may appear in other users' local projects. The user should also uncheck "Keep locks" so that any conflicts stored in [_MMSSync_Package](#) can be released and cleared through someone else.

<src="https://jpltube.jpl.nasa.gov/NetworkFileStore/7617/media_720.mp4">Your browser does not support embedded videos.

Link to Video: https://jpltube.jpl.nasa.gov/NetworkFileStore/7617/media_720.mp4

Video Annotations:

1. Begin with a project opened
2. MMS>>Update and Commit>>Commit to MMS
3. *For teamwork projects, a message dialog will appear recommending to update the project from teamwork first and commit immediately after
 1. For this screencast, the user chose to "Continue with MMS" since the project was just updated from teamwork
4. Wait for project to update from MMS
5. Resolve any conflicts
 1. Once conflicts are resolved, rerun "Commit to MMS"
6. MMS change requests are added to the queue
7. Notice the job that appears in the MMS Queue window

3.2.2.3 Commit with Deletes to MMS

Update and Commit has been removed from MDK 2.4, and is no longer available for use. It has been replaced with [Coordinated Syncing](#).

This documentation only applies to MDK 2.3.

IMPORTANT: It is highly suggested that a MagicDraw project be fully synced before using "[Commit with Deletes to MMS](#)" for the first time. "Fully synced" means that the model has undergone a full [Manual Syncing](#).

The "Commit to MMS with Deletes" function does exactly what it sounds like: it goes through the same process as [Commit to MMS](#) but with also committing DELETES. Deleting elements from the MMS can sometimes take longer than other commits, so this function has been separated to allow users to continue work with the MMS at a substantial pace and choose to commit the "deletes" at a more convenient time. "Deletes" are committed last in order to satisfy the typical [CRUD](#) order of operations. More information can be found here: [Manual Order of Operation](#).

In this clip, the user calls the "Commit with Deletes to MMS" function. The user then watches as the model updates from MMS, updates the model, commits to MMS, and finally deletes elements successfully.

<src="https://jpltube.jpl.nasa.gov/NetworkFileStore/8728/media_720.mp4">Your browser does not support embedded videos.

Link to video: https://jpltube.jpl.nasa.gov/NetworkFileStore/8728/media_720.mp4

Video Annotations:

1. Begin with a project opened
2. MMS>>Update and Commit>>Commit with Deletes to MMS
3. *For teamwork projects, a message dialog will appear recommending to update the project from teamwork first and commit immediately after
 1. For this screencast, the user chose to "Continue with MMS" since the project was just updated from teamwork
4. Wait for project to Update from MMS
5. Resolve any conflicts
 1. Once conflicts are resolved, rerun "Commit to MMS with Deletes"
6. MMS change requests are added to the queue
7. Notice the jobs that appear in the MMS Queue window, including the deletes - deletes were too fast, but notice "Delete Successful"

3.2.2.4 Generate All Documents and Commit to MMS

Update and Commit has been removed from MDK 2.4, and is no longer available for use. It has been replaced with [Coordinated Syncing](#).

This documentation only applies to MDK 2.3.

IMPORTANT: It is highly suggested that a MagicDraw project be fully synced before using "[Commit with Deletes to MMS](#)" for the first time. "Fully synced" means that the model has undergone a full [Manual Syncing](#).

The MMS option to "Generate All Documents and Commit to MMS" allows the user to **generate every Document in the model**. The generation process creates and updates all the View Instances for each Document. The [Commit to MMS](#) process is the same as previously stated, where it updates the whole model and commits the changes to MMS.

So overall, "Generate All Documents and Commit to MMS" includes the following steps:

1. Update from MMS
2. Generate Views from the Document level for every document in the model
 1. This includes generating and updating View Instances in the model)
3. Validate View Hierarchy on every Document
4. Commit to MMS

In this clip, the user edits 3 different documents and then runs this function. Through regeneration, new views get added, view hierarchies get updated, and other commits are completed, not only on the edited documents, but on all documents in the model.

Link to Video: https://jpltube.jpl.nasa.gov/NetworkFileStore/8750/media_720.mp4

Video Annotations:

1. Edit one or multiple documents (In this video, the user edits 3 documents)
 1. Add a View
 2. Move a view to be a subview
 3. Edit an exposed diagram
2. Go to "MMS" Menu>>"Update and Commit">>"Generate All Documents and Commit to MMS"
3. Wait for Update, Generate, Validate View Hierarchy, and Commit to MMS to automatically run and then complete
 1. If conflicts, resolve them, and rerun "Generate All Documents and Commit to MMS"
 2. Commit or Accept View Hierarchies
 3. Commit outdated images
4. Navigate to View Editor to verify view changes
 1. See new view
 2. See new view hierarchy
 3. See updated image

3.2.3 Manual Syncing

Manual validation is deprecated in MDK 2.3 and above. It is retained for use in resolving specific issues, but models should generally be synced with a more robust operation like [Coordinated Syncing](#) (MDK 2.4) or [Update and Commit](#) (MDK 2.3).

If you have to manually validate, it is best to follow the [CRUD](#) rules about order of operations: 1) Add 2) Update 3) Delete. Using this theory, a [Manual Order of Operations](#) was created to help the user with validation. Although it is not absolutely critical for a user to follow the given order, it tends to save confusion, frustration, and accidental user error. An example of Manual Syncing can be found in the [Create and Generate Documents](#) tutorial.

Manual Order of Operations

Order of Operations for Manual Generation:

1. Validate the document
2. Generate Views
3. Validate Document and Instances
4. Validate View Hierarchy

Order of Operations for Validations:

1. "Does not exist on MMS"
 - Expose[CAE Design::Documents::VE Design Doc::Sear... error | [EXIST] This doesn't exist on MMS | Not Ignored
2. "Exist on MMS" but not in the model
 - Elements created on MMS that needs to be created on MagiDraw
 - Solution: Create MagicDraw Element
3. Changes of attributes associations (i.e. owners, names, docs, values, etc...)
 - 1.2 Critical Deployments error | [NAME] model: 1.2 Critical Deployments, web: Critical Deployments | Not Ignored
4. View Instances
5. View Constraints
 - Latest Release error | [VIEW CONSTRAINT] View constraint is different | Not Ignored
6. "Exist on MMS" but Deleted from MagicDraw
 - <model> error | [EXIST on MMS] Package holding_bin holding_bin_site__18_0_2_bec02f9_1444944898365_672044_69972_no_ project exists on MMS but was deleted from magicdraw | Not Ignored

3.2.4 Resolving Conflicts and Errors

During any of the MMS synchronization operations, the MDK checks for conflicts between the MMS and the MagicDraw model. These conflicts may indicate a difference between what is in the current project model and an edit in View Editor, or they may be caused by data being updated on the MMS by a different application.

- Conflicts will appear either in a Validation window, and each one may be interacted with via the context (right-click) menu in order to commit the MagicDraw version or accept the MMS version.
 - Additional information about the conflict may be available in the context menu as well.
- Error messages will appear on the Notification window, and may not be interacted with.





MDK 2.4:

- Conflicts will not appear for elements that can not be edited by the current user, so they will not appear for any unlocked elements in a Teamwork model.
 - In order to update elements in a Teamwork model, the elements must be locked by the current user. This will also allow the current user to resolve any conflicts that have been pending resolution.
- Images are automatically committed when the View in which they reside is generated, and will never generate a conflict.

MDK 2.3:

- Images are always treated as out of date, and will generate a conflict each time the View they reside in is generated.

Sample Conflicts and other errors that may appear while syncing include:

Error	About	Example
Difference in an element's attributes	Attributes include: Name, Documentation, and/or Values	 1.2 Critical Deployments  error
Difference in ownership, or enumerated value, etc	Ownership has been changed	 Development Wiki  error
[Error] Cannot create element (references or owner not found)	<ul style="list-style-type: none"> • It means there are no model elements for the owner • This usually occurs when doing manual validation • The user needs to "Create MagicDraw Element" of the owner <u>first</u> before resuming validation 	

<p>Server response: 404 {"elements":[{"sysmlid":"(MMS_145....etc)"}]}</p> <p>[Error] Get elements failed.</p>	<ul style="list-style-type: none"> When this occurs, it means that an element has not been successfully posted If this error occurs, the current process will abort and changes won't be compared The solution: <ol style="list-style-type: none"> Locate element in model <ol style="list-style-type: none"> Copy element ID File>>Open Element from URL>>"mdl:/" + {elementID} Validate said element Commit said element from validation window Run operation with MMS again 	
<p>Background Job Is Still Active</p>	<ul style="list-style-type: none"> This occurs when a user tries to commit to the MMS while another request is still sending The commit that was sent when this message was viewed, will be added to the queue Usually a user can just wait until the processes are finished If the job is truly hung, the user can go to the Alfresco link provided in the error message, and edit the state. See instructions here: Background Jobs 	<p>Server message: 409 Background job for project started at T https://cae-ems-uat-origin.jpl.nasa.gov/share/page/document</p>
<p>Not authorized/Don't have permission</p>	<ul style="list-style-type: none"> This error occurs if the user has not logged (at all, or correctly) This error may occur if the user does not have "Contributor" (or higher) permissions for a site The user should make sure that the login is correct and that the correct permissions are given 	<p>[2016.02.18::17:54:15] [INFO] Getting changes from MMS...</p> <p>[2016.02.18::17:54:22] You are not authorized or don't have permission,</p> <p>[2016.02.18::17:54:22] [ERROR] Getting changes from mms failed: Could not connect to broker URL: tcp://[server]. Reason: java.net.ConnectException: Connection timed out: connect</p>
<p>[ERROR] getting changes from mms failed: Could not connect to broker URL: tcp://[server]. Reason: java.net.ConnectException: Connection timed out: connect</p>	<ul style="list-style-type: none"> This is an error that requires a ticket to CAE to remedy. Please write a ticket to CAE Support (SSCAES) on JIRA reporting the server this error occurred on. This error occurs when the server's Java Message Service (JMS) Queue is not turned on, thus requiring DevOps administration to issue a repair. 	

