

onSaveSubPreset

onSaveSubPreset(section)

Description

This callback function is called when saving a subpreset with a corresponding [Preset Browser](#) template. The callback will only be called if the scope of the [Preset Browser](#) template is set correctly.

- If the MacroPage with the [Preset Browser](#) template is attached to an element other than the script module (e.g., the program), the scope must be set to the script module (e.g., @0:Script Module).
- If the MacroPage with the [Preset Browser](#) template is attached to the script module itself, the scope does not need to be set.

In addition, the [Preset Browser Custom](#) template allows to define a section for the subpreset. When you save a subpreset, the section will be passed on from the [Preset Browser Custom](#) template to the callback. You can manage different subsets of parameters by using the section as condition for an `if` statement that stores only the parameters of interest. The data you pass on to the `return` statement will be stored with the subpreset. The data can be of any type, but it is common practice to use a table that can easily be extended with more fields. When the subpreset is restored, the [onLoadSubPreset](#) callback will receive the stored data.

Scope and section are template parameters. You can set them in the MacroPage Designer on the properties pane of the [Preset Browser](#) and [Preset Browser Custom](#) templates.

Available in: Controller.

Arguments

section	The section as defined in the Preset Browser Custom template.	string
----------------	---	--------

Return Values

The returned data will be stored in a subpreset.

Example

```

--[[
    The corresponding MacroPage needs two preset templates: One with
    the section set to Pitch and the other with the section set to Filter.

    The preset template with the section Pitch stores/restores the
    parameters Pitch.Coarse and Pitch.Fine.

    The preset template with the section Filter stores/restores the
    parameters Filter.Cutoff and Filter.Resonance.
--]]

-- get the values of p1 and p2 from the first zone of the program
function getZoneData(p1, p2)
    local data = {}
    local zone = this.program:findZones(true)
    if zone[1] then
        data.p1 = zone[1]:getParameter(p1)
        data.p2 = zone[1]:getParameter(p2)
    end
    return data
end

-- set the values of p1 and p2 in the first zone of the program
function setZoneData(p1, p2, data)
    local zone = this.program:findZones(true)
    if zone[1] then
        zone[1]:setParameter(p1, data.p1)
        zone[1]:setParameter(p2, data.p2)
    end
end

-- save data with the subpreset
function onSaveSubPreset(section)
    if section == "Pitch" then
        return getZoneData("Pitch.Coarse", "Pitch.Fine" ) -- called from preset template Pitch
    elseif section == "Filter" then
        return getZoneData("Filter.Cutoff", "Filter.Resonance") -- called from preset template Filter
    end
end

-- restore data from the subpreset
function onLoadSubPreset(section, data)
    if section == "Pitch" then
        setZoneData("Pitch.Coarse", "Pitch.Fine", data) -- called from preset template Pitch
    elseif section == "Filter" then
        setZoneData("Filter.Cutoff", "Filter.Resonance", data) -- called from preset template Filter
    end
end

```

See Also: [onLoadSubPreset](#)