

XML_UpdateObjects

%XML_UpdateObjects function

Function	The function makes a mass update of the object configuration.					
Declaration	<pre>INT %XML_UpdateObjects(INT in _updateXMLHandle, INT in _filterXMLHandle)</pre>					
Parameters	<table><tr><td>_updateXMLHandle</td><td>Handle to the XML document that specifies the values of configuration parameters which are to be changed.</td></tr><tr><td>_filterXMLHandle</td><td>Handle to the XML document that contains a filter for limitation of the objects group which are to be changed.</td></tr></table>		_updateXMLHandle	Handle to the XML document that specifies the values of configuration parameters which are to be changed.	_filterXMLHandle	Handle to the XML document that contains a filter for limitation of the objects group which are to be changed.
_updateXMLHandle	Handle to the XML document that specifies the values of configuration parameters which are to be changed.					
_filterXMLHandle	Handle to the XML document that contains a filter for limitation of the objects group which are to be changed.					
Return value	Number of objects that matched the filter. If an error occurs during using the function, the function will get back the return value.					
Description	The function makes a mass update of the object configuration.					
Example						

```

TEXT _docFilterFilePath
TEXT _docDataFilePath

INT _docFilter
INT _docData

INT _elmRoot
INT _elmLev1
INT _elmLev2
INT _elmLev3
INT _elmText
BOOL _setValueOk

INT _text

INT _count

BOOL _docFilterFreeOk
BOOL _docDataFreeOk

;empty document creation for filter and for formula which will be updated

_docFilter := %XML_CreateDocument()
_docData := %XML_CreateDocument()

;document formation for filter

_elmRoot := %XML_AddElement(_docFilter,"ROOT")           ;<ROOT>
_elmLev1 := %XML_AddElement(_elmRoot,"FILTER_OPTIONS")   ;<FILTER_OPTIONS>
_elmLev2 := %XML_AddElement(_elmLev1,"Operator")         ;<Operator>
_elmText := %XML_AddTextNode(_elmLev2,"#text")
_setValueOk := %XML_SetNodeValue(_elmText,"AND")         ;AND
_elmLev1 := %XML_AddElement(_elmRoot,"CFGRECORDS")       ;<CFGRECORDS>
_elmLev2 := %XML_AddElement(_elmLev1,"TObjItemData")     ;<TObjItemData>
_elmLev3 := %XML_AddElement(_elmLev2,"Name")             ;<Name>
_elmText := %XML_AddTextNode(_elmLev3,"#text")
_setValueOk := %XML_SetNodeValue(_elmText,"H._S*")       ;H._S*
_elmLev3 := %XML_AddElement(_elmLev2,"Typ")              ;<Typ>
_elmText := %XML_AddTextNode(_elmLev3,"#text")
_setValueOk := %XML_SetNodeValue(_elmText,"ARCH_VAL")    ;ARCH_VAL

;document formation for formula which will be updated

_elmRoot := %XML_AddElement(_docData,"ROOT")           ;<ROOT>
_elmLev1:= %XML_AddElement(_elmRoot,"CFGRECORDS")       ;<CFGRECORDS>
_elmLev2 := %XML_AddElement(_elmLev1,"TObjItemData")     ;<TObjItemData>
_elmLev3 := %XML_AddElement(_elmLev2,"Name")             ;<Name>
_elmText := %XML_AddTextNode(_elmLev3,"#text")
_setValueOk := %XML_SetNodeValue(_elmText,"filter.xml")  ;filter.xml
_elmLev3 := %XML_AddElement(_elmLev2,"Descript")        ;<Descript>
_elmText := %XML_AddTextNode(_elmLev3,"#text")
_setValueOk := %XML_SetNodeValue(_elmText,"description_01") ;
description_XX

;update of the objects description convenient to filter

_count := %XML_UpdateObjects(_docData,_docFilter)       ;document import

;save on disc and release

_docFilterFreeOk := %XML_FreeDocument(_docFilter)
_docDataFreeOk := %XML_FreeDocument(_docData)

```

Note

Error description can be called by means of ESL function [%XML_GetLastErrorMessage](#) (if the function gets back the invalid value).



Related pages:

[XML file manipulation functions](#)

[Function arguments - types](#)