<model> error [CREATE FAILED] Owner or chain of owners not found Not Ignored	<p>When the MDK tries to create an element from the update, it can't because the element's owner isn't found in the model (and the owner is not also in the list of elements to create)</p> <p>RESOLUTION: It is possible that this error does not require a resolution, as it might not affect your model. To be sure you need to determine who the owner is supposed to be, which can be found in the detail popup. The qualifiedName attribute should give an idea of where it's supposed to be and what's missing in the model.</p>	Screenshot of error (TBD)
--	--	---------------------------

3.2.5 Dynamic Syncing

Dynamic Syncing is removed from MDK 2.4, and is not available for use.

This documentation only applies to MDK 2.3.

Dynamic Syncing is known as real-time Syncing. It allows a user to constantly update the MagicDraw model, the MMS Server, and the View Editor all at the same time. The following subviews demonstrate how this function works in terms of model elements and Views.

Requirements for this function:

1. The model must be completely synced
2. (Teamwork Project) The user needs to have the entire model locked
 1. It is recommended to use Dynamic Syncing when working with a very small team (1-3 people)
 2. Alternatively, for bigger groups, it is recommended to do this at a time when the team is unlikely to be working (like 2 AM)
3. User must be ready for all changes to automatically take place
4. (Not required, but suggested) The user can work in a separated Teamwork branch synced with a View Editor task in order to do dynamic syncing. Then the user can merge changes back into "trunk" branch in MagicDraw.
 1. However, since the Teamwork branching is not as robust, refer to Step 2 for using the model, but in the context of the "trunk" branch

Remember: Always check your specific project modeling practices with regard to tasks, sites, content control, etc.

3.2.5.1 Dynamic Sync Elements

Dynamic Syncing is removed from MDK 2.4, and is not available for use.

This documentation only applies to MDK 2.3.

In this clip, the user demonstrates the Dynamic Syncing different types of model elements by editing on the MagicDraw side as well as the View Editor side. This is the real time syncing that has been mentioned because the whole system is being constantly being updated with each edit.

Note: As of now the screencast includes "live" typing, however when the screencasts are rerecorded at some point, this video will be updated. For now, since no functionality has changed, the video will remain the same.

Link to Video: https://jpltube.jpl.nasa.gov/NetworkFileStore/6464/media_720.mp4

Video Annotations:

1. Open branched teamwork server project in Magic Draw
2. Go to the tag in View Editor with the same name as the magic draw teamwork server project name
3. The project branch is called Demo.
4. Navigate to the document in the tag.
5. Start Dynamic Sync
 - If Project version is out of date, Click MMS on the top right of the window pane and select "Send Project Version".
6. Text of elements are automatically synced.
7. You can also sync mass values.
8. New Elements created in Magic Draw are automatically synced.
9. New Elements created in View Editor are synced back to the magic draw project.
10. Stop Dynamic sync when you are done.
11. To Merge tags, go back to the root of the site.
12. Click the compare button.
13. Type the tags you want to merge.

14. If the comparison is too big, an email will be sent.
15. You can see the differences between the branches.
16. To Merge click the merge button.

3.2.5.2 Dynamic Sync Views

Dynamic Syncing is removed from MDK 2.4, and is not available for use.

This documentation only applies to MDK 2.3.

In this clip, the user demonstrates the Dynamic Syncing of Views by editing on the MagicDraw side as well as the View Editor side. This is the real time syncing that has been mentioned because the whole system is being constantly being updated with each edit.

Note: As of now the screencast includes "live" typing, however when the screencasts are rerecorded at some point, this video will be updated. For now, since no functionality has changed, the video will remain the same.

<src="https://jppltube.jpl.nasa.gov/NetworkFileStore/4361/media_720.mp4">Your browser does not support embedded videos.

https://jppltube.jpl.nasa.gov/NetworkFileStore/5430/media_720.mp4

Video Annotations

1. Start Dynamic Sync
2. Creating a view on MD
3. View element automatically gets uploaded to the model repository.
4. Validate the view hierarchy of the document.
5. Commit the View hierarchy.
6. The view shows up on View Editor.
7. Changes to the documentation automatically get sent. They appear on View editor.
8. View Editor changes automatically get sent to the MD model.
9. Elements created on View Editor also get created on MD.
10. Model information can also be edited and changed .

3.3 MDK Specific Features

The following subviews are designed to provide users with information about key MDK features, including explanations of their use and how they affect a modeler on a daily basis.

3.3.1 Documents and Views

Documents and Views can be created and modeled in MagicDraw and generated to View Editor.

Documents are a key part of systems engineering; through the practice of Model Based System Engineering (MBSE), documents have also been adapted so that they are able to be generated and produced from a model. Documents consist of Views, which are sections of a document. Both Documents and Views are based on Classes in UML and therefore are treated as elements in the model, with their own associated attributes and formatting metadata.

Implications:

A SysML model is not required to produce a document; however, the main interface between MagicDraw and View Editor, the web application, is primarily done through the interactive capabilities of documents and views. They are used to provide access to modeling data outside of the model itself.

Documents and views are built and configured using [Viewpoints and Viewpoint Methods](#) . More information can be found in the next view.

3.3.2 Viewpoints and Viewpoint Methods

One of the defining moments for widespread adoption of SysML at JPL was when the community created "DocGen" (see [DocGen User's Guide](#)), a precursor of the MDK plug-in, which gives MBSE practitioners the ability to produce documents from their models. In order to create these documents, the view and viewpoint method was introduced.

A pattern is a set of rules governing model construction that provide standardization and consistency across models. The method for document creation is one such pattern adopted by the Object Management Group (OMG, the standards body behind SysML) and was incorporated into SysML 1.4.

Viewpoints and Viewpoint Methods can be created in MagicDraw and will result in View Editor. Refer to [Create and Generate Documents](#) and [Create a Reusable Cover Page](#) for examples of using Documents, Views, Viewpoints, Viewpoint Methods, and Exposing elements.

Viewpoints:

- Can be thought of as the "compiler" for a view.
- Dictate what will actually be displayed in a view.
- Uses the viewpoint methods and the exposed elements to produce a view.

Viewpoint Methods:

- Are a set of rules that govern model construction to give standardization and consistency across models.
- Can be thought of as the "constructor".
- Consist of activity flows that are specialized to be in charge of building what the view will be.
- The most common viewpoint method are those that make Rapid Tables - they take the exposed elements and iteratively go through them to produce the desired table. More information on how to build such a table can be found here: [Create and Generate a Rapid Table](#) .

Implications:

Using the model, a user will consistently utilize Viewpoints and Viewpoint Methods to construct documents and views. These documents and views may contain any number of important modeling information, based on what the user desires. The user will then generate these formatted and configured documents and views so that users on View Editor can have access to the model information and data.

3.3.3 _MMSSync_ Package

The _MMSSync_ package is part of a number of sync solutions for the MMS and View Editor. Its purpose is to allow continual collaboration while a project is consistently being updated by multiple users. It does this by capturing changes between the model and the MMS server and persisting them in the model. The idea is that when an element is updated from the server, whoever is using the syncing options will capture the changes and the MDK will try to update the model itself. However, if there is some non-editable content or errors of any sort that would prevent the MDK from automatically updating, these changes will be saved in the _MMSSync_ package. Once stored, the changes will be tried to be resolved/updated the next time it is run.

Although the _MMSSync_ package looks like an ordinary folder, **users should NOT try to edit the content**. A user may try to unlock the elements (for Teamwork), but should not edit the elements themselves. If the user does not solely use syncing options for updating this folder, there could be a slight possibility for data loss.

