# How to Resize GUI using Allegro SKILL

Product Version: SPB 22.1
January 2023

Copyright Statement

# Contents

## Purpose

This Application Note describes how to resize graphical user interfaces (GUI) and the individual controls in Allegro® SKILL using examples. The attached ZIP archive contains SKILL file and different form files with the examples described in this document.

## Audience

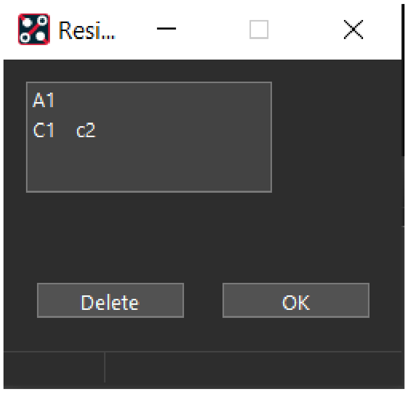
This document is intended for Allegro® SKILL developers.

## SKILL Files

SKILL files can be found at 'Attachments' section. This pdf can be searched with the document title on https://support.cadence.com

## Making Forms Resizable

Although graphical user interfaces (GUI) created in Allegro® SKILL are fixed in size by default, you can make them resizable.



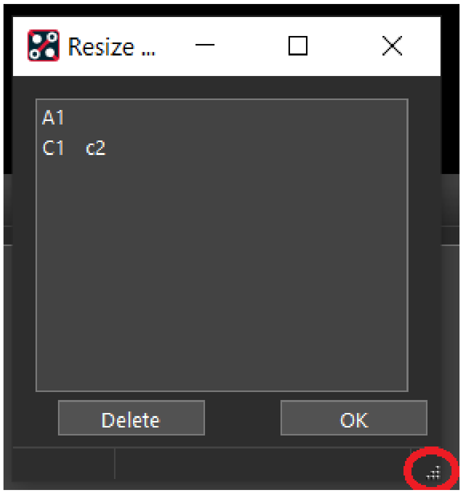
To make a form resizable, add `FLEXMODE`, optionally with `FLEX`, to the form file before `ENDTILE`, `ENDFORM`.

```
[FLEXMODE ]

[FLEX fx fy fw fh]
```

FLEXMODE makes the form resizable, and a few dots appear on the lower-right corner of the form, as shown in the following figure. The cursor changes to a double-sided arrow, on pointing to these dots.

FLEX can then be used to manage the sizing and movement of individual form controls.



So, FLEXMODE represents the general rules that apply to all controls in the form except those with specific overrides (FLEX).

The attached zip archive contains a skill file (test_resize.il) and various form files (noresize.form, resize.form, resize_ex1.form and so on) for the different examples in this document. Call the corresponding function (noresize, ex, to see the various forms)

In the resize.form file, choose the appropriate flex mode by uncommenting FLEXMODE statements; remove the pound symbol (#) before the FLEXMODE statements. In the following example, *StandButtons* FLEXMODE is active.

```
#FLEXMODE EdgeGravity
#FLEXMODE EdgeGravityOne
FLEXMODE StandButtons
```

