MediaMatrix®

A Division of Peavey Electronics Corporation

NWare

NWare User Guide

Version 2.0.0.0

July 7, 2017

Copyright notice

The information contained in this manual is subject to change without notice. Peavey Electronics is not liable for improper installation or configuration. The information contained herein is intended only as an aid to qualified personnel in the design, installation and maintenance of engineered audio systems. The installing contractor or end user is ultimately responsible for the successful implementation of these systems.

Email:*mmtechsupport@peavey.com* (*mailto:mmtechsupport@peavey.com*).

Scope

This guide introduces and explains how to use the features of NWare, the network design and management tool.

If you want to start working with audio devices on the network, it is assumed that you have already installed them, as described in the individual manuals for those devices.

Contents

Chapter 1

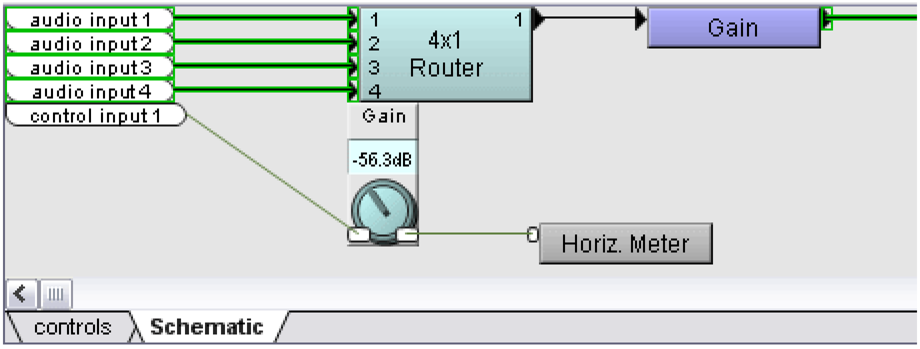
Introduction

In This Chapter

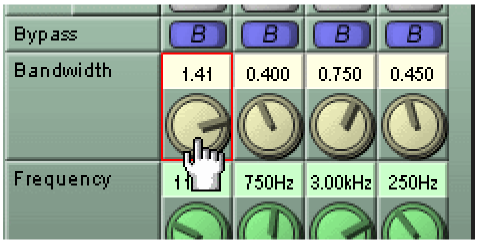
What is NWare?

NWare is a Windows application that allows you to design and manage networked audio systems.

Within NWare, an audio system is referred to as a *project*, and is created by dragging and dropping components and wiring them together on screen. Components you would normally find in an audio system, like gain controls, meters, equalizers, etc. are represented by on screen objects and wired together within NWare.



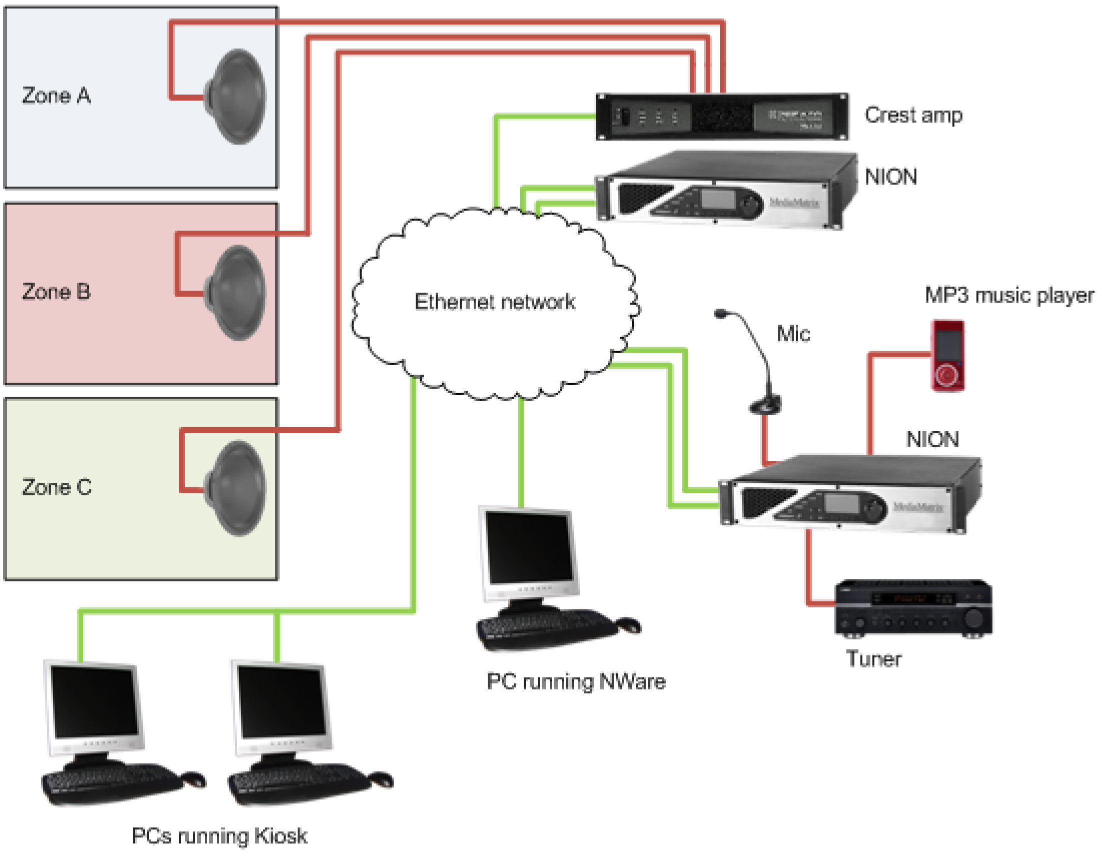
You can test out your design within NWare without using any hardware devices on the network (NIONs, CAB 4ns etc.) using the *emulate* feature. No audio processing is actually performed, but you can test out the controls in your design (move sliders, turn knobs etc.), to see how the user will interact with them when the project is running.



When you are ready to actually start using the audio system, you *deploy* it to a NION on the local area network. Once running, the project is independent of NWare. Users can connect to it using NWare or NWare:Kiosk, or third party systems and use the controls.

Example network set up

The diagram below shows an example set up that you could manage using NWare.



Audio from the different sources: microphone, MP3 music player and tuner can be made available in the different zones. Users of Kiosk can use a simple user interface, designed using NWare, to select audio sources and zones, and also control the volume and tone, for example. When announcements need to be made, the music in a particular zone can be muted so the person using the microphone can be heard.

An audio system like this one can be scaled up to provide audio from further sources and to more locations.

The devices on the network that use CobraNet or Dante – the NIONs and Crest amplifiers in the above example – are on a separate VLAN to the other devices. This ensures that there are no timing issues with the network traffic.

Tip: For information on working with CobraNet networks, refer to the *CobraNet Networking Guide*. For information on working with Dante networks, refer to the *Dante Networking Guide*.

NWare Kiosk

NWare:Kiosk or just *Kiosk* is an NWare client that allows a user to make adjustments to some control values in an audio system, such as volume levels or preset recall, but does not allow any changes to the design of the system. Kiosk is typically used by end users in an installation, once the designer has completed the project.

For more information, see *Running NWare in Kiosk mode* (on page 205).

Third party control systems

Projects that you deploy to NIONs can be controlled by third party systems, as well as NWare and Kiosk.

The third party systems, which can include building management systems, for example, can interact with the MediaMatrix products using the RATC and RATC2 protocols.

Finding out more

The *MediaMatrix website* (*http://www.peaveycommercialaudio.com/applications*) is a good source of information on projects that have been designed and implemented in NWare. Installations include sports arenas, business centers, airports, and many more places.

If you attend the *MediaMatrix training courses* (*http://www.peaveycommercialaudio.com/education.cfm*), you will learn how to build up an NWare project from start to finish. This documentation will also guide you through the stages involved in creating a project, but in practice, of course, projects will have varying levels of complexity, and we recommend that you are properly trained and experienced before attempting to design or edit large scale projects.

The *Peavey Oxford forum* (*http://www.peaveyoxford.com/forum*) is an online community that discusses all things MediaMatrix. It is also a good source of information on the kinds of challenges consultants face when designing and implementing projects and a good place to provide feedback to help us improve our products in the future.

Chapter 2

Installing NWare and Kiosk

In This Chapter

Before you start

It is important that you read and understand the information in this section before you install and run NWare or Kiosk.

Examples in this section may be based on a different version of software to the one you are installing, but will still guide you through the process.

NWare minimum operating system requirements

	Windows XP	**Windows Vista (32 bit)**	**Windows 7**
Service packs	Service Pack 1 or higher	Latest service pack	Latest service pack
Processor	1GHz Pentium 4 processor	1 GHz or faster 32-bit (x86) or 64-bit (x64) processor	1 GHz or faster 32-bit (x86) or 64-bit (x64) processor
Memory	512MB RAM minimum	2GB RAM minimum	2GB RAM minimum
Networking	10/100/1000 BaseT	10/100/1000 BaseT	10/100/1000 BaseT
Disk space	170 MB plus space for application files, project files and associated media files (pictures, audio files, etc.).		

Notes:

- NWare may run on other operating systems, in addition to Windows XP, Windows Vista and Windows 7, but they are not supported or recommended.
- You can use a wireless network with NWare and Kiosk, but we recommend that you use a wired network whenever possible, especially when deploying projects. Wi-fi connections are generally not as reliable as wired connections for transferring large amounts of data.

Kiosk minimum operating system requirements

	Windows XP	**Windows Vista (32 bit)**	**Windows 7**
Service packs	Service Pack 1 or higher	Latest service pack	Latest service pack
Processor	1GHz Pentium 4 processor	1 GHz or faster 32-bit (x86) or 64-bit (x64) processor	1 GHz or faster 32-bit (x86) or 64-bit (x64) processor
Memory	512MB RAM minimum	2GB RAM minimum	2GB RAM minimum
Networking	10/100/1000 BaseT	10/100/1000 BaseT	10/100/1000 BaseT
Disk space	80 MB.		

Notes:

- Kiosk may run on other operating systems, in addition to Windows XP, Windows Vista and Windows 7, but they are not supported or recommended.
- You can use a wireless network with NWare and Kiosk, but we recommend that you use a wired network whenever possible, especially when deploying projects. Wi-fi connections are generally not as reliable as wired connections for transferring large amounts of data.

User privileges

Actions you can perform	**Windows account type**	
	Limited	**Computer administrator**
Install NWare	No	Yes
Run NWare	Yes	Yes
Install Kiosk	No	Yes
Run Kiosk	Yes	Yes

Software, firmware and manual versions

The information in this manual is based on a specific software version (for PC software, like NWare) or firmware version (for hardware devices, like NION). The version of the manual, shown on the front page, includes the software or firmware version and the revision number of the manual itself.

Manual version 1.6.1f.0 (example)	
1.6.1f	Software or firmware version.
0	Manual revision number. Revision 0 is the first version to be published for a release. Revision 1,2 etc. indicate that the manual is a replacement for the previous version published for this release.

Note: To reduce the risk of compatibility problems, we recommend that all the hardware devices (NioNodes, nControl nodes etc.) on the network run the same firmware version, and that the version matches the version of NWare you are running.

Each version of NWare includes firmware for all the MediaMatrix hardware devices it supports.

You can download the latest software and earlier versions of software from *http://www.peaveycommercialaudio.com/products.cfm/Software/NWare-Software/NWare-Software* (*http://www.peaveycommercialaudio.com/products.cfm/Software/NWare-Software/NWare-Software*).

You can install more than one version of NWare or Kiosk on the same PC, but they must be in separate folders. A separate Start menu shortcut will be created for each one automatically by the installer.

Installing hardware devices

NWare can be used to design an audio system without having access to hardware devices like NIONs and CAB4ns. You can do some initial testing within NWare using *emulation mode*, before *deploying* your project to NIONs. As part of the design process you will be able to determine exactly what hardware devices are required, depending on the number of audio input and output connections and DSP processing power needed. Once the hardware devices are available on the local area network, you can test your project on them.

Tip: Information on installing the hardware devices is provided in the individual hardware manuals / installation guides.

NWare and Kiosk installers

When you want to run NWare, you must firstly install the NWare software using the NWare installer, then start NWare. However, when you want to run Kiosk, you have two options:

- Install Kiosk using the Kiosk installer, then start the Kiosk software.

- Install NWare using the NWare installer, then run NWare in a special Kiosk mode.

The Kiosk installer only installs the components required to run Kiosk and is considerably smaller than the NWare installer, so will take less time to download from the MediaMatrix website.

The NWare installer installs all the components required to run NWare, plus the components required to run Kiosk. This means the user could run NWare on the PC as well as Kiosk and have the ability to edit project files, which might present a security risk.

Upgrading software and firmware in a running MediaMatrix system

If you need to update NWare and the firmware running on MediaMatrix nodes on your network, it is important to complete the steps carefully and in the correct order. This will ensure minimum disruption to the running system.

Follow the steps in this section, referring to the linked topics if you need detailed information.

1. Install the latest version of NWare on the administrator's PC. See *Running the NWare installer* (on page 16).
2. Make sure you have access to the most recent version of the running project. If you cannot find the NWare project file, you can upload it from a node in the system where it is hosted. See *Uploading a project to NWare and connecting to it* (on page 168).
3. Update all the *out of date* devices and *obsolete* devices. See *Dealing with project compatibility issues* (on page 274).
4. Save the project using a new file name so that you still have a copy of the original.
5. Update the firmware on the hardware devices in the project. See *Updating firmware on MediaMatrix devices* (on page 280).
6. Once the hardware devices have rebooted after being upgraded, deploy the updated version of the project. See Deploying a project.

Downloading the installation files

If you have been supplied a CD or DVD containing the NWare and Kiosk installation software, you can start the installation process. Otherwise, follow the procedure below to download the installation files from the MediaMatrix website before you begin installing.

⏩ *To download the installation files*

1. Open a web browser.
2. Go to *http://www.peaveycommercialaudio.com/products.cfm/Software/NWare-Software/NWare-Software* (*http://www.peaveycommercialaudio.com/products.cfm/Software/NWare-Software/NWare-Software*).
3. Under **Software**, click the version of NWare or Kiosk you want to download and install.

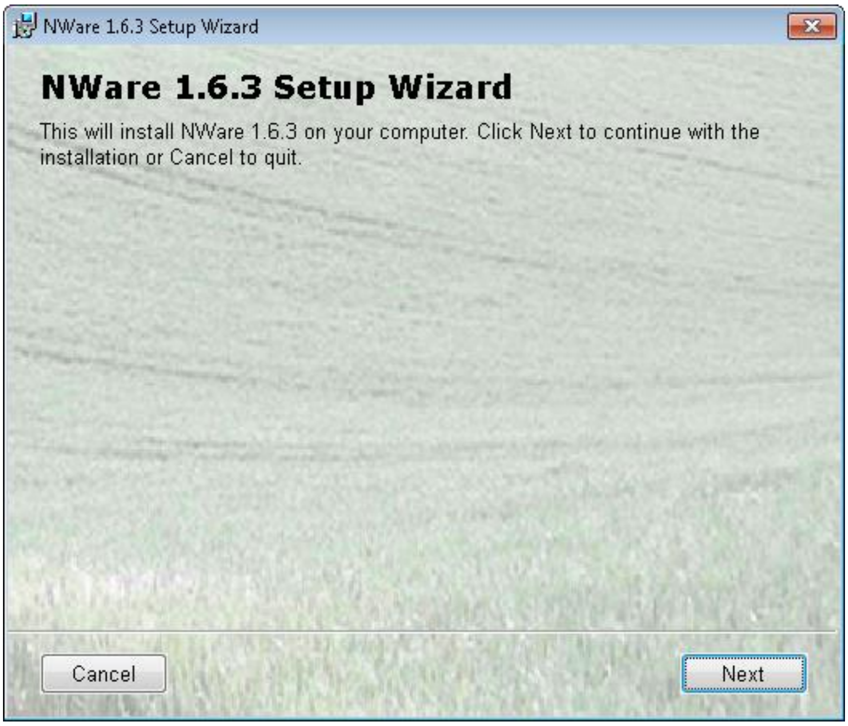
Note: NWare and Kiosk do not run on Mac (Macintosh) operating systems.

4. If a security warning is displayed, follow the onscreen instructions to continue with the download.
5. Save the installer MSI file to your PC hard disk.
6. Repeat the above steps for the other MSI files you want to download.

Running the NWare installer

1. Run the NWare installer MSI file.

   The NWare Setup Wizard is displayed.



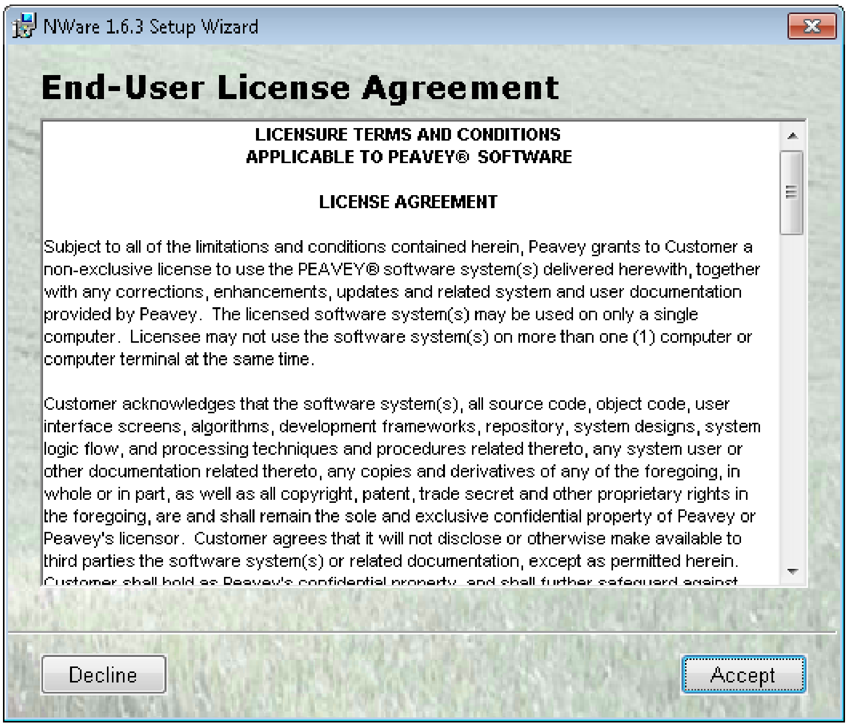
NWare 1.6.3 Setup Wizard

NWare 1.6.3 Setup Wizard

This will install NWare 1.6.3 on your computer. Click Next to continue with the installation or Cancel to quit.

2. Click **Next**.

   The **End-User License Agreement** screen is displayed.



NWare 1.6.3 Setup Wizard

End-User License Agreement

LICENSURE TERMS AND CONDITIONS
APPLICABLE TO PEAVEY® SOFTWARE

LICENSE AGREEMENT

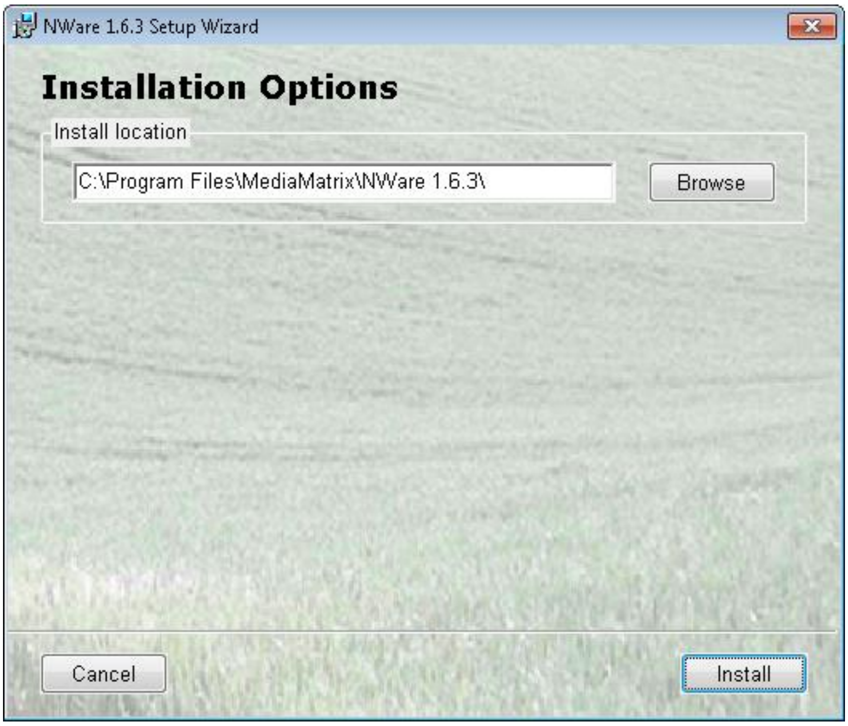
Subject to all of the limitations and conditions contained herein, Peavey grants to Customer a non-exclusive license to use the PEAVEY® software system(s) delivered herewith, together with any corrections, enhancements, updates and related system and user documentation provided by Peavey. The licensed software system(s) may be used on only a single computer. Licensee may not use the software system(s) on more than one (1) computer or computer terminal at the same time.

Customer acknowledges that the software system(s), all source code, object code, user interface screens, algorithms, development frameworks, repository, system designs, system logic flow, and processing techniques and procedures related thereto, any system user or other documentation related thereto, any copies and derivatives of any of the foregoing, in whole or in part, as well as all copyright, patent, trade secret and other proprietary rights in the foregoing, are and shall remain the sole and exclusive confidential property of Peavey or Peavey's licensor. Customer agrees that it will not disclose or otherwise make available to third parties the software system(s) or related documentation, except as permitted herein. Customer shall hold as Peavey's confidential property, and shall further safeguard against

3. Read the end-user license agreement, then click **Accept**.

If you click **Decline**, the installation process will stop.

The **Installation Options** screen is displayed.

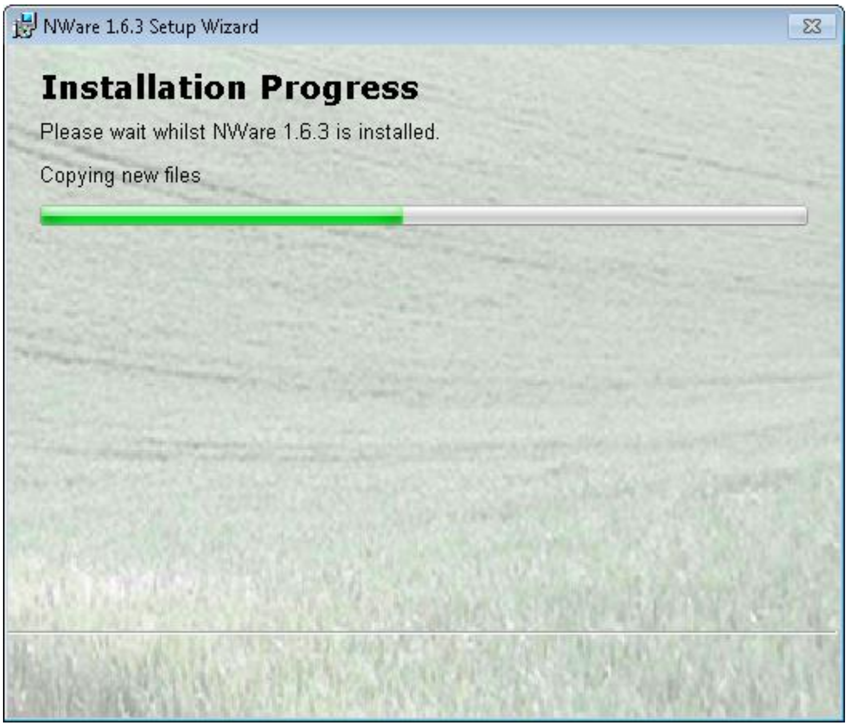


4. If you want to use the default install location, click **Install**.

   If you want to change the install location, type the new location in the **Install location** box, or click **Browse** to choose a location, then click **Install**.

Tip: We recommend that you install the software under *c:\program files\MediaMatrix\<subfolder>*. If you install a future version of NWare on the same PC, the installer will be able to automatically copy settings over from the previously installed version, but not if it is installed in a different location.

The installation process will start. As it continues, the progress will be shown.



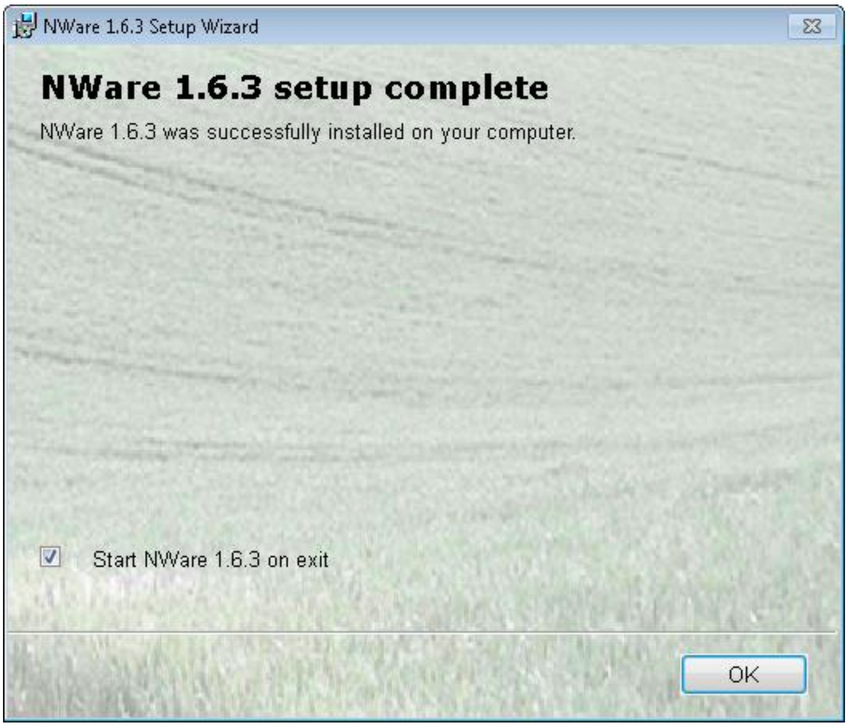
NWare 1.6.3 Setup Wizard

Installation Progress

Please wait whilst NWare 1.6.3 is installed.

Copying new files

The **NWare (version) setup complete** screen is displayed.



NWare 1.6.3 Setup Wizard

NWare 1.6.3 setup complete

NWare 1.6.3 was successfully installed on your computer.

☑ Start NWare 1.6.3 on exit

5. If you want to start NWare immediately, click **OK**.

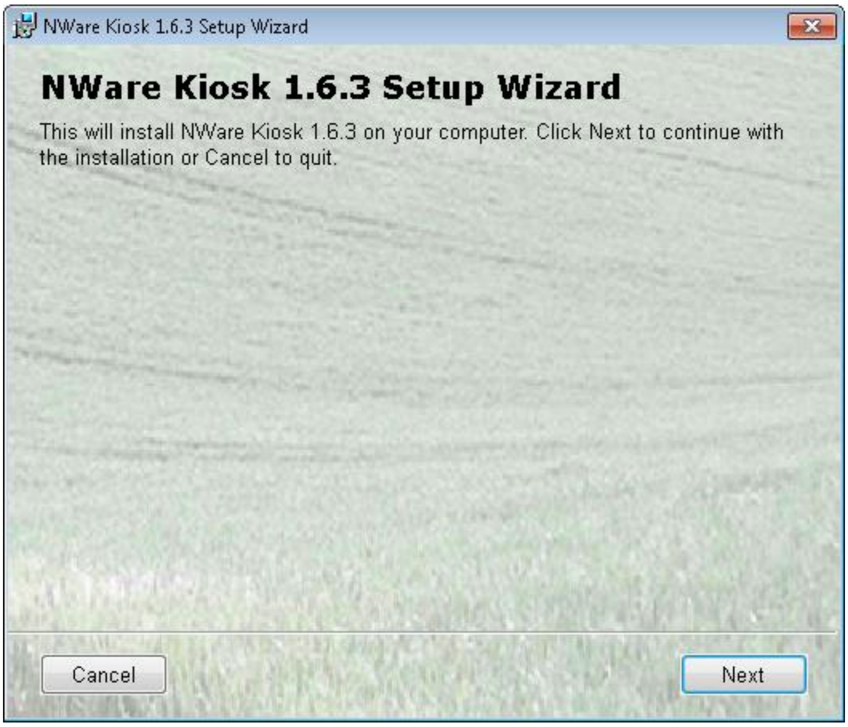
   If you want to close the NWare Setup Wizard without starting NWare, clear the **Start NWare (version) on exit** check box, then click **OK**.

Tip: When you start NWare for the first time, and a previous version of NWare is already installed on the PC, you will be asked if you want to copy settings from the previous installation to the new one. For more information, see *Starting NWare* (on page 32).

Running the Kiosk installer

1. Run the NWare Kiosk installer MSI file.

   The NWare Kiosk Setup Wizard is displayed.



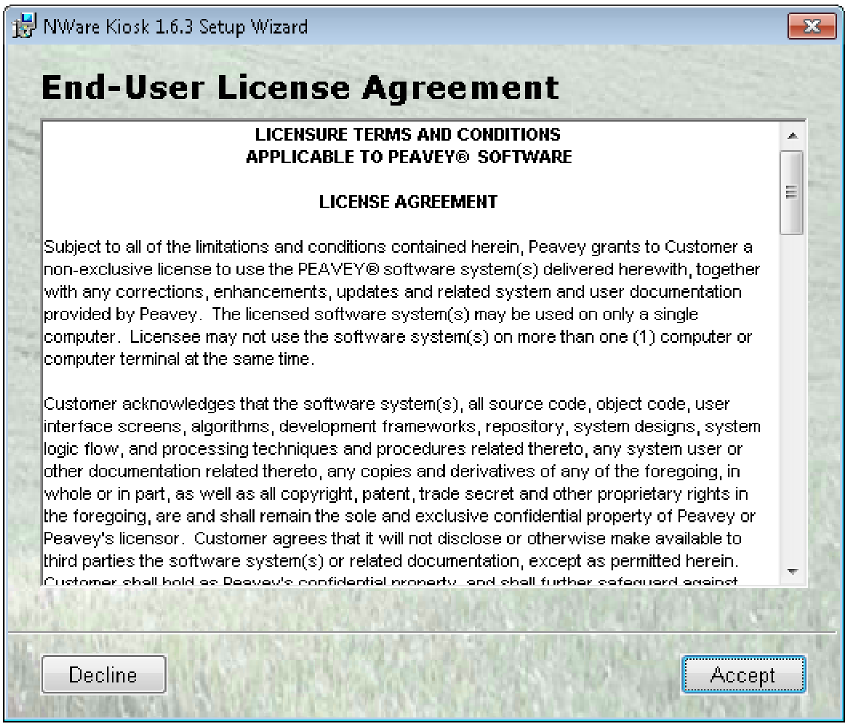
NWare Kiosk 1.6.3 Setup Wizard

NWare Kiosk 1.6.3 Setup Wizard

This will install NWare Kiosk 1.6.3 on your computer. Click Next to continue with the installation or Cancel to quit.

2. Click **Next**.

The **End-User License Agreement** screen is displayed.



NWare Kiosk 1.6.3 Setup Wizard

End-User License Agreement

LICENSURE TERMS AND CONDITIONS
APPLICABLE TO PEAVEY® SOFTWARE

LICENSE AGREEMENT

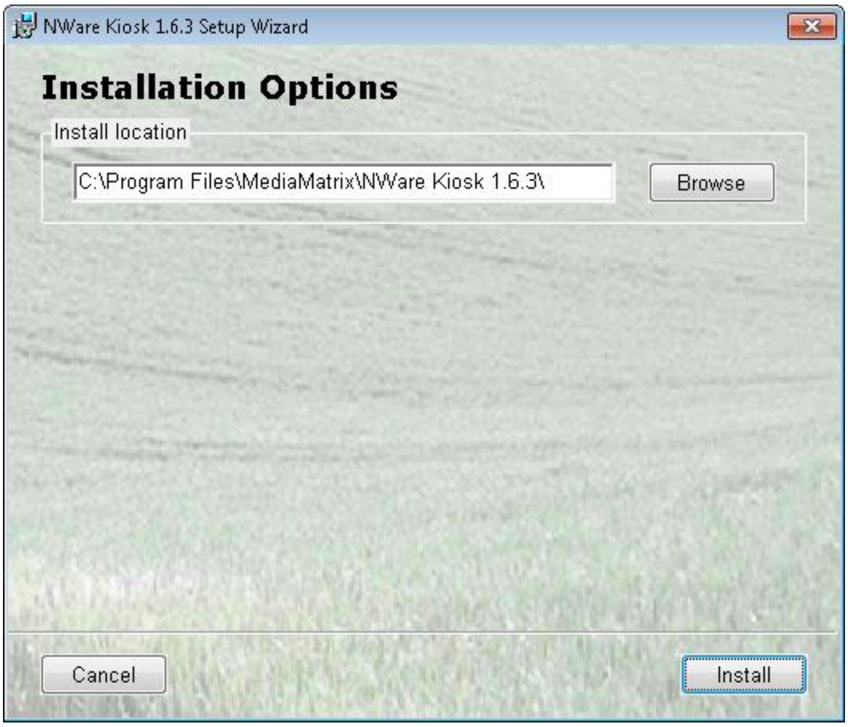
Subject to all of the limitations and conditions contained herein, Peavey grants to Customer a non-exclusive license to use the PEAVEY® software system(s) delivered herewith, together with any corrections, enhancements, updates and related system and user documentation provided by Peavey. The licensed software system(s) may be used on only a single computer. Licensee may not use the software system(s) on more than one (1) computer or computer terminal at the same time.

Customer acknowledges that the software system(s), all source code, object code, user interface screens, algorithms, development frameworks, repository, system designs, system logic flow, and processing techniques and procedures related thereto, any system user or other documentation related thereto, any copies and derivatives of any of the foregoing, in whole or in part, as well as all copyright, patent, trade secret and other proprietary rights in the foregoing, are and shall remain the sole and exclusive confidential property of Peavey or Peavey's licensor. Customer agrees that it will not disclose or otherwise make available to third parties the software system(s) or related documentation, except as permitted herein. Customer shall hold as Peavey's confidential property, and shall further safeguard against

3. Read the end-user license agreement, then click **Accept**.

   If you click **Decline**, the installation process will stop.

   The **Installation Options** screen is displayed.



NWare Kiosk 1.6.3 Setup Wizard

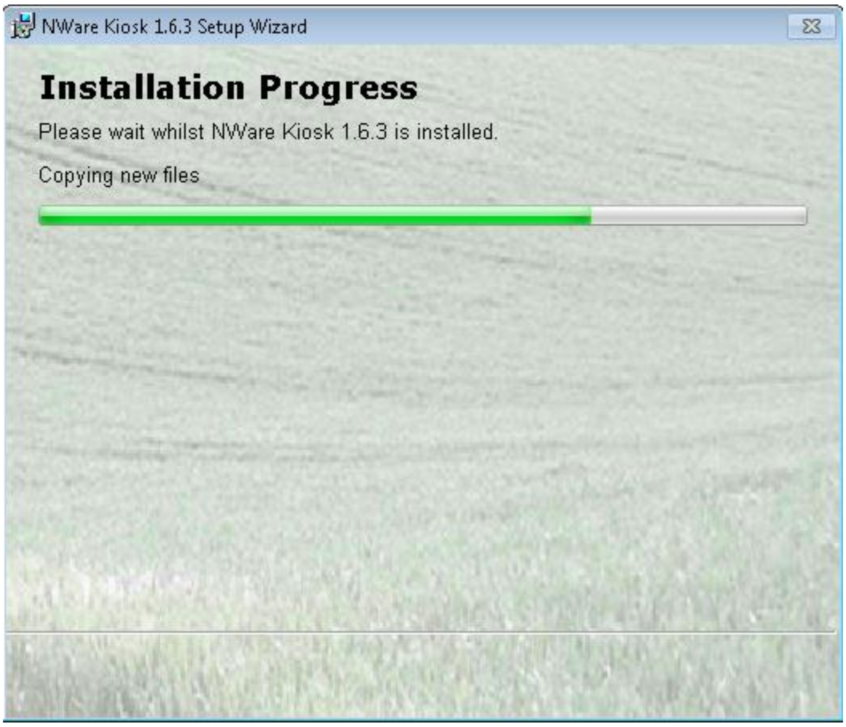
Installation Options

Install location: C:\Program Files\MediaMatrix\NWare Kiosk 1.6.3\

4. If you want to use the default install location, click **Install**.

   If you want to change the install location, type the new location in the **Install location** box, or click **Browse** to choose a location, then click **Install**.

The installation process will start. As it continues, the progress will be shown.



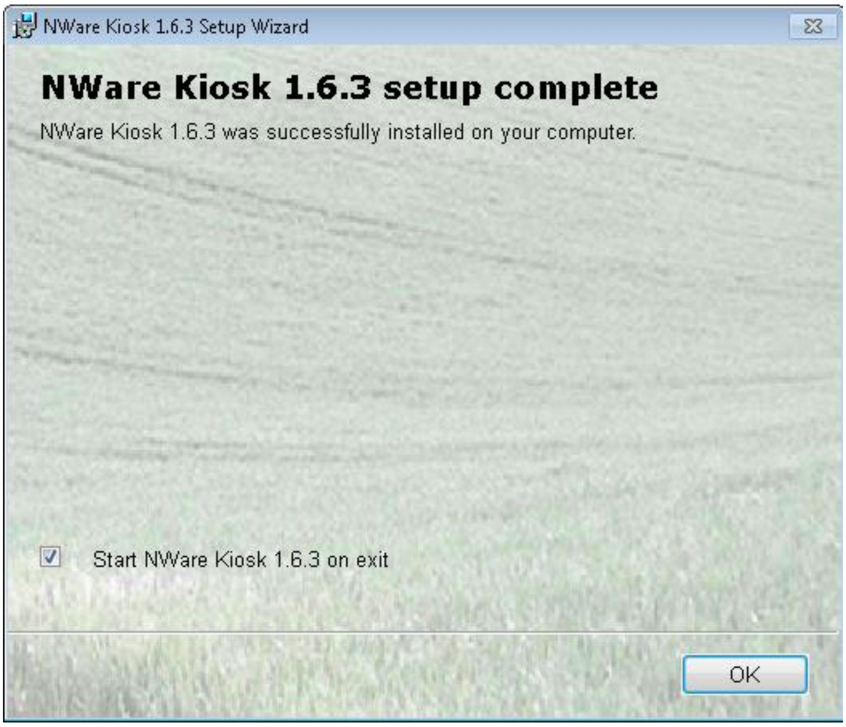
NWare Kiosk 1.6.3 Setup Wizard

Installation Progress

Please wait whilst NWare Kiosk 1.6.3 is installed.

Copying new files

The **NWare Kiosk (version) setup complete** screen is displayed.



NWare Kiosk 1.6.3 Setup Wizard

NWare Kiosk 1.6.3 setup complete

NWare Kiosk 1.6.3 was successfully installed on your computer.

☑ Start NWare Kiosk 1.6.3 on exit

5. If you want to start Kiosk immediately, click **OK**.

   If you want to close the NWare Kiosk Setup Wizard without starting Kiosk, clear the **Start NWare Kiosk (version) on exit** check box, then click **OK**.

Tip: You can customize the start up process for Kiosk, so the user is logged on automatically, a particular project is loaded, and more. For more information, see *Running NWare in Kiosk mode* (on page 205).

Copying settings from a previous installation of NWare

Normally, project files, plugins, user devices, firmware, scripts and preferences from a previously installed version of NWare are automatically copied over to a newly installed version when it is first run. However, it is not always possible for the copy process to be performed automatically – for example, if the previous installation was made on a different PC, or outside the *c:\program files\MediaMatrix* folder. In these cases, you will need to copy over the files manually, before you run NWare.

Note: You will need administrator rights on both the PC you are copying settings from and the PC you are copying settings to.

⏩ *To backup the files in the new installation*

1. Navigate to `<drive>:\<new NWare folder>\plugins\nware\xml`.
2. Make backup copies of the files `user_script.xml` and `preferences.xml`.

   You can call the backup files `user_script.bak` and `preferences.bak`, for example. Creating backups allows you to restore them in the future if you need to.

⏩ *To copy settings from a previous installation of NWare*

1. If it is not possible to reach the NWare project *.npa* files from the new installation of NWare, copy them to a location that is accessible.
2. Open Windows Explorer and navigate to `<drive>:\<old NWare folder>\plugins\nware`.
3. Copy the `\user_devices` folder and the `\firmware` folder over to `<drive>:\<new NWare folder>\plugins\nware`.
4. Navigate to `<drive>:\<old NWare folder>\plugins\nware\xml`.
5. Copy the files `user_script.xml` and `preferences.xml` over to `<drive>:\<new NWare folder>\plugins\nware\xml`.

Allowing non-admin users to run NWare and Kiosk

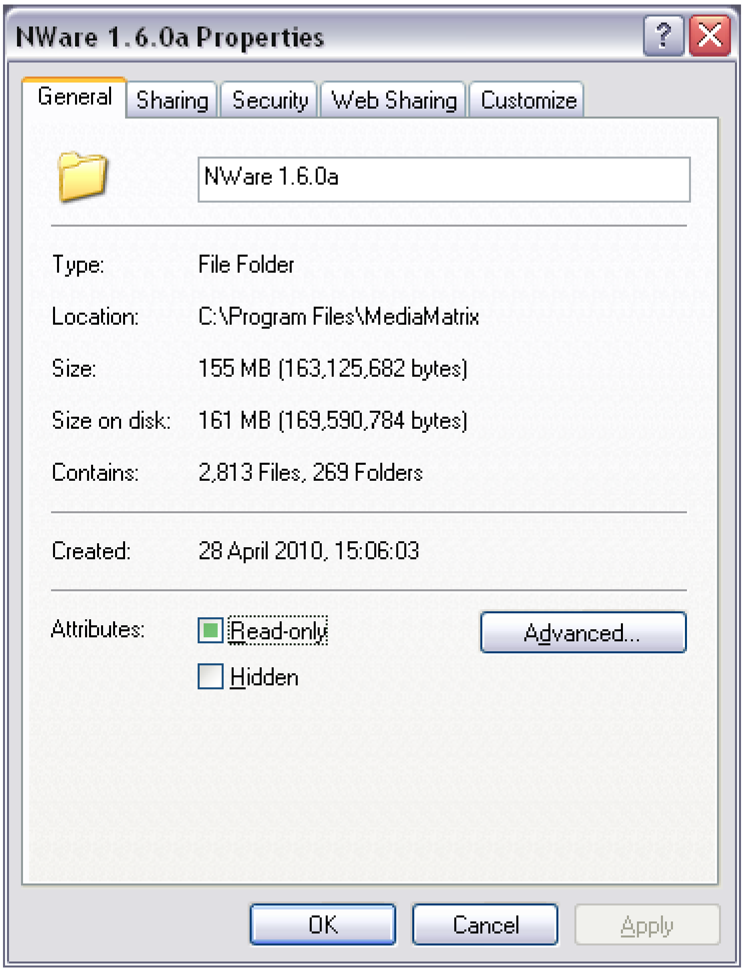
Currently, in order to run NWare or Kiosk as a non-admin user, you must complete some additional steps after the install.

Note: You need administrator rights in order to complete this procedure.

Windows XP

1. Log on to the PC as an administrator user.
2. Open Windows Explorer.
3. Browse to the *NWare <version>* folder.
4. Right-click the folder, and then click **Properties**.

   The **NWare <version> Properties** dialog box is displayed.



NWare 1.6.0a

Type: File Folder

Location: C:\Program Files\MediaMatrix

Size: 155 MB (163,125,682 bytes)

Size on disk: 161 MB (169,590,784 bytes)

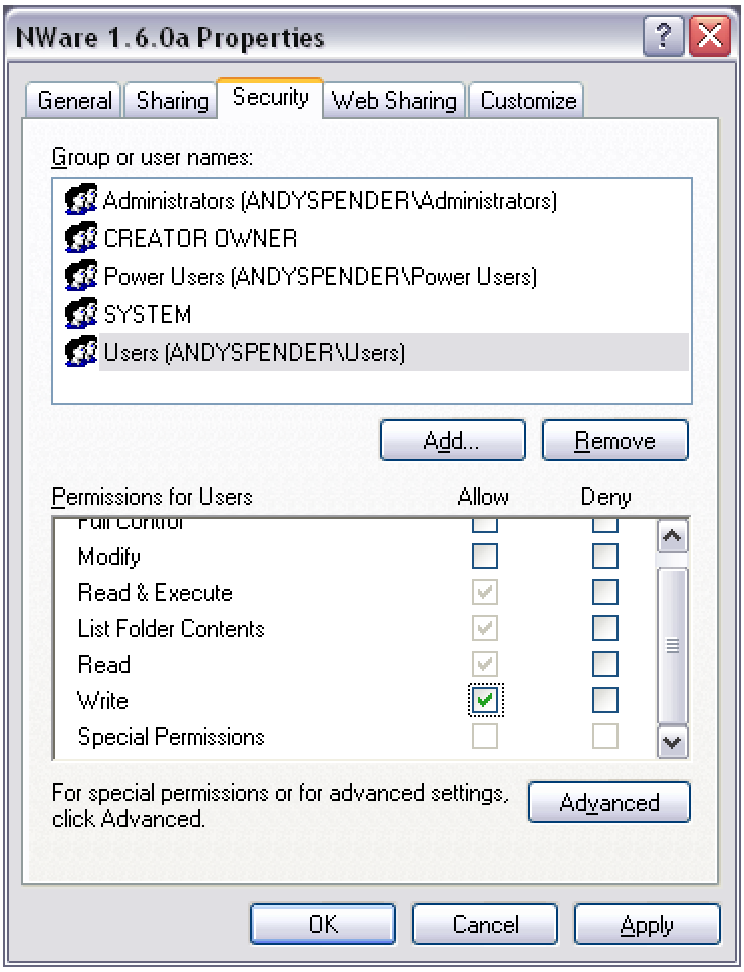
Contains: 2,813 Files, 269 Folders

Created: 28 April 2010, 15:06:03

Attributes: ▣ Read-only

☐ Hidden

5. Click the **Security** tab.



Group or user names:

- Administrators (ANDYSPENDER\Administrators)
- CREATOR OWNER
- Power Users (ANDYSPENDER\Power Users)
- SYSTEM
- Users (ANDYSPENDER\Users)

Permissions for Users	Allow	Deny
Full Control	□	□
Modify	□	□
Read & Execute	☑	□
List Folder Contents	☑	□
Read	☑	□
Write	☑	□
Special Permissions	□	□

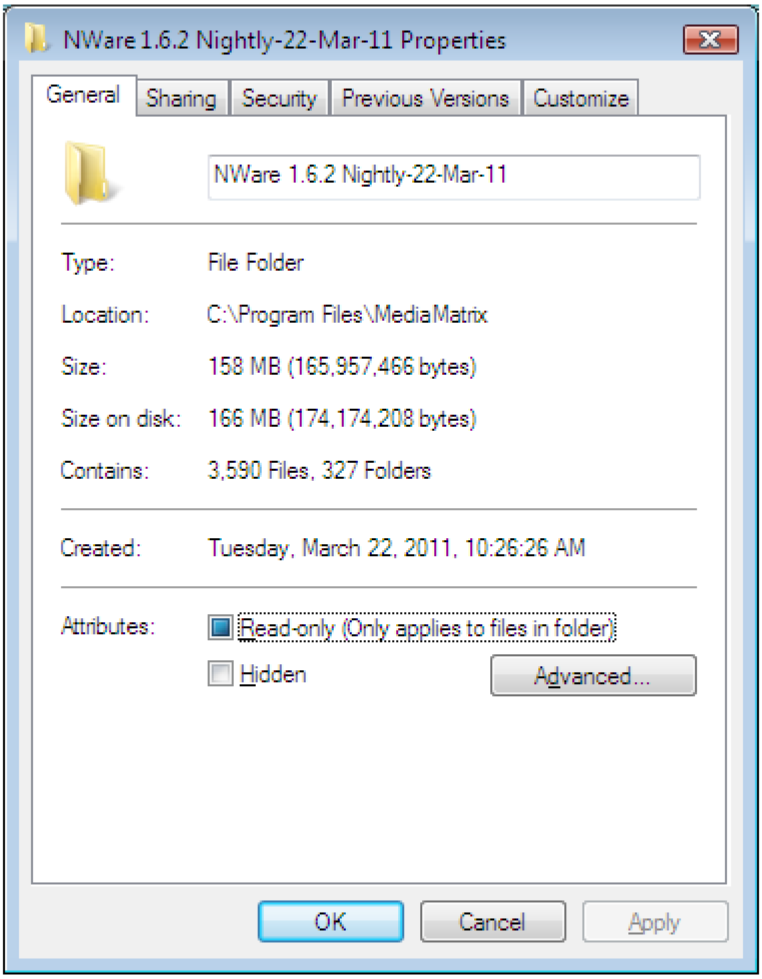
For special permissions or for advanced settings, click Advanced.

6. Under **Group or user names**, click **Users (<computer name>\Users)**.
7. Under **Permissions for Users**, next to **Write**, select the **Allow** check box.
8. Click **OK**.

Windows Vista

1. Log on to the PC as an administrator user.
2. Open Windows Explorer.
3. Browse to the *NWare <version>* folder.
4. Right-click the folder, and then click **Properties**.

The **NWare <version> Properties** dialog box is displayed.



NWare 1.6.2 Nightly-22-Mar-11

Type: File Folder

Location: C:\Program Files\MediaMatrix

Size: 158 MB (165,957,466 bytes)

Size on disk: 166 MB (174,174,208 bytes)

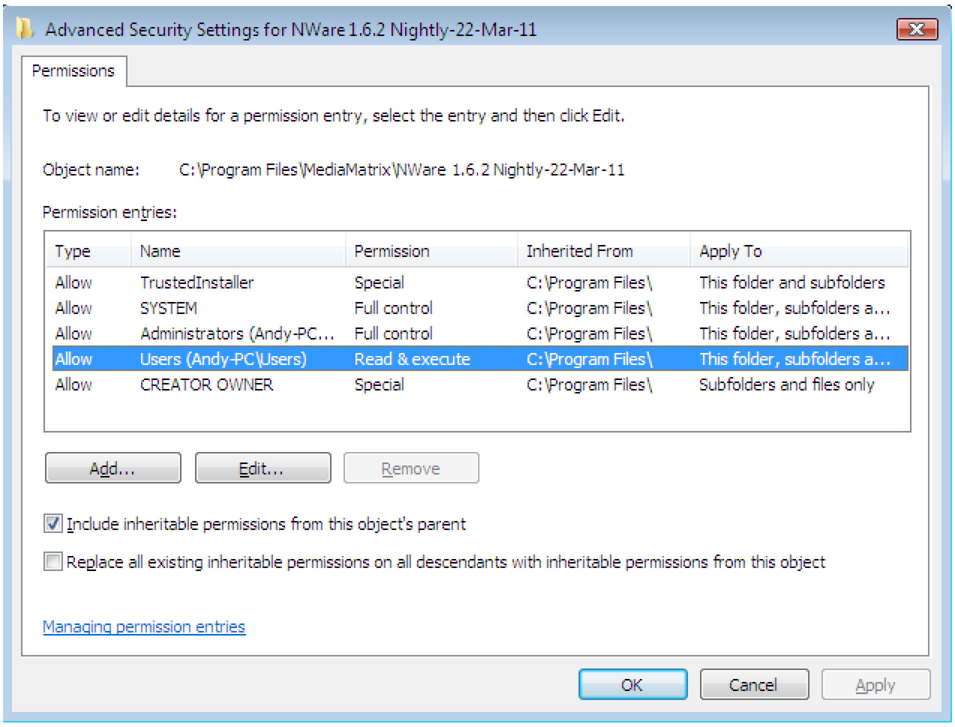
Contains: 3,590 Files, 327 Folders

Created: Tuesday, March 22, 2011, 10:26:26 AM

Attributes: ■ Read-only (Only applies to files in folder)
□ Hidden

5. Click the **Security** tab.
6. Click **Advanced**.

The **Advanced Security Settings for NWare <version>** dialog box is displayed.



Advanced Security Settings for NWare 1.6.2 Nightly-22-Mar-11

Permissions

To view or edit details for a permission entry, select the entry and then click Edit.

Object name: C:\Program Files\MediaMatrix\NWare 1.6.2 Nightly-22-Mar-11

Permission entries:

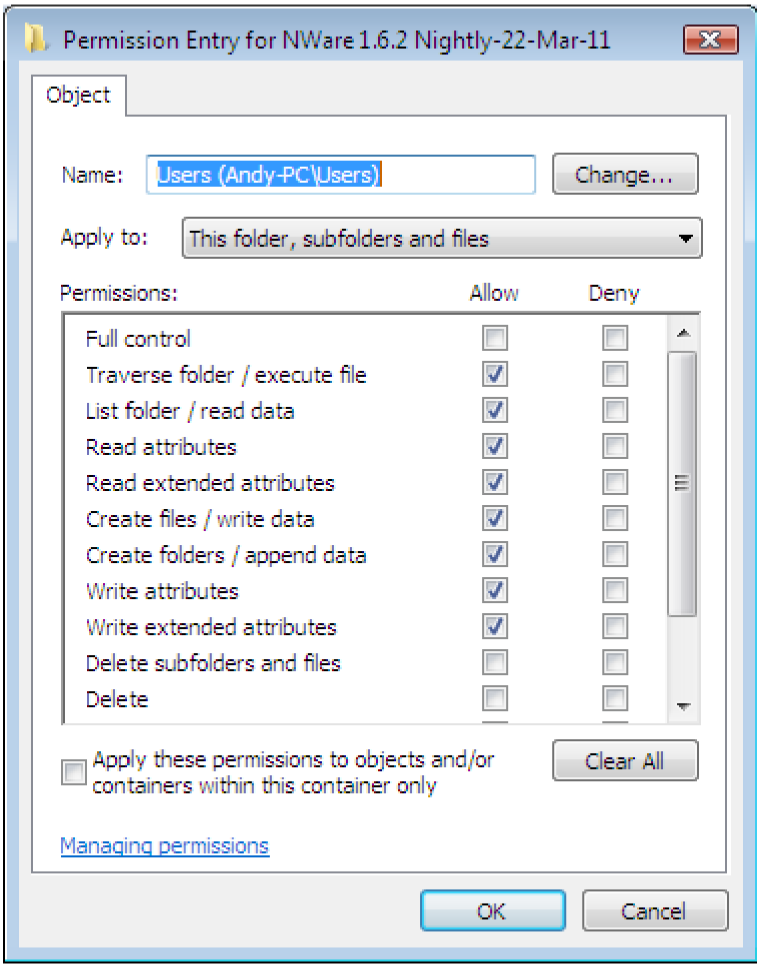
Type	Name	Permission	Inherited From	Apply To
Allow	TrustedInstaller	Special	C:\Program Files\	This folder and subfolders
Allow	SYSTEM	Full control	C:\Program Files\	This folder, subfolders a...
Allow	Administrators (Andy-PC...	Full control	C:\Program Files\	This folder, subfolders a...
Allow	Users (Andy-PC\Users)	Read & execute	C:\Program Files\	This folder, subfolders a...
Allow	CREATOR OWNER	Special	C:\Program Files\	Subfolders and files only

☑ Include inheritable permissions from this object's parent

□ Replace all existing inheritable permissions on all descendants with inheritable permissions from this object

7. Click **Edit**.
8. Click **Continue**.
9. Under **Permission entries**, click **Users (<computer name>\Users)**.
10. Click **Edit**.

The **NWare <version> Properties** dialog box is displayed.



Permission Entry for NWare 1.6.2 Nightly-22-Mar-11

Object

Name: Users (Andy-PC\Users)

Apply to: This folder, subfolders and files

Permissions:	Allow	Deny
Full control	□	□
Traverse folder / execute file	☑	□
List folder / read data	☑	□
Read attributes	☑	□
Read extended attributes	☑	□
Create files / write data	☑	□
Create folders / append data	☑	□
Write attributes	☑	□
Write extended attributes	☑	□
Delete subfolders and files	□	□
Delete	□	□

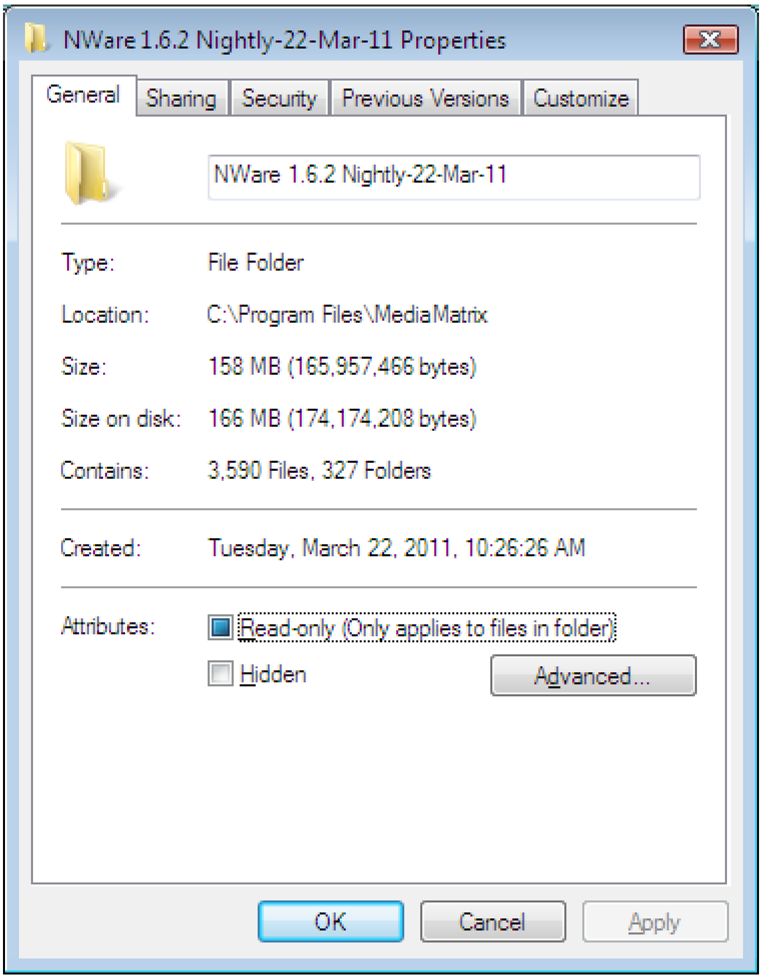
□ Apply these permissions to objects and/or containers within this container only

11. Under **Permissions**, select the **Allow** check boxes next to the following permissions:
    - Traverse folder / execute file
    - List folder / read data
    - Read attributes
    - Read extended attributes
    - Create files / write data
    - Create folders / append data
    - Write attributes
    - Write extended attributes
12. Click **OK** to close the **Permission Entry** dialog box, and then repeatedly to close the remaining dialog boxes.

Windows 7

1. Log on to the PC as an administrator user.
2. Open Windows Explorer.
3. Browse to the *NWare <version>* folder.
4. Right-click the folder, and then click **Properties**.

The **NWare <version> Properties** dialog box is displayed.



NWare 1.6.2 Nightly-22-Mar-11

Type: File Folder

Location: C:\Program Files\MediaMatrix

Size: 158 MB (165,957,466 bytes)

Size on disk: 166 MB (174,174,208 bytes)

Contains: 3,590 Files, 327 Folders

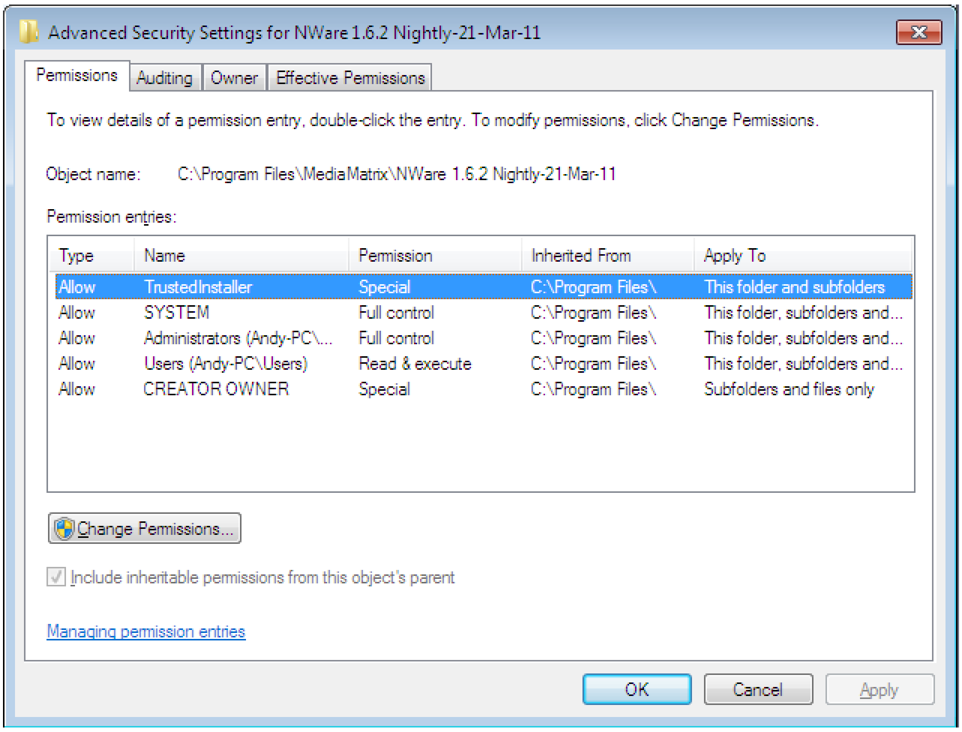
Created: Tuesday, March 22, 2011, 10:26:26 AM

Attributes: ■ Read-only (Only applies to files in folder)

□ Hidden

5. Click the **Security** tab.
6. Click **Advanced**.

The **Advanced Security Settings for NWare <version>** dialog box is displayed.



Advanced Security Settings for NWare 1.6.2 Nightly-21-Mar-11
Permissions
Auditing
Owner
Effective Permissions

To view details of a permission entry, double-click the entry. To modify permissions, click Change Permissions.

Object name: C:\Program Files\MediaMatrix\NWare 1.6.2 Nightly-21-Mar-11

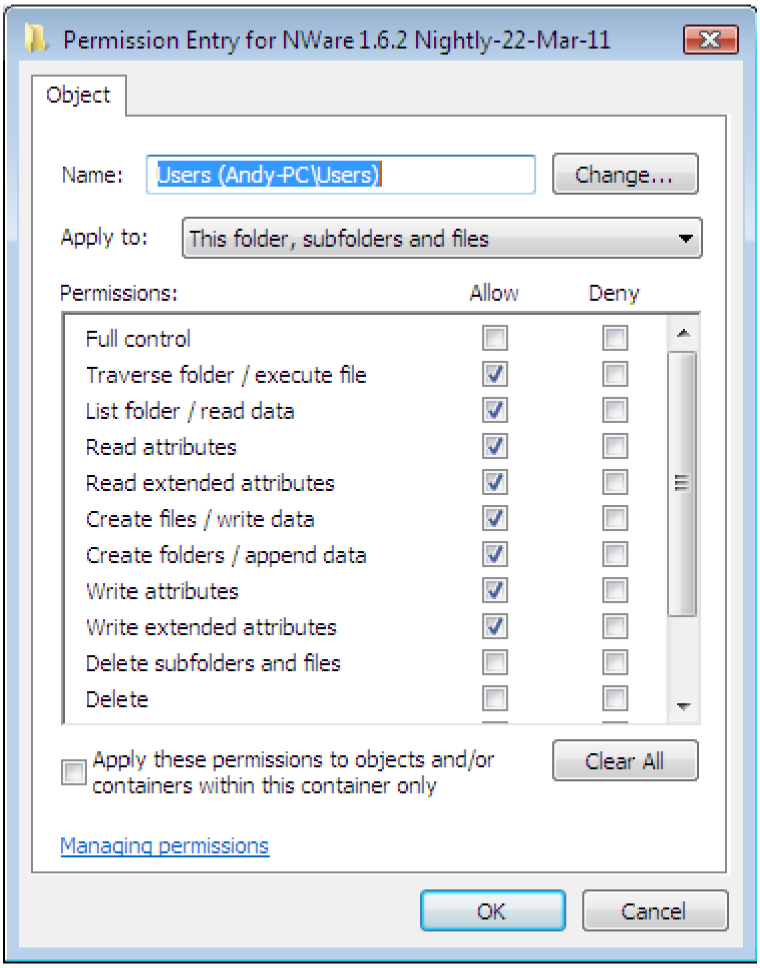
Permission entries:

Type	Name	Permission	Inherited From	Apply To
Allow	TrustedInstaller	Special	C:\Program Files\	This folder and subfolders
Allow	SYSTEM	Full control	C:\Program Files\	This folder, subfolders and...
Allow	Administrators (Andy-PC\...	Full control	C:\Program Files\	This folder, subfolders and...
Allow	Users (Andy-PC\Users)	Read & execute	C:\Program Files\	This folder, subfolders and...
Allow	CREATOR OWNER	Special	C:\Program Files\	Subfolders and files only

☑ Include inheritable permissions from this object's parent

7. Click **Change Permissions**.
8. Under **Permission entries**, click **Users (<computer name>\Users)**.
9. Click **Edit**.

The **NWare <version> Properties** dialog box is displayed.



Name: Users (Andy-PC\Users)

Apply to: This folder, subfolders and files

Permissions:	Allow	Deny
Full control	□	□
Traverse folder / execute file	☑	□
List folder / read data	☑	□
Read attributes	☑	□
Read extended attributes	☑	□
Create files / write data	☑	□
Create folders / append data	☑	□
Write attributes	☑	□
Write extended attributes	☑	□
Delete subfolders and files	□	□
Delete	□	□

□ Apply these permissions to objects and/or containers within this container only

10. Under **Permissions**, select the **Allow** check boxes next to the following permissions:
    - Traverse folder / execute file
    - List folder / read data
    - Read attributes
    - Read extended attributes
    - Create files / write data
    - Create folders / append data
    - Write attributes
    - Write extended attributes
11. Click **OK** to close the **Permission Entry** dialog box, and then repeatedly to close the remaining dialog boxes.

Chapter 3

Getting started

In This Chapter

Starting NWare

When you start NWare for the first time, and a previous version of NWare is already installed on the PC, you will be asked if you want to copy settings from the previous installation to the new one. The settings include project files, plugins, user devices, firmware, scripts and preferences.

Note: NWare will not prompt you to copy settings if the previous version of NWare was installed outside the *c:\program files\MediaMatrix* folder. If you have used a different folder, you will need to copy the settings over manually. For more information, see *Copying settings from a previous installation of NWare* (on page 22) in the webhelp or Word format versions of the NWare User Guide.

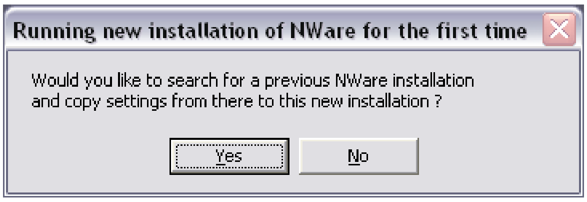
To start NWare

- On the **Start** menu, point to **All Programs**. Point to **NWare**, and then click **NWare <version>**.

  If you are starting NWare for the first time:

- If a **Windows Security Alert** dialog box is displayed, click **Unblock**.

  The **Running new installation of NWare for the first time** dialog box is displayed.



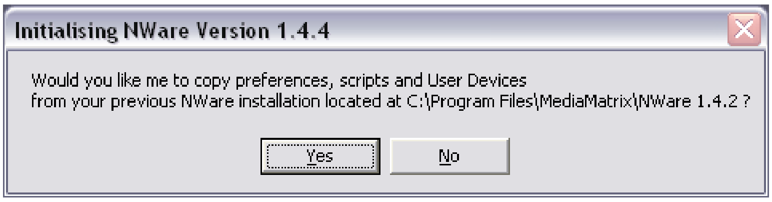
Running new installation of NWare for the first time

Would you like to search for a previous NWare installation
and copy settings from there to this new installation ?

If you want to copy settings used with a previous version of NWare over to the new installation:

a) Click **Yes**.

The **Initializing NWare Version (version)** dialog box is displayed.

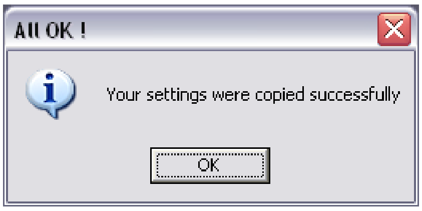


Initialising NWare Version 1.4.4

Would you like me to copy preferences, scripts and User Devices
from your previous NWare installation located at C:\Program Files\MediaMatrix\NWare 1.4.2 ?

b) Click **Yes** to copy the settings over to the new installation.

When the copying process has finished the **All OK !** dialog box is displayed.



All OK !

Your settings were copied successfully

c) Click **OK**.

If the PC is fitted with more than one network card:

If the PC running NWare is fitted with more than one network card, you can choose which card to use when communicating with the network. This is useful if, for example, your PC is connected to two separate networks via two separate network cards or via a network card and a wireless network interface.

When you start NWare, you will be asked which card you want to use. Once you choose a card, the setting is used each time you start the program. You will only be asked to choose a card at a later time if the IP address of any of the cards has changed.

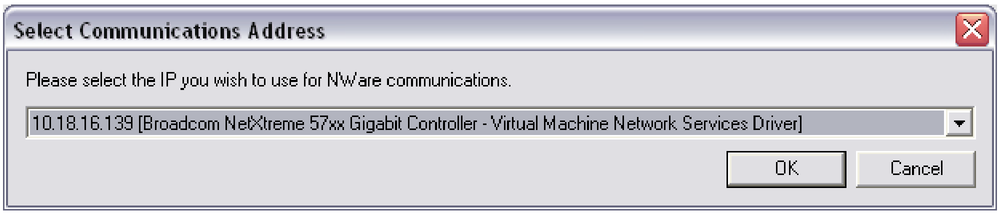
Notes:

- If you start using a new IP address to connect to a network, or disconnect and then reconnect to the network, you may need to restart Pandad. For more information, refer to the section *Recovering Pandad after it has stopped functioning* in the document *Pandad Administrator Guide*.
- You can use a wireless network with NWare and Kiosk, but we recommend that you use a wired network whenever possible, especially when deploying projects. Wi-fi connections are generally not as reliable as wired connections for transferring large amounts of data.

⏩ *To choose a PC network card*

1. If the **Select Communications Address** dialog box is not displayed, on the **Tools** menu, click **Communication Preferences**.

   The **Select Communications Address** dialog box is displayed.



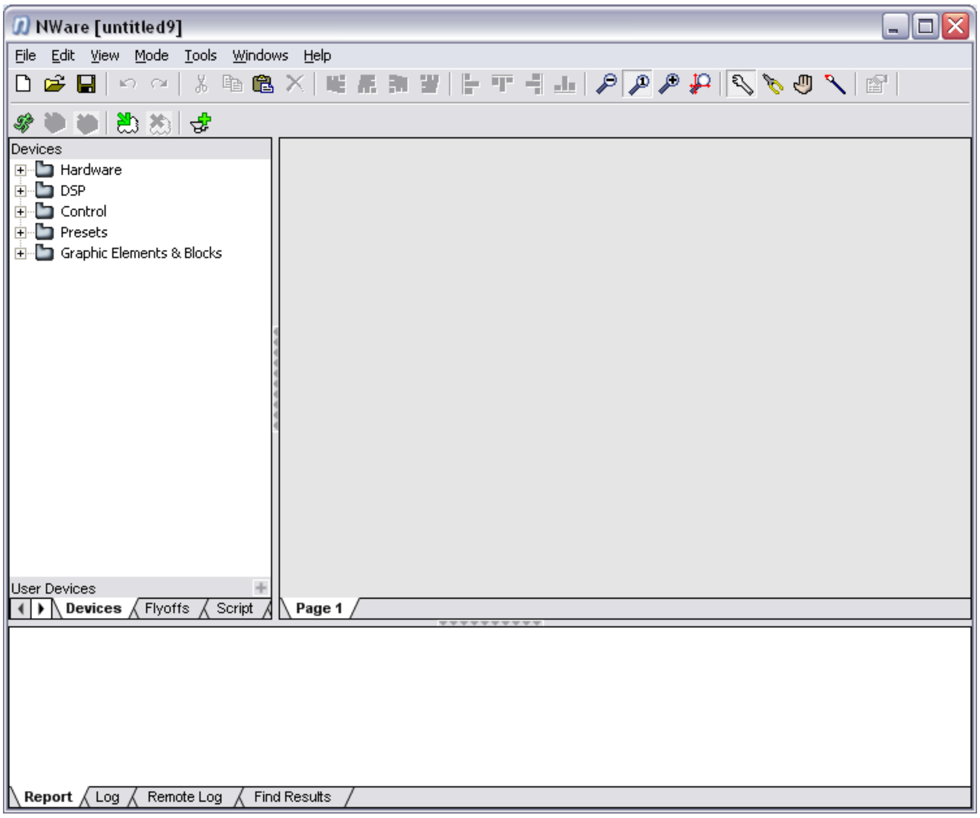
Select Communications Address

Please select the IP you wish to use for NWare communications.

10.18.16.139 [Broadcom NetXtreme 57xx Gigabit Controller - Virtual Machine Network Services Driver]

2. In the list, click the details of the network card you want to use.
3. Click **OK**.

The NWare application is displayed showing a blank project, ready for editing.



Tip: If you would like the NWare window to be maximized when the application is started, see *Starting NWare in full screen mode* (on page 272).

Loading a project

The procedure below explains how to load a project from a mapped drive listed in Windows Explorer, but you can also load a project from a node to which it has been deployed. For more information, see *Uploading a project to NWare and connecting to it* (on page 168).

Note: If you load a project that has been created using a different version of NWare, it is important to understand the compatibility issues that may arise with the different devices in the project. For more information, see *Dealing with project compatibility issues* (on page 274).

To load a project

1. On the toolbar, click the **Open** button.
2. Select an NWare Project Archive file (.npa) from the **Open** dialog box.
3. Click **Open**.

   The project is loaded into a new NWare window.

Tip: You can also load projects by opening the **File** menu, pointing to **Recent**, and then making a selection.

Saving a project

The procedure below explains how to save a project (.npa file) to a mapped drive listed in Windows Explorer, but you can also save a project to a node when you deploy it. It can then be loaded from the node for editing later on. For more information, see Deploying a project.

Tip: When you save a project file, details about the version of NWare used are saved with the project. When you subsequently load the project file, the version details are displayed on the *Log* tab.

⏩ *To save a project*

1. On the toolbar, click the **Save** button 💾.
2. If you have not already specified a file name:

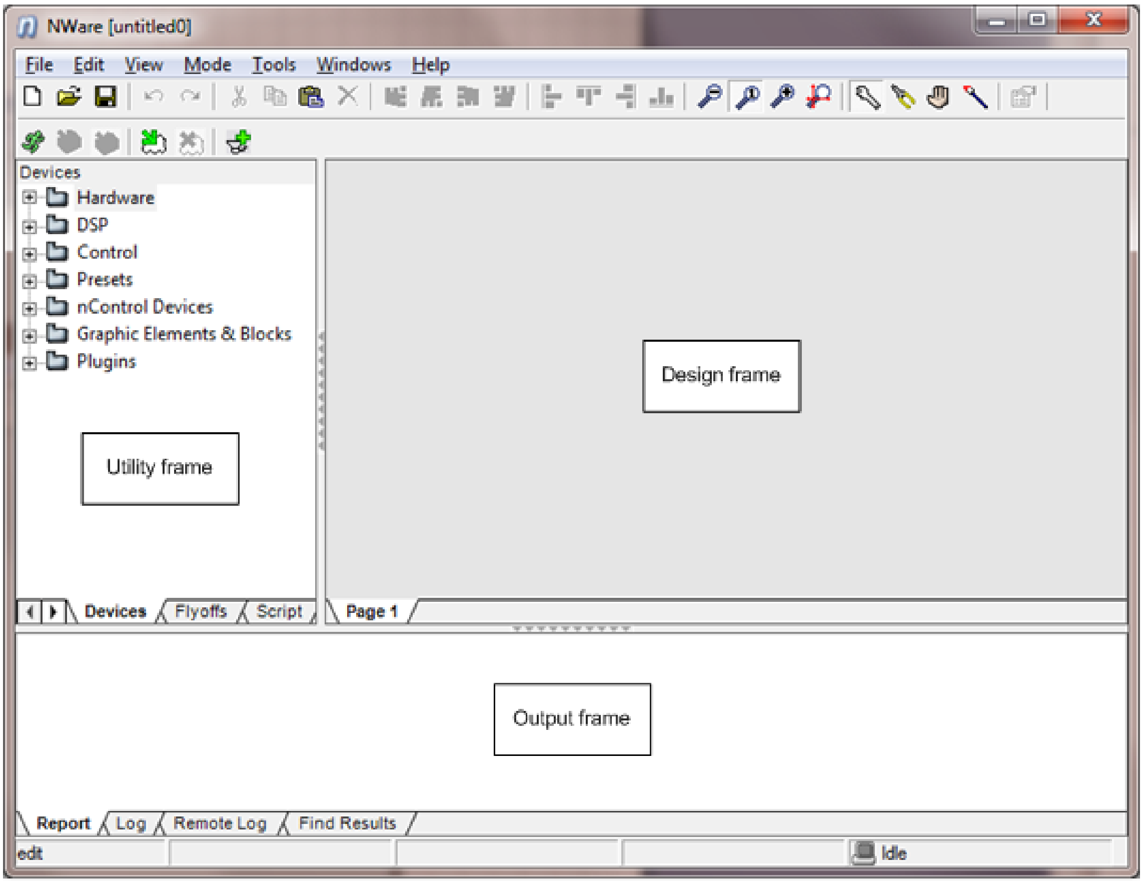
   The **Save As** dialog box will be displayed.

   a) Select or type an NWare Project Archive file (.npa).

   b) Click **Save**.

The NWare window

The different parts of the NWare window are indicated below.



Tip: You can show or hide the utility frame by pressing F11. You can show or hide the output frame by pressing F10.

Tip: You can reorder the tabs in the Utility frame (Devices tab, Flyoffs tab, and so on) by dragging with the mouse.

Project designs are created by dragging devices from the Devices tab in the utility frame over to pages in the design frame, and then wiring the devices together. The Output frame at the bottom of the window displays status information organized onto different tabs.

The device tree

Some of the devices in the device tree represent hardware elements of the target system, like NIONs; others represent DSP-based signal processing algorithms and others represent control functions.

Tip: To filter the contents of the device tree, type in the *Filter* box above the device tree and click *Apply*. NWare will display only devices with names that match the search phrase. Click *Clear* to clear the filter.

Tip: To expand or collapse the entire device tree, right-click the device tree, and then click *Expand All* or *Collapse All*.

Hardware

MediaMatrix hardware devices, including NIONs, nControl units and CABs, and third party hardware, including JL Cooper devices.

DSP

A large number of Digital Signal Processing (DSP) devices you can include in your design and wire to other devices. These include filters, white noise generators, level meters and many more.

Control

A number of different types of devices for working with ControlMatrix, logic gates (AND, OR, NOT etc.), Python Script, and more.

Under *Knobs, Faders, Etc.*, you can find *Generic Controls*. You can use this object to create several different kinds of controls, including knobs, faders, meters etc. Drag *Generic Controls* over to the design page and click *OK* to create the control. Open the new Controls block, then edit the control properties of the knob control inside. On the *Style* tab, select a new control style to create a fader, meter etc.

Presets

Devices for capturing and saving the values of controls in your design. Each preset device can save and restore several sets of control values in datasets.

nControl Devices

The **nControl Devices** part of the device tree contains devices for use exclusively with nControl nodes and nTouch 180 nodes.

Note: These devices cannot be used with NioNodes.

The majority of the devices are SNMP based and return status information about different types of hardware devices on the network, including switches and UPS devices.

Graphic Elements & Blocks

Objects for enhancing the graphics in your design, including:

- bump panels (buttons in up and down states)
- labels of various sizes.

Blocks that can contain groups of objects, including:

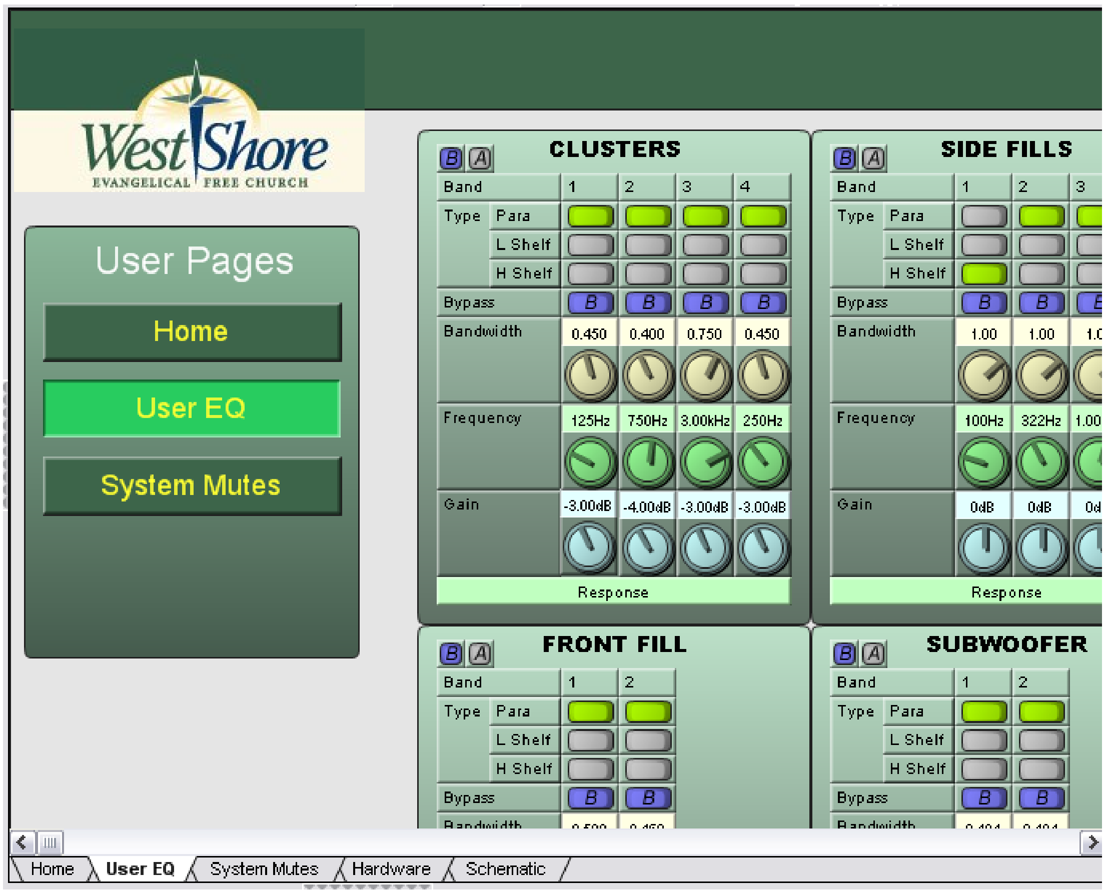
- Kiosk screens with predefined sizes.
- DSP devices and control devices.