Usage:

MDK 2.4 and MDK 2.3:

- Every project that is connected to MMS will have an _MMSSync_ package.
- References to all changed (categorized as created, updated, and deleted) elements on both MagicDraw and MMS will be persisted in elements in this package.
- These changed elements will be analyzed during sync operations to generate the necessary operations to sync the model.
- Each element in this package will be categorized and timestamped.
- Each element in this package is automatically managed and deleted by the MDK. No user intervention is required or recommended. See version specific implications for more information.
- In the case that elements in this package cannot be deleted due to insufficient locks, a second element is created that signifies to sync operations that the first one can be safely ignored as its contents have already been processed.

MDK 2.4

- Elements in this package are **not** automatically locked by the MDK or deleted. As such, it is recommended with this version of MDK for users to lock the _MMSSync_ package recursively before running a sync operation at regular intervals.

MDK 2.3:

- Elements in this package are automatically locked by the MDK and deleted. As such, it is recommended with this version of MDK for users to unlock these elements whenever possible so that they can be cleared by the next person who runs a sync operation. This can be done by recursively unlocking from the _MMSSync_ package or through the main menu by using "Collaboration" > "Unlock All".

3.3.4 holding_bin

The holding_bin package is created for every project (or package) synced to an alfresco site. Its purpose is to provide an easy to locate package where documents and elements created in ViewEditor will be stored in MagicDraw until a more appropriate location is chosen by the user.

The root level holding_bin will contain hierarchical subfolders to organize elements by their creation time: YY_#### (year), MM_## (month of year, starting from 0 for January), DD_## (day of month, starting from 1), HH_## (hour or day, in 24 hour notation). Elements created in ViewEditor will be stored in the folder corresponding to their creation time, down to the hour.

Usage:

- The holding_bin package is created after the site's Cover Page is generated.
 - This automatic generation occurs when a user navigates to a site for the first time, as there is no need to store elements generated in ViewEditor until the first one is generated.
- Elements imported into the the holding_bin through a synchronization action can be moved to other locations in the project as appropriate.

3.4 Selected Modeling Tutorials

The following subviews are dedicated to guiding users through some of the most commonly used documentation features that MagicDraw and View Editor have to offer.

NOTE: These tutorials were made for MDK 2.3, and practices in MDK 2.4 may deviate from them slightly. Any differences in procedure will be noted in the tutorial proper.

Table of Contents (Summarized Highlights)

1. [Create and Generate Documents](#) - Collection of fundamental tutorials for the average user
 1. [Create A Document With A View](#) - The most in depth and comprehensive tutorial is the in the sense that it walks through the steps of creating, generating, and validating through the Update and Commit options as well as the Manual syncing options.
 2. [Insert Diagram as Image](#) - The simplest demonstration of Viewpoints and Viewpoint methods
 3. [Create and Generate a Rapid Table](#) - The demonstration of one of the most common and useful tools that enable View Editor users to truly interact with MagicDraw model elements
2. [Site Characterization](#) - Provides a means of organization of elements for a modeler and documents for a View Editor user
3. [Create Enumerated Values](#) - Enables an element value to have a specific set of choices that can be reflected in View Editor as a drop down list
4. [Create Toggable Boolean Values](#) - Enables an element value to be boolean and be reflected in View Editor as a check box
5. [Systems Reasoner](#) - Brief introduction to the practice of Block Specific Type (BST) and the usefulness it provides for modelers

3.4.1 Create and Generate Documents

The following subviews focus on foundational training to get any MagicDraw/MDK user to be able to interact with the MMS and subsequently View Editor. The goal at the end of these subviews is for the MDK user to be able to [Create A Document With A View](#) , [Generate Views and Commit to MMS](#) , and [Manually Generate Views and Validate with MMS](#) .

3.4.1.1 Create A Document With A View

In this clip, the user creates a new (blank) document in MagicDraw using MDK's tools for Documents and Views. Committing these new elements is required for proceeding to the next step, [Generate Views and Sync with MMS](#) , where the user will generate the document so it appears on View Editor.

NOTE: This tutorial was put together for MDK 2.3. MDK 2.4 uses a more streamlined synchronization workflow. The below tutorial can be used, but all references to "Update and Commit", "Commit to MMS" should be replaced with a save (local model) or commit to teamwork (teamwork model) to trigger a coordinated sync. References to "Generate and Commit to MMS" should be replaced with a save (local model) or commit to teamwork (teamwork model) to trigger a coordinated sync followed by a "Generate Views" on the appropriate target. References to "Validate View Hierarchy" may simply be skipped.

Setup: Model has been synced to it's own Alfresco site (See [Create an MDK-Enabled Project](#) for more information on this process) A new package is created for the purpose of simple organization. Each modeler should follow the modeling practices of his/her project.

<src="https://jpltube.jpl.nasa.gov/NetworkFileStore/8662/media_720.mp4">Your browser does not support embedded videos.

Link to video: https://jpltube.jpl.nasa.gov/NetworkFileStore/8662/media_720.mp4

Video Instructions:

1. Create a new View Diagram
 1. Name new diagram
2. Create new Document
 1. Name Document
3. Create new View

1. Name new view
4. Create a "Directed Composition" relation from the new Document to the new View

3.4.1.1.1 Generate Views and Sync with MMS

NOTE: This tutorial was put together for MDK 2.3. MDK 2.4 uses a more streamlined synchronization workflow. The below tutorial can be used, but all references to "Update and Commit", "Commit to MMS" should be replaced with a save (local model) or commit to teamwork (teamwork model) to trigger a coordinated sync. References to "Generate and Commit to MMS" should be replaced with a save (local model) or commit to teamwork (teamwork model) to trigger a coordinated sync followed by a "Generate Views" on the appropriate target. References to "Validate View Hierarchy" may simply be skipped.

There are two ways to generate views and interact with the MMS in general - using "Generate Views and Commit to MMS" or manually generating views/validating. CAE highly encourages the use of the "Generate Views and Commit to MMS". It is designed to be a "one-click" interaction where MDK takes care of validations and "order of validations" for the user. However, for the convenience of the user and to illustrate all the steps that "Generate Views and Commit to MMS" does, a second screencast has been provided for manually generating and validating a document/view.

Generate Views and Commit to MMS

IMPORTANT: It is highly suggested, though not absolutely required, that a MagicDraw project be fully synced before using these options for the first time.

In this clip, the user is working on a teamwork hosted project and has already created a document and view ([Create A Document With A View](#)). The user now demonstrates how to use "Generate Views and Commit to MMS" from the document's View Diagram. The user then shows the newly generated document on View Editor.

<src="https://jpltube.jpl.nasa.gov/NetworkFileStore/8664/media_720.mp4">Your browser does not support embedded videos.

Link to video: https://jpltube.jpl.nasa.gov/NetworkFileStore/8664/media_720.mp4

Note to any user who is working on a project on a teamwork server: There will be a pop up window regarding updating the project before interacting with MMS. Please refer to [Teamwork and MMS](#) for more information, including what is the best option to choose.

Video Annotations:

1. Generate Views and Commit to MMS
 1. "Generate Views and Commit to MMS" on the document level
 2. When message pops up about updating from teamwork, click "Cancel, I will update from teamwork first"
 3. Go to "Collaborate" drop down menu and click "Update Project"
 4. Upon completion, "Generate Views and Commit to MMS" again
 5. This time, when message comes up, click "Continue with MMS"
2. *Watch it all go
3. If there are conflicts
 1. Resolve conflicts by accepting from the web or committing the model
 2. Re-run "Generate Views and Commit to MMS"
4. Validate the hierarchy via the Validation Hierarchy window
5. *Navigate to document on view editor
6. Show that the view is there
7. Go back to MD, go to "Collaborate" dropdown, "Commit to server", Comment "Created and generated(Document name) "

Manually Generate Views and Validate with MMS

In this clip, the user is working on a teamwork hosted project and has already created a document and view ([Create A Document With A View](#)). The user now manually generates views, validates the document/views, validates hierarchy, validates instances/constraints, and confirms the document appears on View Editor.

<src="https://jpltube.jpl.nasa.gov/NetworkFileStore/8666/media_720.mp4">Your browser does not support embedded videos.

