

in

COLLABORATORS

	<i>TITLE :</i> in		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		October 9, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	in	1
1.1	RICommoditiesLib	1
1.2	overview	2
1.3	exchangelistchange	2
1.4	cxlistchange	3
1.5	makecommodity	3
1.6	sethotkey	3
1.7	hotkeyhit	3
1.8	commodityevent	4
1.9	setcxstatus	4
1.10	exchangemessage	4
1.11	cxappear	5
1.12	cxdisappear	5
1.13	cxenable	5
1.14	cxdisable	5
1.15	cxkill	6
1.16	exchangelist	6
1.17	cxunique	6
1.18	exchangeappear	6
1.19	exchangedisappear	7
1.20	exchangeenable	7
1.21	exchangedisable	7
1.22	exchangekill	7
1.23	exchangechangelist	8
1.24	exchangeunique	8
1.25	commoditymsgport	8
1.26	Example Programs	8

Chapter 1

in

1.1 RICommoditiesLib

RI Commodities Library V2.5

©1996 Red When Excited Ltd

Undocumented commands added by Toby Zuijdveld 28/02/1999.
mailto: hotcakes@abacus.net.au

Overview

Command Index

MakeCommodity

CxKill

SetHotKey

CxUnique

HotKeyHit

CxChangeList

CommodityEvent

ExchangeAppear

SetCXStatus

ExchangeDisAppear

ExchangeMessage

ExchangeEnable

1.4 cxlistchange

Functions : CxListChange

Modes : Amiga

This is used in conjunction with ExchangeMessage, see CxAppear for more information.

1.5 makecommodity

Function : MakeCommodity

Modes : Amiga

Syntax : success=MakeCommodity(name\$,title\$,description\$)

This command attempts to add your Commodity to the list of commodities. A return value of -1 indicates success, 0 means failure. (not enough memory)

name\$ refers to the name of the Commodity and it should be unique. This is the name that appears when running the Commodity Exchange program.

title\$ is the title of your program, e.g. "My Screen Blanker".

description\$ is a brief description of your program.

The Commodity Exchange program will then have 'name\$' in its list of Commodities and when a user clicks on your commodity, it will display the title\$ and description\$.

1.6 sethotkey

Function : SetHotKey

Modes : Amiga

Syntax : success=SetHotKey(hotkey#,hotkeydescription\$)

This will add a hotkey event to your commodity so that after a hotkey has been pressed you can find out which one.

e.g.

```
success=SetHotKey(0,"lalt lshift a")
```

1.7 hotkeyhit

Function : HotKeyHit

Modes : Amiga

Syntax : hitkeynum=HotKeyHit

This will return the number of the hot key which has been hit since the last 'CommodityEvent' was called, or -1 if no such hotkey has been activated.

1.8 commodityevent

Function : CommodityEvent

Modes : Amiga

Syntax : anyevent=CommodityEvent

This looks to see if either:

- a) A hotkey has been pressed
- b) A message from Exchange has been received

and returns -1 if such an event occurred, 0 if nothing has happened yet. This should be inside a Repeat-Until loop

e.g.

```
Repeat
  VWait
  ev.l=Event
  ce.l=CommodityEvent
  hk.l=HotKeyHit          ; This must be used after CommodityEvent
Until ev or ce or hk
```

1.9 setcxstatus

Statement : SetCXStatus

Modes : Amiga

Syntax : SetCXStatus on|off

This sets the status of your Commodity to either Active (on) or Inactive (off) - this can be seen by running the Commodities Exchange program.

NOTE: WAS SetStatus -> Changed to clarify the fact its a Commodity Lib command.

1.10 exchangemessage

Function : ExchangeMessage

Modes : Amiga

Syntax : messnum.l=ExchangeMessage

This looks to see if the Commodities Exchange has issued you with a message e.g. Hide Interface, Show Interface. It returns the message ID of the incoming message or 0 for no message.

1.11 cxappear

Functions: CxAppear

Modes : Amiga
Syntax : CxAppear

This is used in conjunction with ExchangeMessage

e.g.

```
em.l=ExchangeMessage
Select em
  Case CxAppear
    Gosub _appear
  Case CxDisAppear
    Gosub _disappear
End Select
```

The function merely returns the ID value associated with that particular Commodities Exchange message.

1.12 cxdisappear

Functions: CxDisAppear

Modes : Amiga
Syntax : CxDisAppear

This is used in conjunction with ExchangeMessage, see CxAppear for more information.

1.13 cxenable

Functions: CxEnable

Modes : Amiga
Syntax : CxEnable

This is used in conjunction with ExchangeMessage, see CxAppear for more information.

1.14 cxdisable

Functions: CxDisable

Modes : Amiga
Syntax : CxDisable

This is used in conjunction with ExchangeMessage, see CxAppear for more information.

1.15 cxkill

Functions: CxKill

Modes : Amiga
Syntax : CxKill

This is used in conjunction with ExchangeMessage, see CxAppear for more information.

1.16 cxchangelist

Functions: CxChangeList

Modes : Amiga
Syntax : CxChangeList

This is used in conjunction with ExchangeMessage, see CxAppear for more information.

1.17 cxunique

Functions: CxUnique

Modes : Amiga
Syntax : CxUnique

This is used in conjunction with ExchangeMessage, see CxAppear for more information.

1.18 exchangeappear

Functions: ExchangeAppear

Modes : Amiga
Syntax : ExchangeAppear

This is used in conjunction with ExchangeMessage

e.g.

```
em.l=ExchangeMessage
If em
  If ExchangeAppear then Gosub _appear
```

```
If ExchangeDisAppear then Gosub _disappear
EndIf
```

This is intended as an alternative way of acting upon Exchange Messages.

1.19 exchangedisappear

Functions: ExchangeDisAppear

Modes : Amiga
Syntax : ExchangeDisAppear

This is used in conjunction with ExchangeMessage, see ExchangeAppear for more information.

1.20 exchangeenable

Functions: ExchangeEnable

Modes : Amiga
Syntax : ExchangeEnable

This is used in conjunction with ExchangeMessage, see ExchangeAppear for more information.

1.21 exchangedisable

Functions: ExchangeDisable

Modes : Amiga
Syntax : ExchangeDisable

This is used in conjunction with ExchangeMessage, see ExchangeAppear for more information.

1.22 exchangekill

Functions: ExchangeKill

Modes : Amiga
Syntax : ExchangeKill

This is used in conjunction with ExchangeMessage, see ExchangeAppear for more information.

1.23 exchangechangelist

Functions: ExchangeChangeList

Modes : Amiga

Syntax : ExchangeChangeList

This is used in conjunction with ExchangeMessage, see ExchangeAppear for more information.

1.24 exchangeunique

Functions: ExchangeUnique

Modes : Amiga

Syntax : ExchangeUnique

This is used in conjunction with ExchangeMessage, see ExchangeAppear for more information.

1.25 commoditymsgport

Function: CommodityMsgPort

Modes : Amiga/Blitz

Syntax : msgport.l=CommodityMsgPort

This function returns the address of the Commodity's message port. Useful for doing extra things such as AddWaitEvent PortSigBit(CommodityMsgPort),xxxx

1.26 Example Programs

Example Programs

EXAMPLE 1 - a general Commodities example :

```
Load Example 1
Compile It!
```