Plugins

Plugins store collections of devices from your designs. By creating a series of plugins you can reuse and share custom functionality with other users.

The design frame

The design frame contains all the individual pages in your design. The pages in turn store all the components that the design is going to use; these might include a *NION n3* and a *CAB 8n* on a page called *Hardware*, for example, and controls for adjusting the gain and equalizer settings on a page called *User EQ*.



Pages are edited within NWare, but once the project has been deployed or *emulated* (on page 167), you can interact with it either from NWare or using Kiosk. The NWare designer has control over which pages can be viewed and edited by other NWare designers, and which pages can be viewed in Kiosk.

Zooming and panning the display

You can use the zoom tool to control the magnification of the design area, and the pan tool to move the design area. These tools are handy when you are working on a large design.

Tip: You can split the design across several pages to make the display easier to work with.

To zoom the display in and out

- On the toolbar, click the **Zoom In** button or the **Zoom Out** button .

Tip: If your mouse is fitted with an IntelliMouse wheel, you can hold down CTRL and rotate the mouse wheel backwards and forwards to zoom in and out on the area where the mouse pointer is located.

To return the display to 100% magnification, click the **Zoom Normal** button .
To shrink all the devices to fit the design area, press CTRL+0 (zero).

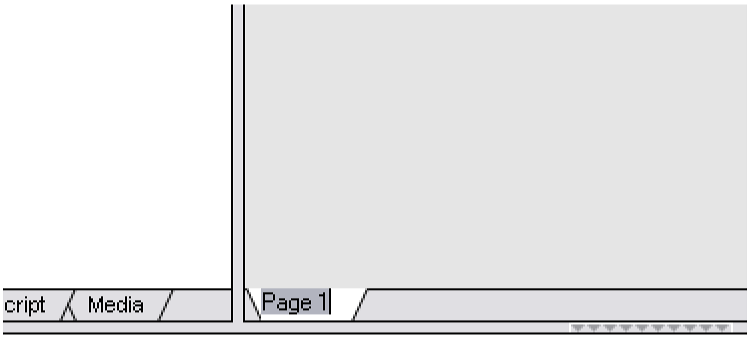
Note: The zero key on the numeric keypad cannot be used to select this function.

To pan the display

1. Hold down the space bar. The cursor will change to ✥.
2. Drag the mouse.

Renaming a page

1. At the bottom of the design frame, double-click the page name to highlight it.



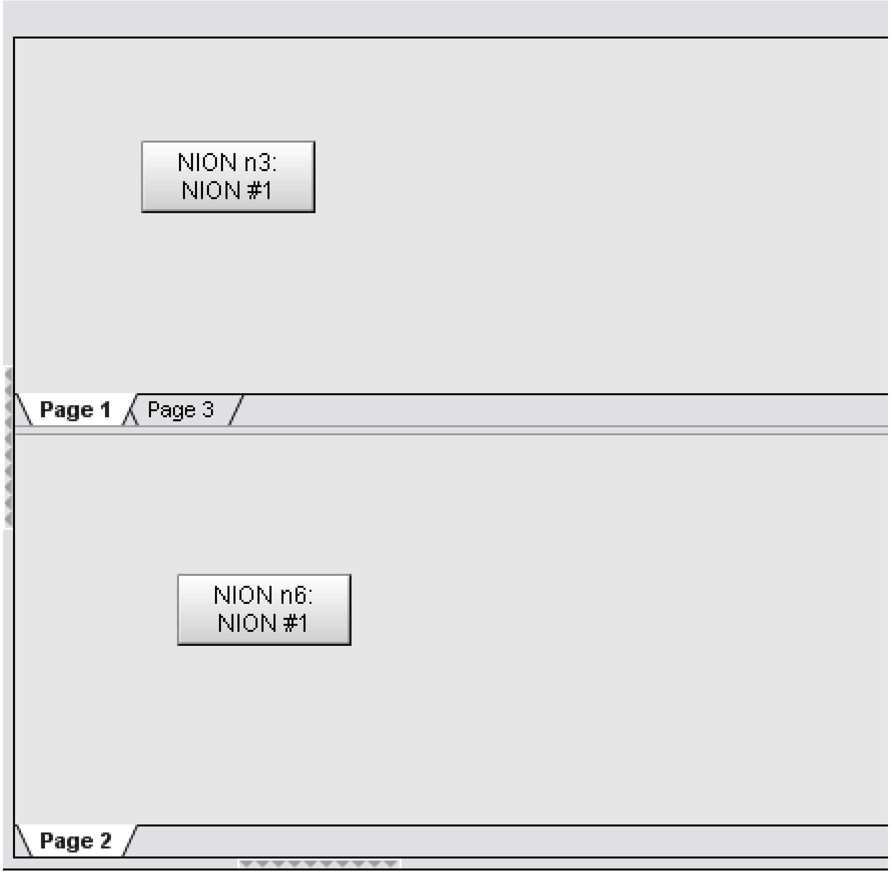
2. Type the new name for the page and press Enter.

Displaying two pages simultaneously

NWare can display two design pages simultaneously, allowing you to compare the pages and copy and paste components from one page to another.

⏩ *To display two pages simultaneously*

- Right-click the page tab at the bottom of the page, and then click **Move to lower pane**.



You can remove the split from the display by moving all the pages to the lower pane or by moving all the pages to the upper pane using the **Move to upper pane** command.

Hiding pages from the display

You can hide a page so that it is no longer visible in the NWare window. It can be displayed again at any time and no password is required.

⏩ *To hide a page from the display*

- Right-click the page tab at the bottom of the page, and then click **Hide Page x**.

Controlling access to pages in the design frame

You can prevent NWare users from viewing and editing pages in your project by protecting them with a password.

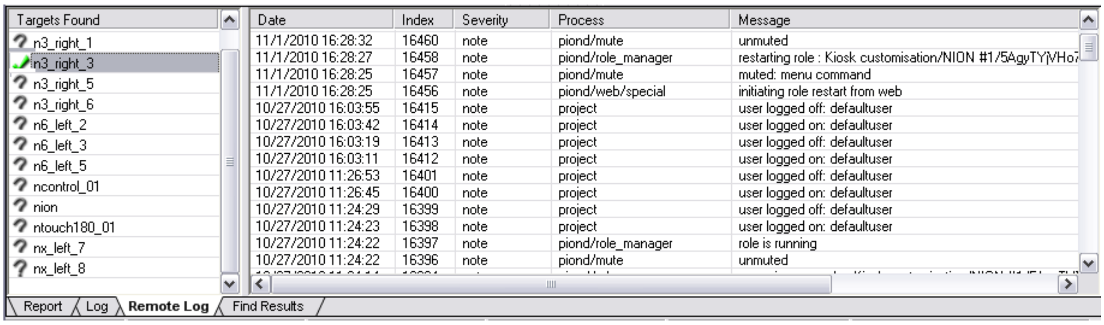
Users may have access to your project file on your PC or network, and they may also be able to access the file from the MediaMatrix device hosting the project if the *Save Project to Target* option was selected during deployment. If a project with no password protection on its pages has been saved on a node, someone could download NWare from the MediaMatrix website, then use the *Upload and Connect* (on page 168) option on the File menu to load the project on their PC and edit it. It could then be redeployed.

Notes:

- The password controls access to pages within NWare, not Kiosk. Page access in Kiosk is controlled by exporting pages to a page group and then allowing users access to that group. For more information, see *Exporting pages from an NWare project* (on page 208).
- Do not lock pages within the device or the page on which the device is located in your design. You will not be able to access the locked pages from the iPhone or iPad.

The Remote Log tab

The Remote Log tab at the bottom of the NWare window displays a list of MediaMatrix nodes that NWare has discovered on the network. If logging is enabled, you can view log messages for each node, as shown in the example below.



Targets Found
n3_right_1
n3_right_3
n3_right_5
n3_right_6
n6_left_2
n6_left_3
n6_left_5
ncontrol_01
nion
ntouch180_01
nx_left_7
nx_left_8

Date	Index	Severity	Process	Message
11/1/2010 16:28:32	16460	note	piond/mute	unmuted
11/1/2010 16:28:27	16458	note	piond/role_manager	restarting role : Kiosk customisation/NION #1/5AgyTYjVHo7
11/1/2010 16:28:25	16457	note	piond/mute	muted: menu command
11/1/2010 16:28:25	16456	note	piond/web/special	initiating role restart from web
10/27/2010 16:03:55	16415	note	project	user logged off: defaultuser
10/27/2010 16:03:42	16414	note	project	user logged on: defaultuser
10/27/2010 16:03:19	16413	note	project	user logged off: defaultuser
10/27/2010 16:03:11	16412	note	project	user logged on: defaultuser
10/27/2010 11:26:53	16401	note	project	user logged off: defaultuser
10/27/2010 11:26:45	16400	note	project	user logged on: defaultuser
10/27/2010 11:24:29	16399	note	project	user logged off: defaultuser
10/27/2010 11:24:23	16398	note	project	user logged on: defaultuser
10/27/2010 11:24:22	16397	note	piond/role_manager	role is running
10/27/2010 11:24:22	16396	note	piond/mute	unmuted

For information on enabling logging, see *Displaying log messages for a node* (on page 284).

Tip: The names of the nodes shown in the Targets Found list can be specified using the web interfaces for the nodes or front panel interfaces, if available. The names are also displayed in Dante Controller, if you are using a Dante network. For more information, refer to the hardware manual for the type of node in each case.

Working with devices

Devices are positioned and configured while in *Edit* mode, and are wired together in *Wire mode* (on page 135). You can change the color of a device in *Paint mode* (on page 44). When NWare is connected to a project that has been deployed or *emulated* (on page 167), *Gesture* mode is selected, so you can operate the controls.

Notes:

- When you double-click a device block and display its settings, for some devices you will see properties with values, or spaces where values can be specified. You cannot change these values or type in the spaces until you have deployed the project (see Deploying a project) or emulated the project (see *Emulating a project without hardware* (on page 167)), and the cursor is displayed (showing *Gesture* mode).

- If you edit the device properties of a device, when you click OK to confirm any changes, the device will be rebuilt. This means that all the changes you have made on the control surface will be lost. This is an important consideration, especially when working with devices that allow a lot of customization. We recommend that you add a new page to the device, and then copy new or customized controls to that page. The new page will not change when you click OK.

Moving, copying and duplicating objects

NWare allows you to cut, copy and paste objects in your design, and also to duplicate them. Duplication of an object is carried out in just one step, but performs the same function as copy and paste.

When a generic control, such as a knob, is copied and pasted or duplicated, the copy is assigned the same ID as the original. This means that gesturing the copy will also gesture the original and vice versa.

The behavior is different, however, when you copy a generic control block or DSP device block. Controls inside the copied block will be assigned *new* IDs, ending the association with the original controls.

When you add controls to a control block or DSP block, and then copy and paste the block, the controls inside the copied block will operate in exactly the same way as the controls inside the original block. Controls inside blocks within the control block or DSP block will be assigned new IDs, and controls copied outside blocks will keep the original IDs.

⏩ *To move objects*

1. Click an object, or click-drag to make a selection box around the objects you want to move.
2. Press CTRL+X.
3. Press CTRL+V to choose a location to paste the objects, or CTRL+SHIFT+V to paste the objects directly under the cursor.

⏩ *To copy objects*

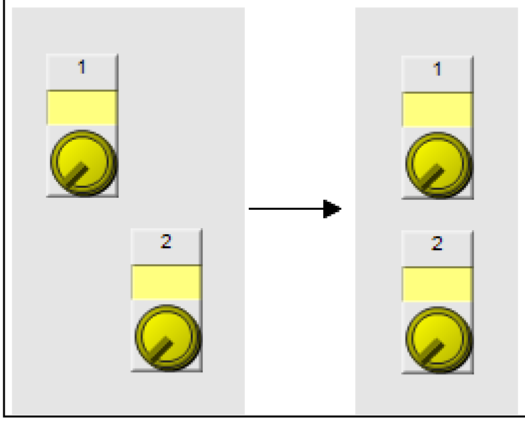
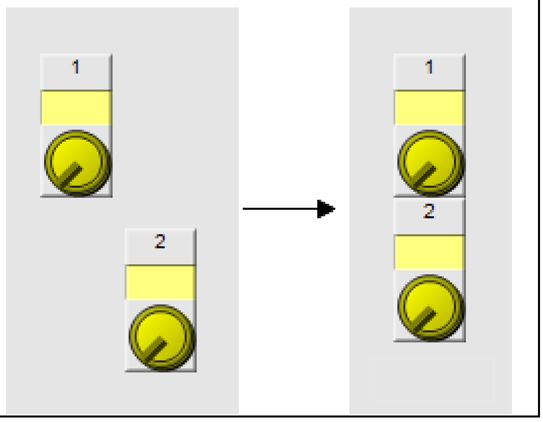
1. Click an object, or click-drag to make a selection box around the objects you want to copy.
2. Press CTRL+C.
3. Press CTRL+V to choose a location to paste the objects, or CTRL+SHIFT+V to paste the objects directly under the cursor.

⏩ *To duplicate objects*

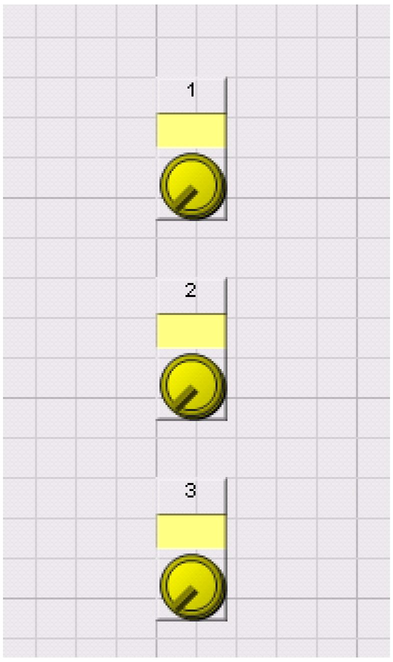
1. Click an object, or click-drag to make a selection box around the objects you want to duplicate.
2. Press CTRL+D to choose a location to place the duplicated objects, or CTRL+SHIFT+D to choose the number of copies to make, as well as the arrangement of the copies.

Aligning devices on the design page

When you align devices with each other, they are positioned on the design page so that the top, bottom, left or right edges are in the same position. When you pack devices, they are aligned, and the space between them is also removed.

Left aligned devices	Left packed devices
	

You can also use an alignment grid to help you align devices automatically, as you drag them across the design page.



Aligning devices

1. Click-drag, CTRL+click or SHIFT+click to select the objects on the design page you want to align.
2. Click a button on the toolbar to align the objects.

	Align left
	Align right

	Align top
	Align bottom

Packing devices

1. Click-drag, CTRL+click or SHIFT+click to select the objects on the design page you want to pack together.
2. Click a button on the toolbar to pack the objects.

	Pack left
	Pack right
	Pack top
	Pack bottom

Showing and hiding the alignment grid

- On the toolbar, click **Grid** Grid.

Controlling how objects snap to the grid

You can display an alignment grid on the NWare design page to help you align the objects in your design. NWare can automatically align objects to the grid using the *snap* feature.

You can choose the distance between the lines of the grid, and also control whether objects snap to the grid lines, or to points between the lines.

The width and height of the grid cells are specified using the *Snap X* and *Snap Y* settings. The *grid interval* specifies the number of snap points from one grid line to the next. If the snap interval is 1, an object snaps to each grid line as you drag it, but if the snap interval is 2, an object snaps to the mid point between each grid line, as well as to each grid line

⏩ *To switch the snap feature on or off*

1. On the toolbar, click **Snap On/Off** Snap.

⏩ *To set the grid interval*

1. On the toolbar, click **Edit grid** Edit Grid.
2. In the **Grid interval X** box, type the number of snap points from one horizontal grid line to the next.
3. In the **Grid interval Y** box, type the number of snap points from one vertical grid line to the next.

4. Click **OK**.

Setting the alignment grid origin

The grid origin is the position, defined by x and y co-ordinates, of the top left-hand corner of the grid, relative to the top left-hand corner of the design page.

If you specify 10 and 10 as the x and y co-ordinates, for example, the top left-hand corner of the grid will start 10 pixels to the right and 10 pixels down from the top left corner of the design page.

To set the alignment grid origin

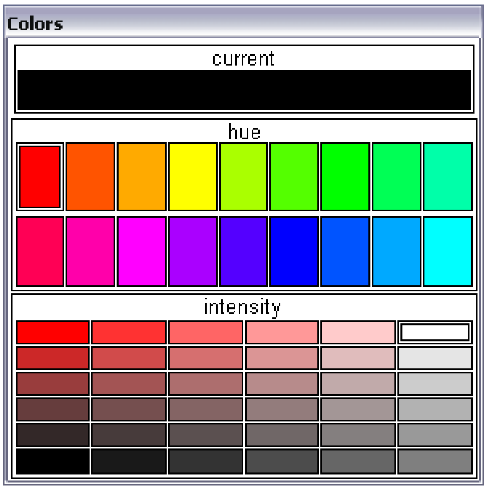
1. On the toolbar, click **Set Origin** Set Origin.
2. In the **Origin X** box, type the x co-ordinate of the center of the alignment grid.
3. In the **Origin Y** box, type the y co-ordinate of the center of the alignment grid.

Changing the color of device blocks

Choosing a color for a device block from a palette

1. Click the button to switch to Paint mode.

   The color palette is displayed.



2. Under **hue**, click a color sample.
3. Under **intensity**, click an intensity sample.
4. Click the device block you want to recolor.

Copying the color of one device block to another

1. Click the button to switch to Paint mode.
2. Hold down the SPACEBAR (the cursor will change to), and then click the block with the color you want to copy.

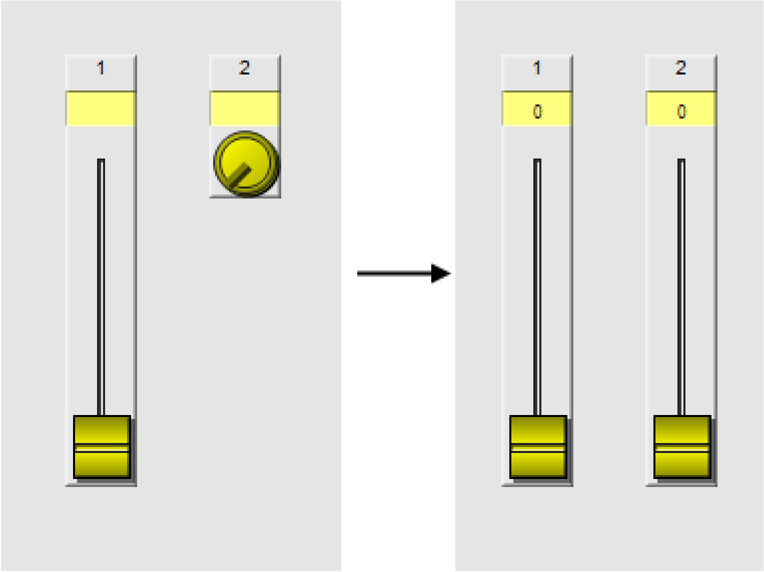
   The color palette will show the color copied from the block.
3. Click the device block you want to recolor.

Tip: If you want to copy the color from a very small or narrow object that is difficult to select, zoom the display first, by clicking the button on the toolbar.

Copying the style of a control to another control

You can copy the style settings from one control in your design to another control, to give them the same appearance and functionality.

In the example below, the style of the fader is copied to the knob control to turn it into a fader.



Note: When the control style is copied, the control that the style is copied to is not assigned the same ID as the original control. It keeps its current ID.

To copy the style of a control to another control

1. Right-click the control with the style you want to copy, and then click **Copy Control Style**.
2. Right-click the control that you want to copy the style to, and then click **Paste Control Style**.

Managing media files

The NWare Media tab is used for importing and maintaining a list of files you plan to use in your project. The following types of files are supported.

Audio files	wav, mp3
Image files	bmp, jpg/jpeg, png, gif, tif/tiff, wmf, emf, svg
Kiosk2Go help files	csv

Audio files can be played back via the NION Wave File Player and nControl Media Player.

Notes:

- NION supports wav files, but does not currently support mp3 files.
- Wav files must be recorded using a sample rate of 48kHz. Other sample rates, such as 44.1kHz are not supported.

- An *AudioScience* (*http://www.audioscience.com/internet/products/sound_cards/asi54xx.htm*) card is required when playing back audio files via an nControl.

Image files can be used to create backgrounds for pages in the design. If you are using Kiosk2Go, you can use images to customize the appearance of knobs, faders and other controls. For more information, see *Skinning controls to improve their appearance* (on page 240).

Kiosk2Go help files store help text for controls, so that you do not need to add the text directly to each control in your design. This allows the help text to be produced and managed using an external help tool. The help text is linked to the Kiosk2Go controls using help IDs. For more information, see *Adding help text using a topic file* (on page 251) in the *NWare User Guide*.

Once the files have been imported, they will be available for selection on page and control properties.

Loading the media files into NWare

1. Click the **Media** tab.
2. Right-click the **Media** tab, and then click **Import**.
3. Browse to the files you want to import, then select the files.
4. Click **Open**.

Renaming media files in the project

1. Click the **Media** tab.
2. Right-click the media file you want to rename.
3. Click **Rename**.
4. Type the new name for the file and then press **Enter**.

Tip: Each object that was configured to use the old file name will be updated to use the new file name automatically.

Deleting unused media files from the project

To minimize the size of the NWare project file, we recommend that you remove media files that are loaded into the project, but are not used by any objects.

⏩ *To delete unused media files from the project*

1. Click the **Media** tab.
2. Right-click the list of files, and then click **Delete Unused**.

Note: You cannot undo the delete action.

Exporting media files from the project

You can export media files from NWare for use in other applications.

⏩ *To export media files from the project*

1. Click the **Media** tab.
2. Select the files you want to export.
3. Right-click the **Media** tab, and then click **Export**.
4. Choose a location for the files, and then click **OK**.

Chapter 4

Adding hardware devices to your design

In This Chapter

Introduction

Your project must contain device blocks that represent the physical network devices you are going to use, such as NIONs and CABs. These devices will process the audio in your design, so it is important to understand which models are needed, based on their capabilities and the requirements of your project.

If you are unfamiliar with the available MediaMatrix devices, we recommend you read the entries in the glossary for a quick overview, or refer to more detailed information in brochures or on the *MediaMatrix website* (*http://www.peaveycommercialaudio.com/products.cfm*).

NWare supports devices fitted with CobraNet CM-1 or CM-2 modules, or Dante DLMs (Dante Legacy Modules). If you add a NioNode or a CAB to your design, you will need to choose which type of module you are using and the configuration options for passing audio will be different in each case.

Note: Currently, only the NioNode and CAB devices support both CobraNet and Dante. Other devices can only use CobraNet.

Once you have added the hardware devices to your design, refer to *Routing audio between devices* (on page 136) for detailed information on transferring audio between devices.

Adding a NioNode to your design

Introduction

NION is a digital audio processor. The following variants are currently available: the n3, the n6, the nX and the nE, which are all supported by NWare. The NION is compatible with the CAB audio break-out boxes (CAB 4n, CAB 16d etc.).

Tips:

- Within NWare, a NION is referred to as a NioNode. This term is use both in the NWare user interface and in the user documentation.
- If you click the Cancel button on the *Configure NioNode GPIO* dialog box when you are adding a NioNode to your design, this will cancel the process of adding the NioNode, not just close the dialog box. If you do not want to configure the GPIO, but still want to add the NioNode, click OK.

The NioNode device has a number of properties, which allow you to specify:

- The types cards installed in the expansion slots.
- A protocol to use for control over the IP network, e.g. RATC.
- A protocol to use for control via the serial port, e.g. PASHA.
- How the GPIO port will be used.
- etc.

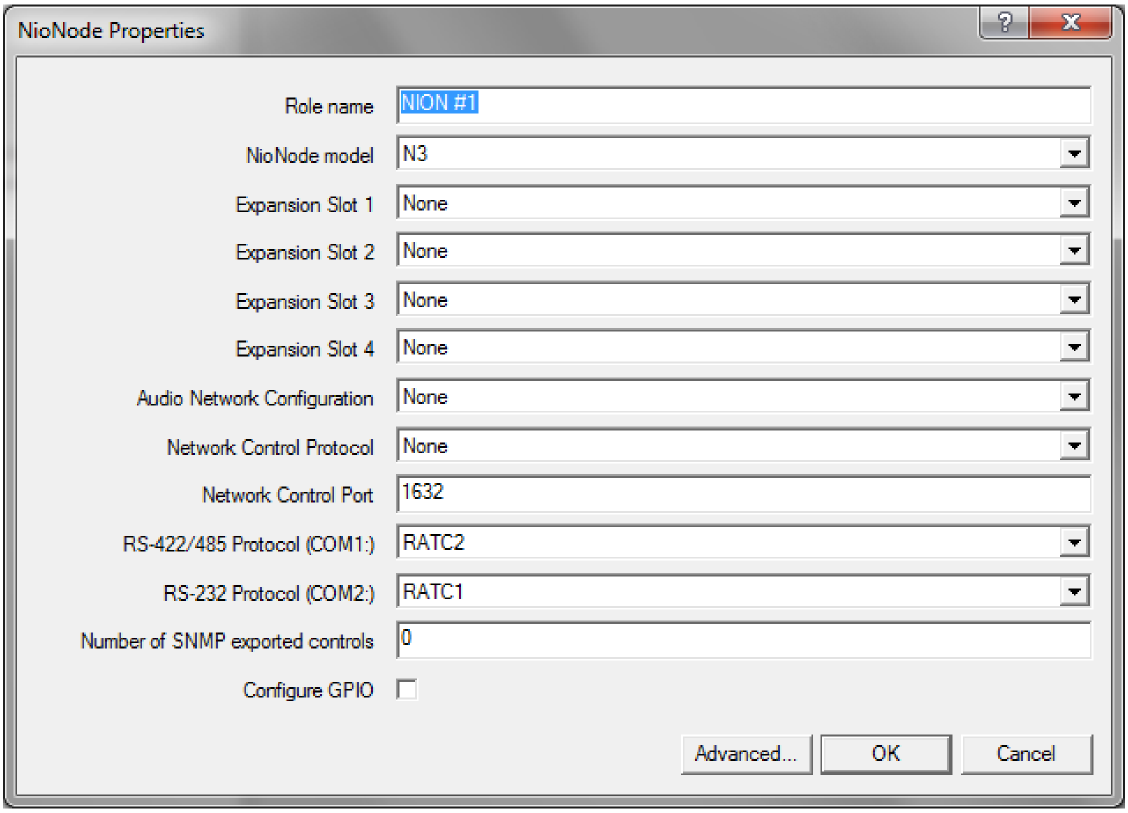
If you require reference information on the NioNode properties, see *NION Audio Processor* in the *NWare Device Reference*.

Adding the NioNode

1. In the device tree, expand the **Hardware** folder.

2. Expand the **NION Audio Processor** folder.
3. Drag an **N3**, **N6** or **nX/nE** object over to the design page.

   The **NioNode Properties** dialog box is displayed.



Role name	NION #1
NioNode model	N3
Expansion Slot 1	None
Expansion Slot 2	None
Expansion Slot 3	None
Expansion Slot 4	None
Audio Network Configuration	None
Network Control Protocol	None
Network Control Port	1632
RS-422/485 Protocol (COM1:)	RATC2
RS-232 Protocol (COM2:)	RATC1
Number of SNMP exported controls	0
Configure GPIO	☐

Note: The latest version of the NION NX has four expansion slots. However, older versions of the product only had two slots. If you are using an older model with two slots, you must only specify cards for slots 1 and 2.

4. If you want to change the role name from the default, type a new name in the **Role name** box.

   Note: You must specify a unique role name in the **Role name** box. If you do not do this, an error will be displayed when you deploy the project.

   Devices in your design are allocated to roles (either manually, or automatically by NWare). Roles are then assigned to MediaMatrix hardware devices for processing during deployment.
5. In the **Expansion Slot** boxes, select the card types that are installed or will be installed in the NioNode.

   Flyoffs for each card input channel and output channel will be added to the **Flyoffs** tab. For information on the NION and the available cards, see *http://mm.peavey.com/products/series.cfm?cat_id=54 (NION)* (*http://www.peaveycommercialaudio.com/products.cfm/mm/Audio-Networking/NION-Series*).
6. If you intend to make connections from this NioNode to other devices on the network, such as CAB 4ns or CAB 8ns, for example, click an option in the **Audio Network Configuration** list.

If you are using CobraNet, click one of the following options to specify how you want to organize the 32 input channels and 32 output channels into CobraNet network bundles.

CobraNet CM1: 4 8 Channel Bundles	The channels are organized into four bundles of eight individual channels.
CobraNet CM1: 8 4 Channel Bundles	The channels are organized into eight bundles of four individual channels.
CobraNet CM1: Advanced	NWare allows you to specify which channels are in which bundles on a per channel basis.

If you are using Dante network, click **Dante DLM**.

7. If you want to use the RATC protocol to control the NioNode over the IP network:
   a) In the **Network Control Protocol** list, click the protocol name.
   b) If you want the RATC server on the NioNode to listen on a different port to the default, type the new port number in the **Network Control Port** box.

      The default is 1632. The valid range is 1-32767.

      For detailed information on the available protocols, refer to the *External Control User Guide*.
8. If you want to use the RATC or PASHA protocol to control the NioNode via its serial ports, click the protocol name in the **RS-422/485 Protocol (COM1:)** and/or **RS-232 Protocol (COM2:)** lists.

None/Comms processor	Used when the project contains a Comms Processor device, or when you do not want to use the serial port. **Note:** If you select one of the other options – RATC2, for example, then try to use the Comms Processor device, it will not respond as quickly.
RATC2	RATC2 is the improved version of RATC that was introduced with the NION platform, but is also supported by nControl and nTouch 180. Very similar to RATC1, RATC2 introduces shortened commands, and several extra functions.
RATC1	RATC1 is the first generation protocol that was used in Classic frame-based MediaMatrix systems. RATC1 for NioNodes, nTouch 180 nodes and nControl nodes is equivalent to what was called RATC in Classic MediaMatrix.
PASHA/PageMatrix	Protocol used with the PageMatrix Command Center. It supports a four character control ID sent with the (T) trigger command.
PASHA/XControl	Provides basic S (Set) PASHA functionality with X-Net2-style trigger commands.

PASHA/Legacy	Designed for projects that contain legacy MediaMatrix nodes that are controlled by external programs using the classic PASHA protocol, as implemented on MainFrames and MiniFrames. When you want to replace the legacy nodes with NioNodes, select the *PASHA/Legacy* option and you will not need to update your external program code. This option ensures all hex values are returned from nodes in lower case instead of upper case to match the original MediaMatrix PASHA protocol.

9. If you want to use SNMP to manage controls in your project, type the number of controls in the **Number of SNMP exported controls** box.

   A flyoff for each control will be added to the **Flyoffs** tab. The maximum number of controls you can manage via SNMP is 512 per NioNode.

10. If you want to use the GPIO connector on the rear of the NioNode to communicate with external devices, select the **Configure GPIO** box.

11. If you want to manually set the CobraNet CM-1 latency, click **Advanced**, and then click a value in the **CobraNet CM-1 latency** list.

    The default is 48kHz / 5.333ms.

Note: We recommend that you do not change the latency from the default setting, as it affects the timing of traffic on the CobraNet network. For more information, see *Timing* in the *CobraNet Networking Guide*.

12. If you want to manually set the Conductor priority of this NioNode, click **Advanced**, and then click a value in the **CobraNet Conductor Priority** list.

    The default is **64..65**. The NioNode will automatically select a value in the specified range. If you leave this setting at the default value, a CobraNet Conductor will be selected automatically. The device with the highest value will become the Conductor, therefore selecting a higher value makes it more likely that this device will become the Conductor.

13. If you want to manually set the XDAB clock master priority, click **Advanced**, and then click a value in the **XDAB Clock Master Priority** list.

    The default is **0..1**. The NioNode will automatically select a value in the specified range. The NioNode with the highest value will become the XDAB master, therefore selecting a higher value makes it more likely that this NioNode will become the XDAB master.

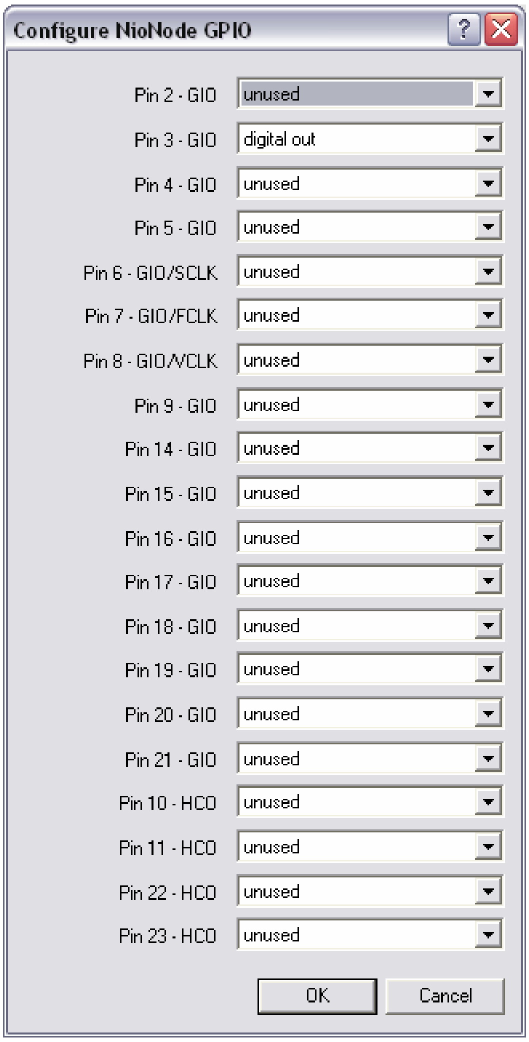
14. If you want to manually specify the source for the CobraNet clock signal, click **Advanced**, and then click an option in the **Clock Source** list.

Clock Source	Specifies the source for the CobraNet clock signal. This is used to synchronize devices on the network.	
	Automatic	If a CM-1 card is fitted to the NioNode, the clock signal will be received via this interface. If no CM-1 card is fitted, the signal will be generated by the AES card. This is the default setting.

	CM-1	The clock signal will be received via the CM-1 interface.
	I/O slot x	The clock signal will be received from an external source via an AES card in slot x. For information on using this setting, see *Using an external clock source to synchronize devices on a CobraNet network* (on page 55) in the *NWare User Guide*.

15. Click **OK**.
16. If you selected the **Configure GPIO** option:

    The Configure NioNode GPIO dialog box is displayed.



Pin 2 - GIO: unused
Pin 3 - GIO: digital out
Pin 4 - GIO: unused
Pin 5 - GIO: unused
Pin 6 - GIO/SCLK: unused
Pin 7 - GIO/FCLK: unused
Pin 8 - GIO/VCLK: unused
Pin 9 - GIO: unused
Pin 14 - GIO: unused
Pin 15 - GIO: unused
Pin 16 - GIO: unused
Pin 17 - GIO: unused
Pin 18 - GIO: unused
Pin 19 - GIO: unused
Pin 20 - GIO: unused
Pin 21 - GIO: unused
Pin 10 - HCO: unused
Pin 11 - HCO: unused
Pin 22 - HCO: unused
Pin 23 - HCO: unused

- Select the function for each pin.

General purpose ports 2-9,14-21	These ports represent the bulk of the GPIO functionality. Each of these ports can be configured as follows: ▪ Digital In (3.0V TTL logic - Low: 0 VDC - 0.8 VDC; High: 2.0 VDC - 24 VDC)

	▪ Digital Out (3.0V TTL logic - Low: 0V DC - 0.4 VDC; High: 2.4 VDC - 3.3 VDC) ▪ Analog In 1K, 12V (using external 12 VDC power source) ▪ Analog In 10K, 12V (using external 12 VDC power source) ▪ Analog In 10K, 24V (using external 24 VDC power source) ▪ Analog In 1K, self powered (pin feeds required voltage through pot or switch to common) ▪ Analog In 10K, self powered (pin feeds required voltage through pot or switch to common) ▪ Rotary Encoder (requires 2 pins and a common) ▪ Raw (all modes available, software configurable) ▪ Clock signals on pins 6-8.
High current output ports 10-11,22-23	Each high current port provides 11.5VDC at 0.5A. High current outputs can be configured for straight logic (on/off) or PWM (Pulse Width Modulation) operation.

Note: If you want to use the rotary encoder function, you must select *rotary encoder* for the pin, and the following pin must be set to *unused*.

For more information on the GPIO connector, see *GPIO overview* in the *NION Hardware Manual*.

Using an external clock source to synchronize devices on a CobraNet network

You can configure the NioNode AES card to receive a clock signal provided by a device connected to the card. This signal is transferred to the CobraNet network and used in place of the CM-1 clock signal from the CobraNet Conductor to synchronize devices on the network. When this set up is used, the NioNode fitted with the AES card automatically becomes the CobraNet Conductor.

The procedure below assumes you have added a NioNode device to your design and selected an AES card for one of the available slots.

Notes:

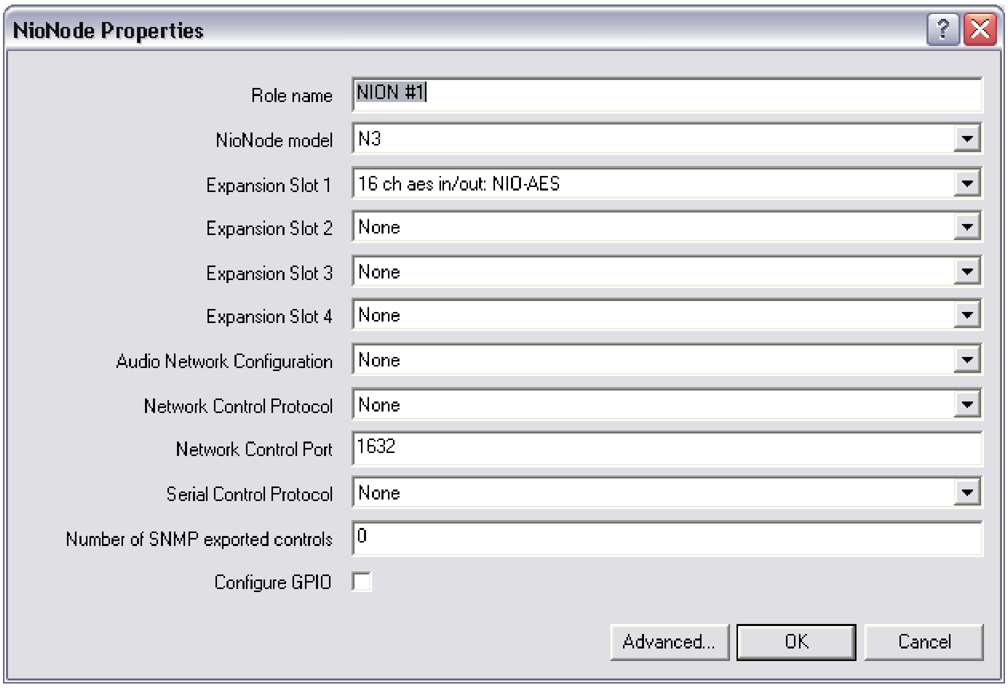
- The NioNode must be fitted with an AES card. The clock synchronization feature is not currently supported by other cards.
- Only channel 1 on the card can be used to receive a clock signal.
- Currently, when using an external clock source, you can only use a project sample rate of 48KHz – the other sample rate options are not supported. To view the project sample rate, on the *File* menu, click *Project Properties*.
- If you are using XDAB, the NION fitted with the AES card must be configured as the XDAB master.

Tip: The digital signal received by the AES card can contain audio as well as a clock signal. This allows you to make connections using the flyoff for channel 1 to other devices in your design.

⏩ *To use an external clock source to synchronize devices on a CobraNet network*

1. Connect the external clock source to the channel 1 connector on the AES card.
2. In NWare, right-click the NioNode, and then click **Device Properties**.

   The **NioNode Properties** dialog box is displayed.

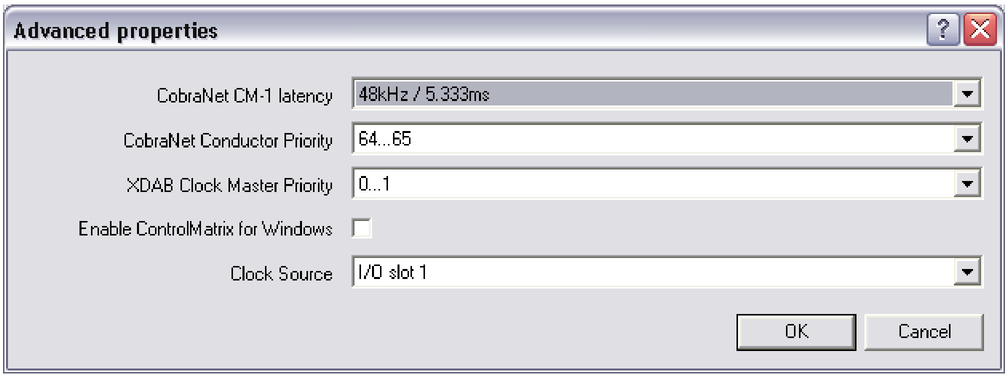


NioNode Properties

Role name	NION #1
NioNode model	N3
Expansion Slot 1	16 ch aes in/out: NIO-AES
Expansion Slot 2	None
Expansion Slot 3	None
Expansion Slot 4	None
Audio Network Configuration	None
Network Control Protocol	None
Network Control Port	1632
Serial Control Protocol	None
Number of SNMP exported controls	0
Configure GPIO	□

3. Click **Advanced**.

   The **Advanced properties** dialog box is displayed.

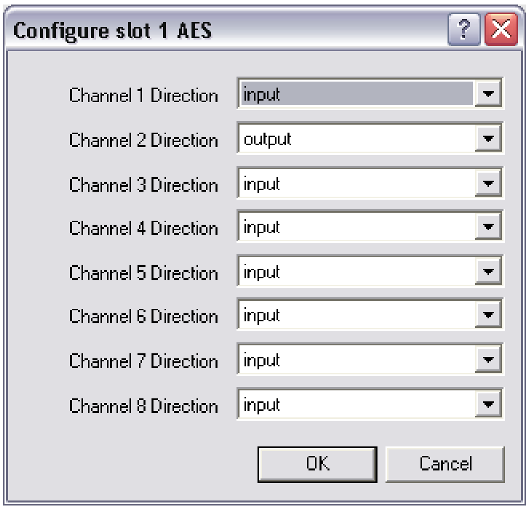


Advanced properties

CobraNet CM-1 latency	48kHz / 5.333ms
CobraNet Conductor Priority	64...65
XDAB Clock Master Priority	0...1
Enable ControlMatrix for Windows	□
Clock Source	I/O slot 1

4. In the **Clock Source** list, click an **I/O Slot** option corresponding to the slot number in which the AES card is inserted.
5. Click **OK**.
6. Click **OK**.

The **Configure Slot x AES** dialog box is displayed.



Channel 1 Direction: input
Channel 2 Direction: output
Channel 3 Direction: input
Channel 4 Direction: input
Channel 5 Direction: input
Channel 6 Direction: input
Channel 7 Direction: input
Channel 8 Direction: input

7. In the **Channel 1 Direction** list, click **input**.

   When a clock signal from an external source is detected on this input, the sample rate converters will be disabled automatically.
8. Click **OK**.

Understanding the AES card control surface when using an external clock

When a NioNode has been configured to receive an external clock signal via the AES card, an LED is shown on the control surface (*Slot x: AES* tab) next to *Ext clock source*; this is for input channels 1 and 2.

When the system is working normally and a clock signal is received, the *Ext clock source* and *Lock* LEDs are lit (green). If the clock signal is interrupted, the *Lock* LED will be unlit and the *Error* LED will be lit (red); all channels on the AES card will be muted automatically; the last received clock signal will continue to be transmitted from the AES card on to the network so that devices can be synchronized and there is no audio disruption.

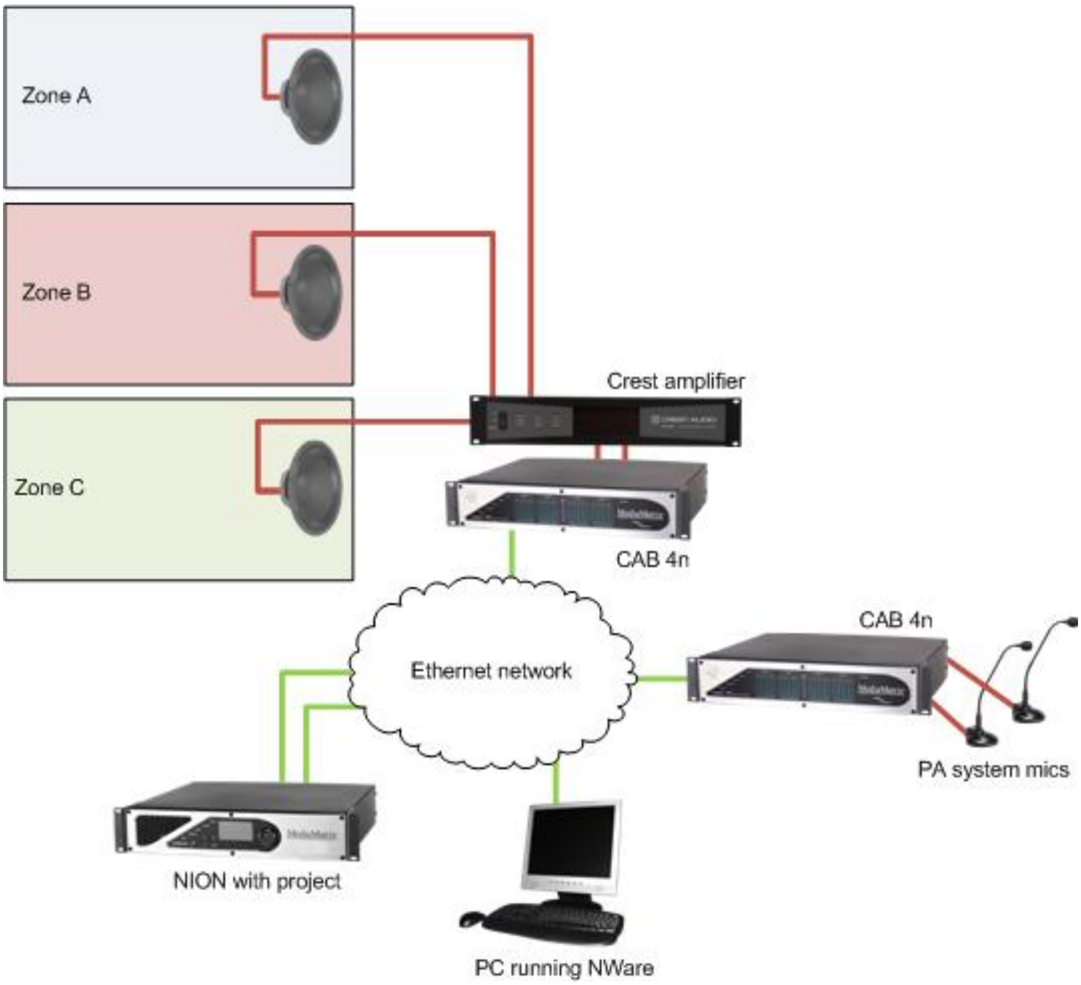
Note: If the signal received via the AES card contains audio as well as a clock signal, and the audio is being passed to other devices in the project using flyoffs, we recommend that the devices connected to these flyoffs are muted automatically should the clock signal no longer be received. When the clock signal is recovered, after a short delay – 0.5 seconds, for example, the devices can be unmuted. The delay allows the AES clock to resync to the external clock signal.

Adding a CAB 4n node to your design

Introduction

The CAB 4n is a professional digital audio processor intended for fixed installation applications. This 2U rack-mount package is designed to provide high quality audio performance and easy to use controls. Engineered from the ground up with the commercial sound systems contractor in mind, the CAB 4n includes removable screw connectors for easy installation and cost effective servicing, as well as front panel hidden ID assignment.

CABs provide a bridge for audio devices so that they can be connected together over large distances via a network. The example below shows how two CAB units can be used to transfer audio from microphones at one end of the network to speakers at a different location. The project is managed using NWare and hosted by the NioNode.



Tip: If you want to control external devices using GPIO, but not pass audio via the CAB, it can be controlled using an nControl or an nTouch 180 without a NioNode in the project. For more information, see *Interfacing with a CAB device* in the *nControl Hardware Manual* or *Interfacing with a CAB device* in the *nTouch 180 Hardware Manual*.

The CAB 4n has a number of properties, which allow you to:

- specify network settings.
- specify gain settings for inputs and outputs.
- monitor the GPIO connections
- control data flow through the serial port.

Each CAB 4n can receive up to four separate audio bundles of 4 channels each. You can control the analog gain stage by gesturing the gain controls on the control surface. The range is -95 dB to +30 dB.

If you require reference information on the CAB 4n device properties, see *CAB 4n* in the *NWare Device Reference*.

Hardware and software devices

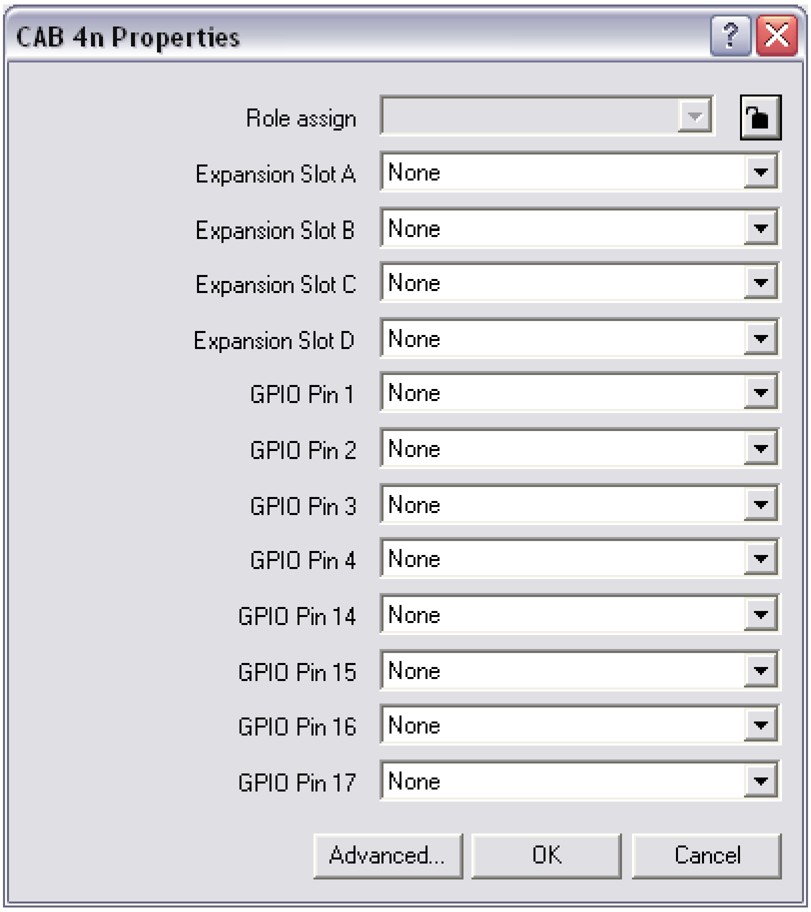
The CAB 4n is an integration of physical hardware and software. In software, the equipment is represented by a yellow *CAB 4n* or *CAB Dante* block in NWare.

There is a very basic relationship between the physical CAB product and its accompanying NWare device. Essentially, the NWare device is the *control panel* for the hardware. There are no physical controls on the CAB itself, except for the Hardware ID controls. For information on setting the Hardware ID, see *Setting the Hardware ID* in the *CAB 4n Hardware Manual*.

Adding the CobraNet CAB 4n

1. In the device tree, expand the **Hardware** folder.
2. Expand the **CAB Audio I/O** folder.
3. Drag a **CAB 4n** object over to the design page.

   The **CAB 4n Properties** dialog box is displayed.



CAB 4n Properties

Role assign:
Expansion Slot A: None
Expansion Slot B: None
Expansion Slot C: None
Expansion Slot D: None
GPIO Pin 1: None
GPIO Pin 2: None
GPIO Pin 3: None
GPIO Pin 4: None
GPIO Pin 14: None
GPIO Pin 15: None
GPIO Pin 16: None
GPIO Pin 17: None

4. If you want to change the role to which this device belongs, click the 🔓 button to change it to the 🔒 button, and then select a role name in the list.
5. In the **Expansion Slot** boxes, select the card types that are installed or will be installed in the CAB.

   The CAB cards are color-coded to help you identify them.
6. If you want to use the GPIO connector on the rear of the CAB to communicate with external devices, select the function for each pin in the **GPIO Pin** lists.

For more information on the GPIO connector, see *GPIO pin assignment* in the *CAB 4n Hardware Manual*.

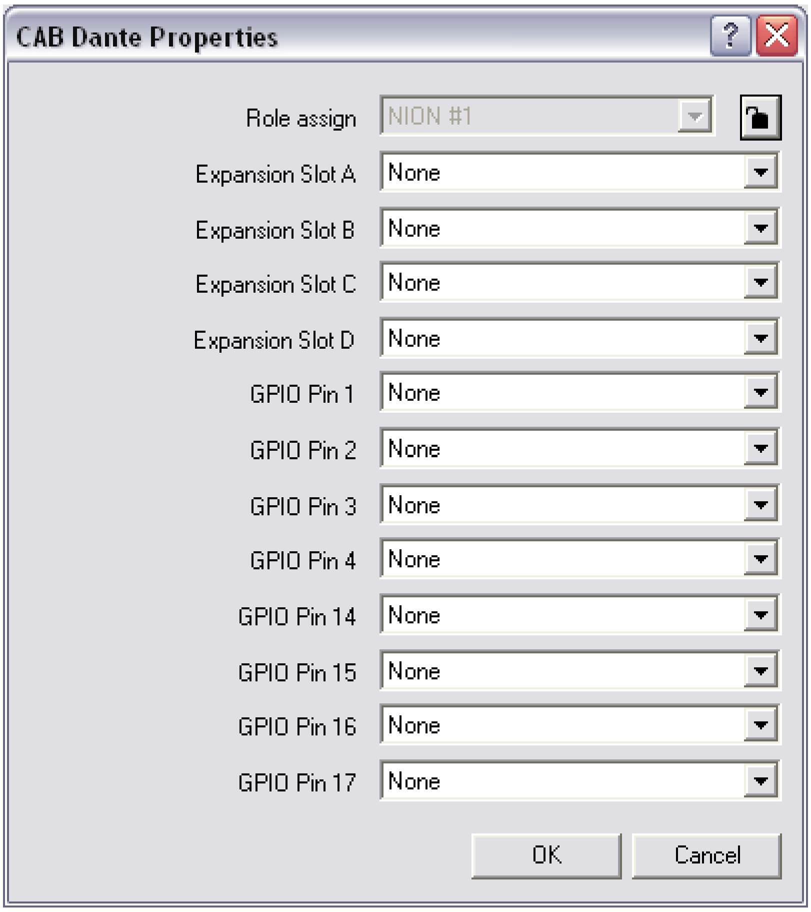
7. If you want to allow a raw, RS-485 MAC configuration to send and receive from non-CobraNet MACs, click **Advanced**, and then select the **RS-485 Raw MAC Enable** check box.
8. If you intend to use the CAB with ControlMatrix, click **Advanced**, and then select the **ControlMatrix Mode** check box.

   This will set the baud rate and format defaults and add input wiring nodes to the block for serial bridging MACs when used with ControlMatrix.
9. If you want to specify how the CobraNet input and output channels will be organized into bundles, click **Advanced**, and then select the **Advanced Subchannel Mapping** check box.
10. Click **OK**.

Adding the Dante CAB 4n

1. In the device tree, expand the **Hardware** folder.
2. Expand the **CAB Audio I/O** folder.
3. Drag a **CAB 4n Dante** object over to the design page.

   The **CAB Dante Properties** dialog box is displayed.



Role assign: NION #1
Expansion Slot A: None
Expansion Slot B: None
Expansion Slot C: None
Expansion Slot D: None
GPIO Pin 1: None
GPIO Pin 2: None
GPIO Pin 3: None
GPIO Pin 4: None
GPIO Pin 14: None
GPIO Pin 15: None
GPIO Pin 16: None
GPIO Pin 17: None

4. If you want to change the role to which this device belongs, click the 🔓 button to change it to the 🔒 button, and then select a role name in the list.

5. In the **Expansion Slot** boxes, select the card types that are installed or will be installed in the CAB.

   The CAB cards are color-coded to help you identify them.
6. If you want to use the GPIO connector on the rear of the CAB to communicate with external devices, select the function for each pin in the **GPIO Pin** lists.

   For more information on the GPIO connector, see *GPIO pin assignment* in the *CAB 4n Hardware Manual*.
7. Click **OK**.

Setting the Hardware ID

In order for the NWare CAB device to communicate with the physical CAB hardware, you must set the *Id* field on the CAB device to the hardware address specified on the physical CAB hardware. Before this can be done, the project must be deployed or *emulated* (on page 167).

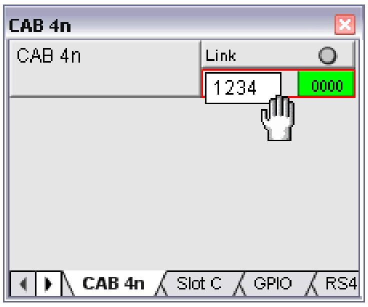
⏩ *To set the Hardware ID*

1. Open the device tree and drag a NioNode, nControl node or nTouch 180 node over to the design page.
2. Double-click the CAB 4n device to open the block.
3. Click the **CAB 4n** tab.

   The **id** box is the Hardware ID. This is a four digit hexadecimal number between 0000 and FFFF (0000 is off). This number uniquely identifies the CAB on the network. It is specified on the front panel of the CAB hardware using rotary switches.
4. On the toolbar, click **Emulate** .

   The project will be emulated and NWare will switch to *Gesture mode*.
5. Hover the mouse pointer over the **id** box so the red selection rectangle appears, then type the four digits of the Hardware ID.

   Acceptable values are in hex and between 0000 and FFFF (0000 is off).



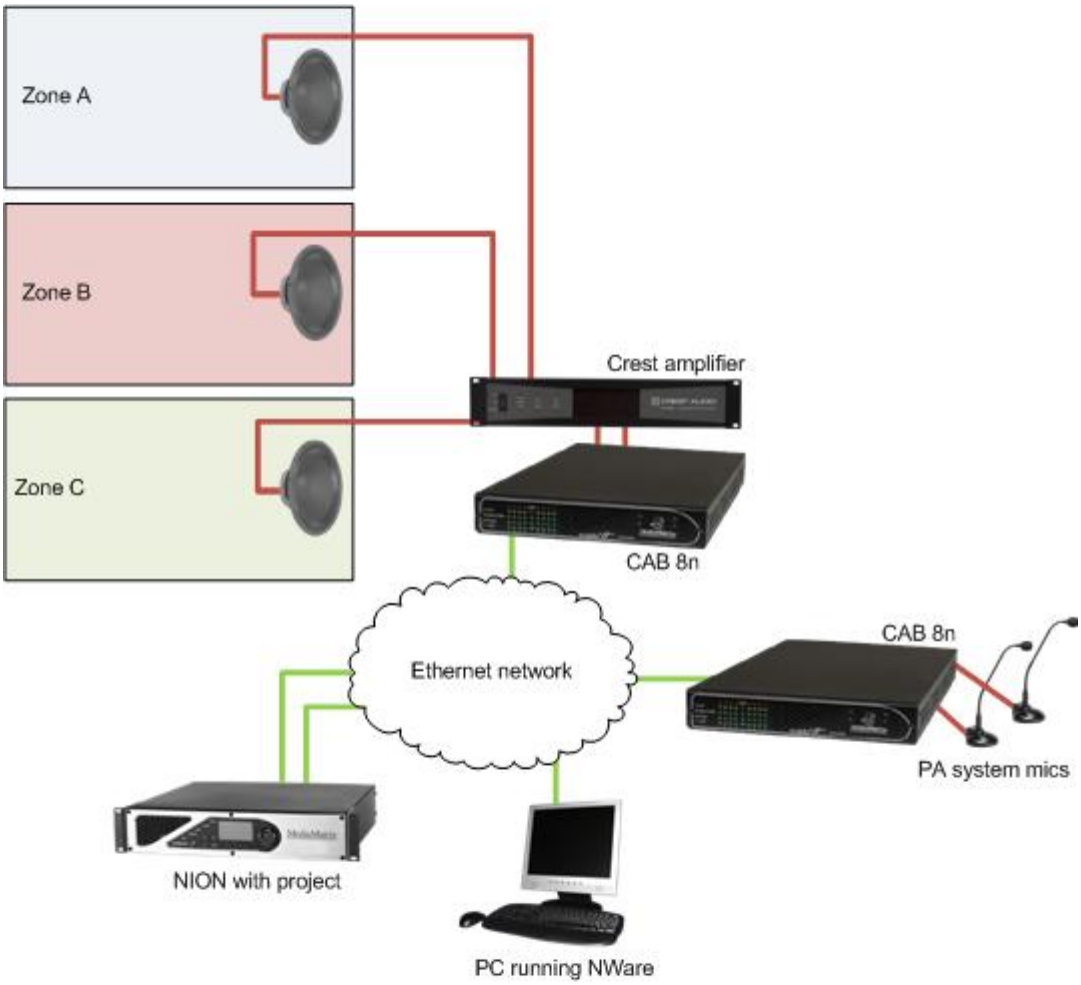
Tip: After deploying the project to a NioNode, and when a valid number is entered, the green Link LED next to the ID number will light. On the front of the CAB, the LEDs will stop showing the Hardware ID and become meters.

Adding a CAB 8n node to your design

Introduction

The CAB 8n Configurable Audio Bridge is an eight channel audio input and output device. It is designed to be used with MediaMatrix NIONs in professional and commercial audio and communications applications.

CABs provide a bridge for audio devices so that they can be connected together over large distances via a network. The example below shows how two CAB units can be used to transfer audio from microphones at one end of the network to speakers at a different location. The project is managed using NWare and hosted by the NioNode.



Tip: If you want to control external devices using GPIO, but not pass audio via the CAB, it can be controlled using an nControl or an nTouch 180 without a NioNode in the project. For more information, see *Interfacing with a CAB device* in the *nControl Hardware Manual* or *Interfacing with a CAB device* in the *nTouch 180 Hardware Manual*.

The CAB 8n device has a number of properties, which allow you to:

- specify network settings.
- specify gain settings for inputs and outputs.
- monitor the GPIO connections

- control data flow through the serial port.

If you require reference information on the CAB 8n device properties, see *CAB 8n* in the *NWare Device Reference*.

Hardware and software devices

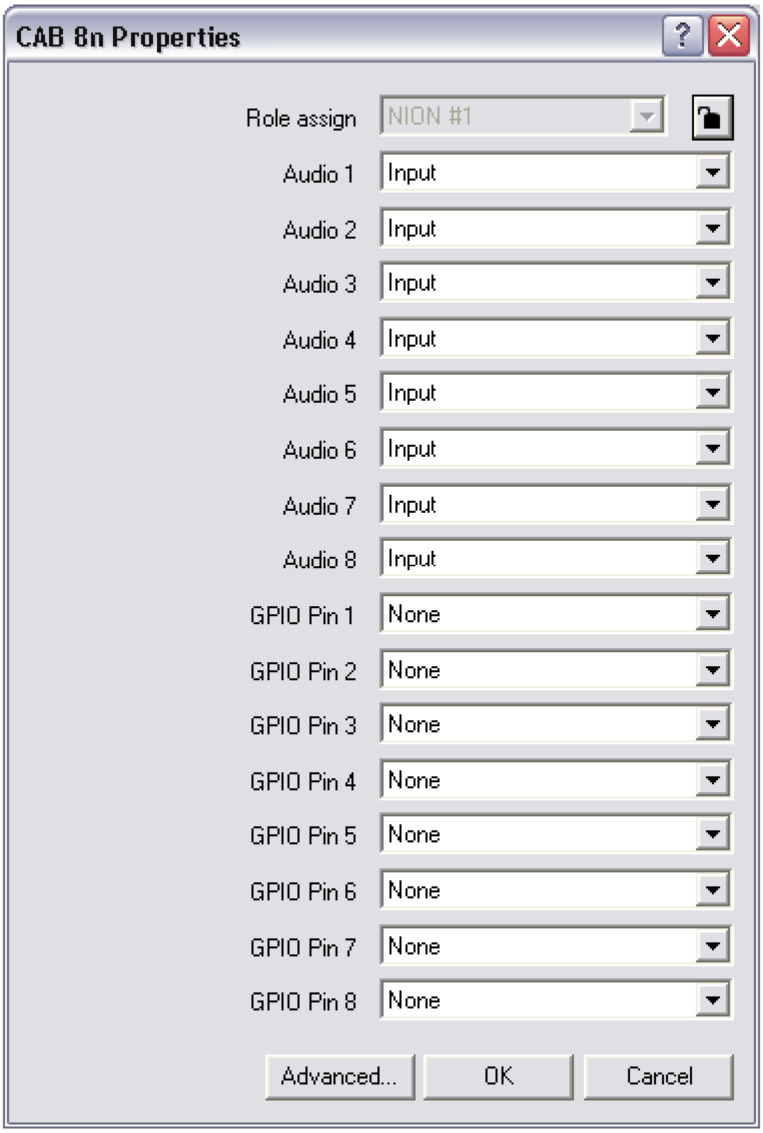
The CAB 8n is an integration of physical hardware and software. In software, the equipment is represented by a yellow *CAB 8n* block in NWare.

There is a very basic relationship between the physical CAB product and its accompanying NWare device. Essentially, the NWare device is the *control panel* for the hardware. There are no physical controls on the CAB itself, except for the Hardware ID controls. For information on setting the Hardware ID, see *Setting the Hardware ID* in the *CAB 8n Hardware Manual*.

Adding the CAB 8n

1. In the device tree, expand the **Hardware** folder.
2. Expand the **CAB Audio I/O** folder.
3. Drag a **CAB 8n** object over to the design page.

The **CAB 8n Properties** dialog box is displayed.



CAB 8n Properties

Role assign	NION #1
Audio 1	Input
Audio 2	Input
Audio 3	Input
Audio 4	Input
Audio 5	Input
Audio 6	Input
Audio 7	Input
Audio 8	Input
GPIO Pin 1	None
GPIO Pin 2	None
GPIO Pin 3	None
GPIO Pin 4	None
GPIO Pin 5	None
GPIO Pin 6	None
GPIO Pin 7	None
GPIO Pin 8	None

4. If you want to change the role to which this device belongs, click the 🔓 button to change it to the 🔒 button, and then select a role name in the list.
5. In the **Audio** lists, click **Input** or **Output** to specify the function of each audio connector on the device.
6. Click **Advanced**, and then select the **Advanced Subchannel Mapping** check box. Click **OK**.
7. If you want to use the GPIO connector on the rear of the CAB to communicate with external devices, select the function for each pin in the **GPIO Pin** lists.

   For more information on the GPIO connector, see *Configurable general purpose ports* in the *CAB 8n Hardware Manual*.
8. Click **OK**.

Setting the Hardware ID

In order for the NWare CAB device to communicate with the physical CAB hardware, you must set the *Id* field on the CAB device to the hardware address specified on the physical CAB hardware. Before this can be done, the project must be deployed or *emulated* (on page 167).

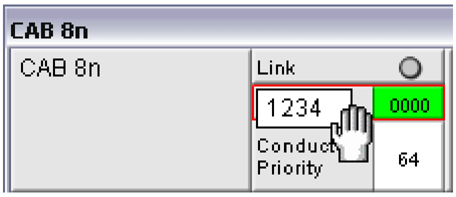
To set the Hardware ID

1. Open the device tree and drag a NioNode, nControl node or nTouch 180 node over to the design page.
2. Double-click the CAB 8n device to open the block.
3. Click the **CAB 8n** tab.

   The **id** box is the Hardware ID. This is a four digit hexadecimal number between 0000 and FFFF (0000 is off). This number uniquely identifies the CAB on the network. It is specified on the front panel of the CAB hardware using rotary switches.
4. On the toolbar, click **Emulate** .

   The project will be emulated and NWare will switch to *Gesture mode*.
5. Hover the mouse pointer over the **id** box so the red selection rectangle appears, then type the four digits of the Hardware ID.

   Acceptable values are in hex and between 0000 and FFFF (0000 is off).



Tip: After deploying the project to a NioNode, and when a valid number is entered, the green Link LED next to the ID number will light. The meter LEDs on the front of the CAB will start to represent the signal levels for the audio channels.

Setting up the gain structure

If you are planning to use the connectors on the CAB 8n as inputs, we recommend that you configure them to provide optimal performance for the expected input signal levels. The settings you need to adjust are located on the *Audio* tab of the NWare CAB 8n device.

If the maximum output level of each audio source is known, you can set the *Maximum Input Signal* setting on the CAB 8n device to match this level. You can also add some headroom, if desired, to prevent the signal from being clipped automatically if it exceeds the maximum level.

When the audio arrives into the CAB 8n, the gain level will be adjusted automatically to make optimal use of the A to D converter. For example, if the input signal level equals the maximum, which is 0.25V, but full scale in the A to D converter is 2.5V, the signal level will be multiplied by 10 to equal 2.5V.

Once the *Maximum Input Signal* setting has been specified for each input channel, it should not need adjustment. Only the *Level Control* setting needs to be adjusted.

Using a CAB 8n in standalone mode

The CAB 8n can be configured so that when the project has been deployed, the CAB 8n does not require a connection to a NION, nControl or nTouch 180 in order to send and receive audio data on the CobraNet network. Even when the CAB 8n is power cycled, it still continues to send and receive audio.

Note: You will need a NION, nControl or nTouch 180 initially, in order to deploy the project and send the configuration to the CAB, but once this is done they will no longer be required.

⏩ *To use a CAB 8n in standalone mode*

1. Open the device tree and drag a NioNode, nControl node or nTouch 180 node over to the design page.
2. Drag a CAB 8n over to the design page.

   When you add the CAB 8n, select the **Advanced Subchannel Mapping** option on the **Device Properties / Advanced properties** dialog box.
3. Deploy or emulate the project.
4. Open the **CAB 8n** block.
5. On the **CAB 8n** tab, specify the settings you want the CAB 8n to use when it is running in standalone mode.

   Tip: These will need to include the Id (hardware id) and TX or RX bundle number and subchannel settings.
6. Click **Persistent configuration**.
7. Click **Standalone mode**.

Setting up redundancy

The CAB 8n redundancy system allows a standby unit to replace an active unit on the network if it stops functioning.

This section assumes you have physically installed and connected two CAB 8ns, as described in the section *Operating modes and redundancy* in the *CAB 8n Hardware Manual*. It also assumes you have dragged two CAB 8n devices and a NioNode, nControl node or nTouch 180 node over to the design page in NWare.

⏩ *To set up redundancy*

1. Deploy the project.
2. Open the CAB 8n blocks to display the control surfaces.
3. Type the hardware IDs for the master and slave CABs into the **Id** boxes on the CAB control surfaces.

   Each CAB will need a unique ID, and in each case, when a valid number is entered, the green Link LED next to the ID number will light. The meter LEDs on the front of the CAB will start to represent the signal levels for the audio channels.
4. On the master CAB 8n control surface, in the **Failover Mode** list, click **Master**.
5. On the slave CAB 8n control surface, in the **Failover Mode** list, click **Slave**.
6. Type the bundle numbers for transmission and reception of audio into the **Bundle** boxes on the master CAB control surface.

7. Type same bundle numbers for transmission and reception of audio used for the master CAB into the **Bundle** boxes on the slave CAB control surface.

   The **TX** and **RX** LED(s) will be lit on the master CAB, but not the slave.
8. Type the bundle numbers for transmission and reception of audio into the **Bundle** boxes on the other devices in the design that will receive audio from the CAB or send audio to the CAB. Check that the **TX** and **RX** lights are lit showing that audio is flowing through the system.
9. On the master CAB 8n control surface:
   a) Click **Persistent configuration**.
   b) Click **Standalone mode**.
   c) Click **Persistent configuration**.
10. Repeat step 9 on the slave CAB 8n control surface.

    The units will click when these settings are changed and the display on the slave CAB will change to show arrows continuously moving from left to right. The unit is now in standby mode.

Adding an nWall node to your design

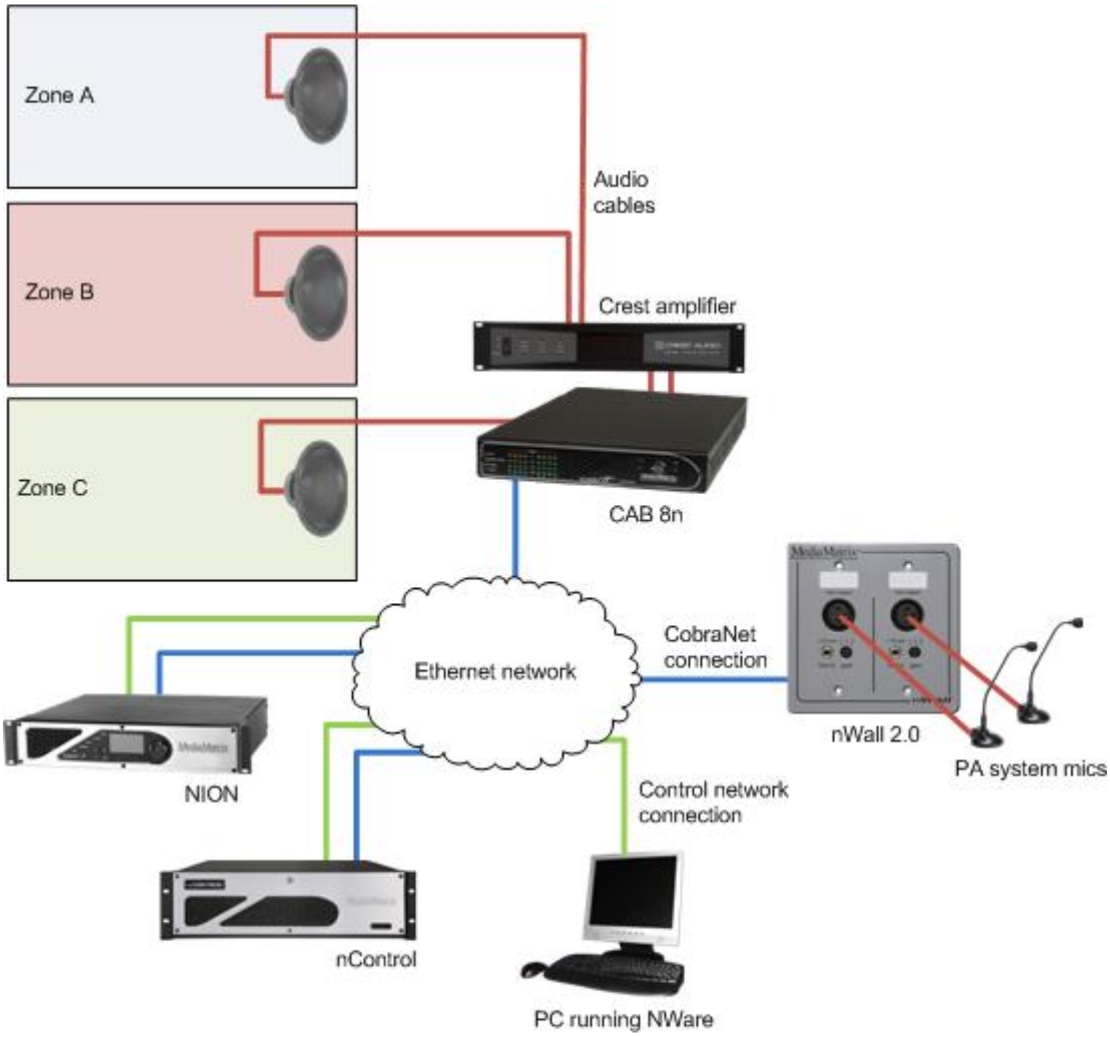
Introduction

The nWall is a CobraNet interface panel with both XLR and TRS mini jack connectors. The nWall 2.0 has two microphone/line level input channels. The nWall 0.2 has two analog line level output channels. The nWall 1.1 has two analog line level input channels and two analog line level output channels.

The nWall eliminates the need for long runs of analog audio cables terminated in racks of patch bays, as typically seen in ballrooms and convention centers. As the connection to the nWall uses CAT5e UTP cable with network standard RJ45 crimp connectors, the installation time, number of terminations required and associated cost are all minimized.

In the example below, the audio from the mics is made available directly onto the CobraNet network without the need for a CAB or NION in the same physical location. The NWare project is hosted on the nControl, allowing the NWare user to remotely assign a bundle number to the nWall for audio transmission.

Note: If you want to control and monitor an nWall using an NWare project, you will need an nControl or nTouch 180. If you simply want to transmit or receive audio using an nWall, these devices are not required. You can use CobraNet Discovery to configure and monitor the nWall.



There is a different nWall device for each nWall model. These devices allow you to specify an IP address, which links the NWare device to the physical device on the network, and a bundle number for transmission and / or reception of data via CobraNet.

Note: Specifying an IP address for an nWall device in NWare does not assign the IP address to the physical device. CobraNet Discovery must be used to assign an IP address to the nWall.

Each nWall device has a Link LED that shows when NWare has established a connection to the physical device. When the device is receiving or transmitting data, the *Receiving* or *Transmitting* LED is lit.

If you require reference information on the nWall device properties, see *nWall* in the *NWare Device Reference*.

Understanding how IP addresses and bundle numbers are used

In order to contact a physical nWall device from an NWare nWall device, the physical device must first be assigned an IP address. This is done using CobraNet Discovery. For more information, see *Assigning bundle numbers and an IP address* in the *nWall Hardware Manual*.

To make audio data available to other CobraNet devices on the network, or to receive audio from other network devices, the nWall must be assigned a bundle number. This can be done using CobraNet Discovery, or, once NWare is connected to the nWall, from the NWare nWall device control surface.

For an nWall 0.2, the bundle will contain two output channels:

- The line output for the first XLR connector and first mini-jack.
- The line output for the second XLR connector and second mini-jack.

For an nWall 1.1, two bundle numbers are used. When the device is transmitting audio onto the network, the two channels inside the transmitter bundle contain two *separate* audio streams:

- audio from the mini-jack on channel 1.
- audio from the XLR connector on channel 2.

When the device is receiving audio from the network, the two channels of audio inside the receiver bundle are used as follows:

- The mini-jack receives audio from channel 1.
- The XLR connector receives audio from channel 2.

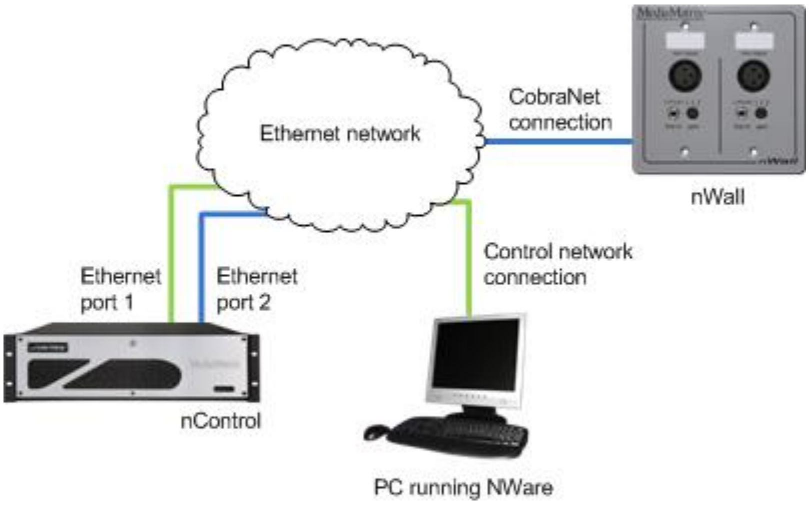
For an nWall 2.0, the bundle will contain two input channels:

- The mic input from the first XLR connector summed with the line input from the first mini-jack.
- The mic input from the second XLR connector summed with the line input from the second mini-jack.

Note: An nWall 1.1 or nWall 2.0 will start to transmit audio data onto the CobraNet network as soon as a transmit bundle number is assigned.

Setting up the nControl or nTouch 180

In order to monitor and configure an nWall from NWare, you will need an nControl or nTouch 180 connected to both the control network and the CobraNet network. The example below shows the required setup for an nControl.



Before you start using NWare to connect to the nWall, it is important to check that the network settings for the connection to the CobraNet network are correct. In the procedure below, it is assumed that *Local Area Connection 1* is connected to the control network and *Local Area Connection 2* will be used for the CobraNet network. For information on the nControl network ports, see *Rear network ports* in the *nControl Hardware Manual*.

⏩ *To set up the nControl or nTouch 180*

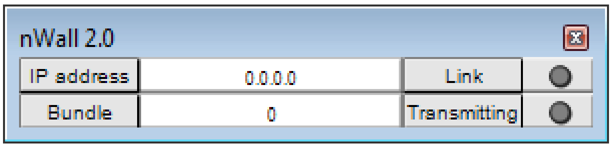
1. Open a web browser, then connect to the nControl or nTouch 180 web interface.
2. On the **Network** screen, under **Local Area Connection 2**, check that the network settings are correct for the CobraNet network.

   If no DHCP server is available, you will need to use a static IP address.
3. Click **Set** to confirm the action.

   You will be asked to log on.
4. Specify your username and password. The default username is *superuser*; it has no password.

Adding the nWall device to your NWare project

1. In the device tree, expand the **Plugins/Peavey/Hardware/nWall** folder.
2. Drag an **nWall** device over to the design page.
3. Double-click the device block to show the control surface.

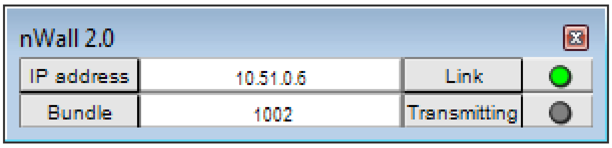


nWall 2.0

IP address	0.0.0.0	Link	●
Bundle	0	Transmitting	●

4. Deploy the project.
5. In the **IP address** box, type the IP address of the nWall.

   When the nWall has been contacted, the *Link* LED will be lit, and if a bundle number has been assigned for transmitting or receiving audio, it will be displayed in the *Bundle* box.



nWall 2.0

IP address	10.51.0.6	Link	🟢
Bundle	1002	Transmitting	●

When another device on the CobraNet network starts to transmit or receive audio using the bundle number assigned to the nWall, the *Transmitting* or *Receiving* LED will be lit on the nWall control surface.

If you want to use a different bundle number, type the new number in the **Bundle** box. If you are also running CobraNet Discovery, the new bundle number will be displayed as soon as you refresh the display.

Tip: If you do not know the IP address of the nWall, but know other information assigned to it, such as the *sysName* or *sysDescription*, you can use an Agent Discovery device to contact the nWall and retrieve the IP address. For more information, see *Connecting to the nWall when you do not know the IP address* (on page 71).

Connecting to the nWall when you do not know the IP address

If you do not know the IP address of the nWall, you can use the Agent Discovery device in NWare to discover the nWall on the CobraNet network, and retrieve its IP address.