Link to video: https://jpltube.jpl.nasa.gov/NetworkFileStore/8666/media_720.mp4

Video Annotations:

1. Manually Generate Views and Validate Models with MMS
 1. From the Document Diagram or Containment Tree, "Generate Views" from the Document level
 2. See new View Instances package

2. More info on View Instances can be found in the description
 1. In the containment tree, "Validate Models" on the parent element most appropriate
 2. (For this clip) Validate Models on the package that contains the documents/views and View Instances
3. Note: 404 (aka "element not found") message will appear because the Instances do not exist on MMS...yet
4. Order errors in validation window by "Message" column
 1. Follow normal CRUD order of operations rule: Add, Update, Delete
 2. Commit "[EXIST] This doesn't exist on MMS"
 3. (Explanation of Export on MagicDraw vs Export on Server can be found in the description)
 4. (For this clip) Export on MagicDraw
 5. Accept from Web or Commit from Model for other conflicts
 6. Commit View Constraints
 7. Create MagicDraw element or Delete from MMS - not applicable here
5. Validate View Hierarchy
 1. Commit Hierarchy
6. (View Editor) Navigate to document
 1. See View
7. If teamwork project - return to MagicDraw, "Commit to server" with comment

3.4.1.2 Insert Diagram as Image

NOTE: This tutorial was put together for MDK 2.3. MDK 2.4 uses a more streamlined synchronization workflow. The below tutorial can be used, but all references to "Update and Commit", "Commit to MMS" should be replaced with a save (local model) or commit to teamwork (teamwork model) to trigger a coordinated sync. References to "Generate and Commit to MMS" should be replaced with a save (local model) or commit to teamwork (teamwork model) to trigger a coordinated sync followed by a "Generate Views" on the appropriate target. References to "Validate View Hierarchy" may simply be skipped.

In this clip, the user demonstrates one of the most basic ways of using viewpoints and viewpoint methods to construct what will be in a view. For more information about what viewpoints and viewpoint method are, refer to [Viewpoints and Viewpoint Methods](#). More detailed information about creating specific viewpoints and methods can be found in the [DocGen User's Guide](#).

<src="https://jpltube.jpl.nasa.gov/NetworkFileStore/6607/media_720.mp4">Your browser does not support embedded videos.

Link to video: https://jpltube.jpl.nasa.gov/NetworkFileStore/6607/media_720.mp4

Video Annotations:

1. Create a new View
2. Create new View Point
 1. Name new View Point
3. Create a "Conforms" relation from the new View to the new View Point
4. Select a diagram from the containment tree (to insert as an image) and drag it into the view diagram of the chosen document
5. Create an "Expose" relation from the chosen view to the diagram
6. Select the Viewpoint in the containment tree and create a new Viewpoint Method Diagram
7. Navigate to the new Viewpoint Method Diagram
8. Create a View Point activity
 1. Insert "Initial Node"
 2. Insert "Image Template"
 3. Insert "Activity Final"
 4. Connect activity flow (use the stamp feature to speed connections)
9. From the View Diagram, on the document level, "Generate Views and Commit to MMS"
 1. *If teamwork popup, refer to [Teamwork and MMS](#) for more information, including which option to choose and when.
 2. Address conflicts/issues by going through the validation windows that appeared during the update/generate/commit process - including committing view hierarchy and outdated image.
 3. Once conflicts are resolved, re-run update/generate/commit

3.4.1.3 Create and Generate a Rapid Table

NOTE: This tutorial was put together for MDK 2.3. MDK 2.4 uses a more streamlined synchronization workflow. The below tutorial can be used, but all references to "Update and Commit", "Commit to MMS" should be replaced with a save (local model) or commit to teamwork (teamwork model) to trigger a coordinated sync. References to "Generate and Commit to MMS" should be replaced with a save (local model) or commit to teamwork (teamwork model) to trigger a coordinated sync followed by a "Generate Views" on the appropriate target. References to "Validate View Hierarchy" may simply be skipped.

In this screencast, the user creates a table in MagicDraw (also called "rapid table") using viewpoints and viewpoint methods. The table is then pushed to a view and shown in View Editor. For more detailed information, about using [Viewpoints and Viewpoint Methods](#) refer to the [DocGen User's Guide](#).

<src="https://jpltube.jpl.nasa.gov/NetworkFileStore/6604/media_720.mp4">Your browser does not support embedded videos.

Link to video: https://jpltube.jpl.nasa.gov/NetworkFileStore/6604/media_720

Video Annotations:

1. Create a new View in the Document for the table
2. Create new Viewpoint
 1. Name new Viewpoint
 2. Connect Viewpoint to new View via "Conforms"
3. Drag existing package of blocks onto diagram
 1. Connect package via "Expose"
4. Create View Point Method Diagram
 1. Right click the view point in the containment tree->Create Diagram->MDK->View Point Method Diagram
5. Start building the activity - the end result will create a table that has the name of the blocks. Columns for two of their properties will be added in advanced screencasts
 1. Create "Initial Node"
 2. Create "CollectOwnedElements"
 3. Create "FilterByStereotypes"
 1. Double click the new block, or right click and go to specification
 2. Find "Stereotypes" -> search for "Block" and select Block[Class] Sysml::Blocks
 4. Create "Table Structure"
 1. Name the table. This name will display in the view
 2. Create "Initial Node" inside the table.
 3. Create "TableAttributeColumn" inside the table.
 1. Name the column. This name will display in the view
 2. Double click the new block
 3. Find "Desired Attribute" -> select desired attribute from options
 4. Create "Activity Final" inside the table.
 5. Create "Activity Final" outside the table, in the activity.
 6. Connect all control flows
6. From the View Diagram, on the document level, "Generate Views and Commit to MMS". If working on a Teamwork server, remember to update first.
 1. Address conflicts/issues by going through the validation windows that appeared during the update/generate/commit process - including committing view hierarchy and outdated image.
 2. Once conflicts are resolved, re-run update/generate/commit

3.4.2 Site Characterization

NOTE: This tutorial was put together for MDK 2.3. MDK 2.4 uses a more streamlined synchronization workflow. The below tutorial can be used, but all references to "Update and Commit", "Commit to MMS" should be replaced with a save (local model) or commit to teamwork (teamwork model) to trigger a coordinated sync. References to "Generate and Commit to MMS" should be replaced with a save (local model) or commit to teamwork (teamwork model) to trigger a coordinated sync followed by a "Generate Views" on the appropriate target. References to "Validate View Hierarchy" may simply be skipped.

Site Characterization offers users the ability to better organize projects by allowing Packages within the model to be appear as if they were sub-sites. There are two advantages to this, better organization for large models and the ability to fine tune access permissions from whole project scale down to individual packages.

In this clip, the user demonstrates how to create these characterized sites and how it allows organization on View Editor.

<src="https://jpltube.jpl.nasa.gov/NetworkFileStore/8742/media_720.mp4">Your browser does not support embedded videos.

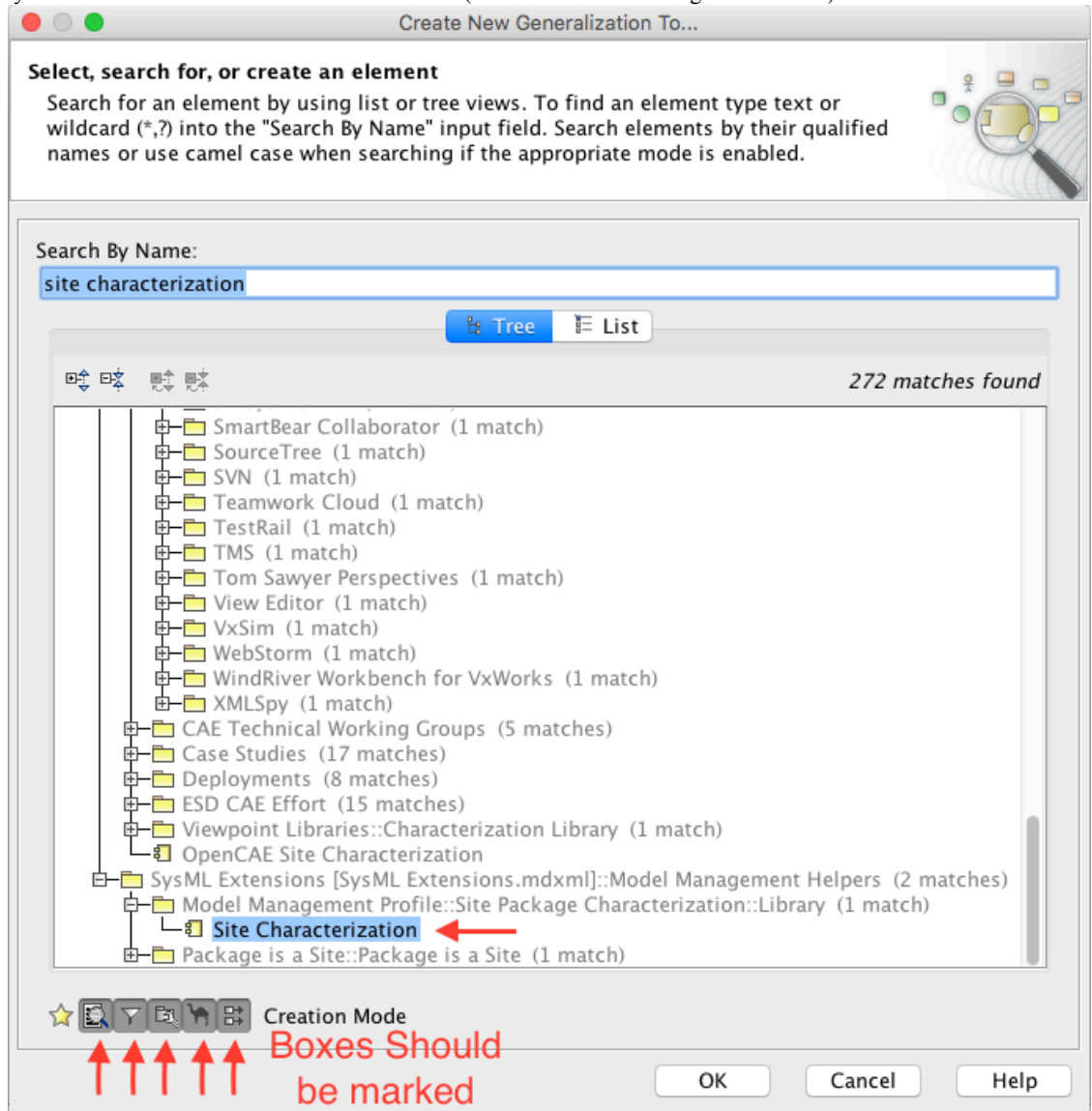
Link: https://jpltube.jpl.nasa.gov/NetworkFileStore/8742/media_720.mp4

