HI_OpenLogDBByMask

%HI_OpenLogDBByMask function

Function

The function **%HI_OpenLogDBByMask** opens the dialog box for browsing the log database (control function).

Declaration

```
%HI_OpenLogDBByMask(
   TEXT in mask,
   TEXT in skupinaMask,
   INT in priorityMask,
   BOOL in bASC,
   INT in pageLen,
   INT in intervalType
   [, HBJ in refToLG1, ...]
 )
```

or

Declaration

```
%HI_OpenLogDBByMask(
   TEXT in mask,
   TEXT in skupinaMask,
   INT in priorityMask,
   BOOL in bASC,
   INT in pageLen,
   TIME in bt,
   TIME in et
   [, HBJ in refToLG1....]
 )
```

Parameters

mask	Mask (in the dialog box **System logging - settings**).
groupMask	Filter for events source, it must be different from 0 and must be formed by sum of the values of predefined variables **_LOGF_*.**
priorityMask	Event type filter.
bASC	Order: • @TRUE - ascending • @FALSE - descending
pageLen	Page size within the interval of 5 ... 200.
intervalType	Time interval for browsing events. Possible values: • **1** - last hour • **8** - last 8 hours • **12** - last 12 hours • **24** - last 24 hours
bt	Beginning time (when you define other time interval than the intervals defined by the parameter *intervalType*).
et	End time (when you define other time interval than the intervals defined by the parameter *intervalType*).
refToLG1, ...	References to the logical groups.

Description

The function opens the dialog box for browsing the log database according to defined parameters from the ESL script. The function parameters copy behaviour of corresponding dialog box for opening the log database when you browse events for the objects matching given mask.

The parameter *groupMask* represents the filter for events source. It must be other than 0 and must be the sum of the values of the predefined variables _LOG_PRTY_*.

The parameter *priorityMask* represents the filter for *Event type.* It must be other than 0 and must be the sum of the values of the predefined variables _LOG_PRTY_*.

The parameter *intervalType* is **INT** type and allows to define the time interval for browsing events.

To define other time interval use the parameters *bt and et* of **Absolute time** type.

Objects matching the selection condition *mask* can be further filtrated by their membership in logical groups using the optional parameters *refLG1*, *refLG2*, ...

Example

```
%HI_OpenLogDBByMask("*s*", _LOGF_LOGONLOGOFF, _LOG_PRTY_INFO, @TRUE, 20,
8, AI_1_KB03\HBJ)
```

Note

The parameter *mask* may be replaced by a content of XML file representing an extended filter. The filter filtrates all the text columns in log database.
ROOT element contains the column names together with the filter conditions. Each element which represents the filter condition can also contain an attribute *strict*. This attribute defines if the filtering in some column is necessary.

The rules for elements which represent the filter conditions:

- *name, descript, person, oldvalue, newvalue* or *comment* are the reserved elements,
- ROOT element must not contain other elements than reserved one,
- each reserved element can occur only once,
- if the reserved element is not in XML file it does not belong to filter,
- if attribute *strict* has not been defined for the element or it has a different value than "off", then it is always "on"
- the records must conform to all the elements containing the attribute *strict* "on" and at least to one element containing the attribute *strict* "off" (if it exists),
- the filter condition in element enables the same syntax as mask.

Example: XML file:

<?xml version="1.0" encoding="utf-8" ?>
  <ROOT>
    <name strict="**on**">**!TF2\.**<\name>
    <descript strict="**off**">**SystemD2000***<\descript>
    <oldvalue strict="**off**">**Run**<\oldvalue>
  <\ROOT>

may be assembled according to this example:

```
 PROCEDURE AddElement(IN INT _parent, IN TEXT _name, IN TEXT _value, IN
TEXT _attribute)

 INT _eName
  INT _eValue
  BOOL _bAttribute

 _eName := %XML_AddElement(_parent, _name)
 _eValue := %XML_AddTextNode(_eName, _value)
 _bAttribute := %XML_SetAttribute(_eName, "strict", _attribute)

 END AddElement

 ENTRY btnOpenLogDB_OnClick

 INT _eDoc
 INT _eRoot

 TEXT _filterAsXML
  BOOL _ok

 _eDoc := %XML_CreateDocument()
 _eRoot := %XML_AddElement(_eDoc, "ROOT")

 CALL AddElement(_eRoot, "name", "!TF2\.", "on")
  CALL AddElement(_eRoot, "descript", "SystemD2000*", "off")
 ;CALL AddElement(_eRoot,"person", "*", "on")
  CALL AddElement(_eRoot, "oldvalue", "Run", "off")
 ;CALL AddElement(_eRoot, "newvalue", "*", "on")
  ;CALL AddElement(_eRoot, "comment", "*", "on")

 _filterAsXML := %XML_ToString(_eDoc)
 %HI_OpenLogDBByMask(_filterAsXML, _LOGF_LOGONLOGOFF, _LOG_PRTY_INFO,
@TRUE)
 _ok := %XML_FreeDocument(_eDoc)

 END btnOpenLogDB_OnClick
```

Related pages:

Active picture manipulation functions
Indexed local variables
Function arguments - types