

1.

A feature is a UInt32 data item. it is stored in a feature table
Rom has a
table of os features which is copied to a RAM feature table when
machine 1st starts

can use to publish what features it provides
will not survive a soft reset

```
FtrGet(sysFtrCreator, sysFtrNumRomVersion, &romVersion);  
useful to indicate what features are available
```

2.

```
static Boolean FormhandleEvent(...)  
{  
    switch(event->type)  
    {  
        case keyDownEvent:  
            switch(event->data.keyDown.char)  
            {  
                case h:  
                    //do something  
                    break  
            }  
        }  
    }  
}
```

3.

```
for(i = 0; i < 4; i++)  
{  
    TblSetItemStyle(table, i, 0, checkboxTableItem);  
    TblSetItemInt(table, i, 0, 0);  
    TblSetItemStyle(table, i, 1, labelTableItem);  
    TblSetItemPtr(table, i, 1, gLabel[i]);  
    TbleSetItemStyle(table, i, 2, textTableItem);  
}
```

5.

```
typedef struct{  
    UInt8 x;  
}MyType;  
  
MyType pref;  
UInt16 size = sizeof(MyType);
```

```
pref.x = 3;
```

```
//to set
```

```
prefSetAppPreferences(creatorID, myPrefID, version, &pref, size,  
true);
```

```
//to get
```

```
prefGetAppPreferences(creatorID, myPrefID, &pref, size, saved);
```

6.

```
h = DmNewhandle(dbRef, sizeof(MyAppInfo));  
id = MemHandleToLocalID(h);  
DmSetDatabaseInfo(0, dbRef, NULL, NULL, NULL,  
NULL, NULL, NULL, NULL,  
&id, NULL, NULL, NULL);  
appInfo = MemLocalIDToLockedPtr(id, 0);  
DmSet(appInfo, 0, sizeof(MyAppInfo), 0);  
MemPtrUnlock(appInfo);
```

8.

```
Err error = ExgRegisterDatatype(myCreatorID, exgRegTypeID, "text/  
plain", 'text', 0);
```

9.

```
_btobex://99:88:77:66:55:44,FF:EE:DD:CC:BB:AA/myRecord.txt
```

10.

```
NetIPAddr address = 0;  
NetIPAddr *addressP = NULL;  
NetHostInfoBuffTye hostInfoBuff;  
NetHostInfoBufPtr HostInfoBufP = &HostInfoBuf;  
NetInfoPtr hostInfoP = NULL;  
HostInfoP = NetLibGetHostByName(gNetRefNum, "bob.the.builder",  
HostInfoBufP, 1000, &error);  
addressP = (NetIPaddress*)HostInfoP->addressLstP[0];  
address = NetNtoHL(*addressP);
```

4) Err error = DmCreateDatabase(0, "Bob Stuff", "bobs",
"DATA", false);

7) typedef UInt6 DmCompare F(void* rec1, void* rec2,
Int16 other, SortRecordInfoPtr recSortInfo,
SortRecordInfoPtr rec2SortInfo, MemHandle appInfo);