hotkey

COLLABORATORS			
	TITLE : hotkey		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		January 17, 2023	

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

hotkey

1.1 hotkey.doc

```
--background--

Activate()

AddKey()

CheckCVersion()

DeleteBroker()

GetCMsg()

Init()
```

1.2 HotKey/--background--

```
                  DESCRIPTION
This module allows easy access to hotkey-specific functions of the
commodity.library.

 NOTE

              DeleteBroker()
               and
              CheckCVersion()
               are the only procedures which
are very kind to OS1.x, because ONLY they are supposed to be called on
Operating Systems other than Release 2+.

Therefore you MUST NOT call any of the module's
procedures except
              DeleteBroker()
               and
              CheckCVersion()
```

```
                on
Systems without commodities.library.

In addition the version of commodities.library must be 37
or higher, because this version is demanded by Oberon 3.00
Interfaces for the commodities.library
( OpenLibrary(commodities.library,37) ).

After version check failed, you may gracefully exit, e.g.
  IF ~ CheckCVersion(0) THEN HALT(20) END;

There should be no need to IMPORT Commodities, because all
important constants are redefined. Some constants are even
defined wrong in Interfaces for Oberon3.00!


 EXAMPLE
Refer to HotKeyTest for a example how to use this module.

 AUTHOR
Program:  V1.0  28-Apr-92  Thomas Igracki
                first release
          V2.0  21-Dec-92  Thomas Wagner [tom]
                removed some bugs (no ReplyMsg ...)
                and completely revised.
                NOTE: ** Interface has changed! **

Autodocs: Thomas Wagner [tom]
```

1.3 HotKey/Activate

```
 NAME
Activate -- (De)activates hotkeys

 SYNOPSIS
Activate (ON: BOOLEAN)

 FUNCTION
This function activates or deactivates the broker and its hotkeys.

 INPUTS
ON - TRUE to activate, FALSE to deactivate
```

1.4 HotKey/AddKey

```
                  NAME
AddKey -- Add a hotkey description to the broker

 SYNOPSIS
AddKey (descr: ARRAY OF CHAR; ID: LONGINT):BOOLEAN
```

```
 FUNCTION
Adds the given hotkey to the broker.

 INPUTS
descr - containing the hotkey description
ID    - any number to recognize this hotkey. Not
        used by the module or by the operating system.

 RESULT
BOOLEAN - indicates the success of the hotkey installation. If
          FALSE, ErrorSet contains the error code.
          Note: Multiple error code bits are possible.
          The most important error:
           badDescription: wrong hotkey description

 SEE ALSO

              GetCMsg()
```

1.5 HotKey/CheckCVersion

```
 NAME
CheckCVersion -- Checks the version of the Commodities.library.

 SYNOPSIS
CheckCVersion (version: SHORTINT): BOOLEAN;

 FUNCTION
  Check whether the required version is available.

 INPUTS
version - The required version. 0 to check if there is
          any usable version of commodities.library available.
          (See NOTE!).

 RESULT
BOOLEAN - TRUE  if Commodity has the required version or higher
          FALSE if not available

 NOTE
V36 is considered out of date. So if you use the Interfaces for
commodity.library delivered with Oberon3.00 (or later versions)
CheckCVersion(0) will return FALSE!
```

1.6 HotKey/DeleteBroker

```
                 NAME
DeleteBroker -- Removes a broker

 SYNOPSIS
DeleteBroker()
```

```
 FUNCTION
This function removes the broker (and all attached hotkeys) and
closes the MsgPort.

 NOTES
This function is automatically called. No need to call it
yourself except you want to delete the broker earlier.

 SEE ALSO

              Init()
```

1.7 HotKey/GetCMsg

```
                  NAME
GetCMsg -- Get commodity message

 SYNOPSIS
GetCMsg(VAR type: LONGSET; VAR id: INTEGER):BOOLEAN

 FUNCTION
If you get the signal (value returned by
              Init()
              ) then you must
call GetCMsg() to determine which hotkey was pressed or which
command was sent.

 INPUTS
VAR type - Contains hotkey if an Hotkey was pressed or command
           if a command was sent. Note: cDisable and cEnable
           are handled automatically by this module. So you
           should ignore them, except you want to add some
           more functionality to them.
VAR id   - An integer value to store the id of a hotkey (if there
           is any Msg). Only valid if type equals command.

 RESULT
BOOLEAN - TRUE if there was a Msg and type and id is valid.
          Call GetCMsg() again, because there might be more Msg's.

 EXAMPLE
IMPORT Exec, HotKey;
VAR HotID  : LONGINT;
    HotType: LONGSET;
    HotSig : SHORTINT;
    Signals: SHORTINT;
...
IF ~HotKey.CheckCVersion(0) THEN HALT(20) END;
HotSig := HotKey.Init(...);
...
Signals := Exec.Wait(LONGSET{HotSig});
...
IF HotSig IN Signals THEN
```

```
  WHILE HotKey.GetCMsg(HotType,HotID) DO
         IF HotKey.hotkey IN HotType THEN
      CASE HotID OF
        ...
      END; (* CASE *)
    ELSIF HotKey.command IN HotType THEN
      CASE HotID OF
        hot.cAppear   : (* ShowInterface *) |
        hot.cDisappear: (* HideInterface *) |
        hot.cKill     : (* Quit          *) |
        hot.cUnique   : (* ShowInterface or Quit *) |
        ELSE            (* ignore cDisable and cEnable *)
      END; (* CASE *)
    END; (* IF *)
  END; (* WHILE *)
END; (* IF *)

For a full example see HotKeyTest.mod.
```

1.8 HotKey/Init

```
                  NAME
Init -- Initiate a broker.
InitX -- Initiate a broker (extended parameters).

 SYNOPSIS
InitX (name,title,descr: ARRAY OF CHAR;
       unique          : SET;
       window          : BOOLEAN;
       pri             : SHORTINT       ): SHORTINT;

Init (name,title,descr: ARRAY OF CHAR): SHORTINT;

 FUNCTION
  Initiates some structures needed by all other procedures
of this module and installs a broker.

 INPUTS
name   - The name of the broker. You should not exceed the length
         defined in Commodities.nameLen.
title  - Name, version, ... of the commodity. You should not exceed
         the maximum length defined in Commodities.titleLen.
descr  - A short summary of the functionality of this commodity.
         You should not exceed the length defined in
         Commodities.descrLength.

Only for InitX():

unique - Set to one or more of Flags for Unique.
         Init: set to {notify}.
window - Set this to TRUE when your commodity is able to open
         a window. If this is set to FALSE the Show/Hide gadgets
         of the exchange program are deactivated.
         Init: set to TRUE.
```

```
pri    - The priority of the broker. Init: set to 0.

 RESULT
SHORTINT - A positive number is the Signalbit for hotkey
           activity. -1 indicates an error. See the global
           readonly variable Error for the cause.
           Here are some important errors:
            systemError : usually no memory
            calledTwice : called this routine twice
            brokerNIL   : usually already another broker with
                          same name

 NOTES
- After Init() or InitX() you must call
              Activate()
               to activate
  the broker.
- You can't call Init() twice.

 SEE ALSO

              DeleteBroker()
```