

```
/* *****  
UEnhancedIcon.h
```

Written by Ron Conescu
Copyright ©1993 Information Management Incorporated. All rights reserved.

TEnhancedIcon:

Uses separate ICON or cicon resources for the icon's unhighlighted (normal, unselected), highlighted (selected), and dimmed (inactive) states. The resources are expected to occur in the resource file as follows:

Res#	meaning
n	Normal, unclicked
n+1	Selected, clicked
n+2	Dimmed

TRadioIcon:

Works like a radio button: put several TRadioIcons in a TCluster, and set their EventNumbers to mRadioHit (in TControl).

Has two states, on (selected) and off (deselected). Each state has an icon associated with it, as follows:

Res#	meaning
n	Off (unselected)
n+5	On (selected)

They are 5 numbers apart to leave room for the TEnhancedIcon's resources: n+1, n+2, (n+5)+1, (n+5)+2.

When setting up your application, call HideOnSuspend to tell the icon whether or not you want it to HIDE or DIM when its window is suspended; default is DIM.

Modifications:

9/9/93 Ron Conescu Added TToggleIcon.

```
***** */
```

```
#ifndef __UEnhancedIcon__  
#define __UEnhancedIcon__
```

```
#ifndef __MACAPP__  
#include <MacApp.h>  
#endif
```

```
const Boolean kSelected = true;  
const Boolean kDeselected = false;
```

```
/* =====  
TEnhancedIcon  
===== */
```

```
class TEnhancedIcon : public TIcon  
{  
public:  
virtual void pascal Initialize (void);  
virtual void pascal Activate (Boolean entering);  
virtual void pascal DimState (Boolean state, Boolean redraw);  
virtual void pascal Hilite (void);
```

```

virtual void pascal ReadFields (TStream* aStream); // override of TIcon

virtual void On (Boolean redraw = true);
virtual void Off (Boolean redraw = true);
virtual void HideOnSuspend (Boolean hideMe = true);

protected:
ResNumber fNormalRsrcID; // Res# n
ResNumber fHilitedRsrcID; // Res# n+1
ResNumber fDimmedRsrcID; // Res# n+2

ResNumber fCurrentRsrcID;
ResNumber fInactiveRsrcID; // either fDimmedRsrcID or kNoResource
};

```

```

/* =====
TRadioIcon

```

TRadioIcon is effectively two TEnhancedIcons -- one for the radio's "on" state, one for the "off" state. This means that this object should have six (count 'em, SIX) cicons associated with it. The first three are the "off" state, starting at resource ID n; the second three are the "on" state, starting at n+5. Thus, the icons and their states are:

Res ID	meaning
n	state = off, normal
n+1	state = off, selected
n+2	state = off, dimmed
n+5	state = on, normal
n+6	state = on, selected
n+7	state = on, dimmed

If you want to use the "on" icon in your view editor, make sure to call TRadioIcon::SetStartState. See TRadioIcon::SetStartState.

TRadioIcon mimics the properties of a radio button (TRadio). However, since TIcon is not a subclass of TControlMgr, Toggle, SetState, and IsOn use fState, not SetLongVal().

Modifications:

9/9/93 Ron Conescu Initial implementation.

```

===== */

```

```

class TRadioIcon: public TEnhancedIcon
{
public:
virtual void pascal Initialize (void);
virtual void pascal DoPostCreate (TDocument* itsDocument);
virtual void pascal DoEvent (EventNumber eventNumber,
TEventHandler* source,
TEvent* event);

// Call this if you want to use the "on" icon in the view editor
// instead of the "off" icon:
virtual void SetStartState (Boolean state);

// Adapted from TRadio:
virtual void pascal Toggle ();
virtual void pascal SetState (Boolean state);
virtual Boolean pascal IsOn (void);

```

protected:

```
// fOffID and fOnID represent the "off" and "on" states of this
// "radio button"; each is the first of a set of three
// TEnhancedIcon icons.
Boolean      fState;          // On or off (active or inactive).
ResNumber    fOffID;         // Res# n (initialized to fNormalRsrcID)
ResNumber    fOnID;         // Res# n+5
};

#endif
```