A range of different identifiers can be used to locate the nWall. These are specified on the control surface of the Agent Discovery device, in order to start the discovery process.

- MAC address
- Sys Object ID
- Sys Name
- Sys Contact
- Sys Location.

This method utilizes the BOOTP and SNMP features of the nControl or nTouch 180. It is especially helpful and suited for larger installations, as it uses switch port discovery.

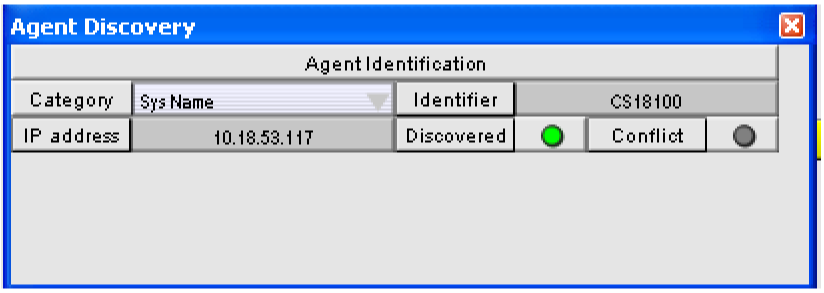
Note: Even though the discovery process will find the IP address of the nWall, you will still need to specify an IP address range that includes the IP address of the nWall.

⏩ *To contact the nWall when you do not know the IP address*

1. In the device tree, expand the **Hardware** folder.
2. Drag an **nControl** or **nTouch 180** device over to the design page.
3. In the device tree, expand the **Plugins / Peavey / Hardware / nWall** folder.
4. Drag an **nWall** device over to the design page.
5. Open the device properties on the nControl / nTouch 180.
6. Select the **CobraNet BOOTP** check box.
7. Select the **SNMP Discovery** check box.
8. Click **OK**.
9. On the **Configure BOOTP Server** dialog box, specify an IP address range for the nWall nodes on the CobraNet network.
10. Click **OK**.
11. On the **Configure Discovery Server** dialog box, specify the IP address range to use with BOOTP, and select the **Switch port support** check box.
12. In the **Update period (mins)** box, type a value for the interval between scans for devices on the network.
13. Click **OK**.

14. In the device tree, expand the **nControl Devices / SNMP Components** folder, and then drag an **Agent Discovery** device over to the design page.
15. Select Wire mode , then wire the first output wiring node on the **Agent Discovery** block to the input wiring node on the **nWall** block.
16. Deploy the project and specify an identifier for the nWall using the **Agent Discovery** device.

Tip: You can use the *Sys Name* setting, for example, which is *CS18100*, by default.



Agent Discovery

Agent Identification

Category	Sys Name	Identifier	CS18100		
IP address	10.18.53.117	Discovered	●	Conflict	●

When the nWall has been discovered, its IP address will be displayed in the *IP address* box on the Agent Discovery device, and on the nWall device. When NWare contacts the nWall using the IP address, the Link LED on the nWall device will be lit.

Adding an iPad device to your design

Introduction

The iPad is a touch screen device made by Apple Inc. For more information, see *apple.com/ipad* (*http://www.apple.com/ipad*).

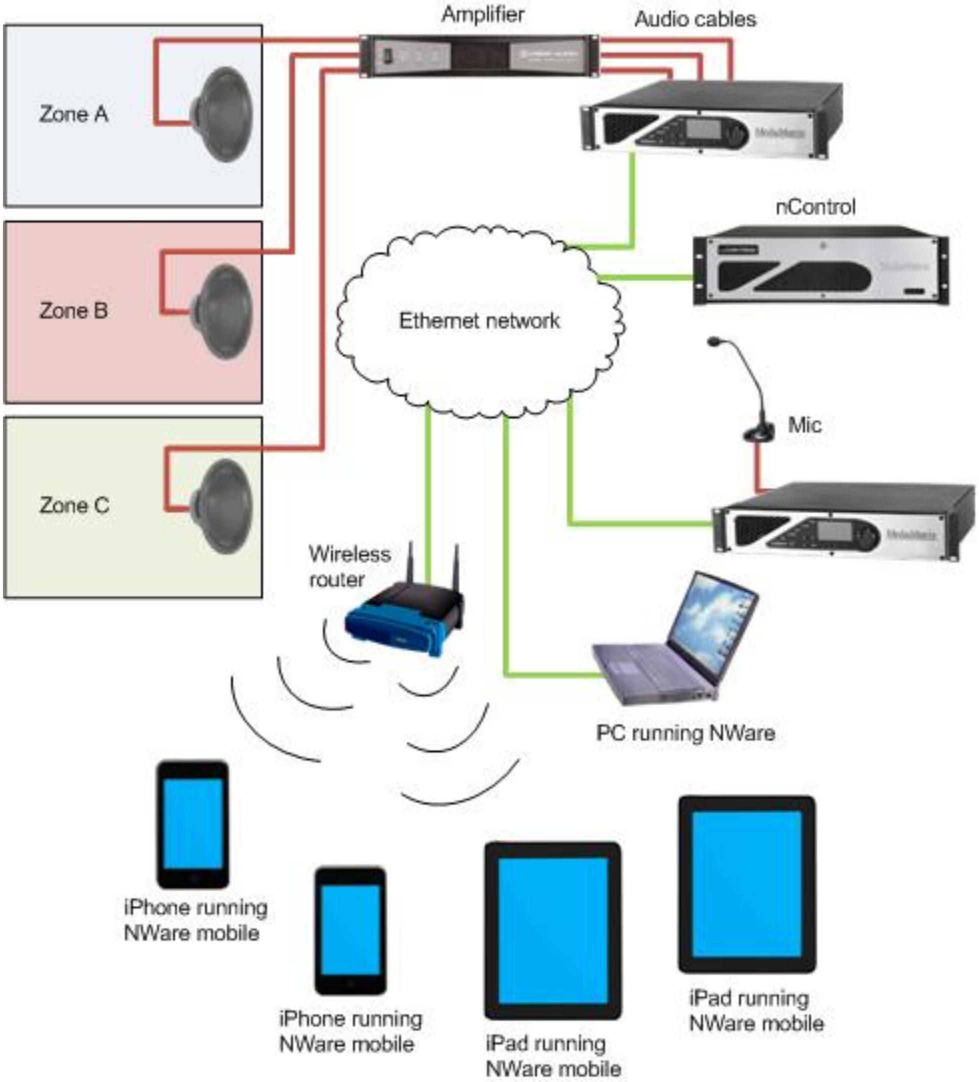
The NWare iPad device allows you to control an NWare project from an iPad. It works in conjunction with an nControl unit or an nTouch 180 unit, which hosts the Kiosk pages from the NWare project.

Notes:

- The iPad requires iOS version 4.2 or later.
- The *NWare Mobile* application or *NWare Mobile Designer Edition* application must be installed on the iPad. These are both available via *iTunes* (*http://itunes.apple.com/gb/artist/peavey-electronics-corporation/id436001313*).
- NWare Mobile Designer Edition is fully functional, but has a time limit. You can use it to connect to a project as many times as you want for one hour; afterwards, connections will be refused until the project has been redeployed or reemulated, or the host (nControl or nTouch 180) has been power cycled.
- If you are going to deploy the project, Apple Bonjour must be installed on the nControl or nTouch 180.
- If you are going to use emulation mode, Apple Bonjour must be installed on the NWare PC. This application is installed automatically when iTunes is installed.
- There is no set limit on the number of iPad devices that can be added to a project. However, if you add several of these devices and users are interacting with them simultaneously, you may find that the responsiveness of the Kiosk screens is affected.

- We recommend that you place nTouch 60 devices, Kiosk Layout blocks, Kiosk2Go Layout blocks and Mobile Display blocks on a separate page to the other devices in the project, and do not export the page.

In the example below, users can control the audio system via any of the Apple devices. To use a set up like this, your NWare project will need to contain two iPhone Mobile Display devices and two iPad Mobile Display devices, each configured to contact the corresponding Apple device. You will also need an nControl node, two NioNodes and an amplifier device.

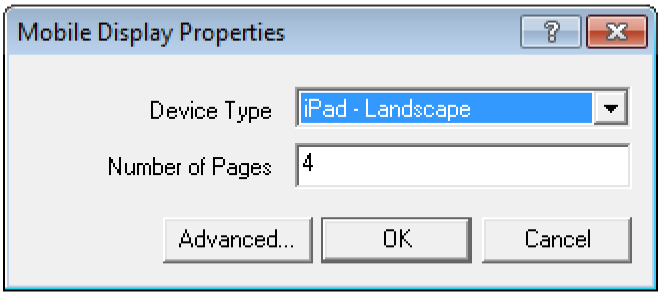


The iPad sections in the NWare User Guide will explain how to add an iPad to your design and configure it. If you require only reference information on the iPad device, see *iPad Device* in the *NWare Device Reference*.

Adding the iPad device

1. In the device tree, expand the **Hardware** folder.
2. Expand the **Mobile Display** folder, then drag an **iPad Device** node over to the design page.

The **Mobile Display Properties** dialog box is displayed.



Mobile Display Properties

Device Type: iPad - Landscape

Number of Pages: 4

The **Device Type** is selected automatically.

3. In the **Number of Pages** box, type the number of individual design pages to be displayed on the device when it is created.

   These are the pages that will be visible on the device itself when the project is running.

Notes:

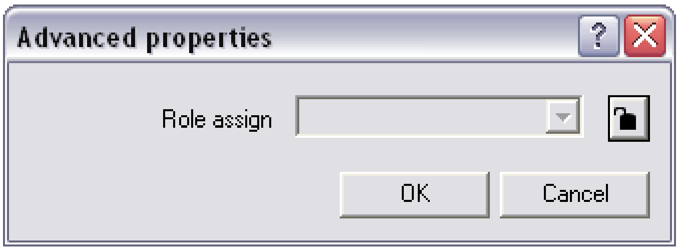
- If you change the number of pages after the device has been created, all the controls you have added to the pages will be deleted. To avoid this problem, you can use a GUI block with the same dimensions as the device, finish the design in the GUI block, then copy the controls over to the device.
- We recommend that you place nTouch 60 devices, Kiosk Layout blocks, Kiosk2Go Layout blocks and Mobile Display blocks on a separate page to the other devices in the project, and do not export the page.

---

4. If you want to change the role to which this device belongs:

   a) Click **Advanced**.

   The **Advanced properties** dialog box is displayed.



Advanced properties

Role assign

   b) Click the 🔓 button to change it to the 🔒 button, and then select a role name in the list.

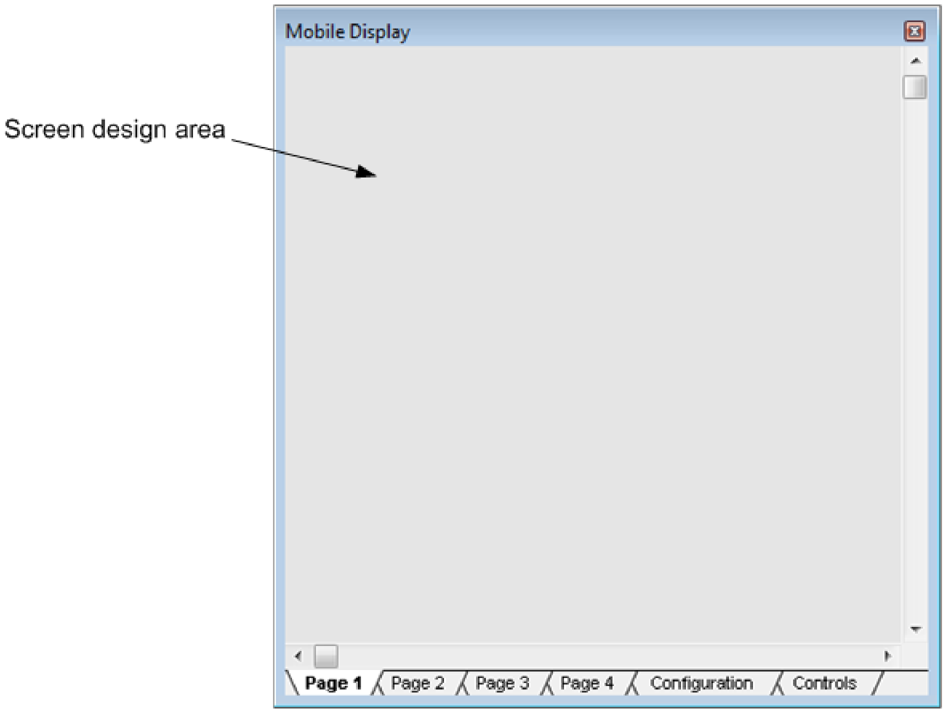
   c) Click **OK**.

5. Click **OK**.

Understanding the layout of the device

Once you have added the Mobile Display device to your design, you can start to add controls to the individual pages, so end users can interact with them.

When you open the Mobile Display device, you will see that NWare has created a number of individual pages automatically.



Each page labeled *Page 1*, *Page 2*, and so on, shows the area that will be visible on the iPad screen – the *screen design area* – at the top. Any controls you place in the screen design area will be visible on the iPad screen when the project is running.

Notes:

- Do not lock pages within the device or the page on which the device is located in your design. You will not be able to access the locked pages from the iPad.
- We recommend that you place nTouch 60 devices, Kiosk Layout blocks, Kiosk2Go Layout blocks and Mobile Display blocks on a separate page to the other devices in the project, and do not export the page.

The *Configuration* tab is used to specify a name for this NWare device, so you can recognize it when you connect to it from the iPad. You can also set up a security pin code that must be entered before the numbered pages can be viewed. On the *Controls* tab are controls you can add to the screen design area on each page.

Connecting the iPad to the NWare Mobile Display device

In order to view the pages in an NWare Mobile Display device on the screen of an iPad, you must set up a connection between the iPad and the device hosting the project.

The iPad will automatically discover all the Mobile Display devices (referred to simply as *mobile displays*) in all the projects running on hosts on the local subnet. However, it will not be able to discover mobile displays running on hosts on other subnets. In order to connect to these displays, you must specify the IP addresses of the hosts manually.

Notes:

- If a static IP address has been specified for a mobile display and it can also be discovered automatically, it will only be listed once on the iPad screen. The static IP address will be used to connect to the mobile display.
- When a static IP address has been configured, but the host cannot be reached, or the mobile display name cannot be found on the host, the mobile display name will be listed on the iPad screen, but it will be grayed out and cannot be selected.

⏩ *To connect the iPad to the NWare Mobile Display device*

1. Deploy the project.
2. Open the **Mobile Display** block.
3. Click the **Configuration** tab.
4. In the **Mobile Display Name** box, type a name to identify this individual iPad device in your project.

   This name will be available for selection in the NWare mobile app on the iPad; when it is selected, the pages in the device block can be viewed on the iPad screen.

   Note: Only one iPad can connect to the iPad device at a time. If a user attempts to connect another iPad, an error will be displayed on the iPad.

5. If you want the iPad user to have to specify a pin code in order to access the screens on the NWare iPad device, type the pin in the **Authentication PIN** boxes.

   This setting is optional.
6. On the iPad, start the NWare mobile application.
7. Press the cog to display the **Mobile Displays** screen.

   There may be a short delay while all available displays are found.
8. If the mobile display you want to use is listed, press the item to connect to it.

   If you want to add a mobile display to the list by specifying the IP address of its host:

   a) Press **Other**.

   b) Next to **IP Address,** specify the IP address of the host (nControl / nTouch 180) running the project.

   c) Next to **Description**, type the name of the mobile display.

   Note: The name is case-sensitive.

   d) Press **Save**.
9. If you have specified a pin code for connecting to the NWare Mobile Display device, type in the pin code.
10. You can now interact with the project on the iPad.

Tips:

- If you want to display the configuration settings when you are using the application, press three fingers anywhere on the screen at the same time, then press the cog button.
- You can set up the iPad to remember the pin number of the NWare iPad device so you won't have to type it next time you connect, by switching on Remember PINs on the configuration screen.
- When you have finished using the NWare mobile application, press the Home button to return to the main Home screen.

In NWare, on the iPad device **Configuration** tab, the IP address of the connected iPad is displayed under **Client IP Address**. The **Client Connected LED** will be lit (green).

Adding an iPhone device to your design

Introduction

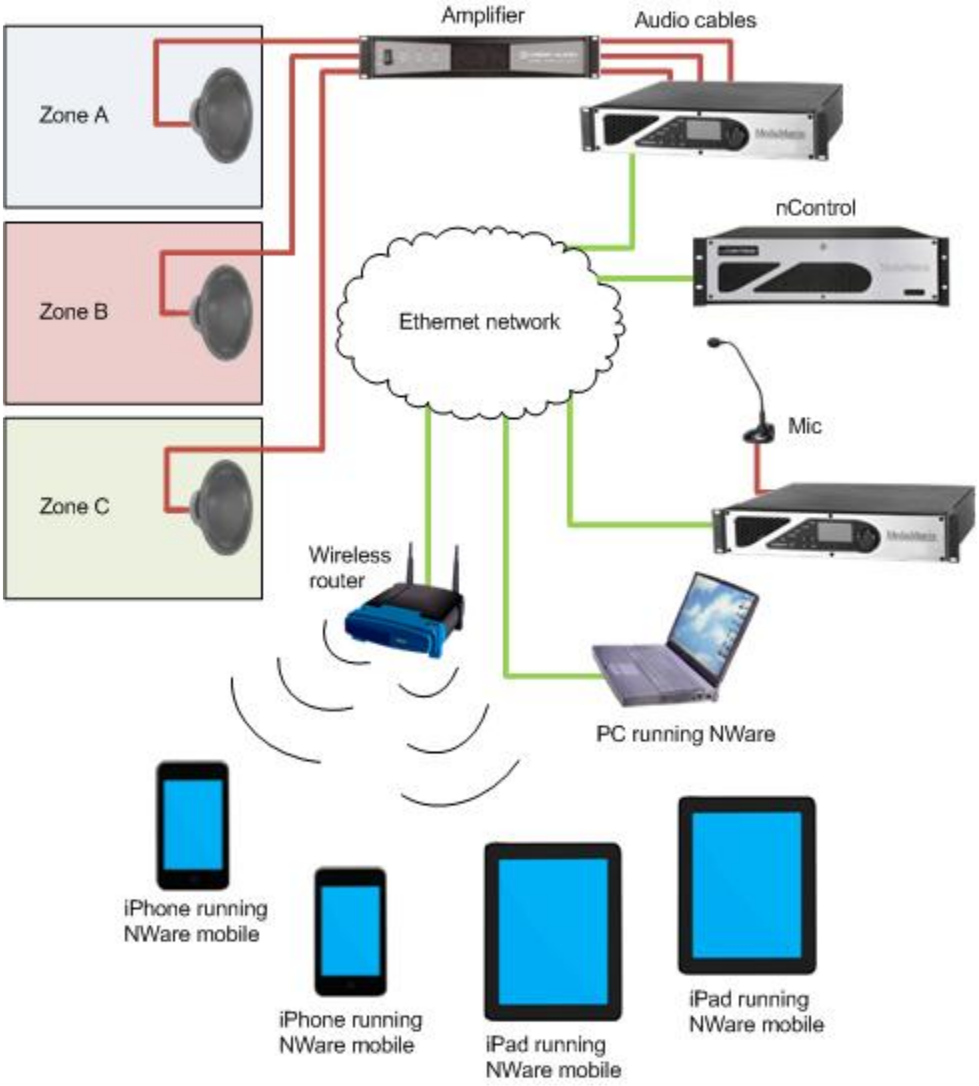
The iPhone is a touch screen device made by Apple Inc. For more information, see *apple.com/iphone http://www.apple.com/iphone*.

The NWare iPhone device allows you to control an NWare project from an iPhone. It works in conjunction with an nControl unit or an nTouch 180 unit, which hosts the Kiosk pages from the NWare project.

Notes:

- The iPhone requires iOS version 4.2 or later.
- The *NWare Mobile* application or *NWare Mobile Designer Edition* application must be installed on the iPhone. These are both available via *iTunes* (*http://itunes.apple.com/gb/artist/peavey-electronics-corporation/id436001313*).
- NWare Mobile Designer Edition is fully functional, but has a time limit. You can use it to connect to a project as many times as you want for one hour; afterwards, connections will be refused until the project has been redeployed or reemulated, or the host (nControl or nTouch 180) has been power cycled.
- If you are going to deploy the project, Apple Bonjour must be installed on the nControl or nTouch 180.
- If you are going to use emulation mode, Apple Bonjour must be installed on the NWare PC. This application is installed automatically when iTunes is installed.
- There is no set limit on the number of iPhone devices that can be added to a project. However, if you add several of these devices and users are interacting with them simultaneously, you may find that the responsiveness of the Kiosk screens is affected.
- We recommend that you place nTouch 60 devices, Kiosk Layout blocks, Kiosk2Go Layout blocks and Mobile Display blocks on a separate page to the other devices in the project, and do not export the page.

In the example below, users can control the audio system via any of the Apple devices. To use a set up like this, your NWare project will need to contain two iPhone Mobile Display devices and two iPad Mobile Display devices, each configured to contact the corresponding Apple device. You will also need an nControl node, two NioNodes and an amplifier device.

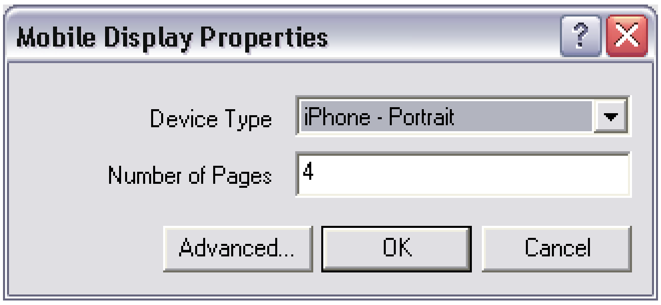

Amplifier
Audio cables
Zone A
Zone B
Zone C
nControl
Ethernet network
Mic
Wireless router
PC running NWare
iPhone running NWare mobile
iPhone running NWare mobile
iPad running NWare mobile
iPad running NWare mobile

The iPhone sections in the NWare User Guide will explain how to add an iPad to your design and configure it. If you require only reference information on the iPhone device, see *iPhone Device* in the *NWare Device Reference*.

Adding the iPhone device

1. In the device tree, expand the **Hardware** folder.
2. Expand the **Mobile Display** folder, then drag an **iPhone Device** node over to the design page.

The **Mobile Display Properties** dialog box is displayed.



Mobile Display Properties

Device Type: iPhone - Portrait

Number of Pages: 4

The **Device Type** is selected automatically.

3. In the **Number of Pages** box, type the number of individual design pages to be displayed on the device when it is created.

   These are the pages that will be visible on the device itself when the project is running.

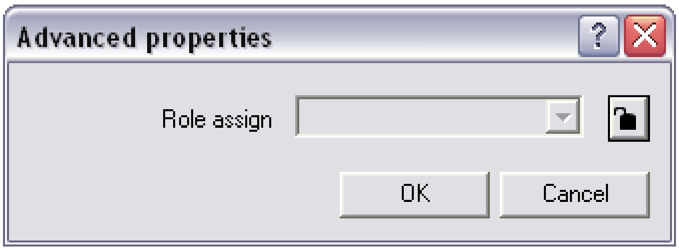
Notes:

- If you change the number of pages after the device has been created, all the controls you have added to the pages will be deleted. To avoid this problem, you can use a GUI block with the same dimensions as the device, finish the design in the GUI block, then copy the controls over to the device.
- We recommend that you place nTouch 60 devices, Kiosk Layout blocks, Kiosk2Go Layout blocks and Mobile Display blocks on a separate page to the other devices in the project, and do not export the page.

4. If you want to change the role to which this device belongs:

   a) Click **Advanced**.

   The **Advanced properties** dialog box is displayed.



Advanced properties

Role assign

   b) Click the 🔓 button to change it to the 🔒 button, and then select a role name in the list.

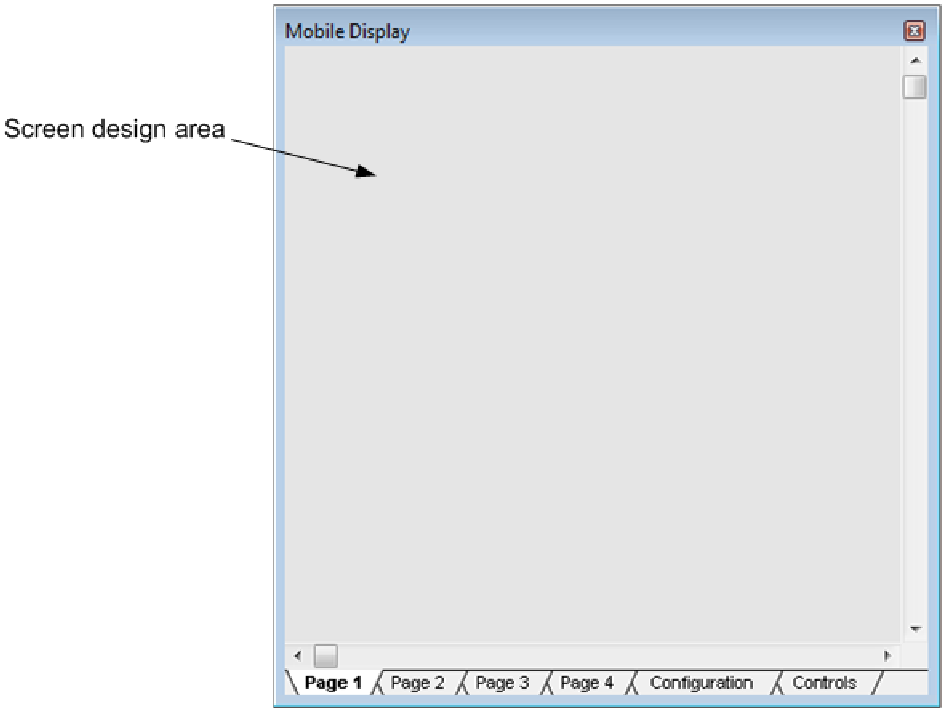
   c) Click **OK**.

5. Click **OK**.

Understanding the layout of the device

Once you have added the Mobile Display device to your design, you can start to add controls to the individual pages, so end users can interact with them.

When you open the Mobile Display device, you will see that NWare has created a number of individual pages automatically.



Each page labeled *Page 1*, *Page 2*, and so on, shows the area that will be visible on the iPhone screen – the *screen design area* – at the top. Any controls you place in the screen design area will be visible on the iPhone screen when the project is running.

Notes:

- Do not lock pages within the device or the page on which the device is located in your design. You will not be able to access the locked pages from the iPhone.
- We recommend that you place nTouch 60 devices, Kiosk Layout blocks, Kiosk2Go Layout blocks and Mobile Display blocks on a separate page to the other devices in the project, and do not export the page.

The *Configuration* tab is used to specify a name for this NWare device, so you can recognize it when you connect to it from the iPhone. You can also set up a security pin code that must be entered before the numbered pages can be viewed. On the *Controls* tab are controls you can add to the screen design area on each page.

Connecting the iPhone to the NWare Mobile Display device

In order to view the pages in an NWare Mobile Display device on the screen of an iPhone, you must set up a connection between the iPhone and the device hosting the project.

The iPhone will automatically discover all the Mobile Display devices (referred to simply as *mobile displays*) in all the projects running on hosts on the local subnet. However, it will not be able to discover mobile displays running on hosts on other subnets. In order to connect to these displays, you must specify the IP addresses of the hosts manually.

Notes:

- If a static IP address has been specified for a mobile display and it can also be discovered automatically, it will only be listed once on the iPhone screen. The static IP address will be used to connect to the mobile display.
- When a static IP address has been configured, but the host cannot be reached, or the mobile display name cannot be found on the host, the mobile display name will be listed on the iPhone screen, but it will be grayed out and cannot be selected.

⏩ *To connect the iPhone to the NWare Mobile Display device*

1. Deploy the project.
2. Open the **Mobile Display** block.
3. Click the **Configuration** tab.
4. In the **Mobile Display Name** box, type a name to identify this individual iPhone device in your project.

   This name will be available for selection in the NWare mobile app on the iPhone; when it is selected, the pages in the device block can be viewed on the iPhone screen.

   Note: Only one iPhone can connect to the iPhone device at a time. If a user attempts to connect another iPhone, an error will be displayed on the iPhone.

5. If you want the iPhone user to have to specify a pin code in order to access the screens on the NWare iPhone device, type the pin in the **Authentication PIN** boxes.

   This setting is optional.
6. On the iPhone, start the NWare mobile application.
7. Press the cog to display the **Mobile Displays** screen.

   There may be a short delay while all available displays are found.
8. If the mobile display you want to use is listed, press the item to connect to it.

   If you want to add a mobile display to the list by specifying the IP address of its host:

   a) Press **Other**.

   b) Next to **IP Address,** specify the IP address of the host (nControl / nTouch 180) running the project.

   c) Next to **Description**, type the name of the mobile display.

   Note: The name is case-sensitive.

   d) Press **Save**.
9. If you have specified a pin code for connecting to the NWare Mobile Display device, type in the pin code.
10. You can now interact with the project on the iPhone.

Tips:

- If you want to display the configuration settings when you are using the application, press three fingers anywhere on the screen at the same time, then press the cog button.
- You can set up the iPhone to remember the pin number of the NWare iPhone device so you won't have to type it next time you connect, by switching on Remember PINs on the configuration screen.
- When you have finished using the NWare mobile application, press the Home button to return to the main Home screen.

In NWare, on the iPhone device **Configuration** tab, the IP address of the connected iPhone is displayed under **Client IP Address**. The **Client Connected LED** will be lit (green).

Adding an nControl node to your design

Introduction

nControl is an industrial network PC server running ControlManager application software. ControlManager provides a platform for MediaMatrix control systems. It can be custom configured to monitor and control multiple NWare projects, consisting of NIONs, CABs, amplifiers, and other devices. Additionally, it can also monitor and control 3rd party devices using SNMP and execute Python scripts.

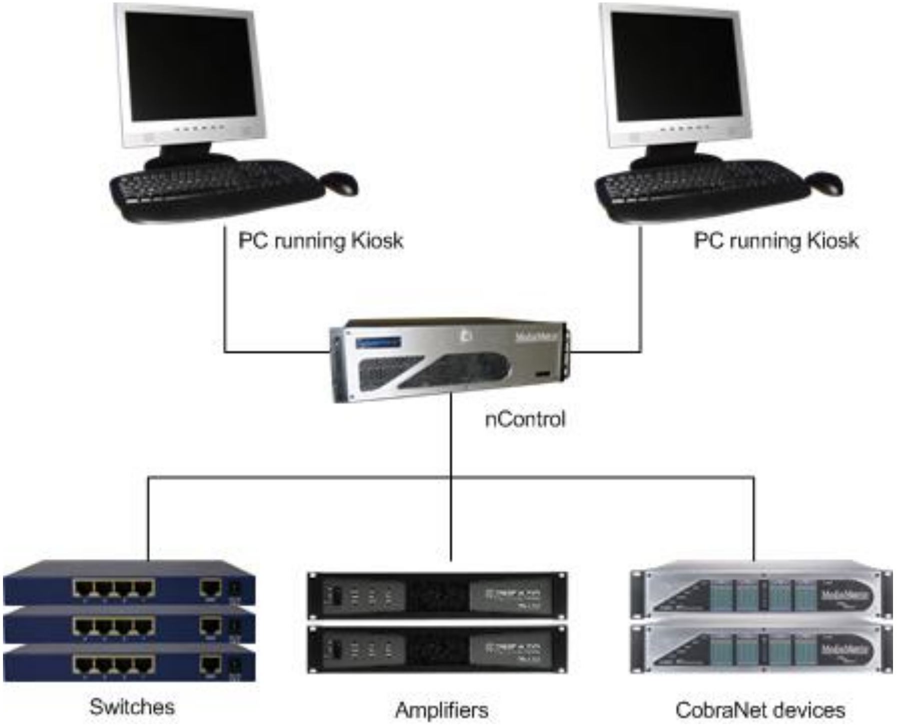
nControl supports RATC and PASHA protocols, allowing you to interact with an NWare project via an Ethernet or serial connection.

After nControl has been installed and given initial configuration settings, it is managed using NWare.

Notes:

- nControl units host projects for control and monitoring, but do not have any DSP capability.
- The available functionality is controlled using a USB hardware key. With this key, you can deploy projects containing devices from the *nControl Devices* section of the devices tree to the nControl. Without the key, you cannot use these devices in your project, but it can contain nTouch 60 devices.
- If you are using a hardware key issued prior to the 1.7.2 release, you will also need an activation key, which is a special code that you specify on the Hardware page of the web interface.
- Using all the capabilities of nControl requires a firm understanding of NWare, SNMP and Python scripting.

When you work with nControl units, your project can contain devices that act as proxies for hardware devices, such as Ethernet switches, UPSs and temperature monitoring systems.



Activation keys

If you create a project containing devices from the *nControl Devices* section of the NWare devices tree, and deploy it to the nControl, it will only function if an activation key has been specified.

If your project does not contain devices from the *nControl Devices* section of the devices tree, an activation key is not required. The nControl can host nTouch 60 projects without a key being specified.

SNMP discovery

When you add an nControl device to your design, you can select **SNMP Discovery**. This allows you to dynamically locate and identify SNMP enabled devices on your network. The discovery process will retrieve a set of information from each device that includes the MAC address, SNMP Object ID, name, location and contact.

The **SNMP Agent Discovery** device allows you to apply search criteria to the discovery information so that you can identify specific devices on the network by their SNMP configuration, rather than by their IP address. This is useful, for example, when you are using dynamic IP address assignment via BOOTP or DHCP.

Note: In order to use the Agent Discovery device, you must select the **SNMP Discovery** check box on the device properties of the nControl or nTouch 180 device.

Switch port discovery

You can use switch port discovery to detect SNMP enabled devices on the network by specifying the switch port they are connected to. In order for this feature to work, you must select the **Switch port support** check box on the **Configure Discovery Server** dialog box when you add the nControl device to your design, and you must also select the **Switch Port Discovery** check box on the **Advanced Properties** of the Agent Discovery device.

You can identify the SNMP devices simply by the switch port number. The IP address of the switch is not required, as the switch itself can be discovered on the network. However, you can specify further identification criteria, such as the IP address, if desired. The number of criteria is specified when you create the Agent Discovery device.

When you deploy or emulate the project and connect to it, on the Agent Discovery device you will be able to specify the port number for the SNMP device and the other criteria. The port number refers to the port number listed in the SNMP Interfaces table on the switch.

Note: Typically, the required port number will be the same as the physical port number used on the switch. However, in the case of Cisco switches, a different port number may be required. This is because of the numbering scheme used in the SNMP interfaces table. If you are using a Cisco switch, we recommend that you check the entries in the table before using switch port discovery.

Understanding the redundancy system

The nControl redundancy system allows a standby unit to replace an active unit on the network if it stops functioning.

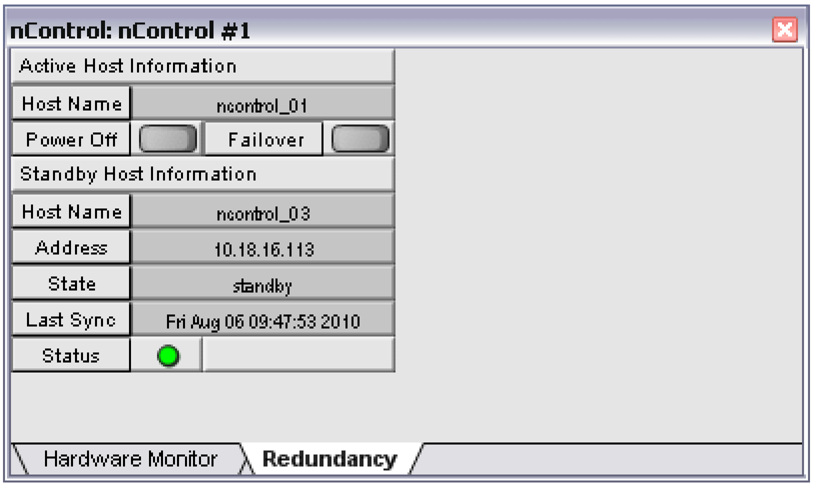
When the system is used, the configuration settings from the active unit are automatically copied to the standby unit to keep them synchronized. The two units are responsible for a single role.

Note: In order to use the redundancy system, you must configure the units via the web interfaces before you use them in your project. For more information, see *Setting up redundancy* in the *nControl Hardware Manual*.

The explanations below assume that you have already added an nControl device to your project.

A system with two peer units

When you deploy the project, you will see the active nControl unit listed on the *Role Deployment* dialog box, but the standby unit will not be listed. When you deploy the project to the active unit, the *Redundancy* tab on the nControl device will show the host name of the active unit under *Active Host Information*. The name and IP address of the standby unit will be displayed under *Standby Host Information*.



nControl: nControl #1

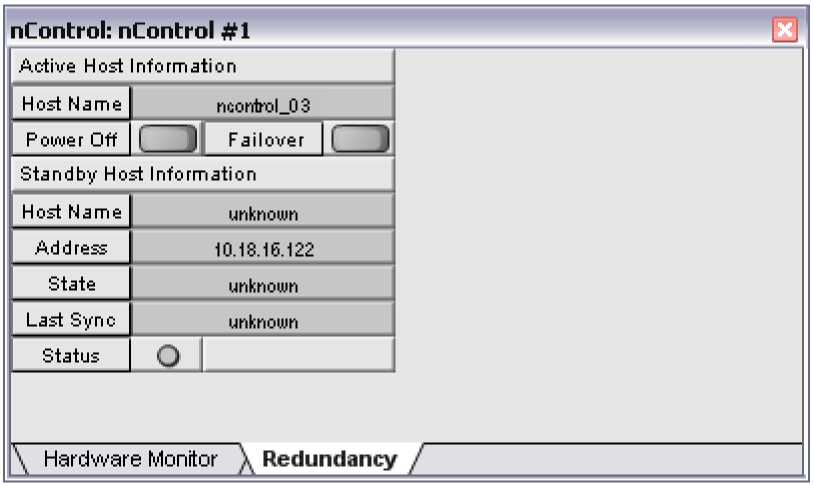
Active Host Information

Host Name	ncontrol_01		
Power Off	○	Failover	○

Standby Host Information

Host Name	ncontrol_03
Address	10.18.16.113
State	standby
Last Sync	Fri Aug 06 09:47:53 2010
Status	🟢

If the active unit fails, after a short delay, the standby unit will start hosting the role; under *Active Host Information*, the *Host Name* field will change to show the name of the standby unit; and under *Standby Host Information*, the *Address* field will show the IP address of the unit that has just failed.



nControl: nControl #1

Active Host Information

Host Name	ncontrol_03		
Power Off	○	Failover	○

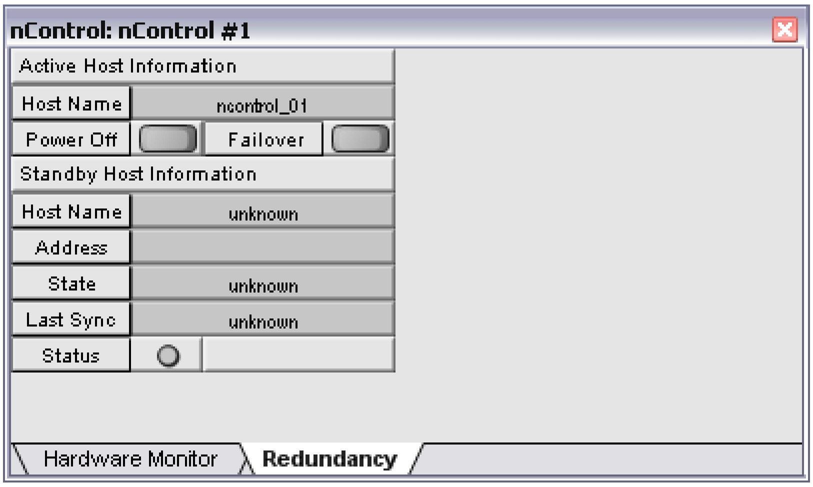
Standby Host Information

Host Name	unknown
Address	10.18.16.122
State	unknown
Last Sync	unknown
Status	○

If the failed unit starts functioning, its details will be shown under *Standby Host Information*. It will not resume hosting the role.

A system with a primary unit and a backup unit

When you deploy the project, you will see the active nControl unit listed on the *Role Deployment* dialog box, but the standby unit will not be listed. When you deploy the project to the active unit, the *Redundancy* tab on the nControl device will show the host name of the active unit under *Active Host Information*, but the *Standby Host Information section* will not show information from the standby unit.



nControl: nControl #1

Active Host Information

Host Name: ncontrol_01

Power Off ○ Failover ○

Standby Host Information

Host Name: unknown

Address:

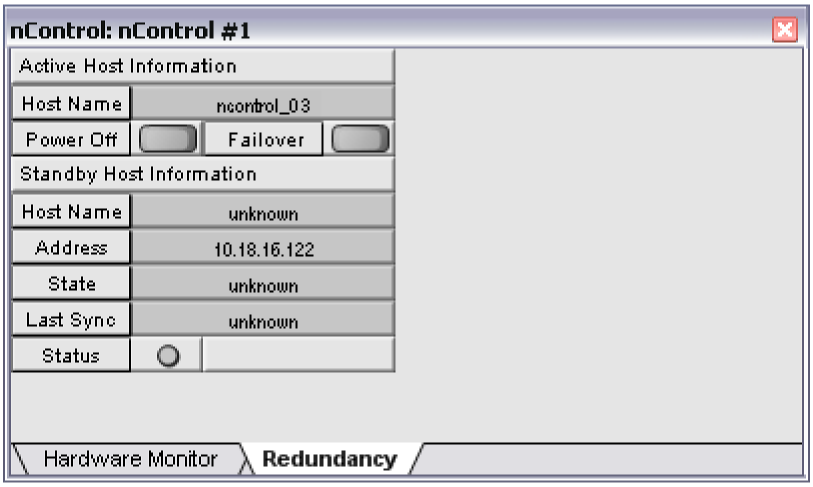
State: unknown

Last Sync: unknown

Status: ○

Hardware Monitor | **Redundancy**

If the active unit fails, after a short delay, the standby unit will start hosting the role; under *Active Host Information*, the *Host Name* field will change to show the name of the standby unit; and under *Standby Host Information*, the *Address* field will show the IP address of the unit that has just failed.



nControl: nControl #1

Active Host Information

Host Name: ncontrol_03

Power Off ○ Failover ○

Standby Host Information

Host Name: unknown

Address: 10.18.16.122

State: unknown

Last Sync: unknown

Status: ○

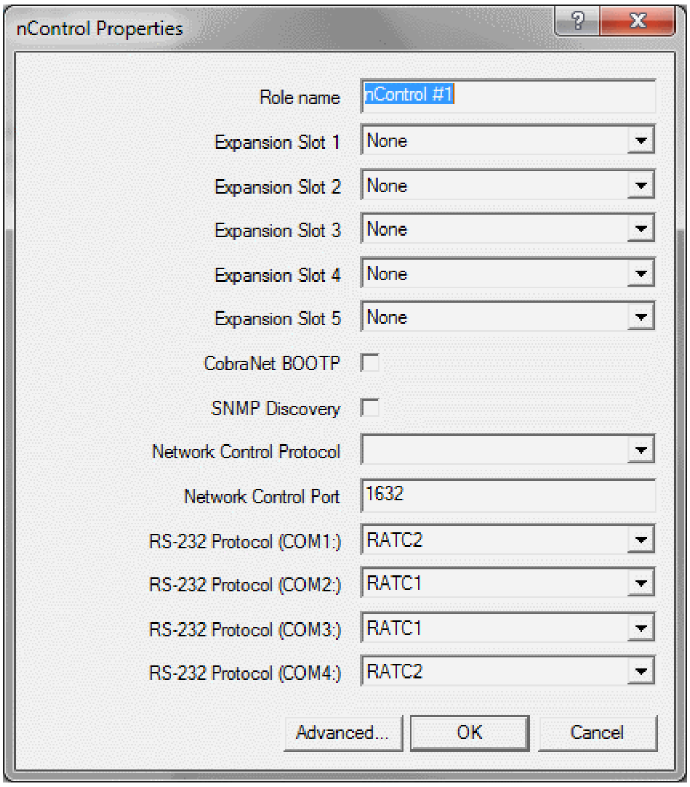
Hardware Monitor | **Redundancy**

If the failed unit starts functioning, it will resume hosting the role and, under *Active Host Information*, the *Host Name* field will change to show the name of the unit that was originally hosting the role. The unit that is now back in standby mode will be restarted.

Adding the nControl node

1. In the device tree, expand the **Hardware** folder.
2. Drag the **nControl** object over to the design page.

   The **nControl Properties** dialog box is displayed.



nControl Properties

Role name: nControl #1

Expansion Slot 1: None

Expansion Slot 2: None

Expansion Slot 3: None

Expansion Slot 4: None

Expansion Slot 5: None

CobraNet BOOTP: □

SNMP Discovery: □

Network Control Protocol:

Network Control Port: 1632

RS-232 Protocol (COM1:): RATC2

RS-232 Protocol (COM2:): RATC1

RS-232 Protocol (COM3:): RATC1

RS-232 Protocol (COM4:): RATC2

3. If you want to change the role name from the default, type a new name in the **Role name** box.

   Note: You must specify a unique role name in the **Role name** box. If you do not do this, an error will be displayed when you deploy the project.

   Devices in your design are allocated to roles (either manually, or automatically by NWare). Roles are then assigned to MediaMatrix hardware devices for processing during deployment.

4. In the **Expansion Slot** boxes, click the names of the cards installed in the nControl unit.

8 ch relay/digital in: PCI-1761	Eight channel digital input/output card. For more information, see *advantech.com* (*http://www.advantech.com.tw/products/8-ch-Relay-and-8-ch-Isolated-Digital-Input-PCI-Card/mod_1-2MLHK7.aspx*).
CobraNet: AudioScience-5416	CobraNet sound card. For information on For more information, see *audioscience.com* (*http://www.audioscience.com/internet/products/sound_cards/asi54xx.htm*).

5. If you want to use the nControl unit as a CobraNet BOOTP server, providing IP addresses to other devices automatically, select the **CobraNet BOOTP** check box.

Notes:

- The BOOTP server is recommended for use with CobraNet devices only.
- It is unlikely that any device will consistently receive the same address after the nControl is rebooted.
- The BOOTP server is designed to support CobraNet devices and uses the more recent BOOTP extensions detailed in RFC 1497. The server is not guaranteed to be backwardly compatible with all BOOTP clients.
- It is possible, although unlikely, that the BOOTP server will allocate IP addresses to non-CobraNet devices on the network that are configured to use DHCP. If this problem occurs, we recommend that you switch off the BOOTP server on the nControl unit and use an alternative server.

6. If you want the nControl unit to be able to discover other SNMP devices on the network, select the **SNMP Discovery** check box.

Notes:

- This feature can generate a large amount of network traffic.
- This feature must be enabled in order to use the SNMP Agent Discovery device in your design.

7. If you intend to control the nControl unit using RATC over the IP network:
   a) Click the RATC version in the **Network Control Protocol** list.
   b) If you want the RATC server on the nControl unit to listen on a different port to the default, type the new port number in the **Network Control Port** box.

   The default is 1632. The valid range is 1-32767.

   Note: If you change the port number, you will have to reconfigure the clients to connect using the new port.

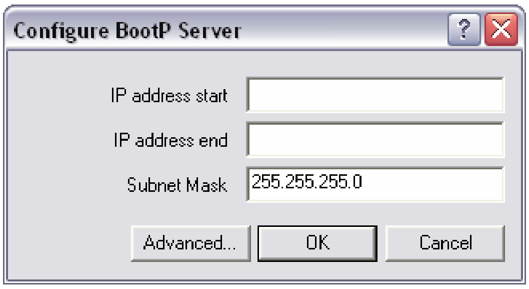
8. If you want to use the RATC or PASHA protocol to control the nControl via its serial ports, click the protocol name in the **RS-232 Protocol** list(s) for the ports you intend to use.

None/Comms processor	Used when the project contains a Comms Processor device, or when you do not want to use the serial port. **Note:** If you select one of the other options – RATC2, for example, then try to use the Comms Processor device, it will not respond as quickly.
RATC2	RATC2 is the improved version of RATC that was introduced with the NION platform, but is also supported by nControl and nTouch 180. Very similar to RATC1, RATC2 introduces shortened commands, and several extra

	functions.
RATC1	RATC1 is the first generation protocol that was used in Classic frame-based MediaMatrix systems. RATC1 for NioNodes, nTouch 180 nodes and nControl nodes is equivalent to what was called RATC in Classic MediaMatrix.
PASHA/PageMatrix	Protocol used with the PageMatrix Command Center. It supports a four character control ID sent with the (T) trigger command.
PASHA/XControl	Provides basic S (Set) PASHA functionality with X-Net2-style trigger commands.
PASHA/Legacy	Designed for projects that contain legacy MediaMatrix nodes that are controlled by external programs using the classic PASHA protocol, as implemented on MainFrames and MiniFrames. When you want to replace the legacy nodes with NioNodes, select the *PASHA/Legacy* option and you will not need to update your external program code. This option ensures all hex values are returned from nodes in lower case instead of upper case to match the original MediaMatrix PASHA protocol.

9. Click **OK**.
10. If you selected the **CobraNet BOOTP** option:

    The **Configure BootP Server** dialog box is displayed.



Configure BootP Server

IP address start

IP address end

Subnet Mask 255.255.255.0

a) In the **IP address start** box, type the start IP address for the range that can be allocated to clients requesting IP addresses.

b) In the **IP address end** box, type the end IP address for the range that can be allocated to clients requesting IP addresses.

c) In the **Subnet Mask** box, type the subnet mask for the IP address range.

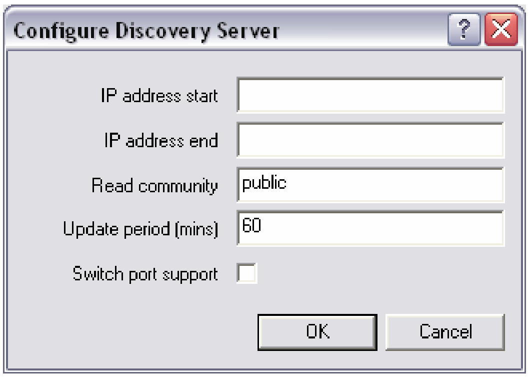
d) If you want to change the maximum number of IP address requests the BOOTP server can process simultaneously, click **Advanced**, type a new number in the **Threads** box, and then click **OK**.

Tip: We recommend that you leave the thread count setting set to 2.

e) Click **OK**.

11. If you selected the **SNMP Discovery** option:

The **Configure Discovery Server** dialog box is displayed.



Configure Discovery Server

IP address start

IP address end

Read community: public

Update period (mins): 60

Switch port support □

a) In the **IP address start** box, type the start IP address for the range that will be scanned during the discovery process.

If you want to use a broadcast address, which will provide better performance when scanning a large range of addresses, type the broadcast address.

When a large number of devices are discovered, using a broadcast address will greatly reduce the impact on the network. The format of a broadcast address is x.x.x.255. This will broadcast to the entire /24 subnet. You can also use x.x.255.255 to broadcast to a /16 subnet, but the impact of addressing such a large number of devices must be considered carefully.

Note: Not all SNMP devices will respond to a broadcast addresses. Check that the devices you are using are compatible.

b) In the **IP address end** box, type the end IP address for the range that will be scanned during the discovery process.

If you are using a broadcast address, leave this box blank.

c) In the **Read community** box, type the read community string (password) for accessing devices on the network during the discovery process.

d) In the **Update period (mins)** box, type the interval to use between scans for devices on the network.

e) If you want to enable switch port discovery, select the **Switch port support** check box.

This feature allows devices to be detected via the switch port to which they are connected. Both the switch and the device must support SNMP. Furthermore, the device must respond to the SNMP discovery process.

f) Click **OK**.

Forcing a failover to the standby unit

The *Redundancy tab* on the nControl device has a *Power Off* button that will power down the active unit, and a *Failover* button that will restart the active unit. These are useful for testing the redundancy system.

⏩ *To restart the active unit*

- Click **Failover**.

⏩ *To power down the active unit*

- Click **Power Off**.

Using Crest amplifiers in your design

NWare allows you to control and monitor Crest CKi and Ci series amplifiers. For detailed information on these amplifiers, refer to the *Crest website* (*http://www.peaveycommercialaudio.com/products.cfm/cr/Power-Amplifiers*).

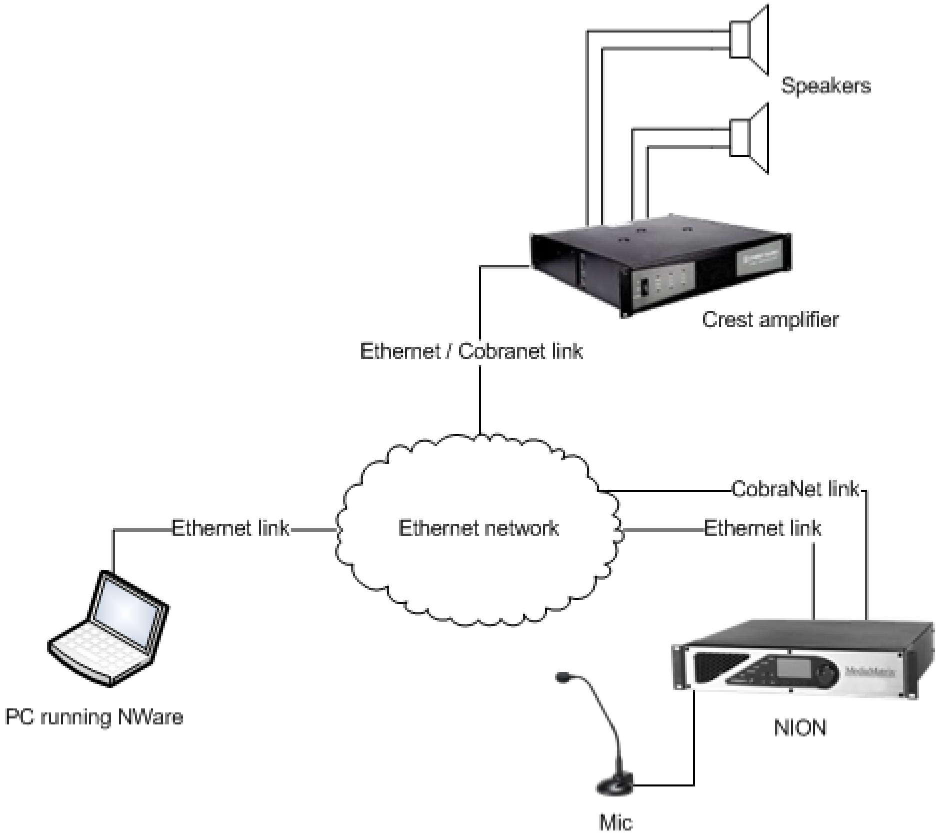
Crest Audio offers two features to monitor system operation and help systems comply with safety regulations: *Real Time Load Monitoring (RTLM)* and *Speaker Tone Sensing (STS)*. Both are available in addition to the monitoring systems built in to the amplifiers. RTLM is available via the NWare amplifier devices, which are described in the section *Amplifiers* in the *NWare Device Reference*. For detailed information on how both RTLM and STS work, see *Monitoring Crest amplifiers* (on page 112).

By default, a check box called **Live Taper** is selected on the device properties of each amplifier device you add to your design. This feature is designed to make it easier to make fine adjustments at higher gain levels (by decreasing the rate at which the gain changes when it is near the maximum). You can clear this check box to disable this feature.

To start working with Crest amplifiers, complete one of the procedures following this section. It is assumed that the NioNode (or nControl node or nTouch 180 node), the amplifiers and the PC running NWare are all connected to the same network, as shown in the diagram below.

Notes:

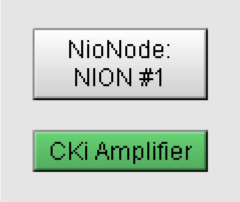
- Before you use an amplifier with NWare, it is important to check that it is installed correctly and is running the latest firmware. If you need to upgrade the firmware, use the Crest NexSys software to do this.
- The Crest Ci amplifier can receive up to four eight channel bundles, but the current version of NWare only supports one eight channel bundle.
- You cannot control and monitor amplifiers by running NWare in emulation mode. You must deploy your project.
- If you are using a NioNode to host the project, control signals will be transferred from the NioNode to the amplifiers via the LAN port on the NioNode (and not the CobraNet port). This means you will need to configure a router to route the control signals from the control network to the CobraNet network, or use an nControl node to host the project and connect it to both the control network and the CobraNet network.
- When CobraNet and Ethernet devices share the same network, the use of managed switches is strongly recommended. CobraNet traffic can cause other traffic to be transferred very slowly, unless traffic management systems are in place, such as VLANs. For more information, refer to the section on NION use cases in the NION Hardware Manual.



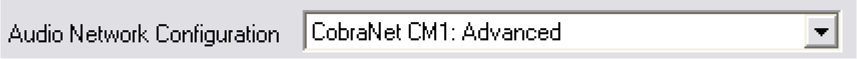
Adding a CKi amplifier to your design

1. Drag a **Crest CKi Amplifier** object from the device tree over to the page.

   It is assumed that your project already contains a NioNode or nControl node. This procedure uses a NioNode as an example.



2. Display the NioNode properties, then specify **CobraNet CM1:Advanced** as the **Audio Network Configuration** setting.



Audio Network Configuration CobraNet CM1: Advanced

3. On the toolbar, click the **Emulate** button.
4. Open the **Cki Amplifier** block.

   Here is a summary of the available fields.

Channels	
Field name	**Meaning**
General	
ID	The two octets of the amplifier ID number in hex format. This is shown on the display at the back of the unit.
IP Address	The IP address of the amplifier. This is shown on the display at the back of the unit.
Power	Lit (green) when the amplifier is switched on. If remote mode is selected on the amplifier, you can switch it on or off using the Power button within NWare.
Rmte (Remote)	Lit (green) when the amplifier is running in remote mode.
Link	Lit (green) when a link is established between NWare and the device across the network.
Fault	Lit (red) when a fault condition, e.g. thermal or short circuit, has been detected by the amplifier. Please refer to the amplifier documentation for information on fault conditions.
CobraNet	
Input (A / D)	Selects either analog audio input using Phoenix connectors or digital audio input from the CobraNet network.
Bundle Num.	The audio bundle number for the device. This must correspond with the bundle number specified for the NION associated with the amplifier. You can then make connections to the amplifier using the NION flyoffs.
Channel x	
Fault	Lit (red) if the amplifier enters ACL or IGM or experiences a thermal, DC voltage, or short circuit fault.
Name	A name for the channel. This label can be used to improve the clarity of the user interface for the end user.
Gain	Adjusts the amount of amplification or attenuation to be applied to the channel. Range is -97 to 0 dB.
Mute	Mutes the channel output. The LED is lit (red) when the

	channel is muted.
Alerts	
User Temp	The user-specified upper limit for the channels' heat sink temperature. If the limit is exceeded, the light will be lit (yellow).
IGM	Lit (yellow) when the Instantaneous Gain Modulation (IGM) circuit detects a load that overstresses the output stage. The channel gain is automatically adjusted to a safe level.
ACL	Lit (yellow) when the Active Clip Limiting (ACL) circuit is active because a channel has reached the clipping point.
Cobranet	
Subchannel	The subchannel numbers for the audio channels.
Faults	
Temperature	Lit (red) when a channel’s heat sink temperature reaches 75°C (which may indicate an obstructed air supply). The channel will independently protect itself by disconnecting its load and shutting down until it has cooled.
DC	Lit (red) when the amplifier channel detects DC voltage or subsonic frequencies at a channel output. The respective output relay will immediately open to prevent loudspeaker damage.
RTLM (Real Time Load Monitoring)	Lit (red) when the load for the channel exceeds the upper or lower limit. If this indicator is lit, and the RTLM training process has successfully completed, it indicates that the load on the amplifier has altered significantly. This could be because a loud speaker has blown, for example.
RTLM	
Upper	The upper limit for the load level. If the limit is exceeded, the Fault light will be lit.
Lower	The lower limit for the load level. If the limit is exceeded, the Fault light will be lit.

Advanced	
Field name	**Meaning**
Subnet Mask	The subnet mask for the Ethernet / CobraNet network.
Gateway	The IP address of the router. If there is no router, this will be set to 0.0.0.0.
Gain/Sensitivity	Position of the Gain switch at the rear of the amplifier. This can be set to give constant sensitivity, or a constant gain of x20 (26 dB) or x40 (32 dB). Display only.
Version	Firmware version running on the amplifier. Display only.
AC Power Info	
Volt	Voltage provided to the amplifier. Display only.
Amp	Current provided to the amplifier. Display only.
Watt	Input power provided to the amplifier. Display only.
RTLM	
Channel	The channel number.
Train	Starts the training process for a channel. Used once speakers have been connected to the outputs and audio is running through the amplifier. The process takes between 5 and 10 minutes. The Training light will switch off automatically when the process is complete. **Notes:** ▪ If the load level is too low, the training process will not complete.

	▪ If you are using a Ci amplifier, the training process will not complete successfully unless all the channels are loaded and driven appropriately.
Abort	Aborts the training process for a channel. Use this button to stop the training process if more than 10 minutes have elapsed since the process was started, and the Training light is still lit.

5. Type the amplifier id in the **Id** boxes.

   This is shown at the back of the unit.

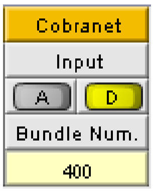
6. Type the IP address in the **IP Address** boxes.

   This is shown at the back of the unit.

Tip: The subnet mask and gateway settings are read from the amplifier automatically when it is contacted by NWare. These can be changed on the **Advanced** tab if necessary.

If the network settings are correct, the **Link** light will be lit (green) and information will be received from the amplifier, including the gain setting, temperature etc., and displayed in the fields of the **Cki Amplifier** block.

7. Under **Cobranet** / **Input**, click the **D** button to select digital audio input from the CobraNet network (via the NION).



8. In the **Bundle Num.** box, type a bundle number.

   Each bundle can manage between 1 and 8 channels. The range is 0 to 65,535 (0 is off). For multicast bundles, the number must be in the range 1 to 255. For unicast bundles, the number must be in the range 256 to 65,279. Numbers in the range 65280 to 65535 are reserved and cannot be used.

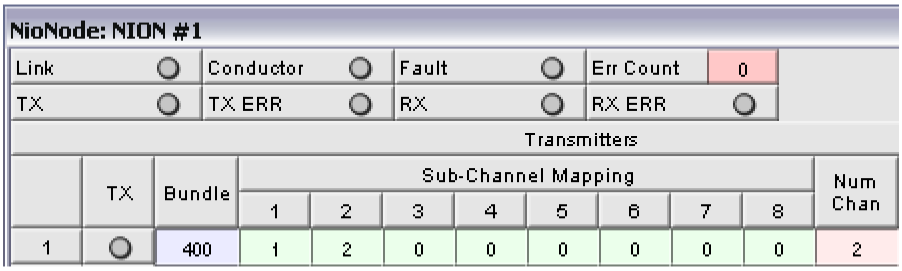
9. In the **Subchannel** boxes, specify the subchannel numbers to use when referring to the amplifier channels (A and B). These can simply be 1 and 2.
10. Open the **NioNode** block and, in the first available **Bundle** box in the **Transmitters** section, specify the bundle number you specified in the **CKi Amplifier** block.
11. In the corresponding **Sub-Channel Mapping** boxes to the right of the **Bundle** box, specify the CM-1 output numbers (1-32) you want to send to the amplifier.

Note: Ci and Cki amplifier default sub-channel mapping is one for one. Sub-channel 1 matches amplifier channel 1, sub-channel 2 matches amplifier channel 2, sub-channel 3 matches amplifier channel 3, and so on.

Any of the NION CM-1 audio output numbers may be entered in any sub-channel box. Click the sub-channel box and type the CM-1 output number you want to use.

12. In the **Num Chan** (number of channels) box, type the actual number of audio channels in the CobraNet bundle.

Specifying this value avoids overloading the network with unnecessary data.



NioNode: NION #1

Link ○	Conductor ○	Fault ○	Err Count 0
TX ○	TX ERR ○	RX ○	RX ERR ○

Transmitters

	TX	Bundle	Sub-Channel Mapping								Num Chan
			1	2	3	4	5	6	7	8	
1	○	400	1	2	0	0	0	0	0	0	2

Tip: Values in the **Sub-Channel Mapping** boxes may be changed on a dynamic basis. This allows any hardware channel number to be mapped to any bundle sub-channel(s) at any time, greatly expanding signal routing flexibility. However, only users familiar with advanced CobraNet practices should use this feature.

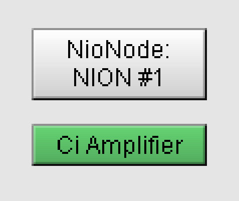
13. On the toolbar, click the **Disemulate** button.
14. In the utility frame, click the **Flyoffs** tab, then drag flyoffs for the CM-1 output channels you are using over to the page.
15. For test purposes, wire the flyoffs to an audio generator - a sine generator, for example.
16. Save the project, then deploy it.

    Audio can now pass between the NioNode and the Crest amplifier(s).

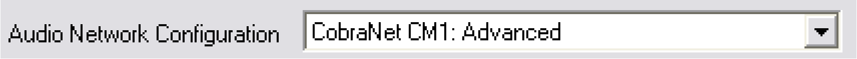
Adding a Ci amplifier to your design

1. Drag a **Crest Ci Amplifier** object (4 channels or 8 channels) from the device tree over to the page.

   It is assumed that your project already contains a NioNode or nControl node. This procedure uses a NioNode as an example.



2. Display the NioNode properties, then specify **CobraNet CM1:Advanced** as the **Audio Network Configuration** setting.



Audio Network Configuration: CobraNet CM1: Advanced

3. On the toolbar, click the **Emulate** button.
4. Open the **Ci Amplifier** block.

   Here is a summary of the available fields.

Channels	
Field name	**Meaning**
General	

ID	The two octets of the amplifier ID number in hex format. This is shown on the display at the back of the unit.
IP Address	The IP address of the amplifier. This is shown on the display at the back of the unit.
Power	Lit (green) when the amplifier is switched on. If remote mode is selected on the amplifier, you can switch it on or off using the Power button within NWare.
Rmte (Remote)	Lit (green) when the amplifier is running in remote mode.
Link	Lit (green) when a link is established between NWare and the device across the network.
Fault	Lit (red) when a fault condition, e.g. thermal or short circuit, has been detected by the amplifier. Please refer to the amplifier documentation for information on fault conditions.
CobraNet	
Input (A / D)	Selects either analog audio input using Phoenix connectors or digital audio input from the CobraNet network.
Bundle Num.	The audio bundle number for the device. This must correspond with the bundle number specified for the NION associated with the amplifier. You can then make connections to the amplifier using the NION flyoffs.
Channel x	
Fault	Lit (red) if the amplifier enters ACL or IGM or experiences a thermal, DC voltage, or short circuit fault.
Input level	Input level in dBu.
Output level	Full scale output level in dBu. Zero dB on the meter is reached at full amp power (assuming an 8 Ohm load). When in 70V mode, full power is reached at about 5dB on the output meter.
Name	A name for the channel. This label can be used to improve the clarity of the user interface for the end user.
Gain	Adjusts the amount of amplification or attenuation to be applied to the channel. Range is -97 to 0 dB.

Mute	Mutes the channel output. The LED is lit (red) when the channel is muted.
Alerts	
User Temp	The user-specified upper limit for the channels' heat sink temperature. If the limit is exceeded, the light will be lit (yellow).
IGM	Lit (yellow) when the Instantaneous Gain Modulation (IGM) circuit detects a load that overstresses the output stage. The channel gain is automatically adjusted to a safe level.
ACL	Lit (yellow) when the Active Clip Limiting (ACL) circuit is active because a channel has reached the clipping point.
Cobranet	
Subchannel	The subchannel numbers for the audio channels.
Faults	
Temperature	Lit (red) when a channel’s heat sink temperature reaches 75°C (which may indicate an obstructed air supply). The channel will independently protect itself by disconnecting its load and shutting down until it has cooled.
DC	Lit (red) when the amplifier channel detects DC voltage or subsonic frequencies at a channel output. The respective output relay will immediately open to prevent loudspeaker damage.
RTLM (Real Time Load Monitoring)	Lit (red) when the load for the channel exceeds the upper or lower limit. If this indicator is lit, and the RTLM training process has successfully completed, it indicates that the load on the amplifier has altered significantly. This could be because a loud speaker has blown, for example.
RTLM	
Upper	The upper limit for the load level. If the limit is exceeded, the Fault light will be lit.
Lower	The lower limit for the load level. If the limit is exceeded, the Fault light will be lit.

Advanced	
Field name	**Meaning**
Subnet Mask	The subnet mask for the Ethernet / CobraNet network.
Gateway	The IP address of the router. If there is no router, this will be set to 0.0.0.0.
Version	Firmware version running on the amplifier. Display only.
AC Power Info	
Volt	Voltage provided to the amplifier. Display only.
Amp	Current provided to the amplifier. Display only.
Watt	Input power provided to the amplifier. Display only.
RTLM	
Channel	The channel number.
Norm Load	Impedance of the channel normalized to 100 ohms.
Char Load	Full spectrum load impedance. The value can change (along the impedance curve) according to the dominant frequency content of the program material. However, if you feed a sine wave into the amplifier, a more accurate impedance can be measured (at that frequency and level).
Enabled	Lit (green) when RTLM reporting is enabled.
Active	Lit (green) when training is in progress and the load on the channel is actively being checked.

Below Threshold	Lit (green) when the channel voltage is below the minimum threshold required to measure load data.
Progress	Indicates the percentage completion of load data collection during training and after a reboot.
Enable	Switches RTLM reporting on. Enables collection of RTLM data points.
Disable	Switches RTLM reporting off. Stops the training process, keeping previous training data intact.
Train	Starts the training process for a channel. Used once speakers have been connected to the outputs and audio is running through the amplifier. The process takes between 5 and 10 minutes. The Training light will switch off automatically when the process is complete. **Notes:** ▪ If the load level is too low, the training process will not complete. ▪ If you are using a Ci amplifier, the training process will not complete successfully unless all the channels are loaded and driven appropriately.
Abort	Aborts the training process for a channel. Use this button to stop the training process if more than 10 minutes have elapsed since the process was started, and the Training light is still lit.

5. Type the amplifier id in the **Id** boxes.

   This is shown at the back of the unit.

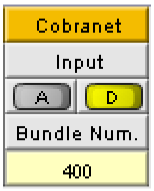
6. Type the IP address in the **IP Address** boxes.

   This is shown at the back of the unit.

Tip: The subnet mask and gateway settings are read from the amplifier automatically when it is contacted by NWare. These can be changed on the **Advanced** tab if necessary.

If the network settings are correct, the **Link** light will be lit (green) and information will be received from the amplifier, including the gain setting, temperature etc., and displayed in the fields of the **Cki Amplifier** block.

7. Under **Cobranet** / **Input**, click the **D** button to select digital audio input from the CobraNet network (via the NION).



8. In the **Bundle Num.** box, type a bundle number.

   Each bundle can manage between 1 and 8 channels. The range is 0 to 65,535 (0 is off). For multicast bundles, the number must be in the range 1 to 255. For unicast bundles, the number must be in the range 256 to 65,279. Numbers in the range 65280 to 65535 are reserved and cannot be used.
9. In the **Subchannel** boxes, specify the subchannel numbers to use when referring to the amplifier channels.

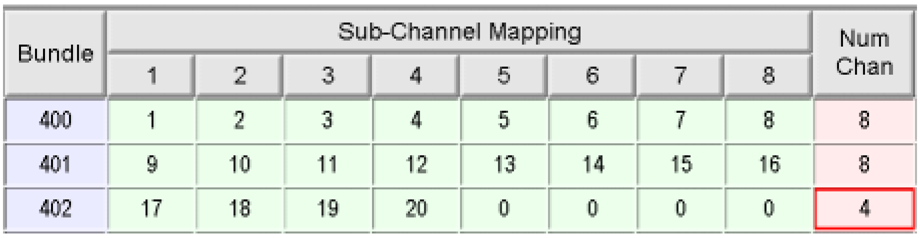
   The channel numbers can simply be 1, 2, 3 etc.
10. Open the **NioNode** block and, in the first available **Bundle** box in the **Transmitters** section, specify the bundle number you specified in the **Ci Amplifier** block.
11. In the corresponding **Sub-Channel Mapping** boxes to the right of the **Bundle** box, specify the CM-1 output numbers (1-32) you want to send to the amplifier.

Note: Ci and Cki amplifier default sub-channel mapping is one for one. Sub-channel 1 matches amplifier channel 1, sub-channel 2 matches amplifier channel 2, sub-channel 3 matches amplifier channel 3, and so on.

Any of the NION CM-1 audio output numbers may be entered in any sub-channel box. Click the sub-channel box and type the CM-1 output number you want to use.

12. In the **Num Chan** (number of channels) box, type the actual number of audio channels in the CobraNet bundle.

    Specifying this value avoids overloading the network with unnecessary data.



Bundle	Sub-Channel Mapping								Num Chan
	1	2	3	4	5	6	7	8	
400	1	2	3	4	5	6	7	8	8
401	9	10	11	12	13	14	15	16	8
402	17	18	19	20	0	0	0	0	4

Tip: Values in the **Sub-Channel Mapping** boxes may be changed on a dynamic basis. This allows any hardware channel number to be mapped to any bundle sub-channel(s) at any time, greatly expanding signal routing flexibility. However, only users familiar with advanced CobraNet practices should use this feature.

13. On the toolbar, click the **Disemulate** button.
14. In the utility frame, click the **Flyoffs** tab, then drag flyoffs for the CM-1 output channels you are using over to the page.
15. For test purposes, wire the flyoffs to an audio generator - a sine generator, for example.
16. Save the project, then deploy it.

    Audio can now pass between the NioNode and the Crest amplifier(s).

Adding an amplifier stack to your design

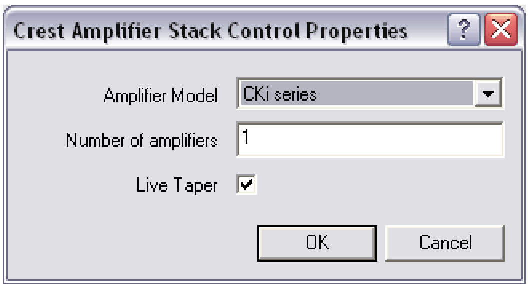
If you are working with multiple amplifiers that are located in the same local area of the network, you can create an amplifier stack and specify settings for these devices at the same time.

It is assumed that your project already contains a NioNode or nControl node. This procedure uses a NioNode as an example.

⏩ *To add an amplifier stack to your design*

1. Drag a **Crest Amplifier Stack** object from the device tree over to the page.

   The **Crest Amplifier Stack Control Properties** dialog box is displayed.



Crest Amplifier Stack Control Properties

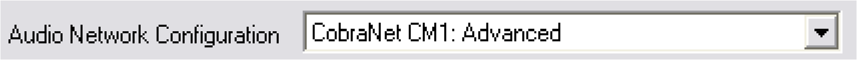
Amplifier Model: CKi series

Number of amplifiers: 1

Live Taper ☑

2. In the **Amplifier Model** list, click the amplifier model name.
3. In the **Number of amplifiers** box, type the number of amplifiers in the stack.
4. If you want to make it easier to make fine adjustments at higher gain levels (by decreasing the rate at which the gain changes when it is near the maximum), leave the **Live Taper** check box checked.
5. Click **OK**.

   The **CKi Amplifier Stack** block or **Ci Amplifier Stack** block will be created on the page.
6. Display the NioNode properties, then specify **CobraNet CM1:Advanced** as the **Audio Network Configuration** setting.



Audio Network Configuration: CobraNet CM1: Advanced

7. On the toolbar, click the **Emulate** button.
8. Open the **CKi Amplifier Stack** block or **Ci Amplifier Stack** block.

   Here is a summary of the available fields for each individual amplifier.

Main	
Field name	**Meaning**
Amp	The amplifier number in the stack.
ID	The two octets of the amplifier ID number in hex format. This is shown on the display at the back of the unit.
IP Address	The IP address of the amplifier. This is shown on the display at the back of the unit.
Link	Lit (green) when a link is established between NWare and the amplifier across the network.
Fault	Lit (red) when a fault condition, e.g. thermal or short circuit, has been detected by the amplifier. Please refer to the amplifier documentation for information on fault conditions.

Rmte (Remote)	Lit (green) when the amplifier is running in remote mode.
Power	Lit (green) when the amplifier is switched on. If remote mode is selected on the amplifier, you can switch it on or off using the Power button within NWare.
Mute (number/letter)	Mutes the channel output. The LED is lit (red) when the channel is muted.

Channels	
Field name	**Meaning**
Amp	The amplifier number in the stack.
ID	The two octets of the amplifier ID number in hex format. This is shown on the display at the back of the unit.
IP Address	The IP address of the amplifier. This is shown on the display at the back of the unit.
Link	Lit (green) when a link is established between NWare and the amplifier across the network.
Fault	Lit (red) when a fault condition, e.g. thermal or short circuit, has been detected by the amplifier. Please refer to the amplifier documentation for information on fault conditions.
Rmte (Remote)	Lit (green) when the amplifier is running in remote mode.
Power	Lit (green) when the amplifier is switched on. If remote mode is selected on the amplifier, you can switch it on or off using the Power button within NWare.
Chan	Channel letter or number.

Channel Name	A name for the channel. This label can be used to improve the clarity of the user interface for the end user.
Cobranet Sub Channel	The subchannel numbers for the audio channels.
Input Level (Ci amps only)	Input level in dBu.
Gain	Adjusts the amount of amplification or attenuation to be applied to the channel. Range is -97 to 0 dB.
Mute	Mutes the channel output. The LED is lit (red) when the channel is muted.
Output Level (Ci amps only)	Full scale output level in dBu. Zero dB on the meter is reached at full amp power (assuming an 8 Ohm load). When in 70V mode, full power is reached at about 5dB on the output meter.
Temp	Channel heat sink temperature.
Ther	Lit (red) when a channel’s heat sink temperature reaches 75°C (which may indicate an obstructed air supply). The channel will independently protect itself by disconnecting its load and shutting down until it has cooled.
DC	Lit (red) when the amplifier channel detects DC voltage or subsonic frequencies at a channel output. The respective output relay will immediately open to prevent loudspeaker damage.
RTLM (Real Time Load Monitoring)	Lit (red) when the load for the channel exceeds the upper or lower limit. If this indicator is lit, and the RTLM training process has successfully completed, it indicates that the load on the amplifier has altered significantly. This could be because a loud speaker has blown, for example.
User Temp	The user-specified upper limit for the channels' heat sink temperature. If the limit is exceeded, the light will be lit (yellow).
IGM Alert	Lit (yellow) when the Instantaneous Gain Modulation (IGM) circuit detects a load that overstresses the output stage. The channel gain is automatically adjusted to a safe level.
ACL Alert	Lit (yellow) when the Active Clip Limiting (ACL) circuit is active because a channel has reached the clipping point.

RTLM (CKi amp)	
Field name	**Meaning**
Amp	The amplifier number in the stack.
ID	The two octets of the amplifier ID number in hex format. This is shown on the display at the back of the unit.
IP Address	The IP address of the amplifier. This is shown on the display at the back of the unit.
Link	Lit (green) when a link is established between NWare and the amplifier across the network.
Fault	Lit (red) when a fault condition, e.g. thermal or short circuit, has been detected by the amplifier. Please refer to the amplifier documentation for information on fault conditions.
Rmte (Remote)	Lit (green) when the amplifier is running in remote mode.
Power	Lit (green) when the amplifier is switched on. If remote mode is selected on the amplifier, you can switch it on or off using the Power button within NWare.
Chan	Channel letter or number.
Upper	The upper limit for the load level. If the limit is exceeded, the Fault light will be lit.
Lower	The lower limit for the load level. If the limit is exceeded, the Fault light will be lit.
Train	Starts the training process. Used once speakers have been connected to the outputs and audio is running through the amplifier. The process takes between 5 and 10 minutes. The Training light will switch off automatically when the process is complete. **Note:** If the load level is too low, the training process will not complete.
Abort	Aborts the training process. Use this button to stop the

	training process if more than 10 minutes have elapsed since the process was started, and the Training light is still lit.
Training	Lit (green) when Training is in progress.
RTLM (Ci amp)	
Field name	**Meaning**
Amp	The amplifier number in the stack.
ID	The two octets of the amplifier ID number in hex format. This is shown on the display at the back of the unit.
IP Address	The IP address of the amplifier. This is shown on the display at the back of the unit.
Link	Lit (green) when a link is established between NWare and the amplifier across the network.
Fault	Lit (red) when a fault condition, e.g. thermal or short circuit, has been detected by the amplifier. Please refer to the amplifier documentation for information on fault conditions.
Rmte (Remote)	Lit (green) when the amplifier is running in remote mode.
Power	Lit (green) when the amplifier is switched on. If remote mode is selected on the amplifier, you can switch it on or off using the Power button within NWare.
Chan	Channel letter or number.
Enable	Switches RTLM reporting on. Enables collection of RTLM data points.
Disable	Switches RTLM reporting off. Stops the training process, keeping previous training data intact.
Norm Load	Impedance of the channel normalized to 100 ohms.

Char Load	Full spectrum load impedance. The value can change (along the impedance curve) according to the dominant frequency content of the program material. However, if you feed a sine wave into the amplifier, a more accurate impedance can be measured (at that frequency and level).
Enabled	Lit (green) when RTLM reporting is enabled.
Active	Lit (green) when training is in progress and the load on the channel is actively being checked.
Below Threshold	Lit (green) when the channel voltage is below the minimum threshold required to measure load data.
Progress	Indicates the percentage completion of load data collection during training and after a reboot.
Upper	The upper limit for the load level. If the limit is exceeded, the Fault light will be lit.
Lower	The lower limit for the load level. If the limit is exceeded, the Fault light will be lit.
Train	Starts the training process. Used once speakers have been connected to the outputs and audio is running through the amplifier. The process takes between 5 and 10 minutes. The Training light will switch off automatically when the process is complete. **Note:** If the load level is too low, the training process will not complete.
Abort	Aborts the training process. Use this button to stop the training process if more than 10 minutes have elapsed since the process was started, and the Training light is still lit.
Training	Lit (green) when Training is in progress.
Advanced	
Field name	**Meaning**
Amp	The amplifier number in the stack.
ID	The two octets of the amplifier ID number in hex format. This is shown on the display at the back of the unit.

IP Address	The IP address of the amplifier. This is shown on the display at the back of the unit.
Link	Lit (green) when a link is established between NWare and the amplifier across the network.
Fault	Lit (red) when a fault condition, e.g. thermal or short circuit, has been detected by the amplifier. Please refer to the amplifier documentation for information on fault conditions.
Rmte (Remote)	Lit (green) when the amplifier is running in remote mode.
Power	Lit (green) when the amplifier is switched on. If remote mode is selected on the amplifier, you can switch it on or off using the Power button within NWare.
Subnet	The subnet mask for the Ethernet / CobraNet network.
Gateway	The IP address of the router. If there is no router, this will be set to 0.0.0.0.
Input (Line / Cobra)	Selects either analog audio input using Phoenix connectors or digital audio input from the CobraNet network.
Bundle Num.	The audio bundle number for the device. This must correspond with the bundle number specified for the NION associated with the amplifier. You can then make connections to the amplifier using the NION flyoffs.
Gain/Sensitivity (CKi amps only)	Position of the Gain switch at the rear of the amplifier. This can be set to give constant sensitivity, or a constant gain of x20 (26 dB) or x40 (32 dB). Display only.
Volt	Voltage provided to the amplifier. Display only.
Amp	Current provided to the amplifier. Display only.
Watt	Input power provided to the amplifier. Display only.

