

For example, the input description string “rawkey -caps -lalt -relativemouse -upstroke ralt tab” matches a tab upstroke or downstroke with the right alt key pressed whether or not the left alt, either shift, or the capslock keys are down. The following IX structure corresponds to that input description string:

```
IX ix = {
    IX_VERSION,                /* The version */
    IECLASS_RAWKEY,            /* We're looking for a RAWKEY event */
    0x42,                      /* The key the usa0 keymap maps to a tab */
    0x00FF & (~IECODE_UP_PREFIX), /* We want up and down key presses */
    IEQUALIFIER_RALT,          /* The right alt key must be down */
    0xFFFF & ~(IEQUALIFIER_LALT | IEQUALIFIER_LSHIFT |
        IEQUALIFIER_RSHIFT | IEQUALIFIER_CAPSLOCK | IEQUALIFIER_RELATIVEMOUSE),
    /* don't care about left alt, shift, capslock, or relativemouse qualifiers */
    IXSYM_CAPS /* The shift keys and the capslock key qualifers are all equivalent */
};
```

The CxFilter() macro only accepts a description string to describe an input event. A commodity can change this filter, however.

```
void SetFilter( CxObj *filter, char *descrstring );
void SetFilterIX( CxObj *filter, IX *ix );
```

SetFilter() and SetFilterIX() change which input events a filter CxObject diverts. SetFilter() accepts a pointer to an input description string. SetFilterIX() accepts a pointer to an IX input expression. A commodity that uses either of these functions should check the filter's error code with CxObjError() to make sure the change worked.

```
errorcode = LONG ParseIX( char *descrstring, IX *ix );
```

The function ParseIX() parses an input description string and translates it into an IX input expression. Commodities Exchange uses ParseIX() to convert the description string in CxFilter() to an IX input expression. As was mentioned previously, as of *commodities.library* version 37.3, ParseIX() does not work with certain kinds of input strings.

This article is by no means an exhaustive description of Commodities Exchange. For more information, see the Commodities Exchange Autodocs (*commodities.doc* for *commodities.library* and *cx.doc* for the macros and linker library functions) and the sample commodities from the 1990 Atlanta DevCon disks.