Q: I've written an application which instantiates a Listener subclass and checks in its listenPort under a well-known name so that other applications can find it using NXPortNameLookup().   In Release 2.x, applications on a remote hosts were able to look up this port without problems, but when I run my application under Release 3.0, it can't be found.   What's changed?

Q: I find that in Release 3.0 my applications which make use of Speaker/Listener messages between two hosts can only do so if Public Window Server is enabled for the host on the Listener side.   Why is this so?

A: Prior to Release 3.0,   Speaker/Listener messages were not secure between hosts (i.e., it was possible to obtain and send messages to the port of a Listener on another machine).   This was possible because the Listener's port was registered with the local Network Name Server under the name passed to Listener's **checkInAs:** method exactly as specified.

In Release 3.0, the same is true only if Public Window Server is enabled (this is documented in the AppKit Release Notes).   If not, the name passed to the   **checkInAs:** method is modified so as to make it unique to the local host before the port is registered.   Similarly, **NXPortFromName()** and **NXPortNameLookup()** attempt to find registered Listener ports under the requested name and the local modified form of that name.   An application which attempts to look up a port on a remote host which has Public Window Server disabled will not be able to find it because that port will be registered under a name unique to that remote host.

It's wise to have this restriction on the application's default Listener and to have Public Window Server

disabled for security purposes.   If this is so, however, and you instantiate another Listener (or Listener subclass) in your application which will be registered under a name which is different from that of the application itself and must be accessible to remote applications, it can't be registered using Listener's **checkInAs:** method.   The following category of Listener implements an additional method (**publicCheckInAs:**) which forces the registered name of the Listener's port to be the name requested whether Public Window Server is enabled or not.

```
#import <appkit/appkit.h>
#import "Controller.h"

#import <servers/netname.h>

@interface Listener(PublicPort)

- (int)publicCheckInAs:(const char *)name;

@end

@implementation Listener(PublicPort)

/*
 * This method should be used INSTEAD OF -checkInAs:, not in addition to it.
 * Using both may lead to problems later if checked in again under the same
 * name.
 */
```

```objc
- (int)publicCheckInAs:(const char *)name
{
    kern_return_t result;

    if (![self listenPort]) {
        /*
         * Causes Listener to alloc a port.
         */
    [self usePrivatePort];
  }

    /*
     * Now register the port with the Network Name Server.
     */
    result = netname_check_in(name_server_port,
    (char *)name,[self signaturePort],[self listenPort]);
 if (result != NETNAME_SUCCESS)
            return result;

    /*
     * Set my portName to be the registed name.
     */
    free(portName);
    portName = NXCopyStringBufferFromZone(name, [self zone]);
    return 0;
}
```

@end

QA870

Valid for 3.0