Version	Firmware version running on the amplifier. Display only.

9. Type the amplifier ids in the **Id** boxes.

   These are shown at the back of each unit.

10. Type the IP addresses in the **IP Address** boxes.

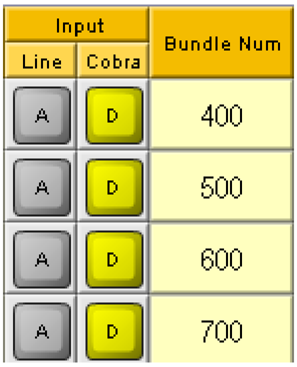
    These are shown at the back of each unit.

Tip: The subnet mask and gateway settings are read from the amplifiers automatically when they are contacted by NWare. These can be changed on the **Advanced** tab if necessary.

If the network settings are correct, the **Link** lights will be lit (green) and information will be received from the amplifiers, including the gain settings, temperature etc., and displayed in the fields of the **Crest Amplifier Stack** block.

11. Click the **Advanced** tab.

12. Under **Input**, click each **D** button to select digital audio input from the CobraNet network (via the NION).



Input		Bundle Num
Line	Cobra	
A	D	400
A	D	500
A	D	600
A	D	700

13. In each **Bundle Num.** box, type the bundle number that will be used to transmit audio to the amp.

    Each bundle can manage between 1 and 8 channels. The range is 0 to 65,535 (0 is off). For multicast bundles, the number must be in the range 1 to 255. For unicast bundles, the number must be in the range 256 to 65,279. Numbers in the range 65280 to 65535 are reserved and cannot be used.

Tip: If you are using four channel amplifiers, a single bundle number can be used for two amplifiers. Specify bundle numbers for either multicast or (multi-)unicast (as described above), and then specify the require settings for the NioNode CobraNet configuration, as described later in this procedure.

14. Click the **Channels** tab.

15. In the **CobraNet Sub Channel** boxes, specify subchannel numbers to use when referring to the amplifier channels.

    The channel numbers can simply be 1, 2, 3 etc.

16. Open the **NioNode** block and, in the first available **Bundle** boxes in the **Transmitters** section, specify the bundle numbers you specified in the **CKi Amplifier Stack** block or **Ci Amplifier Stack** block.

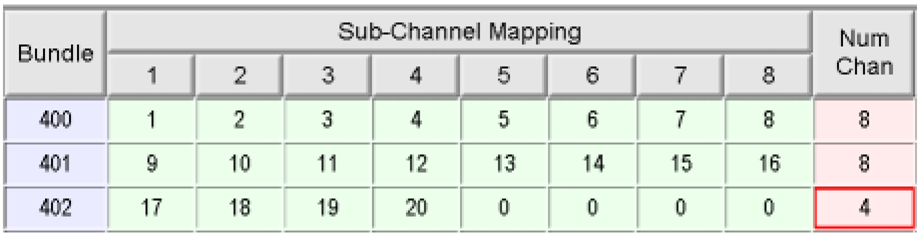
> **Tip:** To use multicast, so that two amplifiers can receive audio via the same bundle number, make sure the bundle number is in the range 1-255. The **Unicast Mode** and **Max Unicast** settings can be set to the default values. Be aware, however, that multicast is not recommended, as it will result in excessive processing for all network nodes that receive the transmitted data.

17. In the corresponding **Sub-Channel Mapping** boxes to the right of the **Bundle** boxes, specify the CM-1 output numbers (1-32) you want to send to the amplifier.

> **Note:** Ci and Cki amplifier default sub-channel mapping is one for one. Sub-channel 1 matches amplifier channel 1, sub-channel 2 matches amplifier channel 2, sub-channel 3 matches amplifier channel 3, and so on.

18. Any of the NION CM-1 audio output numbers may be entered in any sub-channel box. Click the sub-channel box and type the CM-1 output number you want to use.
19. In the **Num Chan** (number of channels) box, type the number of audio channels to be included in the bundle.

    The Num Chan setting, which has a default of 8, controls how many audio channels are sent in a bundle. The value should be set to the number of contiguous channels that will be used, as defined in the Sub-channel Mapping boxes. If Num Chan is set to a value that is greater than the number of channels used, then the bundle will be larger than necessary and will result in wasted bandwidth and processor time. Unused channels with a value of 0 are represented in the bundle by header data only (and no audio data will be included). If the bundle contains subchannel numbers that are not used, they will be sent as full audio channels and consume 1 Mbit of bandwidth per channel.



Bundle	Sub-Channel Mapping								Num Chan
	1	2	3	4	5	6	7	8	
400	1	2	3	4	5	6	7	8	8
401	9	10	11	12	13	14	15	16	8
402	17	18	19	20	0	0	0	0	4

> **Tip:** Values in the **Sub-Channel Mapping** boxes may be changed on a dynamic basis. This allows any hardware channel number to be mapped to any bundle sub-channel(s) at any time, greatly expanding signal routing flexibility. However, only users familiar with advanced CobraNet practices should use this feature.

> **Tip:** If you want to use multi-unicast, so that two amplifiers can receive audio via the same bundle number, set **Unicast Mode** to 2 and **Max Unicast** to 2. For more information on these settings, see *Routing audio on CobraNet networks* (on page 141).

20. On the toolbar, click the **Disemulate** button.

21. In the utility frame, click the **Flyoffs** tab, then drag flyoffs for the CM-1 output channels you are using over to the page.
22. For test purposes, wire the flyoffs to an audio generator - a sine generator, for example.
23. Save the project, then deploy it.

    Audio can now pass between the NioNode and the Crest amplifier(s).

Monitoring Crest amplifiers

Crest Audio offers two features to monitor system operation and help systems comply with safety regulations: *Real Time Load Monitoring (RTLM)* and *Speaker Tone Sensing (STS)*. Both are available in addition to the monitoring systems built in to the amplifiers. These features are set up differently and monitor different parts of the system. Each feature has its pros and cons. This section will discuss these features as separate entities and a hybrid approach will also be discussed for use with the CKi amplifier series.

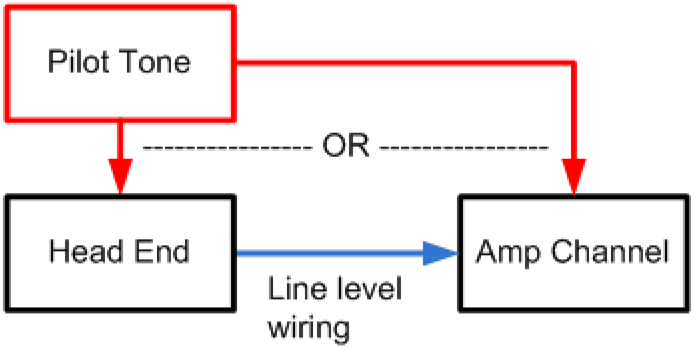
Note: The ideas and solutions in this section are specific to CKi amplifiers, NxCobraNet and NxEthernet modules, but they are also relevant, and indeed expanded, in the Ci/NxCobraNet8 hardware.

Speaker tone sensing

Speaker Tone Sensing (STS) allows monitoring of the system from the input of the pilot tone to the amplifier outputs. The pilot tone is the stimulus that is continuously measured at the output of the amplifier and is defined as the system steady state level. If the pilot tone is not detected at the amplifier output, an alert is issued.

The pilot tone is input into the system at or before the amplifier input and is typically placed at or near 20KHz to be out of the audible range. The pilot tone level is kept relatively low but must be detected by the remote output VU meters. A typical level as measured at the amplifier output is around 1.5V RMS. The Impedance of most speakers at the upper end of the audio spectrum is very high. Therefore, not much power is dissipated.

Software monitors the output VU meter for the presence of the pilot tone. The STS software has entry for a threshold level (with hysteresis) for the output VU meter for each amplifier channel. If the output VU level falls below this threshold, a STS alert is issued.

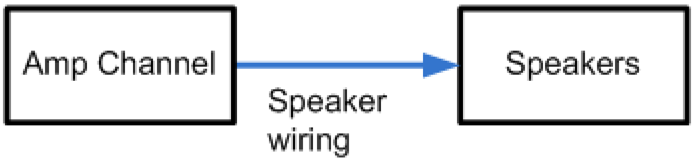


STS continuously verifies the system from the entry point of the pilot tone to the amplifier output.

Real time load monitoring

Real Time Load Monitoring allows monitoring of the speaker lines and loads. Each amplifier channel calculates a full spectrum average impedance from the output voltage and current. The impedance is averaged through a 4 minute moving average to stabilize the reading. This impedance is checked to ensure it is between an upper and lower threshold. If it falls outside this range, an RTLM alert is issued.

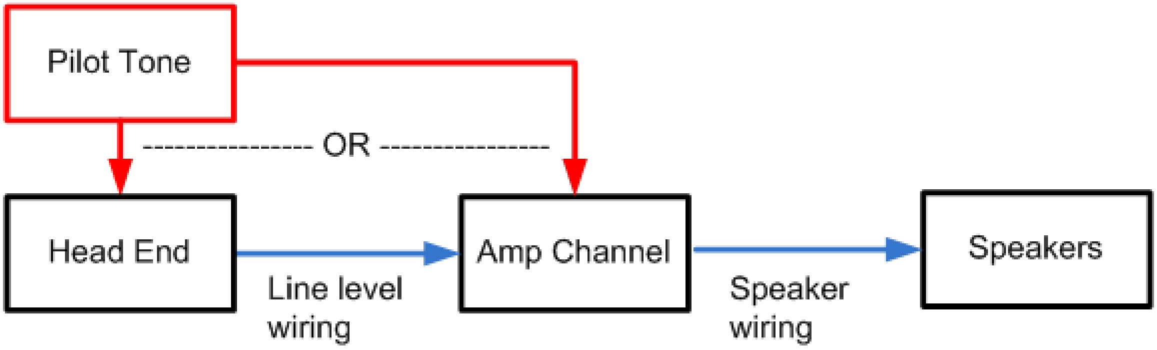
A reference impedance is obtained by clicking the *Train* button on the *Advanced* tab of the amplifier device in NWare, then playing appropriate program material into the channel/system. Without the program material (stimulus), RTLM or training is not actively getting impedance data to check the load. The amp channel collects impedance data for roughly 5 minutes (when the signal is constantly above the threshold). Once training is complete, the reference impedance is automatically saved in non-volatile memory. This impedance is then used to compare against new impedance data. When the difference between the reference impedance and new impedance data is above the upper threshold or below the lower threshold, an RTLM alert is issued. The upper and lower RTLM limits are % changes of the reference impedance.



RTLM verifies the amplifier output, speaker lines and loads when there is viable program level.

Combining STS and RTLM

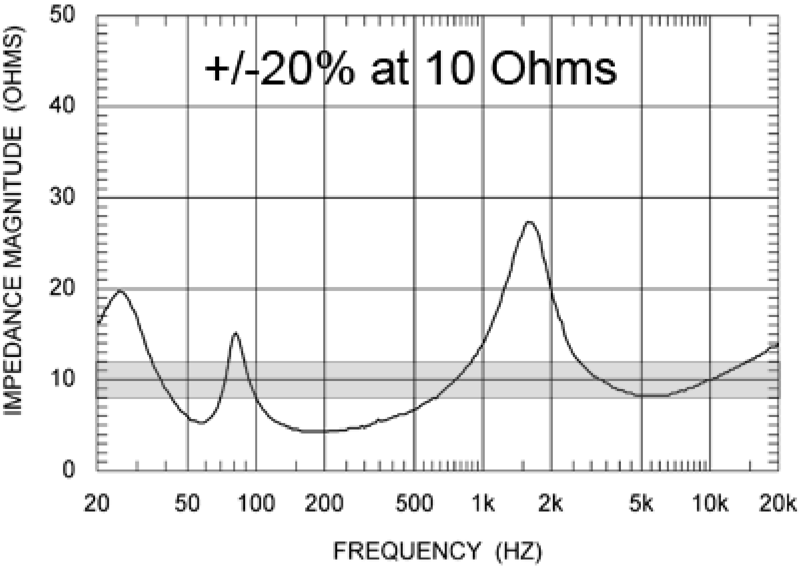
Combining these features for *speaker tone driven load monitoring* is a feasible idea. This allows verification from the input of the pilot tone through any head end gear and wire, through the amplifier, through the speaker wires to the load.



A pilot tone is used to provide a constant stimulus for both load monitoring and STS. This allows load monitoring even without program material.

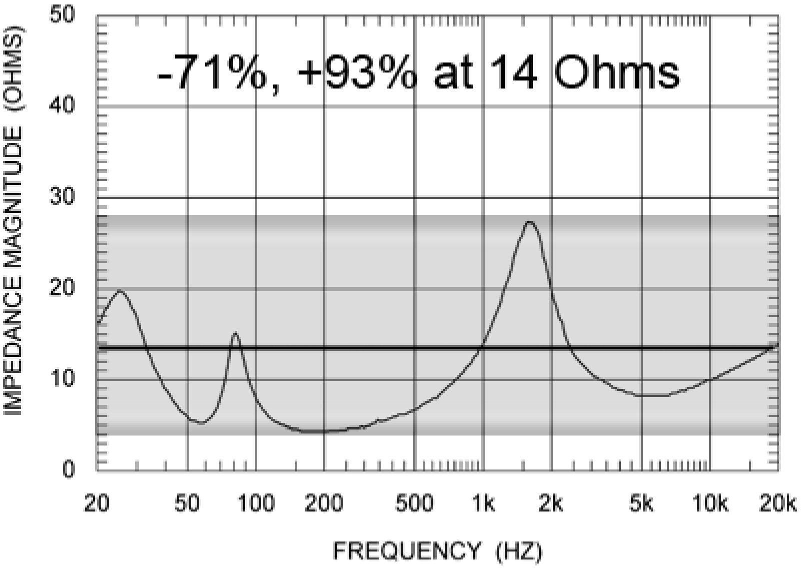
The solution involves some work creating STS functionality in NWare for deployment to an nControl. This involves reading the output VU information from the amplifier and creating STS threshold and hysteresis fields. An alert is then generated when the output VU meter level falls below the STS threshold.

No changes to RTLM are needed. However, some effort is required to properly configure it to avoid false alerts. The averaged impedance measured by RTLM is influenced by the program material due to the loads impedance plot. With this plot, we can better set the upper and lower RTLM limits.



The plot shows a vented box speaker impedance. The gray area (safe operating area) shows a +/-20% margin centered about 10 Ohms. As you can see, this area does not cover a large portion of the curve. It would be wise to increase the RTLM upper and lower limits to allow more of the curve in the *safe operating zone*.

The speaker plot below has an impedance of about 14 Ohms at 19KHz. The highest impedance is 27 Ohms at 1.5KHz and the lowest impedance is about 4 Ohms at 200 Hz. So, if we set a reference impedance (train) with a 19KHz tone, we should expect to get a 14 ohm result. Since the upper and lower RTLM limits are % change thresholds, we can calculate the limits to suit the load as such.



$impedance_{ref} := 14$ Speaker impedance at 19kHz

$impedance_{min} := 4$ Minimum speaker impedance

$impedance_{max} := 27$ Maximum speaker impedance

$$RTLM_{lower} := 100 \frac{impedance_{ref} - impedance_{min}}{impedance_{ref}}$$

$RTLM_{lower} = 71.429$ RTLM lower limit (%)

$$RTLM_{upper} := 100 \frac{impedance_{max} - impedance_{ref}}{impedance_{ref}}$$

$RTLM_{upper} = 92.857$ RTLM upper limit (%)

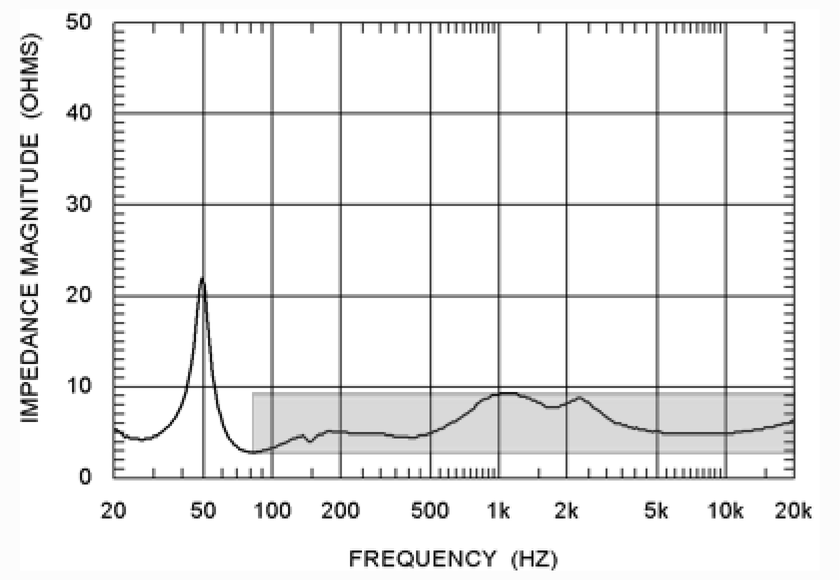
Setting the RTLM limits to these values will ensure that all measured impedances measured are in the *safe operating area* (gray area in graph above). If needed, the upper limit can be refined and lowered thus increasing the RTLM performance.

Note: This is based on the program material not containing 4 minutes of 1.5KHz tones.

The pilot tone dissipates less than 0.2 watts in the load above.

$$\frac{1.5V^2}{14ohms} = 0.160714 \text{ watts dissipated in load}$$

While the limit calculations above include the entire frequency (and impedance) range, the range can be narrowed to deal with frequencies contained in the program material. For instance, if the system is to be used as a paging system (voice only), the frequency range can be narrowed to deal with only the vocal fundamental frequencies. The impedance plot below shows the *safe operating area* of a speaker with a 4th order high-pass filter at 100 Hz. The large peak at 50 Hz can effectively be ignored, thus increasing the effectiveness of the RTLM measurement. The closer the load impedance minimum and maximum values (the smaller the *safe operating area*), the more effective RTLM will be.



When using RTLM without a pilot tone, the *safe operating area* can be narrowed even further for better RTLM performance. A typical music reinforcement system can be set up to run with a value of 20 for upper and lower limits (+/- 20% of reference impedance). Typical RTLM setup is as follows:

1. Run program material at typical levels for about 10-15 minutes through selected channel(s) to warm up speakers.
2. Run pink noise at about -26dB of typical program material level through selected channel(s). (Raise the level to a comfortable listening level.)
3. Click *Train* on selected channels.

   If training is taking more than 5 minutes, raise the pink noise level for the channel.
4. Once training is complete, set the RTLM upper and lower limits to 20.

Besides the RTLM set up described above, the STS settings must be entered as well. As mentioned earlier, the pilot tone signal must be high enough to allow it to be detected by the output VU meter for the amplifier channel. A reading of 1.5V RMS at the output of the amplifier is sufficient. The STS software must be configured to send an alert if the output VU level falls below the user-defined threshold.

By using the suggested methods in this section, Crest CKi amplifiers can employ STS and RTLM features to ensure your system is continuously monitored from the input to the speakers.

Notes:

- The speaker impedance curves are helpful when making these calculations. However, measuring the actual load impedance, which may include many speakers of various types, is recommended. The *WT3 Woofer Tester* (*http://www.parts-express.com/pe/showdetl.cfm?Partnumber=390-804*) has been found to be effective for this purpose.
- The RTLM alert is generated by the amplifier, whereas the STS alert is generated by the NWare / nControl system.
- The closer the amplifier channels rated load impedance is to the actual load impedance, the lower the pink noise (or reference stimulus) level can be for accurate impedance training. If a channel is rated at 8ohms, but only has a 312ohm load (16W tap at 70V), the channel may need a higher pink noise level to train properly.

Adding an XControl LCD device to your design

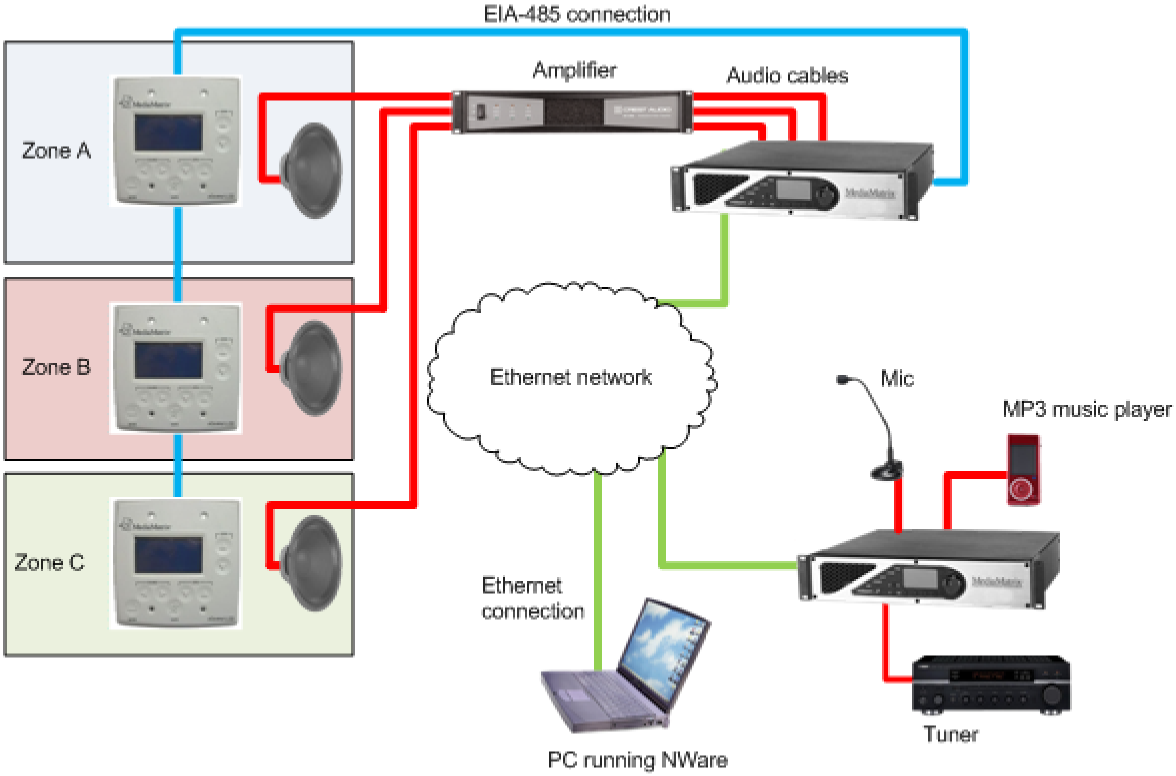
Introduction

xControl LCD is a wall-mounted control panel for controlling MediaMatrix audio systems.

The unit has capacitive touch buttons for selecting the audio source, zone and level, as well as muting the output and confirming a menu choice or setting change (Enter button). There is a two color LCD display for presenting information to the user.

Note: Currently, the xControl LCD is limited to controlling a maximum of 4 input sources and 4 zones.

The xControl LCD works in conjunction with a NION, which communicates with the unit via an EIA-485 connection.



To use a set up like the one in the example, your NWare project will need to contain three xControl LCD devices, each configured to contact an xControl LCD via a NION on the network. You will also need two NioNodes and an amplifier device.

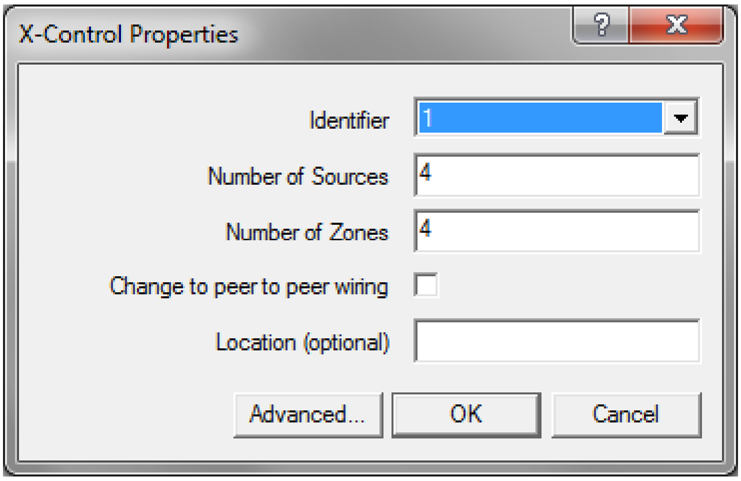
The xControl LCD sections in the NWare User Guide will explain how to add the device to your design and configure it, but if you require only reference information, see *xControl LCD* in the *NWare Device Reference*.

Adding the XControl LCD node

1. In the device tree, expand the **Hardware** folder.
2. Expand the **XControl** folder.
3. Expand the **LCD** folder.
4. If you want to wire the xControl LCD device to control devices in your design, drag a **Generic Controls (PASHA)** device over to the design page.

   If you want to wire the xControl LCD device to audio devices in your design, drag a **Zone Source and (PASHA)** device over to the design page.

The **X-Control Properties** dialog box is displayed.



X-Control Properties

Identifier: 1

Number of Sources: 4

Number of Zones: 4

Change to peer to peer wiring: □

Location (optional):

5. In the **Identifier** list, click the base code for the xControl LCD you want to control.

   The base code is selected on the *Set Base Code* screen of the xControl LCD. For more information, see *Setting the unit base code* in the *xControl LCD Hardware Manual*.
6. In the **Number of Sources** box, type the number of inputs to use as program sources. An input wiring node will be added for each one.

Note: Currently, the xControl LCD can control a maximum of four sources.

7. In the **Number of Zones** box, type the number of zones that the xControl LCD will manage. These could be individual rooms in a building, for example.

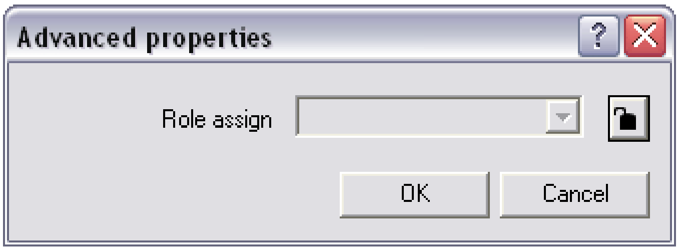
Note: Currently, the xControl LCD can control a maximum of four zones.

8. If you are using a Generic Controls (PASHA) device, and you want the output wiring nodes to use peer-to-peer wiring, as opposed to master-slave wiring (the default), select the **Change to peer to peer wiring** check box.

Note: The *Change to peer to peer wiring* check box is only available on the *Generic Controls (PASHA)* device.

9. If you want to display the physical location of the device on the device block, type the location details in the **Location** box.
10. If you want to change the role to which this device belongs:
    a) Click **Advanced**.

    The **Advanced properties** dialog box is displayed.



Advanced properties

Role assign:

    b) Click the 🔓 button to change it to the 🔒 button, and then select a role name in the list.

11. Click **OK**.
12. Click **OK**.

Adding an nTouch 180 node to your design

Introduction

nTouch 180 is a touch screen control panel for controlling MediaMatrix systems. The device runs an embedded version of NWare Kiosk and allows the user to control a project designed in NWare. It can host the project and media files or connect to another device on the network where they are being hosted. The unit provides several mounting options, including surface, panel, flush and general VESA mounting.

After nTouch 180 has been installed and given initial configuration settings, the majority of its features are managed via the web interface and touch screen interface. Software upgrades are managed using NWare.

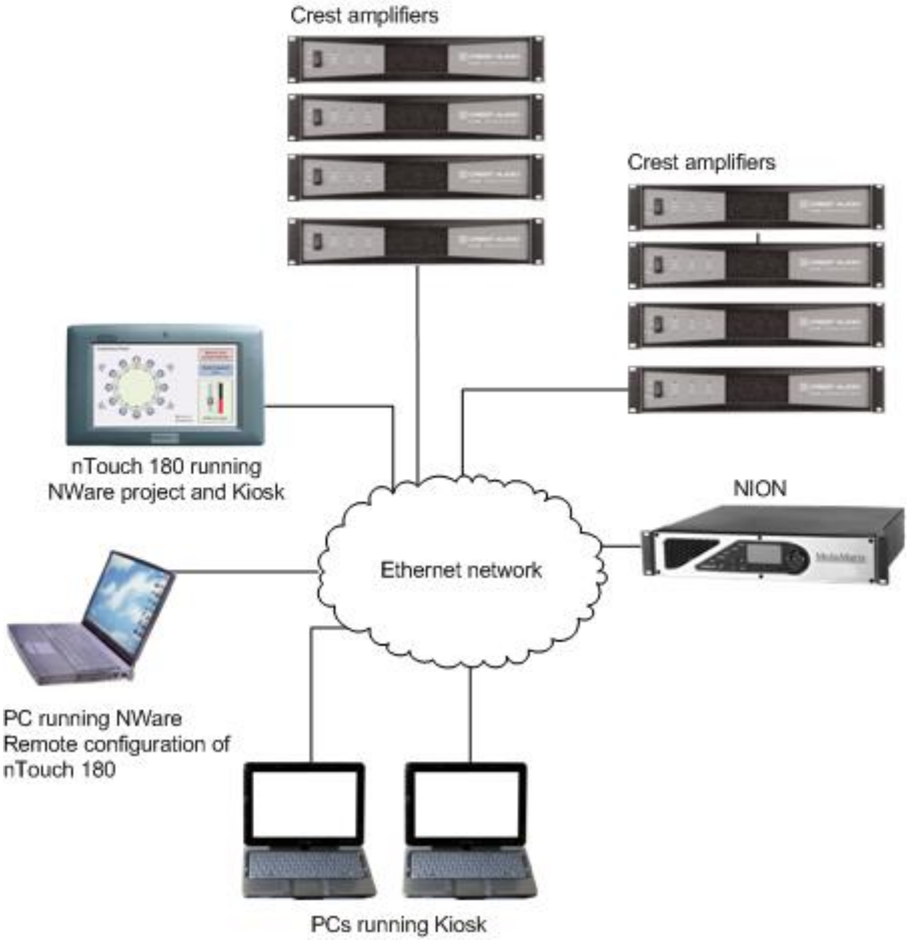
Notes:

- nTouch 180 units host projects for control and monitoring, but do not have any DSP capability.
- Using all the capabilities of nTouch 180 requires a firm understanding of NWare, SNMP and Python scripting.

The diagram below shows a typical scenario where an nTouch 180 allows users to control a number of amplifiers remotely. The project created in NWare on the laptop gives the Kiosk users graphical controls for altering the volume, selecting an input source, or muting the audio output, for example.

The nTouch 180 can also be used to monitor network devices, such a switches, via SNMP and other protocols, and this information can then be made available to the system administrator.

The NION supplies audio to the amplifiers across the network, and complements the role of the nTouch 180 in providing an overall audio management and networking solution.



Understanding the redundancy system

The nTouch 180 redundancy system allows a standby unit to replace an active unit on the network if it stops functioning.

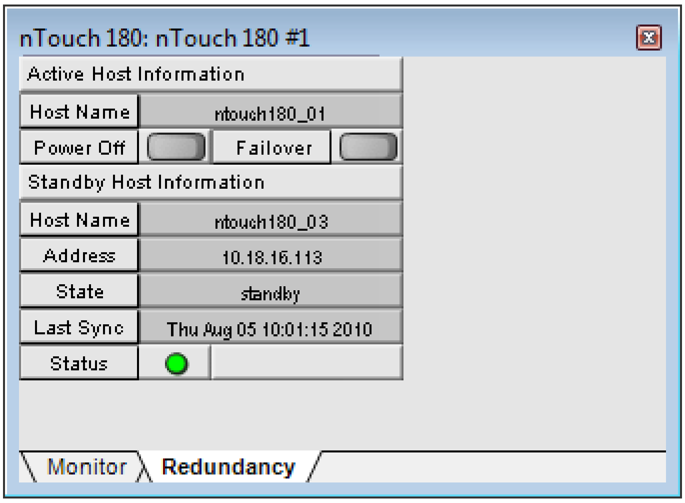
When the system is used, the configuration settings from the active unit are automatically copied to the standby unit to keep them synchronized. The two units are responsible for a single role.

Note: In order to use the redundancy system, you must configure the units via the web interfaces before you use them in your project. For more information, see *Setting up redundancy* in the *nTouch 180 Hardware Manual*.

The explanations below assume that you have already added an nTouch 180 device to your project.

A system with two peer units

When you deploy the project, you will see the active nTouch 180 unit listed on the *Role Deployment* dialog box, but the standby unit will not be listed. When you deploy the project to the active unit, the *Redundancy* tab on the nTouch 180 device will show the host name of the active unit under *Active Host Information*. The name and IP address of the standby unit will be displayed under *Standby Host Information*.



nTouch 180: nTouch 180 #1

Active Host Information

Host Name: ntouch180_01

Power Off | Failover

Standby Host Information

Host Name: ntouch180_03

Address: 10.18.16.113

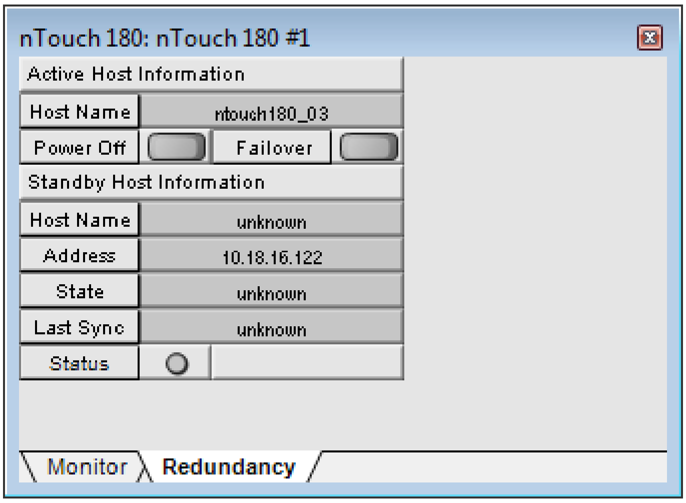
State: standby

Last Sync: Thu Aug 05 10:01:15 2010

Status: ●

Monitor | Redundancy

If the active unit fails, after a short delay, the standby unit will start hosting the role; under *Active Host Information*, the *Host Name* field will change to show the name of the standby unit; and under *Standby Host Information*, the *Address* field will show the IP address of the unit that has just failed.



nTouch 180: nTouch 180 #1

Active Host Information

Host Name: ntouch180_03

Power Off | Failover

Standby Host Information

Host Name: unknown

Address: 10.18.16.122

State: unknown

Last Sync: unknown

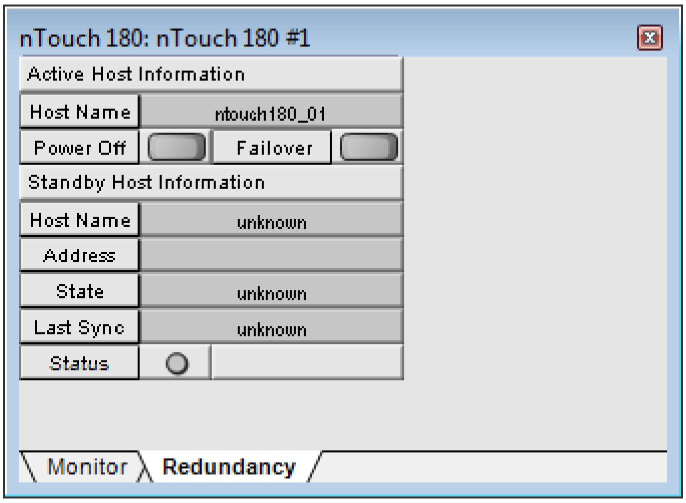
Status: ○

Monitor | Redundancy

If the failed unit starts functioning, its details will be shown under *Standby Host Information*. It will not resume hosting the role.

A system with a primary unit and a backup unit

When you deploy the project, you will see the active nTouch 180 unit listed on the *Role Deployment* dialog box, but the standby unit will not be listed. When you deploy the project to the active unit, the *Redundancy* tab on the nTouch 180 device will show the host name of the active unit under *Active Host Information*, but the *Standby Host Information section* will not show information from the standby unit.



nTouch 180: nTouch 180 #1

Active Host Information

Host Name: ntouch180_01

Standby Host Information

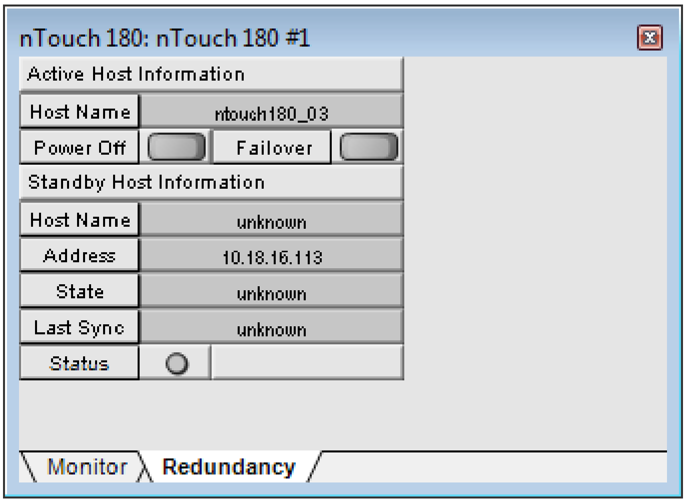
Host Name: unknown

Address:

State: unknown

Last Sync: unknown

If the active unit fails, after a short delay, the standby unit will start hosting the role; under *Active Host Information*, the *Host Name* field will change to show the name of the standby unit; and under *Standby Host Information*, the *Address* field will show the IP address of the unit that has just failed.



nTouch 180: nTouch 180 #1

Active Host Information

Host Name: ntouch180_03

Standby Host Information

Host Name: unknown

Address: 10.18.16.113

State: unknown

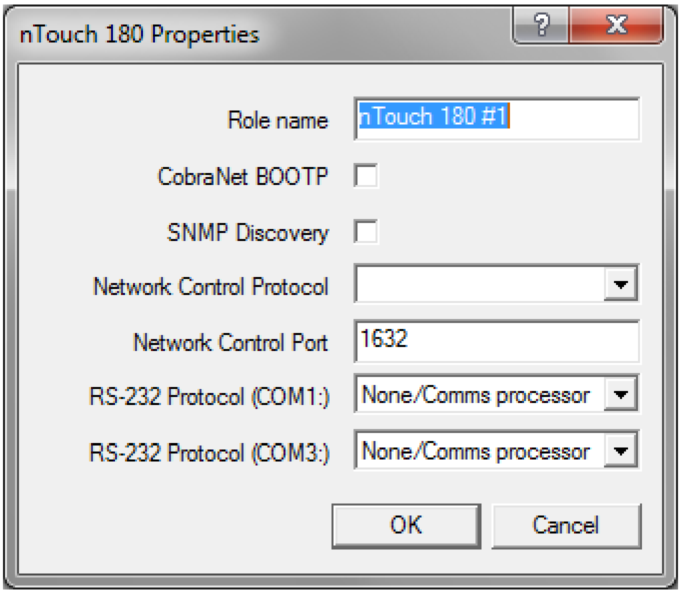
Last Sync: unknown

If the failed unit starts functioning, it will resume hosting the role and, under *Active Host Information*, the *Host Name* field will change to show the name of the unit that was originally hosting the role. The unit that is now back in standby mode will be restarted.

Adding the nTouch 180 node

1. In the device tree, expand the **Hardware** folder.
2. Drag the **nTouch 180** object over to the design page.

The **nTouch 180 Properties** dialog box is displayed.



nTouch 180 Properties

Role name: nTouch 180 #1

CobraNet BOOTP □

SNMP Discovery □

Network Control Protocol:

Network Control Port: 1632

RS-232 Protocol (COM1:): None/Comms processor

RS-232 Protocol (COM3:): None/Comms processor

3. If you want to change the role name from the default, type a new name in the **Role name** box.

Note: You must specify a unique role name in the **Role name** box. If you do not do this, an error will be displayed when you deploy the project.

4. If you want to use the nTouch 180 unit as a CobraNet BOOTP server, providing IP addresses to other devices automatically, select the **CobraNet BOOTP** check box.

Notes:

- The BOOTP server is recommended for use with CobraNet devices only.
- It is unlikely that any device will consistently receive the same address after the nTouch 180 is rebooted.
- The BOOTP server is designed to support CobraNet devices and uses the more recent BOOTP extensions detailed in RFC 1497. The server is not guaranteed to be backwardly compatible with all BOOTP clients.
- It is possible, although unlikely, that the BOOTP server will allocate IP addresses to non-CobraNet devices on the network that are configured to use DHCP. If this problem occurs, we recommend that you switch off the BOOTP server on the nTouch 180 unit and use an alternative server.

5. If you want the nTouch 180 unit to be able to discover other SNMP devices on the network, select the **SNMP Discovery** check box.

Notes:

- This feature can generate a large amount of network traffic.
- This feature must be enabled in order to use the SNMP Agent Discovery device in your design.

6. If you intend to control the nTouch 180 unit using RATC over the IP network:

a) Click the RATC version in the **Network Control Protocol** list.

b) If you want the RATC server on the nTouch 180 unit to listen on a different port to the default, type the new port number in the **Network Control Port** box.

The default is 1632. The valid range is 1-32767.

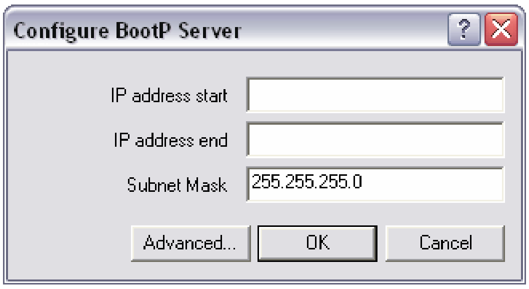
Note: If you change the port number, you will have to reconfigure the clients to connect using the new port.

7. If you want to use the RATC or PASHA protocol to control the nTouch 180 via its serial ports, click the protocol name in the **RS-232 Protocol** list(s) for the ports you intend to use.

None/Comms processor	Used when the project contains a Comms Processor device, or when you do not want to use the serial port. **Note:** If you select one of the other options – RATC2, for example, then try to use the Comms Processor device, it will not respond as quickly.
RATC2	RATC2 is the improved version of RATC that was introduced with the NION platform, but is also supported by nControl and nTouch 180. Very similar to RATC1, RATC2 introduces shortened commands, and several extra functions.
RATC1	RATC1 is the first generation protocol that was used in Classic frame-based MediaMatrix systems. RATC1 for NioNodes, nTouch 180 nodes and nControl nodes is equivalent to what was called RATC in Classic MediaMatrix.
PASHA/PageMatrix	Protocol used with the PageMatrix Command Center. It supports a four character control ID sent with the (T) trigger command.
PASHA/XControl	Provides basic S (Set) PASHA functionality with X-Net2-style trigger commands.
PASHA/Legacy	Designed for projects that contain legacy MediaMatrix nodes that are controlled by external programs using the classic PASHA protocol, as implemented on MainFrames and MiniFrames. When you want to replace the legacy nodes with NioNodes, select the *PASHA/Legacy* option and you will not need to update your external program code. This option ensures all hex values are returned from nodes in lower case instead of upper case to match the original MediaMatrix PASHA protocol.

8. Click **OK**.
9. If you selected the **CobraNet BOOTP** option:

The **Configure BootP Server** dialog box is displayed.



IP address start

IP address end

Subnet Mask 255.255.255.0

a) In the **IP address start** box, type the start IP address for the range that can be allocated to clients requesting IP addresses.

b) In the **IP address end** box, type the end IP address for the range that can be allocated to clients requesting IP addresses.

c) In the **Subnet Mask** box, type the subnet mask for the IP address range.

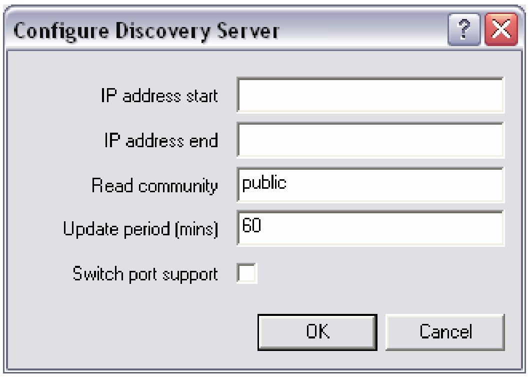
d) If you want to change the maximum number of IP address requests the BOOTP server can process simultaneously, click **Advanced**, type a new number in the **Threads** box, and then click **OK**.

Tip: We recommend that you leave the thread count setting set to 2.

e) Click **OK**.

10. If you selected the **SNMP Discovery** option:

The **Configure Discovery Server** dialog box is displayed.



IP address start

IP address end

Read community public

Update period (mins) 60

Switch port support □

a) In the **IP address start** box, type the start IP address for the range that will be scanned during the discovery process.

If you want to use a broadcast address, which will provide better performance when scanning a large range of addresses, type the broadcast address.

When a large number of devices are discovered, using a broadcast address will greatly reduce the impact on the network. The format of a broadcast address is x.x.x.255. This will broadcast to the entire /24 subnet. You can also use x.x.255.255 to broadcast to a /16 subnet, but the impact of addressing such a large number of devices must be considered carefully.

Note: Not all SNMP devices will respond to a broadcast addresses. Check that the devices you are using are compatible.

b) In the **IP address end** box, type the end IP address for the range that will be scanned during the discovery process.

If you are using a broadcast address, leave this box blank.

c) In the **Read community** box, type the read community string (password) for accessing devices on the network during the discovery process.

d) In the **Update period (mins)** box, type the interval to use between scans for devices on the network.

e) If you want to enable switch port discovery, select the **Switch port support** check box.

This feature allows devices to be detected via the switch port to which they are connected. Both the switch and the device must support SNMP. Furthermore, the device must respond to the SNMP discovery process.

f) Click **OK**.

Forcing a failover to the standby unit

The *Redundancy tab* on the nTouch 180 device has a *Power Off* button that will power down the active unit, and a *Failover* button that will restart the active unit. These are useful for testing the redundancy system.

⏩ ***To restart the active unit***

- Click **Failover**.

⏩ ***To power down the active unit***

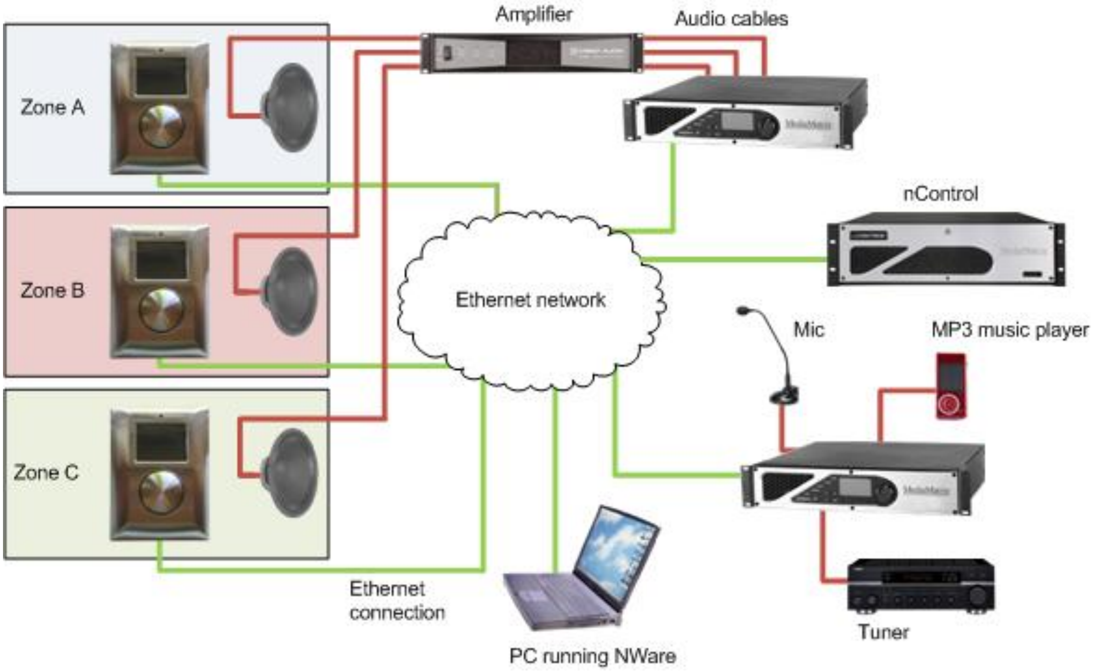
- Click **Power Off**.

Adding an nTouch 60 node to your design

Introduction

The nTouch 60 is a wall-mounted control panel for controlling MediaMatrix audio systems. The unit has a touch-sensitive screen for selecting options from an audio control project, which has been designed in NWare. It also has a knob for adjusting individual settings, like volume, and selecting options.

The nTouch 60 works in conjunction with an nControl unit or an nTouch 180 unit, which hosts the Kiosk pages from the NWare project.



To use a set up like the one in the example, your NWare project will need to contain three nTouch 60 devices, each configured to contact an nTouch 60 on the network. You will also need an nControl node, two NioNodes and an amplifier device.

The nTouch 60 sections in the NWare User Guide will explain how to add the device to your design and configure it, but if you require only reference information, see *nTouch 60* in the *NWare Device Reference*.

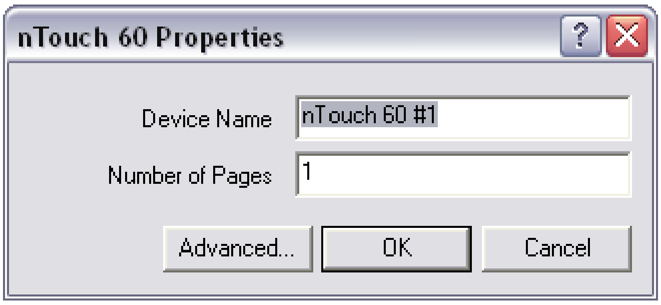
Compatibility with NWare 1.6.2a

If you have created a project using NWare 1.6.2a that contains nTouch 60 devices, each nTouch 60 device will need to be recreated in the current version. The nTouch 60 devices in the project will not be updated automatically when it is opened in NWare.

Adding the nTouch 60 node

1. In the device tree, expand the **Hardware** folder.
2. Drag an **nTouch 60** object over to the design page.

   The **nTouch 60 properties** dialog box is displayed.



nTouch 60 Properties

Device Name: nTouch 60 #1

Number of Pages: 1

3. In the **Device Name** box, type a name for the device.

Tip: The name does not need to be unique, but we recommend that you assign unique names to each device to help with the organization of your project.

4. In the **Number of Pages** box, type the number of individual design pages to be displayed on the device when it is created.

   These are the pages that will be visible on the device itself when the project is running.

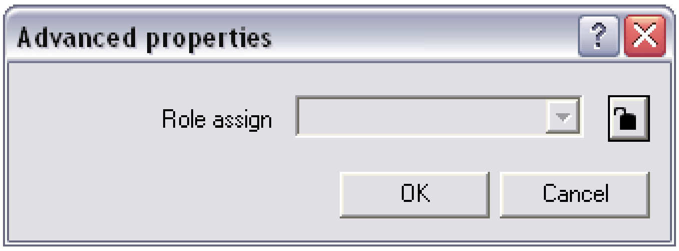
Notes:

- If you change the number of pages after the device has been created, all the controls you have added to the pages will be deleted. To avoid this problem, you can use a GUI block with the same dimensions as the device, finish the design in the GUI block, then copy the controls over to the device.
- We recommend that you place nTouch 60 devices, Kiosk Layout blocks, Kiosk2Go Layout blocks and Mobile Display blocks on a separate page to the other devices in the project, and do not export the page.

5. If you want to change the role to which this device belongs:

   a) Click **Advanced**.

   The **Advanced properties** dialog box is displayed.



   b) Click the 🔓 button to change it to the 🔒 button, and then select a role name in the list.

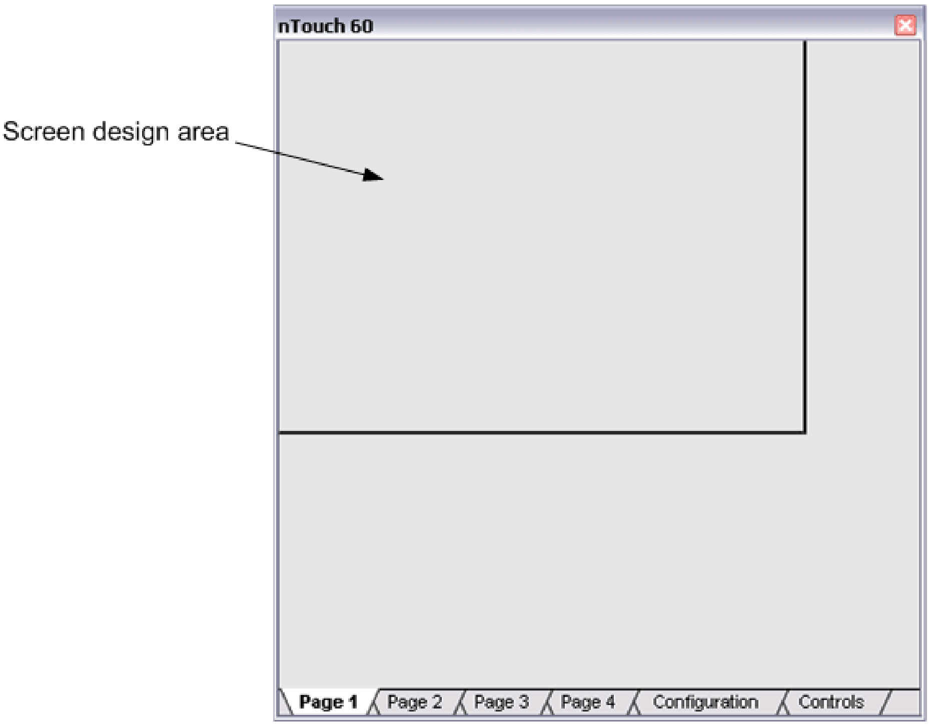
   c) Click **OK**.

6. Click **OK**.

Understanding the layout of the device

Once you have added the nTouch 60 device to your design, you can start to add controls to the individual pages, so end users can interact with them.

When you open the nTouch 60 device, you will see that NWare has created a number of individual pages automatically.



Each page labeled *Page 1*, *Page 2*, and so on, shows the area that will be visible on the nTouch 60 screen – the *screen design area* – at the top. Any controls you place in the screen design area will be visible on the nTouch 60 screen when the project is running.

Notes:

- Do not lock pages within the device or the page on which the device is located in your design. You will not be able to access the locked pages from the nTouch 60.
- We recommend that you place nTouch 60 devices, Kiosk Layout blocks, Kiosk2Go Layout blocks and Mobile Display blocks on a separate page to the other devices in the project, and do not export the page.

The *Configuration* tab is used to contact the physical nTouch 60 unit that this NWare device will control; it also allows you to specify settings to control the behavior of the screen and update the firmware. On the *Controls* tab are controls you can add to the screen design area on each page; these include knob controls for interacting with the knob on the nTouch 60 unit and status LEDs.

Connecting NWare to the nTouch 60 unit

In order for the NWare nTouch 60 device to communicate with the physical nTouch 60 unit, you must specify the IP address of the physical device on the control surface of the NWare device. You will then be able to see controls you have added to the pages of the NWare device on the screen of the nTouch 60 unit, and you will be able to interact with them using the touch screen and control knob.

⏩ *To connect NWare to the nTouch 60 unit*

1. Make sure the project contains an nControl unit or nTouch 180 unit, as well as an nTouch 60.
2. Double-click the nTouch 60 device block to open it.

Tip: We recommend that you test the responsiveness of the control knob the first time you connect to the unit from NWare. Click the **Controls** tab, click the **Page 1** knob control to select it, then copy and paste it onto the screen design area on page 1.

3. Deploy or emulate the project.
4. Click the **Configuration** tab.
5. Under **Communications**, type the IP address of the nTouch 60 in the **Device IP Address** box.

   When the nTouch 60 is contacted, the **Device Responding** LED is lit (green) and the **Temperature** box shows a current reading from the device.

Tip: If the **Device Responding** LED does not light, check that the IP address is correct (you can display the IP address of the nTouch 60 on its screen by pressing the set up button at the rear of the unit, and then pressing **IP Config** on the screen); also check that you can contact the unit using a PING command from a Command Prompt window.

   For information on the controls, see *Controls* in the *NWare Device Reference*.

6. If you want to test the responsiveness of the control knob, open page 1, and then move the control knob on the front of the nTouch 60 unit.

   If communications are working properly, the control knob on the NWare device will move at the same time as the knob on the nTouch 60 unit.

Using the knob controls to store values

On the *Controls* tab, the NWare nTouch 60 device has a knob control for each individual page and a single, global knob control. It is important to understand what each knob control does and when to use it.

Type of knob control	What it does	When to use it
Knob control for each page	Stores a value for that page on the nTouch 60 device. Each knob control of this type can store a different value.	When you want to wire the knob control on one page to one device, and the knob control on another page to a different device. For example, if you want to control the gain for amplifier A

		on page 1 and the gain for amplifier B on page 2: 1. Wire the Page 1 knob control to the gain setting for amplifier A. 2. Wire the Page 2 knob control to the gain setting for amplifier B. 3. Copy the Page 1 knob control to the screen design area on that page. 4. Copy the Page 2 knob control to the screen design area on that page.
Global knob control	Stores a single value for the entire nTouch 60 device.	When you want to control the same device using the knob control on more than one page of the nTouch 60 device. For example, if you have a single amplifier in your project and you want to allow the user to control the gain from more than one page on the nTouch 60 device: 1. Wire the global knob control to the gain setting for the amplifier. 2. Copy the global knob control to the screen design area on each of the pages.

Sizing the screen design area to match the physical display area

The touch screen on the nTouch 60 measures 2.4" (60mm) diagonally and features a display resolution of 320 x 240 pixels. The screen design area on the NWare nTouch 60 device has the same pixel size, so when you add controls to this area, you will be able to see the proportion of the screen area used by each control. However, you will not be able to see how large the controls will appear on the screen of the physical device until it is connected to your project.

To help with this problem, you can use the image below, which shows the physical screen area of the nTouch 60.

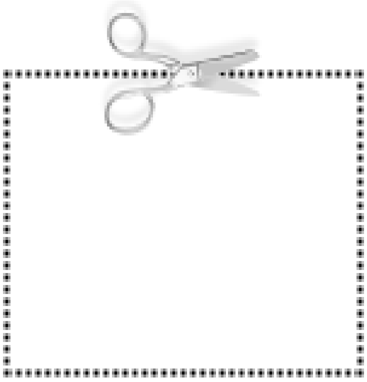
Note: You can only print out the image from the PDF version of this manual. If you are viewing the online version, click this PDF icon to display the PDF version.

To size the screen design area to match the physical display area

1. Print out this topic from the PDF (with the *Page Scaling* setting set to *none*).

2. Cut out the rectangle along the dotted line and then place it on your monitor.
3. Zoom the NWare display using the 🔍 and 🔍 buttons on the toolbar until the screen design area of the nTouch 60 device matches the size of the rectangle.

   Controls you add to the screen design area will now appear the same size as they will on the physical nTouch 60.



Recommendations for arranging and sizing controls

To help you design screens for the nTouch 60 that will be as useful and effective as possible, we have devised a number of recommendations for controls.

Type of control	Recommendations
All controls	Arrange controls at least 16 pixels apart to avoid the user pressing the wrong control.
Button	Set the size to at least 87 (w) x 93 (h) to allow for a range of finger sizes and for a label to be displayed above the button.
Rotary control	This type of control is difficult to gesture using the touch screen. Wire the control to the global knob control or the knob control for one of the pages. Alternatively, use a separate up arrow button and a down arrow button to the user to increase or decrease the value.
Fader	This type of control is difficult to gesture using the touch screen. Wire the control to the global knob control or the knob control for one of the pages. Alternatively, use a separate up arrow button and a down arrow button to the user to increase or decrease the value.

If you are looking for more general recommendations for screen design, you can find information here: *http://www.asktog.com/basics/firstPrinciples.html* (*http://www.asktog.com/basics/firstPrinciples.html*).

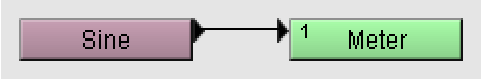
Chapter 5

Routing audio between devices

In This Chapter

Introduction

You can route audio between certain types of devices in your design by simply wiring them together. In the example below, the sine generator has been wired to the meter.



To wire devices together like this, switch to Wire mode by clicking the button on the toolbar, then simply drag the wiring node ▶ on the first device over to the wiring node on the second device and release the mouse button. If the link is valid, a solid line will be drawn.

Tip: Audio wiring nodes look like this: ▶ and control wiring nodes look like this: ○.

If the sine and meter devices were on different pages in the design, you would need to create *flyoffs* and use them to wire the devices together. For more information, see *Wiring DSP devices together using flyoffs* (on page 136).

If two NioNodes are installed in the same equipment room (and they are n3 or n6 models), you can use the XDAB interfaces at the rear of the units to physically wire them together, and then manage the connections using an XDAB device in NWare.

If you want to transfer audio from a card in a NioNode to a device in your design, you need to use audio card flyoffs from the Flyoffs tab. See *Routing audio to and from NioNode audio cards* (on page 140).

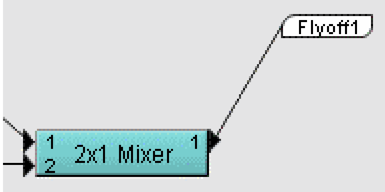
If you want to transfer audio from a NioNode to another NioNode or a CAB, and the connection runs across a network, you need to use CM-1 (CobraNet) or DLM (Dante) flyoffs. See *Routing audio on CobraNet networks* (on page 141) or *Routing audio on Dante networks* (on page 153).

Wiring DSP devices together using flyoffs

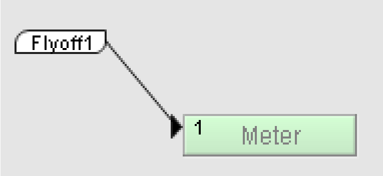
Introduction

Flyoffs are used in NWare to allow signals to be transferred between devices in different parts of a design (within a page, between pages, between devices on the network, and so on) without there being a graphical wire connection.

In the example below, the output from a mixer device is connected to a flyoff called *Flyoff1*.



In another part of the design, a second flyoff is also labeled *Flyoff1* . Connecting this flyoff to the meter allows the signal to flow from the mixer device to the meter.



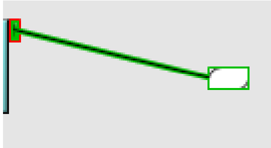
Note: If you make a wire connection between two devices that is invalid, such as wiring two audio outputs together, NWare will allow the connection to be made. An error message will not be displayed, but the wire attached to the flyoff will be colored red. Similarly, when a flyoff is unresolved, such as when an audio output is not flyoff wired to an audio input, the attached wire is colored green, but no error message is displayed.

Creating a flyoff

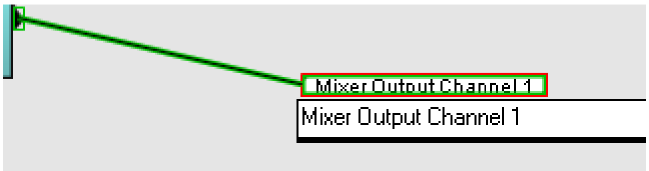
1. On the toolbar, click the button to switch to Wire mode.
2. Move the mouse pointer over a wire node (▯ or ▶) on a device.
3. Hold down the left mouse button and drag the mouse over to a space next to the device.

   A wire will be drawn from the wire node to the mouse pointer position.
4. Click the right mouse button.

   A flyoff is created. Audio flyoffs are white; control flyoffs are dark green.



5. Click the flyoff to select it and then type a name.

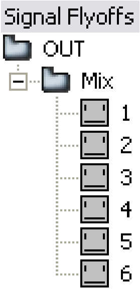


Mixer Output Channel 1

Mixer Output Channel 1

The flyoff will be added to the **Flyoffs** tab, so you can drag it into your design when you want to make a connection to the device it is wired to.

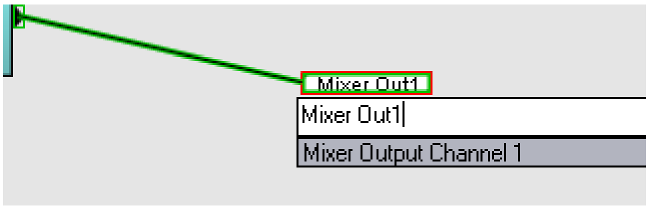
Tip: Using a dot in the flyoff name creates branches in the flyoffs tree to help you organize your flyoffs more easily. Creating flyoffs with the names *OUT.mix1*, *OUT.mix2,...OUT.mix6*, for example, would update the flyoffs tree to look like the picture below.



Signal Flyoffs
- OUT
  - Mix
    - 1
    - 2
    - 3
    - 4
    - 5
    - 6

Renaming a flyoff

- Click the flyoff to select it and then type a new name.



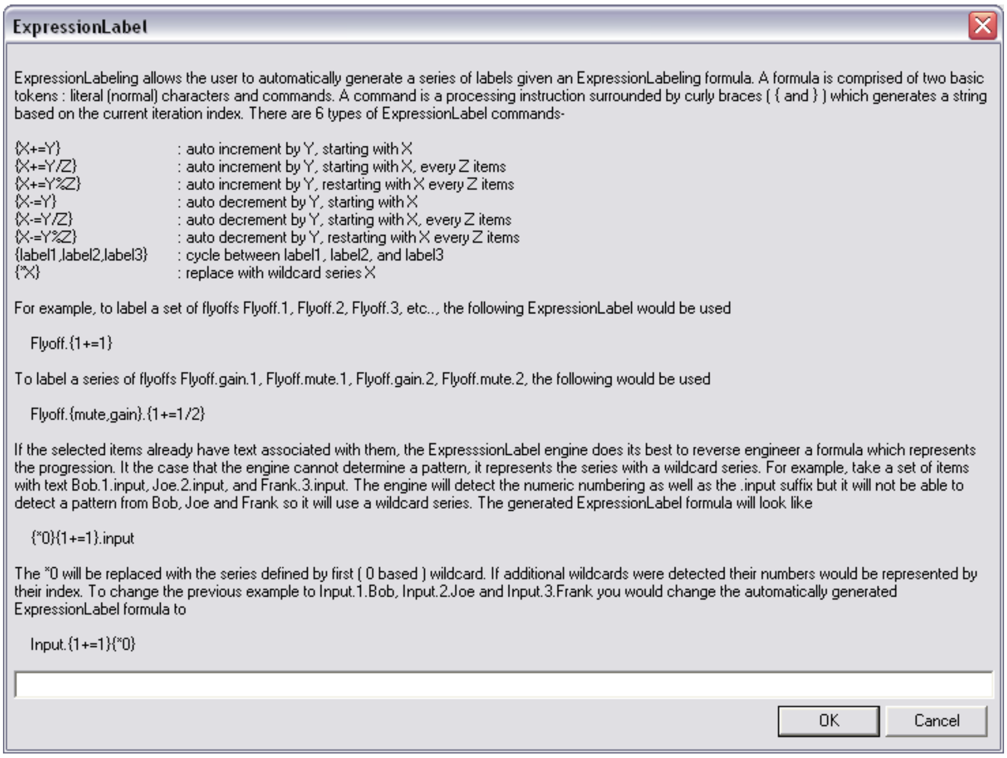
Naming a group of flyoffs automatically

When your design contains a large number of flyoffs, you can save time by naming them using the Expression Labeler. For example, if a mixer has six outputs, you might want to label them *OUT.mix1*, *OUT.mix2*, and so on. Rather than typing the names individually, you can specify a formula in the Expression Labeler that will name them for you.

⏩ *To name a group of flyoffs automatically*

1. Select a group of flyoffs.
2. Press CTRL+SHIFT+F.

   The **ExpressionLabel** dialog box is displayed, showing instructions for creating labels.



ExpressionLabeling allows the user to automatically generate a series of labels given an ExpressionLabeling formula. A formula is comprised of two basic tokens : literal (normal) characters and commands. A command is a processing instruction surrounded by curly braces ({ and }) which generates a string based on the current iteration index. There are 6 types of ExpressionLabel commands-

{X+=Y}	: auto increment by Y, starting with X
{X+=Y/Z}	: auto increment by Y, starting with X, every Z items
{X+=Y%Z}	: auto increment by Y, restarting with X every Z items
{X-=Y}	: auto decrement by Y, starting with X
{X-=Y/Z}	: auto decrement by Y, starting with X, every Z items
{X-=Y%Z}	: auto decrement by Y, restarting with X every Z items
{label1,label2,label3}	: cycle between label1, label2, and label3
{*X}	: replace with wildcard series X

For example, to label a set of flyoffs Flyoff.1, Flyoff.2, Flyoff.3, etc.., the following ExpressionLabel would be used

```
Flyoff.{1+=1}
```

To label a series of flyoffs Flyoff.gain.1, Flyoff.mute.1, Flyoff.gain.2, Flyoff.mute.2, the following would be used

```
Flyoff.{mute,gain}.{1+=1/2}
```

If the selected items already have text associated with them, the ExpresssionLabel engine does its best to reverse engineer a formula which represents the progression. It the case that the engine cannot determine a pattern, it represents the series with a wildcard series. For example, take a set of items with text Bob.1.input, Joe.2.input, and Frank.3.input. The engine will detect the numeric numbering as well as the .input suffix but it will not be able to detect a pattern from Bob, Joe and Frank so it will use a wildcard series. The generated ExpressionLabel formula will look like

```
{*0}{1+=1}.input
```

The *0 will be replaced with the series defined by first (0 based) wildcard. If additional wildcards were detected their numbers would be represented by their index. To change the previous example to Input.1.Bob, Input.2.Joe and Input.3.Frank you would change the automatically generated ExpressionLabel formula to

```
Input.{1+=1}{*0}
```

3. Type a labeling expression in the box.

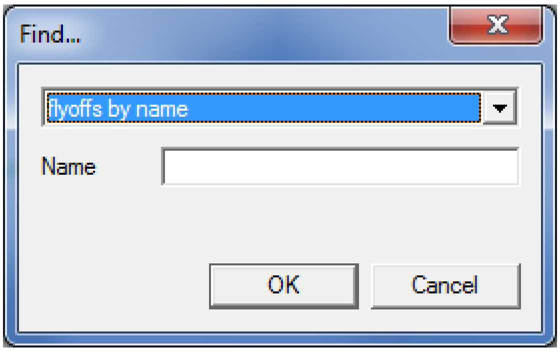
   To label the flyoffs as *OUT.mix1*, *OUT.mix2*, and so on, type:

   ```
   OUT.mix{1+=1}
   ```

Finding a flyoff

1. Press **CTRL+F** to display the **Find** dialog box.

2. In the list, click **flyoffs by name**.



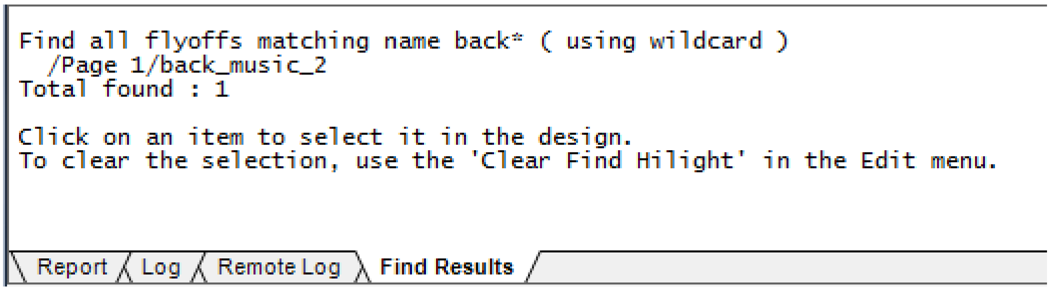
flyoffs by name

3. In the **Name** box, type the name of the flyoff to find.

Tip: You can use an asterisk as a wildcard character.

4. Click **OK**.

   The results will be displayed on the **Find Results** tab at the bottom of the NWare window.



```
Find all flyoffs matching name back* ( using wildcard )
  /Page 1/back_music_2
Total found : 1

Click on an item to select it in the design.
To clear the selection, use the 'Clear Find Hilight' in the Edit menu.
```

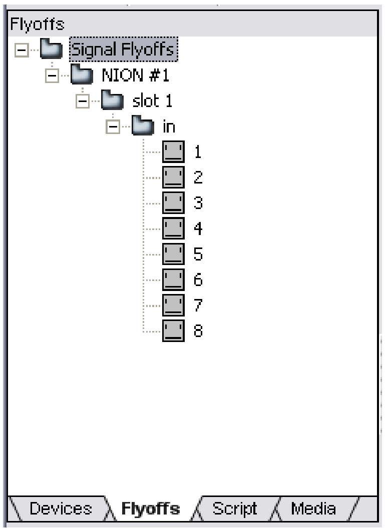
5. Click a result to jump directly to the flyoff in the design.

   The flyoff will be highlighted on the design page. Other devices will be dimmed.

Tip: To switch off the highlighting of the flyoff and dimming of the other controls, right-click the design page, and then click **Clear Find Hilight**.

Routing audio to and from NioNode audio cards

When you add a NioNode to your design and specify input and output cards on the device properties, NWare automatically adds flyoffs to the Flyoffs tab. Here are the flyoffs for a NIO-8ml II card.



When an input card like the NIO-8ml II is present, flyoffs are added under the *in* folder in the tree. When and output card is present, they are added under the *out* folder.

You can wire devices to the flyoffs so that audio can flow from the input connectors on a card to the devices in your design, or from the devices in your design to the output connectors on a card.

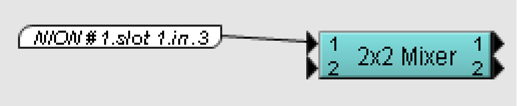
Note: You cannot rename flyoffs from the Flyoffs tab.

Tip: You can locate a flyoff in your design by double-clicking the flyoff on the Flyoffs tab. To switch off the highlighting of the flyoff and dimming of the other controls, right-click the design page, and then click **Clear Find Hilight**.

⏩ *To route audio to and from NioNode audio cards*

1. Drag a flyoff from the **Flyoffs** tab over to the page.
2. On the toolbar, click the button to switch to Wire mode.
3. Drag the flyoff over to the input or output wiring node ▶ on the device.

   A connection line is added. In the example below, a connection is made from connector 3 on an input card inserted in slot 1.

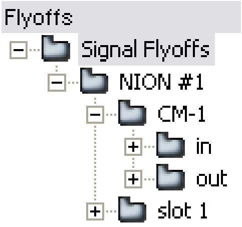


   The flyoff is lit (green) 🟩 on the Flyoffs tab.

Routing audio on CobraNet networks

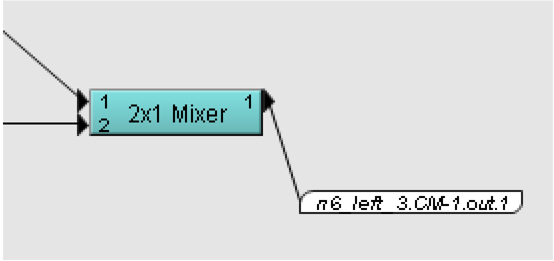
Routing audio via a NioNode

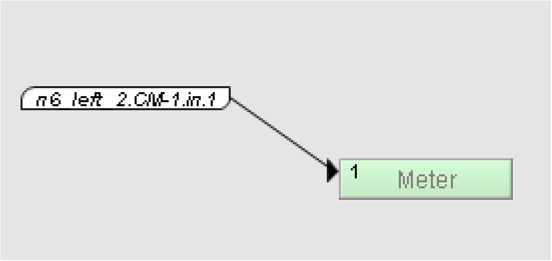
In order to route audio between a NioNode and other devices via a CobraNet network, you must select a CobraNet option in the *Audio Network Configuration* list on the NioNode properties. This adds a CM-1 folder under the NioNode on the Flyoffs tab, and a CM-1 tab to the NioNode block.



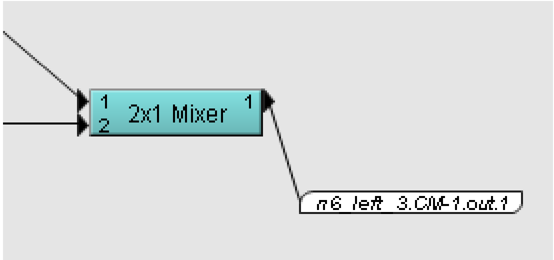
You can wire devices to the CM-1 flyoffs so that audio can flow from devices on the CobraNet network to devices in your design, or vice versa.

In the example below, signals flow from the Mixer device on one NioNode called *n6 left 3*, out to the network, then back in from the network to a NioNode called *n6 left 2*, and finally on to a Meter device.



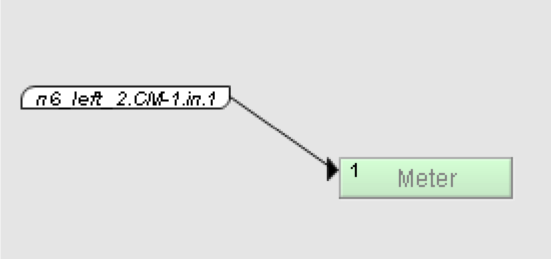


Firstly, you wire the audio output from a device to the input connector on one of the CM-1 output flyoffs, as shown below.



Wiring to the CM-1 output flyoff indicates that the audio data will be transferred to the CM-1 interface on the NioNode *n6 left 3* and then on to the network where it can be received by another device.

In another part of the design, a CM-1 input flyoff for the *n6 left 2* NioNode is wired to a device to indicate that audio data will be received from the network and transferred to the device.



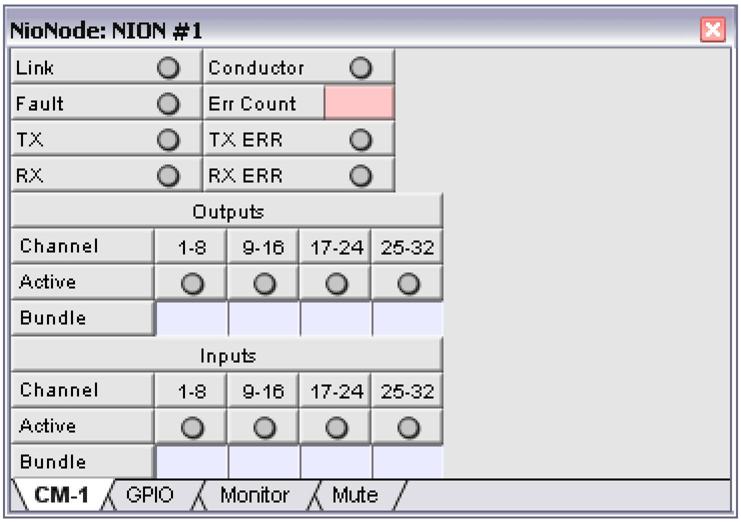
The CM-1 interface supports a maximum of 32 input sub-channels and 32 output sub-channels. In the screenshots above, the example uses output channel 1 on the first NioNode and input channel 1 on the second NioNode.

In order for audio to flow between the 2x1 Mixer device and the Meter device, you need to specify settings on the CM-1 tab on both the transmitting NioNode and the receiving NioNode. Settings on the tab control how the sub-channels are organized into bundles for transmitting and receiving data. You can choose whether to arrange the sub-channels into 4 eight channel bundles or 8 four channel bundles, or you can choose the Advanced option, which allows you to choose how many channels each bundle will contain.

When you use the 4x8 or 8x4 options, data is sent using unicast, so the option you choose determines how many devices can receive the audio data in the bundles. If there are five different audio channels that are to be sent to five different receiving devices, for example, you will need to use the 8x4 option or the Advanced option. The 8x4 option allows you to specify up to eight receivers; the Advanced option, up to 16.

Tip: If the **CM-1** tab is not displayed on a NioNode device, it is because the **Audio Network Configuration** setting on the device properties is set to **None**. It needs to be changed to one of the other values, which are explained below.

When the **CobraNet CM:1 4 8 Channel Bundles** option is selected, the CM-1 tab looks like this:


NioNode: NION #1

Link	○	Conductor	○
Fault	○	Err Count	
TX	○	TX ERR	○
RX	○	RX ERR	○

Outputs

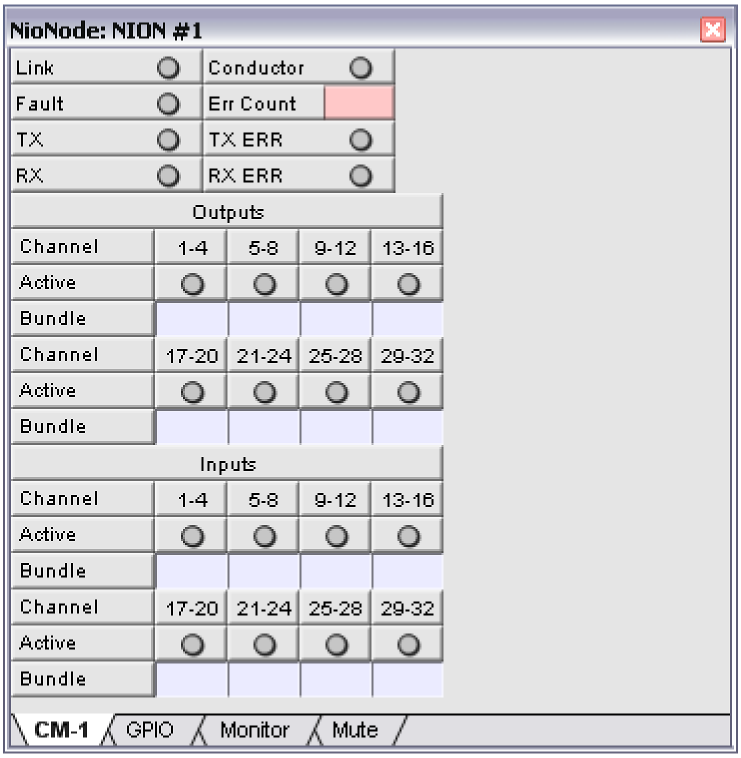
Channel	1-8	9-16	17-24	25-32
Active	○	○	○	○
Bundle				

Inputs

Channel	1-8	9-16	17-24	25-32
Active	○	○	○	○
Bundle				

The CM-1 sub-channels are organized into bundles, with up to eight allowed in each. The bundles are identified by bundle numbers, which are typed into the **Bundle** boxes. The Active LED above each bundle box will light green when the bundle is actually being transmitted or received.

