

# loadPresetAsync

```
loadPresetAsync(path, callback)
```

## Description

Function to load the elements of a VST preset in a separate, parallel thread. Loading the VST preset in a separate thread can be necessary if the preset is too big to be loaded in a short time. The function returns a [LoadProgress](#) object that can be used to get information on the load progress and the loaded elements, for example. After the preset is loaded, the callback function is called. The callback function gets the [LoadProgress](#) object as default argument.

**Available in:** Controller.

## Arguments

<b>path</b>	The path and file name of the VST preset.	string
<b>callback</b>	Callback function that is called after the preset is loaded. The callback function gets the <a href="#">LoadProgress</a> object as argument.	function, optional

## Return Values

Returns a [LoadProgress](#) object.

## Example

```

-- start with an empty program, remove all existing layers
layers = this.parent:findLayers()

if layers then
  for i, layer in ipairs(layers) do
    this.parent:removeLayer(layer)
  end
end

-- table with layer presets from Skylab
layerPresets = {
  { name = "Ambient Pad 01", path = "vstsound://724ACB205EFF46F885735D1B216C37AD/.AppData/Steinberg/Skylab/Sub
Presets/Layer Presets/Ambient Pads/Ambient Pad 01.vstpreset" },
  { name = "Ambient Pad 02", path = "vstsound://724ACB205EFF46F885735D1B216C37AD/.AppData/Steinberg/Skylab/Sub
Presets/Layer Presets/Ambient Pads/Ambient Pad 02.vstpreset" },
  { name = "Ambient Pad 03", path = "vstsound://724ACB205EFF46F885735D1B216C37AD/.AppData/Steinberg/Skylab/Sub
Presets/Layer Presets/Ambient Pads/Ambient Pad 03.vstpreset" },
  { name = "Ambient Pad 04", path = "vstsound://724ACB205EFF46F885735D1B216C37AD/.AppData/Steinberg/Skylab/Sub
Presets/Layer Presets/Ambient Pads/Ambient Pad 04.vstpreset" },
}
-- create table with the preset names
function getPresetNames()
  presetNames = {}
  for i, preset in ipairs(layerPresets) do
    presetNames[i] = preset.name
  end
end

getPresetNames()

-- remove the old layer after the new one was added
function removeOldLayer(progressInfo)
  local newPreset = progressInfo.root
  if oldPreset then
    this.parent:removeLayer(oldPreset)
    print(oldPreset.name.." removed.")
  end
  oldPreset = newPreset
end

-- append the preset in a separate thread
function appendNewLayer(progressInfo)
  if progressInfo.root then
    this.parent:appendLayerAsync(progressInfo.root, removeOldLayer)
    print("Appending "..progressInfo.root.name.."...")
  end
end

-- load the preset in a separate thread
function onSelectPresetChanged()
  progress = 0
  progressInf = loadPresetAsync(layerPresets[SelectPreset].path, appendNewLayer)
  print("Loading "..layerPresets[SelectPreset].name.."...")
end

-- define a parameter for selecting the preset to be loaded
defineParameter("SelectPreset", "Select Preset", 1, presetNames, onSelectPresetChanged)

-- monitor the progress with onIdle
progress = 1
function onIdle()
  if progress < 1 then
    progress = progressInf.progress
    print("Progress: "..(progressInf.progress * 100).."%")
  end
end
end

```

**See Also:** [loadPreset](#), [LoadProgress](#)