Description: In order to better organize your EMS Projects on the web, MDK incorporates a feature to allow Packages within the model to appear as if they were sub-sites. . Site Characterization can be performed on any number of packages to create a MagicDraw Containment Tree like model on the view editor.

Video Instructions:

1. Begin with a new Package that you would like to have characterized as a site.
2. Select the intended Package and create a "Component" element within it. This element will define the site characterization.
 1. Name the new element (normally "{package name} Site Characterization").
3. Create a Generalization relation (outgoing) for the new component.
 1. Right click the component block ->Create Relation ->Outgoing -> Generalization

2. Select the Sysml Extension element “Site Characterization” (within the Model Management Profile).



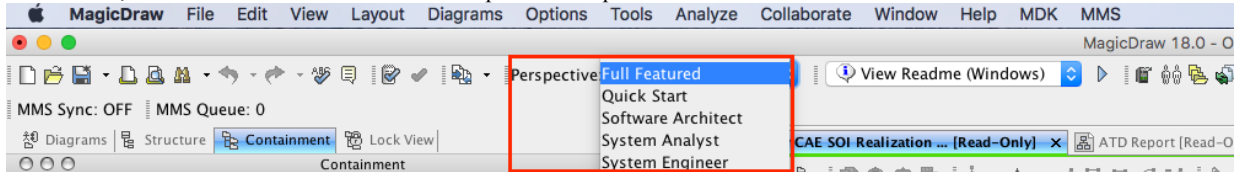
2. If you don't see that specific profile, make sure that all the bottom filters are turned on.
4. Create a Dependency relation (outgoing) for the new component.
 1. Right click the component block ->Create Relation ->Outgoing -> Dependency
 2. Select the package that is to be characterized. (The package that you want the site to be under)
5. Apply the “characterizes” Stereotype to the Dependency relation.
 1. Right click the dependency relation
 2. select Stereotype
 3. choose characterizes
 4. click apply
6. (Optional) Create a Block Definition Diagram for the relationships. This provides a visual reference for the characterization.
 1. Drag out the different components of the site package onto the BDD.
7. Commit to MMS.
 1. Collaborate (located Top Bar) -> Commit Changes to server
 2. Add comment
 3. If popup occurs asking to update, update (say yes)
8. Navigate to View Editor to verify that the package now appears. Wait for the element to fully appear, and for the MMS elements to be generated.
9. Return to MagicDraw and "Commit to MMS". This will import the elements that were automatically created by MMS.
 1. Click "Continue to MMS" when message pops up
10. Commit your changes to teamwork.

FAQs:

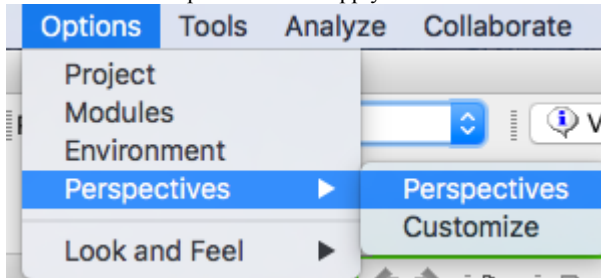
Why can't I see the "Component" element?

Sometimes a user cannot see a "Component" element because of the MagicDraw "Perspective" that is selected. A user needs to be in the "Full Featured" Perspective to see the "Component Element". There are two ways to check/change the perspective:

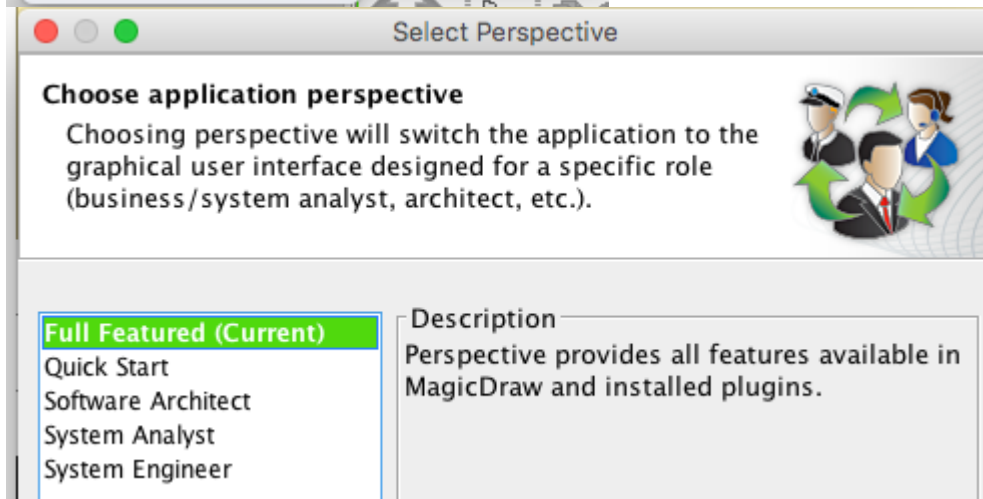
1. From the toolbar menu, select "Full Featured" from the "Perspective" dropdown



2. The selected Perspective will then be shown
2. Go to "Options toolbar>>Perspectives" and Apply "Full Featured"



1.