When the **CobraNet CM:1 8 4 Channel Bundles** option is selected, the CM-1 tab looks like this:



NioNode: NION #1

Link	○	Conductor	○
Fault	○	Err Count	
TX	○	TX ERR	○
RX	○	RX ERR	○

Outputs

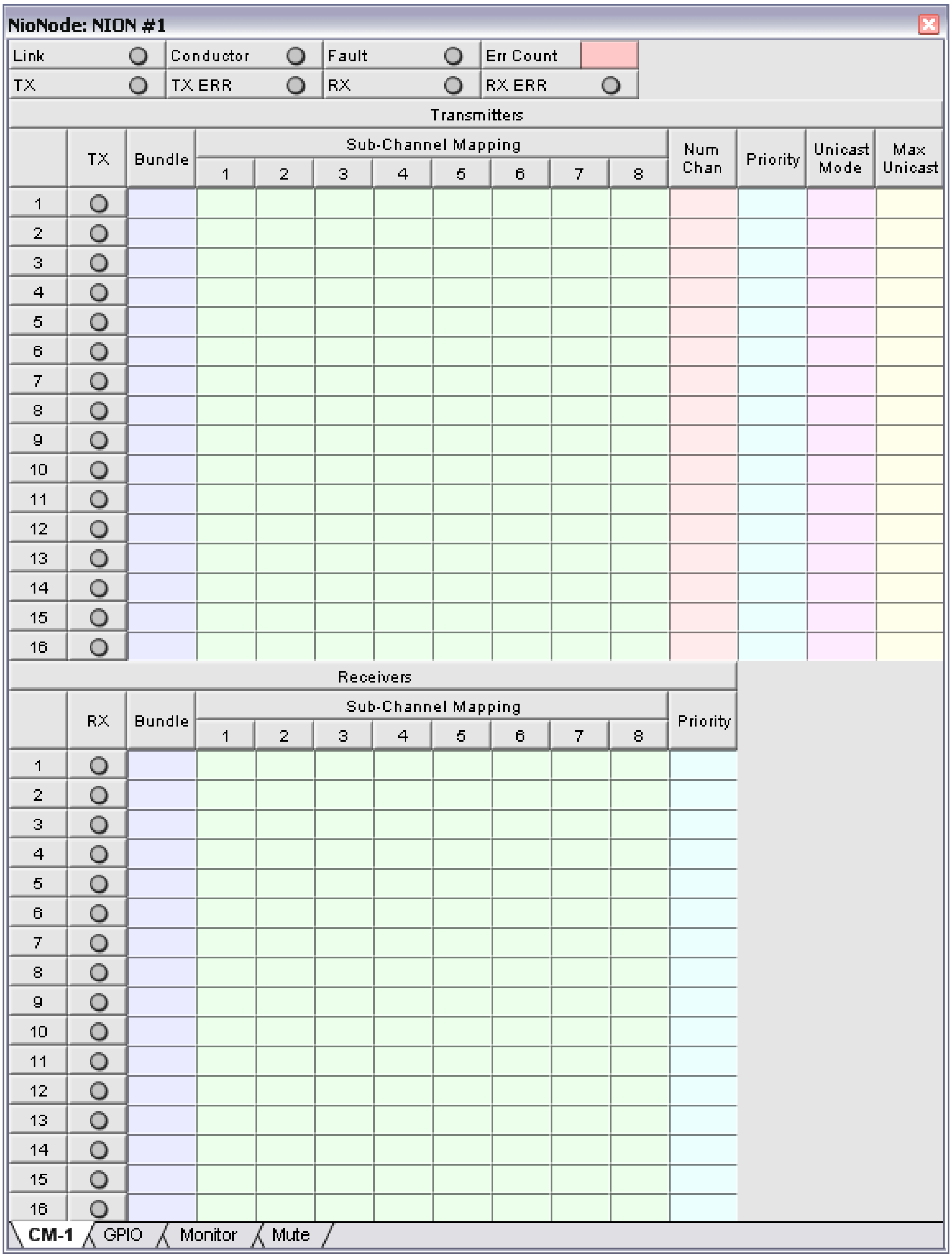
Channel	1-4	5-8	9-12	13-16
Active	○	○	○	○
Bundle				
Channel	17-20	21-24	25-28	29-32
Active	○	○	○	○
Bundle				

Inputs

Channel	1-4	5-8	9-12	13-16
Active	○	○	○	○
Bundle				
Channel	17-20	21-24	25-28	29-32
Active	○	○	○	○
Bundle				

The CM-1 sub-channels are organized into a larger number of bundles, each containing four sub-channels.

When the **CobraNet CM:1 Advanced** option is selected, the CM-1 tab looks like this:



NioNode: NION #1

Link ○	Conductor ○	Fault ○	Err Count
TX ○	TX ERR ○	RX ○	RX ERR ○

Transmitters

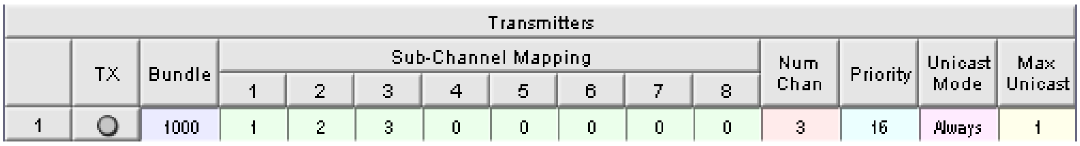
	TX	Bundle	Sub-Channel Mapping								Num Chan	Priority	Unicast Mode	Max Unicast
			1	2	3	4	5	6	7	8				
1	○													
2	○													
3	○													
4	○													
5	○													
6	○													
7	○													
8	○													
9	○													
10	○													
11	○													
12	○													
13	○													
14	○													
15	○													
16	○													

Receivers

	RX	Bundle	Sub-Channel Mapping								Priority
			1	2	3	4	5	6	7	8	
1	○										
2	○										
3	○										
4	○										
5	○										
6	○										
7	○										
8	○										
9	○										
10	○										
11	○										
12	○										
13	○										
14	○										
15	○										
16	○										

Tip: This layout is also used on CAB and nControl devices.

The display is arranged in a 2-D grid with the transmitters and receivers for the CM-1 interface shown as rows and the sub-channel positions shown as columns. The bundles do not have a set number of CobraNet sub-channels, just a maximum of eight. You can specify exactly which sub-channels to put into each bundle and arrange them into the eight available positions. In the example below, transmission bundle number 1000 contains three sub-channels; the remaining five sub-channels are unused.



Transmitters

	TX	Bundle	Sub-Channel Mapping								Num Chan	Priority	Unicast Mode	Max Unicast
			1	2	3	4	5	6	7	8				
1	○	1000	1	2	3	0	0	0	0	0	3	16	Always	1

As there are unused sub-channels in the bundle, the **Num Chan** setting has been set to 3 to indicate that only the first three sub-channel mapping columns are in use, and columns 4-8 can be ignored. The Num Chan setting, which has a default of 8, controls how many audio channels are sent in a bundle. The value should be set to the number of contiguous channels that will be used, as defined in the Sub-channel Mapping boxes. If Num Chan is set to a value that is greater than the number of channels used, then the bundle will be larger than necessary and will result in wasted bandwidth and processor time. Unused channels with a value of 0 are represented in the bundle by header data only (and no audio data will be included). If the bundle contains subchannel numbers that are not used, they will be sent as full audio channels and consume 1 Mbit of bandwidth per channel.

The TX or RX LED next to each bundle assignment will light green when the bundle is actually being transmitted or received.

Each transmitter bundle can optionally have a priority associated with it. This is used in certain redundancy scenarios and allows one machine to backup another. If one of the machines should fail, the other takes over automatically. When there is more than one transmitter on the network with the same bundle assignment, the transmitter with higher priority is used.

Transmitters can send data onto the network using unicast, multi-unicast or multicast. The **Unicast Mode** field controls this. Values for this field are **Always** (always use unicast), and **1-4** (1 meaning there is only one receiver - so use unicast, 2-4 meaning there are multiple receivers, so use multi-unicast).

Multi-unicast is provided because multicast is disruptive to most networks. In the multi-unicast scenario, if the transmitter has multiple receivers it will send each a unicast version of the packet unless the number of receivers is higher than the **Max Unicast** value specified for the transmitter. If that value is exceeded, it will revert to multicast. Multicast can also be specifically requested by using a bundle number in the range 1-255.

Each receiver can also have a priority, but its use is slightly different from the transmitter side. On the receiver side, this priority is used to determine which receivers will be serviced in the event there are not enough resources to fulfill all receiver requests. In this scenario, receivers with higher priorities will be serviced first.

Routing audio via a CAB 4n

Once you have established communications between NWare and the CAB, you will need to prepare the network to pass audio. First, plan out your system. Good system design and careful planning of your resources and audio distribution requirements will ensure a seamless installation of the audio network. You should have a systematic method of assigning your hardware IDs and audio bundles. For example, you may want to number all incoming audio as odd numbered bundles, and outgoing audio as even numbered bundles. You may want to create a numbering scheme that synchronizes the Hardware IDs with the audio bundles so that there is some consistency in the way your system is identified.

A bundle is defined as a group of digital audio channels that can have between 1 and 8 members. For the CAB 4n, we recommend using the 8 4 channel bundles in the NioNode (1 bundle per MM 4 Ch card). These channels are transferred into and out of the NioNode via the individual CM-1 Flyoffs.

To route audio via a CAB 4n

1. Double-click the CAB 4n device to open the block.
2. Click the **Slot x** tab, where x is the slot number where the MM 4 Ch card is installed.
3. On the toolbar, click **Emulate** .

   The project will compile. NWare will switch to *Gesture mode*.

Note: You must emulate or deploy your project before you can set the bundle numbers.

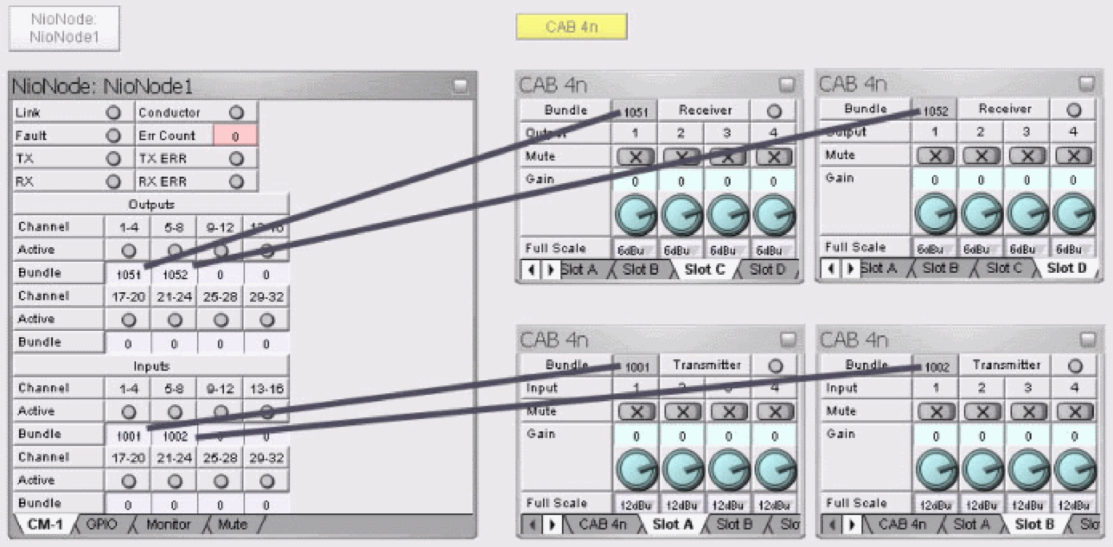
4. Hover the mouse pointer over the first **Bundle** box so the red selection rectangle appears, then type the digits of the bundle number.

   Each bundle can manage between 1 and 8 channels. The range is 0 to 65,535 (0 is off). For multicast bundles, the number must be in the range 1 to 255. For unicast bundles, the number must be in the range 256 to 65,279. Numbers in the range 65280 to 65535 are reserved and cannot be used.
5. Repeat the previous step for the **Bundle** boxes on the other card tabs.
6. Double-click the NioNode device to open the block.
7. On the NioNode CM-1 tab, in the **Bundle** boxes, specify the bundle numbers used in the CAB **Bundle** boxes.

   Receiver bundle numbers on the CAB are specified in the **Outputs** bundle boxes on the NioNode.

   Transmitter bundle numbers on the CAB are specified in the **Inputs** bundle boxes on the NioNode.

In the example below, we have specified audio bundle numbers on the NioNode CM-1 tab. The same bundle numbers are set on the associated CAB 4n slot tabs.



Tip: There are four bundle numbers on a CAB 4n, one for each set of four channel cards that may be loaded.

You can now make audio connections to the CAB by wiring devices to the NioNode CM-1 flyoffs. Using the example above, you could make a connection to the first output connector on the card in slot C by wiring a device to *NioNode1.CM-1.out.1*. Similarly, you could make a connection to the third input connector on the card in slot B by wiring a device to *NioNode1.CM-1.in.7*.

There are many additional variables in larger systems, but this is the basic configuration. What is very important to understand is the relationship between the CAB 4n hardware, the CAB 4n block in NWare and the CobraNet audio transport. Remember, the bundle is specified twice: once in the NioNode to associate the CM-1 CobraNet flyoffs with the selected audio, and again in the CAB 4n block, to connect those flyoffs with the input or output slot of the actual CAB 4n.

Routing audio via a CAB 8n

Once you have established communications between NWare and the CAB, you will need to prepare the network to pass audio. First, plan out your system. Good system design and careful planning of your resources and audio distribution requirements will ensure a seamless installation of the audio network. You should have a systematic method of assigning your hardware IDs and audio bundles. For example, you may want to number all incoming audio as odd numbered bundles, and outgoing audio as even numbered bundles. You may want to create a numbering scheme that synchronizes the Hardware IDs with the audio bundles so that there is some consistency in the way your system is identified.

A bundle is defined as a group of digital audio channels that can have between 1 and 8 members. If all the audio connectors on the CAB 8n are being used as inputs or all as outputs, we recommend using the 4 8 channel bundles configuration in the NioNode (1 bundle per CAB 8n). However, if half the outputs on the CAB 8n are being used as inputs and half as outputs, we recommend using the 8 4 channel bundles configuration. These channels are transferred into and out of the NioNode via the individual CM-1 Flyoffs.

⏩ *To route audio via a CAB 8n*

1. Double-click the CAB 8n device to open the block.
2. Click the **CAB 8n** tab.
3. On the toolbar, click **Emulate** .

   The project will compile. NWare will switch to *Gesture mode*.

Note: You must emulate or deploy your project before you can set the bundle numbers.

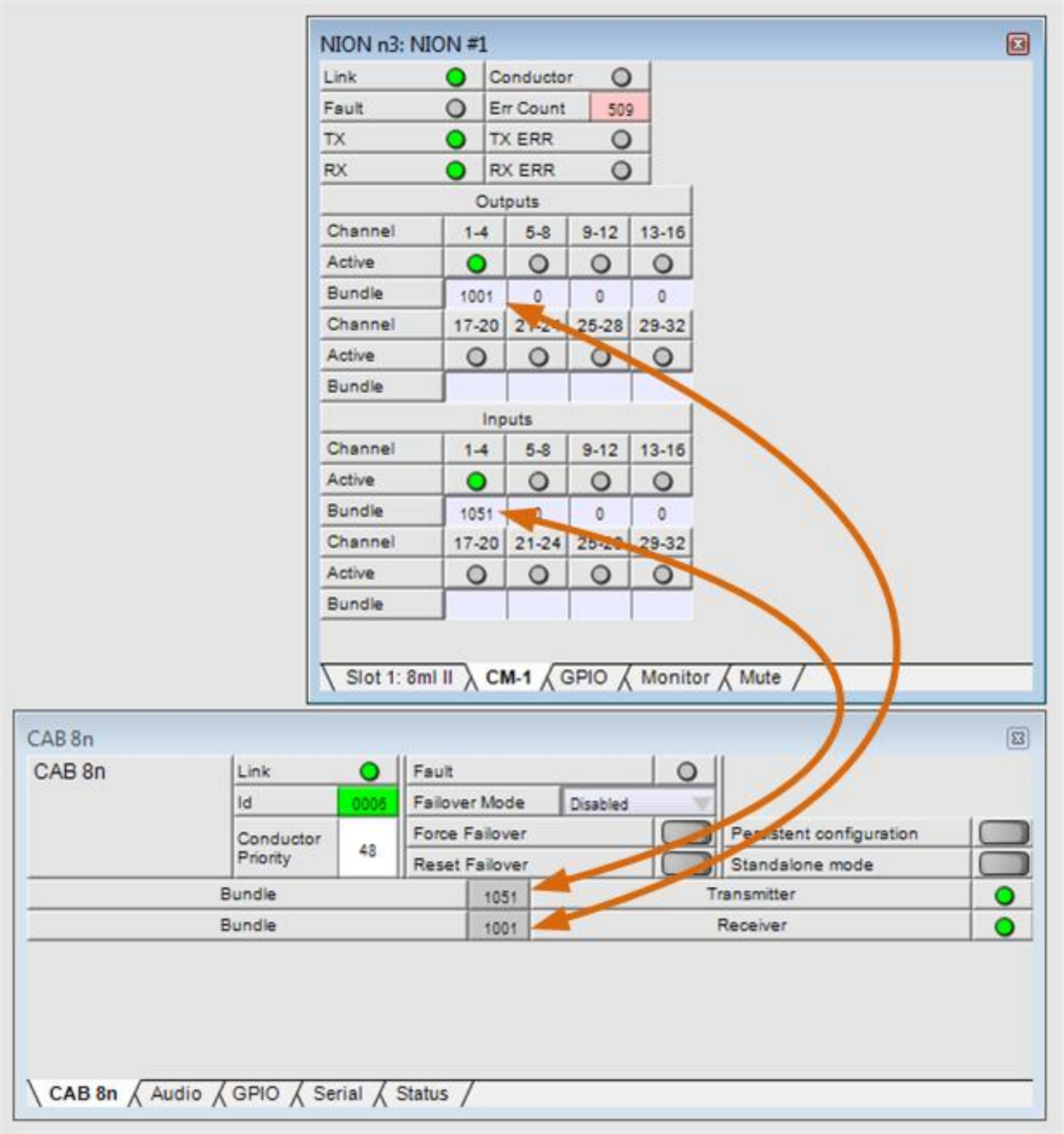
4. If the CAB will be receiving audio into its connectors (from microphones, for example), hover the mouse pointer over the first **Bundle** box (Transmitter), then type the digits of the bundle number.

   Each bundle can manage between 1 and 8 channels. The range is 0 to 65,535 (0 is off). For multicast bundles, the number must be in the range 1 to 255. For unicast bundles, the number must be in the range 256 to 65,279. Numbers in the range 65280 to 65535 are reserved and cannot be used.
5. If the CAB will be transmitting audio from its connectors (to amps and then speakers, for example), hover the mouse pointer over the second **Bundle** box (Receiver), then type the digits of the bundle number.
6. Double-click the NioNode device to open the block.
7. On the NioNode CM-1 tab, in the **Bundle** boxes, specify the bundle numbers used in the CAB **Bundle** boxes.

Receiver bundle numbers on the CAB are specified in the **Outputs** bundle boxes on the NioNode.

Transmitter bundle numbers on the CAB are specified in the **Inputs** bundle boxes on the NioNode.

In the example below, we have specified audio bundle numbers in the NioNode CM-1 tab to match those specified for the CAB 8n.



NION n3: NION #1

Link	●	Conductor	○
Fault	○	Err Count	509
TX	●	TX ERR	○
RX	●	RX ERR	○

Outputs

Channel	1-4	5-8	9-12	13-16
Active	●	○	○	○
Bundle	1001	0	0	0
Channel	17-20	21-24	25-28	29-32
Active	○	○	○	○
Bundle				

Inputs

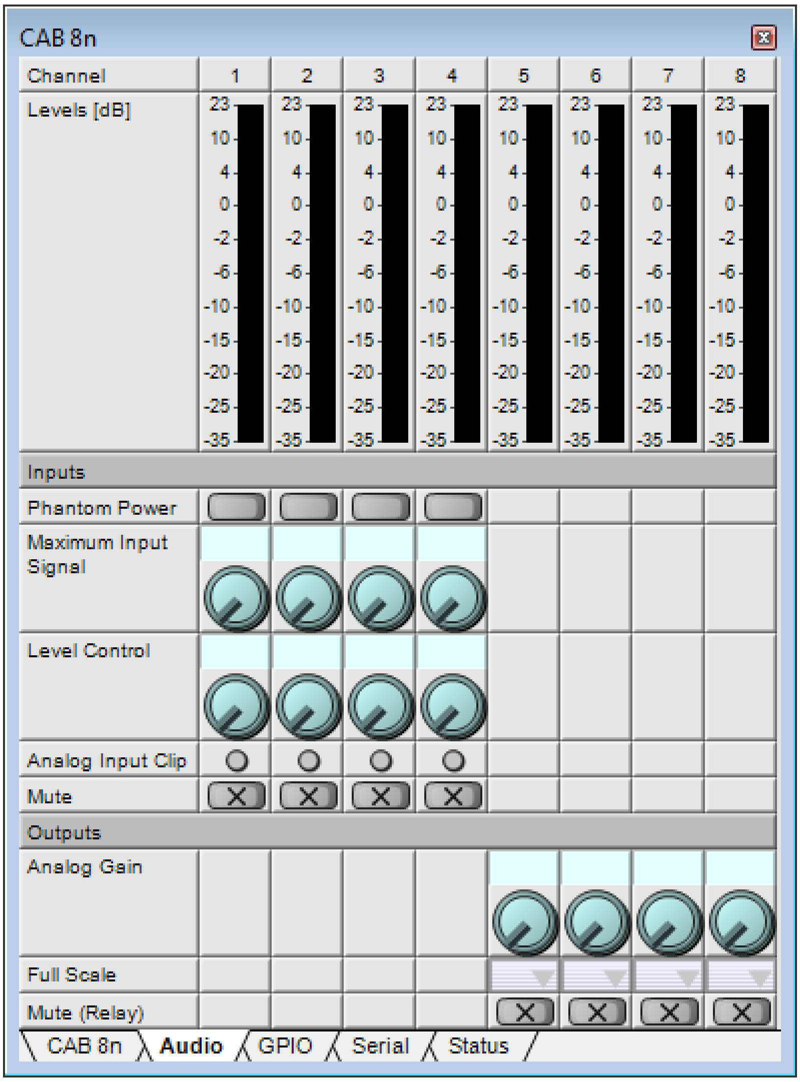
Channel	1-4	5-8	9-12	13-16
Active	●	○	○	○
Bundle	1051	0	0	0
Channel	17-20	21-24	25-28	29-32
Active	○	○	○	○
Bundle				

CAB 8n

CAB 8n	Link	●	Fault	○	
	Id	0006	Failover Mode	Disabled	
	Conductor Priority	48	Force Failover		Persistent configuration
			Reset Failover		Standalone mode

Bundle	1051	Transmitter	●
Bundle	1001	Receiver	●

The CAB 8n has been configured with four inputs and four outputs.



You can make audio connections to the CAB by wiring devices to the NioNode CM-1 flyoffs. Using the example above, you could make a connection to connector 5 on the CAB, which has been set up as a receiver, by wiring a device to *NION #1.CM-1.out.5*. Similarly, you could make a connection to connector 2 on the CAB, which has been set up as a transmitter, by wiring a device to *NION #1.CM-1.in.2*.

There are many additional variables in larger systems, but this is the basic configuration. What is very important to understand is the relationship between the CAB 8n hardware, the CAB 8n block in NWare and the CobraNet audio transport. Remember, the bundle is specified twice: once in the NioNode to associate the CM-1 CobraNet flyoffs with the selected audio, and again in the CAB 8n block, to connect those flyoffs with the input or output connector of the actual CAB 8n.

Routing audio via an nControl AudioScience card

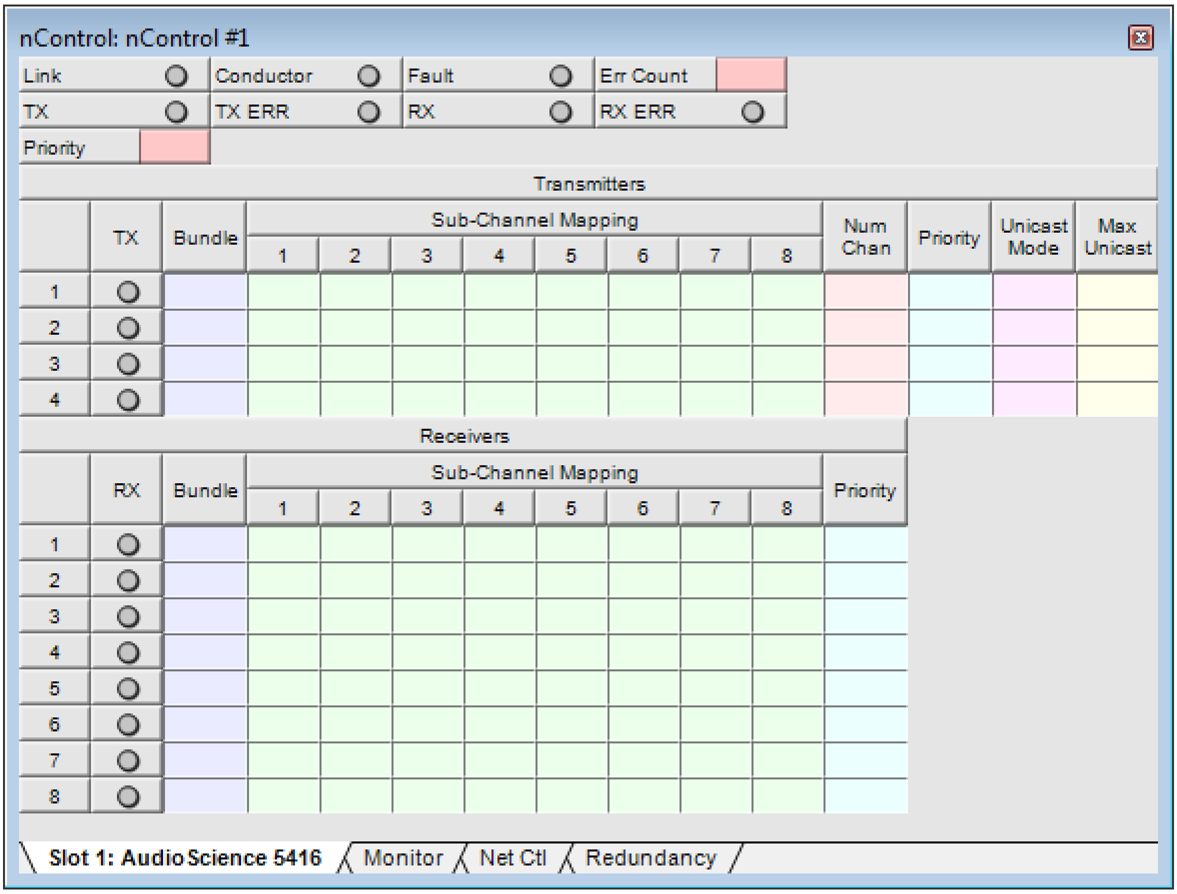
If you want to route audio via AudioScience CobraNet cards installed in the nControl, you need to specify how audio channels will be organized for transmission and reception via the network.

You can use the Media Player device to play back audio files that are stored on the nControl. The data is sent over the CobraNet network via the AudioScience card to one or more receiver devices. The Media Recorder device records audio received via the AudioScience card onto the hard disk inside the nControl.

Note: The AudioScience AS5416 supports 16 input channels and 16 output channels, and up to four of these cards can be installed into an nControl. This creates a theoretical maximum channel count of 64 input channels and 64 output channels. However, the nControl supports a maximum of 64 output channels (for playback), but only 32 input channels (for recording).

The nControl device settings

Settings on the **AudioScience 5416** tab(s) on the nControl device control how transmit and receive channels are organized into bundles.



nControl: nControl #1

Link	○	Conductor	○	Fault	○	Err Count	
TX	○	TX ERR	○	RX	○	RX ERR	○
Priority							

Transmitters

	TX	Bundle	Sub-Channel Mapping								Num Chan	Priority	Unicast Mode	Max Unicast
			1	2	3	4	5	6	7	8				
1	○													
2	○													
3	○													
4	○													

Receivers

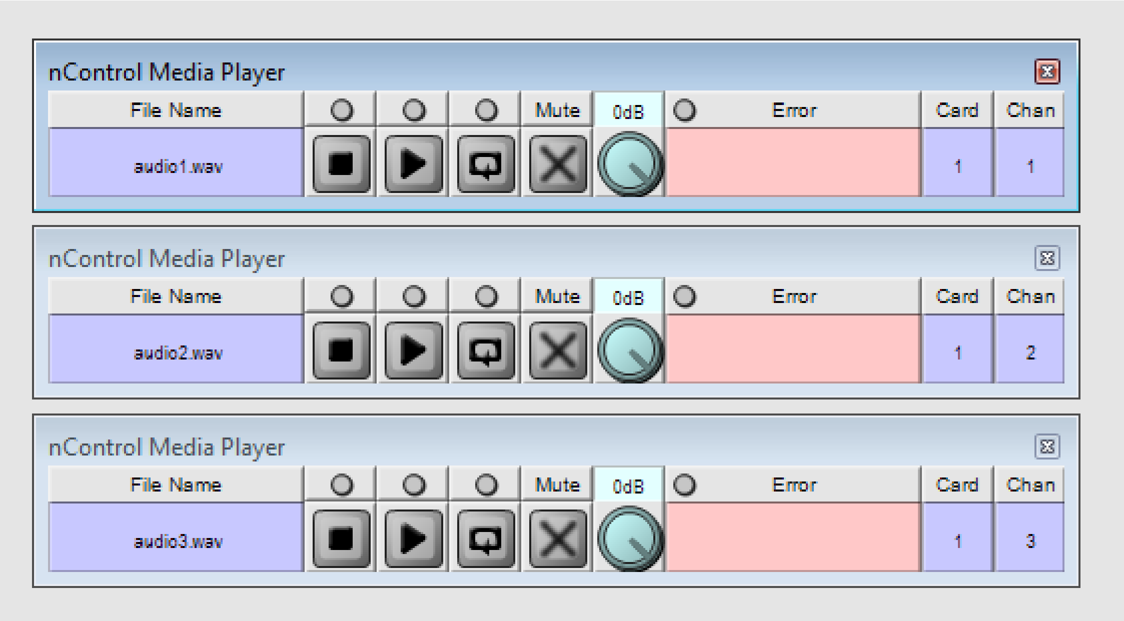
	RX	Bundle	Sub-Channel Mapping								Priority
			1	2	3	4	5	6	7	8	
1	○										
2	○										
3	○										
4	○										
5	○										
6	○										
7	○										
8	○										

Tip: If no **AudioScience 5416** tabs are displayed on an nControl device, it is because the **Expansion Slot 1-5** settings on the device properties are set to **None**. You need to select **CobraNet AudioScience-5416** in at least one of these lists.

Transmit settings

The display is arranged in a 2-D grid with the transmitters for the CobraNet interface shown as rows and the sub-channel positions shown as columns.

Copy the channel number you have specified in the *Chan* box on each nControl Media Player device into an available sub-channel slot under *Sub-Channel Mapping*.



nControl Media Player

File Name	○	○	○	Mute	0dB	○	Error	Card	Chan
audio1.wav	■	▶		✕				1	1

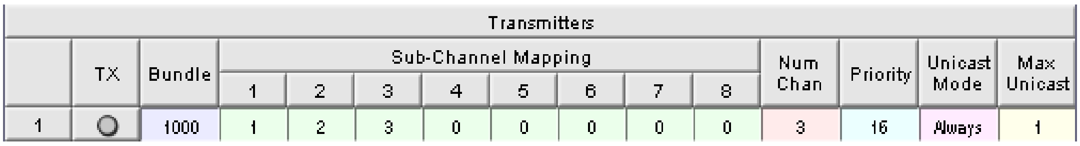
nControl Media Player

File Name	○	○	○	Mute	0dB	○	Error	Card	Chan
audio2.wav	■	▶		✕				1	2

nControl Media Player

File Name	○	○	○	Mute	0dB	○	Error	Card	Chan
audio3.wav	■	▶		✕				1	3

The bundles do not have a set number of CobraNet sub-channels, just a maximum of eight. You can specify exactly which sub-channels to put into each bundle and arrange them into the eight available positions. In the example below, transmission bundle number 1000 contains three sub-channels; the remaining five sub-channels are unused.



Transmitters

	TX	Bundle	Sub-Channel Mapping								Num Chan	Priority	Unicast Mode	Max Unicast
			1	2	3	4	5	6	7	8				
1	○	1000	1	2	3	0	0	0	0	0	3	16	Always	1

As there are unused sub-channels in the bundle, the **Num Chan** setting has been set to 3 to indicate that only the first three sub-channel mapping columns are in use, and columns 4-8 can be ignored. The Num Chan setting, which has a default of 8, controls how many audio channels are sent in a bundle. The value should be set to the number of contiguous channels that will be used, as defined in the Sub-channel Mapping boxes. If Num Chan is set to a value that is greater than the number of channels used, then the bundle will be larger than necessary and will result in wasted bandwidth and processor time. Unused channels with a value of 0 are represented in the bundle by header data only (and no audio data will be included). If the bundle contains subchannel numbers that are not used, they will be sent as full audio channels and consume 1 Mbit of bandwidth per channel.

The TX LED next to each bundle assignment will light green when the bundle is actually being transmitted.

Each transmitter bundle can optionally have a priority associated with it. This is used in certain redundancy scenarios and allows one machine to backup another. If one of the machines should fail, the other takes over automatically. When there is more than one transmitter on the network with the same bundle assignment, the transmitter with higher priority is used.

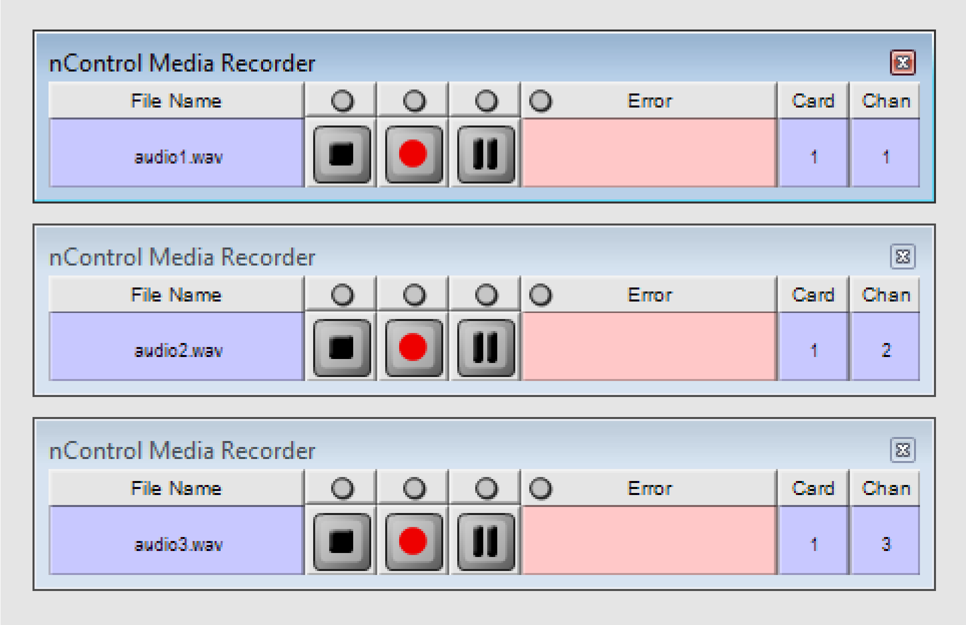
Transmitters can send data onto the network using unicast, multi-unicast or multicast. The **Unicast Mode** field controls this. Values for this field are **Always** (always use unicast), and **1-4** (1 meaning there is only one receiver - so use unicast, 2-4 meaning there are multiple receivers, so use multi-unicast).

Multi-unicast is provided because multicast is disruptive to most networks. In the multi-unicast scenario, if the transmitter has multiple receivers it will send each a unicast version of the packet unless the number of receivers is higher than the **Max Unicast** value specified for the transmitter. If that value is exceeded, it will revert to multicast. Multicast can also be specifically requested by using a bundle number in the range 1-255.

Receive settings

The display is arranged in a 2-D grid with the receivers for the CobraNet interface shown as rows and the sub-channel positions shown as columns.

Copy the channel number you have specified in the *Chan* box on each nControl Media Recorder device into an available sub-channel slot under *Sub-Channel Mapping*.



nControl Media Recorder

File Name	○	○	○	○ Error	Card	Chan
audio1.wav	■	●	❚❚		1	1

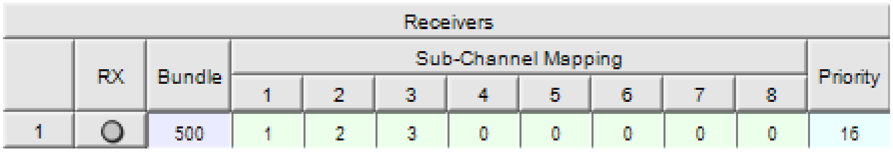
nControl Media Recorder

File Name	○	○	○	○ Error	Card	Chan
audio2.wav	■	●	❚❚		1	2

nControl Media Recorder

File Name	○	○	○	○ Error	Card	Chan
audio3.wav	■	●	❚❚		1	3

The bundles do not have a set number of CobraNet sub-channels, just a maximum of eight. You can specify exactly which sub-channels to use from the available positions in each bundle. In the example below, receive bundle number 500 contains three sub-channels; the remaining five sub-channels are unused.



Receivers

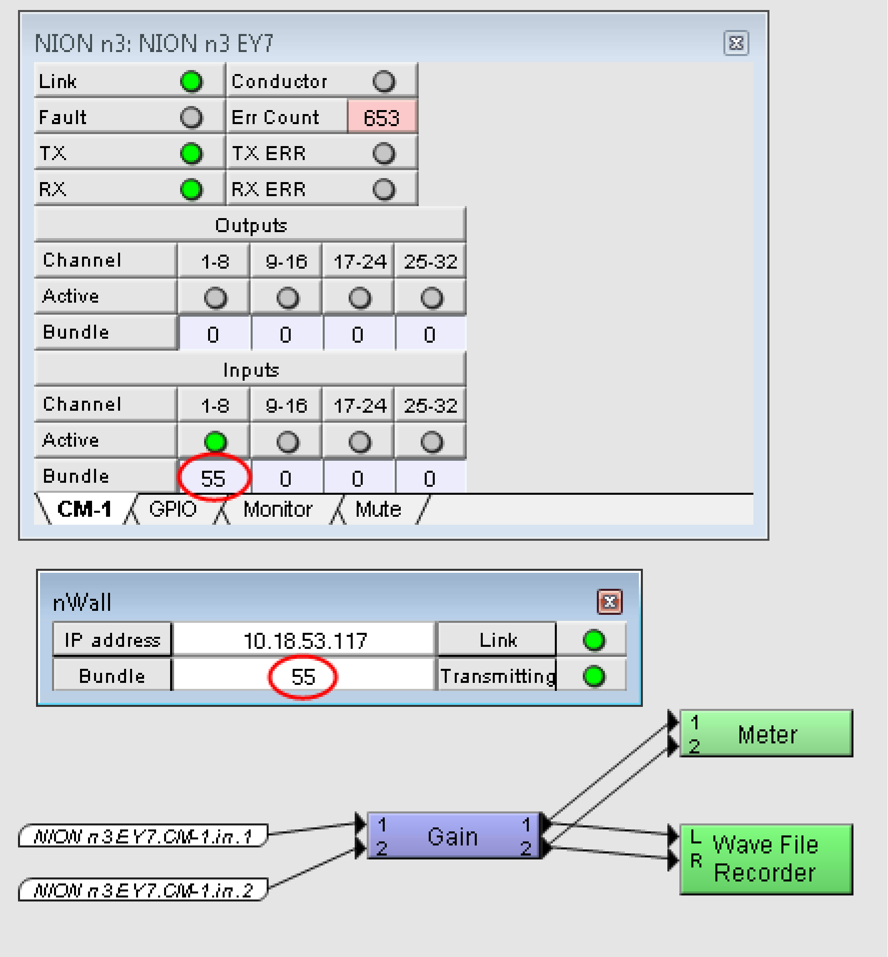
	RX	Bundle	Sub-Channel Mapping								Priority
			1	2	3	4	5	6	7	8	
1	○	500	1	2	3	0	0	0	0	0	16

Each receiver can also have a priority, but its use is slightly different from the transmitter side. On the receiver side, this priority is used to determine which receivers will be serviced in the event there are not enough resources to fulfill all receiver requests. In this scenario, receivers with higher priorities will be serviced first.

Routing audio via an nWall

Once you have assigned a bundle number to the nWall (see *Adding the nWall device and assigning an IP address and bundle number* (on page 69)), it is ready to transmit audio onto the CobraNet network. There is no further set up to do on the nWall device itself, but you will need to specify the bundle configuration on the receiving CobraNet device, as described below.

In our example, we are setting up a NioNode to receive the audio from the nWall via the CobraNet network. The nWall has been configured to transmit using bundle number 55, and the NioNode has been configured to receive audio using the same bundle number. The CM-1 flyoffs can then be wired to other devices to transfer the audio from the nWall to other parts of the design.



NION n3: NION n3 EY7

Link	●	Conductor	○
Fault	○	Err Count	653
TX	●	TX ERR	○
RX	●	RX ERR	○

Outputs

Channel	1-8	9-16	17-24	25-32
Active	○	○	○	○
Bundle	0	0	0	0

Inputs

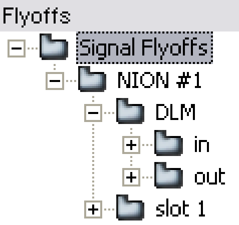
Channel	1-8	9-16	17-24	25-32
Active	●	○	○	○
Bundle	55	0	0	0

nWall

IP address	10.18.53.117	Link	●
Bundle	55	Transmitting	●

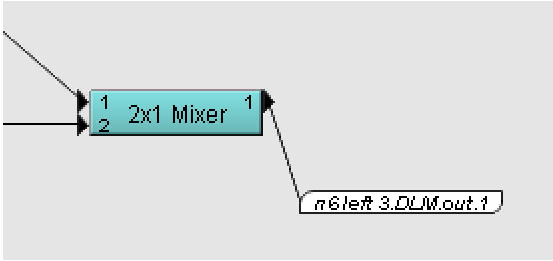
Routing audio on Dante networks

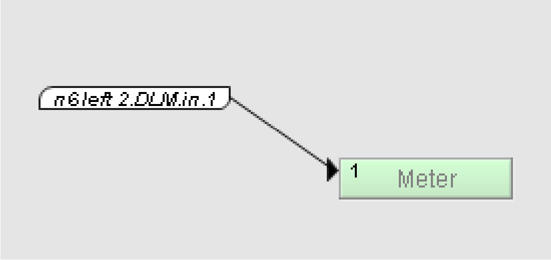
In order to route audio between a NioNode and other devices via a Dante network, you must select the *Dante DLM* option in the *Audio Network Configuration* list on the NioNode properties. This adds a DLM folder under the NioNode on the Flyoffs tab.



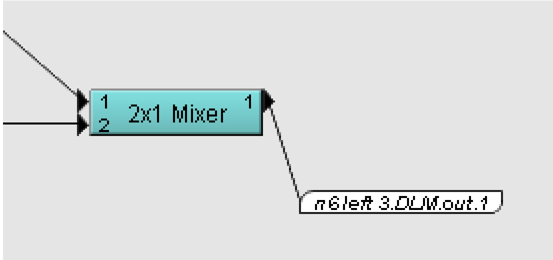
You can wire devices to the DLM flyoffs so that audio can flow from devices on the Dante network to devices in your design, or vice versa. Once you have wired up the devices in the NWare project, you will need to complete the set up process using *Dante Controller* (*http://www.audinate.com/index.php?option=com_content&view=article&id=305&Itemid=167*).

In the example below, signals flow from the Mixer device on one NioNode called *n6 left 3*, out to the network, then back in from the network to a NioNode called *n6 left 2*, and finally on to a Meter device.



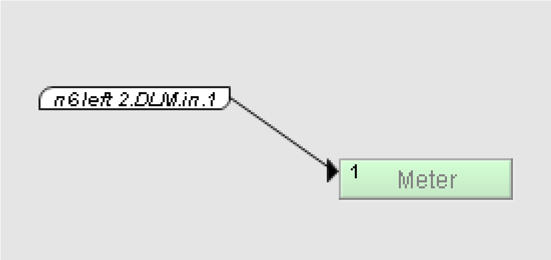


Firstly, you wire the audio output from a device to the input connector on one of the DLM output flyoffs, as shown below.



Wiring to the DLM output flyoff indicates that the audio data will be transferred to the DLM interface on the NioNode *n6 left 3* and then on to the network where it can be received by another device.

In another part of the design, a DLM input flyoff for the *n6 left 2* NioNode is wired to a device to indicate that audio data will be received from the network and transferred to the device.



The DLM interface supports a maximum of 32 input sub-channels and 32 output sub-channels. In the screenshots above, the example uses output channel 1 on the first NioNode and input channel 1 on the second NioNode.

Once the wiring is complete in NWare, open the Dante Controller program and create audio connections between Dante transmitters and receivers with the same numbers as the DLM flyoffs you used in your NWare design. These will be listed under the NioNodes you are using (*n6left 2* and *n6left 3* in the example). This completes the procedure.

Note: Configuration of Dante audio network connections is only supported through 3rd party applications, such as Audinate's Dante Controller software. This software supports 32 channels of 48k audio in and out.

For more information on using the Dante Controller, refer to the Audinate documentation and the *Audinate website* (*http://www.audinate.com*). For a simple worked example in this manual, see *Worked example* (on page 155).

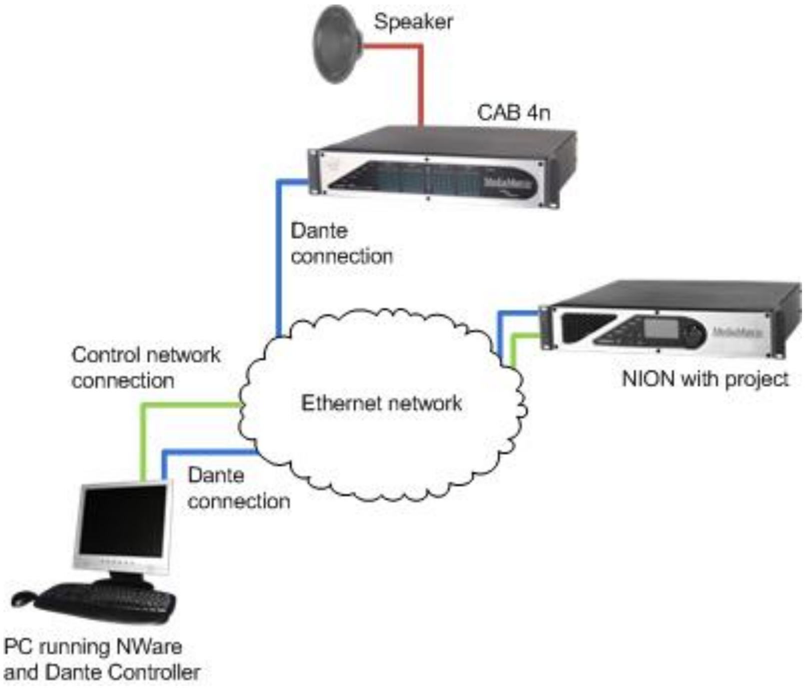
Routing audio via a CAB 4n

Once you have established communications between NWare and the CAB, you can make connections to the CAB inputs and outputs using the NioNode DLM flyoffs. By default, DLM input flyoffs 1-4 are used for the first CAB TX card , 5-8 for the second, and so on. Similarly, output flyoffs 1-4 are used for the first CAB RX card, 5-8 for the second, and so on.

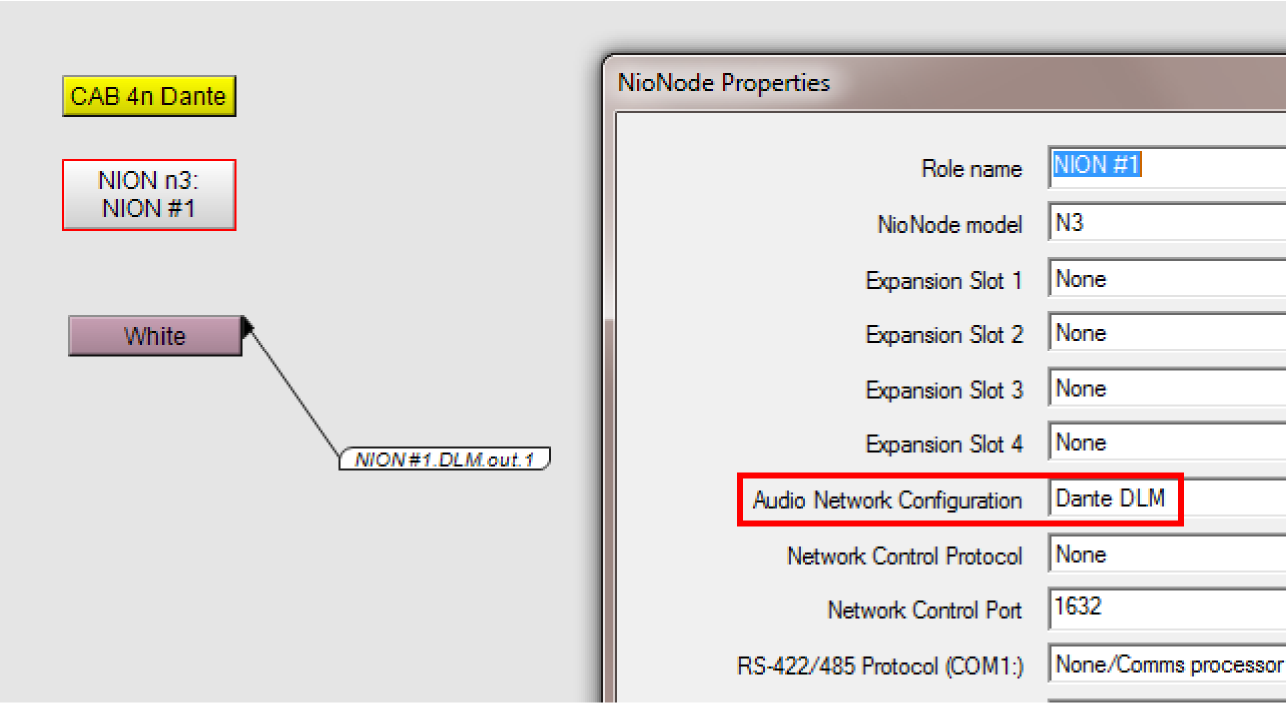
If you want to use different DLM flyoffs, you will need to configure the NioNode and CAB inputs and outputs in the Dante Controller software. For more information on using the Dante Controller, refer to the Audinate documentation and the *Audinate website* (*http://www.audinate.com*).

Worked example

In the example below, a NioNode and a CAB 4n fitted with an RX (blue) card are wired to the Dante network. The CAB has a speaker wired to the first output channel on the RX card.



In NWare, a NioNode is added to the project, with the *Audio Network Configuration* set to *Dante DLM*. We will be routing audio via a Dante CAB 4n, so one of these blocks is also added. The audio source will be a white noise generator.



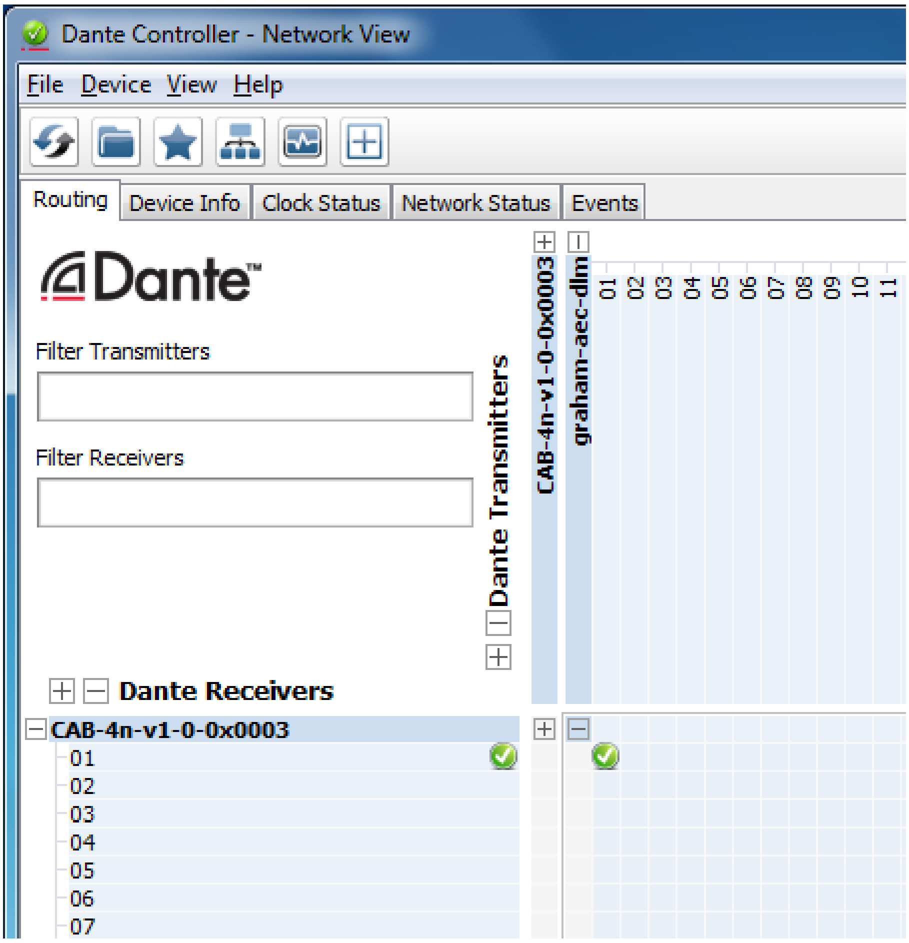
NioNode Properties

Role name	NION #1
NioNode model	N3
Expansion Slot 1	None
Expansion Slot 2	None
Expansion Slot 3	None
Expansion Slot 4	None
Audio Network Configuration	Dante DLM
Network Control Protocol	None
Network Control Port	1632
RS-422/485 Protocol (COM1:)	None/Comms processor

The noise generator is wired to the first DLM output flyoff (dragged over to the page from the Flyoffs tab). This means the audio will be generated on the NioNode, then made available on the Dante network via the first *transmitter*.

In Dante Controller, a connection is made between the NioNode (graham-aec-dlm) and the CAB (CAB-4n-v1-0-0x0003) simply by clicking on the matrix. The first transmitter from the NioNode is connected to the first receiver on the CAB.

Receivers 1-4 for the CAB correspond to the four output channels on the first RX card fitted to the unit. Therefore, to use channel 1, we make a connection to receiver 1 on the grid. Transmitter 1 corresponds to the first output channel on the DLM fitted to the NioNode, which is wired to the noise generator in the project.



When the connection is established, and the gain adjusted on the white noise generator, audio will be heard from the speaker (it will also be indicated by bars on the meter on the front of the CAB). You can use this simple example to experiment, and pass audio through a more complex project.

Using the signal probe to check connections in a project

The signal probe is used to check the audio paths between devices by connecting to the wires in the project.

The *Probe point valid* LED lights when the audio wire or wiring node ▶ you are testing is connected to an audio device, e.g. a pink noise generator, and not a logic device, e.g. an AND gate. The device also displays the device ID and port type.

Note: The Signal Probe is designed to be used when the project has been deployed and audio can pass between the devices. The *Probe point valid* LED only works when the project has been deployed.

Tips:

- You can connect the output wiring node on the Signal Probe block to a meter to instantly see when a signal is present at the test point.
- As an alternative to using the Signal Probe device, you can connect an oscilloscope to an output connector on a NioNode to check for the presence of a signal.

To use the signal probe to check connections in a project

1. Emulate or deploy the project.
2. Double-click the Signal Probe device block to open it.
3. Click the signal probe icon .

   The cursor will change to a signal probe.
4. Click a wire or wiring node ▶ to view the device ID and port type, and to monitor the audio at that place in the design.

   The signal probe will stay connected to the point that was clicked, allowing continuous monitoring.

   If the point you have clicked is connected to an audio device (and not a logic device, for example), the *Probe point valid LED* will be lit.

   To release the probe, click the Signal probe icon.

   To stop using the probe, drag the Signal Probe icon over to the Signal Probe device block, then click the gray box in the top left corner.

Monitoring audio in your design using your PC sound card

If you are hosting your project on a NioNode, you can monitor the audio output from devices in the design using headphones or speakers connected to your PC sound card. This is handy for debugging.

The device that allows monitoring in this way is called the Wave File Recorder. It is available for NioNodes, but not other node types. There are different variants of the device for mono and stereo audio and different sample rates.

Notes:

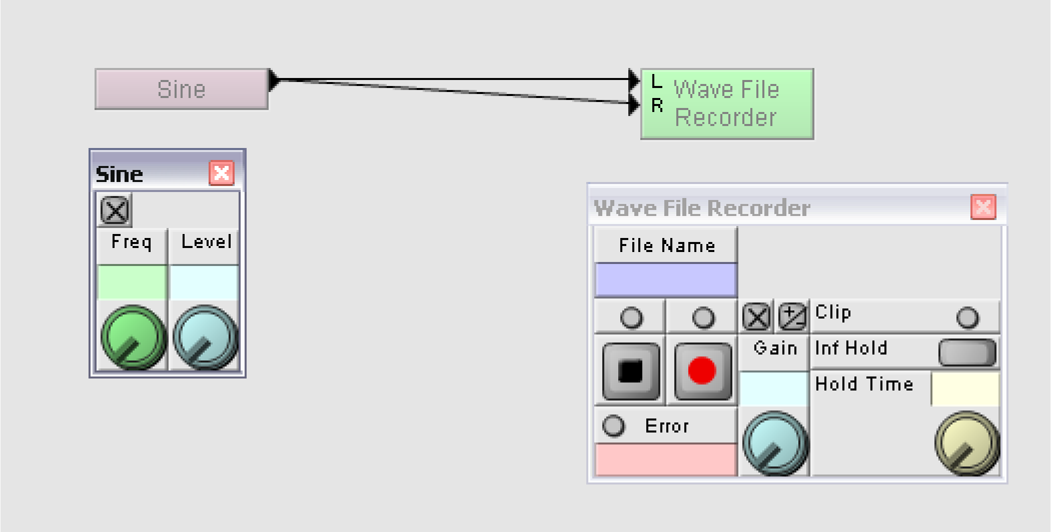
- The sample rate for the player or recorder device must be set to a multiple of the project sample rate. For example, if the system sample rate is 48KHz (which is typical), then realistic values for the sample rate are 48000, 24000, or 12000.
- The interpolation value (for the player device) or decimation value (for the recorder device) when multiplied with the sample rate must equal the project sample rate. For example, if the sample rate is set to 12000 and the project sample rate is 48000, then the correct setting for interpolation or decimation is 4x.
- When you choose a player or recorder device with a particular sample rate, the interpolation or decimation is set automatically by NWare, according to the project sample rate you are using.

- The number of Wave File Players and Wave File Recorders you can add to your project is limited by the available memory on the NioNode that is hosting the project. If you add too many of these devices, an error will be displayed when the project is deployed. The memory usage is affected by a number of factors, including the sample rate and whether mono or stereo is selected.
- You cannot monitor audio in emulation mode – you must deploy the project.

To monitor audio in your design using headphones

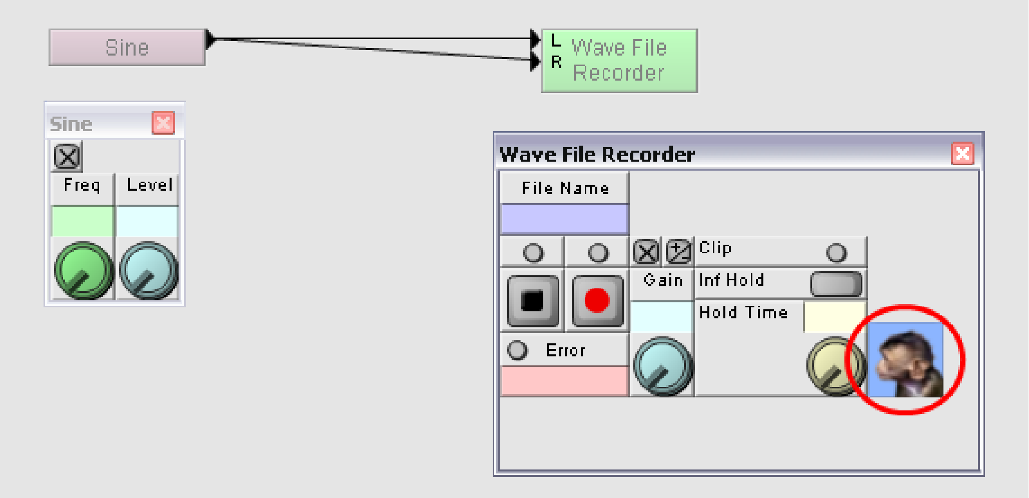
1. In the device tree, expand the **DSP** / **Wave File Recorders** folder.
2. Expand the **Stereo** or **Mono** folder to show the Wave File Recorder for the sample rate used in your project.
3. Drag a **Wave File Recorder** device over to the design page.
4. Wire the Wave File Recorder to the device you want to monitor.

   In the simple example below, we are monitoring the output from a Sine Wave Generator.



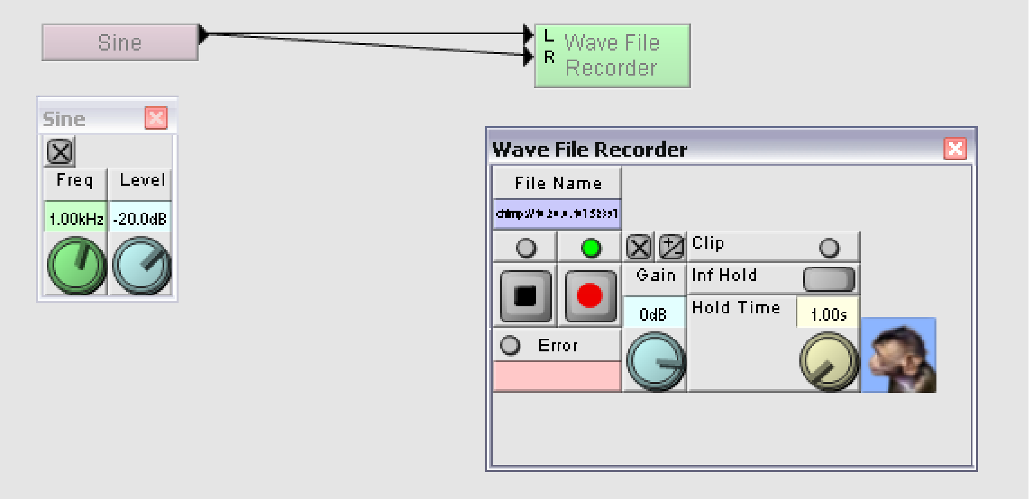
5. Right-click the **Wave File Recorder** device block, and then click **Device Properties**.
6. Click **Advanced**, and then select the **enable CHIMP** check box.
7. Click **OK**, and then again to close the Wave File Recorder device.

The **CHIMP** button is shown on the Wave File Recorder control surface, allowing you to switch to CHIMP mode.



8. Deploy the project.
9. When you are ready to start listening to the audio, click the **CHIMP** button.

   The LED above the record button will be lit (green). The *File Name* box will be updated to show the IP address of the PC and port number used. If you click the button again, a new port number will be used.



Freq 1.00kHz
Level -20.0dB
File Name chimp://[illegible]
Gain 0dB
Hold Time 1.00s

Tip: If you cannot hear the audio you expect to hear, check that the frequency is in audible range. You can use a meter if you want a visual indication that a signal is present.

Chapter 6

Deploying and emulating projects

In This Chapter

Deploying a project

When you want to run your project on a device, such as a NioNode or nControl node, you need to deploy it.

Tips:

- If you are connected to a project on a node and you make changes to the project, you must deploy the project in order to see the outcome of the change.
- NioNodes, nControl nodes and nTouch 180 nodes can all host projects. If none of these types of nodes are available on the network, you can use the Emulate function to test out some of the functionality in your project by running it on your PC. For more information, see *Emulating a project without hardware* (on page 167).

When a project is deployed, the following tasks are performed:

1. A design check is carried out.
2. The project is compiled and DSP code is generated.
3. The results are downloaded to the target nodes.
4. The NWare PC is connected to the target nodes, ready for control.

   The pointer will change to a ✋, NWare will switch to gesture mode, and you will be able to gesture controls.

The design check generates a report in the output frame which displays results indicating hardware usage and any problems encountered in the design.

This phase requires that the NWare PC be networked with the target set of nodes onto which a project is to be installed. During deployment, attached nodes are discovered and displayed to the designer. The designer then assigns the roles that exist within the project to the nodes. Role assignment results in each node downloading, and storing locally, the configuration data that it needs to fulfill its role during runtime.

Once a project has been deployed, the set of nodes targeted by the project perform the signal and control processing contained in the design. One or more PCs running NWare or Kiosk can connect to the target nodes to control and monitor them. Third-party control systems can connect as well.

Tips:

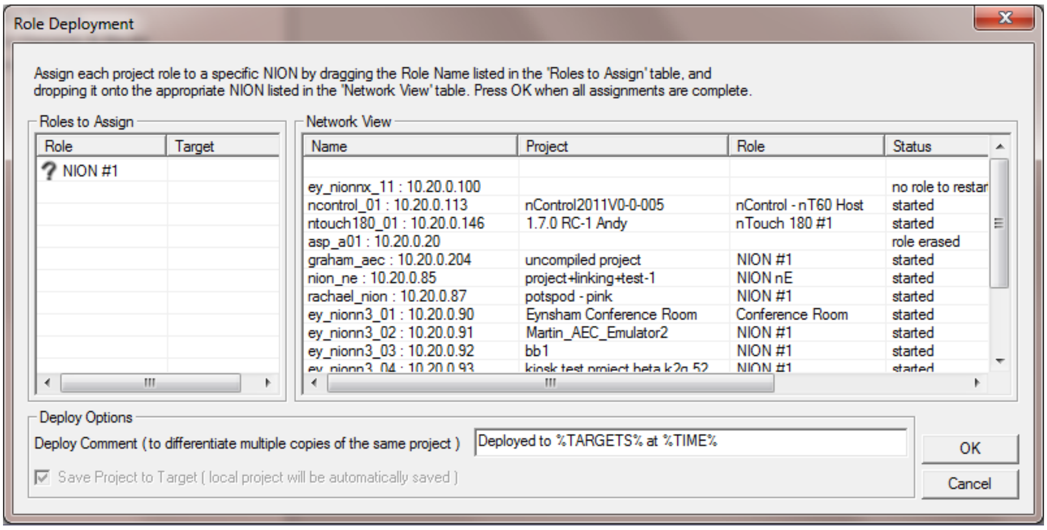
- Once a project has been deployed to a node, if the node is power cycled, the project is restarted automatically.
- If you deploy a project to a node that does not meet the hardware specification of the corresponding NWare device, a warning will be displayed. You can click OK to continue or cancel the operation. This warning is displayed, for example, if a NIO-4x4 card has been specified on the properties of a NioNode, but the NION you are deploying to is not fitted with this card. In the case of different NION models with different numbers of DSP chips, you will be able to deploy the project to the NioNode if the requirements of the project do not exceed the capacity of the DSPs fitted to the NION.

To deploy a project

1. On the toolbar, click the **Deploy** button .

The project will be compiled and a compilation report will be produced. For more information, see *Understanding the compilation report* (on page 164).

The **Role Deployment** dialog box is displayed.



Role Deployment

Assign each project role to a specific NION by dragging the Role Name listed in the 'Roles to Assign' table, and dropping it onto the appropriate NION listed in the 'Network View' table. Press OK when all assignments are complete.

Roles to Assign

Role	Target
? NION #1	

Network View

Name	Project	Role	Status
ey_nionnx_11 : 10.20.0.100			no role to restar
ncontrol_01 : 10.20.0.113	nControl2011V0-0-005	nControl - nT60 Host	started
ntouch180_01 : 10.20.0.146	1.7.0 RC-1 Andy	nTouch 180 #1	started
asp_a01 : 10.20.0.20			role erased
graham_aec : 10.20.0.204	uncompiled project	NION #1	started
nion_ne : 10.20.0.85	project+linking+test-1	NION nE	started
rachael_nion : 10.20.0.87	potspod - pink	NION #1	started
ey_nionn3_01 : 10.20.0.90	Eynsham Conference Room	Conference Room	started
ey_nionn3_02 : 10.20.0.91	Martin_AEC_Emulator2	NION #1	started
ey_nionn3_03 : 10.20.0.92	bb1	NION #1	started
ey_nionn3_04 : 10.20.0.93	kiosk test project beta k2a 52	NION #1	started

Deploy Options

Deploy Comment (to differentiate multiple copies of the same project) Deployed to %TARGETS% at %TIME%

☑ Save Project to Target (local project will be automatically saved)

The **Roles to Assign** list shows the roles you have created by adding nodes to your design (and naming them uniquely).

The **Network View** list shows the nodes that NWare has found on the network.

Tip: If you have deployed a role to a device recently, NWare will prompt you to deploy it to the same device, and will do this automatically after a few seconds. This feature can be switched off on the **Advanced** tab of the **User Preferences** dialog box, by clearing the **Enable Autodeploy countdown** check box.

2. In the **Roles to Assign** list, drag each role over to a node in the **Network View**.

   A green tick ✓ will be shown next to each role to show it has been assigned successfully.

3. If you want to change the comment that is displayed next to the project details in the Kiosk project list, type in the **Deploy Comment** box.

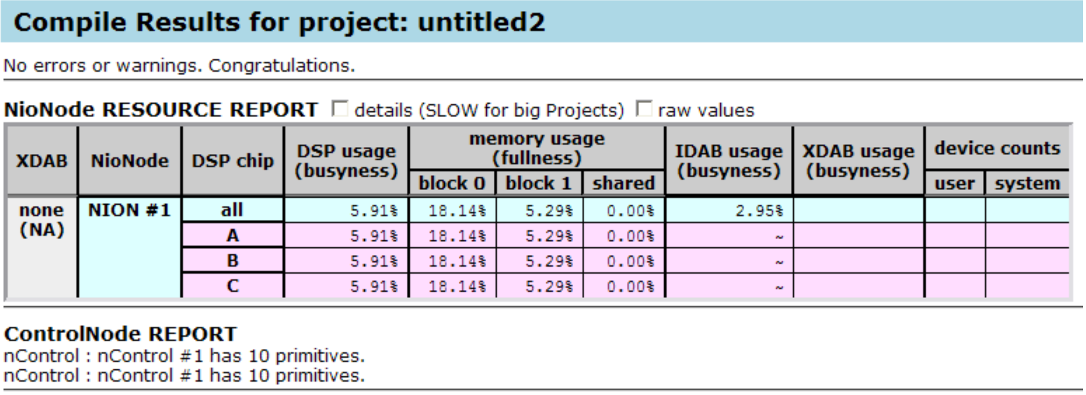
   The project list is displayed when you start Kiosk. For more information, see *Starting Kiosk and selecting a project* (on page 212).

   The **Save Project to Target** check box is automatically selected for new projects. This will save the project file (.npa) to the target device, enabling you to upload it after deployment and make changes. For more information, see *Uploading a project to NWare and connecting to it* (on page 168). If you load a project created in an older version of NWare, the check box will be selected or cleared according to its state when the project was last saved.

4. Click **OK**.

Understanding the compilation report

When the project design is checked at compile time, an HTML-based report is generated. This report includes notes, warnings, errors and a resource report table showing the target resources used by the design.



Compile Results for project: untitled2

No errors or warnings. Congratulations.

NioNode RESOURCE REPORT □ details (SLOW for big Projects) □ raw values

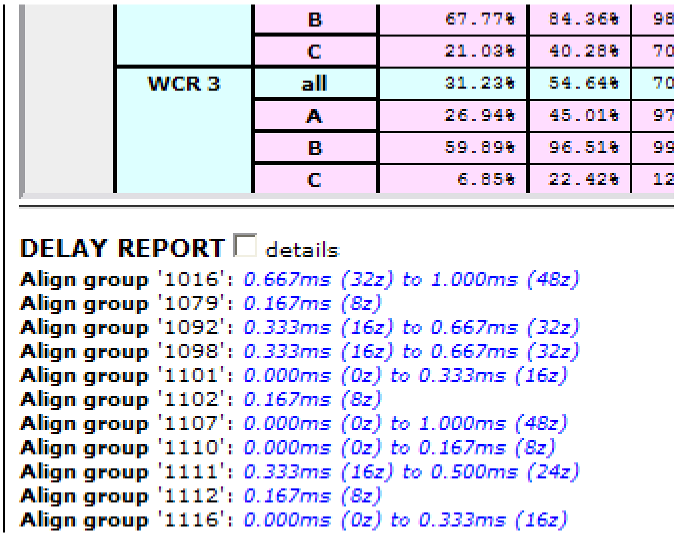
XDAB	NioNode	DSP chip	DSP usage (busyness)	memory usage (fullness)			IDAB usage (busyness)	XDAB usage (busyness)	device counts	
				block 0	block 1	shared			user	system
none (NA)	NION #1	all	5.91%	18.14%	5.29%	0.00%	2.95%			
		A	5.91%	18.14%	5.29%	0.00%	~			
		B	5.91%	18.14%	5.29%	0.00%	~			
		C	5.91%	18.14%	5.29%	0.00%	~			

ControlNode REPORT
nControl : nControl #1 has 10 primitives.
nControl : nControl #1 has 10 primitives.

Many of the notes, warnings and errors, and the entries in the resource report table are clickable links that will jump to the associated objects in the design. When you click cells in the table, they will be highlighted. You can switch off the highlighting by clicking in another part of the table.

Interpreting the delay report

The compilation report includes a section called the delay report, which provides important information about the signal processing delays in your design.



		B	67.77%	84.36%	98
		C	21.03%	40.28%	70
	WCR 3	all	31.23%	54.64%	70
		A	26.94%	45.01%	97
		B	59.89%	96.51%	99
		C	6.85%	22.42%	12

DELAY REPORT □ details
Align group '1016': *0.667ms (32z) to 1.000ms (48z)*
Align group '1079': *0.167ms (8z)*
Align group '1092': *0.333ms (16z) to 0.667ms (32z)*
Align group '1098': *0.333ms (16z) to 0.667ms (32z)*
Align group '1101': *0.000ms (0z) to 0.333ms (16z)*
Align group '1102': *0.167ms (8z)*
Align group '1107': *0.000ms (0z) to 1.000ms (48z)*
Align group '1110': *0.000ms (0z) to 0.167ms (8z)*
Align group '1111': *0.333ms (16z) to 0.500ms (24z)*
Align group '1112': *0.167ms (8z)*
Align group '1116': *0.000ms (0z) to 0.333ms (16z)*

The delays are computed by tracing the signal path from a source (typically an analog input, or possibly a signal generator) to a sink (typically an analog output, but possibly a device that combines inputs such as a mixer).

Signals pass through an A/D converter (which has a delay that is included in the computation) to a DSP; they may then be passed on one or more times to other DSPs. Each of these transfers adds to the delay. These transfers may be between DSPs on the same NioNode, or may be to another NioNode via an XDAB connection. The signal is then sent out through a D/A converter, which again adds to the delay. All these delays are totaled. All possible paths are computed, from every input to every output. The delay report identifies the maximum delay found, as well as the minimum delay. The information is shown as a period of time (in milliseconds) and number of samples (for example, 72z is 72 samples of delay).

Tips:

- If you select the *details* check box, each individual delay is reported for every input and output connector on each device.
- You can click on the delay time to highlight the device doing the combining in your project. This information can be useful if you are doing delay sensitive combining, such as a mix-minus. Typically, if you are not doing a mix-minus you can ignore all these groups.

The report is presented in alignment groups. The most important alignment group is the *analog output group*. This group reports the minimum and maximum delay from any analog input to any analog output. For each combining device, such as a mixer, additional alignment groups are created. Each of these groups is identified by a device ID.

Notes:

- If you select *Auto-compensate signal delays* in the project properties, the report will still report minimum delays, but will indicate they have been compensated to the longest delay time.
- The delay reported for the analog outputs is slightly high, about 410uS (.41ms) at 48k.
- AES cards are included in the analog output group. As above, the actual delay will be slightly less than the reported delay.
- Delay lines in the design are not included in the delay calculation as the delay length is variable and under the user's control.
- CobraNet delays are not computed. You will need to add the actual CobraNet delay to the delays reported in the CobraNet group.

Minimizing delay

Reducing the *processing vector length* or increasing the *sample rate* in the project properties, can reduce the delay. However, each of these decreases the amount of processing each DSP can perform. Because of this, the compiler may need to add DSP to DSP transfers, causing the delay to increase.

Automatic signal delay compensation

NWare provides a way to automatically compensate the signal propagation delay through components in a project. This can be switched on using the *Auto-compensate signal delays* option on the project properties.

When you turn on delay compensation, delay alignment devices are added in front of every output, and in front of every multi-input device that combines the inputs. The compiler then places the devices on the DSP chips. Finally, based on the placement, the compiler sets the alignment delay lengths to whatever is necessary to achieve the equal arrival time of signals. Because this is done after the devices are placed, it is possible that when these delays are set the DSP memory will be over allocated. When this happens, there are several solutions:

- Increase the *delay comp average size* setting on the project properties (this setting provides a hint to the compiler about how much memory should be reserved for compensating for delays). This will reserve more memory, making the compile more likely to succeed.
- Reduce the *Processing vector length, samples* setting on the project properties. Less memory will be used, potentially reducing the entire system delay, but more DSP resources will be consumed. If your compile results show failure due to lack of memory, but you have spare DSP cycles, this solution may work.
- Hard assign some devices to DSP chips.

Internal latencies and alignment

The table below shows latency measurements for particular operations performed by a NioNode. These figures are useful when you want to calculate delays manually when setting up alignment between objects.

IO card serial port receive	1 vector
IO card serial port transmit	1 sample
DSP processing	1 vector
DSP to DSP transfer (within a NioNode)	1 vector
XDAB transfer (NioNode to NioNode)	3 vectors
IO Card ADC	34 samples @ 48/96 KHz
IO Card DAC	37 samples @ 48 KHz, 20 samples @ 96 KHz

It is not always easy to manually calculate the latency of a particular signal relative to another. For example, in the design below, the alignment must be set to 4 vectors (32 samples):

- DSP A processing (sine)
- transfer A to B
- DSP B processing (Gain)
- transfer B to A
- DSP A processing (2x1 mixer)

The alignment figure can be calculated by counting up the vectors between the 5 stages listed, but it is easier to create your design and check the delay report to see what manual delays you need to insert.

Emulating a project without hardware

If you do not have access to MediaMatrix hardware devices, such as NIONs or nControls, or do not want to disturb a currently running system, the target hardware can be emulated on the NWare PC. This makes it possible to gesture controls, save and load presets, test the control programming and verify external control, but some of the functionality of a deployed project is not available.

Notes:

- The emulator does not support audio processing.
- A maximum of 32 hardware devices can be emulated.
- Each time you emulate your project, Windows Task Manager will show a program called **pion_emulator.exe** on the **Processes** tab. Sometimes, when the Emulate and Disemulate buttons are clicked several times in succession, multiple instances of **pion_emulator.exe** are shown before each one is closed.

To emulate a project without hardware

- On the toolbar, click the **Emulate** button .

  The project will be compiled and a compilation report will be produced. For more information, see *Understanding the compilation report* (on page 164).

  The pointer will change to a ✋, NWare will switch to gesture mode, and you will be able to gesture controls.

To stop emulating

- On the toolbar, click the **Disemulate** button .

Connecting to the current project

When the current project has been deployed to a node, such as an nControl node, for example, NWare or a third party control system can connect directly to it for control and monitoring.

Note: In order to connect to the project, you must have the exact same version of the project loaded in NWare.

To connect to a project

- On the toolbar, click the **Connect** button .

Specifying a preferred user for connecting to projects

You can specify a default username to use when connecting to projects after selecting the emulate, deploy or upload and connect features. This is called the *preferred user*.

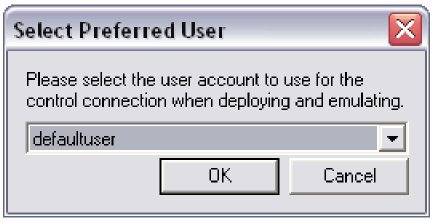
NWare will also use this username when it connects to nodes to perform other actions, such as updating firmware. NWare logs on to the node using a particular username. The node matches this username with the username of the same name stored on the unit. The privileges of the username on the node determine whether the action may be carried out.

Tip: Users often log on to nodes with the username *defaultuser*. If the settings for this username on the node show the **Deploy** privilege set to **Disallow**, it will not be possible to deploy a role to the node even when they are logged on.

⏩ *To specify a preferred user for connecting to projects*

1. In NWare, on the **File** menu, click **Preferred User**.

   The **Select Preferred User** dialog box is displayed.



Please select the user account to use for the control connection when deploying and emulating.

defaultuser

2. In the list, click a user account to specify as the default user.
3. Click **OK**.

Disconnecting from a project

- On the toolbar, click the **Disconnect** button.

Uploading a project to NWare and connecting to it

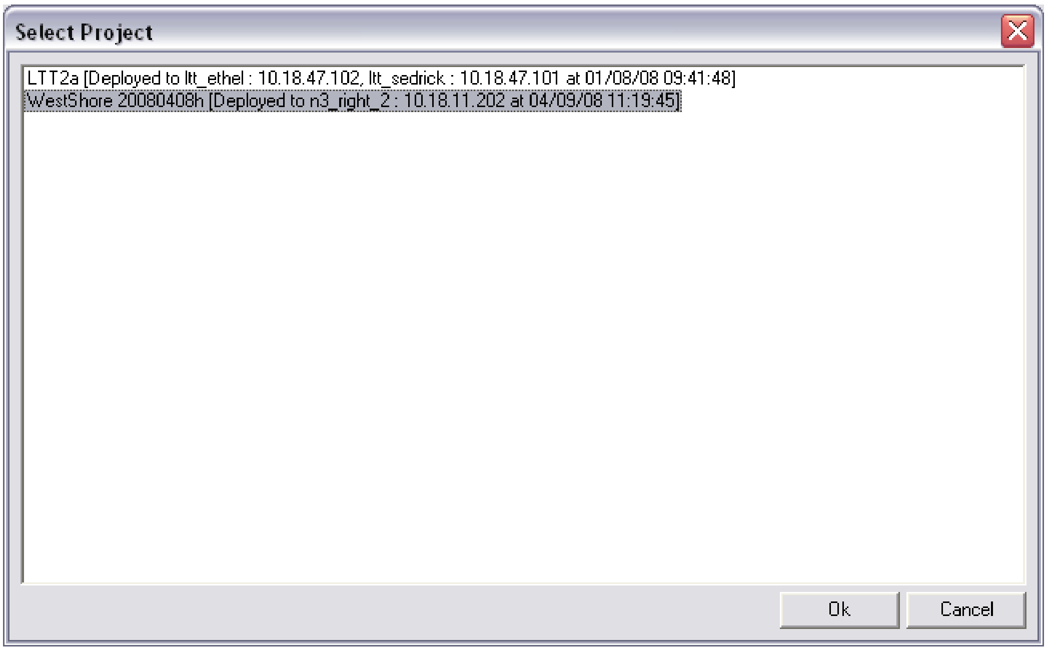
You can upload a project file (.npa) from a node for editing on the NWare PC, and connect to it for control purposes, all in one operation.

