Example of work with lists of objects (LST_... actions)

Structure of object SD.ListData mentioned in an example must correspond with structure of output parameter **_data** of ESL action LST_CREATE.

```
RECORD NOALIAS (SD.ListData) _data

TEXT _mask
INT _objType
ALIAS _logGroup
INT _sortBy
INT _lines
```

```
INT _h
INT _nrObjs
INT _nrPages

_mask    := "P.*"               ; mask of objects that are required
_objType :=  @OTYPE_ALL_TYPES   ; all types of objects
_sortBy  :=  @SORT_LIST_NAME     ; sorting according to object name
_lines   := 7                   ; number of elements on one page
```

```
; instead of _logGroup there can be filter by logical group (by entering its name)
```

```
LST_CREATE _mask, _objType, _logGroup, _sortBy, _lines, _h, _data
```

```
 IF  _h <> 0 THEN   ; if the list of objects has been created (if h=0 => error)
```

```
LST_GETINFO _h, _nrObjs  ; detection of number of all objects in list
```

```
 IF %Mod( _nrObjs, _lines ) = 0 THEN
 _nrPages := _nrObjs / _lines
ELSE
 _nrPages := _nrObjs / _lines + 1
 ENDIF
```

```
 ; if LEN(_data) = 0, then page does not exist
 ; if LEN(_data) = 0 follows after LST_CREATE, then _nrObjs = 0 and it is not important to pass through list
of objects
 ; or if LST_GO_TOP is followed by LST_GO_PREV, then LST_GO_PREV will be followed by LEN(_data) = 0
```

```
 LST_GO_PAGE   _h, _data, 2
 LST_GO_FIRST _h, _data
 LST_GO_NEXT   _h, _data
LST_GO_PREV   _h, _data
 LST_GO_LAST   _h, _data
```

```
 LST_CLOSE _h   ; closing the list of objects
```

```
ENDIF
```

Related pages:

Script action