

HI_PopUpMenu_AddItem

%HI_PopUpMenu_AddItem function

Function

The function allows to add an item to the popup menu ([Control function](#)).

Declaration

```
%HI_PopUpMenu_AddItem(  
    INT in type,  
    TEXT in text,  
    INT in itemId,  
    INT in parentId  
    [, BOOL in bEnabled,  
    BOOL in bChecked,  
    HBJ in bitmapHBJ,  
    HBJ in bitmapCheckedHBJ]  
)
```

Parameters	<table><tr><td>type</td><td>Item type. Possible values:<ul style="list-style-type: none">• 0 - normal item• 1 - Nested popup menu• 2 - menu separator</td></tr><tr><td>text</td><td>Item text.</td></tr><tr><td>itemId</td><td>Unique identifier.</td></tr><tr><td>parentId</td><td>Unique identifier of the parent popup menu (popup menu one level above).</td></tr><tr><td>bEnabled</td><td>Enable/disable the item.</td></tr><tr><td>bChecked</td><td>Select/unselect the item.</td></tr><tr><td>bitmapHBJ</td><td>Object of Bitmap type.</td></tr><tr><td>bitmapCheckedHBJ</td><td>Object of Bitmap type.</td></tr></table>	type	Item type. Possible values: <ul style="list-style-type: none">• 0 - normal item• 1 - Nested popup menu• 2 - menu separator	text	Item text.	itemId	Unique identifier.	parentId	Unique identifier of the parent popup menu (popup menu one level above).	bEnabled	Enable/disable the item.	bChecked	Select/unselect the item.	bitmapHBJ	Object of Bitmap type.	bitmapCheckedHBJ	Object of Bitmap type.
type	Item type. Possible values: <ul style="list-style-type: none">• 0 - normal item• 1 - Nested popup menu• 2 - menu separator																
text	Item text.																
itemId	Unique identifier.																
parentId	Unique identifier of the parent popup menu (popup menu one level above).																
bEnabled	Enable/disable the item.																
bChecked	Select/unselect the item.																
bitmapHBJ	Object of Bitmap type.																
bitmapCheckedHBJ	Object of Bitmap type.																

Description

The function adds the item into the popup menu. It must be called (repeatedly if necessary) after calling the function [%HI_PopUpMenu_Create](#).

```
graph TD
    L1[Text 1] --- L2[Text 2.1]
    L1 --- L2_3[Text 2.3]
    L4[Text 4]
```

The parameter *type* defines the type of the item to be inserted. For each item (except Menu separator type), there must be defined an unique identifier (the parameter *id*) within defined menu (that compounds of all defined popup menus). If the item is inserted into a nested popup menu (e.g. the item "Text 1.1" in the figure above), the parameter *parentId* contains *id* of the item in the parent popup menu:

```
; definition of the item "Text 2"  
  
%HI_PopUpMenu_AddItem(1, "Text 2", 2, 0, @TRUE, @FALSE, 0, 0)  
  
; definition of the item "Text 2.1"  
  
%HI_PopUpMenu_AddItem(0, "Text 2.1", 21, 2, @TRUE, @FALSE, 0, 0)
```

The parameter *bEnabled* enables (the value @TRUE), or disables (the value @FALSE) the item. In the figure above, the item "Text 2.2.2".
The parameter *bChecked* selects (the value @TRUE) the item. In the figure above, the item "Text 2.2.1".
The parameters *bitmapHBJ* and *bitmapCheckedHBJ* are the reference to objects of [Bitmap](#) type, that will be used (if are defined) if the item is selected (checked), or is not selected (unchecked).

In script, items defined by the function **%HI_PopUpMenu_AddItem** may not be organized as they will be displayed in the popup menu - they will be automatically organized before showing the menu.

Regulations:

- maximum number of items in menu - 200
- if item has reference to non-existing parent popup menu, it will be shown in the main menu
- if parent popup menu is not submenu, the item will be shown in the main menu
- if parent popup menu is the same item (reference to itself), it will be shown in the main menu
- if there is several items with the same *id*, the first one of them will be parent popup menu
- order of items in main menu and submenus is given by their order in script
- circular references are removed from popup menu (including submenus) and there is displayed error report

Example

The script that creates and displays the popup menu shown in the figure above.

```
; create an empty popup menu
```

```
%HI_PopUpMenu_Create(300,300)
```

```
%HI_PopUpMenu_AddItem(0, "Text 1", 1, 0, @TRUE, @FALSE, 0, 0)  
%HI_PopUpMenu_AddItem(1, "Text 2", 2, 0, @TRUE, @FALSE, 0, Bitmapa.  
BMP\HBJ)
```

```
%HI_PopUpMenu_AddItem(0, "Text 2.1", 21, 2, @TRUE, @FALSE, 0, 0)
```

```
%HI_PopUpMenu_AddItem(0, "Text 2.1.1", 211, 21, @TRUE, @FALSE, 0, 0)
```

```
%HI_PopUpMenu_AddItem(0, "Text 2.1.2", 212, 21, @TRUE, @FALSE, 0, 0)
```

```
%HI_PopUpMenu_AddItem(1, "Text 2.2", 22, 2, @TRUE, @FALSE, 0, 0)
```

```
%HI_PopUpMenu_AddItem(0, "Text 2.2.1", 221, 22, @TRUE, @TRUE, 0, 0)  
%HI_PopUpMenu_AddItem(0, "Text 2.2.2", 222, 22, @FALSE, @FALSE, 0, 0,)
```

```
%HI_PopUpMenu_AddItem(0, "Text 2.3", 23, 2, @TRUE, @FALSE, 0, 0)
```

```
%HI_PopUpMenu_AddItem(2, "Text 3", 3, 0, @TRUE, @FALSE, 0, 0)  
%HI_PopUpMenu_AddItem(0, "Text 4", 4, 0, @TRUE, @FALSE, 0, 0)
```

```
; show popup menu  
%HI_PopUpMenu_Show( )
```

**Related pages:**

[Active picture manipulation functions](#)
[Indexed local variables](#)
[Function arguments - types](#)