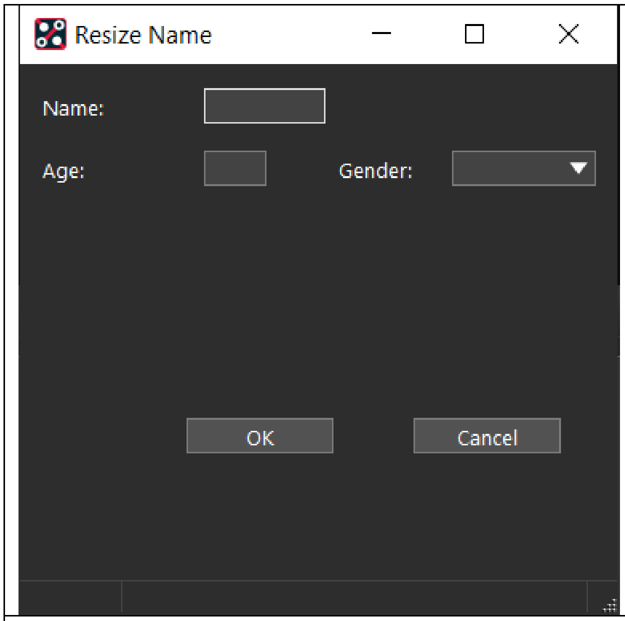
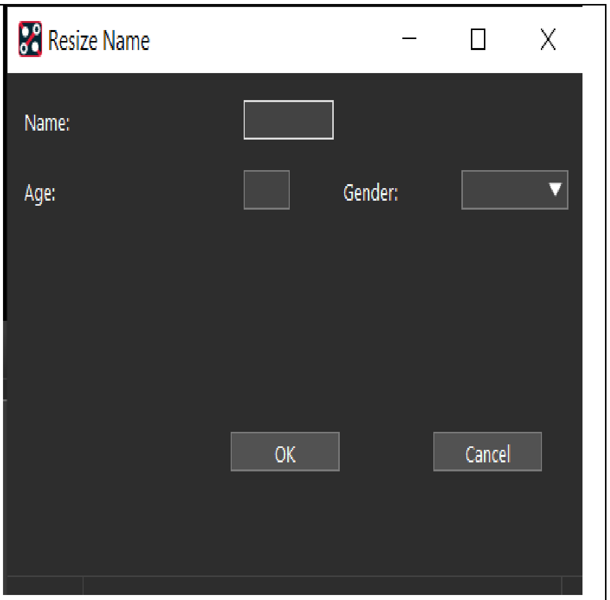
## Resizing Entire GUI

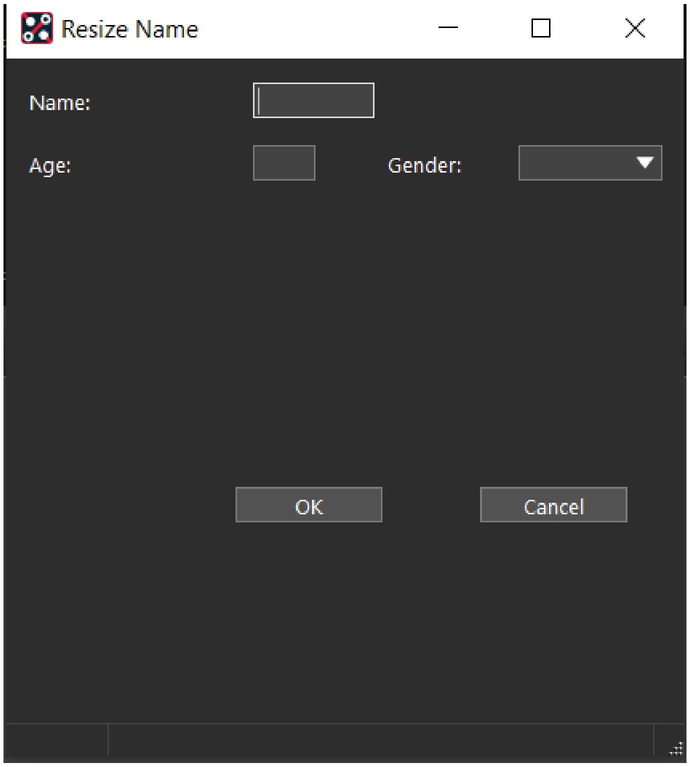
FLEXMODE represents general rules that apply to all the controls in a form. But you can override these rules for specific controls using FLEX.

Only a single `FLEXMODE` is supported per form, and, in case of multiple `FLEXMODE` statements, the last encountered in the form file is used.