Note: In order for this operation to work, you must have already deployed the project. The **Save Project to Target** check box must have been checked on the Role Deployment dialog box, which will download and store the complete project on the node.

⏩ *To upload a project to NWare and connect to it*

1. On the **File** menu, click **Upload and Connect**.

The **Select Project** dialog box is displayed.



Select Project

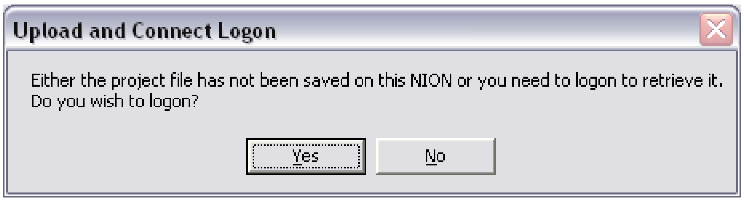
LTT2a [Deployed to ltt_ethel : 10.18.47.102, ltt_sedrick : 10.18.47.101 at 01/08/08 09:41:48]
WestShore 20080408h [Deployed to n3_right_2 : 10.18.11.202 at 04/09/08 11:19:45]

2. Click the project name, and then click **OK**.

   The **Save As** dialog box is displayed.

3. Specify a location and file name for the project file, and then click **Save**.

   The **Upload and Connect Logon** dialog box is displayed.

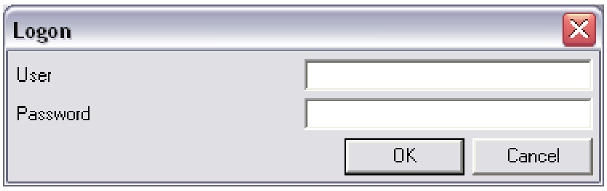


Upload and Connect Logon

Either the project file has not been saved on this NION or you need to logon to retrieve it.
Do you wish to logon?

4. Click **Yes**.

   The **Logon** dialog box is displayed.



Logon

5. Type the username and password, and then click **OK**.

   The pointer will change to a ✋, NWare will switch to gesture mode, and you will be able to gesture controls.

Specifying values for device controls

When you have connected to a project using NWare or Kiosk, you can gesture controls and specify values in edit boxes. When you move the cursor over a control that accepts a value, a red border will be displayed.

⏩ *Specifying individual values for controls*

1. Move the mouse pointer over the control.

A red highlighting box will be displayed.

2. Type the value and press Enter.

⏩ *Specifying values for multiple controls at the same time*

1. Create a spreadsheet and type values into the cells that you want to copy over to the NWare controls.
2. Select the cell values and then press CTRL+C.
3. In NWare, select the controls that are to receive the values.

   You can press CTRL+A to select all the controls on a device.
4. Press CTRL+V to paste in the values.

Troubleshooting

Problem	**Meaning**
Deployment fails. Error: Exception Assigning Role: Timeout During Connect. Completed = No	NWare could not establish a reliable network connection to the remote node. A robust connection is required for deployment. **Notes:** ▪ This problem can occur even if the target node is listed on the Remote Log tab and you can PING the node from a Command Prompt window. ▪ Using a crossover cable between the NWare PC and the remote node can cause this problem. We recommend that you connect the NWare PC and the node to a switch. ▪ Using a wireless connection can also cause this problem. Switch to using network cables where possible.
Deployment fails. Role start failed: adapter not initialized	You are attempting to deploy a project containing a CAB to an nControl node or nTouch 180 node, but no network interface has been specified for connecting to the CAB. Connect to the nControl or nTouch 180 via the web interface. Open the *Network* page. Specify a network interface in the *CAB Control Interface* list, then reboot the unit. For more information, see *Interfacing with a CAB device* in the *nControl Hardware Manual or Interfacing with a CAB device in the nTouch 180 Hardware Manual.*
The Remote Log tab does not list the node I want to deploy to.	Pandad, the program which locates nodes on the network, may have crashed. For more information, see *Determining if Pandad has stopped functioning* in the *Pandad Administrator Guide*.

Chapter 7

Adding more advanced features to your design

In This Chapter

Control devices

Control Programming

Control Programming refers to the set of features that allow the designer to create and customize interactions between device controls. Once the project is deployed, the controls are hosted on a NioNode, nControl node or nTouch 180 node, which means third party systems can interact with them.

Control Settings

A Controls setting is represented in three ways: by position, string and value. Position is the fractional number corresponding to the graphic knob or slider position in the user interface, and ranges between 0 and 1. String is the text display string for the Control in the user interface. Value is the 32-bit internal number representation of the Control, and while not normally portrayed in the user interface, can be of use in Control Programming. As an example, a gain settings position might be 0.55, while its string is -45dB and its value is -45.0.

Control Wiring and Groups

Controls can be graphically wired to one another to form Control Groups. A Controls wiring properties are accessible in its Control Properties dialog in the Wiring tab. In the Control Wiring dialog box, you can independently enable the display of the three available Control Wiring nodes for that Control: slave, peer and master. These nodes display, left to right, underneath the Control.

Here are the types of wired group relationships:

- Master-to-slave - a Control master node is wired to one or more Control slave nodes. The slave Control(s) will track the master, and become read-only, meaning that it (they) cannot be directly gestured.
- Peer-to-peer - a Control peer node is wired to one or more other Control peer nodes. In this case, any of the Controls can be gestured, and they will track each other.
- Slave-to-slave - wiring Control slave nodes to one another can be done, but does not accomplish anything unless a master is also wired into the set.

Controls wired directly to one another are linked by position, that is, the graphical position, between all the way down and all the way up, of the linked Controls will track.

Note: If you make a copy of a control - a fader, for example, and then set up the copy as a slave control, you will not be able to gesture the original control or the copy. It may not be obvious that a particular control has been set up as a slave, as the copy may be on a different page in the design and have a different appearance. You can use the Find command on the Edit menu to locate all the copies of a particular control to see if any are used in a slave configuration. The Find Results tab at the bottom of the screen will display clickable links to the original control and all its copies. You can then click each link and check to see if any of the copies are set up as slaves.

Tip: You can create a peer-to-peer group of two or more controls by selecting the controls and then clicking Control Group on the Tools menu.

Control Processing

A more powerful approach to creating interactions between controls is to use Control Processing. In this case, Controls are wired to Control Processor devices - a Controls master node is wired to the input node of a Control Processor, and one or more Control slave nodes are wired to the output node of a Control Processor. Control Processing supports the linking and processing of Control settings by position, string and value. (Recall that direct Control-to-Control wiring only supports position-based linking.)

There are two types of Control Processor devices: Control Operator and Control Scripter.

Using XDAB to share DSP resources

Clusters of NIONs can be interconnected in an XDAB ring, allowing up to 512 channels of redundant, low-latency audio to be exchanged. Wiring devices together in this way allows available DSP processing capacity to be shared between the group.

Note: The maximum channel count scales proportionally to the chosen sample rate. At a 48KHz sampling rate, 448 channels of audio can be exchanged.

Each NION has two XDAB ports, marked IN and OUT, which are used to physically wire them together in a ring using shielded CAT-6 cabling. The order of the NIONs in the ring is not important, since it is determined, and adjusted for, when the associated NWare project is deployed. However, NIONs that are specified as members of the same XDAB ring must be physically connected, otherwise each unit will indicate that a fault has occurred.

When the project is emulated or deployed, devices are not automatically distributed between the available DSPs in all the connected NioNodes. This has two important implications:

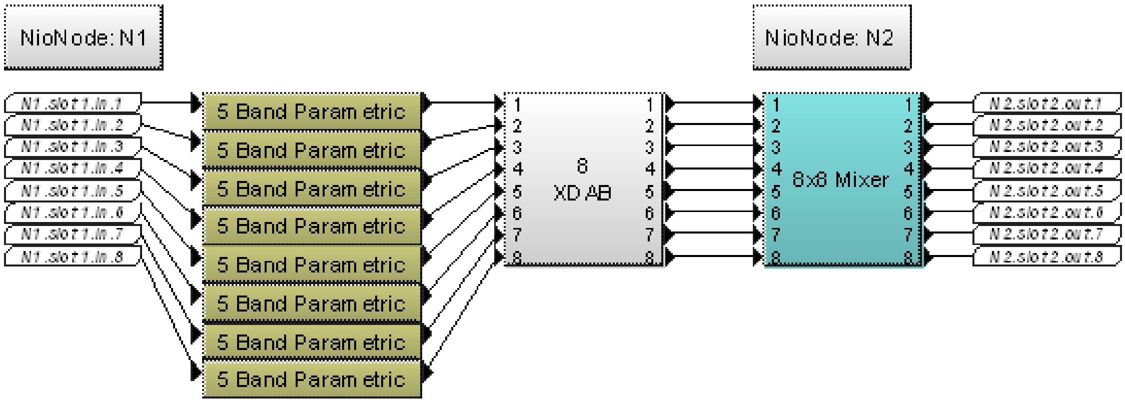
- Since a project can contain multiple XDAB clusters, as well as isolated NioNodes that are not in a cluster (perhaps connected via CobraNet), the designer must indicate to the compiler which NioNodes in the project are connected together via XDAB by adding an XDAB device to the project and wiring it to devices.
- The designer must specify in which role each of the DSP devices resides.

Note: The compiler does not automatically allocate DSP devices in the project to DSP chips within each NioNode.

Example 1

In this example, the compiler can automatically determine which NioNodes will host the devices – it is inferred by the arrangement in the design.

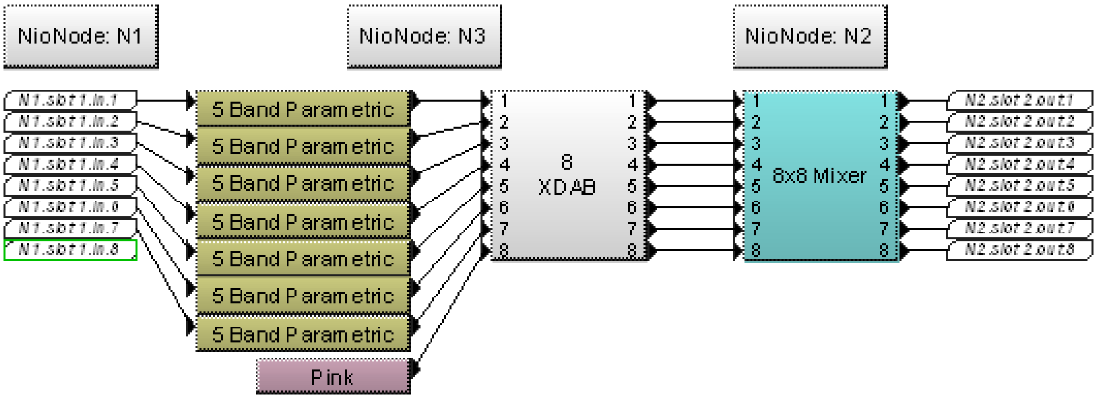
The input flyoffs from NioNode N1 are wired to the parametric equalizers, which are in turn wired to the XDAB device inputs. The compiler will therefore place the parametric equalizers into the N1 role. The output flyoffs from NioNode N2 are wired to the 8x8 mixer, so the mixer will be placed into the N2 role for the second NioNode.



Example 2

In this example, the Pink Noise Generator is not linked to a NioNode flyoff – only the XDAB block. Assuming the device has not been assigned to a role manually by the user (via the Device Placement option on the Tools menu), the compiler will assign it to a role automatically. When this operation is performed, the compiler picks a role at random and a warning message is displayed.

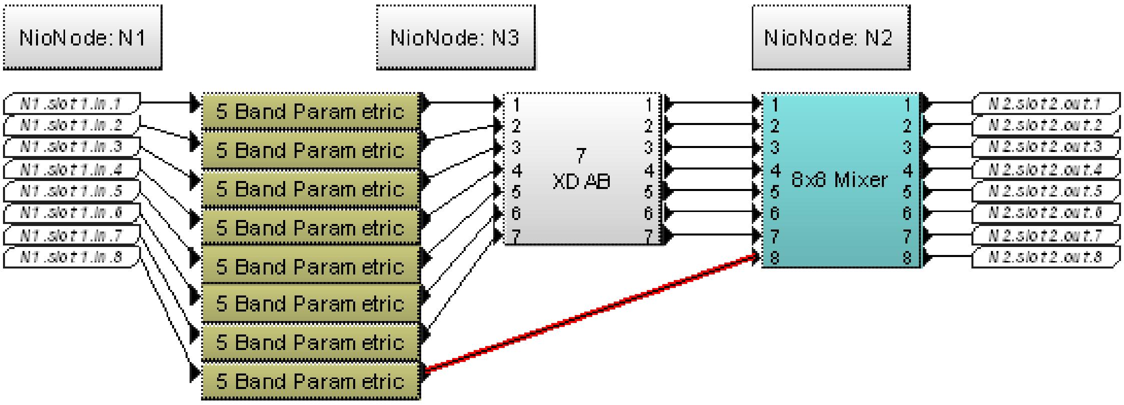
Note: The random assignment is made, even if the NioNode hosting the role runs out of DSP resources when the role is deployed.



Example 3

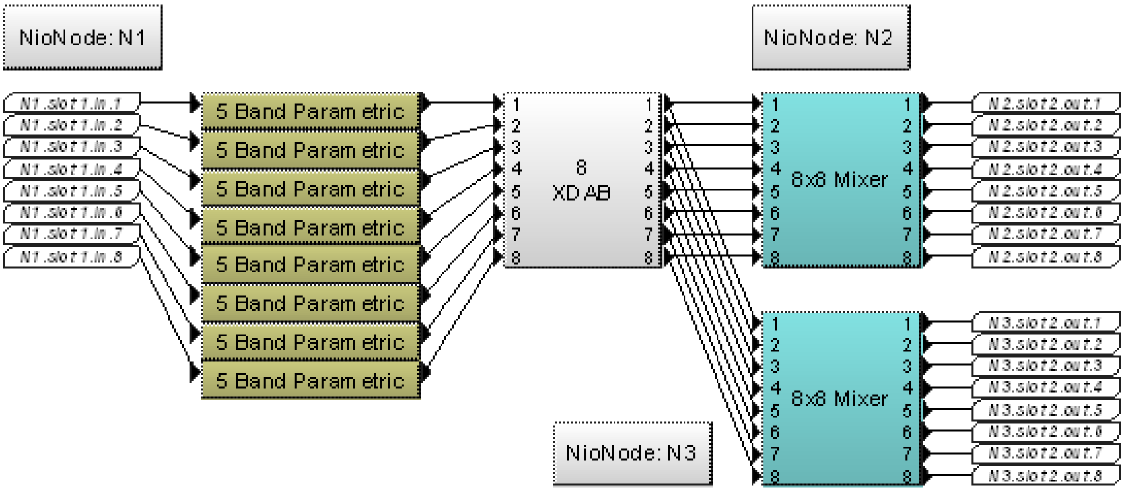
This example demonstrates how incorrectly wiring the devices can cause the compile to fail.

There is a signal path that runs from N1.slot1.In.8, to the parametric equalizer, to the 8x8 mixer and then out on N2.slot2.out8. This path does not use the XDAB device and means that the compiler will not automatically place the devices into the available roles and the compile will fail. It is important that the XDAB device is used as an intermediate step between the devices.



Example 4

This example demonstrates that signals that are transferred on to the XDAB ring are available to all NioNodes in the ring. The inputs from NioNode N1 are made available to both the N2 and N3 nodes.



XDAB considerations

- The source of a particular XDAB channel cannot be dynamically changed – it is fixed at compile time.
- The compiler does not automatically place devices in different roles for all the NioNodes connected to the XDAB ring.
- The XDAB interface has a maximum capacity of 448 channels at a 48 KHz sample rate, but the actual number of channels allocated is specified by the channel count of the XDAB device.
- The level of resources consumed on the NioNodes by XDAB is proportional to the channel count. Each NioNode in a cluster receives, at each of its DSP chips, all of the signals sourced by the other NioNodes in the cluster. This consumes signal buffer memory. In addition, IDAB (DSP-to-DSP transfer bus) channels are used by XDAB channels.
- In order to transfer signals to devices via XDAB, you must wire all the input devices and output devices to a single XDAB device. Using flyoffs for the XDAB inputs and outputs will help improve the clarity of the design.

Tip: To create a flyoff from an XDAB connector, switch to wiring mode, drag one of the connectors away from the XDAB block and then right-click. You can name the flyoff and then it will be added to the Flyoffs tab for use with other devices.

XDAB wiring rules

- A project can contain multiple XDAB devices, but you cannot wire an XDAB device to another XDAB device.
- A NioNode cannot be wired to more than one XDAB device.
- An XDAB input must not be traced to more than one NioNode.
- If an XDAB input is not traced to any NioNode, then the devices in that trace are assigned to the default NioNode.
- If an XDAB output is not traced to any NioNode, then the devices in that trace are assigned to the same NioNode that is wired to the corresponding XDAB input.
- If an input is not wired, that channel is deleted from the XDAB. Any devices fed by the output of that channel are wired to silence.
- If an XDAB output (but not the corresponding XDAB input) is not wired, it is ignored. That XDAB channel is not deleted.
- If a signal is connected to more than one XDAB input, those channels are joined into a single channel.
- If an input and its corresponding output are traced to the same NioNode, the devices on either side are wired directly to one another so that they do not pass across the XDAB. The XDAB channel is not deleted.
- If an XDAB output is wired directly to an XDAB input, the channel corresponding to the input is deleted and any ports fed by the channel are silenced.
- At least two NioNodes must be traced to each XDAB.

Setting conductor and XDAB priority

Introduction

You can use NWare to change the CobraNet Conductor and XDAB master priorities. These settings will ensure that a particular device within an XDAB cluster will become the XDAB master or that a particular CobraNet device on a network or VLAN segment will become a CobraNet Conductor.

Each NION will set the actual priority from within the range specified by the user. Each of these ranges can have two values.

If you want a NION to be an XDAB master or a CobraNet Conductor, set the appropriate range to be higher than any other device in the cluster.

Notes:

- NWare will normally ensure that the default values are set for proper operation. However, it may be necessary to manually adjust the settings in order to meet the constraints outlined in the use cases.
- Be aware of the Conductor priority settings of devices outside the XDAB cluster that reside elsewhere on the network. This point is important when using slave VLANs in order to ensure that a slave device within a cluster remains the CobraNet Conductor.

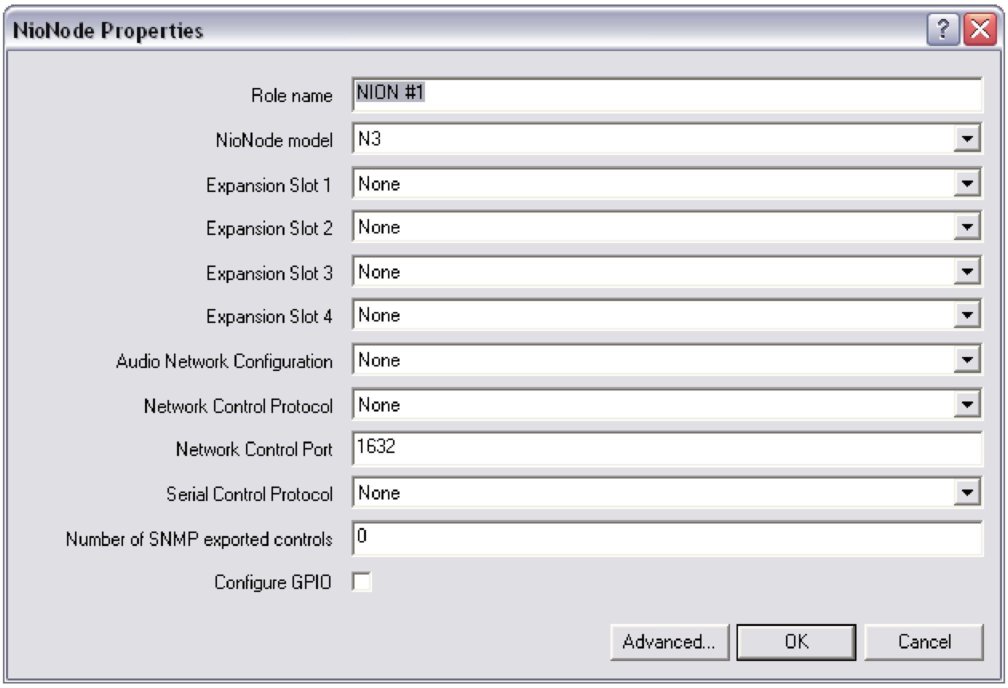
Tips:

- Explicitly set the CobraNet Conductor priority and XDAB master priority in all devices to ensure they operate correctly.
- Locate the CobraNet Conductor outside an XDAB cluster when possible.
- Set all Conductor priorities to 0 in all devices except those that should be allowed to be Conductors. For those devices that should be Conductors, make their priorities the same.
- Devices with CM-1 modules are preferred as Conductors over devices that have CM-2 modules.

Adjusting the CobraNet Conductor and XDAB clock master priorities

1. Right click on the NION on the NWare page, and then click **Device Properties**.

The **NioNode Properties** dialog box is displayed.

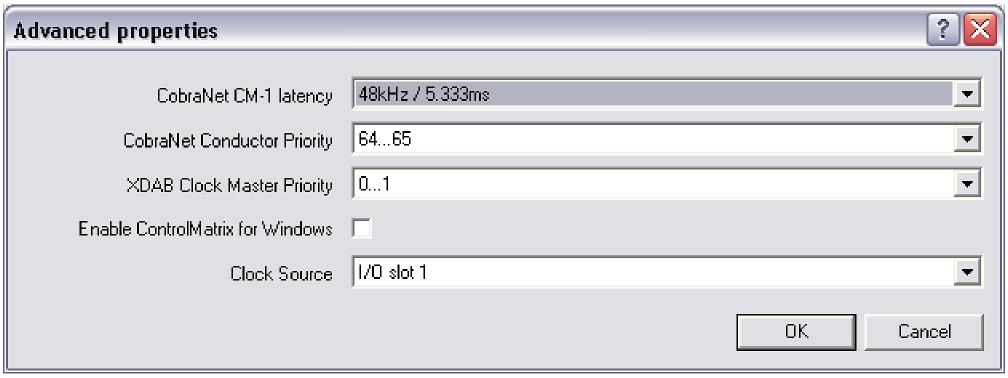


NioNode Properties

Role name	NION #1
NioNode model	N3
Expansion Slot 1	None
Expansion Slot 2	None
Expansion Slot 3	None
Expansion Slot 4	None
Audio Network Configuration	None
Network Control Protocol	None
Network Control Port	1632
Serial Control Protocol	None
Number of SNMP exported controls	0
Configure GPIO	□

2. Click **Advanced**.

   The **Advanced properties** dialog box is displayed.



Advanced properties

CobraNet CM-1 latency	48kHz / 5.333ms
CobraNet Conductor Priority	64...65
XDAB Clock Master Priority	0...1
Enable ControlMatrix for Windows	□
Clock Source	I/O slot 1

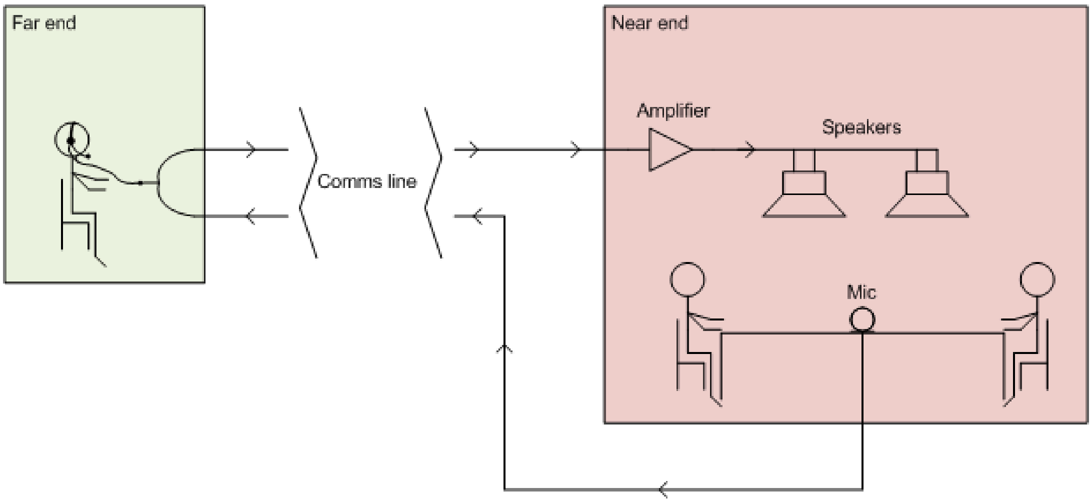
3. In the **CobraNet Conductor Priority** list, click the priority range for the NION.
4. In the **XDAB Clock Master Priority** list, click the priority range for the NION.

Tip: If you want a NION to be an XDAB master or a CobraNet Conductor, set the appropriate range to be higher than any other device in the cluster.

Using echo cancellation in your design

Introduction

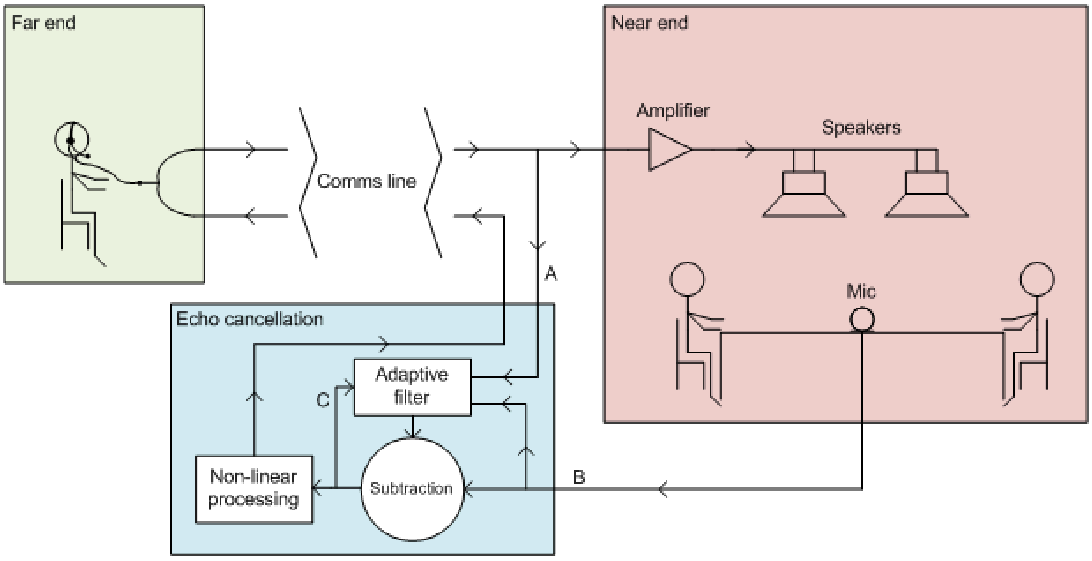
It is quite common to hear an echo when communicating via microphones and speakers, in a set up similar to the one shown below. When the person at the far end speaks, their voice is transmitted to the speakers in the room at the near end. The microphone on the table not only picks up what the people in the room say in reply, but also the sound of the person at the far end as it emerges from the speakers. This means that the person at the far end not only hears what the people at the near end say, but also a delayed echo of what they themselves have just said. The effect can be very distracting. It is important that the echo is eliminated to ensure that the quality of the sound in the conference is kept to an acceptable level.



An echo cancellation system can be introduced into the design to eliminate the echo effect. It will ensure that the person at the far end can only hear the sound made by the people at the near end in their headphones, not the sound of the people at the near end plus the echoed sound of themselves from the far end.

In order to send only the sound of the people at the near end down the line to the person at the far end, the sound of the person at the far end, as it is heard in the room, must be removed (or *subtracted*) from the signal. The problem is, the part of the signal from the near end containing the signal from the far end is no longer identical to the original and cannot be easily recognized. The characteristics of the room have affected it. The echo cancellation system must determine how the room at the near end affects the signal from the far end in order to make it recognizable. It can then be cancelled out.

The diagram below shows the same scenario with the addition of an echo cancellation system.



The echo cancellation process is as follows:

1. Sound from the far end is digitally sampled to form a reference signal (A).
2. The sound from the microphone is digitally sampled (B). This contains the direct path sound from the people at the near end, plus the reverberant sound from the speakers containing the sound from the far end.
3. The adaptive filter builds a mathematical model of how the room at the near end affects the sound from the far end. It does this by comparing the part of the signal from the near end (B) that contains the signal from the far end with the signal that comes directly from the far end (A) and working out how the sound has been changed.
4. In the subtraction phase, the system locates signal A from the far end within signal B from the near end and attempts to remove it.
5. If the sound is subtracted successfully, the adaption filter has effectively determined how the room at the near end affects signal A. It can then use the same set up to cancel echo from all further sound it receives. If the sound is not subtracted, the signal is fed back into the adaptive filter (C), so the filter can adjust the settings. The process of feeding the signal back until a successful subtraction is completed is called *convergence*.
6. Once the subtraction process has completed, the signal is passed to the non-linear processing stage. If the signal still contains some residual sound from the far end that has not been removed by the adaptive filter, it is removed.

For more detailed information on how echo cancellation works, see *http://en.wikipedia.org/wiki/Echo_cancellation* (*http://en.wikipedia.org/wiki/Echo_cancellation*).

Non-linear processing (NLP)

The non-linear processing (NLP) stage of the echo cancellation system will detect when there is audio coming from the near end. If there is no audio coming through, then the system will switch off the output to the far end.

Sometimes, there may be silence at the near end (so that the output is automatically switched off to the far end), but then soon after, the people start to speak again and output is generated. The system must cope with natural pauses in conversation without chopping off the start or end of what is said.

The MediaMatrix echo cancellation system is set up so that the NLP stage shuts off the output when the level of noise at the near end falls below a fixed threshold.

Echo span and latency

The task of the adaption filter is to determine how to turn the signal from the near end (labeled B in the diagram in the *Introduction* (on page 179)) into the reference signal from the far end (labeled A).

In the MediaMatrix system, the adaption filter has a buffer size of 140ms, which is referred to as the *echo span*. If sound is passed into the adaption filter containing an echo longer than the echo span, only the first 140ms of the echo will be cancelled; the remainder will not.

The time taken for sound entering the echo cancellation system to be processed and then finally exit is referred to as the *latency*. It is 11ms for the MediaMatrix system.

Echo return loss enhancement (ERLE)

Due to the acoustics of the room at the near end and the volume setting, the signal level entering the echo cancellation system at B in the diagram (see *Introduction* (on page 179)) is typically lower than the reference level at A. The difference in level is referred to as the *echo return loss*.

In the subtraction phase of the echo cancellation process, the echo in the signal is typically attenuated by 35dB. The attenuation level achieved is referred to as the *echo return loss enhancement*.

If, for example, the reference signal level at A is 0dB (relative level) and the echo return loss is 10dB, the signal level of the echo reaching the far end would be -45dB.

Example MediaMatrix configurations

MediaMatrix provides echo cancellation via the NioNode NIO-AEC card. Each of these cards can provide acoustic echo cancellation for up to eight channels of audio.

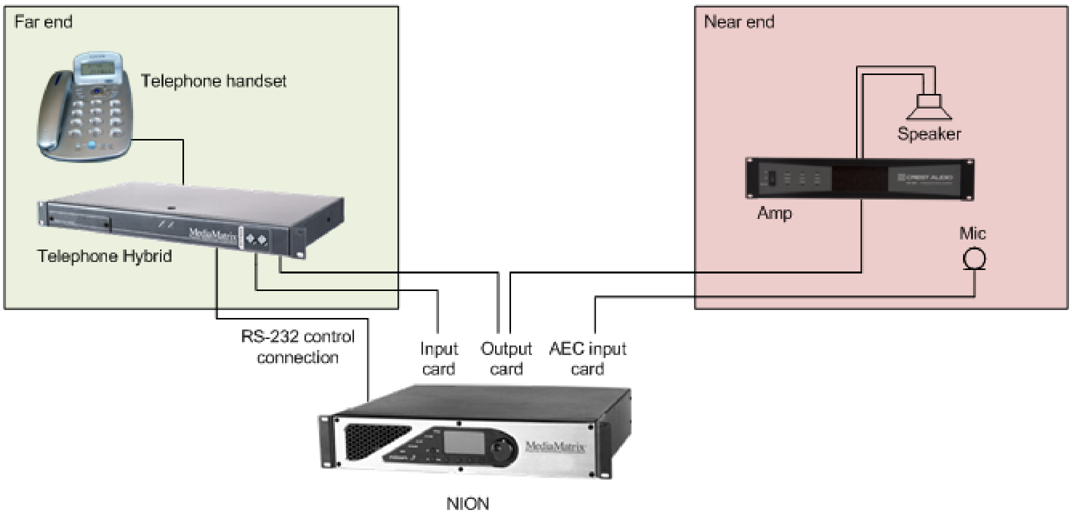
The sections below show how echo cancellation can be provided in different scenarios. In each case it takes just a couple of seconds for the system to converge and start cancelling the echo. The cancellation then works instantly every time audio is passed through.

Note: If the audio characteristics of the room where the microphones are situated changes, because a microphone is moved, for example, the system has to reconverge. That is, repeat the process of analysing the audio characteristics of the room and change the settings of the adaption filter until it successfully cancels the echo. This may take a few seconds.

Scenario 1

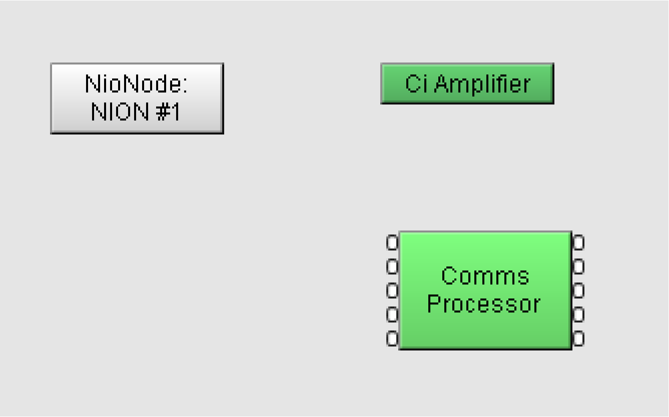
At the far end, a telephone is connected to a MediaMatrix Telephone Hybrid. This device converts the audio from the telephone microphone to line out audio, before it is fed into an input card in the NioNode. The audio is transferred from the NioNode to the near end via a regular NioNode output card, where it can be heard from the speaker.

The microphone input from the room at the near end is fed into the AEC card, where the echo is removed, and then to the output card for transfer to the telephone via the Telephone Hybrid.



NWare project for scenario 1

A NioNode, a Ci Amplifier and a Comms Processor have been added to the project.



Three cards have been specified for the NioNode expansion slots:

- a Nio-8i input card
- a Nio-8o output card
- an AEC card.

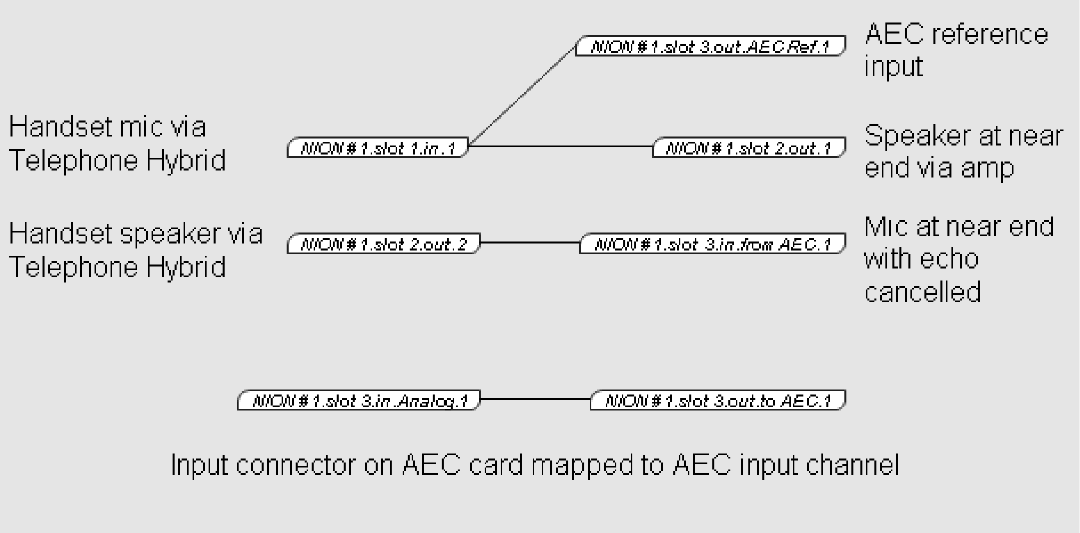
The controls for the AEC card allow the user to enable or disable echo cancellation, set gain levels, and so on.

The Comms Processor device will be used to communicate with the MediaMatrix Telephone Hybrid via the RS-232 link from the NioNode. The device will send commands to ring the phone at the far end when someone initiates a call at the near end, or inform the user at the near end that someone is calling from the far end, for example. For information on the Comms Processor device, refer to the section *Comms Processor* in the *NWare Device Reference*.

The Ci Amplifier device communicates with the amplifier that is driving the speaker at the near end. In the example we assume that this is a Crest Ci Amplifier.The Comms Processor device will be used to communicate with the MediaMatrix Telephone Hybrid via the RS-232 link from the NioNode. The device will send commands to ring the phone at the far end when someone initiates a call at the near end, or inform the user at the near end that someone is calling from the far end, for example. For information on the Comms Processor device, refer to the section *Comms Processor* in the *NWare Device Reference*.

The Ci Amplifier device communicates with the amplifier that is driving the speaker at the near end. In the example we assume that this is a Crest Ci Amplifier.

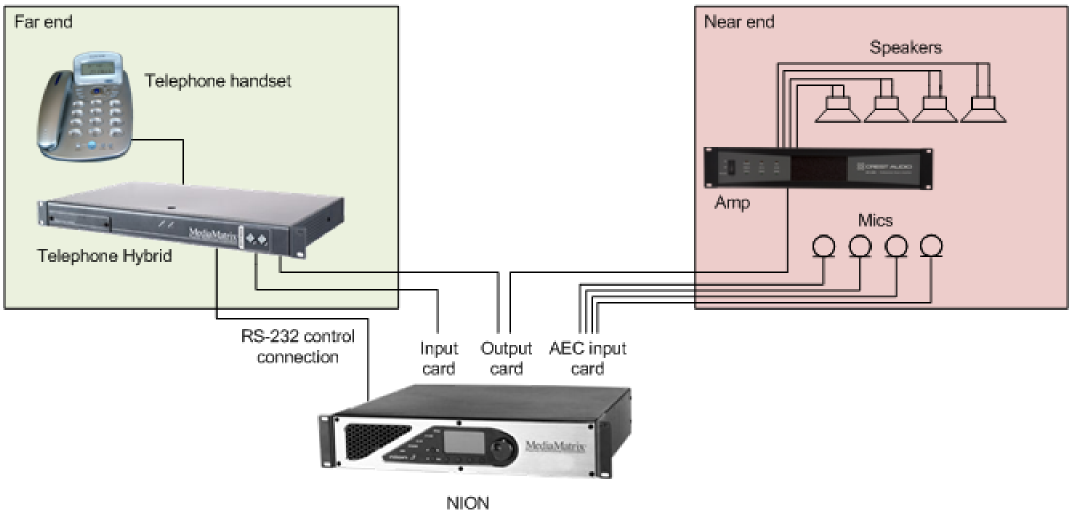
Flyoffs for the NioNode cards have been dragged over to the design and connected as shown below.


NION#1.slot 3.out.AEC Ref.1
AEC reference input
Handset mic via Telephone Hybrid
NION#1.slot 1.in.1
NION#1.slot 2.out.1
Speaker at near end via amp
Handset speaker via Telephone Hybrid
NION#1.slot 2.out.2
NION#1.slot 3.in.from AEC.1
Mic at near end with echo cancelled
NION#1.slot 3.in.Analog.1
NION#1.slot 3.out.to AEC.1
Input connector on AEC card mapped to AEC input channel

As you can see, the audio input from the mic at the far end is used as a reference signal for the echo cancellation system, as well as being transferred directly to the speaker at the near end.

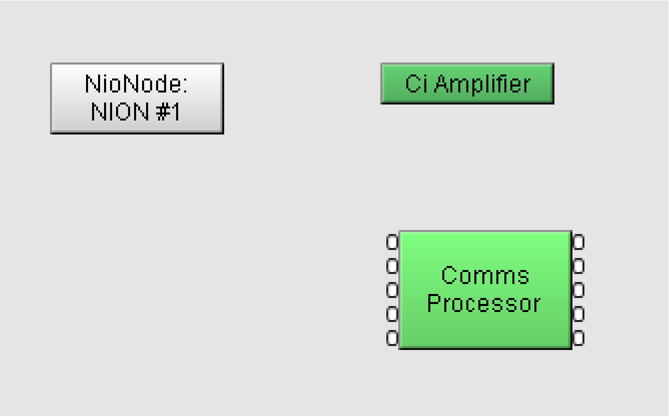
Scenario 2

This example shows how further microphones and speakers can be added at the near end to provide a system that could be used for a large conference room. The AEC card can accept up to eight microphone inputs.

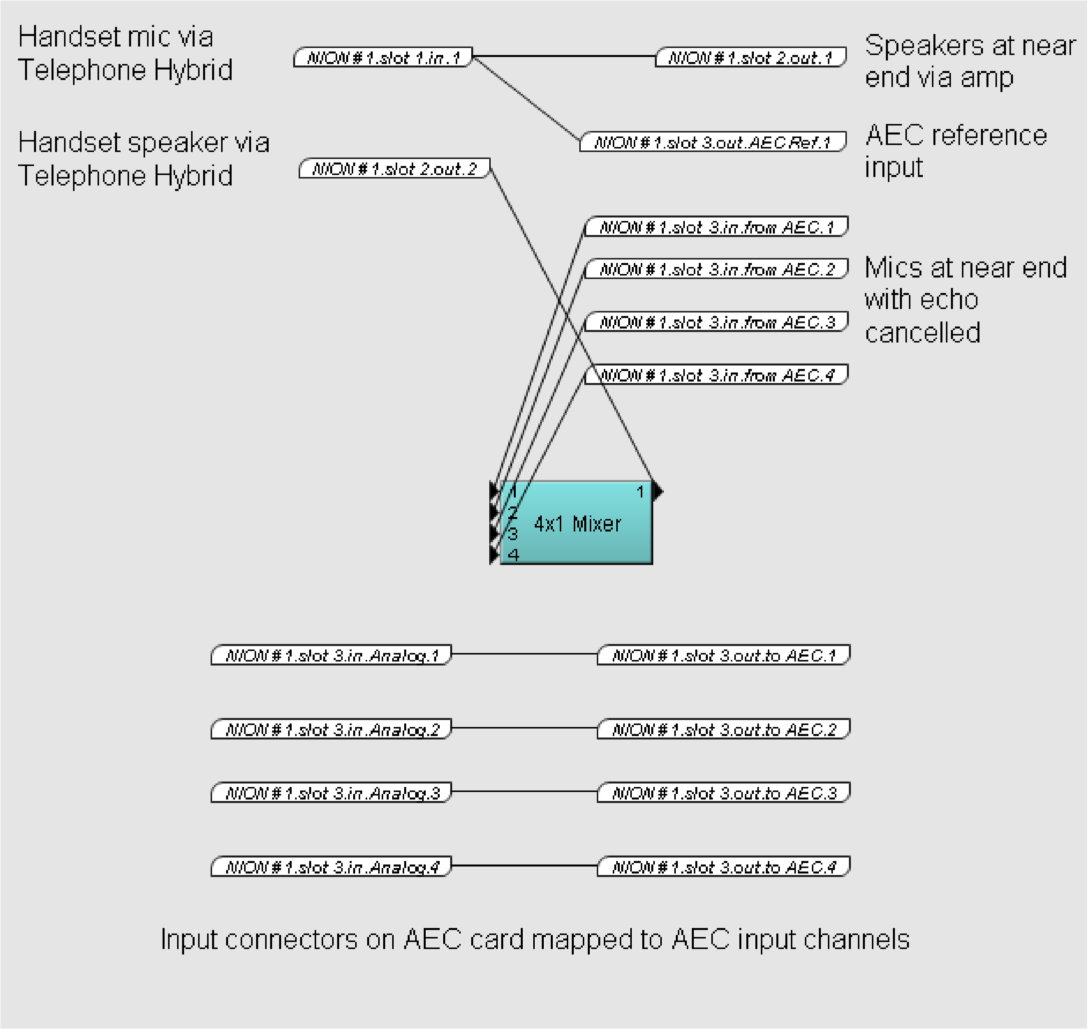


NWare project for scenario 2

In scenario 2, the same basic components are used in the project and the same cards are installed in the NioNode.

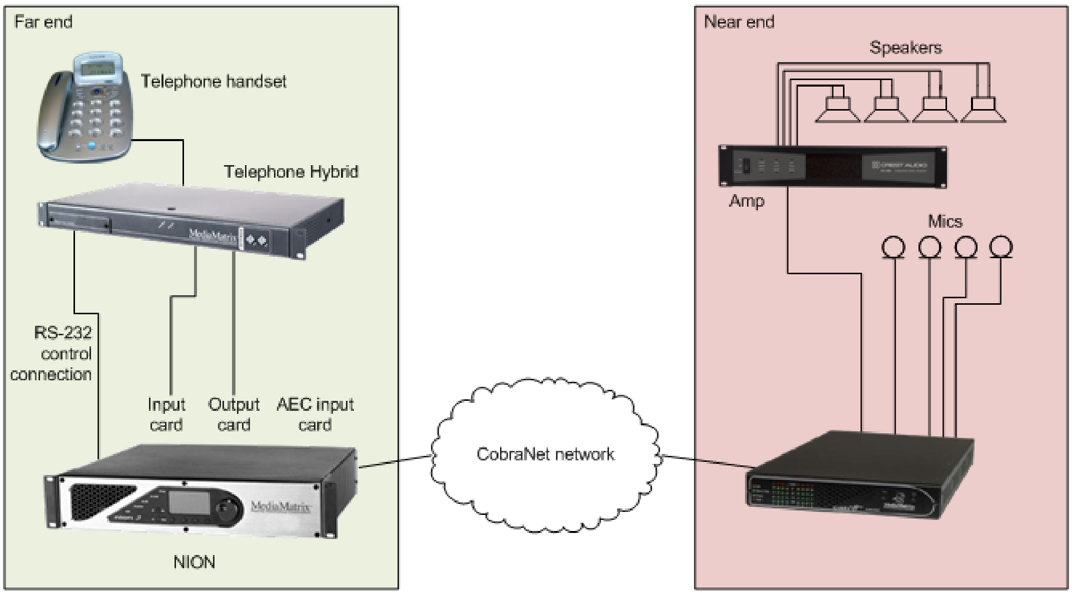


The flyoff configuration has been extended to include the other microphone inputs. These are passed to the handset speaker via a mixer device.


Handset mic via
Telephone Hybrid
NION # 1.slot 1.in.1
NION # 1.slot 2.out.1
Speakers at near
end via amp
NION # 1.slot 3.out.AEC Ref.1
AEC reference
input
Handset speaker via
Telephone Hybrid
NION # 1.slot 2.out.2
NION # 1.slot 3.in.from AEC.1
NION # 1.slot 3.in.from AEC.2
NION # 1.slot 3.in.from AEC.3
NION # 1.slot 3.in.from AEC.4
Mics at near end
with echo
cancelled
1
2
3
4
4x1 Mixer
1
NION # 1.slot 3.in.Analog.1
NION # 1.slot 3.out.to AEC.1
NION # 1.slot 3.in.Analog.2
NION # 1.slot 3.out.to AEC.2
NION # 1.slot 3.in.Analog.3
NION # 1.slot 3.out.to AEC.3
NION # 1.slot 3.in.Analog.4
NION # 1.slot 3.out.to AEC.4
Input connectors on AEC card mapped to AEC input channels

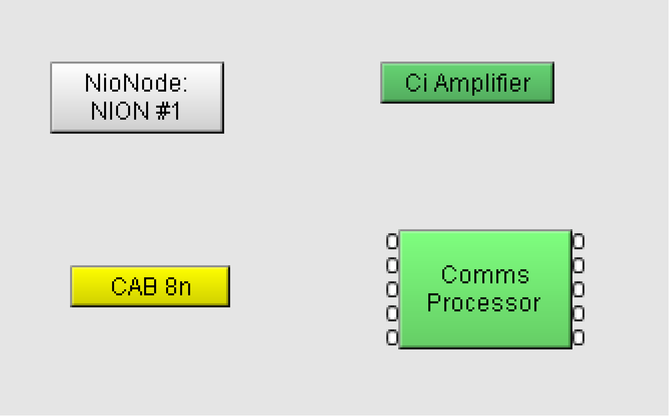
Scenario 3

This example uses a CAB 8n at the near end instead of a NioNode. The AEC card in the NioNode at the far end does not require direct connections to the microphones. Instead the audio is routed across the network to the AEC card from the inputs on the CAB 8n.

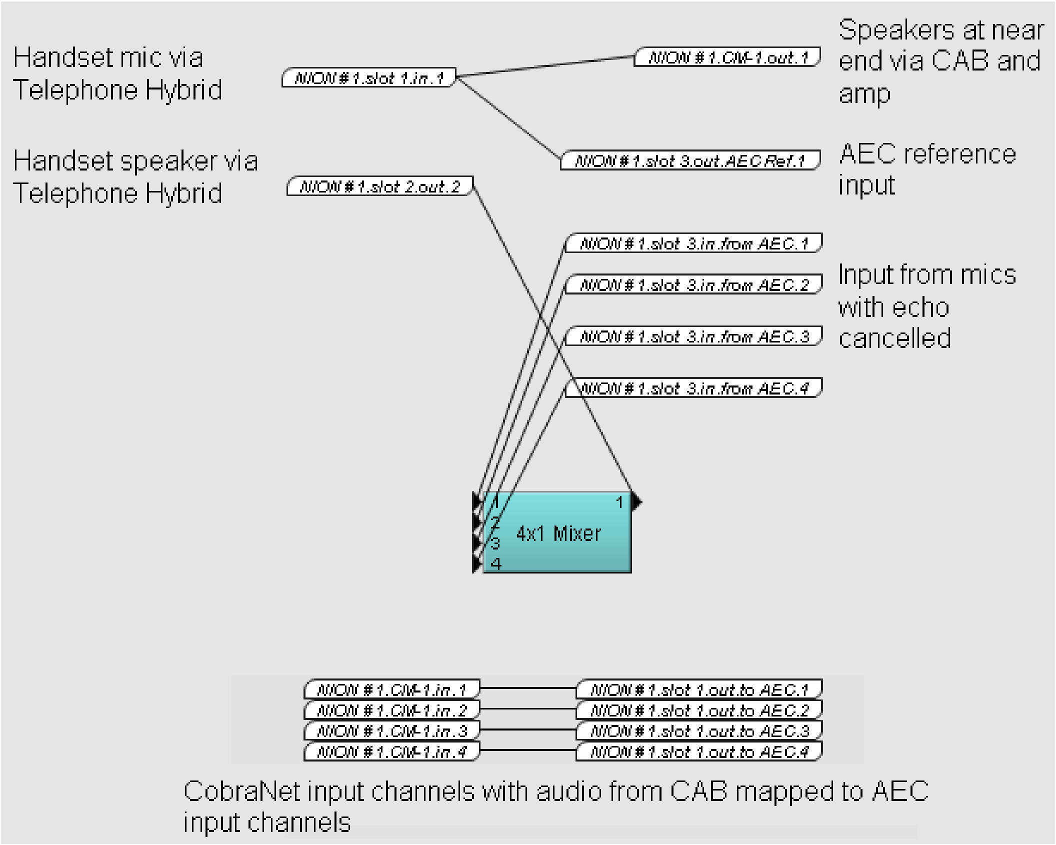


NWare project for scenario 3

In this scenario, a CAB 8n has been added at the near end. This device provides inputs for the microphones and outputs for the speakers, and is controlled by the NioNode from across the network.

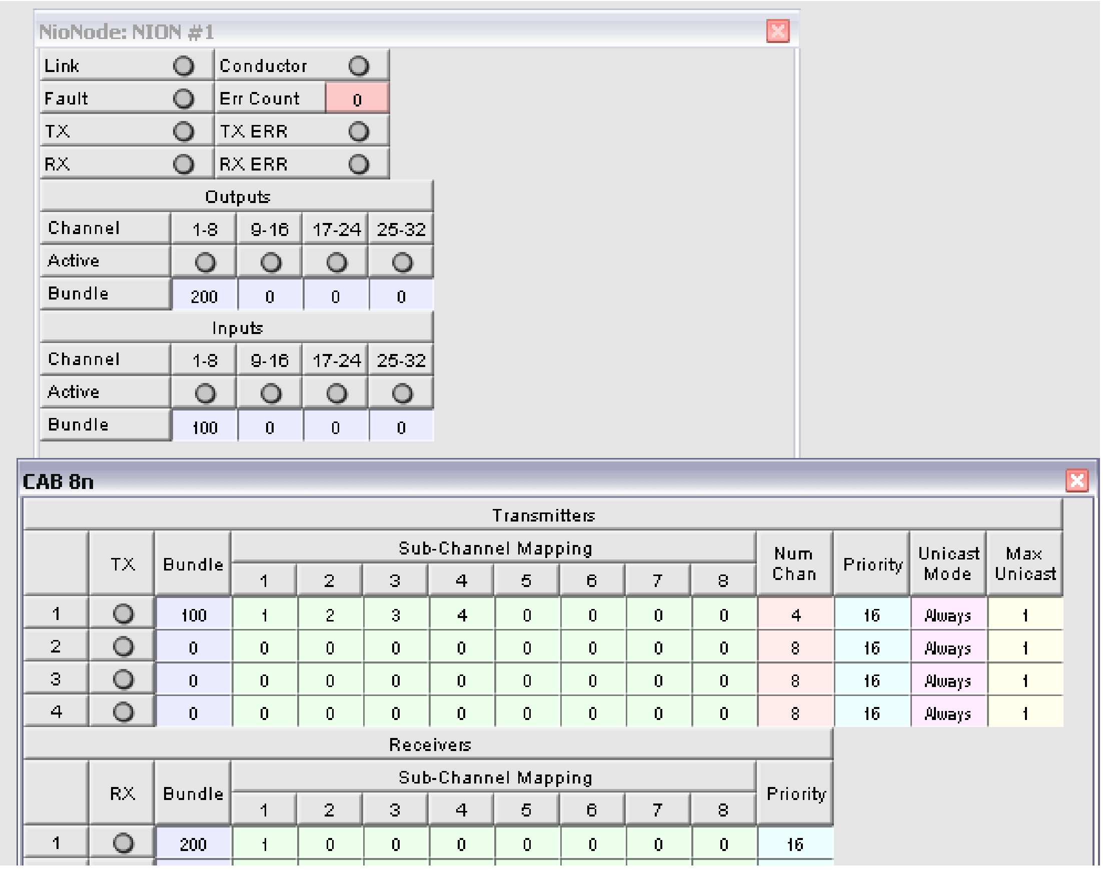


The flyoff configuration for the previous scenario has been altered so that the input from the microphone at the far end is sent to the speakers at the near end via the CM-1 interface of the NioNode (and the CAB 8n).


Handset mic via Telephone Hybrid
NION #1.slot 1.in.1
NION #1.CM-1.out.1
Speakers at near end via CAB and amp
NION #1.slot 3.out.AEC Ref.1
AEC reference input
Handset speaker via Telephone Hybrid
NION #1.slot 2.out.2
NION #1.slot 3.in.from AEC.1
NION #1.slot 3.in.from AEC.2
NION #1.slot 3.in.from AEC.3
NION #1.slot 3.in.from AEC.4
Input from mics with echo cancelled
1
2
3
4
4x1 Mixer
1
NION #1.CM-1.in.1
NION #1.CM-1.in.2
NION #1.CM-1.in.3
NION #1.CM-1.in.4
NION #1.slot 1.out.to AEC.1
NION #1.slot 1.out.to AEC.2
NION #1.slot 1.out.to AEC.3
NION #1.slot 1.out.to AEC.4
CobraNet input channels with audio from CAB mapped to AEC input channels

We are using flyoffs to connect the four microphone inputs from the CAB 8n to the inputs of the internal AEC system. The audio arrives into the NioNode via the CM-1 interface, so CM-1 flyoffs are used in the wiring.

In order for audio to be sent between the NioNode and the CAB 8n, bundle numbers have been specified on the CM-1 tab of the NioNode block and the same numbers are used on the CAB 8n tab of the CAB 8n block.



NioNode: NION #1

Link	○	Conductor	○
Fault	○	Err Count	0
TX	○	TX ERR	○
RX	○	RX ERR	○

Outputs				
Channel	1-8	9-16	17-24	25-32
Active	○	○	○	○
Bundle	200	0	0	0

Inputs				
Channel	1-8	9-16	17-24	25-32
Active	○	○	○	○
Bundle	100	0	0	0

CAB 8n

Transmitters

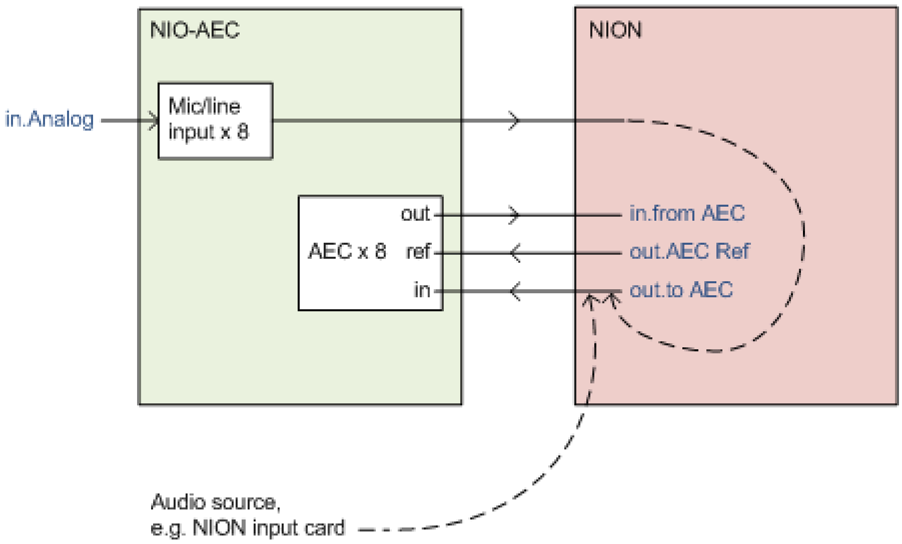
	TX	Bundle	Sub-Channel Mapping								Num Chan	Priority	Unicast Mode	Max Unicast
			1	2	3	4	5	6	7	8				
1	○	100	1	2	3	4	0	0	0	0	4	16	Always	1
2	○	0	0	0	0	0	0	0	0	0	8	16	Always	1
3	○	0	0	0	0	0	0	0	0	0	8	16	Always	1
4	○	0	0	0	0	0	0	0	0	0	8	16	Always	1

Receivers

	RX	Bundle	Sub-Channel Mapping								Priority
			1	2	3	4	5	6	7	8	
1	○	200	1	0	0	0	0	0	0	0	16

Understanding the flyoffs for the NIO-AEC card

When you add a NioNode to your design containing a NIO-AEC card, a number of flyoffs are added to the Flyoffs tab. The blue labels in the diagram below are the flyoff names as they are shown in NWare. Some of the flyoffs relate to the physical connectors on the card, but others relate to the internal routing system used within the card and the NioNode in which it is installed. You will need to be familiar with the flyoffs in order to use the AEC card in your design.



in.Analog	Physical input connector on the AEC card. A microphone is usually attached to this connector. This connector can be used as an input to the AEC system by creating a connection using flyoffs.
in.from AEC	Audio (with the echo removed) passed from the AEC card. This is to be sent to the far end.
out.AEC Ref	The reference signal passed to the card from the far end (labeled A in the diagram in the *Introduction* (on page 179)).
out.to AEC	The signal from the near end (containing the echo) passed to the AEC card (labeled B in the diagram in the *Introduction* (on page 179)). This signal can be sourced from the input connectors on the AEC card or another device by setting up a connecting using flyoffs.

Storing and reusing devices using plugins

Introduction

Plugins store collections of devices from your designs. By creating a series of plugins you can reuse and share custom functionality with other users.

Here are some things you can do with plugins:

- Customize devices to always appear the way you want, with the default values you want.
- Build, reuse, and distribute your own custom design elements and title blocks.
- Create devices to control other products.
- Distribute, either freely or for money, the devices you create.

Plugins can store devices that perform signal processing and control tasks, and can also include images and Python scripts created using the Live Python device.

Notes:

- Plugins cannot include preset datasets, audio .wav files or Python scripts that are not part of Live Python devices.
- In order to create a plugin, Edit mode must be selected. You can switch to this mode by clicking the Edit button on the toolbar.

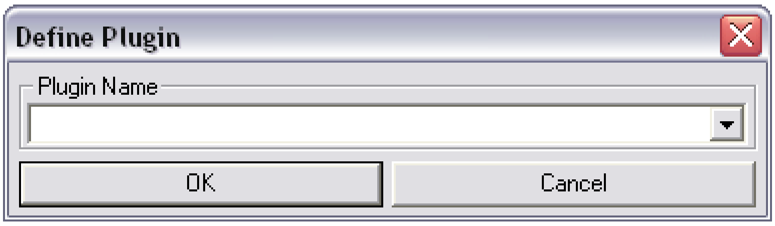
User devices

Earlier versions of NWare used to include a feature called *user devices*, which offered similar functionality to plugins. When you upgrade to a version of NWare that supports plugins, and you run NWare for the first time, you will be prompted to convert the user devices to plugins.

Creating a plugin

1. On the design page, select the devices you want to include in the plugin.
2. Expand the **Plugins** tree to show the folder where you would like to store the plugin.
3. Right-click the **Plugins** tree, and then click **Define Plugin**.

   The **Define Plugin** dialog box is displayed.



4. In the **Plugin Name** box, type a name for the new plugin.

   If you want to store the plugin in a subfolder below the top level Plugins folder in the device tree, use the following format for the plugin name:

   ```
   <subfolder name>/.../<plugin name>
   ```

5. Click **OK**.

The plugin will be added to the device tree under the Plugins folder. If you have specified one or more subfolders as part of the name, the plugin will be located at the bottom of the subfolder tree.

Note: Plugins you have created will be available in the device tree the next time you open a new instance of NWare. However, if you already have other instances of NWare open when you define a plugin, the new plugin will not be shown in those instances until you restart them.

Adding devices from a plugin to your design

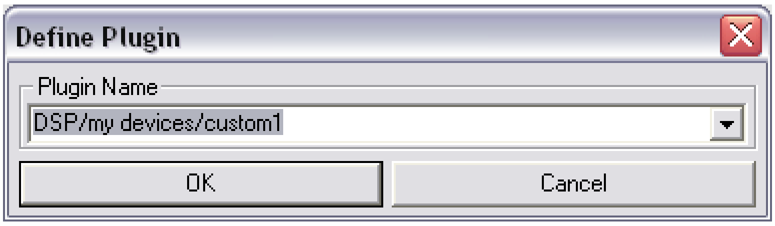
1. Expand the **Plugins** folder in the device tree.
2. Navigate to the plugin you want to add to the design.
3. Drag the plugin over to the design page.

   The devices contained in the plugin will be added to the design.

Modifying a plugin

1. On the design page, select the devices you want to include in the plugin.
2. Expand the **Plugins** tree to show the plugin you would like to modify.
3. Right-click the Plugin, and then click **Define Plugin**.

   The **Define Plugin** dialog box is displayed.



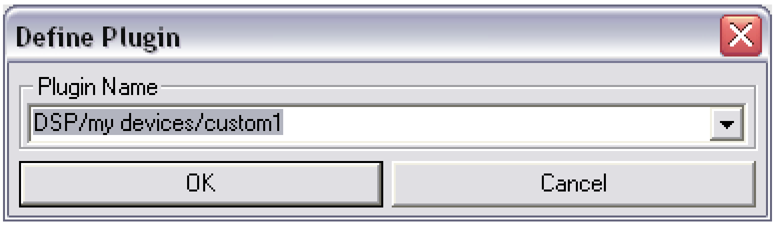
Plugin Name: DSP/my devices/custom1

4. Click **OK**.

Renaming or relocating a plugin

1. Expand the **Plugins** tree to show the plugin you would like to modify.
2. Right-click the Plugin, and then click **Define Plugin**.

   The **Define Plugin** dialog box is displayed.



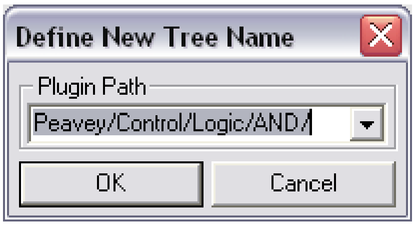
Plugin Name: DSP/my devices/custom1

3. Type the new location path and/or name in the **Plugin Name** box.
4. Click **OK**.

Renaming or relocating a folder in the Plugins tree

1. Expand the **Plugins** tree to show the folder you would like to modify.
2. Right-click the folder, and then click **Rename Tree**.

The **Define New Tree Name** dialog box is displayed.



Peavey/Control/Logic/AND/

3. Type the new path name in the **Plugin Path** box.

   To relocate a folder to another part of the tree, specify the path to the new parent folder, followed by the name of the folder you are moving.
4. Click **OK**.

Deleting plugins

⏩ *To delete a single plugin*

1. Expand the **Plugins** folder in the device tree.
2. Navigate to the plugin you want to delete.
3. Right-click the plugin, and then click **Delete**.
4. Click **Yes**.

Note: If no other plugins are located in the folder, the folder will also be deleted. If any folders in the tree above the folder are empty, they will also be deleted.

⏩ *To delete multiple plugins*

1. Expand the **Plugins** folder in the device tree.
2. Navigate to the section of the tree you want to delete.
3. Right-click the folder, and then click **Delete Tree**.
4. Click **Yes**.

Note: All plugins in the folder and subfolders will be deleted along with the folders. If any folders in the tree above the folder are empty, they will also be deleted.

Importing and Exporting plugins

When plugins are exported from NWare, they are stored in a file with the extension *.npp*. You can email this file to other users of NWare or copy it to a shared area to allow users to access your devices.

During the export process, images and Live Python code used by the plugins are automatically included in the plugin file. When you import the plugin file, the plugins it contains are automatically copied to the Plugins part of the device tree; images stored in the file are copied to the Media tab and given a prefix of *PL_* to help identify them.

⏩ *To export a single plugin*

1. Expand the **Plugins** folder in the device tree.
2. Navigate to the plugin you want to export.
3. Right-click the plugin and then click **Export**.
4. Type a file name for the NION Project Plugin file (.npp), and then click **Save**.

⏩ *To export multiple plugins*

1. Expand the **Plugins** folder in the device tree.
2. Navigate to the section of the tree you want to export.
3. Right-click the folder and then click **Export Tree**.
4. Type a file name for the NION Project Plugin file (.npp), and then click **Save**.

⏩ *To import plugins*

1. On the **Tools** menu, click **Import Plugins**.
2. Select a NION Project Plugin file (.npp) from the **Open** dialog box, and then click **Open**.

   The plugins contained in the file are loaded into NWare. If the plugins were in subfolders when they were exported, the folder structure will be recreated.

Tip: If you attempt to import a plugin file containing a plugin with the same name as one listed in the Plugins tree, NWare will compare the contents of the imported plugin with the one in the Plugins tree. If they are different, a warning will be displayed and you will be able to choose to either overwrite the existing plugin or cancel the import.

Using the Feedback Ferret to eliminate feedback in your design

The Feedback Ferret is an automatic feedback eliminator. Once it has been configured, it automatically detects and controls feedback.

The NWare Feedback Ferret device works in the same way as the hardware version, but uses the DSP resources of the NION instead of dedicated hardware.

Step 1

Upon startup, the Ferret's 16 notch filters are initialized to pass the audio unaffected. As the Ferret analyzes the audio and detects a potential feedback condition, it assigns a notch filter. Filter 16 is the first filter assigned with an attenuation of 3 dB centered at the trouble frequency. If more attenuation is required, the filter depth will be progressively increased until the condition is resolved or maximum attenuation is reached. If the feedback is momentary, the filter will be slowly released in .1 dB steps until it either recurs or the filter response is returned to flat and is available for redeployment at the next emergency situation. If the filter depth is repeatedly increased at the same frequency, the filter will be *locked* at this frequency and from this point on will not be slowly released until there is a recompile of *Filter Reset*. The filter recovery feature ensures that the system uses filters only when needed and thus preserving maximum system fidelity.

Step 2

Feedback control devices are typically used on the audio paths from the mixer to the Power Amps. This was mostly a system cost consideration and the Ferret works great here. In addition, the software flexibility of the Ferret within MediaMatrix allows the Ferret to be placed effectively in many new places. Dedicating a Ferret to a single troublesome lavaliere mic is an excellent insurance policy while minimizing the effect on the overall system sound. In addition, multiple mics can be summed and processed by a single Ferret. For those really cost sensitive applications, the control bussing of the Ferret filters introduces the concept of controlling multiple filter banks with a single Ferret.

Step 3

The controls for the Ferret are minimal, consisting of 2 mode selections, a filter configuration select and "Panic" filter configuration. The Ferret provides all necessary configuration and control of the associated filter banks. No operator control is available for the filters.

Step 4

Panic filters are a special class of filters deployed exclusively by the Peavey Feedback Ferret to address the situation of sudden, uncontrolled full-blast feedback. These filters are dispatched when large amounts of energy are concentrated in a single octave. To deal with this, the Ferret has a 2-step approach. Initially, the ferret detects this potentially dangerous condition and the need to take action. There may not be enough information to accurately place a narrow band notch filter, but something has to be done. The panic filter was designed specifically for this situation. The panic filter is deployed with a bandwidth of 1 octave at the approximate center of the problem. The filter is set aggressively, but to preserve sound fidelity, it is limited to a maximum depth of 6 dB. After the crisis is over, the filter is slowly released at .1 dB increments. If the problem recurs as the filter is released, the feedback will build back slowly and the standard, more accurate filter will be set to "kill" the feedback. The result is immediate response followed up with more accurate long-term control if needed.

Step 5: Panic Threshold

Panic filters are brought in to attack the sudden, unexpected and really loud problems. To do this well, the system designer needs to help the Ferret with the definition of "loud". This is where the Panic Threshold comes in. For a typical MediaMatrix system, this control should be set just below Nominal operating level (0 dB). If, however; your design requires more of less headroom, the Panic Filter Threshold will need to be adjusted to compensate for your special situation. Setting this threshold too low may cause unnecessary filters to be set during normal program material. This control has no effect until Panic Filters are enabled.

Step 6: Filter Reset

Once filters have been allocated and locked by the Ferret, recompiling the view or pressing the "Filter Reset" button are the only 2 ways to clear them. During system setup and ring-out, the engineer may adjust the Eq of the system and then use the reset button as a way to tell the Ferret to start over. Before doing this, the engineer should make sure that the system gain is well below feedback, otherwise, serious system damage is possible when the filters that were restraining the Feedback Demon are released. After system ring-out, this control should NEVER be pressed. It's just too dangerous and the possible consequences are too grave (blown speakers and hearing) to allow an operator access to this control. For this reason, we have included an addition "are you sure" level to help prevent inadvertent operation. This control should never appear on the top level of a system view.

Step 7

Slowly bring up the Microphone input gain control. As the system gain approaches feedback, tones or ringing will be heard and quickly killed by the Ferret. As a tone is heard, temporarily stop increasing the gain and allow the Ferret to complete the adjustment of the filters to accurately address the problem mode before going on to the next. Operation of the Ferret can of course be heard, and can also be seen by examining the filter display. During this part of setup, you should practically expect to increase the gain before feedback by 6 to 10 dB. As filters are set, the engineer should monitor the gain and the number of filters used. Setup should be considered finished when either: maximum input gain has been achieved without Feedback, or the Ferret has locked 6 filters, or system gain before feedback has been increased by 10 dB. Filters set during preshow are meant to control dominant room modes. For this reason, if more than 6 filters are needed to obtain the necessary sound level, then you should carefully review your system design looking for system design problems.

Step 8: What to do about Locked Filters

As mentioned earlier, a recompile will initialize the Feedback control filters to flat. Well then the question is "How do I keep my dominant mode settings that I just used preshow mode to configure?" I'm glad you asked. This brings up one of the most powerful reasons for combining MediaMatrix and the Ferret. The Ferret can be used to determine the correct parametric Eq settings and through the flexibility of MediaMatrix these settings can be transferred to MediaMatrix's native fixed Parametric Equalizers. Now all of the filters of the Ferret are available to control transient modes that occur as the sound terrain changes. Additionally, if dynamic control is not desired, the Ferret can be used as a diagnostic tool during setup and then to save DSP cycles, it can be removed from the final view, just leaving the Parametric Equalizers.

Caution: Never feed the filter control output from the Ferret to an audio output. This signal is NOT audio and may damage sensitive components if routed to an audio output.

Note: The Ferret is not intended as an alternative to good system design. With a good system design it can be an effective insurance policy for those unexpected little problems that always crop up when you least expect them. With a poor sound system design, the Ferret will only "band-aid" the problem and you will probably still not be pleased with the overall system performance. The Ferret is just another tool to help you do a better job easier and quicker.

Using Ambient Sensing Levelers to ensure audio can be heard in a noisy venue

Each Ambient Sensing Leveler device provides an automatic *volume control* when the ambient noise levels in a venue start to affect the perceptibility of paging announcements and background music.

The device measures the following:

- The incoming program material level.
- The ambient level from the incoming zone (using a microphone connected to the sense input).
- The output level of the Ambient Sensing Leveler device.

The device then applies gain to the input signal based on the detected ambient level.

The different variants of the device provide the following features:

Continuous sensing	The ambient sound level is measured continuously.
Gap sensing	The ambient sound level is measured during gaps in the program material. The Gap version of the Ambient Sensing Leveler is easier to set up and consumes fewer DSP resources than the regular version. Also, the device accurately calculates the necessary gain even when there are long gaps in the program material.
Router	The device switches inputs based on the ambient noise level.

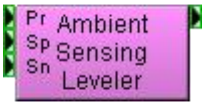
Setting up an Ambient sensing leveler with continuous sensing

Careful sense microphone placement is essential for optimum performance of the Ambient Sensing Leveler. Optimum microphone placement is achieved when the output of the speakers in the zone does not leak back into the sense microphone. This may not be practical in some facilities, so to avoid this problem, the Ambient Sensing Leveler must first go through a system calibration phase before it can accurately estimate the ambient level separately from the speaker leakage (returned through the sense mic).

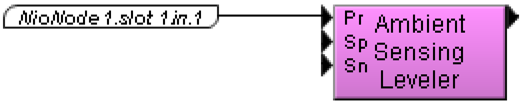
Calibration should be performed after all speakers and microphones are installed and set at their final positions and settings. Moving a microphone or changing the level control on the power amp (or any device after the Ambient Sensing Leveler is in the circuit) will require the system to be recalibrated. Although it is not necessary to have a completely quiet room for calibration, a constant ambient level is required.

Adding the Continuous sensing Ambient sensing leveler to your design and wiring it up

1. In the device tree, expand the **DSP / Dynamics / Ambient Sensing Levelers** folder.
2. Drag a **Continuous Sensing** or **Continuous Sensing w/Router** device over to the design page.

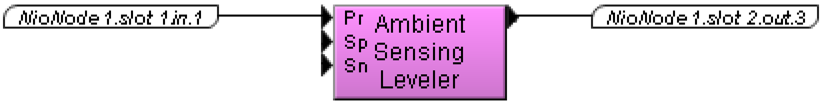


3. Wire the program source (typically background music and/or paging audio) directly to the **Pr** (program source) input wiring node.

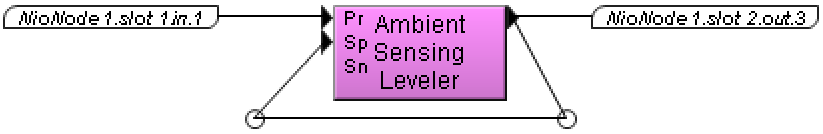


Any gain added or subtracted from this source starts from 0dB (unity gain), which is provided when the Ambient Sensing Leveler is bypassed.

4. Wire the output wiring node to the amplifier that drives the zone.



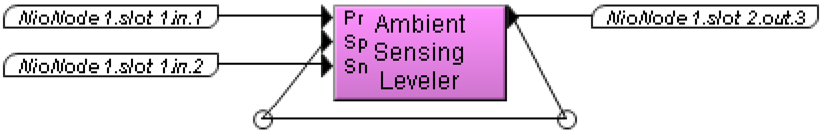
5. Wire the output wiring node to the speaker input (**Sp**) on the block.



   The *Sp* input is the return of the audio you would expect to hear from the speaker. In other words, the audio going into this node should be a duplicate of what the system is delivering to the amplifier that drives the zone.

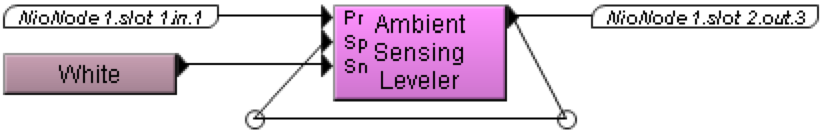
   Tip: The speaker input is only used for measurement and will not pass audio through to the Ambient Sensing Leveler output node.

6. Wire the sense input (**Sn**) to a microphone which has been set up to sense ambience in the zone.



   In more complex projects, the microphone input may be processed by equalizers or filters before being fed into the *Sn* input.

   As an alternative to using the microphone input, you can use a White Noise Generator to test the system.



Note: When properly calibrated, the Ambient Sensing Leveler device will effectively ignore most program material coming into the microphone. Anomalies may occur as a result of locating the sense mic too close to the speakers. The Ambient Sensing Leveler will work best if the microphone is not in the direct or indirect path of any speakers.

Calibrating and using the Ambient sensing leveler

1. Verify the placement, working condition and gain staging of all microphones.
2. Make sure the Ambient Sensing Leveler device has been correctly wired, and that there is a NioNode in your project, then deploy it.
3. Open the **Ambient Sensing Leveler** block.
4. Next to **Gain Update**, click Bypass B.
5. Set the **Sense RMS Time** to 3 seconds, or a larger value for a noisier environment.
6. Verify the placement of all microphones.
7. Gain stage the elements in the circuit properly, so that the output volume (**Gain** meter) in the zone is at the maximum expected level.

   Make sure the **Spkr** (speaker) and **Sense** input levels are in the range 0 to -20dB. This allows good resolution and sufficient headroom for ambient increases.
8. Wait for the **Spkr** (speaker) and **Sense** input levels to settle, and then start the program material or pink noise.

9. Click **Start**.
10. Decrease the level of the source material connected to the **Pr** (program source) input to the proper listening level for the current ambience level, then wait for the RMS levels to settle at this new level.
11. Click **Start** again to deselect the calibration mode.

At this time, *Room Gain* will be updated and the meter under *Ambient* Level will begin to reflect the true ambient level. Upon successful calibration, you may now set the dynamics controls for the gain cell (in the *Gain Update* section) without having to recalibrate the device.

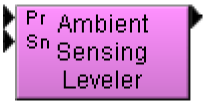
Calibration information will be saved with the NWare project file as default information (for each instance of the Ambient Sensing Leveler device). Presets or SubPresets may also be used to save calibration and device settings.

Setting up a Gap ambient sensing leveler

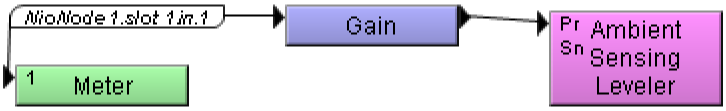
To set up the Gap Ambient Sensing Leveler, minimum and maximum ambient levels are simulated and the device controls are adjusted accordingly.

Adding the Gap Ambient sensing leveler to your design and wiring it up

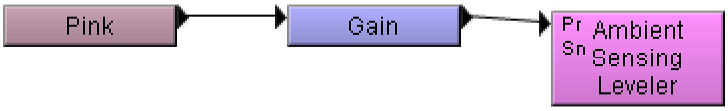
1. In the device tree, expand the **DSP / Dynamics / Ambient Sensing Levelers** folder.
2. Drag a **Gap Sensing** or **Gap Sensing w/Router** device over to the design page.



3. Wire the unprocessed program material to a Gain control, and then to the **Pr** (program source) input wiring node.
4. Wire the unprocessed program material to a meter.



If it is available, the intended program material should be used to configure this device. However, if it is not yet available, you can use a Pink Noise Generator to approximate the effect.



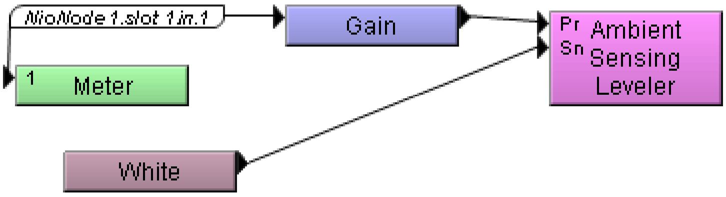
The signal is examined for silence (gaps) that indicate when it is safe to measure the ambient noise sense signal.

5. If you are using the Router version of the device, create the desired processing chains and wire them into the other program input(s).

   The processing chains will be different for different operating conditions, but could be set to have different equalization settings depending on the amount of ambient noise.

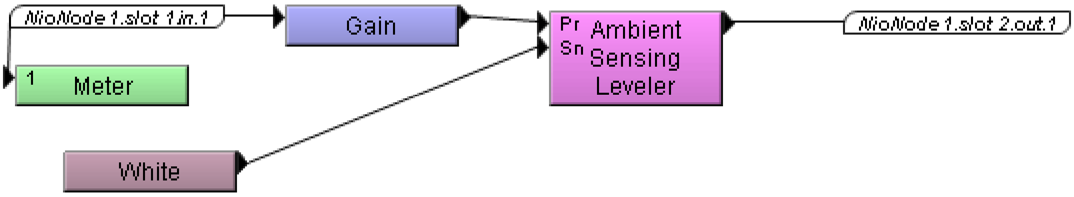
Tip: You can change the number of program inputs in the device properties.

6. Wire the sense input (**Sn**) to a White Noise Generator.



When the program material input detects a gap in the program, the device measures the signal level at this input and applies gain based on this level.

7. Wire the output wiring node to the amplifier that drives the zone.



Using the Gap Ambient Sensing Leveler

All versions of the device

1. Verify the placement, working condition and gain staging of all microphones.
2. Make sure the Ambient Sensing Leveler device has been correctly wired, and that there is a NioNode in your project, then deploy it.
3. Open the **Ambient Sensing Leveler** block.
4. Measure the noise floor on the program input (**Pr**) using the meter.

   If you are using the Router version of the device, use the **P1** input.
5. Start the program material.
6. Set the **Gap Thres Level** to approximately 10 dB above the noise floor measurement.

Note: If you set the Gap Threshold Level too high, the program material will leak into the ambient noise sensor.

