

## Plugins

A plugin is a piece of software that adds features to an existing program.

The grandMA3 supports such plugins, thus allowing customization.

For the plugins used in the grandMA3 software, the scripting language Lua is used.



### Important:

The creation and use of plugins can go deeper into the system as the “normal” usage of a console. Therefore the technical support team of MA Lighting may not be able to help you in all circumstances when using complex Lua plugins and Lua plugins might have to be rewritten when migrating show files to future grandMA3 software version.

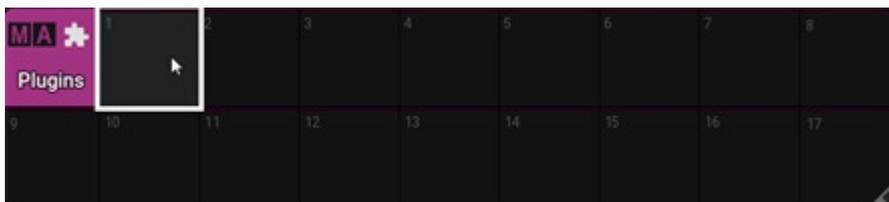
## Add a Plugin Pool Window

- To add a plugin pool window, follow the instructions under [Add windows](#).

In the Add Window pop-up:

1. Tap **Pools**.
2. Tap **Plugins**.

The plugin pool opens.



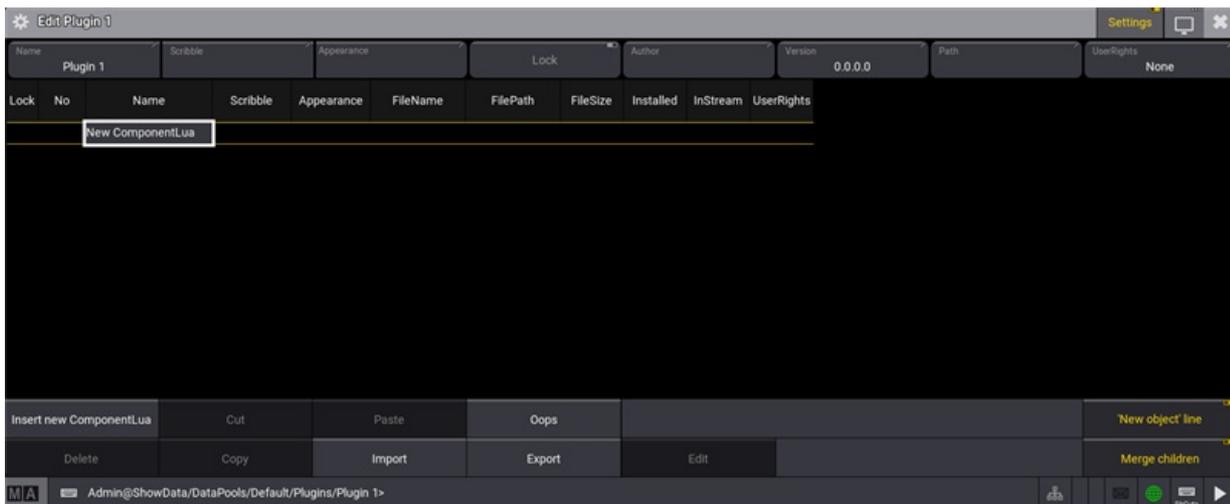
*Plugin pool window*

- To adjust the plugin pool window settings, follow the instructions under [Window settings](#).

## Add a Plugin to the Pool

- To add a plugin to the plugin pool, tap an empty pool object and use the **swipey menu** to select **Edit**.

The Edit Plugin window opens:

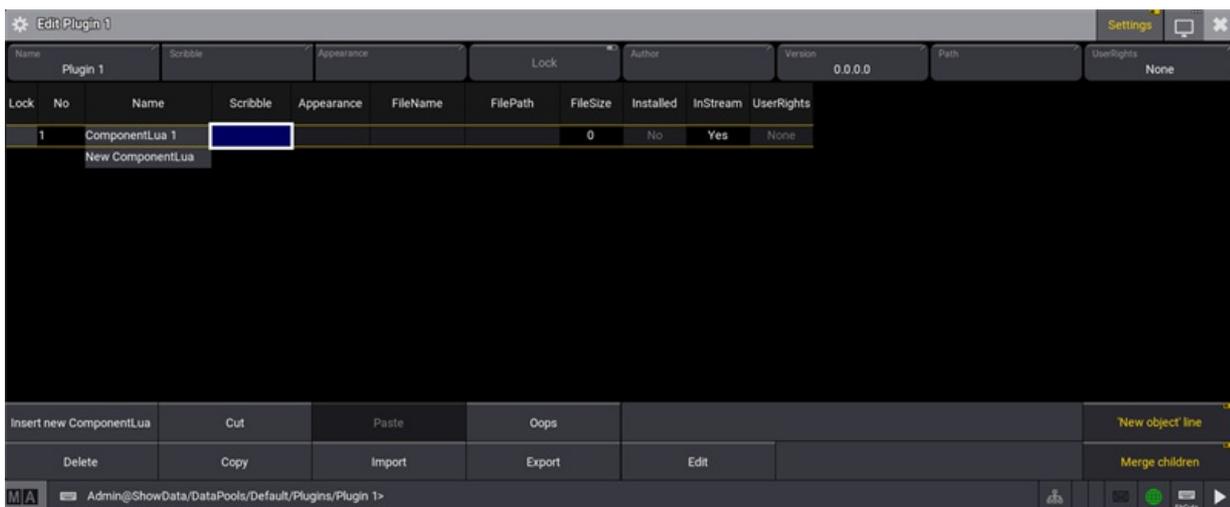


*Edit Plugin window*

## Edit a Plugin

A Lua component is a piece of software code that can be inserted into the console, usually one file.

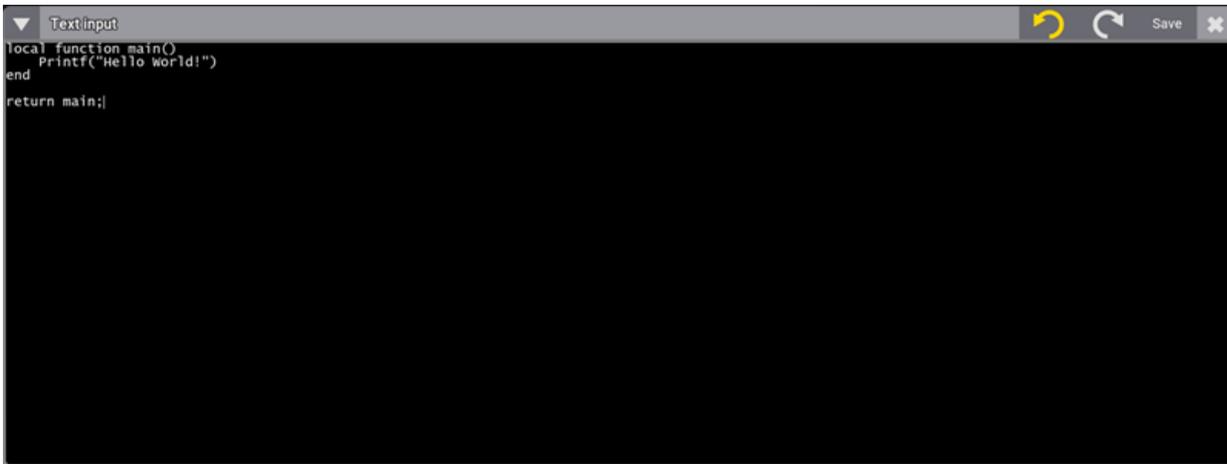
- To add a new Lua component, tap **Insert new ComponentLua**.



The new component is added.

- To add Lua code to the component, tap **Edit**.