The following rules are supported:

- `EdgeGravity`: Controls have an affinity to the closest edge of their immediate container. Exceptions are the *FILLIN* and *INTSLIDEBAR* controls -, the edge gravity for these controls is based on a *TEXT* control positioned to the left.
- `EdgeGravityOne`: Like *EdgeGravity* except that controls are only locked to the right or bottom edge, but not both. The closest edge is used.
- `StandButtons`: Only effects button controls. Uses the same logic as *EdgeGravityOne*.

| | |
|---|---|
|  |  |
| **GUI With FLEXMODE EdgeGravity** | **GUI With FLEXMODE EdgeGravityOne** controls are only locked to the right or bottom edge, but not both. The closest edge is used |



GUI with FLEXMODE StandButtons

## Managing Sizing and Movement of Individual Controls

Use `FLEX` to manage the sizing and movement of individual controls. It has the following syntax:

```
FLEX fx fy fw fh
```

`FLEX` overrides any `FLEXMODE` in effect for that control, and is based upon the parameters `fx`, `fy`, `fw`, and `fh`. The parameters, which are floating point numbers between 0.0 and 1.0, specifies the fraction of the change in container size that the control should move or change in size.

| Parameters | Values | |
|---|---|---|
| | 0 | 1 |
| `fx` and `fy` | Control remains locked to the left or top edge of its container. | Control remains locked to the right or bottom edge of its container. |
| `fw` and `fh` | Control is not resized. | Control is resized in width or height based upon the size change of its container. |

Containers are hierarchical and the size and position of a container affect the member controls. Make sure the container of the control also has a *FLEX* constraint. The sum of the width and height of the immediate controls of a container should not be greater than 1 to prevent overlapping.

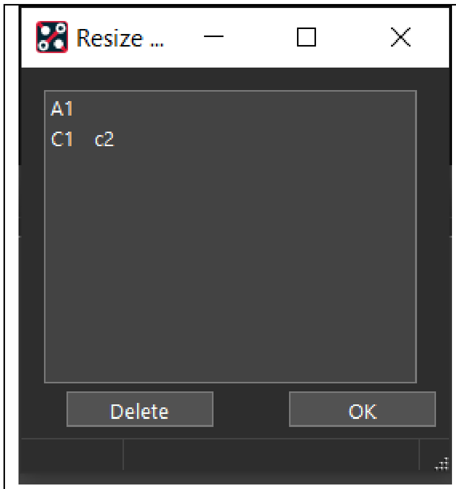
*TABSETS* are slightly different because sizing of their member controls is also based on the *TAB* they belong to.

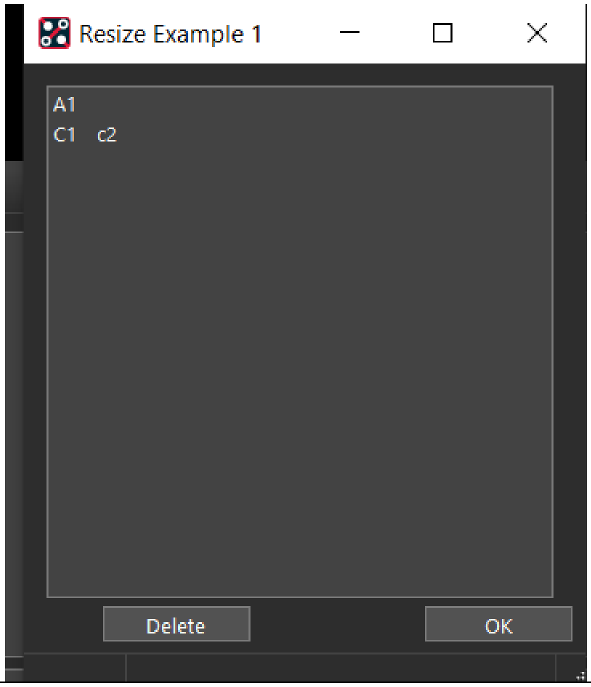
**Caution:** It is possible to create FLEX constraints that result in overlapping controls.

**Example 1**

```
FLEXMODE StandButtons
FLEX list 0 0 1 1
```

Simple list-based form with buttons (label of LIST is list.) The list sizes along with the form, as shown in following image. That is, changes both height and width with the form.