7. If the audio system is installed in an industrial or public transportation environment, set the **Sense RMS Time** to 2 seconds.

   If the audio system is installed in a different environment, such as a casino or dining room, set the **Sense RMS Time** to a value greater than 2 seconds.
8. Next to **Gain Update**, click Bypass B.
9. Switch on the program material (or Pink Noise Generator).
10. Adjust the gain on the Gain device (or Pink Noise Generator) so that there is no detectable ambient noise.
11. Next to **Gain Update**, click Bypass B.

    This switches off bypass, so that the gain controls can now be configured.
12. Using the White Noise Generator, simulate low ambient noise level conditions and adjust the **Amb(**ient) **Thresh**(old) **Level** to the level where the ambient noise begins to affect the intelligibility or perceptibility of the program material.
13. Set the **Min**(imum) **Gain** control to amplify the program material to overcome the ambient noise.
14. Mute the program input (either the gain device or the Pink Noise Generator)

15. Use the Meter device and the **Amb(**ient) level meter on the Ambient Sensing Leveler device to measure the time taken for the program material to decay by at least 10dB below the ambient noise level.
16. Adjust the **Sense Delay Time** to match the length of time it takes for the program material to decay by at least 10dB below the ambient noise level in the room.
17. Using the White Noise Generator, simulate high ambient noise level conditions and set the **Max**(imum) **Gain** control to amplify the program material to overcome the ambient noise at this level.
18. Set the **Ratio** to reflect the amount of program material gain change (in dB) that you would like the device to apply per dB of ambient noise level change.
19. Adjust the **Resp**(onse) **Time**.

    If the ambient levels peak and drop very quickly in your environment, a long response time will ensure that the response time of the device will be more natural.

Additional steps for Gap Ambient Sensing Leveler

1. Under **Router**, adjust the **Thresh Levels** to the desired levels.
2. Set the **Hyst**(eresis).

   This provides a guard band around the router threshold levels in the event of ambient noise hovering around a threshold. A good starting value for Hysteresis is +0.5 dB.
3. Set the **Xfade** (crossfade) **Time**.
4. Generate the program material on all router inputs and test the device to ensure it functions correctly.

   The LEDs in between the threshold level controls indicate which router input is currently selected.

Linking NWare projects together

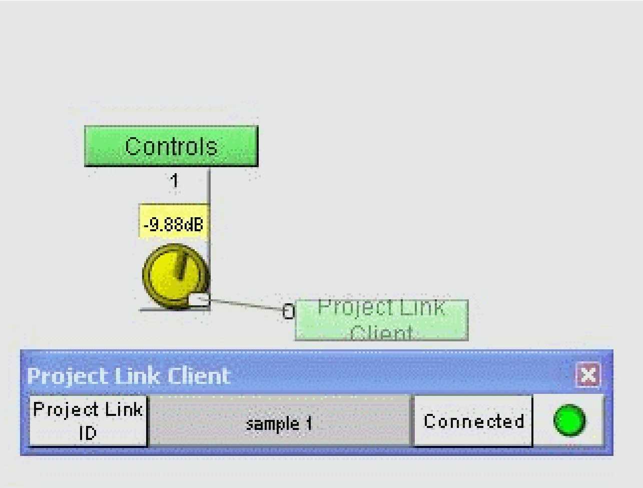
Project linking allow controls in one project to be linked to controls in another project. When the user gestures a knob control in the client project, for example, the value can be passed to a knob control in the server project. As the first knob moves, the second knob moves at exactly the same time.

A Project Link Server device is placed in the server project, and Project Client devices are placed in the client projects (there can be one or more).

Each Project Link Server is given a unique ID. Project Link Clients can then connect to the server using the ID. Once connected, control signals can flow in one direction between the two projects, or in both directions, depending on how you have configured the Project Link devices.

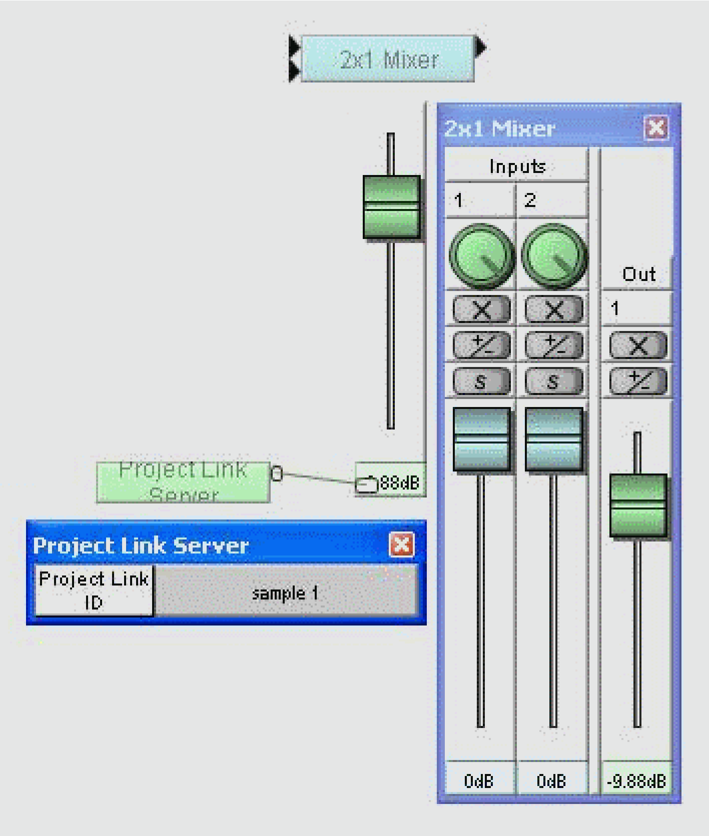
In the example below, the first project – the client – connects to the second project – the server – using the id *sample 1*. The Connected LED shows the projects are linked.

The first project has a knob control linked to a Project Link Client device. In the second project, a Project Link Server device passes the control values from the client project directly to a fader control inside a mixer.



Project Link ID sample 1

Connected



Project Link ID sample 1

Adding a Project Link Server device

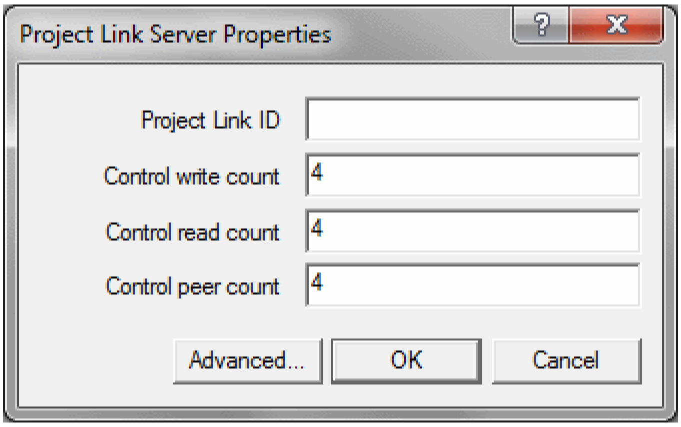
A Project Link Server device is placed in the project that will be linked to the *client* project. There can be several *client* projects for each *server* project.

It is assumed that your *server* project already contains the devices that you would like to link to devices in client projects.

⏩ *To add a Project Link Server device*

1. Make sure you are editing the *server* project.
2. In the device tree, expand the **Control** folder.
3. Drag a **Project Link Server** device over to the design page.

   The **Project Link Server Properties** dialog box is displayed.



Project Link Server Properties

Project Link ID:

Control write count: 4

Control read count: 4

Control peer count: 4

4. If you want to fix the project link identifier that clients need in order to connect to the server, type the name in the **Project Link ID** box.

   If you want the user to be able to specify the identifier when the project is deployed or emulated, leave the box empty.
5. In the **Control write count** box, type the number of controls that will be receiving values via the Project Link Client from the Project Link Server.

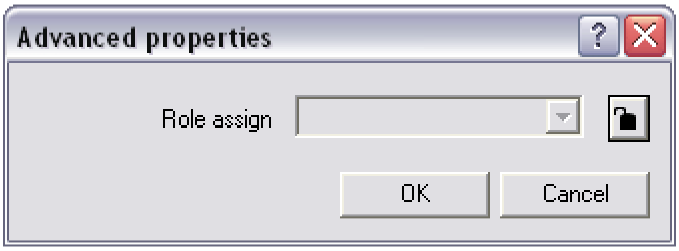
   These controls must be slave wired to the *read* wiring nodes on right of the device block. The valid range is 0 to 200.
6. In the **Control read count** box, type the number of controls that will be sending values via the Project Link Server over to the Project Link Clients.

   These controls must be master wired to the *read* wiring nodes on the left of the device block. The valid range is 0 to 200.
7. In the **Control peer count** box, type the number of controls that will be both sending and receiving values via the Project Link Server to/from the Project Link Client.

   These controls must be *master* wired to the *peer* wiring nodes on the left of the device block. The valid range is 0 to 200.
8. If you want to change the role to which this device belongs:

   a) Click **Advanced**.

   The **Advanced properties** dialog box is displayed.



Advanced properties

Role assign:

   b) Click the 🔓 button to change it to the 🔒 button, and then select a role name in the list.

   c) Click **OK**.

9. Click **OK**.

Wiring controls to the Project Link Server device

You must use the correct type of wiring node on the Project Link Server device when you want to wire the block to devices in your server project. NWare will prevent you from wiring the wrong types of nodes. If this happens:

- Check that you have chosen the correct options on the device properties of the Project Link Server device.
- Check that you are wiring the correct node on the block.
- Check that the device you are wiring has the correct master/slave/peer wiring option selected on the device properties.

Tip: When you hover the mouse pointer over a wiring node on the Project Link Server block, a tooltip is displayed showing the wiring node type.

⏩ *To wire a control that will receive values from linked client controls*

1. Add a slave wiring node to the control.
2. Control wire the wiring node to the *write_x* node on the Project Link Server device.

⏩ *To wire a control that will send values to linked client controls*

1. Add a master wiring node to the control.
2. Control wire the wiring node to the *read_x* node on the Project Link Server device.

⏩ *To wire a control that will send values to, and receive values from, linked client controls*

1. Add a master wiring node to the control.
2. Control wire the master wiring node to the *peer_x* node on the Project Link Server device.

Adding a Project Link Client device

A Project Link Client device is placed in the project that will be linked to the *server* project. There can be several *client* projects for each *server* project.

It is assumed that your *client* project already contains the devices that you would like to link to devices in the server project.

As part of the configuration, you can specify a setting called *Poll period*. This controls the frequency at which the client device will poll the server device. It can impact both network resources and NioNode resources, so it is important to understand the effect specifying different values will have.

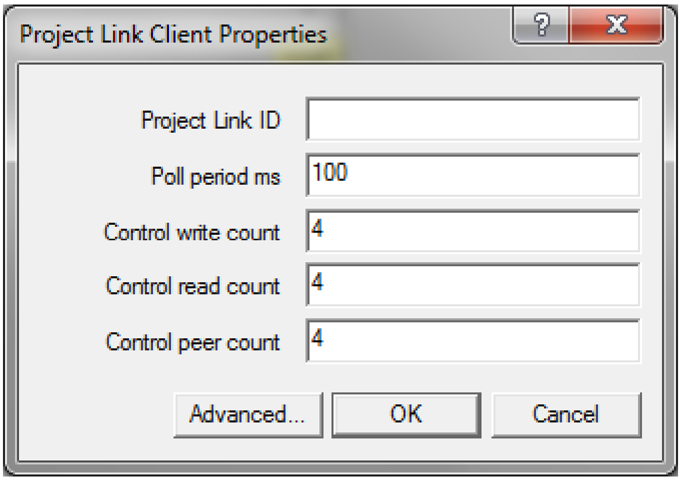
The default value of 100ms means the server is polled 10 times a second. A value of 10ms means the server is polled 100 times per second, and so on. Using the default value usually results in good resource usage and a good GUI response time. Setting the polling value too high can result in a sluggish response from a control, i.e. too much latency between when the control is gestured and when the change is updated in the GUI. Always try to use good judgment when balancing the poll rate to safeguard against overuse of resources while providing a responsive interface.

⏩ *To add a Project Link Client device*

1. Make sure you are editing a *client* project. There may be more than one.
2. In the device tree, expand the **Control** folder.

3. Drag a **Project Link Client** device over to the design page.

   The **Project Link Client Properties** dialog box is displayed.



Project Link Client Properties

Project Link ID

Poll period ms 100

Control write count 4

Control read count 4

Control peer count 4

4. If you already know the project link identifier for the server project and you do not want the user to be able to change it when the project is deployed or emulated, type the name in the **Project Link ID** box.

   If you want the user to be able to specify the identifier when the project is deployed or emulated, leave the box empty.
5. In the **Poll period ms** box, type the period in milliseconds between status update requests from client to server. The valid range is 10 to 60000.
6. In the **Control write count** box, type the number of controls that will be sending values via the Project Link Client over to the Project Link Server.

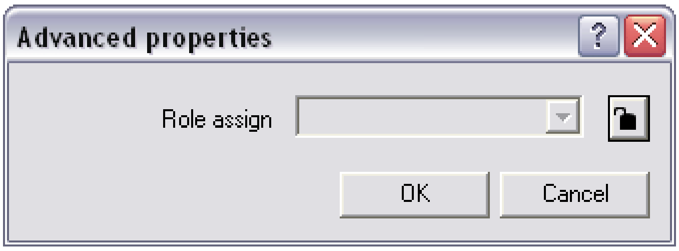
   These controls must be master wired to the *write* wiring nodes on the left of the device block. The valid range is 0 to 200.
7. In the **Control read count** box, type the number of controls that will be receiving values via the Project Link Client from the Project Link Server.

   These controls must be slave wired to the *read* wiring nodes on right of the device block. The valid range is 0 to 200.
8. In the **Control peer count** box, type the number of controls that will be both sending and receiving values via the Project Link Client to/from the Project Link Server.

   These controls must be *master* wired to the *peer* wiring nodes on the left of the device block. The valid range is 0 to 200.
9. If you want to change the role to which this device belongs:

   a) Click **Advanced**.

   The **Advanced properties** dialog box is displayed.



Advanced properties

Role assign

   b) Click the 🔓 button to change it to the 🔒 button, and then select a role name in the list.

   c) Click **OK**.

10. Click **OK**.

Wiring controls to the Project Link Client device

You must use the correct type of wiring node on the Project Link Client device when you want to wire the block to devices in your client project(s). NWare will prevent you from wiring the wrong types of nodes. If this happens:

- Check that you have chosen the correct options on the device properties of the Project Link Client device.
- Check that you are wiring the correct node on the block.
- Check that the device you are wiring has the correct master/slave/peer wiring option selected on the device properties.

Tip: When you hover the mouse pointer over a wiring node on the Project Link Client block, a tooltip is displayed showing the wiring node type.

⏩ *To wire a control that will receive values from linked server controls*

1. Add a slave wiring node to the control.
2. Control wire the wiring node to the *read_x* node on the Project Link Client device.

⏩ *To wire a control that will send values to linked server controls*

1. Add a master wiring node to the control.
2. Control wire the wiring node to the *write_x* node on the Project Link Client device.

⏩ *To wire a control that will send values to, and receive values from, linked server controls*

1. Add a master wiring node to the control.
2. Control wire the master wiring node to the *peer_x* node on the Project Link Client device.

Chapter 8

Running NWare in Kiosk mode

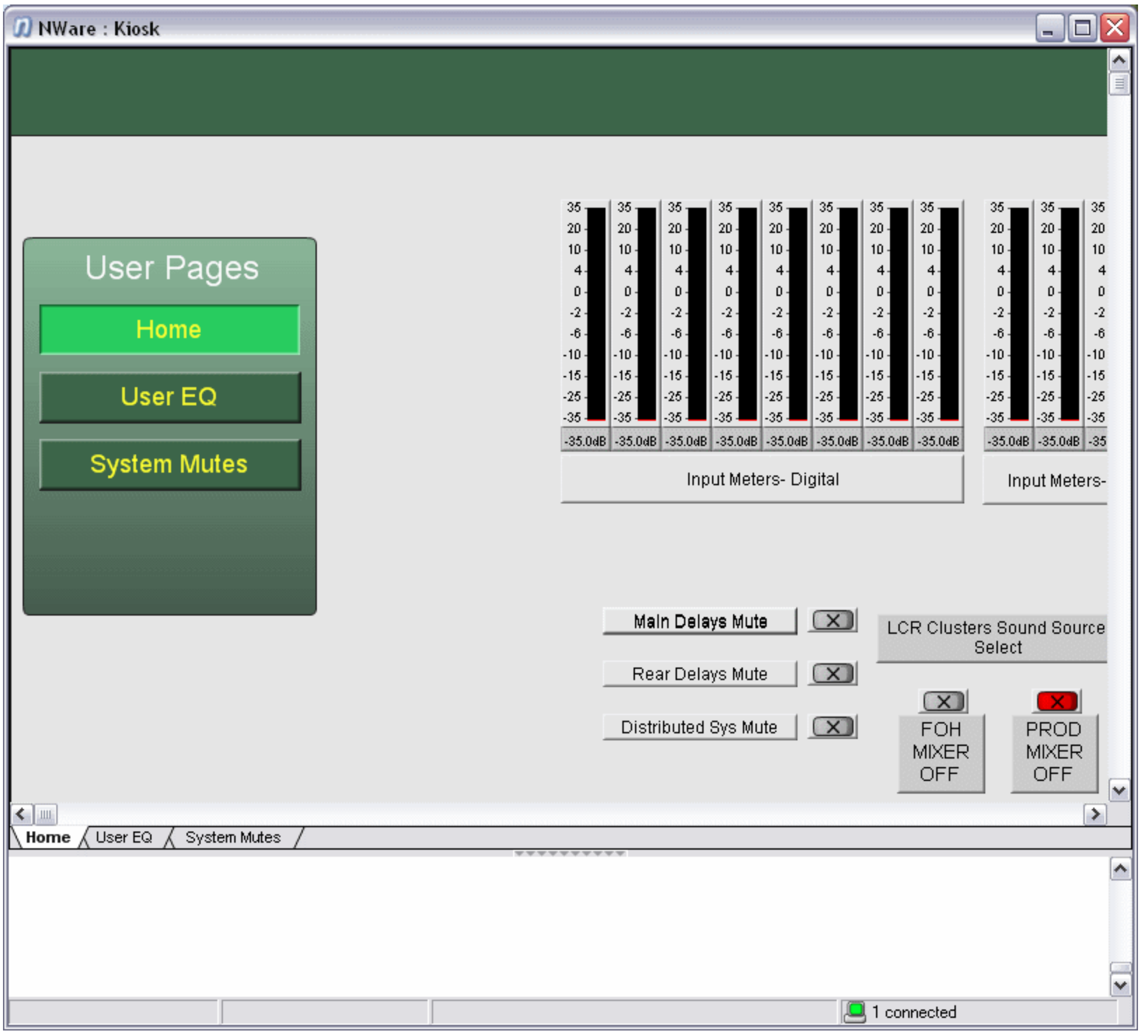
In This Chapter

Introduction

NWare:Kiosk or just *Kiosk* is an NWare client that allows a user to make adjustments to some control values in an audio system, such as volume levels or preset recall, but does not allow any changes to the design of the system.

Note: This is a different program to Kiosk2Go, which is web-based. If you want to know more about Kiosk2Go, see *Setting up Kiosk2Go* (on page 229).

When you run Kiosk, you can use the controls shown on pages like the one below, but some user interface (UI) features normally available in NWare, such as the tree view, will not be available.



Kiosk offers a number of benefits, including:

- Multiple user/password and security options.
- Remote control over Ethernet or wi-fi.
- Multiple project monitoring and control.
- Multiple user connections to the same NWare project.

Tip: Switching between NWare:Kiosk mode and regular NWare mode can be done using a simple command line option. This means it is possible for users to remove the command line option and run the full version of NWare on their PCs. If you would like to prevent users from running the full version of NWare, you can install a special version of NWare that will only run in Kiosk mode. You can download this software from the *MediaMatrix website* (*http://www.peaveycommercialaudio.com/*).

Task overview

Below is a summary of what you will need to do in order to make pages from an NWare project available in Kiosk. We recommend that you read through the steps carefully, even if you are an experienced NWare user.

1. Open NWare.
2. Load the project containing the pages you want to make available in Kiosk.
3. *Check that the page sizes are compatible with the monitors used with Kiosk* (on page 207).
4. *Export pages from the project* (on page 208).
5. *Check the user preferences and make changes as required* (on page 211).
6. Deploy the project.
7. *Run Kiosk and select the project* (on page 212).

If you want Kiosk to start automatically and load a project, see *Logging on and loading a project automatically* (on page 216).

If you want to show or hide specific UI features in Kiosk, or improve password security when logging on automatically, see *Customizing Kiosk* (on page 215).

Preparing pages for display in Kiosk

Sizing pages to fit users' monitors

It is important to consider how the pages in your NWare project will look when they are displayed on your users' monitors. The resolution of a monitor determines how much viewable space is available, so a resolution of 1280x1024 will provide more space than one of 1024x768. You must also consider the on screen elements, including the status bar and the border of the window, as they will take up space. Most XVGA (1280x1024) monitors will therefore provide about 1245x905 pixels of usable screen space.

⏩ *To add a sizing guide to a page*

1. Expand the device tree to display **Graphic Elements & Blocks** / **Labels**.
2. Drag a label over to the page.
3. Right-click the label, and then click **Graphic Properties**.
4. Under **Shape**, click the radio button next to the list to enable it.

   In the list, **Rectangle** is selected by default.
5. Under **Size,** select the **Lock Size** check box.
6. Type the monitor resolution into the **Width** and **Height** boxes.
7. Click **OK**.
8. Drag the rectangle to position it on the page.
9. Right-click the rectangle, and then click **Lock**.

The rectangle will be locked in place, so it cannot be moved accidentally.

10. If possible, before moving forward with the design of the page, set a monitor to the expected resolution for your users and check the appearance of the rectangle in Kiosk to ensure it is clearly visible.
11. Use the rectangle as a sizing guide for the screen area while you complete the rest of the design.
12. When you have finished the design, delete the rectangle.

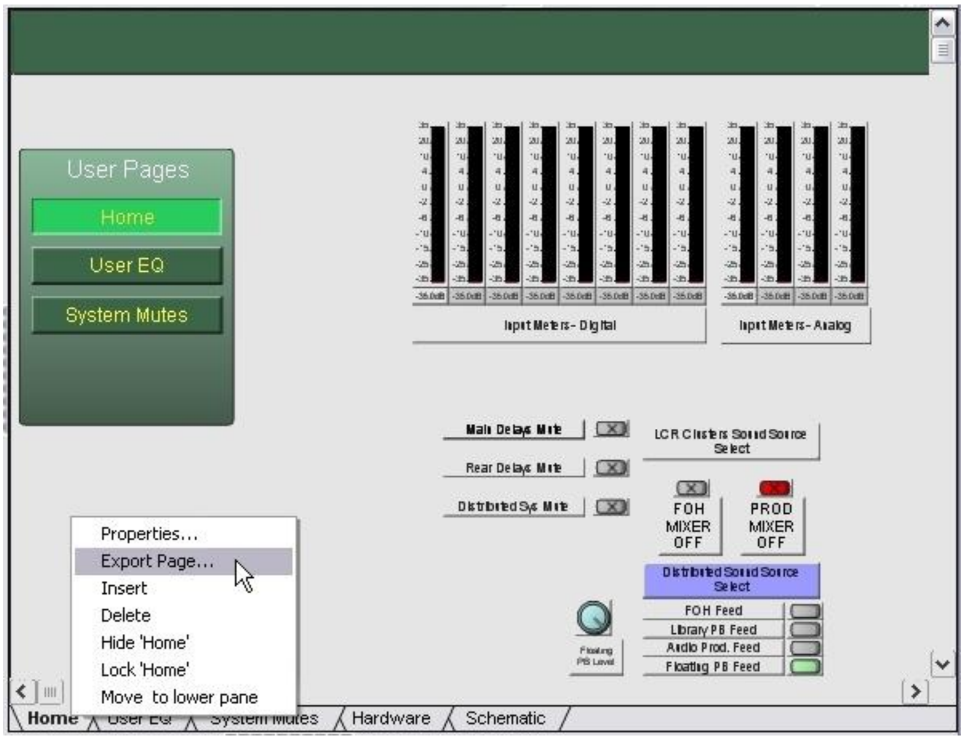
    If you leave the rectangle in place, it may be visible to users with higher resolution monitors.

Exporting pages from an NWare project

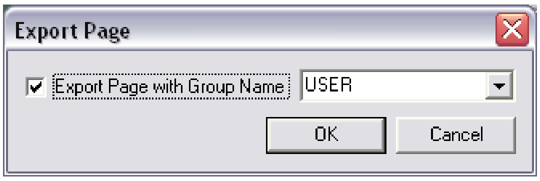
In order to make pages from your NWare project available to Kiosk users, the pages must be exported to a page group, and then the users must be granted access permissions to the page group. The pages are still stored within the NWare *.npa* file, but they will be made available when you load the project into Kiosk.

⏩ *To export pages from an NWare project*

1. Right-click the page tab, and then click **Export Page**.



The **Export Page** dialog box is displayed.


Export Page

☑ Export Page with Group Name: USER

2. Select the **Export page with Group Name** check box.
3. If you want to use an existing group, click the group name in the list.

   If you want to create a new group, type the group name in the list.

Notes:

- The group name cannot contain certain characters, including *, / or \. A warning will be displayed if you use these characters.
- In order for a Kiosk user to be able to access the page you are exporting, they must be a member of the group you specify. For information on adding users to the group, see *Setting the preferences in user accounts* (on page 211).

4. Click **OK**.

Rules for determining the start page

NWare uses rules to determine which exported page appears first when you start Kiosk. Rules are shown in priority order.

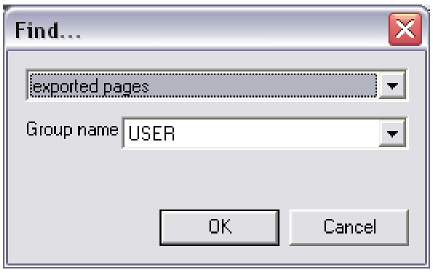
Single page group exported
1. The word *home* is specified in the page name.
2. The page name appears first when the list of page names is sorted into alphabetical order.

Multiple page groups exported
1. The page is a member of a page group. When the list of page group names is sorted into alphabetical order, the page group name appears last.
2. The word *home* is specified in the page name.
3. The page name appears first when the list of page names is sorted into alphabetical order.

Finding exported pages

1. In NWare, on the **Edit** menu, click **Find**.

   The **Find** dialog box is displayed.



Find...

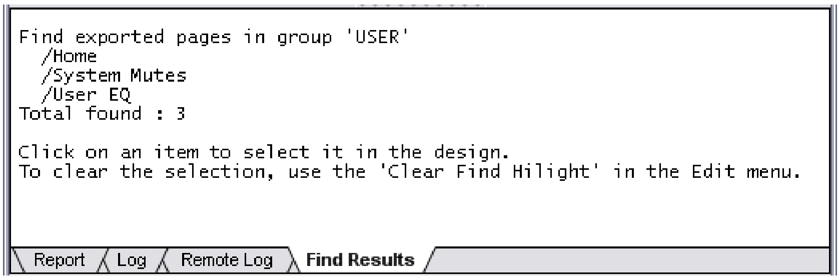
exported pages

Group name USER

2. In the first list, click **exported pages**.
3. If the pages are in a group, click the group name in the **Group name** list, otherwise click ***** in the **Group name** list.

4. Click **OK**.

   The results of the search are shown in the output frame at the bottom of the window.



```
Find exported pages in group 'USER'
  /Home
  /System Mutes
  /User EQ
Total found : 3

Click on an item to select it in the design.
To clear the selection, use the 'Clear Find Hilight' in the Edit menu.
```

Tip: You can jump directly to the pages in the search results by clicking them.

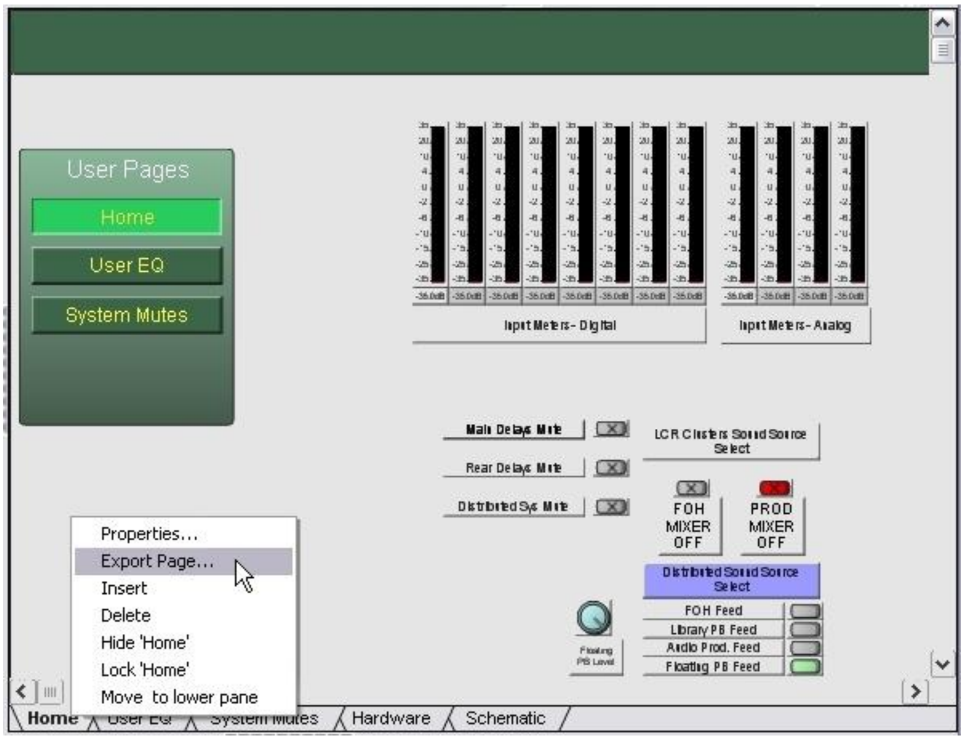
Deleting a page group

Page groups are deleted automatically when they no longer contain any exported pages. In order to delete a page group, you must change the export settings for each page in the group and remove it from the group.

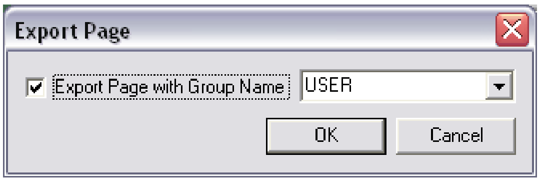
Tip: You can locate all the pages in a particular page group using the Find function. For more information, see *Finding exported pages* (on page 209).

⏩ *To delete a page group*

1. Right-click the page tab of a page in the group you want to delete, and then click **Export Page**.



The **Export Page** dialog box is displayed.



2. Clear the **Export page with Group Name** check box.
3. Click **OK**.
4. Repeat the above steps for all the pages in the group.

   Once you have completed the process, the group will be deleted and the group membership settings of the users that used to belong to the group will be updated.

Setting the preferences in user accounts

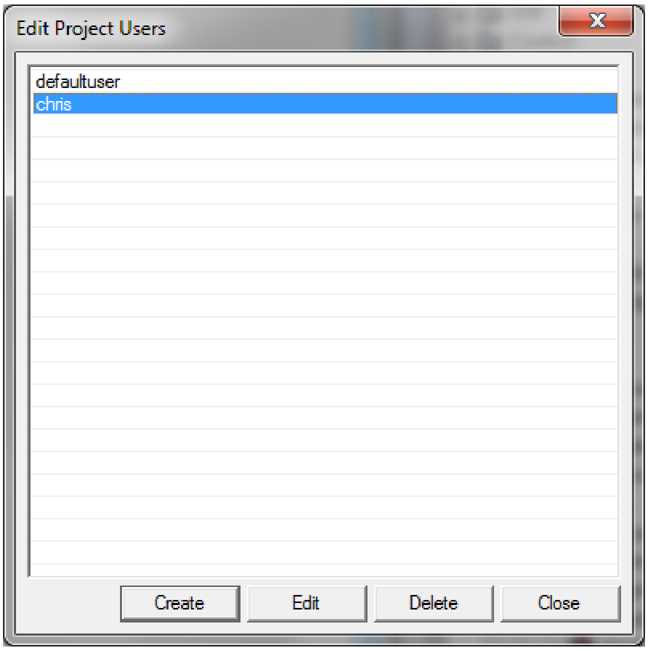
When you run Kiosk, you will be required to log on before you can view exported pages. You can specify any user name from the NWare project, as long as the user has been granted access to the exported pages.

Tip: The procedure below shows how to specify the settings for a typical Kiosk user. For particular installations, however, you may want to use different settings, to restrict user actions further or allow a greater number of actions to be performed.

To set the preferences in user accounts

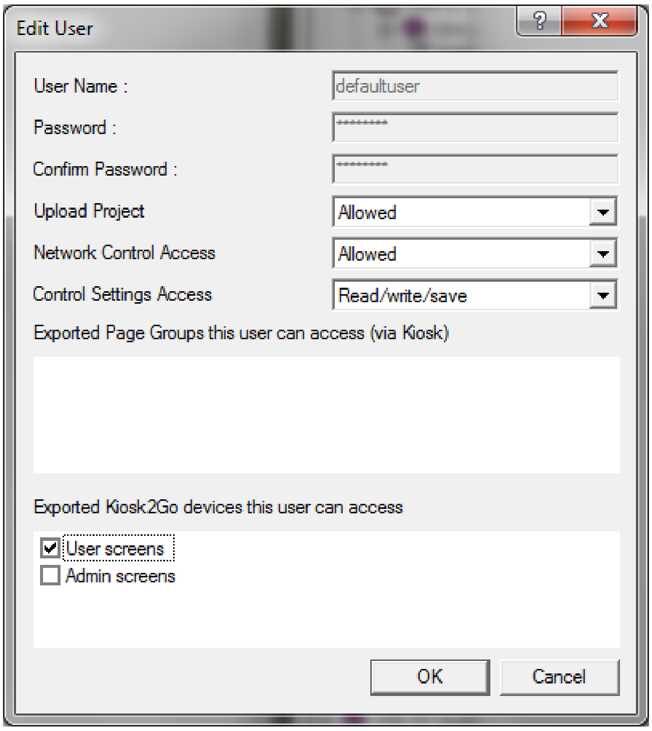
1. On the **File** menu, click **User Accounts**.

   The **Edit Project Users** dialog box is displayed.



2. Click the user you want to configure, then click **Edit**.

The **Edit User** dialog box is displayed.



Edit User

User Name : defaultuser

Password : *********

Confirm Password : *********

Upload Project: Allowed

Network Control Access: Allowed

Control Settings Access: Read/write/save

Exported Page Groups this user can access (via Kiosk)

Exported Kiosk2Go devices this user can access

☑ User screens

□ Admin screens

3. In the **Network Control Access** list, click **Allowed**.
4. In the **Control Settings Access** list, click **Read/write** or **Read/write/save**.
5. In the **Exported Page Groups this user can access (via Kiosk)** list, select the check box next to each page group the user will access.

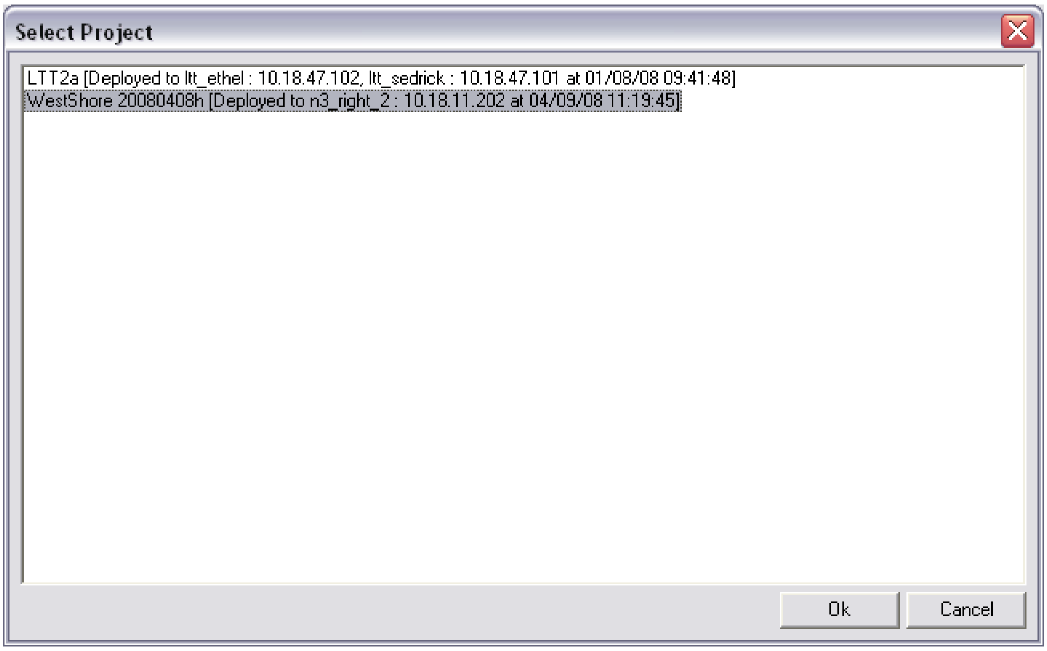
Note: Pages are organized into specific groups when they are exported, so you must specify the groups the user may access, otherwise they will not be able to use the pages in Kiosk.

6. Click **OK**.
7. Click **Close**.

Starting Kiosk and selecting a project

1. On the **Start** menu, point to **All Programs**. Point to **NWare**, and then click **NWare Kiosk <version>**.

The **Select Project** dialog box is displayed.

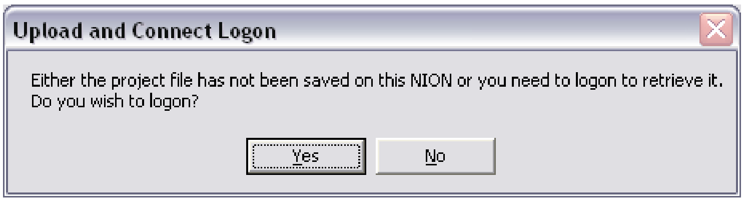


Select Project

LTT2a [Deployed to ltt_ethel : 10.18.47.102, ltt_sedrick : 10.18.47.101 at 01/08/08 09:41:48]
WestShore 20080408h [Deployed to n3_right_2 : 10.18.11.202 at 04/09/08 11:19:45]

2. Click the project you want to run in Kiosk.
3. Click **Ok**.

   If a logon is required, the **Upload and Connect Logon** dialog box will be displayed.



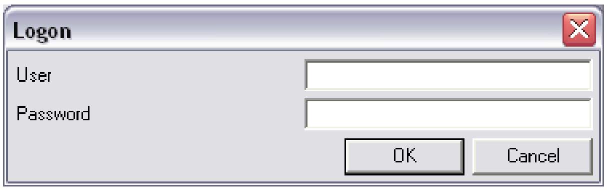
Upload and Connect Logon

Either the project file has not been saved on this NION or you need to logon to retrieve it.
Do you wish to logon?

4. Click **Yes**.

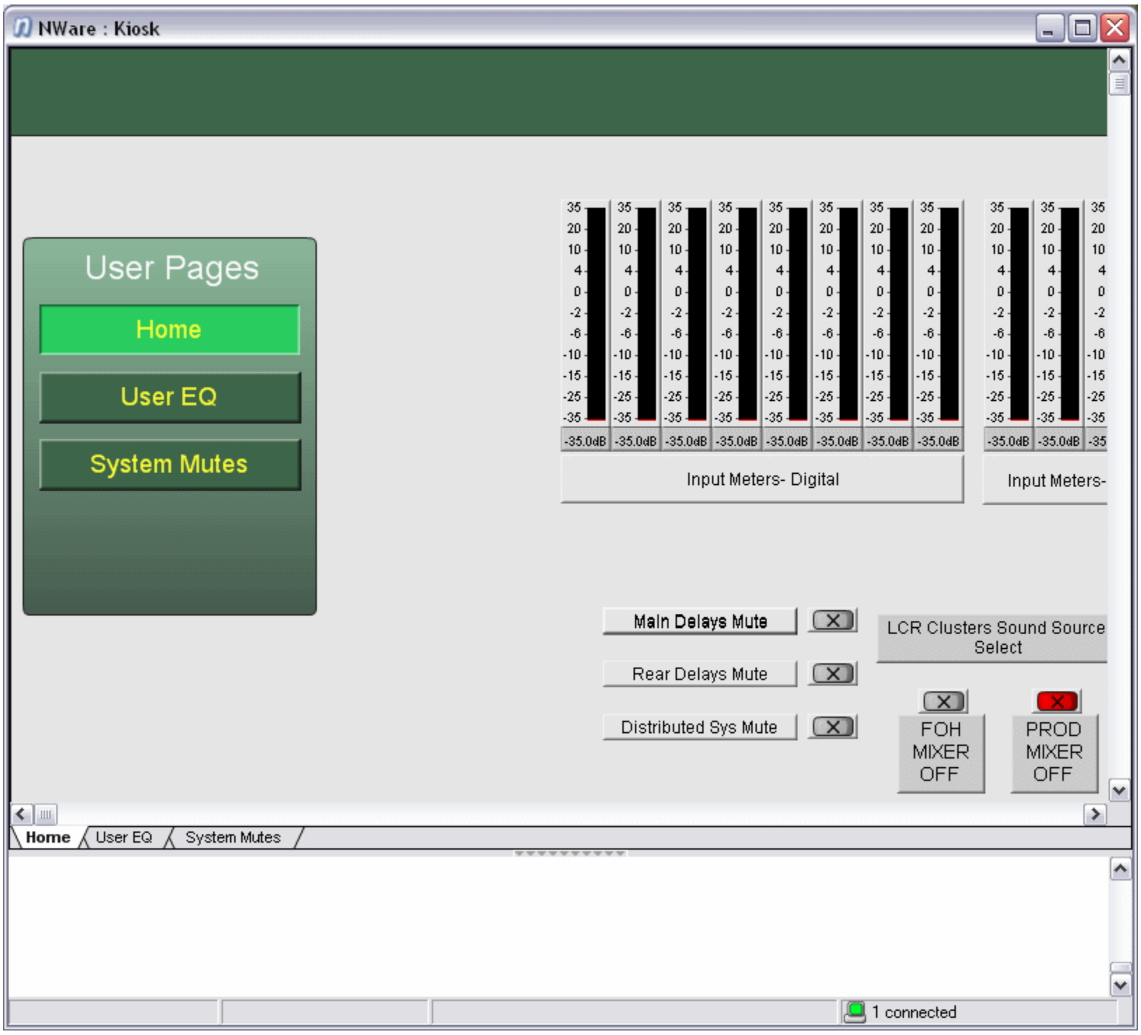
Tip: The part of the message about the project not being saved can be ignored. Only the part about logging on is relevant here.

The **Logon** dialog box is displayed.



5. Type the username and password, and then click **OK**.

The Kiosk main window will be displayed.



The window will show individual tabs for the pages you have exported.

Tips:

- You can drag the page tabs to change the page order. However, when you restart Kiosk and connect to the project, the original order will be restored.
- Some UI features normally available in NWare will be disabled. You can also disable further UI features if necessary. For more information, see *Customizing Kiosk* (on page 215).

Customizing Kiosk

Using command line options

Creating a new shortcut for NWare Kiosk

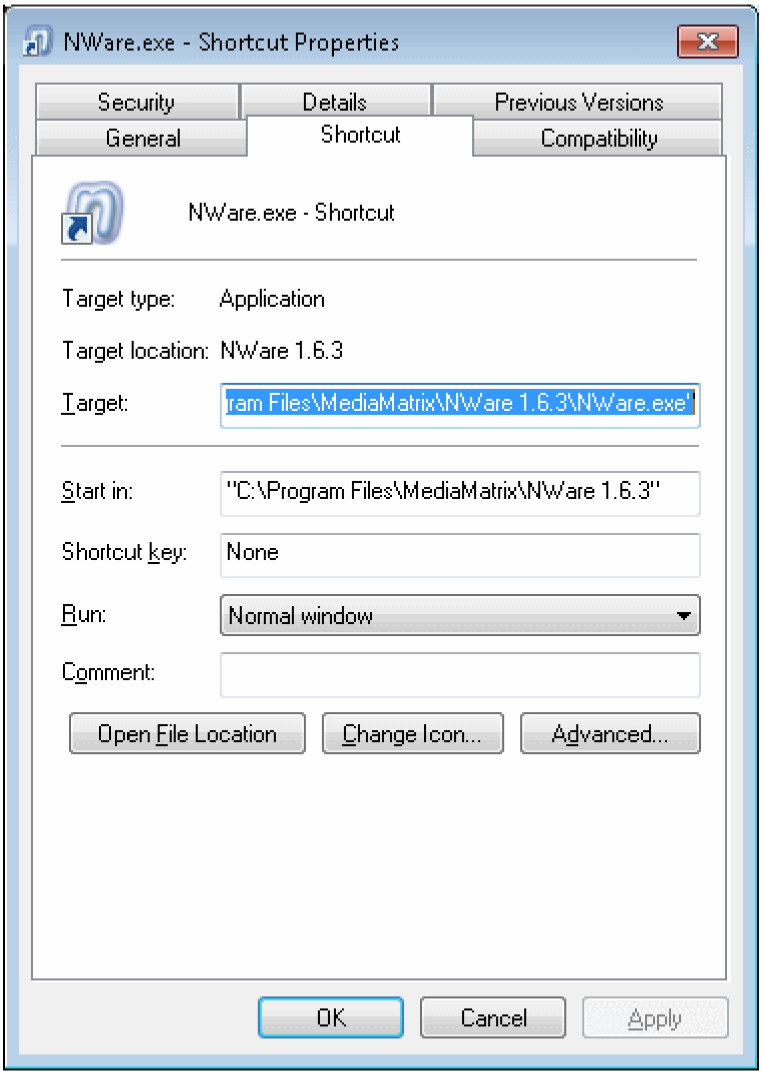
Before you can specify different command line options to customize Kiosk, you must create a new shortcut to *nware.exe*. You cannot use the existing shortcut in the Start menu.

The procedure below is relevant if you are running the version of Kiosk that is installed automatically with NWare, or the special standalone version.

⏩ *To create a new shortcut for NWare Kiosk*

1. Open Windows Explorer and navigate to the *NWare <version>* or *NWare Kiosk <version>* installation folder.
2. Right-click *NWare.exe*, and then click **Create Shortcut**.
3. Right-click the shortcut, and then click **Properties**.

   The **Shortcut to NWare.exe Properties** dialog box is displayed.



NWare.exe - Shortcut Properties

NWare.exe - Shortcut

Target type: Application

Target location: NWare 1.6.3

Target: ram Files\MediaMatrix\NWare 1.6.3\NWare.exe"

Start in: "C:\Program Files\MediaMatrix\NWare 1.6.3"

Shortcut key: None

Run: Normal window

Comment:

4. If you are creating a shortcut to *nware.exe* in the NWare installation folder, add this switch to the command in the **Target** box:

```
/personality=nware_kiosk
```

Note: This switch is not required for the standalone version of Kiosk.

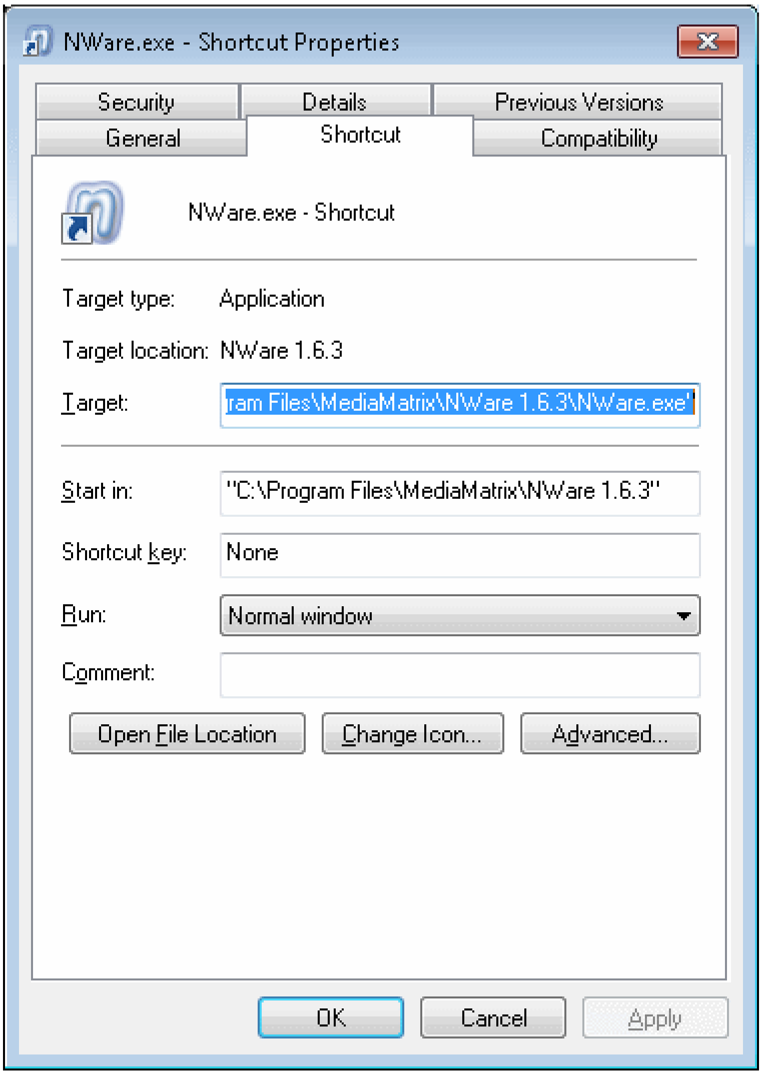
Logging on and loading a project automatically

You can avoid having to specify a username, password and project name each time you start Kiosk, by specifying command line options for the new shortcut you created.

⏩ *To log on and load a project automatically*

1. Open Windows Explorer and navigate to the folder containing the NWare Kiosk shortcut.
2. Right-click the shortcut, and then click **Properties**.

   The **Shortcut to NWare.exe Properties** dialog box is displayed.



NWare.exe - Shortcut Properties

NWare.exe - Shortcut

Target type: Application

Target location: NWare 1.6.3

Target: ram Files\MediaMatrix\NWare 1.6.3\NWare.exe"

Start in: "C:\Program Files\MediaMatrix\NWare 1.6.3"

Shortcut key: None

Run: Normal window

Comment:

3. Add the following switches to the command in the **Target** box:

```
/user=<username>
/pass=<password>
/open=<project name>
```

Where `<project name>` is the name of the NWare *.npa* file without the `.npa` included. Here is an example:

```
"C:\Program Files\MediaMatrix\NWare 1.6.3\NWare.exe" /user=user1
/pass=passwd /open="nion project"
```

Note: If the project name contains spaces, you must enclose it in double quotes.

4. Click **OK**.

   When you start Kiosk using the new settings, the program will attempt to load the project you specified, then log on.

Starting Kiosk in full screen mode

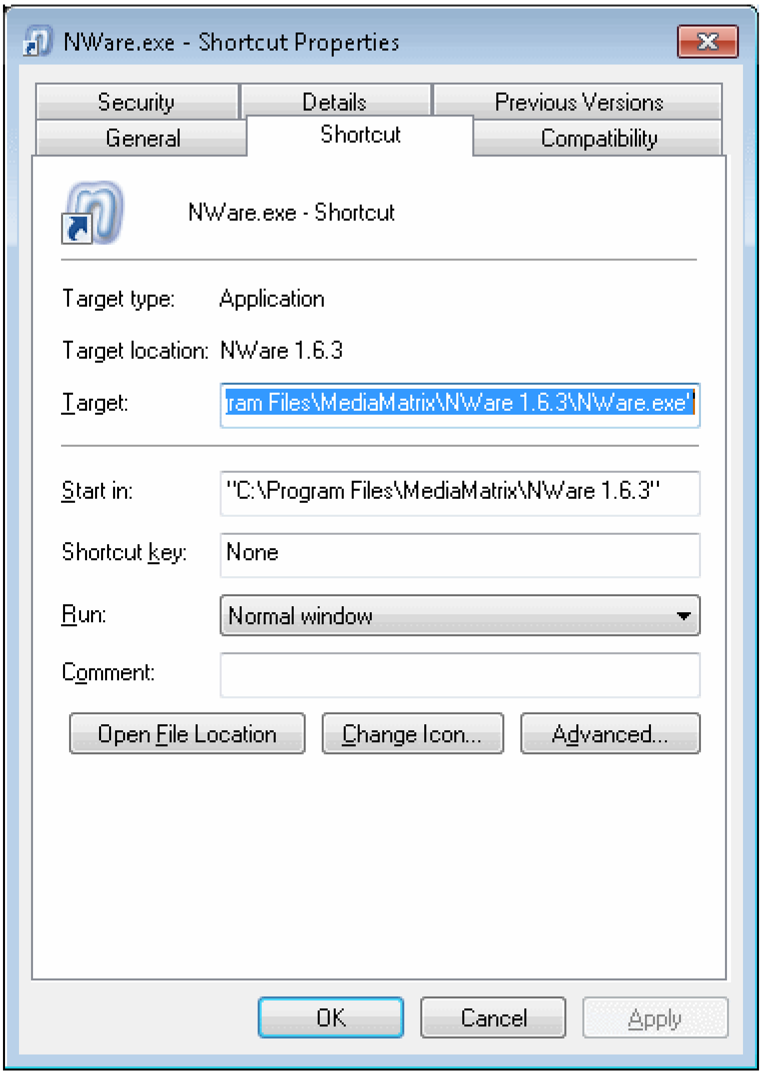
Normally, when you start Kiosk, the size of the application window is the same as it was when you last started the application. However, you can force it to be maximized at start up.

Tip: By default, when the Kiosk window is maximized, there is a small black border around the edge, above the status window. For information on hiding this border, see *Hiding the window border* (on page 218).

⏩ *To start Kiosk in full screen mode*

1. Open Windows Explorer and navigate to the folder containing the NWare Kiosk shortcut.
2. Right-click the shortcut, and then click **Properties**.

The **Shortcut to NWare.exe Properties** dialog box is displayed.



NWare.exe - Shortcut Properties

NWare.exe - Shortcut

Target type: Application

Target location: NWare 1.6.3

Target: ram Files\MediaMatrix\NWare 1.6.3\NWare.exe"

Start in: "C:\Program Files\MediaMatrix\NWare 1.6.3"

Shortcut key: None

Run: Normal window

Comment:

3. Add `/fullscreen=1` after `\NWare.exe"` in the **Target** box.

   Here is an example:

   ```
   "C:\Program Files\MediaMatrix\NWare 1.6.3\NWare.exe" /fullscreen=1
   /personality=nware_kiosk
   ```

4. Click **OK**.

Using nware_kiosk.xml

Hiding the window border

By default, when the Kiosk window is maximized, there is a small black border around the edge, above the status window. If pages in your project stretch to the very edge of the window, you may want to hide the border to ensure the entire area of the page is visible.

⏩ *To hide the window border*

1. Browse to *C:\Program Files\MediaMatrix\NWare [version]\personality*, and then edit the file *nware_kiosk.xml*.
2. Scroll down to the end of the file.

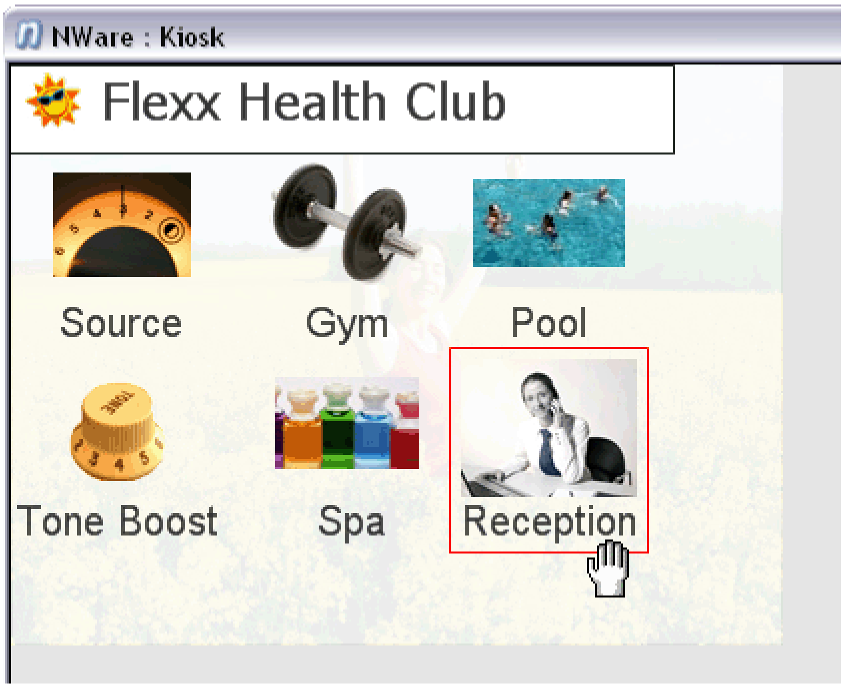
3. Just before the `</preferences>` tag, add this line:

   ```
   <item name="optimize_display_for_full_screen_kiosk" value="true" />
   ```

4. Save the file.

Hiding the selection rectangle

Normally, a selection rectangle is shown when you hover the mouse over a button.



To hide the selection rectangle

1. Browse to *C:\Program Files\MediaMatrix\NWare [version]\personality*, and then edit the file *nware_kiosk.xml*.
2. Scroll down to the end of the file.
3. Just before the `</preferences>` tag, add this line:

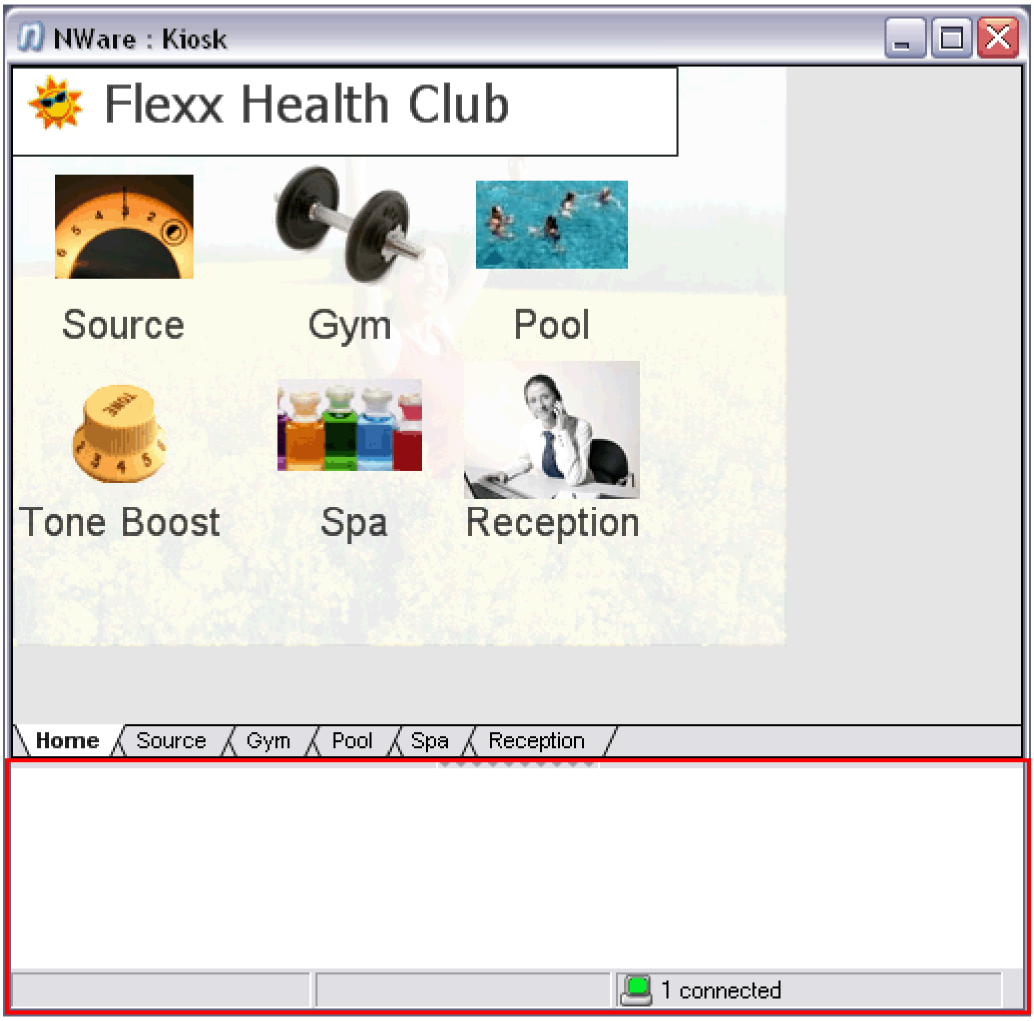
   ```
   <item name="skip_drawing_selection_rectangle" value="true" />
   ```

4. Save the file.

Hiding the status window

Normally, a status window is shown at the bottom of the Kiosk window. You can hide the status window to provide more space for the display of controls.

Tip: If you are running Kiosk with the status window hidden and an error occurs, a Command Prompt window will be displayed showing the error. If you want to prevent the window from being displayed, see *Hiding the Command Prompt window* (on page 221).



To hide the status window

1. Browse to *C:\Program Files\MediaMatrix\NWare [version]\plugins\nware\xml*, and then edit the file *kiosk_windows.xml*.

   The `window` blocks in this file control the display of the frames in the Kiosk window.

2. Add comment symbols (`<!--` and `-->`) to deactivate all the `window` blocks apart from the following section:

```
<windows>

 <window class="app::frame_t" title="NWare : Kiosk"
not_menu_xml="plugins/nware/xml/kiosk_menu.xml" id="main_frame"
icon="plugins/nware/icons/n-ware.ico">

 <window class="win::main_container_t" id="page_container" />

 </window>

</windows>
```

3. Save the file.

Hiding the Command Prompt window

If you are running Kiosk with the status window hidden and an error occurs, a Command Prompt window will be displayed showing the error. This can be distracting for Kiosk users.

⏩ *To hide the Command Prompt window*

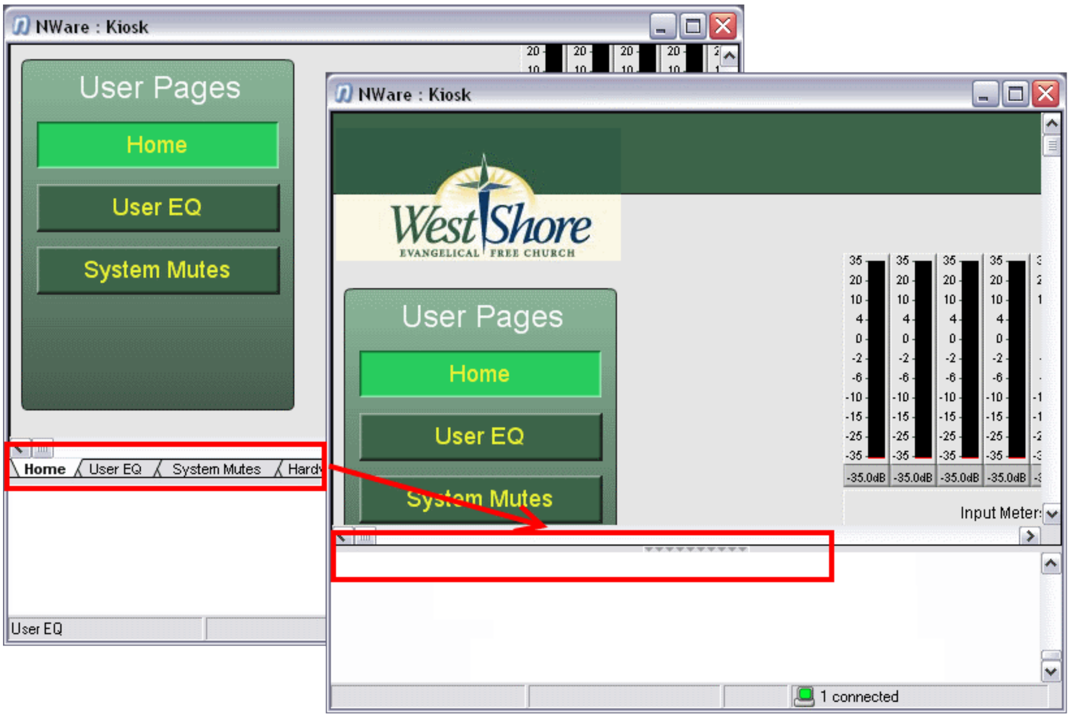
1. Browse to *C:\Program Files\MediaMatrix\NWare [version]\personality*, and then edit the file *nware_kiosk.xml*.
2. Scroll down to the end of the file.
3. Just before the `</preferences>` tag, add this line:

   ```
   <item name="write_console_message_on_comms_exception" value="false"/>
   ```

4. Save the file.

Hiding page tabs

You can hide the page tabs that normally appear at the bottom of the Kiosk window.



⏩ *To hide the page tabs*

1. Browse to *C:\Program Files\MediaMatrix\NWare [version]\personality*, and then edit the file *nware_kiosk.xml*.
2. Scroll down to the end of the file.
3. Just before the `</preferences>` tag, add this line:

   ```
   <item name="skip_drawing_page_tabs" value="true" />
   ```

4. Save the file.

Managing control gesturing

By default, you can drag the mouse to select multiple controls in Kiosk and gesture all of them at the same time. This is a convenient feature when you are running Kiosk on a PC. However, if you are running Kiosk on nTouch 180, it is sometimes preferable to disable this feature, so that users do not accidentally select multiple controls when they touch the screen.

⏩ *To prevent the selection of multiple controls*

1. Browse to *C:\Program Files\MediaMatrix\NWare [version]\personality*, and then edit the file *nware_kiosk.xml*.
2. Scroll down to the end of the file.
3. Just before the `</preferences>` tag, add this line:

   ```
   <item name="enable_marquee_selection" value="false" />
   ```

4. Save the file.

Using kiosk_preferences.xml

Showing or hiding tooltips (hover popups)

The display of tooltips in Kiosk is controlled by a setting in *kiosk_preferences.xml*. By default, tooltips are not shown.

Tip: When you *emulate* (on page 167) or deploy a project and then connect to it using NWare, by default, tooltips are displayed for the different controls. These can be switched on and off from the *User Preferences* dialog box. For more information, see *Showing and hiding tooltips (hover popups) on controls* (on page 273).

⏩ *To show or hide tooltips (hover popups)*

1. Browse to *C:\Program Files\MediaMatrix\NWare [version]\plugins\nware\xml*, and then edit the file *kiosk_preferences.xml*.
2. Change the *enable_hover* setting to true to show tooltips or false to hide tooltips.

   Here is an example showing the setting that will show tooltips.

   ```
   <item name="enable_hover" value="true" />
   ```

3. Save the file and close it.

   Note: It is important not to leave the file open in the editor, even if you have saved it. If you leave the file open, when Kiosk is restarted, the setting will be automatically changed back to *false*.

Using a personality file

You can use an XML file, called a *personality* file, to control the availability of certain UI features in Kiosk, and also for automatically logging on to Kiosk when it is started. The personality file is specified as part of the shortcut for Kiosk.

Note: Before using a personality file, you must create a shortcut to run NWare Kiosk. The shortcut created automatically during installation *cannot* be used. For more information, see *Creating a new shortcut for NWare Kiosk* (on page 215).

⏩ *To create the personality file*

1. Open a text editor and create a new file.
2. Add a `<kiosk>` block to the file and specify tags within the block to control different Kiosk UI features.

   Here is an example showing all the tags you can use. You can base your file on this example and then customize it using information in the later sections.

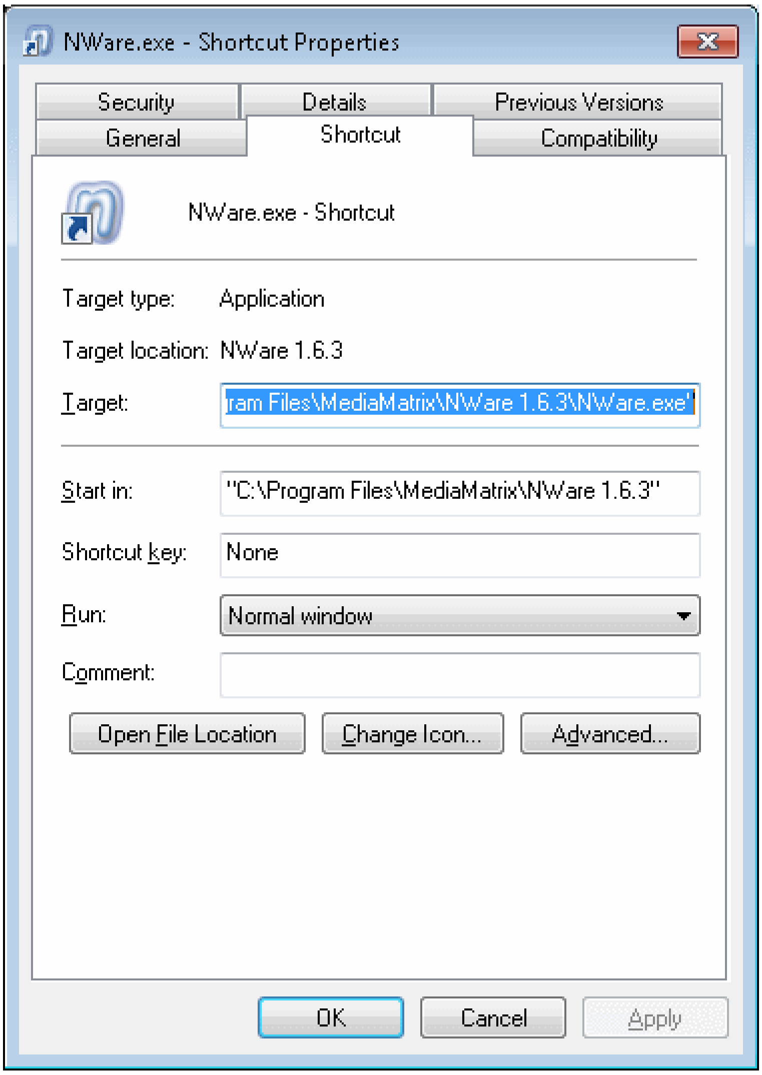
```
<kiosk>

    <name value="kiosk_test" />

    <username value="user1" />

    <hash value="" /> <!-- can use prehashed password if you don't want to
store the password in plain text in this file -->

    <password value="mypass" />

    <hide_root_tab value="true" /> <!-- defaults to false -->

    <hide_cursors value="true" /> <!-- defaults to false -->

</kiosk>
```

3. Save the file with the extension *.XML* and make a note of its name and location.

⏩ *To specify the personality file to use with Kiosk*

1. Open Windows Explorer and navigate to the folder containing the NWare Kiosk shortcut.
2. Right-click the shortcut, and then click **Properties**.

The **Shortcut to NWare.exe Properties** dialog box is displayed.



Target type: Application

Target location: NWare 1.6.3

Target: ram Files\MediaMatrix\NWare 1.6.3\NWare.exe"

Start in: "C:\Program Files\MediaMatrix\NWare 1.6.3"

Shortcut key: None

Run: Normal window

Comment:

3. Add the path and name of the personality XML file to the end of the entry in the **Target** box.

   If you are running the full version of NWare, the entry will look like this:

   ```
   "C:\Program Files\MediaMatrix\NWare 1.6.3\NWare.exe"
   /personality=nware_kiosk my_personality_file.xml
   ```

   If you are running the special Kiosk only version of NWare, the entry will look like this:

   ```
   "C:\Program Files\MediaMatrix\NWare 1.6.3\NWare.exe"
   my_personality_file.xml
   ```

Specifying the basic project settings

1. Add a line to the Kiosk XML file to specify the name of the project *.npa* file (without the *.npa* extension). Here is an example:

   ```
   <name value="kiosk_test" />
   ```

2. If you want to log on automatically using a username and password, specify these settings:

   ```
   <username value="user1" />
   ```

```
<password value="passwd1" />
```

3. Save the personality file.

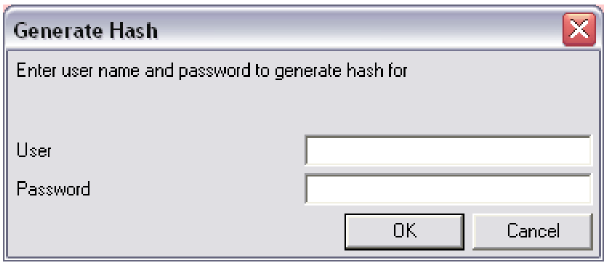
Hiding the password using hashing

The XML personality file is a text file that can be opened in any text editor, such as Windows Notepad. This means it is possible for a user to locate the file and read the password. To protect against this happening, you can encrypt the password using hashing, and specify the encrypted version in the personality file, instead of the original version.

To hide the password using hashing

1. In NWare, press CTRL+SHIFT+H.

   The **Generate Hash** dialog box is displayed.



Generate Hash

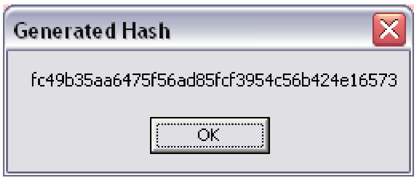
Enter user name and password to generate hash for

User

Password

2. In the **User** box, type the username.
3. In the **Password** box, type the password.
4. Click **OK**.

   The **Generated Hash** dialog box is displayed.



Generated Hash

fc49b35aa6475f56ad85fcf3954c56b424e16573

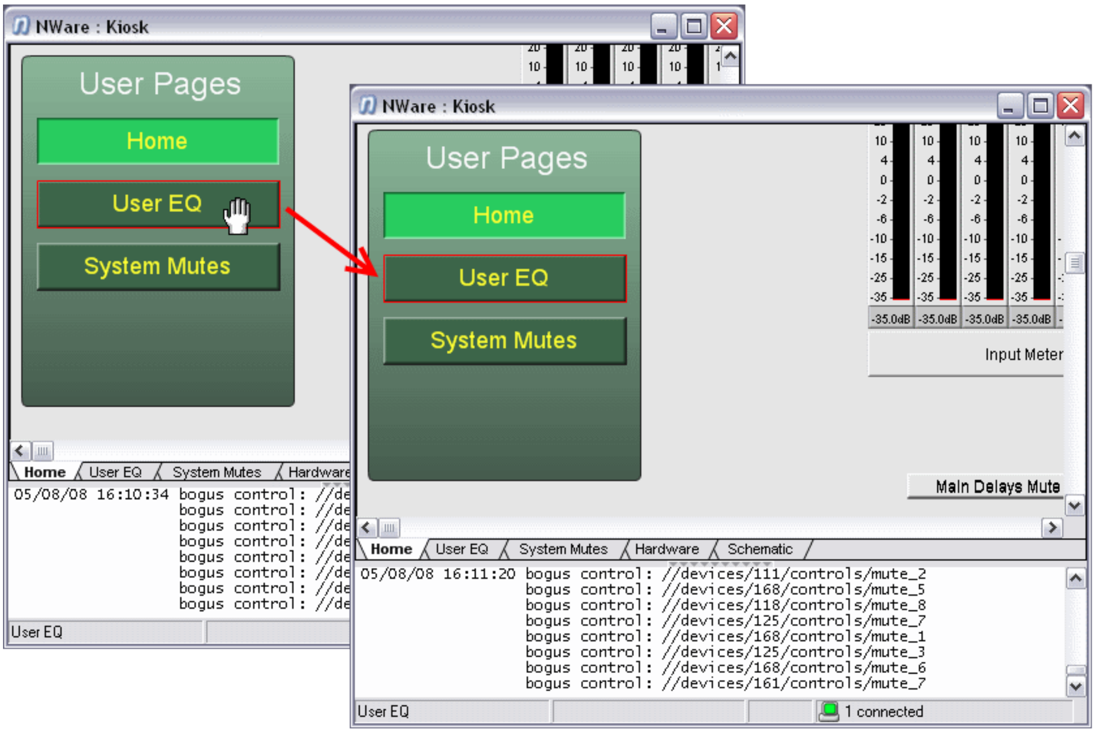
5. Make a note of the hash value.
6. Edit the personality file and remove the `<password value= ... />` line.
7. Add in a new line to specify the hash value. For example:

```
<hash value="fc49b35aa6475f56ad85fcf3954c56b424e16573" />
```

8. Save the personality file.

Hiding the cursor

You can hide the cursor that normally appears when you move the mouse pointer over the Kiosk window.



```
05/08/08 16:10:34 bogus control: //de
                  bogus control: //de
                  bogus control: //de
                  bogus control: //de
                  bogus control: //de
                  bogus control: //de
                  bogus control: //de
                  bogus control: //de
```

```
05/08/08 16:11:20 bogus control: //devices/111/controls/mute_2
                  bogus control: //devices/168/controls/mute_5
                  bogus control: //devices/118/controls/mute_8
                  bogus control: //devices/125/controls/mute_7
                  bogus control: //devices/168/controls/mute_1
                  bogus control: //devices/125/controls/mute_3
                  bogus control: //devices/168/controls/mute_6
                  bogus control: //devices/161/controls/mute_7
```

⏩ *To hide the cursor*

1. Edit the personality file and add the following line:

   ```
   <hide_cursors value = "true" />
   ```

2. Save the personality file.

Starting Kiosk in full screen mode

Normally, when you start Kiosk, the size of the application window is the same as it was when you last started the application. However, you can force it to be maximized at start up.

Tip: By default, when the Kiosk window is maximized, there is a small black border around the edge, above the status window. For information on hiding this border, see *Hiding the window border* (on page 218).

⏩ *To start Kiosk in full screen mode*

1. Browse to *C:\Program Files\MediaMatrix\NWare [version]\personality*, and then edit the file *nware_kiosk.xml*.
2. Scroll down to the end of the file.
3. Just before the `</preferences>` tag, add this line:

   ```
   <item name="full_screen_mode" value="true" />
   ```

4. Save the file.

Troubleshooting

This section contains descriptions of some problems you could run into while using Kiosk, and some suggested solutions.

No projects are listed on the Select Project dialog box when Kiosk starts

Pandad has crashed.

When Pandad stops functioning, it can be recovered without shutting down NWare.

⏩ *To recover Pandad after it has stopped functioning*

1. On the control computer, press CTRL+ALT+DEL to open Windows Task Manager.
2. Click the **Processes** tab.
3. In the **Image Name** column, click **pandad.exe**.

Tip: You can click on the column header to sort all entries alphabetically.

4. Click **End Process** to close *pandad.exe*.
5. Close or minimize Task Manager.
6. Open Windows Explorer, then navigate to the root of the version of NWare you are using (the root is where you find *NWare.exe*).
7. Double-click *pandad.exe* to start Pandad in Interactive Mode.

   A Command Prompt window will open.
8. Open Windows Task Manager and check that *pandad.exe* is listed.
9. Open NWare and check that all the nodes are listed on the **Remote Log** tab.

   It may take a few seconds for the nodes to appear in the list.

If a deploy operation or upload and connect operation was interrupted when Pandad stopped working, it will now continue.

The project you want to use is not listed on the Select Project dialog box when Kiosk starts, but other projects are

Check that you can connect to the node that is hosting the project (NioNode, nControl node or nTouch 180 node).

Check that the project has been deployed to the node you are connected to.

I cannot see a particular page I expected to see in Kiosk

Check that the page has been exported from the project.

Check that the page has been exported to a page group that you are authorized to access.

My user account does not allow me access to Kiosk

Check that the designer has added the page group (to which the exported pages belong) to your user account settings.

Check that the designer has selected **Allowed** next to **Network Control Access** in your user account settings.

Chapter 9

Setting up Kiosk2Go

In This Chapter

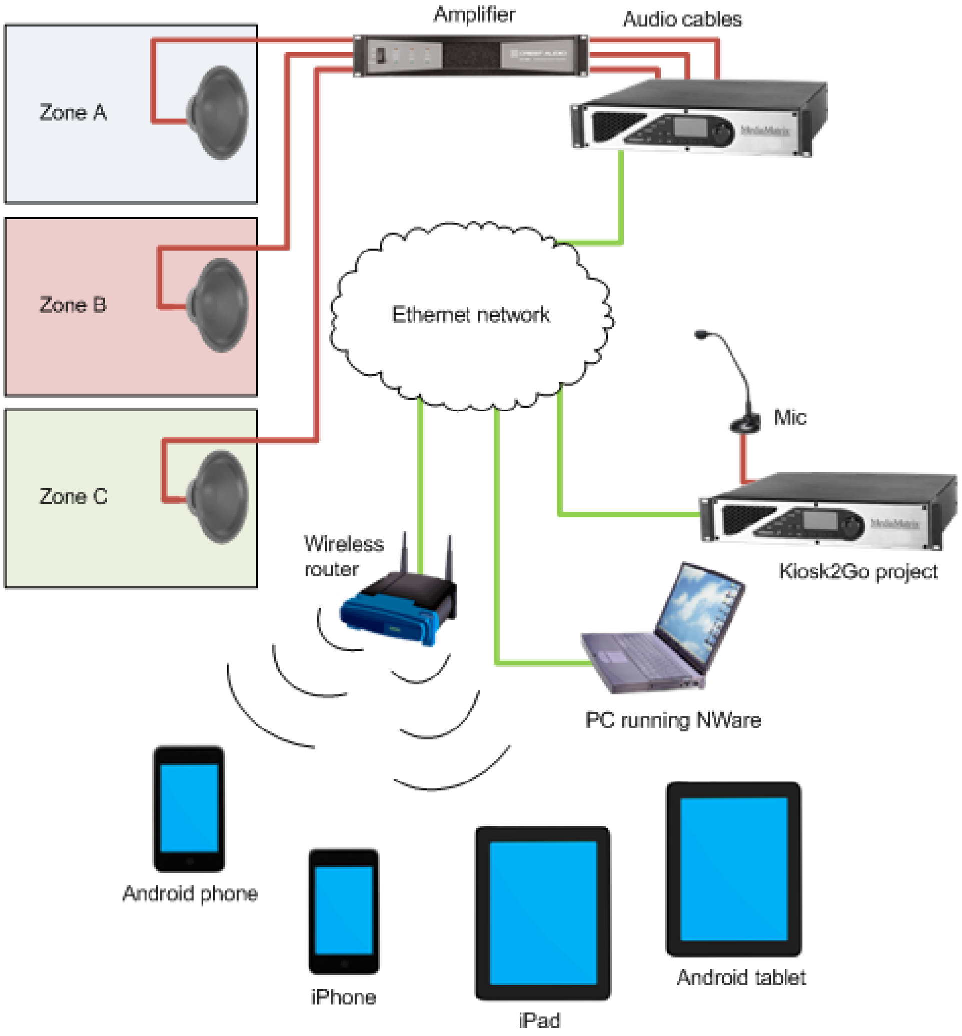
Introduction

Kiosk2Go is a web-based interface for controlling NWare projects. It provides similar functionality to Kiosk, but is optimized for both touch devices, like iPads, iPhones and Android tablets, and PC-based browsers.

You can run Kiosk2Go on a NION, nControl or nTouch 180, and you can also use it with an emulated project.

