

```
.page. newpath
58 72 moveto 0 9 rlineto 0 -9 rmoveto 9 0 rlineto
58 720 moveto 0 -9 rlineto 0 9 rmoveto 9 0 rlineto
stroke
```

MicroPhone II Technical Note



#2: Accessing MicroPhone Variables from an XCMD

See also: Building XCMDs & XFCNs [Resources & Utilities Manual]
 Using Special Callbacks [Resources & Utilities Manual]

Written by: Dave Newman January 24th, 1990

XCMDs and XFCNs developed for use with MicroPhone II can create and access the variables used within MicroPhone II's script language.

This can be achieved by means of the standard Hypercard callback, `GetGlobal` and `SetGlobal`. There is however one exception to this scheme. As noted in the "Building XCMDs & XFCNs" of the "Resources & Utilities" manual, XFCNs can not utilize the `SetGlobal` callback.

The variable name passed to either `GetGlobal`, or `SetGlobal` can be any standard MicroPhone II variable name. If one is calling `SetGlobal` for variable 'nuVar' and that variable does not exist, then MicroPhone II will create 'nuVar' and set it equal to the string passed as a value. All values passed to `SetGlobal` or returned from `GetGlobal` will be strings. Passing anything other than a string is likely to cause MicroPhone II to blow up in your face!

Variable names can also serve as an index into an indexed list (i.e. 'myList[4]') as well. This means that XCMDs and XFCNs can also access the individual elements of an indexed list. If one does a `SetGlobal` for an indexed list variable that does not exist then, as before, MicroPhone II will create the variable, make it an indexed list and assign the string to the specified element.

Please note that, one should not ask for the entire list or a sub list in one callback. Doing so will result in the return of an empty string - a rather useless entity to say the least.

Below are some examples of the ways to use `GetGlobal` and `SetGlobal` from an XCMD or XFCN.

Assuming a script has created the following variables already:

```
myInteger = 2;
myString = 'abc';
myBoolean = false;
myList[1] = 123;
myList[2] = true;
myList[3,1] = 'xyz';
```

```

.page. newpath
58 72 moveto 0 9 rlineto 0 -9 rmoveto 9 0 rlineto
58 720 moveto 0 -9 rlineto 0 9 rmoveto 9 0 rlineto
stroke

```

```
{ Pascal examples }
```

```
Procedure GetMPValues(anXCMD: XCmdPtr);
```

```
Var
```

```
myHdl: Handle;
```

```
Begin
```

```

myHdl := GetGlobal(anXCMD, 'myInteger'); {returns handle to '2'}
myHdl := GetGlobal(anXCMD, 'myString'); {returns handle to 'abc'}
myHdl := GetGlobal(anXCMD, 'myBoolean'); {returns handle to 'false'}
myHdl := GetGlobal(anXCMD, 'myList[3,1]'); {returns handle to 'xyz'}
myHdl := GetGlobal(anXCMD, 'myList[myInteger]'); {returns handle to
                                                    'true'}

myHdl := GetGlobal(anXCMD, 'myList'); {returns handle to empty
                                       string}

```

```
End;
```

```
Procedure SetMPValues(anXCMD: XCmdPtr);
```

```
Var
```

```
aHdl: Handle;
```

```
Begin
```

```

aHdl := Handle(NewString('3'));
SetGlobal(anXCMD, 'myInteger', aHdl); {set myInteger to '3'}
DisposHandle(aHdl);

aHdl := Handle(NewString('456'));
SetGlobal(anXCMD, 'myList[1]', aHdl); {set myList[1] to '456'}
DisposHandle(aHdl);

aHdl := Handle(NewString('true'));
SetGlobal(anXCMD, 'myList[3,1]', aHdl); {set myList[3,1] to 'true'}
DisposHandle(aHdl);

aHdl := Handle(NewString('Hi!'));
SetGlobal(anXCMD, 'newVar', aHdl); {create & set newVar to 'Hi!'}
DisposHandle(aHdl);

aHdl := Handle(NewString('Bye'));
SetGlobal(anXCMD, 'newList[1]', aHdl); {create newList & set
                                       newList[1] to 'Bye'}

DisposHandle(aHdl);
End;
```

```

.page. newpath
58 72 moveto 0 9 rlineto 0 -9 rmoveto 9 0 rlineto
58 720 moveto 0 -9 rlineto 0 9 rmoveto 9 0 rlineto
stroke

```

```

/* C Examples */

void GetMPValues(anXCMD)
XCMDPtr anXCMD;
{
    Handle myHdl;

    myHdl = GetGlobal(anXCMD, "\pmyInteger"); /* returns handle to "2" */
    myHdl = GetGlobal(anXCMD, "\pmyString"); /* returns handle to "abc" */
    myHdl = GetGlobal(anXCMD, "\pmyBoolean"); /* returns handle to
                                                "false" */
    myHdl = GetGlobal(anXCMD, "\pmyList[3,1]"); /* returns handle to "xyz" */
    myHdl = GetGlobal(anXCMD, "\pmyList[myInteger]"); /* returns handle to
                                                        "true" */
    myHdl = GetGlobal(anXCMD, "\pmyList"); /* returns handle to empty
                                            string */
}

void SetMPValues(anXCMD)
XCMDPtr anXCMD;
{
    Handle aHdl;

    aHdl = Handle(NewString("\p3"));
    SetGlobal(anXCMD, "\pmyInteger", aHdl); /* set myInteger to "3" */
    DisposHandle(aHdl);

    aHdl = Handle(NewString("\p456"));
    SetGlobal(anXCMD, "\pmyList[1]", aHdl); /* set myList[1] to "456" */
    DisposHandle(aHdl);

    aHdl = Handle(NewString("\ptrue"));
    SetGlobal(anXCMD, "\pmyList[3,1]", aHdl); /* set myList[3,1] to
                                                "true" */
    DisposHandle(aHdl);

    aHdl = Handle(NewString("\pHi!"));
    SetGlobal(anXCMD, "\pnewVar", aHdl); /* create & set newVar to
                                           "Hi!" */
    DisposHandle(aHdl);

    aHdl = Handle(NewString("\pBye"));
    SetGlobal(anXCMD, "\pnewList[1]", aHdl); /* create newList & set
                                              newList[1] to "Bye" */
    DisposHandle(aHdl);
}

```