

Deleting and executing Macros



You can click the **Delete** or **Run** button in the **Organize Macros** dialog to delete or execute a selected macro. You can also press the pre-designed mnemonic keys to delete or run a macro as described in Organizing Macros Dialog Mnemonic Keys below.

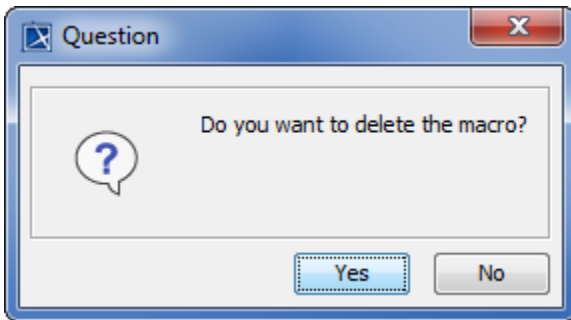
On this page

- [Deleting a Macro](#)
- [Executing a Macro](#)
- [Organizing Macros Dialog Mnemonic Keys](#)

Deleting a Macro

To delete a macro from the **Organize Macros** dialog

1. Click **Tools > Macros > Organize Macros** to open the **Organize Macros** dialog.
2. Select a macro from the table and either click **Delete** or press **Alt + D**. A dialog will open, asking whether you want to delete the macro, see figure below.

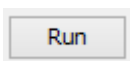
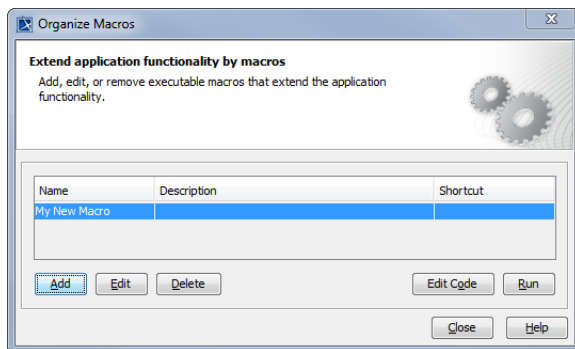


3. Click **Yes** and the macro will be deleted from the **Organize Macros** dialog.

Executing a Macro



To execute a macro from the **Organize Macros** dialog

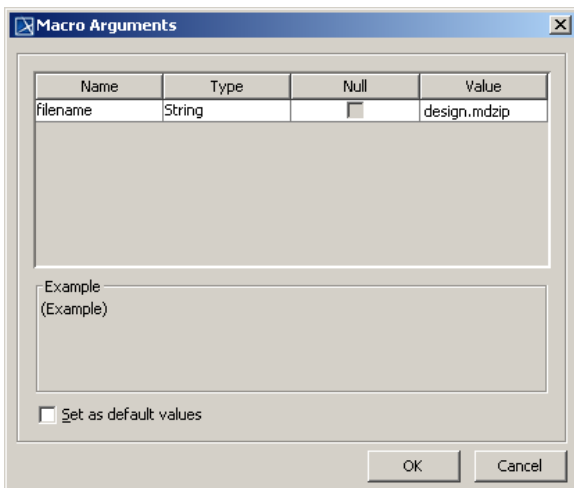
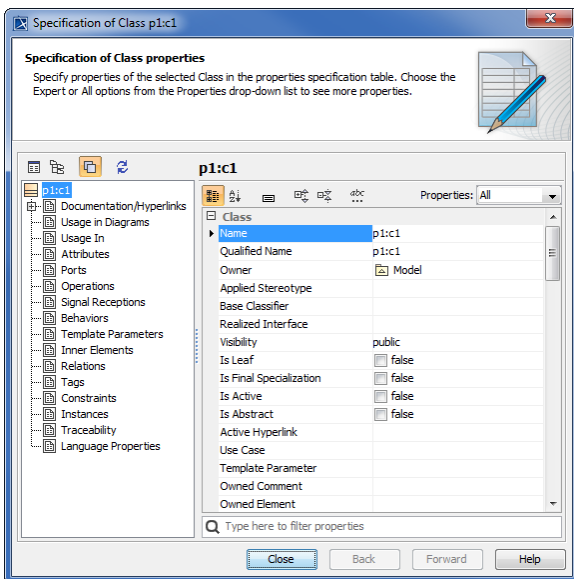
1. Click **Tools > Macros > Organize Macros** The **Organize Macros** dialog will open, see figure below.



