

# HI\_PopUpMenu.AddItem

## %HI\_PopUpMenu.AddItem function

### Function

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

### Declaration

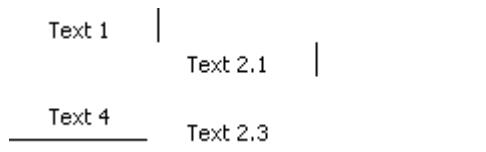
```
%HI_PopUpMenu.AddItem(
    INT in type,
    TEXT in text,
    INT in itemId,
    INT in parentId
    [, BOOL in bEnabled,
    BOOL in bChecked,
    HBJ in bitmapHBJ,
    HBJ in bitmapCheckedHBJ]
)
```

### Parameters

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



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](#)