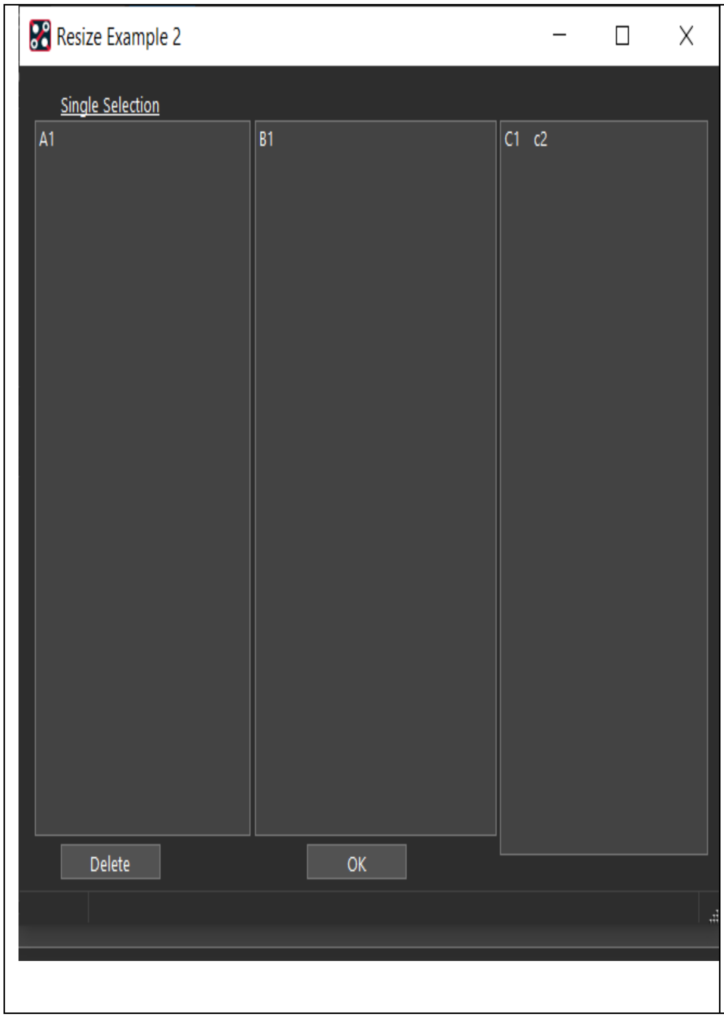


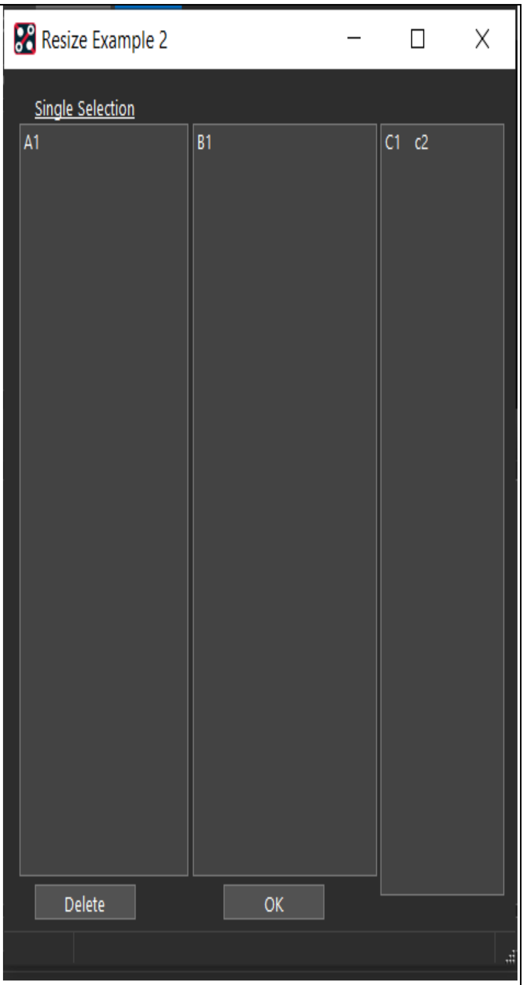


### Example 2

```
FLEXMODE EdgeGravity
FLEX a 0 0 0.33 1
FLEX b 0.33 0 0.67 1
FLEX c 0.67 0 1 1
```

Form containing three lists (a, b, and c) positioned equally across the form. Each list resizes to the height of the form but shares in the increase in form width. Thus, if the form changes width, each control gets 1/3 of this change. Since the list's width changes, the list must move to the right, as shown in following images.

