

# KVIT, BLOCK, UNBLOCK

## KVIT, BLOCK, UNBLOCK actions

### Declaration

```
[_retInt := ] KKIT alarmObjIdent
```

```
[_retInt := ] BLOCK alarmObjIdent[, userObjIdent]
```

```
[_retInt := ] UNBLOCK alarmObjIdent[, userObjIdent]
```

### Parameters

_retInt	out	Identifier of the <i>Int</i> type - action success.
alarmObjIdent	in	Reference to an <i>Alarm</i> type object, or to an object, which can generate a process alarm.
userObjIdent	in	Reference to an <i>User</i> .

### Description

The actions allows to control the alarm states of the D2000 system objects.

- **KKIT** - alarm acknowledgement
- **BLOCK** - alarm blocking or muting for user, if parameter *userObjIdent* is given
- **UNBLOCK** - alarm unblocking or unmuting for user, if parameter *userObjIdent* is given

Every action may be executed in two ways:

- **Synchronously** - notation with an assignment. Action waits for the execution of the given command. Action return code may get one of the following values:
  - `_ERR_TRANS_ABORT`
  - `_ERR_TRANS_ERROR`
  - `_ERR_TRANS_IGNORED`
  - `_ERR_NO_ERROR`
- **Asynchronously** - notation without any assignment. Action will require the system to execute the given command and the script continues executing another actions.

### Example

```
BEGIN
  INT _stav
  _stav := BLOCK alarm
  IF _stav = _ERR_NO_ERROR THEN
    MESSAGE "alarm has been blocked" ON srvskollv.HIP
    _stav := KKIT alarm ;attempt to acknowledge a blocked alarm
  IF _stav = _ERR_NO_ERROR THEN
    MESSAGE "alarm has been acknowledged" ON srvskollv.HIP ;cannot be
executed
  ELSE
    MESSAGE "alarm cannot be acknowledged, it is blocked" ON srvskollv.HIP
  ENDF
ELSE
  MESSAGE "failed to block an alarm" ON srvskollv.HIP
ENDIF
END
```



**Related pages:**

[Script actions](#)

[User Window for Alarm Management](#)