2. Select a macro from the table and click **Run**. After the macro has been executed, a message will open: **The macro <macro name> has been executed.**
3. If you have the parameters in the **Macro Information** dialog, you need to specify the value in the **Macro Arguments** dialog, see figure below, before running the macro.



- For an array datatype, you need to click the  button in the **Value** column in the **Macro Arguments** dialog and enter each value into each line. The value of the first line will be the value in array index 0.
- An ElementPath is a Qualified Name. You can find this information in the specification dialog of each element, see figure below.
- The **Macro Arguments** dialog, see the figure below, will be displayed if the  **Automatically run with default values** check box is not selected.
- If you want to save argument values in the **Macro Arguments** dialog, you need to select the **Set as default values** check box before you click **OK**.

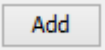
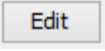
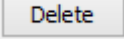
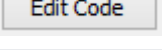
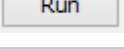
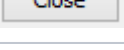
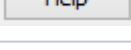


- You can also execute a macro from the main browser in MagicDraw by pressing the shortcut keys that you have defined in the **Organize Macros** dialog.
- You can only run macro one at a time.
- If there is an error while running a macro, for example, syntax error, the following message will open: **MagicDraw cannot execute the <macro language> macro, please make sure that <path, filename, extension> is correct. <error description>.**
- If MagicDraw cannot find a macro file in the location that you have specified in the **Open** dialog, the following message will open: **MagicDraw cannot find the macro: <path, filename, extension>.**

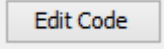
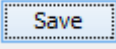
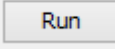
Organizing Macros Dialog Mnemonic Keys

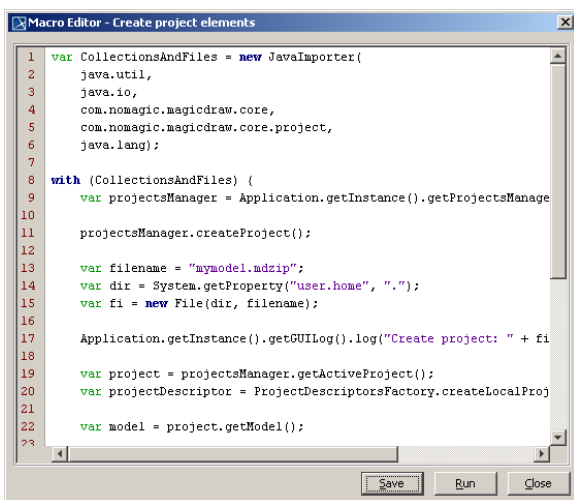
Macro Engine also provides mnemonic keys to add, edit, delete, and run a macro from the **Organize Macros** dialog. The table below shows the **Organize Macros** dialog mnemonic keys and their function.

Table listing of Organize Macros Dialog Mnemonic Keys

Mnemonic keys	Button	Function
Alt + A		To add a macro in the Macro Information dialog.
Alt + E		To edit a macro in the Macro Information dialog.
Alt + D		To delete a macro from the Organize Macros dialog.
Alt + O		Code To edit source code in Macro Editor .
Alt + R		To run a macro from the Organize Macros dialog.
Alt + C		To click the Close button.
Alt + H		To click the Help button.



- You can click the  button in the **Organize Macros** dialog, see figure above, to edit and save source code in the **Macro Editor** dialog, see figure below.
- You can click  to save the source code or click  to run the macro in the **Macro Editor** dialog, see figure below.



```
1 var CollectionsAndFiles = new JavaImporter(  
2     java.util,  
3     java.io,  
4     com.nomagic.magicdraw.core,  
5     com.nomagic.magicdraw.core.project,  
6     java.lang);  
7  
8 with (CollectionsAndFiles) {  
9     var projectsManager = Application.getInstance().getProjectsManage  
10  
11     projectsManager.createProject();  
12  
13     var filename = "mymodel.mzip";  
14     var dir = System.getProperty("user.home", ".");  
15     var fi = new File(dir, filename);  
16  
17     Application.getInstance().getGUILog().log("Create project: " + fi  
18  
19     var project = projectsManager.getActiveProject();  
20     var projectDescriptor = ProjectDescriptorsFactory.createLocalProj  
21  
22     var model = project.getModel();  
23
```