

Q: I would like to have an object in my custom palette do something different when it is unarchived in test mode in Interface Builder. How can I determine if I am in test mode?

A: You can use the **isTestingInterface** method which is defined in the IB (Interface Builder) protocol. See [/NextLibrary/Documentation/NextDev/GeneralRef/08_InterfaceBuilder/Protocols/IB.rtf](#):

Protocol Description

Interface Builder's subclass of the Application class conforms to this protocol. Thus, objects in your custom palette can interact with Interface Builder's main module by sending messages (corresponding to the methods in this protocol) to NXApp.

isTestingInterface

- (BOOL)isTestingInterface

Returns YES if Interface Builder is in Test mode.

However, there is a bug with the **isTestingInterface** method in Release 3.1 and after. Even in test mode, **[NXApp isTestingInterface]** always yields **NO** during unarchiving (**read:**, **awake** and **finishUnarchiving**). One possible workaround is to delay the **isTestingInterface** message by calling **perform:with:afterDelay:cancelPrevious:**. Another workaround you can consider is illustrated in Jeff Martin's StringList MiniExample. You can take a look at the StringList MiniExample available via NeXTAnswers, document #1254. The workaround is implemented in the **awake** method for FilenameList. It is *only* valid in *awake*:

```
// test to see if mainMenu is visible, which it isn't while test
// interface mode is being set up.
```

```
if([NXApp conformsTo:@protocol(IB)])
```

```
if([[NXApp mainMenu] isVisible]) {  
    // do something special for test mode  
    return self;  
}
```

QAxx

Valid for 3.0, 3.1