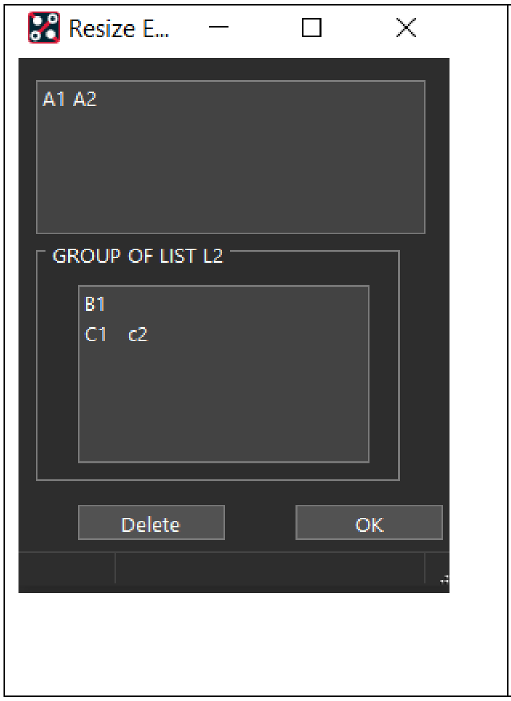


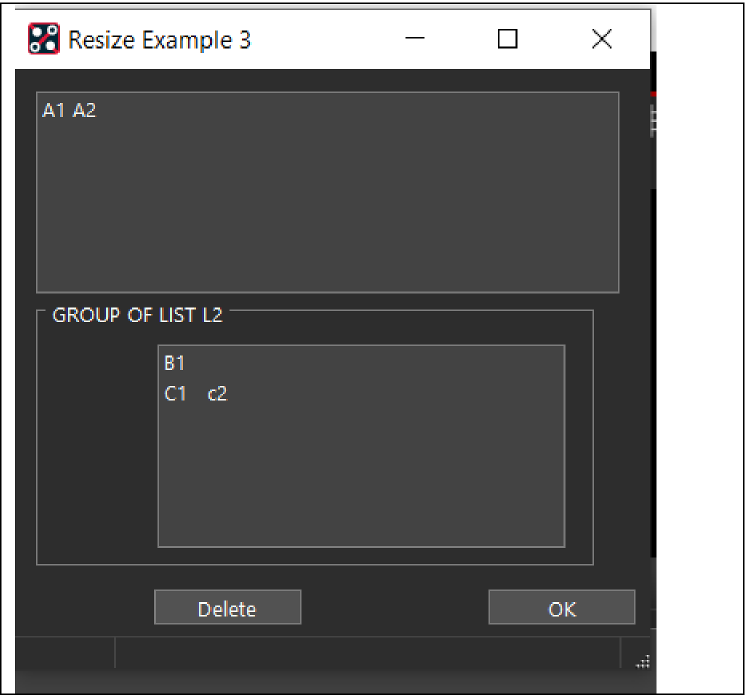


### Example 3

```
FLEX l1 0 0 1 0.5
FLEX g1 0 0.5 1 0.5
FLEX l2 0 0 1 1
```

Form has a group (*g1*) containing a list (*l2*). The group is below another list (*l1*). Both lists share in any change of the form size. The second list (*l2*) is a member of the group container (*g1*), so it moves if the group moves (0 for y) and it gets all the group resizing (h is 1) as shown in the following images.