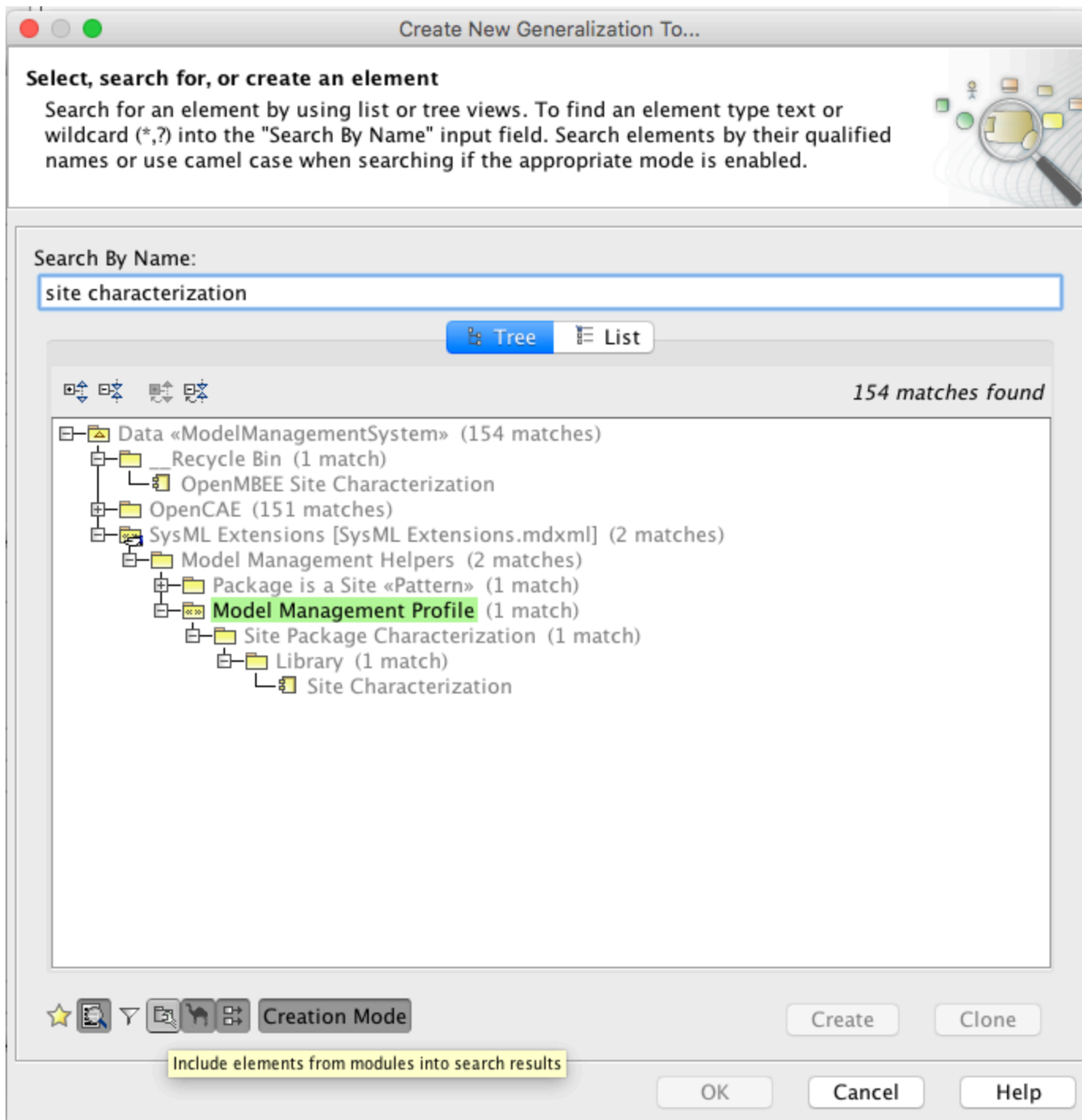


- 2.
3. Select "Full Featured" and click "Apply"
4. Current Perspective is indicated as above

User should now be able to see/create the "Component" element.

[Why can't I see the "Model Management Profile" package when searching for "Site Characterization"?](#)

Sometimes when searching for elements, certain viewing options are automatically selected/deselected to be filtered. In this case, most likely the user needs to click "Include elements from modules into search results" in order to see the "Model Management Profile". It is sometimes also useful to deselect "Filter to reduced list". Both are shown in the picture below.



3.4.3 Create Enumerated Values

NOTE: This tutorial was put together for MDK 2.3. MDK 2.4 uses a more streamlined synchronization workflow. The below tutorial can be used, but all references to "Update and Commit", "Commit to MMS" should be replaced with a save (local model) or commit to teamwork (teamwork model) to trigger a coordinated sync. References to "Generate and Commit to MMS" should be replaced with a save (local model) or commit to teamwork (teamwork model) to trigger a coordinated sync followed by a "Generate Views" on the appropriate target. References to "Validate View Hierarchy" may simply be skipped.

One of the more sophisticated features of View Editor is the option to have values be enumerated values. This means that a user will only be allowed to set an element's value to a specific set of choices. In View Editor, this is shown as a drop down list. This can be extremely useful for elements that are similar in makeup but have different properties and different values.

In this clip, the user demonstrates how to create enumerated values so that they may be seen as drop down lists on View Editor. The user adds a column to an existing rapid table (see [Create and Generate a Rapid Table](#)) to show one of the many options that a user can choose to reveal different model elements.

Note: Unfortunately this clip does not currently provide written "Real time" instructions nor a voice over. However, the functionality is correct so it is still being provided for the user. A new video with comprehensive in-video instructions and edited user friendly footage is on its way.

Link to video: https://jpltube.jpl.nasa.gov/NetworkFileStore/8730/media_720.mp4

3.4.4 Create Toggable Boolean Values

NOTE: This tutorial was put together for MDK 2.3. MDK 2.4 uses a more streamlined synchronization workflow. The below tutorial can be used, but all references to "Update and Commit", "Commit to MMS" should be replaced with a save (local model) or commit to teamwork (teamwork model) to trigger a coordinated sync. References to "Generate and Commit to MMS" should be replaced with a save (local model) or commit to teamwork (teamwork model) to trigger a coordinated sync followed by a "Generate Views" on the appropriate target. References to "Validate View Hierarchy" may simply be skipped.

Another relevant feature of View Editor is the option to have an element's value be Boolean. In View Editor, this is shown as a toggable check box indicating if a value is "√True" or "False". This can be extremely useful for all elements that require a quick Boolean answer.

In this clip, the user demonstrates how to create toggable Boolean values so that they may be seen check boxes on View Editor. The user adds a column to an existing rapid table (see [Create and Generate a Rapid Table](#) for the first column and [Create Enumerated Values](#) for the second column) to show one of the many options that a user can choose to reveal different model elements.

Note: Unfortunately this clip does not currently provide written "Real time" instructions nor a voice over. However, the functionality is correct so it is still being provided for the user. A new video with comprehensive in-video instructions and edited user friendly footage is on its way.

Link to Video: https://jpltube.jpl.nasa.gov/NetworkFileStore/6578/media_720.mp4

3.4.5 Systems Reasoner

NOTE: This tutorial was put together for MDK 2.3. MDK 2.4 uses a more streamlined synchronization workflow. The below tutorial can be used, but all references to "Update and Commit", "Commit to MMS" should be replaced with a save (local model) or commit to teamwork (teamwork model) to trigger a coordinated sync. References to "Generate and Commit to MMS" should be replaced with a save (local model) or commit to teamwork (teamwork model) to trigger a coordinated sync followed by a "Generate Views" on the appropriate target. References to "Validate View Hierarchy" may simply be skipped.

System Reasoner is the tool that provides a modeler with the ability to practice Block Specific Types (BST) modeling. The idea is that a user can create one element, in this case a Block, add some properties and attributes, and then use it as a basis for other specialized versions. It may sound like copying and pasting might achieve this same result, but that is over simplified. Using system reasoner means that there are relations to the foundational blocks and their other specializations and therefore creates a more organic system. If any of the blocks change, the system reasoner set up allows that change to ripple through the rest of the blocks or leave everything else alone.

In this clip, the user creates a base type block, "Generic Flying Machine", and a specialized machine, "Super Flyer 1". The user then demonstrates that changing the value on a part property of Super Flyer 1 does not in fact change the original but rather makes the Super Flyer 1 unique.

Note: Although the clip is about 7 minutes long, most of that time is spent on typing and creating the original block to be specialized. There are also several "Optional" steps to give more context to users. The fundamental part of specializing one block to an already existing block with properties, is only one mouse-click function (see step 7 around the 5:00 minute mark).

Note 2: Step 4 (around 2:52) is also optional when creating the base block; however, it promotes taking full advantage of System Reasoner. This step allows users to add properties to the subblocks which are related to the existing properties. In general, it makes the base type more flexible when specialized.

<src="https://jpltube.jpl.nasa.gov/NetworkFileStore/4335/media_720.mp4">Your browser does not support embedded videos.

Link to Video: https://jpltube.jpl.nasa.gov/NetworkFileStore/4335/media_720.mp4

Video Annotations:

1. Create "Generic Flying Machine" - 00:55
2. Add Part Properties - 01:23
3. Create corresponding type blocks - 02:09
4. Set Part Properties type to corresponding type blocks - 02:52
5. (Optional) Create IBD for Generic Flying Machine - 03:46
6. Create "Super Flyer 1" - 04:33
7. Specialize to Generic Flying Machine - 05:00
8. Redefine Attribute and Specialize Type Recursively - 05:36

9. (Can ignore warning about inconsistent names) - 06:13
10. Set new default for part property specifically for Super Flyer 1 - 06:37
11. (Optional) Create IBD for Super Flyer 1 - 07:21
12. Show default value to see the difference - 08:12
13. (Optional) Create BDD to show inheritance - 08:42
14. Display paths - 09:34

3.4.6 Create a Reusable Cover Page

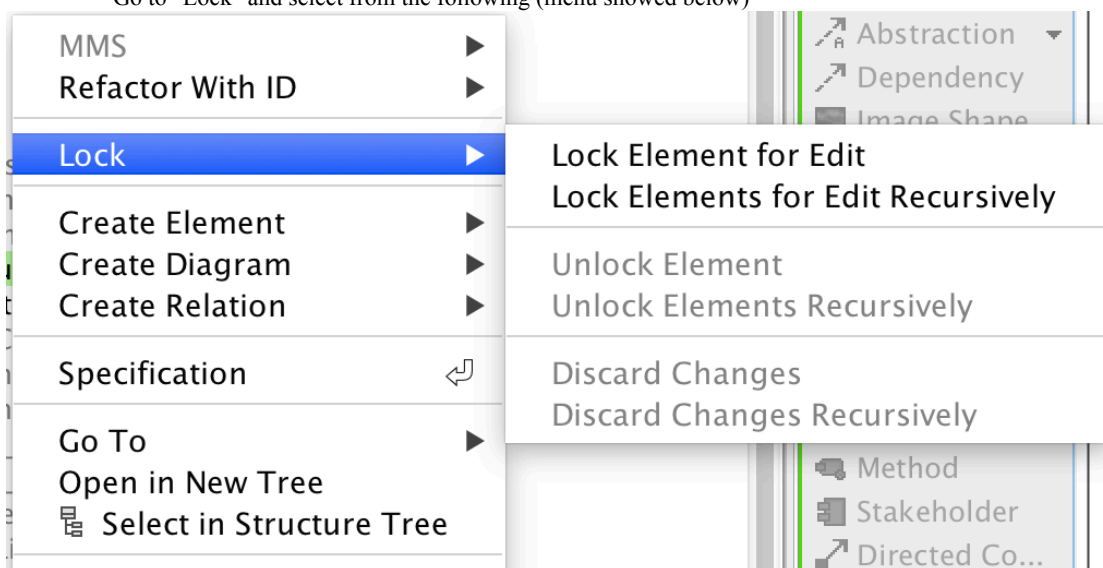
4 Use Teamwork

The following subviews include information and instructions about using a project that is hosted on the teamwork server. More information of the CAE teamwork server can be found on the [Model Development Kit \(MDK\)](#) landing page.