The Text input window opens:

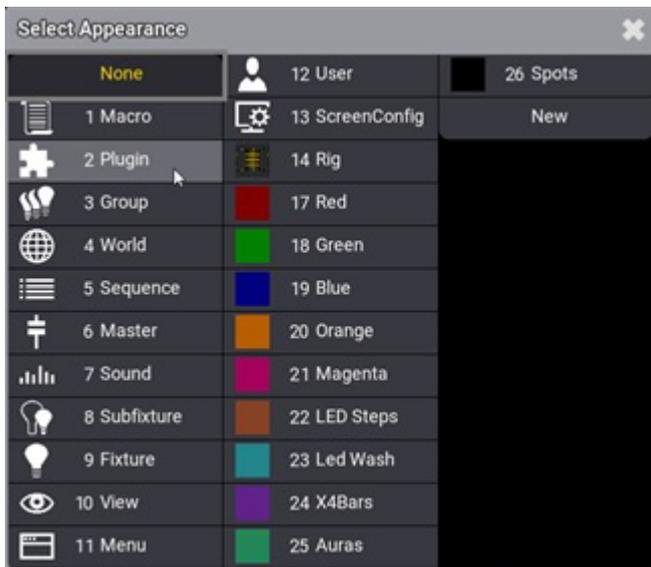


*Lua Text input window*

- Add text or paste the Lua code into the Text input window.
- To save the Lua plugin, tap **Save**.
  
- To add a Scribble, tap **Scribble**, and select the desired scribble from the drop-down list.



- To add an Appearance, tap **Appearance**, and select the desired appearance from the drop-down list.



This file is saved in the show file, unless you export it.

- To enable or disable Installed, tap **Installed** to toggle between Yes and No.  
If Installed is set to Yes, the Lua components will be updated from the file archive they were imported from.  
To update any changes in these Lua components, the **command ReloadPlugins** must be executed.
- **In Stream** describes if the Lua code will be saved in the show file and be streamed in the session or not.  
**In Stream** depends on the setting in **Installed** and cannot be modified by the user.  
**In Stream** Yes means that the Lua code is saved in the show file and streamed in the session but stays as saved in the show file.  
**In Stream** No means that the Lua code is locally saved on the hard drive. The content of this Lua code can be updated by the **command ReloadPlugins**.
- To adjust the user rights, tap **User Rights**, and select the desired user right needed to execute the plugin from the drop-down list.



- To close the Edit Plugin window, tap .

## Run a Plugin

- To run a Lua plugin, tap the desired Plugin object.



## Import a Plugin

- To import a plugin, first create the plugin with your editor.

## Example Plugin

### XML file my\_plugin.xml:

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<GMA3><Plugin Path="/my_plugin" Name="My Plugin"><ComponentLua Name="my_plugin" Installed="Yes" FileName="my_pli
```

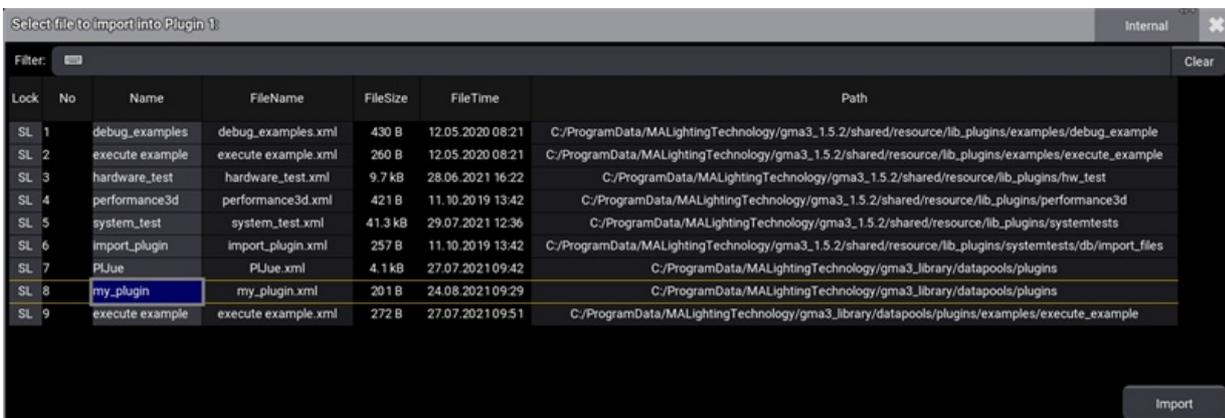
### Lua file my\_plugin.lua:

```
local function main()  
    Printf("Hello, MA user!")  
end
```

```
return main
```

- Save the Lua file and the XML file in the folder  
C:\ProgramData\MALightingTechnology\gma3\_library\datapools\plugins.
- Tap **Import**.

The Import selection menu opens:



- Select the plugin you want to import and tap Import.

The plugin is imported.

