


appendLayerAsync

```
appendLayerAsync(layer, callback)
```

Description

Function to add a layer in the specified destination layer using a separate, parallel thread. Appending a layer in a separate thread can be necessary if the layer is too big to be added in a short time. The layer to be inserted and the destination layer are both determined by their [Layer](#) objects. You can use [getLayer](#) or [findLayers](#) to determine the layer to be inserted. For example, `this.parent` determines the parent layer of the script module as destination layer. The new layer will be added behind the existing layers. To insert a layer at a specific position in the destination layer, use [insertLayer](#) or [insertLayerAsync](#) instead. The function returns a [LoadProgress](#) object that can be used to monitor the load progress. After the layer is added, the callback function is called. The callback function gets the [LoadProgress](#) object as default argument.

 An [Element](#) object can only have one parent. It cannot be child of multiple parents. Therefore, an element object that you retrieved from the running plug-in instance must be removed before it can be inserted again. The element objects you retrieve through [loadPreset](#) or [loadPresetAsync](#) can be inserted freely, because these functions create a copy of the [Element](#) objects when reading them.

Available in: Controller.

Arguments

layer	The Layer object of the layer that you want to append.	Layer
callback	Callback function that is called when the layer is added. The callback function gets the LoadProgress 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: [appendBus](#), [appendEffect](#), [appendLayer](#), [appendMidiModule](#), [appendZone](#), [LoadProgress](#)