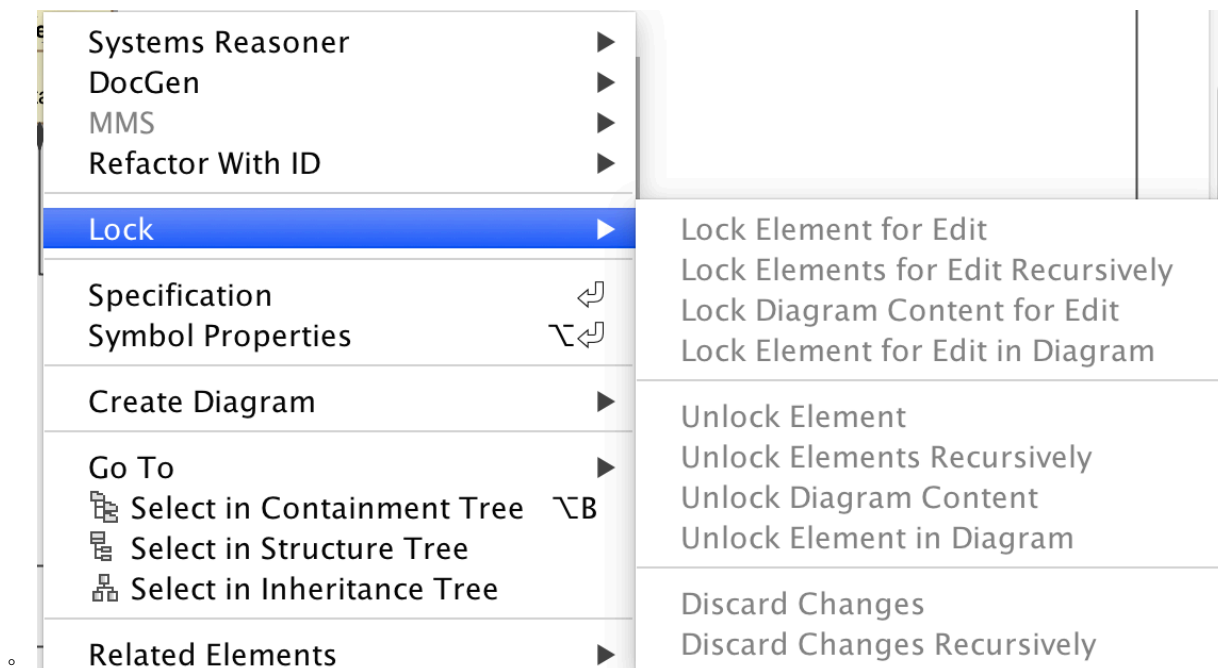
4.1 Everyday Interaction

There are several key actions/practices that a user should know and follow when working on a project model hosted on a teamwork server. The following outlines the most prominent of these as well as those that are used on a daily basis. For more information about other teamwork options and specific teamwork actions, refer to the rest of the subviews in this view, [Use Teamwork](#).

- [Open a Teamwork Project](#)
- When working on a project, the project version aka the commit number, will appear in the MagicDraw header. This will be incremented by 1 every time a user commits the project to the server.
- A user will only be able to edit elements if those elements are **locked**:
 - To lock an element from the containment tree:
 - Find element in the containment tree
 - Right Click
 - Go to "Lock" and select from the following (menu showed below)



- Locking options:
 - Lock Element for Edit - allows user to only edit the selected option
 - Lock Elements for Edit Recursively - allows user to edit the selected option as well as any children/relations of said element
 - Unlock Element - allows user to unlock the selected element only so that other users can lock them. May require a teamwork "commit to server".
 - Unlock Elements Recursively - allows user to unlock the selected element as well as any children/relations of said element, so that other users can lock them. May require a teamwork "Commit to Server".
 - Discard Changes - allows users to discard changes that have been done to only the selected (locked) element
 - Discard Changes Recursively - allows users to discard changes that have been done to the selected (locked) element as well as any children/relations of said element
- To lock an element from a diagram:
 - Find element in a diagram
 - Right Click
 - Go to "Lock" and select from the following (menu showed below)



- Refer to the above "Locking options" for most of the options.
- Diagram specific locking options:
 - Lock Diagram Content for Edit:
 - Lock Element for Edit in Diagram
 - Unlock Diagram Content
 - Unlock Element in Diagram
- Options found on the Collaborate menu. For more information, refer to [Collaborate Menu Overview](#)
 - "Update Project" should be done fairly frequently in order to avoid data loss or overwriting. It allows users to be working on the most up to date version of the project. The commit number in the MagicDraw header will reflect this change if there has been a commit since the user last opened/updated the project.
 - Committing to the teamwork server is essential for collaboration. CAE recommends committing to the server often so that collaboration between users can be facilitated efficiently and the chance of data loss is minimal.
 - When a user is committing, the option "Keep Locks" should only be checked if the user is going to continue actively editing and interacting with the locked elements in the model.
 - A user should commit and uncheck "Keep Locks" anytime the user is done working or is going to be inactive for a significant amount of time (30 minutes or more of inactivity is recommended to commit and unlock).
 - When using [Update and Commit](#) options, the user should commit and uncheck "Keep Locks" as often as is reasonable because of the [_MMSSync_Package](#)
 - "Unlock All" allows a user to unlock everything a user currently has locked. This is highly recommended when using [Update and Commit](#) options because of the [_MMSSync_Package](#)
 - "Update Locked Information" is useful for the user when trying to find out what elements are locked. Locked information will also be updated if a user does "Update Project"

Teamwork and MMS

At the moment, there is no "hard wired" connection between Teamwork and MMS; however, there is a work-around for this, which is demonstrated in the clip below. This work-around is established to prevent accidental data loss that could happen due to outdated information in the project that the user is editing versus what is on the server.

IMPORTANT: Teamwork users that use [Update and Commit](#) , **MUST** always commit to Teamwork, as well, to ensure consistent data. If a user does not properly do so, the changes on View Editor cannot be retrieved with Update anymore as the JMS queue on View Editor will be emptied on update. At that point, changes will require manual syncing.

For this clip, the user was already signed into MMS and is regenerating a document to show the "interaction" between the teamwork server and MMS.

<src="https://jpltube.jpl.nasa.gov/NetworkFileStore/8643/media_720.mp4">Your browser does not support embedded videos.

Link to video: https://jpltube.jpl.nasa.gov/NetworkFileStore/8643/media_720.mp4

Video Annotations:

Generate views on the document level

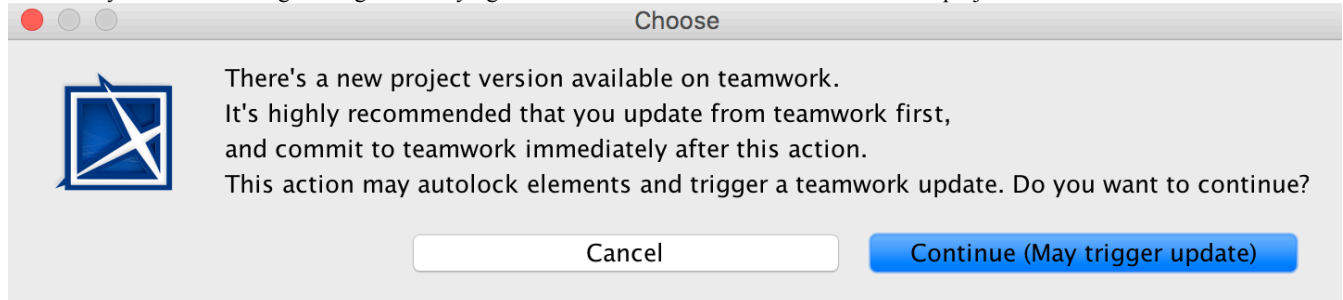
1. When message pops up about updating from teamwork, click "Cancel, I will update from teamwork first"
2. Go to "Collaborate" drop down menu and click "Update Project"
3. Upon completion, Generate again
4. This time, when message comes up, click "Continue with MMS"

