

HI_GetFormatRE

%HI_GetFormatRE, %HI_SetFormatRE functions

Function The function **%HI_SetFormatRE** sets a font format to a selected text in the graphic object [Text entry field](#), type **RichEdit** ([control function](#)).

The function **%HI_GetFormatRE** retrieves a font format of a text in the graphic object [Text entry field](#), type **RichEdit**.

Declaration

```
%HI_SetFormatRE(
    INT in refId,
    TEXT in xmlFormat
)

TEXT %HI_GetFormatRE(
    INT in refId
)
```

Parameters

refId	Reference to the Text entry field , type RichEdit (reference variable).
xmlFormat	A font format of the selected text in the Text entry field , type RichEdit .

Note The font format is always written as XML file.
For example:

```
<?xml version="1.0" encoding="utf-8" ?>
<ROOT>
  <face>Times New Roman</face>
  <color>255</color>
  <size>48</size>
  <charset>0</charset>
  <bold>FALSE</bold>
  <italic>FALSE</italic>
  <underline>TRUE</underline>
  <strikeout>FALSE</strikeout>
  <protected>FALSE</protected>
</ROOT>
```

XML format represents step by step a font type, font color, font size and charset. These parameters are followed by the flags - bold, italic, underline, strikeout or protected (not important so far).

Only entered parameters are formatted. If some parameter is absent, it will not be changed. If a parameter is entered for formatting, it have to contain a valid value and can be entered just once. Otherwise, XML format is invalid.

Example

```

INT _nodeDoc
INT _nodeRoot
INT _nodeElement
INT _nodeText

TEXT _xmlFilter
BOOL _ok

_nodeDoc := %XML_CreateDocument()
_nodeRoot := %XML_AddElement(_nodeDoc, "ROOT")

_nodeElement := %XML_AddElement(_nodeRoot, "face")
_nodeText := %XML_AddTextNode(_nodeElement, "Times New Roman")
_nodeElement := %XML_AddElement(_nodeRoot, "color")
_nodeText := %XML_AddTextNode(_nodeElement, "255")
_nodeElement := %XML_AddElement(_nodeRoot, "size")
_nodeText := %XML_AddTextNode(_nodeElement, "48")
_nodeElement := %XML_AddElement(_nodeRoot, "charset")
_nodeText := %XML_AddTextNode(_nodeElement, "0")
_nodeElement := %XML_AddElement(_nodeRoot, "bold")
_nodeText := %XML_AddTextNode(_nodeElement, "FALSE")
_nodeElement := %XML_AddElement(_nodeRoot, "italic")
_nodeText := %XML_AddTextNode(_nodeElement, "FALSE")
_nodeElement := %XML_AddElement(_nodeRoot, "underline")
_nodeText := %XML_AddTextNode(_nodeElement, "TRUE")
_nodeElement := %XML_AddElement(_nodeRoot, "strikeout")
_nodeText := %XML_AddTextNode(_nodeElement, "FALSE")
_nodeElement := %XML_AddElement(_nodeRoot, "protected")
_nodeText := %XML_AddTextNode(_nodeElement, "FALSE")
_xmlFilter := %XML_ToString(_nodeDoc)
%HI_SetFormatRE(_richEditBox, _xmlFilter)
_ok := %XML_FreeDocument(_nodeDoc)

```



Related pages:

[Graphic object manipulation functions](#)
[Function arguments - types](#)