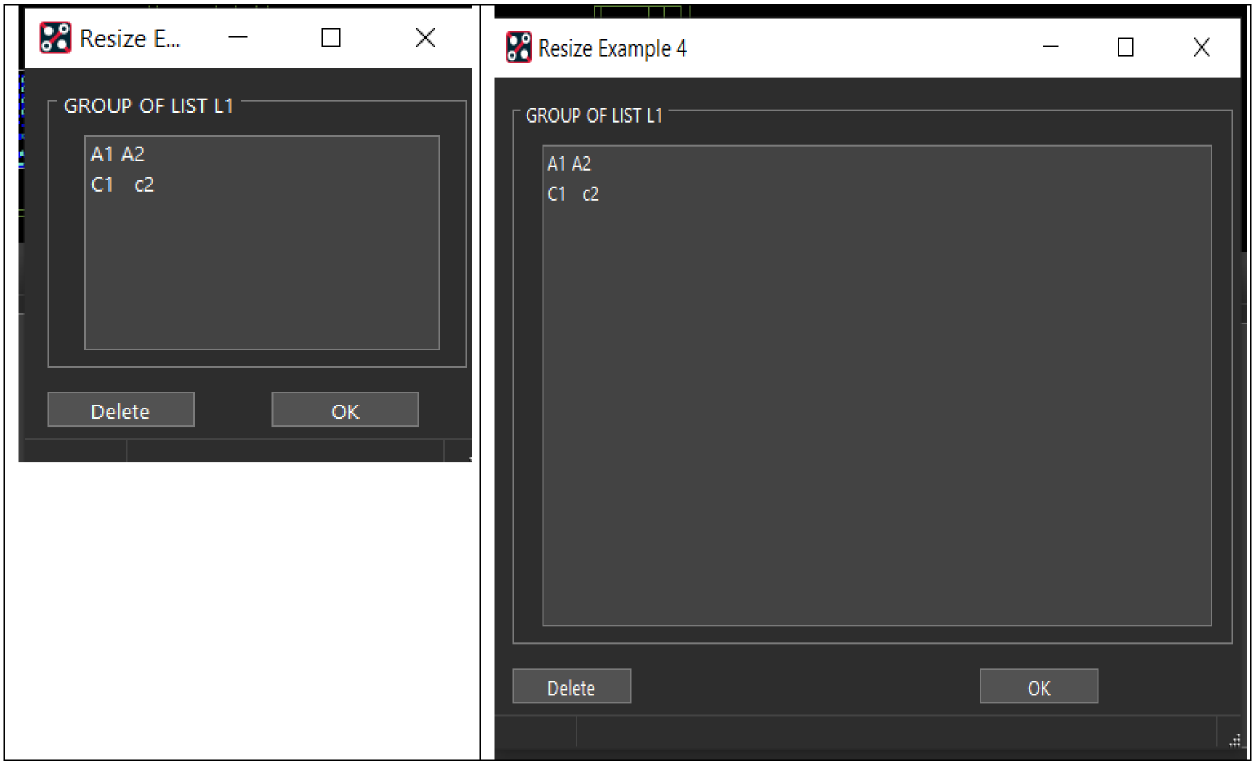




### Example 4

```
FLEX g1 1 1 0 0
FLEX l1 0 0 1 1
```

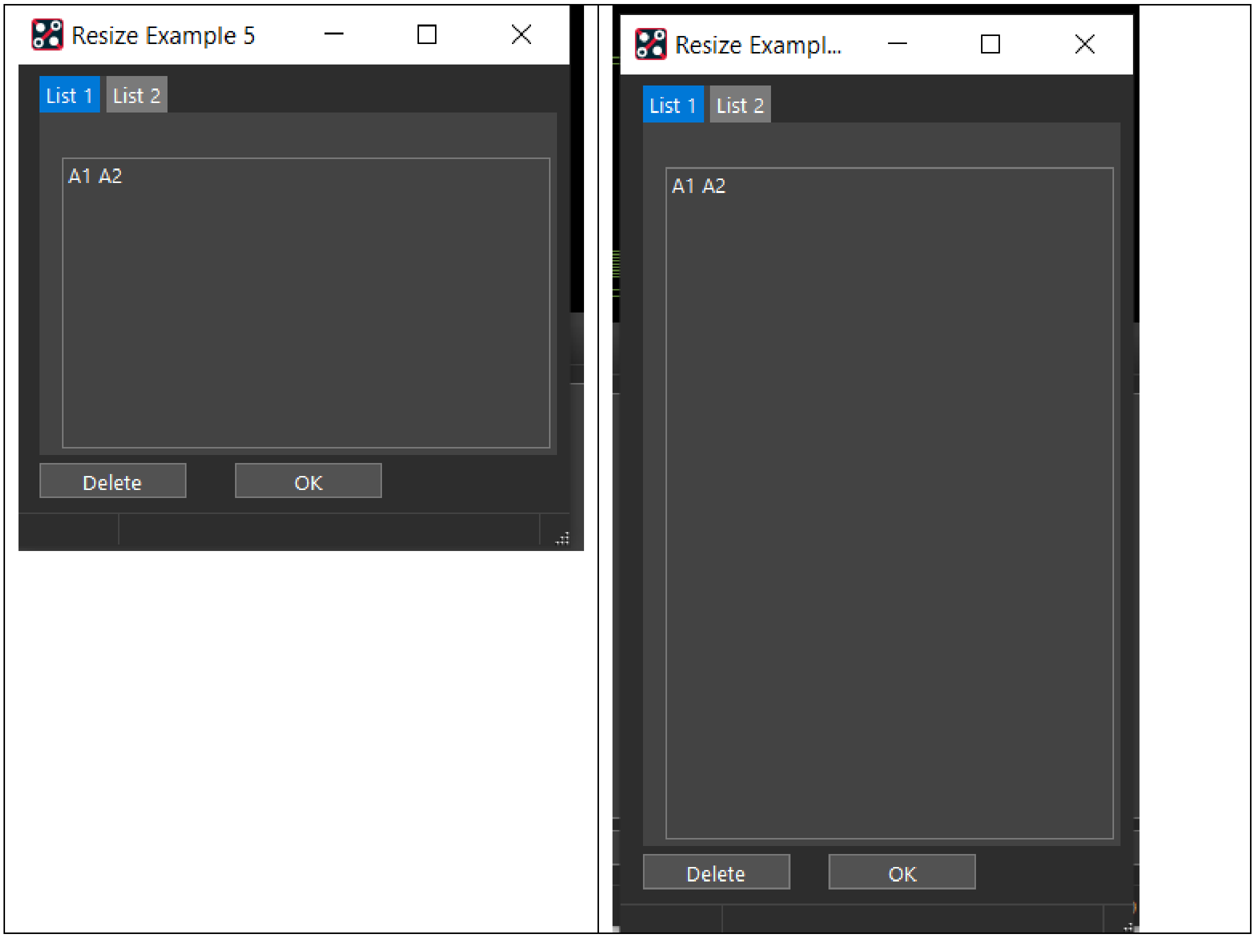
Form has a group (*g1*) with a list member (*l1*).But the list does not resize because it is a member of a group with 0:0 sizing. Though the list has 1:1 sizing, it never changes in size because its container never changes in size. Both the group and its member list move because the group has a 1:1 x/y factor as shown in the following images.



### Example 5

```
FLEX t1 0 0 1 1
FLEX l1 0 0 1 1
FLEX l2 0 0 1 1
```

Form is a TABSET (*t1*) with two tabs. Each tab controls a list (*l1* and *l2*, respectively) that accommodates the maximum change in the form size as shown in the following figures.



## Support

Cadence® Online Support provides access to support resources, including an extensive knowledge base, access to software updates for Cadence products, and the ability to interact with Cadence® Customer Support. Visit *https://support.cadence.com*.

**Note:** Cadence Customer support does not debug Skill code written by users or write Skill code for users.

## Feedback

Email comments, questions, and suggestions to content_feedback@cadence.com.