Possible Connection Error: While working in a Teamwork project, a user may encounter the following problem:

- 2016.01.26::13:05:42] [ERROR] Getting changes from mms failed: Could not connect to broker URL: tcp://cae-ems.jpl.nasa.gov:61616. Reason: java.net.SocketTimeoutException: connect timed out
- Most likely this will be due to the fact that a separate user is also using one of the Update and Commit functions and so the server is busy
- A user should try to wait several minutes for the server to be freed and then retry the chosen Update and Commit function

New Project Version Available message:

- A user may see the following message when trying to interact with MMS on a Teamwork hosted project



- A user will see this when the Teamwork hosted project has been updated. It is highly recommended that the user should update from the Teamwork server and *then* interact with the MMS. The user should then commit these changes to the server.
- Although a user can choose to "Continue (May trigger update)", there may be some model inconsistency or data loss if the user does not continually interact with Teamwork
- To fix this, a user should login to Teamwork and "Update Project"

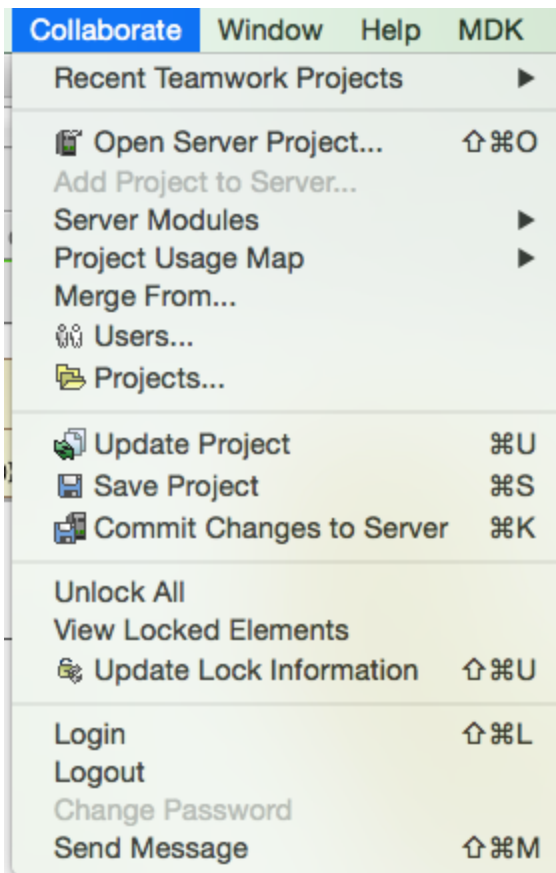
Handling Teamwork Crashes

If MagicDraw crashes during your work on a Teamwork project, there is a chance that MMS will be out of sync with Teamwork. To re-sync MMS with Teamwork, follow these three steps:

1. Re-open project from Teamwork
2. Run "Validate Models" from desired package(s)
3. Accept all changes from MMS

4.2 Collaborate Menu Overview

Below is a general overview of the actions that each option on the Collaborate menu offers:



1. Recent Teamwork Project - quick navigation to projects that you have recently opened

2. **Open Server Project** - allows the following actions: [Open a Teamwork Project](#) , [Create Teamwork Branch](#) , [Open Teamwork Branch](#)
3. **Add Project to Server** - this is an administrator only feature and is not currently discussed in the general users guide. For more information, refer to [Create a Teamwork Project](#)
4. **Server Modules** - "Modules are created with a purpose to reuse them or to decompose projects into parts." ~ MagicDraw User Manual. Refer to the User Manual from your local copy located at: {installation folder}/manual/MagicDraw User Guide
5. **Project Usage Map** - "The Project Usage Map is a live visual graph that represents Teamwork Server project usages as well as identifies potential problem areas" ~ MagicDraw User Manual. Refer to the User Manual from your local copy located at: {installation folder}/manual/MagicDraw User Guide
6. **Merge From...** - "Model Merge enables copying changes between different project versions." ~ [NoMagic](#). Refer to the User Manual from your local copy located at: {installation folder}/manual/MagicDraw User Guide
7. **Users...** - a way to view all the users on the teamwork server and their project permissions. With admin privileges, a user can edit these properties and permissions.
8. **Projects...** - a way to view all the projects on the teamwork server and their properties, including their permitted users. With admin privileges, a user can edit these properties and permissions.
9. ----- See [Everyday Interaction](#) for more information on the below options -----
10. **Update Project** - while working on a teamwork project, a user can (and should do so often) update the project from the latest commit on the server
11. **Save Project** - a user can save the project locally as well as some other actions. See [Saving and committing](#) in the [CAE MagicDraw FAQ](#) .
12. **Commit Changes to Server** - used to commit changes to the teamwork server

13. **Unlock All** - allows users to unlock all elements on server so that other users may start editing and using said elements. Requires the user to commit to server so that all changes are saved before unlocking can occur.
14. **View Locked Elements** - shows the "Locked View" tree that displays which elements are locked by which users at the time of the last update project or Update Lock Information
15. **Update Lock Information** - a way to get the latest list of which users have which elements locked

16. **Login** - login to a specified teamwork server using JPL username and password
17. **Logout** - logout from current teamwork server
18. **Change Password** - *DISABLED* - a user cannot change solely a teamwork password since it is associated with a user's LDAP account

19. **Send Message** - send messages to any user that is currently signed into teamwork

4.3 Create Teamwork Branch

In this screencast, a user (with Project Admin permissions) creates a brand new Teamwork Branch.

Background: A teamwork branch is similar to a Git branch; the goal is to provide an environment for a user to work with real model data and elements but not affect the *true* ("trunk", "master") model. This allows for experimentation and another form of collaboration besides locking elements on a real model. When the user is finished with all wanted changes, a teamwork branch can be merged back into the trunk (master) version.

<src="https://jpltube.jpl.nasa.gov/NetworkFileStore/8644/media_720.mp4">Your browser does not support embedded videos.

Link to video: https://jpltube.jpl.nasa.gov/NetworkFileStore/8644/media_720.mp4

Video Annotations:

1. *Begin with MagicDraw open, but no project open
2. Login to teamwork server
3. Collaborate->Open Server Project
4. Click "..." next to "Trunk" (*master branch) of the project you want to start with
5. Select the trunk branch (*the top one)
6. Click "Edit Branches"
7. Select a point in time to create the branch from
 1. *in video, select 2 commits ago
8. Click "Create Branch"
9. Name the branch
10. Do not toggle the "Branch used modules" check box
11. Click "OK"
12. (Optional) To Open Project on the new branch from this view, expand the branch selection, select the "0" version, click "OK"
13. Select the newly created branch
14. Click OK
15. *Notice that the "Open Server Project" window now shows the branch's name instead of "Trunk"
16. Click "Open"
17. *You can now work within your own branch of the project

*Information statement

4.4 Open Teamwork Branch

A teamwork branch is similar to a Git branch; the goal is to provide an environment for a user to work with real model data and elements but not affect the *true* ("trunk", "master") model. This allows for experimentation another form of collaboration besides locking elements on a real model. When the user is finished with all wanted changes, a teamwork branch can be merged back into the trunk (master) version.

In this screencast, the user demonstrates how to open a teamwork branch so that user may work on a project separately from the "trunk" version.

Link to Video: https://jpltube.jpl.nasa.gov/NetworkFileStore/9734/media_720.mp4

Video Annotations:

1. Login to teamwork server
2. Collaborate->Open Server Project
3. Click "..." next to "Trunk" (*master branch) of the project you want to start with
4. Select wanted branch
5. Click OK
6. *Notice window now shows the selected branch's name instead of "Trunk"
7. Click "Open"

4.5 Syncing with VE Task

In order for a user to work in a fully separate environment, MagicDraw and View Editor, a user needs to work in a teamwork branch that has an associated View Editor task. The association is made when the two have the same exact name. The following are the steps to set up and confirm that a separate environment has been established.

1. [Create Teamwork Branch](#) or [Open Teamwork Branch](#) with the **same name** as View Editor Task
2. [Create/Delete Tasks and Tags](#) or [Navigate and View Options](#) with the **same name** as Teamwork Branch

3. *You can now edit solely in your branch. To do a quick test see video below:
- 4.
5. Link to Video: https://jpltube.jpl.nasa.gov/NetworkFileStore/8701/media_720.mp4
6. Video Annotations:
 1. Create a document in MagicDraw while in teamwork branch.
 1. Generate and Commit to MMS
 2. Look on corresponding VE task and see if document is there
 3. Confirm it is not in another task, especially "master"
 2. Create a document in VE in the same-named task
 1. In MagicDraw, run an update from mms
 2. Notice the new document
 3. Switch to other teamwork branch (could be "trunk"), run an update from MMS, and notice it's not there