Note: If you are intending to use a NION to host a Kiosk2Go project, it is important that the compact flash (CF) memory card is of sufficient capacity. If the NION is fitted with a 256MB card, you may find problems with the display of more complex Kiosk2Go projects in the browser. We recommend that the compact flash card is replaced with a higher capacity version. To check the capacity of the CF memory card, navigate to the NION web interface, then click *Hardware*. The capacity is displayed under *Memory*, as *CFA: xxx meg*. For more information, contact *MediaMatrix Technical Support* (*mailto:mmtechsupport@peavey.com*).

A web browser running on each mobile device connects to the web server on the MediaMatrix node over the wireless network. Web pages generated from the Kiosk2Go NWare project are loaded onto the mobile devices for display on the screen. Controls on the web pages communicate with controls in the NWare project using RATC.



To use a set up like the one in the example, your NWare project will need to contain a *Kiosk2Go Layout* device, which is configured with the controls you want to make available to users of the mobile devices. These can be knobs, faders, buttons etc. arranged on different pages, just as they are in Kiosk. You will also need at least one NioNode, nTouch 180 node or nControl node and, if you require all the features of the above example project, an amplifier device.

The Kiosk2Go sections in the NWare User Guide will explain how to add the device to your design and configure it, but if you require only reference information, see *Kiosk2Go Layouts* in the *NWare Device Reference*.

Touch device browser support

Kiosk2Go is supported by a range of PC browsers, as well as mobile device browsers, including iOS Safari, Chrome and Firefox. It is important to note, however, that the browser you intend to use must provide support for web sockets; this technology is used for communication between the mobile browser and the MediaMatrix node, and is a fundamental requirement of Kiosk2Go.

For a full list of operating systems and browsers and the support they provide for web sockets, see *http://caniuse.com/#search=websocket* (*http://caniuse.com/#search=websocket*).

Browser font support

NWare allows you to specify a number of different fonts for objects that will appear in Kiosk2Go. For example, when you are adding a label, the Control Properties dialog box will show a list of fonts that may include Arial, Arial Black, Arial Narrow etc., depending on what is installed on your PC. The browser that displays the Kiosk2Go screens may support only a subset of the fonts available in NWare, so it is important to check that the fonts you intend to use will display correctly.

You can check to see if fonts are supported by visiting *this website* (*http://www.lalit.org/lab/javascript-css-font-detect/*) using the device which will run Kiosk2Go.

Adding a Kiosk2Go Layout device

A Kiosk2Go Layout device stores the pages that will be displayed in your mobile device browser or PC browser.

There are lots of individual smartphone layouts and tablet layouts under *Graphic Elements & Blocks / Kiosk2Go Layouts* in the device tree. Each is preset with the screen dimensions of the particular device after which it is named. When you drag the device over to the design page and create it, you will be able to see exactly how much screen space you have available for buttons, sliders and other controls.

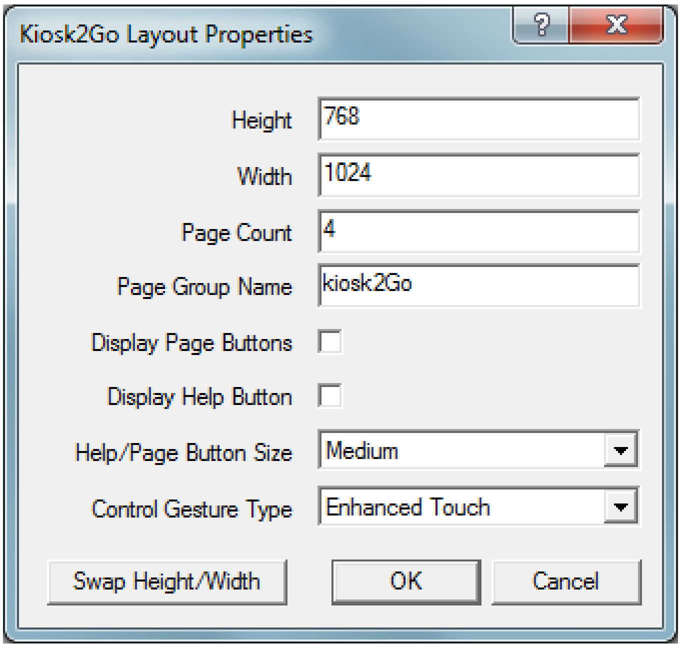
When you add a Kiosk2Go Layout device to your design, you assign it a *page group name*. By default, this is the name of the device, e.g. HTC Thunderbolt. Once you have set up the device, you grant the user, who will log on to Kiosk2Go in the browser, access to this group. If you add several Kiosk2Go Layout devices to the project, provided they have been granted access to the page groups, the user will be asked to choose a page group name before they can access the pages it contains.

⏩ *To add a Kiosk2Go Layout device*

1. In the device tree, expand the **Graphic Elements & Blocks / Kiosk2Go Layouts** folder.
2. If you are designing screens for a specific mobile device, e.g., an Apple iPhone 5, expand the subfolders, and then drag the device over to the design page.

   If the device you are using is not listed, drag a **Generic** device over to the design page.

   The **Kiosk2Go Layout Properties** dialog box is displayed.


Kiosk2Go Layout Properties

Height: 768
Width: 1024
Page Count: 4
Page Group Name: kiosk2Go
Display Page Buttons: □
Display Help Button: □
Help/Page Button Size: Medium
Control Gesture Type: Enhanced Touch

3. If you want to change the dimensions of the screen design area, type new values into the **Height** and **Width** boxes.

   The screen design area is the part of the Kiosk2Go device visible on the screen when Kiosk2Go is running.

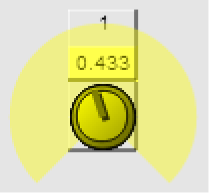
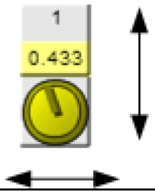
Tip: Specifying a height that is larger than the height of the device screen will allow the user to scroll down to see more controls on the page.

4. If you want to swap the height and width settings over to change the layout between landscape and portrait, click **Swap Height/Width**.
5. In the **Page Count** box, type the number of individual design pages to be displayed on the device when it is created.
6. In the **Page Group Name** box, type the name of the page group that will contain the pages in the Kiosk2Go device.

   This name will be displayed as the title at the top of the browser screen running Kiosk2Go.
7. If you want to allow the user to move to the previous page and next page in the browser by tapping buttons, select the **Display Page Buttons** check box.
8. If you want to allow the user to view help that you have written for controls in the design, select the **Display Help Button** check box.

Tip: The help text is specified for each individual control on the Control Properties / Help tab.

9. If you want to specify smaller or larger help and page buttons, click a new size in the **Help/Page Button Size** list.
10. In the **Control Gesture Type** list, click an option to specify how the user will interact with knob controls in Kiosk2Go.

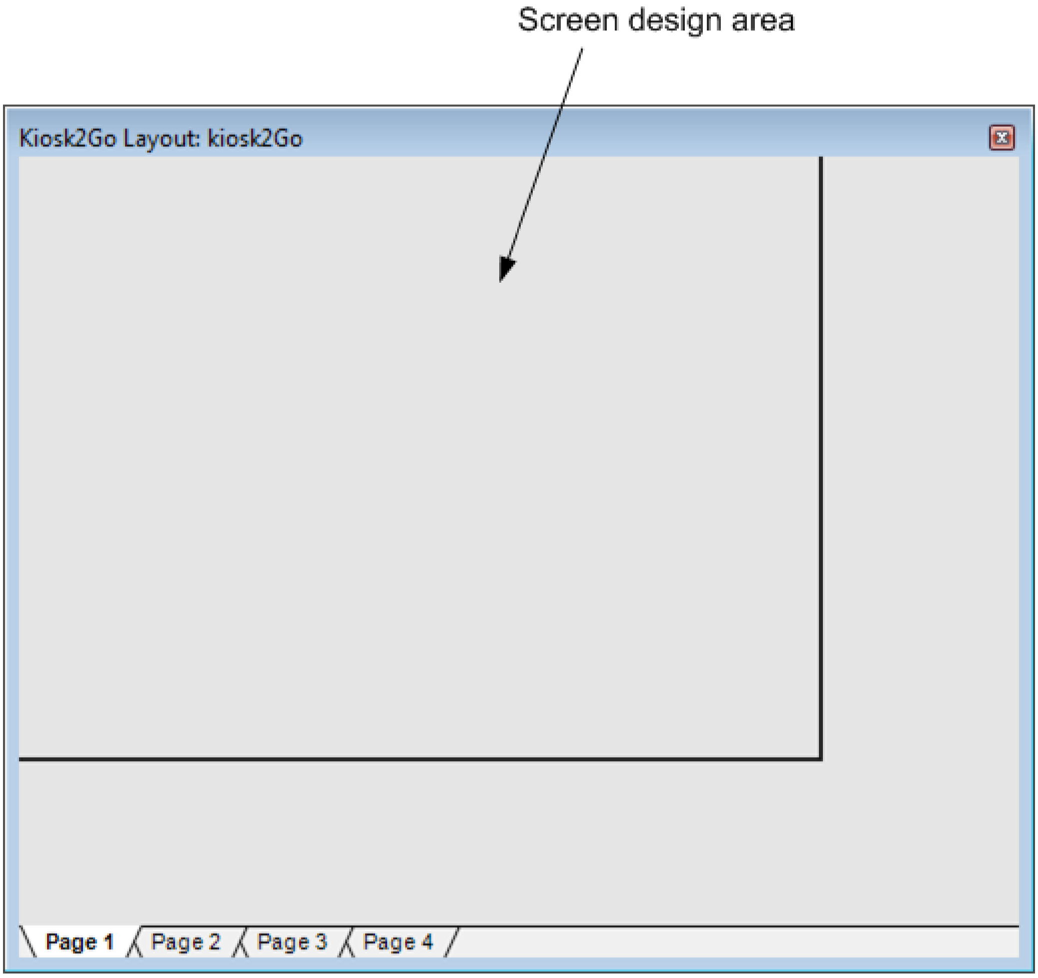
Enhanced Touch	Click/tap the center of the knob to change to gesture mode. A colored gesture area will be displayed around the knob control.  Drag inside this area to gesture the knob and change its value. Click/tap the center of the knob to exit gesture mode.
NWare Compatible	Similar to the style used in regular Kiosk. Tap/hold the center of the knob and then drag to gesture. 
Auto Select	A mode will be selected automatically, based on the type of device running Kiosk2Go.

11. Click **OK**.

Understanding the layout of the device

Once you have added the Kiosk2Go Layout device to your design, you can start to add controls to the individual pages, so end users can interact with them.

When you open the Kiosk2Go Layout device, you will see that NWare has created a number of individual pages automatically.



Each page labeled *Page 1*, *Page 2*, and so on, shows the area that will be visible on the mobile device screen – the *screen design area* – at the top. Any controls you place in the screen design area will be visible on the mobile device screen when the project is running.

Notes:

- Do not lock pages within the device or the page on which the device is located in your design. You will not be able to access the locked pages from the mobile device.
- We recommend that you place nTouch 60 devices, Kiosk Layout blocks, Kiosk2Go Layout blocks and Mobile Display blocks on a separate page to the other devices in the project, and do not export the page.

Adding controls to the Kiosk2Go Layout device

Controls added to the design area of the Kiosk2Go Layout device will be displayed on the mobile device running Kiosk2Go. NWare controls created using the Generic Controls device are supported. These include knobs, meters, buttons, and so on. Some other controls are also supported. For a full list, see *NWare Controls supported by Kiosk2Go* (on page 238).

Kiosk2Go does not display wiring between controls. Therefore, you will need to wire a control on one of the main NWare design pages, so that the control performs the intended function, and then copy and paste the control into the Kiosk2Go block. The user will then see the copied control on their mobile device, and not how it is wired to the rest of the project.

Notes:

- You cannot drag a generic Generic Controls device from the device tree directly onto a page inside a Kiosk2Go block. You must drag the device onto the design page, and then copy and paste it into the Kiosk2Go block (see below).
- Kiosk2Go controls use the RATC2 protocol, which requires a control alias to be assigned to each control. This is done automatically, by default, but can be done manually if preferred. For more information, see *How control aliases are used in Kiosk2Go* (on page 250).
- Use separate paging buttons for Kiosk2Go and regular Kiosk pages. If you copy and paste paging buttons between Kiosk2Go Layout blocks and pages for display in regular Kiosk, when a button is clicked/tapped, it will change the page for both Kiosk2Go users and regular Kiosk users at the same time.

⏩ *To add controls to the Kiosk2Go Layout device*

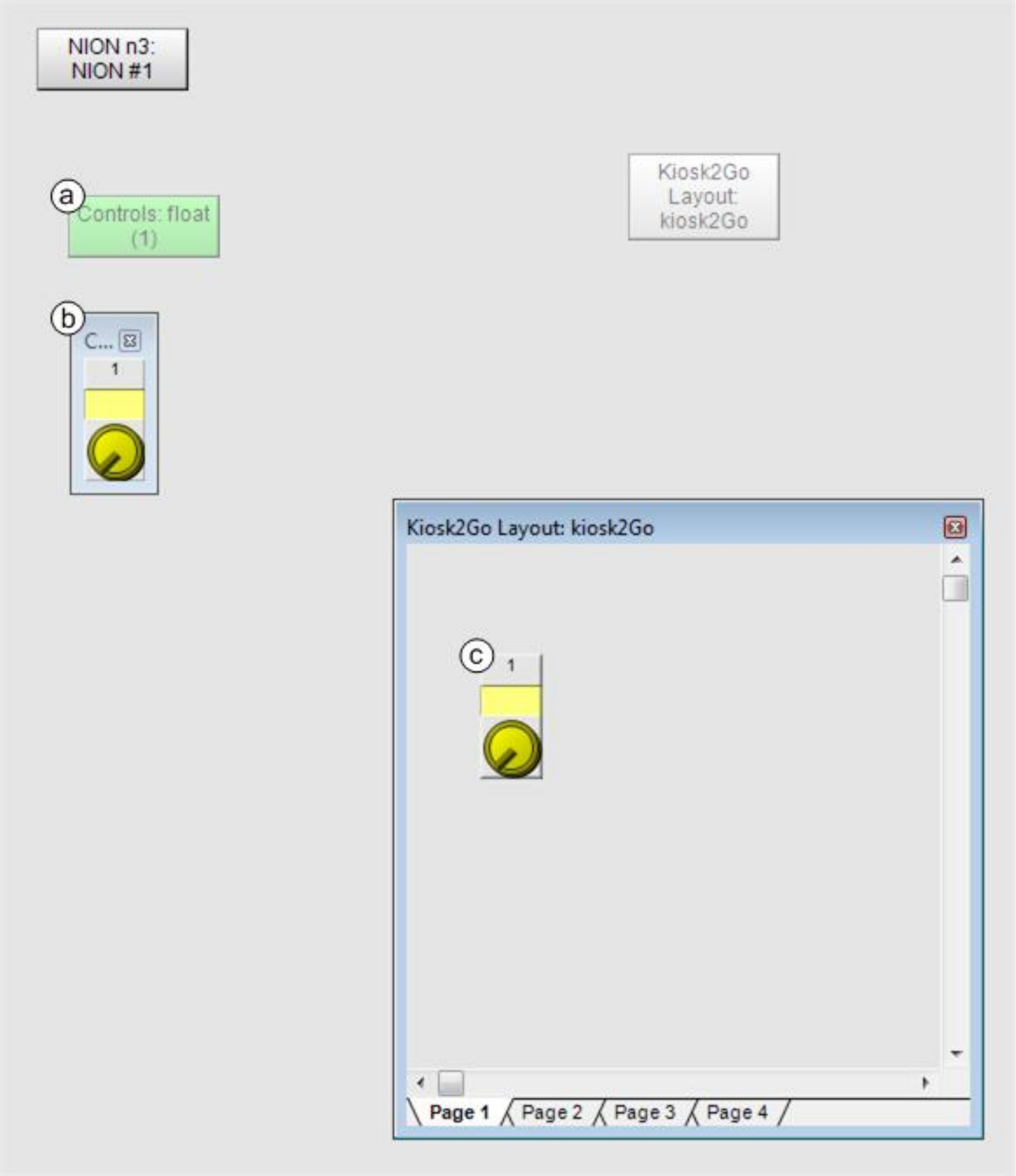
1. Open the **Kiosk2Go Layout** block and click the page where you want to add the control.
2. If you want to add a control that already exists in the project, copy and paste the control onto the Kiosk2Go Layout device page.

   If you want to add a new control from the device tree:

   a) Drag the control onto the main NWare design page (a).

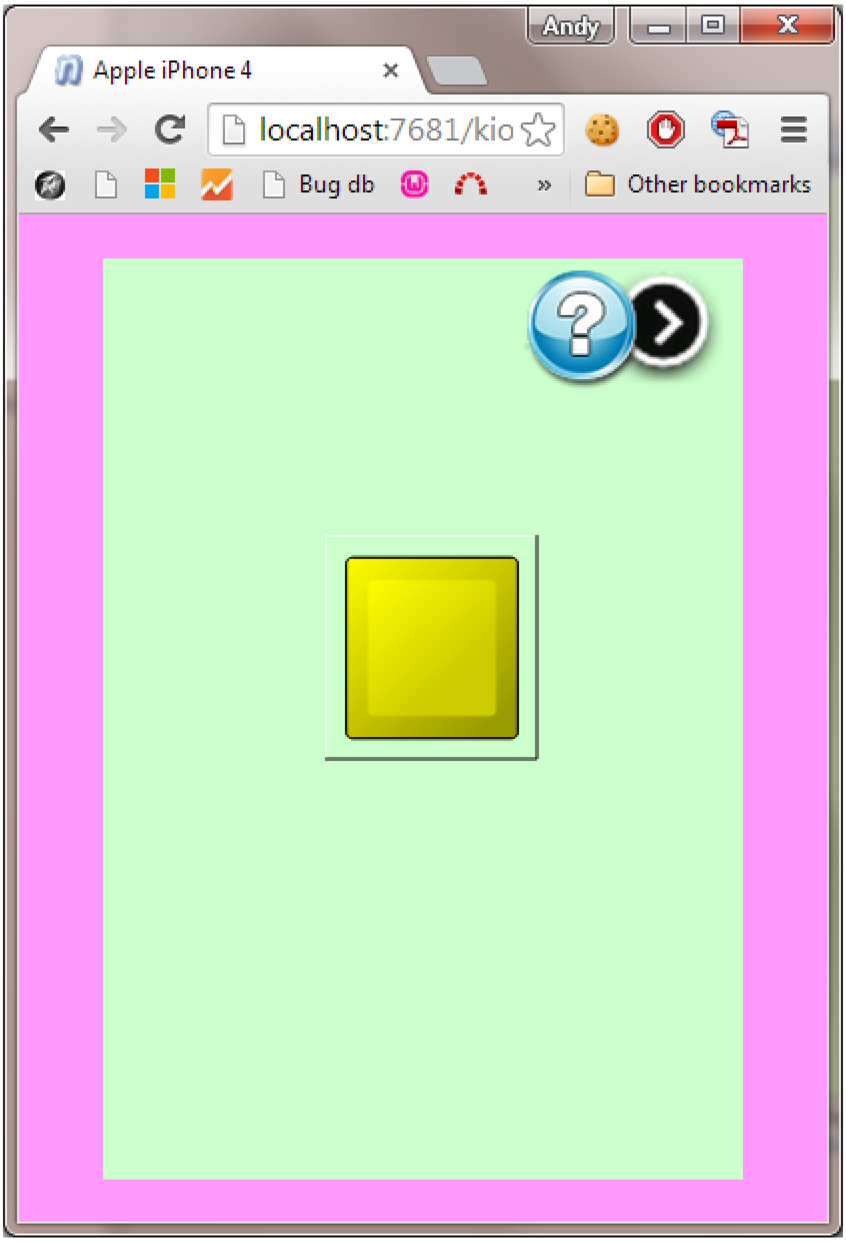
   b) Open the control block (b).

c) Copy and paste the control onto the Kiosk2Go Layout device page (c).



Adding a background color to the Kiosk2Go device

You can specify a color to display for each individual page in the Kiosk2Go device block (green in the example below), as well as the area surrounding the Kiosk2Go device in the browser (pink in the example below).



⏩ *Adding a background color to an individual page*

1. Open the Kiosk2Go block.
2. Right-click a page tab at the bottom, and then click **Properties**.
3. Next to **Color**, click the color sample, and then a click a color hue and intensity.
4. Click **OK**.
5. Click **OK**.

⏩ *Adding a background color to the area outside the Kiosk2Go device in the browser*

1. Right-click the Kiosk2Go device block, and then click **Block Properties**.
2. Next to **Color**, click the color sample, and then a click a color hue and intensity.
3. Click **OK**.
4. Click **OK**.

NWare controls supported by Kiosk2Go

The following controls are supported by Kiosk2Go:

- Generic Controls: all subcontrol types (knob, fader etc. and options), apart from *touchxy* and *text box - externally editable*.
- 2D Control.
- Labels: all types.
- Bump panels.

Grouping of controls is also supported.

Note: File names used with image controls are case-sensitive.

Adding a combo box with page links

When you run Kiosk2Go, you can change between pages one page at a time by tapping the paging buttons ❮ ❯. Alternatively, you can add a combo box to the pages that allows the user to jump directly to a particular page.

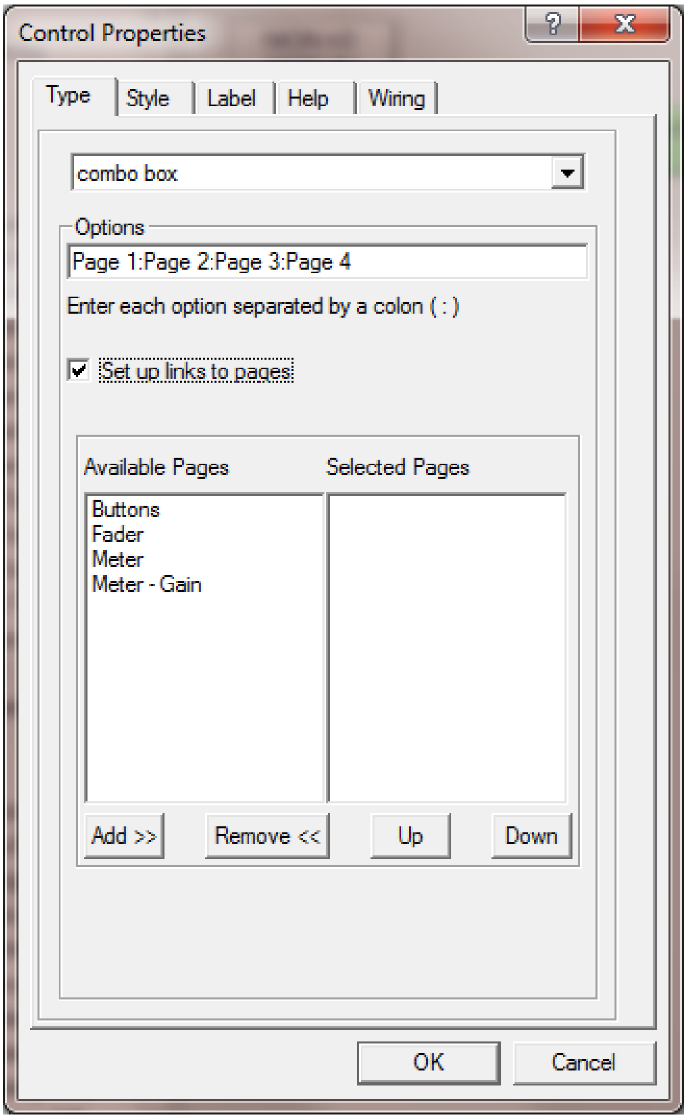
Notes:

- If you rename Kiosk2Go pages (by double-clicking the page tabs inside the Kiosk2Go device and typing new names), you will need to reselect the pages for each paging combo box.
- Page change buttons and page change combo boxes can only be controlled from the browser running Kiosk2Go. You cannot control these devices remotely by wiring other devices to them in NWare or using other methods, such as Python or RATC. This is because there is no way to differentiate between web client instances.

To add a combo box with page links

1. Drag a **Generic Controls** device from the device tree over to the main NWare design page.
2. Specify **string** in the **Type of control(s)** list.
3. Open the **Kiosk2Go Layout block** and click the page where you want to add the combo box.
4. Open the **Generic Controls** control block, and then copy and paste the control onto the **Kiosk2Go Layout** device page.
5. Right-click the control, and then click **Control Properties**.
6. In the list, click **combo box**.
7. Select the **Set up links to pages** check box.

The Available Pages and Selected Pages lists are displayed. The Selected Pages list shows the pages that will appear in the combo box.



combo box

Options

Page 1:Page 2:Page 3:Page 4

Enter each option separated by a colon (:)

☑ Set up links to pages

Available Pages

Buttons
Fader
Meter
Meter - Gain

Selected Pages

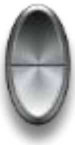
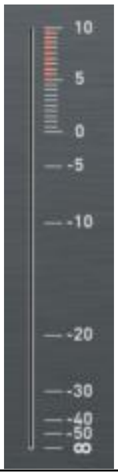
8. In the **Available Pages** list, double-click individual pages to move them to the **Selected Pages** list.
9. If you want to remove a page from the **Selected Pages** list, click the page to select it, and then click **Remove**.
10. If you want to change the order in which the pages will appear in the combo box, click a page to select it, and then click **Up** or **Down**.
11. Click **OK**.
12. Copy and paste the paging combo box onto all the pages in the Kiosk2Go Layout device where you want a list of pages to be shown.

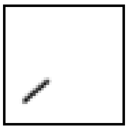
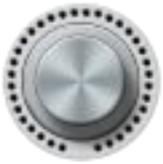
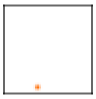
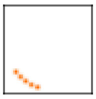
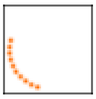
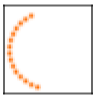
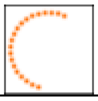
Skinning controls to improve their appearance

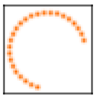
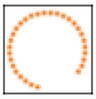
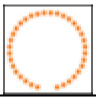
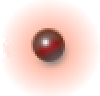
You can enhance the appearance of faders, knobs, meters and LEDs by adding graphics to them. Altering the appearance will help a control fit into a color scheme that you have devised for your Kiosk2Go project.

Control graphics

Below is a table showing the graphics you will need for the different types of controls, along with some examples.

Control	Graphics required	
Fader	Fader handle 	Background 

Knob	Value pointer (rotates as value changes) 	Background image (fixed) 	Image series (changes automatically as value increases)     

			  
Meter	Background image 		
LED	On image 	Off image 	

Specifying the control graphics

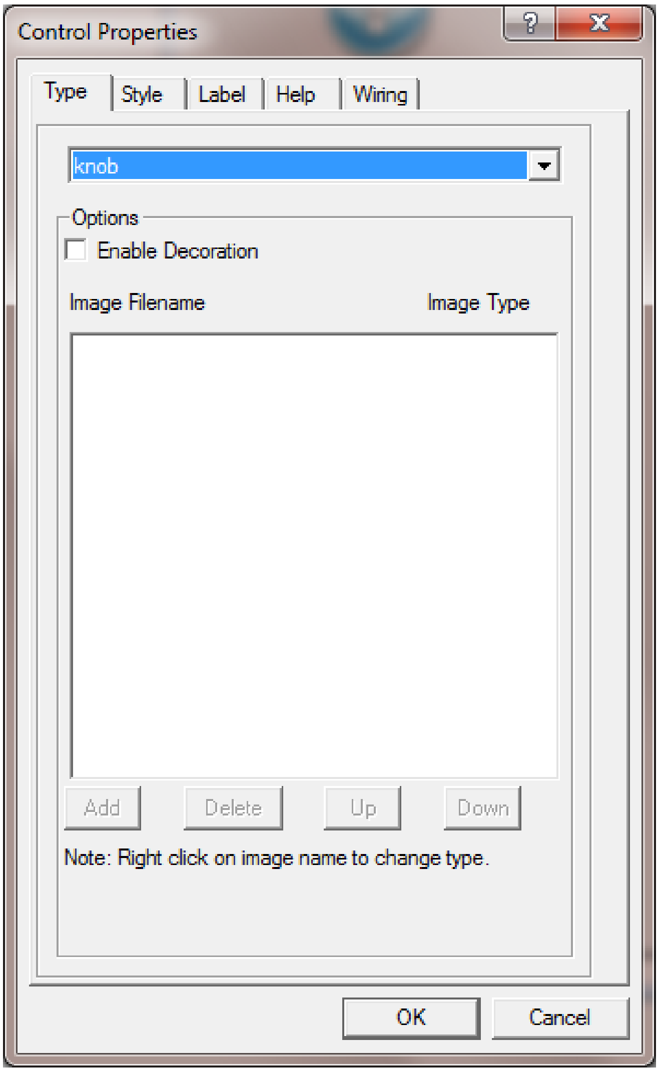
You will need to enable decoration for a control, described in the first procedure below, and then complete one of the other procedures for the type of control you are configuring.

This section assumes you have already imported the images you want to use via the Media tab. For more information, see *Importing files using the Media tab* (on page 45).

To enable decoration for a control

1. Right-click the control, and then click **Control Properties**.

The **Control Properties** dialog box is displayed.



2. On the **Type** tab, under **Options**, select the **Enable Decoration** check box.

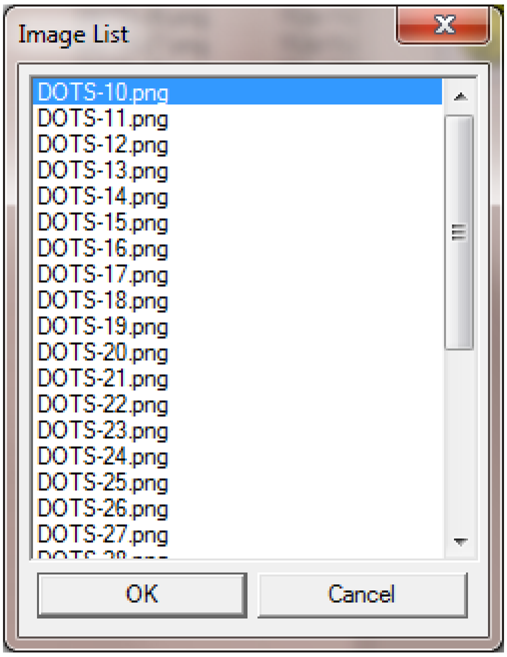
⏩ *To skin a fader*

1. In the **Slider Image** list, click an image for the fader handle.
2. In the **Background Image** list, click an image for the area immediately surrounding the fader.
3. If you want the fader handle to stay the same size when the fader (and background image) is resized on the design page:
   a) Click the **Style** tab.
   b) Under **Advanced Styles**, click **Configure**.
   c) Select the **Don't scale slider image** check box.

⏩ *To skin a knob*

1. Click **Add**.

The **Image List** dialog box will be displayed.



2. Click images for the value pointer, background and image series.

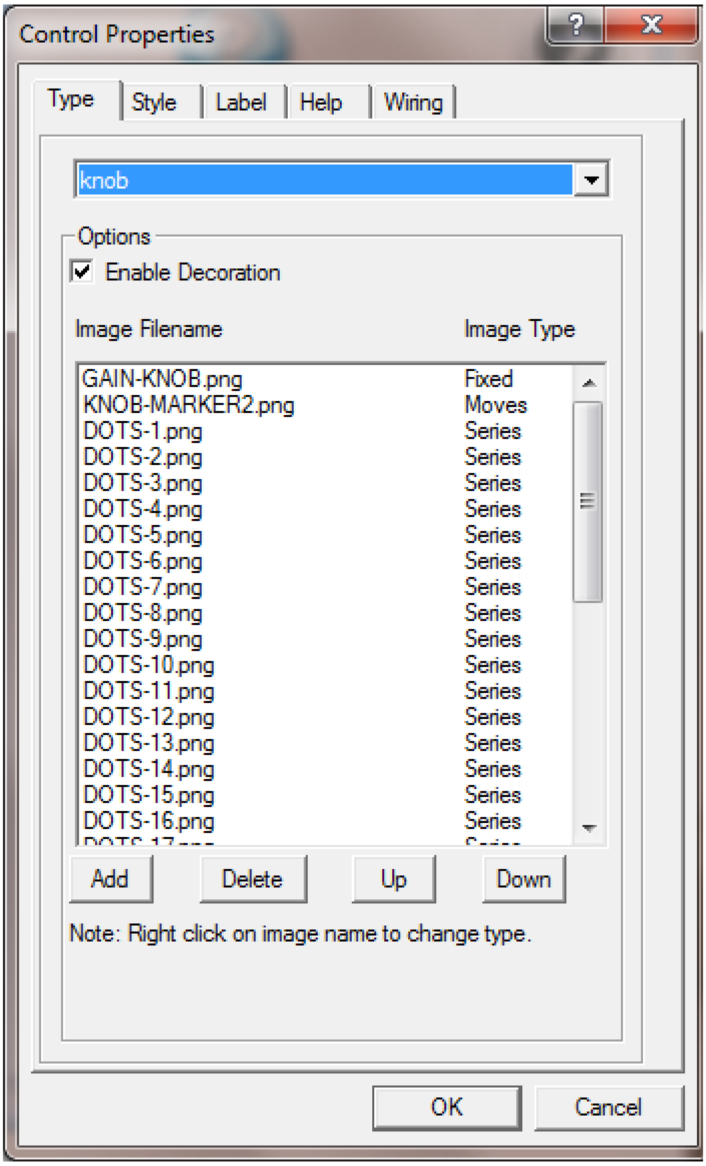
   The value pointer image will be rotated as the value changes. It should point to the minimum value position.

   The background image will remain stationary as the knob value changes.

   The image series will be a number of images that are displayed in turn as the value of the knob increases or decreases.

3. Click **OK**.
4. Next the value pointer image, right-click in the **Image Type** list, and then click **Moves**.
5. Next to the background image, right-click in the **Image Type** list, and then click **Fixed**.
6. Next to the images you want to display as the knob turns, right-click in the **Image Type** list, and then click **Series**.
7. If you want to move an image up or down, click the image, and then click **Up** or **Down**.

Here is an example showing settings for a knob control. There is a fixed image for the background, a pointer image that rotates, and a series of image that are displayed one-by-one as the knob value changes.



knob

Options

☑ Enable Decoration

Image Filename	Image Type
GAIN-KNOB.png	Fixed
KNOB-MARKER2.png	Moves
DOTS-1.png	Series
DOTS-2.png	Series
DOTS-3.png	Series
DOTS-4.png	Series
DOTS-5.png	Series
DOTS-6.png	Series
DOTS-7.png	Series
DOTS-8.png	Series
DOTS-9.png	Series
DOTS-10.png	Series
DOTS-11.png	Series
DOTS-12.png	Series
DOTS-13.png	Series
DOTS-14.png	Series
DOTS-15.png	Series
DOTS-16.png	Series
DOTS-17.png	Series

Note: Right click on image name to change type.

⏩ *To skin a meter*

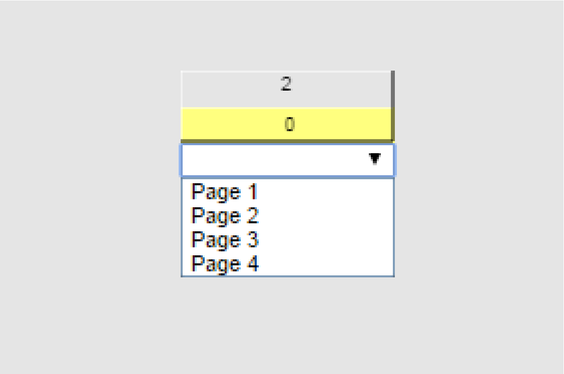
- In the **Meter Image** list, click an image for the area immediately surrounding the meter.

⏩ *To skin an LED*

1. In the **On Image** list, click an image for the LED when it is in the *on* state.
2. In the **Off Image** list, click an image for the LED when it is in the *off* state.

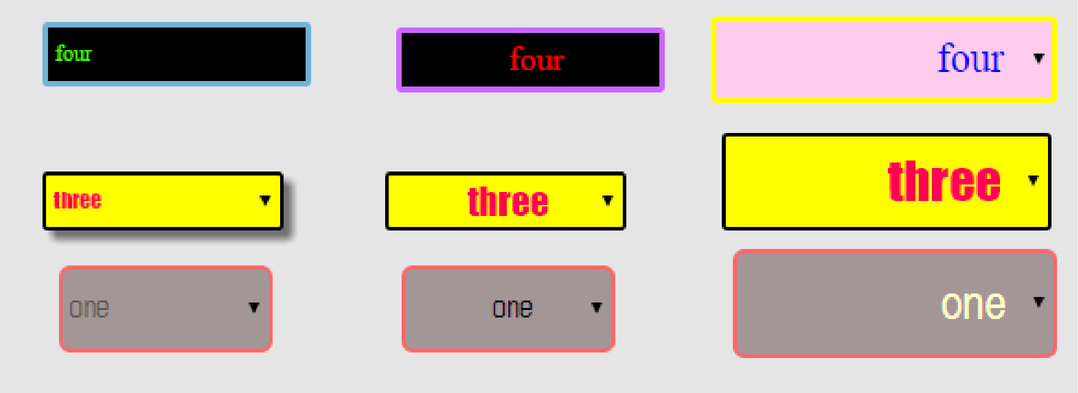
Customizing the appearance of a combo box

By default, a combo box, which is used to present a list of options, looks like this in a browser:



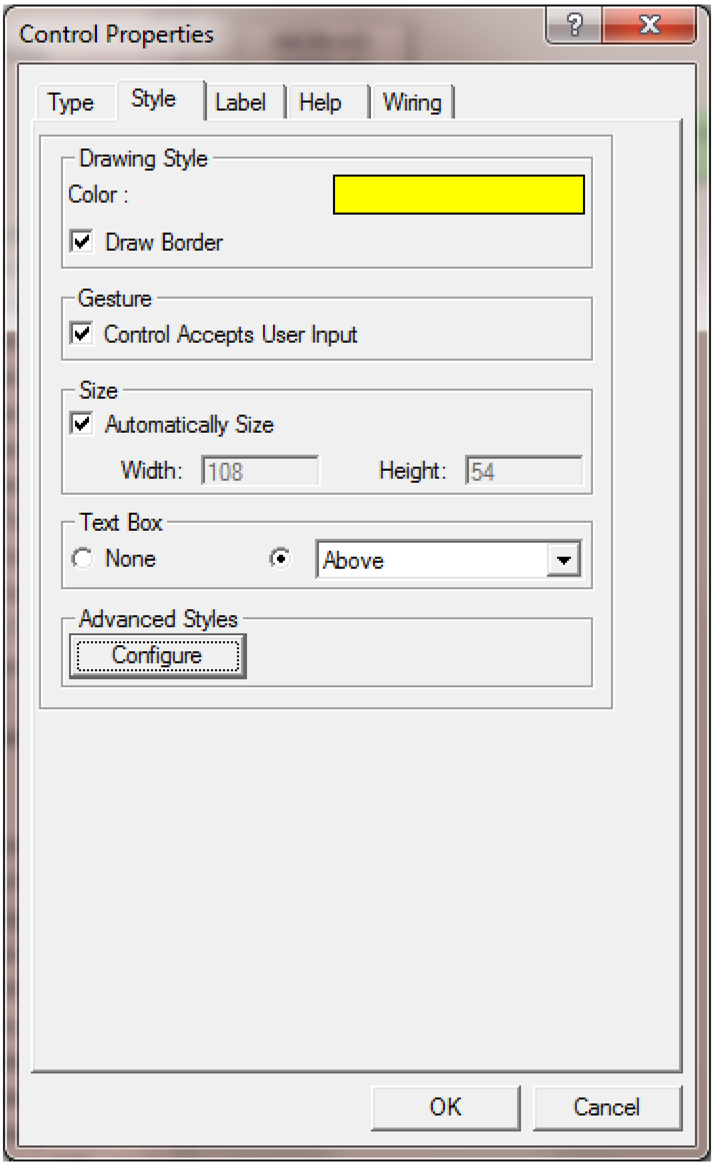
The style of the control is very similar to the combo box shown in NWare and Kiosk.

NWare allows you to customize the appearance of the combo box to help it conform to the design of your Kiosk2Go project. You can change a number of different attributes, including the font name and size, background color and border color. Below are some examples.



⏩ *To customize the appearance of a combo box*

1. Right-click the combo box, and then click **Control Properties**.
2. Click the **Style** tab.



Control Properties

Type | Style | Label | Help | Wiring

Drawing Style

Color :

☑ Draw Border

Gesture

☑ Control Accepts User Input

Size

☑ Automatically Size

Width: 108 Height: 54

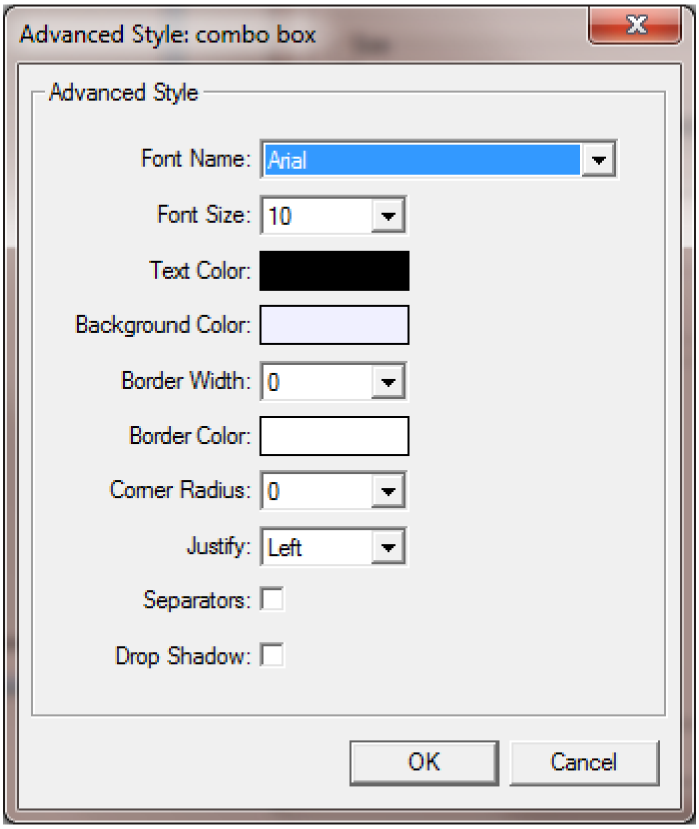
Text Box

○ None ● Above

Advanced Styles

3. Under **Advanced Styles**, click **Configure**.

The **Advanced Style: combo box** dialog box is displayed.



Font Name: Arial

Font Size: 10

Text Color:

Background Color:

Border Width: 0

Border Color:

Corner Radius: 0

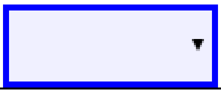
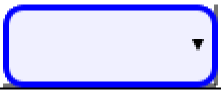
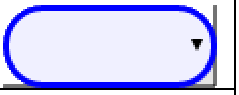
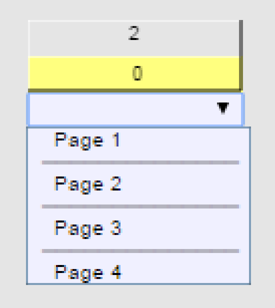
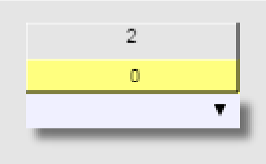
Justify: Left

Separators: □

Drop Shadow: □

4. Select different style options to alter the appearance of the combo box.

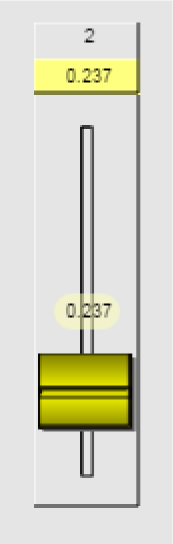
Font Name	The name of the font used to render the items in the combo box list.
Font Size	The size in points of the font used to render the items in the combo box list.
Text Color	The color of the text shown in the combo box list.
Background Color	The color shown behind the combo box. The same color is shown behind the list when it is open or closed.
Border Width	The width of the border around the edge of the combo box when it is closed.
Border Color	The color of the border around the edge of the combo box when it is closed.
Corner Radius	The radius of the curve of the border displayed around the edge of the combo box. A larger number produces a rounder look for the corners. A smaller number produces a sharper look. Some examples are shown below.

	0	10	20
			
Justify	Specifies the alignment for the list entries: Left, Center or Right.		
Separators	When selected, horizontal lines are shown between each of the options in the list.  		
Drop Shadow	When selected, adds a drop shadow to the combo box.  		

5. Click **OK**.

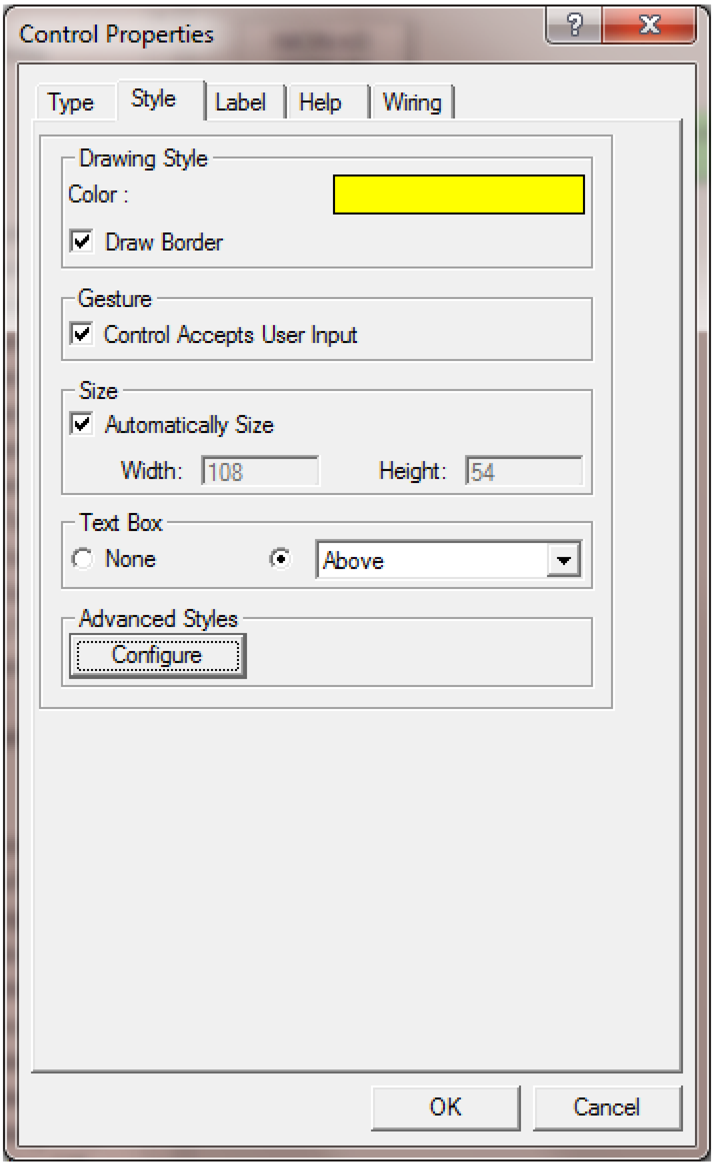
Displaying the value of a fader as a tooltip

You can display the value selected on a fader as a tooltip above the fader handle. As the fader is gestured, the value moves with the handle, so it is always visible. When you stop gesturing, the value is hidden.



⏩ *To display the value of a fader as a tooltip*

1. Right-click the fader control, and then click **Control Properties**.
2. Click the **Style** tab.



Drawing Style
Color :
☑ Draw Border

Gesture
☑ Control Accepts User Input

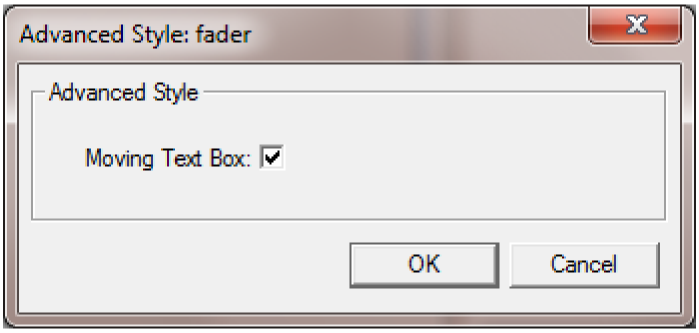
Size
☑ Automatically Size
Width: 108 Height: 54

Text Box
○ None ● Above

Advanced Styles
Configure

3. Click **Configure**.

   The **Advanced Style: fader** dialog box is displayed.



Advanced Style
Moving Text Box: ☑

4. Select the **Moving Text Box** check box.

5. Click **OK**.
6. Click **OK**.

How control aliases are used in Kiosk2Go

By default, each control inside a Kiosk2Go Layout device is assigned a control alias automatically when the project is emulated or deployed. Each alias starts with a tilde (~) and is numbered. For example: ~131915592. The alias for a control is displayed on the Wiring tab.

If you prefer, you can assign your own more meaningful aliases to controls, like *gain1* or *fader3*, for example. This makes it easier to work with the controls in an external program when you are using a control protocol, like RATC.

Adding context sensitive help to a Kiosk2Go control

You can specify help text for individual controls that can be viewed by the Kiosk2Go user. This can help the user to understand when they need to use the control, or explain what will happen if they change the value of a control, for example.

You can add help text directly to the properties of a control, or you can store the help in an external file that must be added to the NWare project before deploy time. Using an external file adds flexibility, allowing the file to be worked on independently of the NWare project, and managed using a help tool, such as *Author-it* (*http://www.author-it.com*) or *Flare* (*http://www.madcapsoftware.com/products/flare/*).

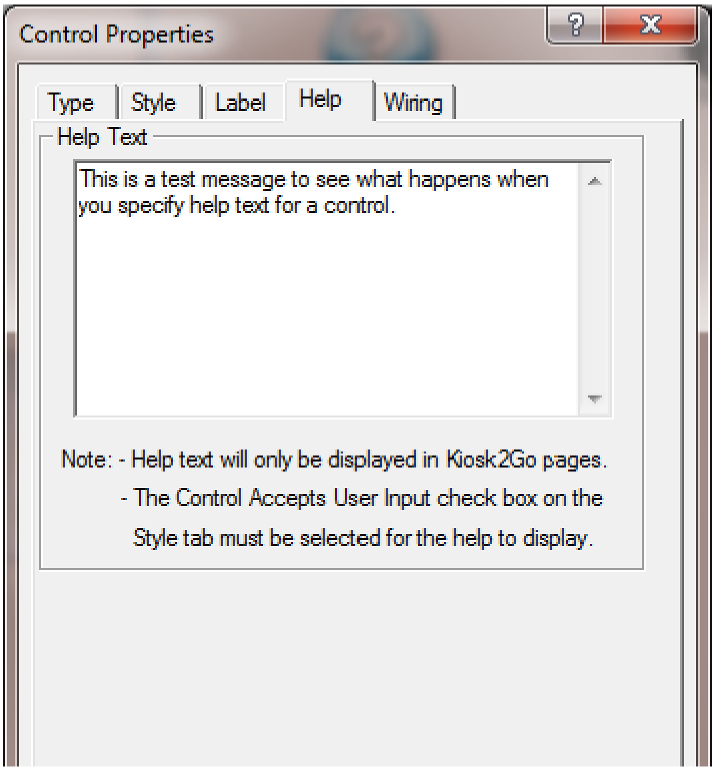
Notes:

- When you add or change help text, the new text will not be available in Kiosk2Go until you have recompiled and deployed / emulated the project. Changing the text does not automatically trigger a recompile, you must hold down SHIFT and click the Deploy button or Emulate button.
- The *Control Accepts User Input* check box must be selected on the Style tab of the Control Properties dialog box in order for the help to display.

Adding help text directly to a control

1. Open the **Kiosk2Go** block and click the page containing the control.
2. Right-click the control, and then click **Control Properties**.
3. Click the **Help** tab.

4. In the **Help Text** box, type the text you want to display for the control.



This is a test message to see what happens when you specify help text for a control.

Note: - Help text will only be displayed in Kiosk2Go pages.
- The Control Accepts User Input check box on the Style tab must be selected for the help to display.

5. Click **OK**.

Adding help text using a topic file

In order to link help from a topic file to controls on a Kiosk2Go page, you need to complete a number of steps, as outlined in this section. The topic file must be written using the format shown, otherwise NWare will be unable to display the help correctly.

The help file format consists of a heading row, followed by individual rows that define the help ID, language and text for an individual topic. Below is an example showing the mandatory heading row, followed by an individual topic.

```
"$HelpId","$Language","$HelpText"
```

```
"HelpID1","EN","This is some help"
```

The data is written using a comma separated values (CSV) format, with each item in double quotes.

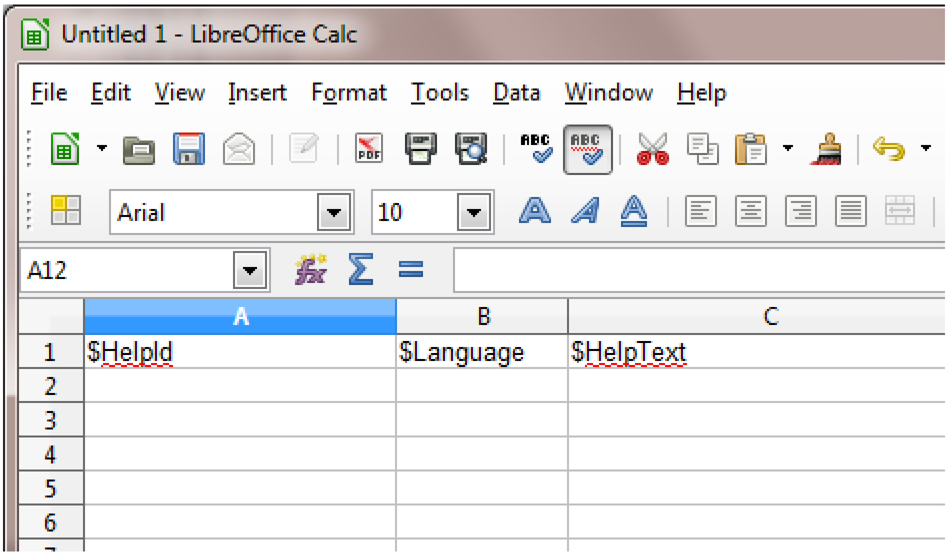
Help text can be created using an editor, such as Windows Notepad or Notepad++, but we recommend that you use either a spreadsheet or help authoring tool to ensure that the format is consistent and no quotes or commas are omitted or accidentally repeated.

Notes:

- If you want to use a help authoring tool, you will need to configure it to produce output in the format shown above. This may involve writing special macros that run after the help has been published to format the output.
- The example below uses *LibraOffice Calc* (*http://www.libreoffice.org/discover/calc/*), a free to download spreadsheet program. At the time of writing, there was no built-in support for saving quoted cell data from Microsoft Excel.
- The only language (and therefore, language code) currently supported is English (EN).

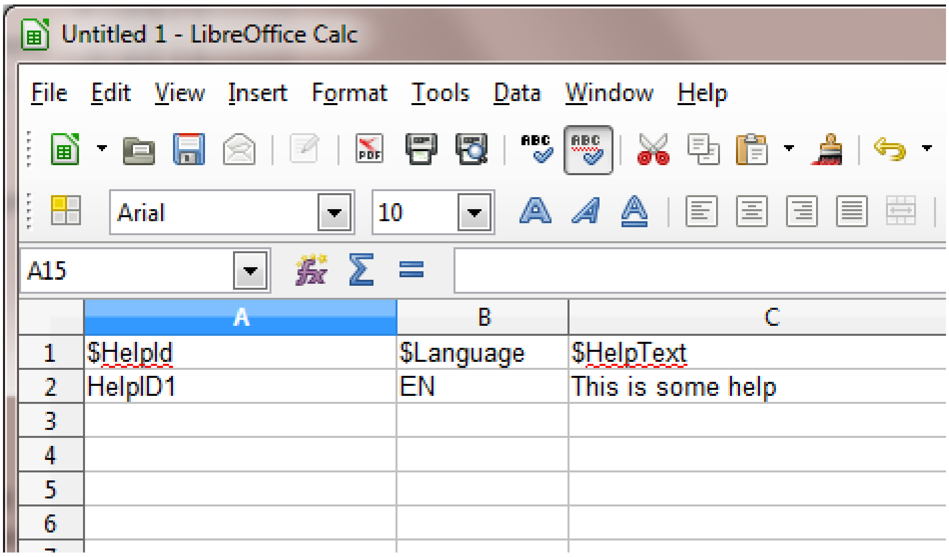
Creating the help text for a control using LibraOffice Calc

1. Open LibraOffice Calc.
2. Create a new spreadsheet.
3. Add the column headings to the spreadsheet.


Untitled 1 - LibreOffice Calc
File Edit View Insert Format Tools Data Window Help
Arial
10
A12

	A	B	C
1	$HelpId	$Language	$HelpText
2			
3			
4			
5			
6			

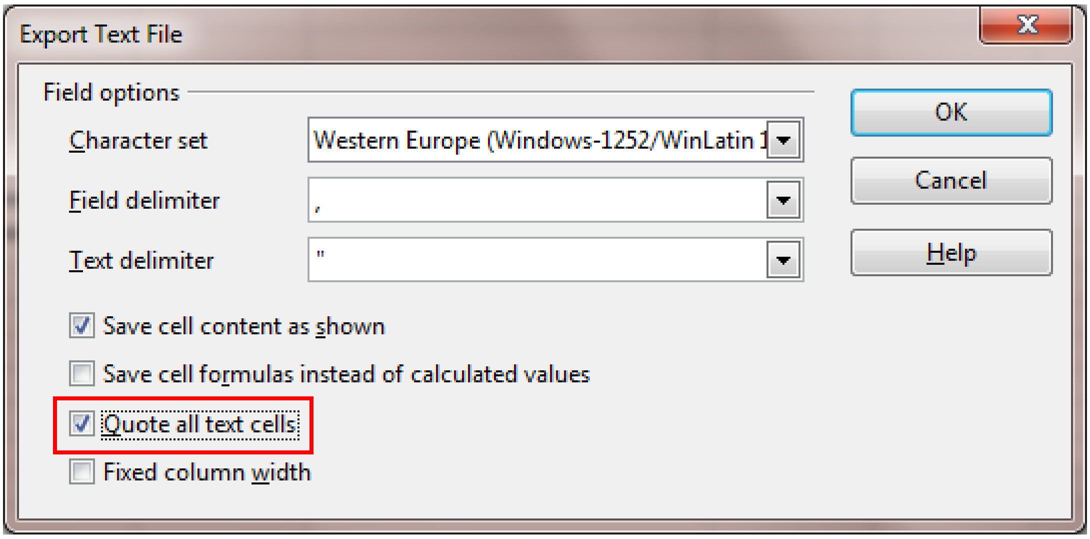
4. Add a row for each Kiosk2Go control containing a help ID, language (EN) and help text.


Untitled 1 - LibreOffice Calc
File Edit View Insert Format Tools Data Window Help
Arial
10
A15

	A	B	C
1	$HelpId	$Language	$HelpText
2	HelpID1	EN	This is some help
3			
4			
5			
6			

5. Once you have finished adding rows to the spreadsheet, click the **Save** button on the toolbar to save the spreadsheet.
6. Choose **Text CSV** as the file format, and specify *HelpText.csv* as the file name.
7. Click **Save**.
8. If a warning is displayed, asking you to choose CSV or ODF format, click **Use Text CSV Format**.

9. On the **Export Text File** dialog box, select the **Quote all text cells** check box.



Export Text File

Field options

Character set: Western Europe (Windows-1252/WinLatin 1)

Field delimiter: ,

Text delimiter: "

☑ Save cell content as shown

□ Save cell formulas instead of calculated values

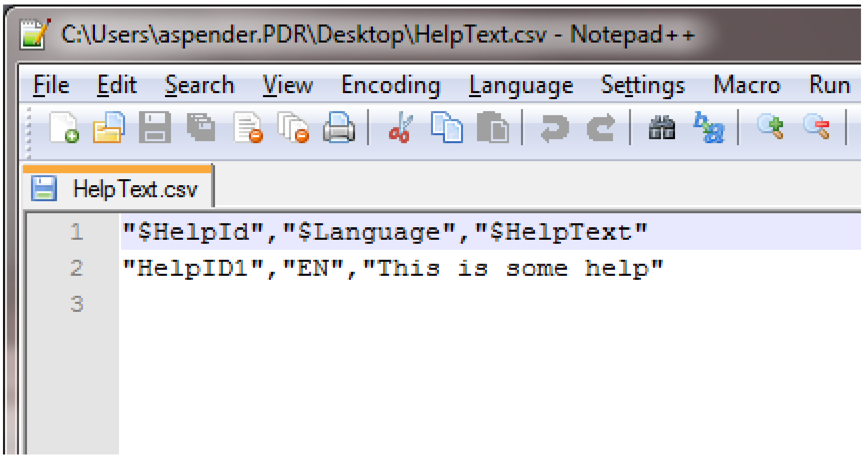
☑ Quote all text cells

□ Fixed column width

10. Click **OK**.

11. Once the export is complete, open the *HelpText.csv* file in a text editor, such as *Notepad++* (*http://notepad-plus-plus.org/*), and check that the formatting is correct.

    Column headings and column entries must be comma-separated and enclosed in double quotes.



C:\Users\aspender.PDR\Desktop\HelpText.csv - Notepad++

```
"$HelpId","$Language","$HelpText"
"HelpID1","EN","This is some help"
```

⏩ *Importing the help file into NWare*

1. Click the **Media** tab.
2. Right-click the **Media** tab, and then click **Import**.
3. Browse to the *HelpText.csv* file, and then click **Open**.

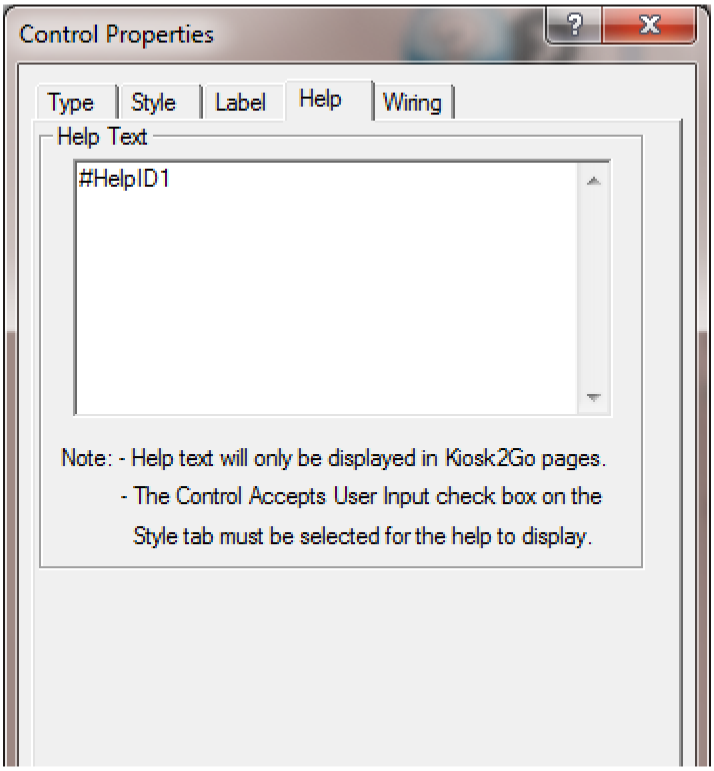
   If the import is successful, information about the file will be displayed on the Media tab.

   If there was a problem with the import, an error will be displayed. Check the format of the csv file and repeat the import process.

⏩ *Referencing a help topic from a Kiosk2Go control*

1. Open the **Kiosk2Go** block and click the page containing the control.
2. Right-click the control, and then click **Control Properties**.
3. Click the **Help** tab.

4. In the **Help Text** box, type a hash (#) followed by the help ID that identifies the help topic for the control.



Help Text

#HelpID1

Note: - Help text will only be displayed in Kiosk2Go pages.
- The Control Accepts User Input check box on the Style tab must be selected for the help to display.

5. Click **OK**.

Checking your help text appears in Kiosk2Go

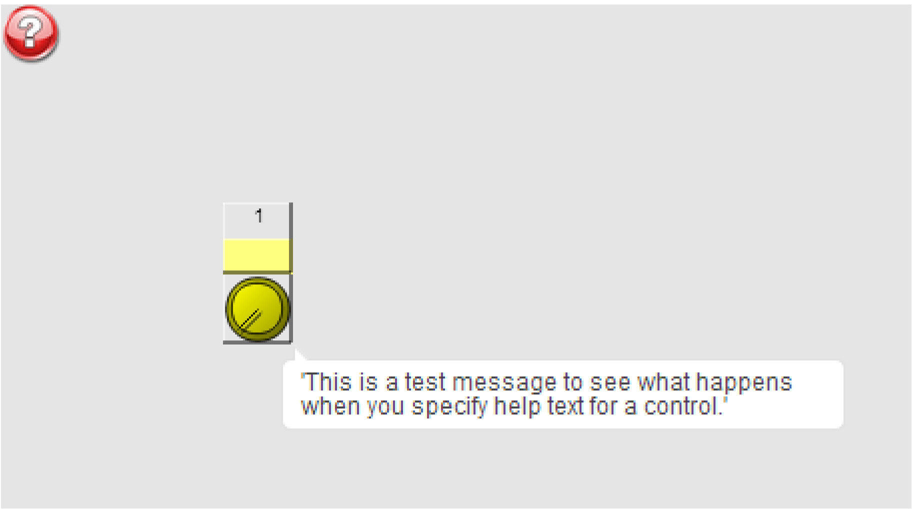
1. Press SHIFT+Deploy .
2. Open a browser, then in the address bar, type the IP address of the MediaMatrix node (or NWare PC if emulating) plus port number 7681 in this format:

```
http://<IP address>:7681
```

3. Log on.
4. Display the page containing the control with help text.
5. Click/tap **Help** .

The button will turn red. The controls cannot be gestured.

6. Click/tap the control to display the help.



'This is a test message to see what happens when you specify help text for a control.'

7. Click/tap **Help** ? to change it back to ?.

   The controls can now be gestured again.

Adding a custom help button

You can configure a button in Kiosk2Go that calls the context-sensitive help. It works in exactly the same way as the default help button, but you can customize its appearance to match the look of your project.

To add a custom help button

1. Drag a **Generic Controls** device from the device tree over to the main NWare design page.
2. Specify **boolean** in the **Type of control(s)** list.
3. Open the **Kiosk2Go Layout** block and click the page where you want to add the help button.
4. Open the **Generic Controls** control block, and then copy and paste the control onto the **Kiosk2Go Layout** device page.
5. Right-click the control, and then click **Control Properties**.
6. Click the **Wiring** tab.
7. Type **helpbutton** in the **ControlAlias** box.
8. Click **OK**.

Tip: Any control can be configured to call the help. You just need to type **helpbutton** in the **ControlAlias** box.

Adding a logout button

You can configure a button in Kiosk2Go that will log out the Kiosk2Go user and display the log on screen.

To add a logout button

1. Drag a **Generic Controls** device from the device tree over to the main NWare design page.

2. Specify **boolean** in the **Type of control(s)** list.
3. Open the **Kiosk2Go Layout** block and click the page where you want to add the logout button.
4. Open the **Generic Controls** control block, and then copy and paste the control onto the **Kiosk2Go Layout** device page.
5. Right-click the control, and then click **Control Properties**.
6. Click the **Wiring** tab.
7. Type **logoutbutton** in the **ControlAlias** box.
8. Click **OK**.

Tip: Any control can be configured to perform the logout function. You just need to type **logoutbutton** in the **ControlAlias** box.

Setting up the MediaMatrix node for Kiosk2Go

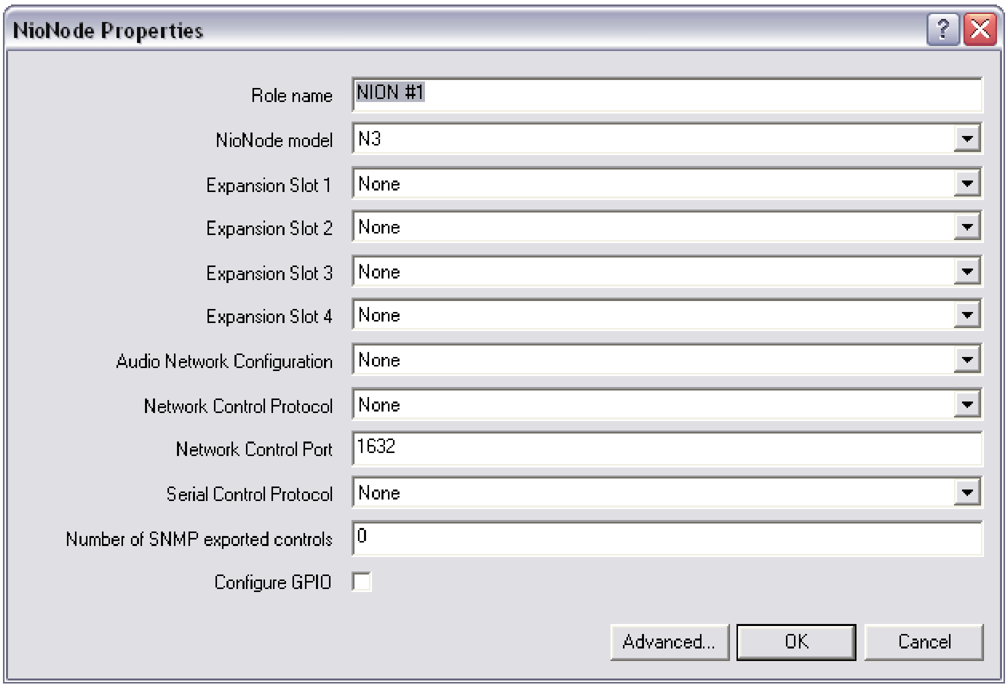
In order to use Kiosk2Go, your project must contain at least one NioNode, nTouch 180 node or nControl node. Furthermore, as Kiosk2Go uses RATC2, you must enable the use of RATC2 on the device properties of the node.

Note: Currently, Kiosk2Go only supports connections on port 1632. You cannot use a different port number.

To set up the MediaMatrix node for Kiosk2Go

1. Right-click the node, and then click **Device Properties**.

   (NioNode example below).



NioNode Properties

Role name	NION #1
NioNode model	N3
Expansion Slot 1	None
Expansion Slot 2	None
Expansion Slot 3	None
Expansion Slot 4	None
Audio Network Configuration	None
Network Control Protocol	None
Network Control Port	1632
Serial Control Protocol	None
Number of SNMP exported controls	0
Configure GPIO	☐

2. In the **Network Control Protocol** list, click **RATC2**.
3. Click **OK**.

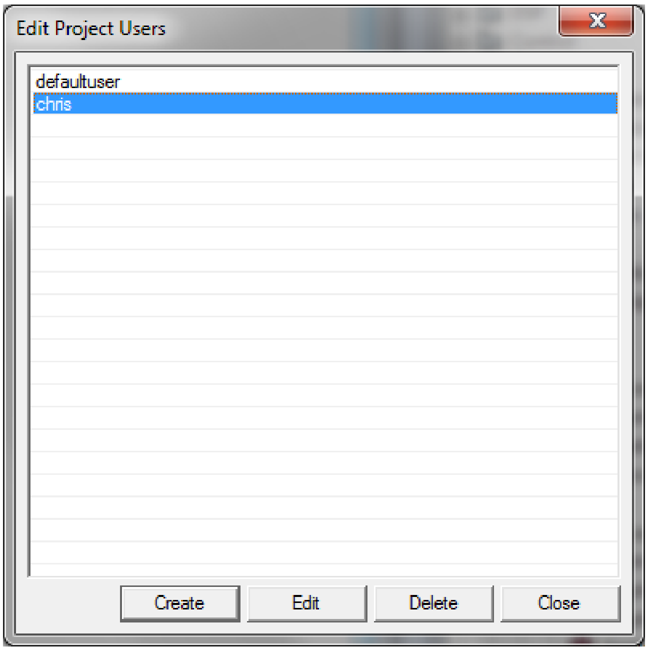
Setting up user access for Kiosk2Go

When you start Kiosk2Go on your mobile device, you will be asked to log on. A valid user name and, if appropriate, password, must be specified. The user must have been granted the *Network Control Access* permission, plus access to the Kiosk2Go devices you want them to use.

⏩ *To set up user access for Kiosk2Go*

1. On the **File** menu, click **User Accounts**.

   The **Edit Project Users** dialog box is displayed.

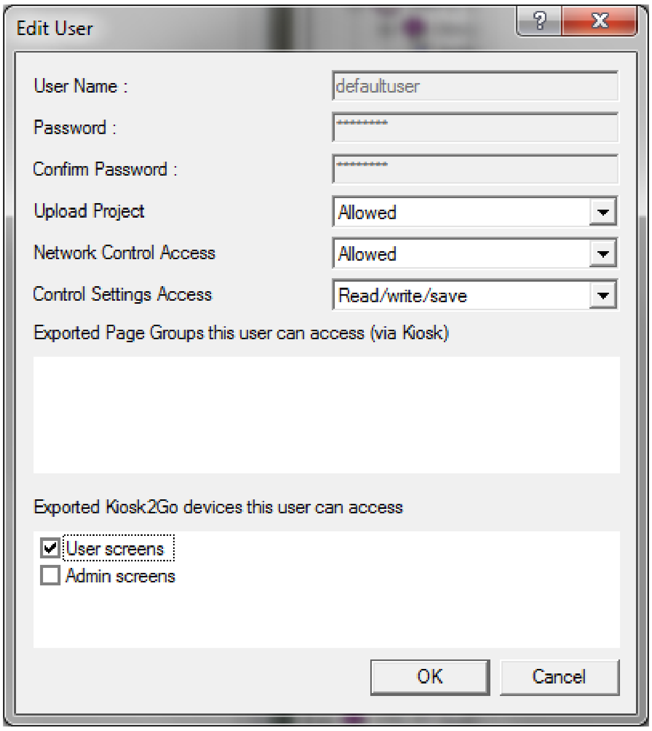


Edit Project Users

defaultuser
chris

2. Double-click the user name you want to use for Kiosk2Go, or click **Create** to create a new user.

The **Edit User** dialog box is displayed.



Edit User

User Name : defaultuser

Password : *********

Confirm Password : ********

Upload Project: Allowed

Network Control Access: Allowed

Control Settings Access: Read/write/save

Exported Page Groups this user can access (via Kiosk)

Exported Kiosk2Go devices this user can access

☑ User screens

□ Admin screens

3. In the **Network Control Access** list, click **Allowed**.
4. Under **Exported Kiosk2Go devices this user can access**, select check boxes next to the devices you want the user to be able to access in Kiosk2Go.

   If you select more than one device, the user will be prompted to choose a device when they log on to Kiosk2Go.
5. Click **OK**.

Displaying an image when the connection to the project is lost

You can specify an image to be displayed in the browser when the connection between Kiosk2Go and the NWare project is lost. This can be a helpful addition to let the users know what has happened when the controls in the project become unresponsive.

An example image, *Kiosk2GoProjectConnectionLost.jpg*, is supplied in the *projects* folder, which is located under the NWare installation folder.

Notes:

- The image file you use must be named *Kiosk2GoProjectConnectionLost.jpg*.
- The size of the image file you use needs to suit the screen resolutions of the devices that will be using Kiosk2Go.

To display an image when the connection to the project is lost

1. In NWare, click the **Media** tab to display the list of imported images.
2. Right-click on the image list, and then click **Import**.
3. Browse to the file *Kiosk2GoProjectConnectionLost.jpg*.

   The one supplied with NWare is located in \<*NWare install folder*>*projects*.

4. Select the image, and then click **Open**.
5. Emulate or deploy the project.

Testing the Kiosk2Go project in a browser

Once you have added controls to the pages of your Kiosk2Go device, set up the user access, and the project contains a MediaMatrix node configured for RATC, you can emulate or deploy the project and test it in a browser. This can be on a PC or on a mobile device.

Note: If more than one instance of NWare is running on your PC and you emulate a project in each instance, Kiosk2Go can only connect to the first emulated project.

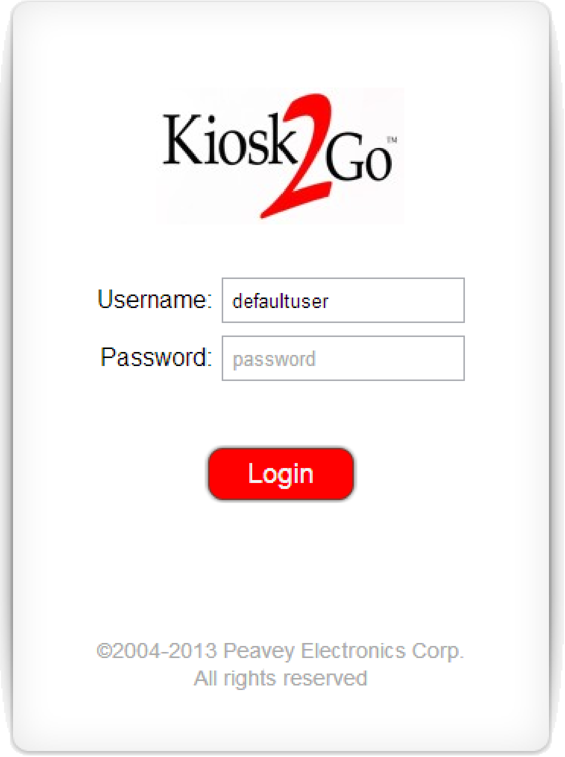
⏩ *To test the Kiosk2Go project in a browser*

1. *Emulate* (on page 167) or Deploy the project.
2. Open a browser, then in the address bar, type the IP address of the MediaMatrix node (or NWare PC if emulating) plus port number 7681 in this format:

```
http://<IP address>:7681
```

Tip: There is a direct link to Kiosk2Go in the web interface of the node.

You will be prompted to log on.



Username: defaultuser

Password: password

©2004-2013 Peavey Electronics Corp.
All rights reserved

3. In the **Username** and **Password** boxes, type the user name and password for the Kiosk2Go user, and then click/tap **Login**.

Tips: If you cannot log on to Kiosk2Go, check the following:

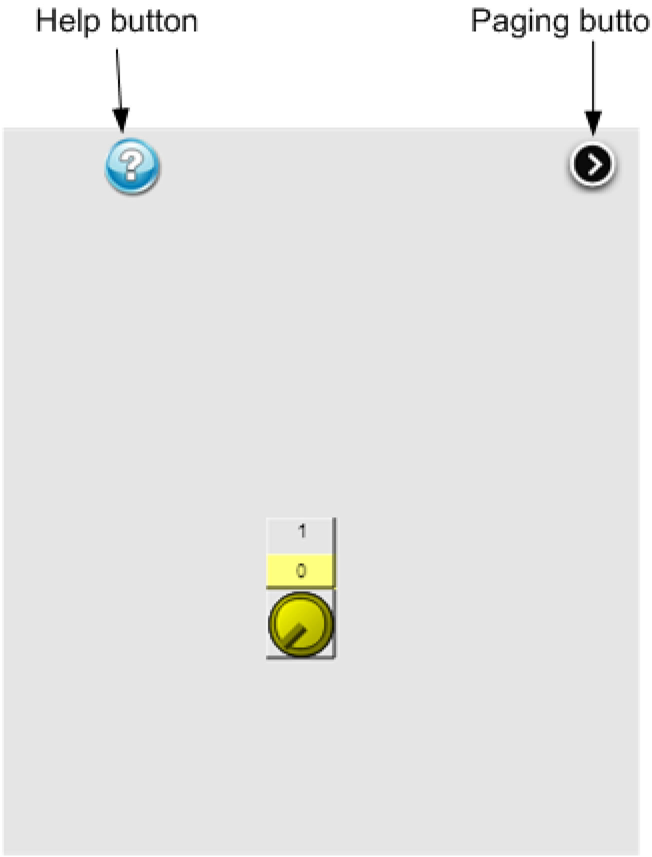
- The password you are typing is correct.

- The user has been granted access to at least one of the Kiosk2Go devices (page groups).
- The node hosting the role has RATC2 specified as the network control protocol.

4. If you are prompted to select a Kiosk2Go device name, click/tap a device name in the list.

   A list of device names is shown when the user has been granted access to more than one Kiosk2Go device (page group).

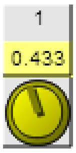
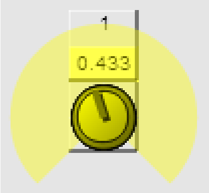
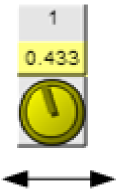
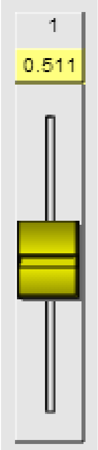
   The first page from the Kiosk2Go Layout block will be displayed. Help and paging buttons are shown if the corresponding options were selected on the device properties of the block.

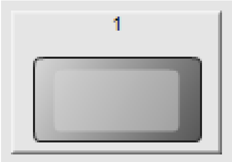


   Click/tap the controls to interact with them. To display the help text specified for a control, click/tap the help button, and then click/tap the control.

   You can use the paging buttons to move between pages.

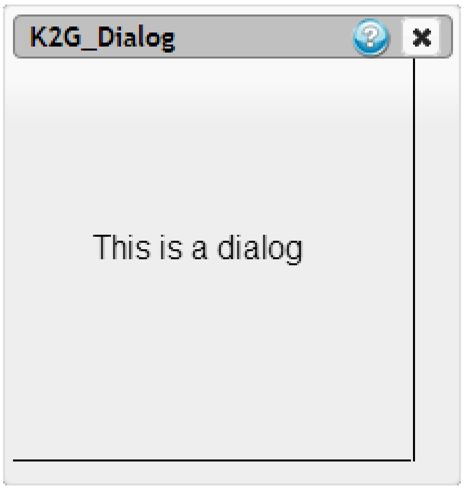
Using controls in Kiosk2Go

Control	Enhanced Touch mode	NWare Compatible mode
Previous page and next page ❮ ❯	Click/tap to change to the next or previous page.	
Knob control 	Click/tap the center of the knob to change to gesture mode. A colored gesture area will be displayed around the knob control.  Drag inside this area to gesture the knob and change its value. Click/tap the center of the knob to exit gesture mode.	Click/tap/hold the center of the knob to change to gesture mode. Drag vertically or horizontally to gesture the knob and change its value. 
	Click/tap the value box to specify a new value. If you are using a PC, you can double-click the value box to edit the current value.	
Fader control 	Drag the fader handle to change the value. Click/tap the value box to specify a new value. If you are using a PC, you can double-click the value box to edit the current value. If you are using a touch device, you can drag multiple faders at the same time.	
Button control	Click/tap to operate. A button can be a toggle button (staying	

 1	down or up when you gesture it) or a momentary button (staying in the down state only momentarily).
Help ?	Click/tap the button to change to help mode ?. You will not be able to gesture controls. Click/tap a control to display the context sensitive help. Click/tap the Help button again to switch back to gesture mode.

Using dialogs to interact with the user

You can use dialogs to display controls to the user, and just with dialogs in Windows and other operating systems, they can be closed by tapping the close (X) button in the top corner.


K2G_Dialog

This is a dialog

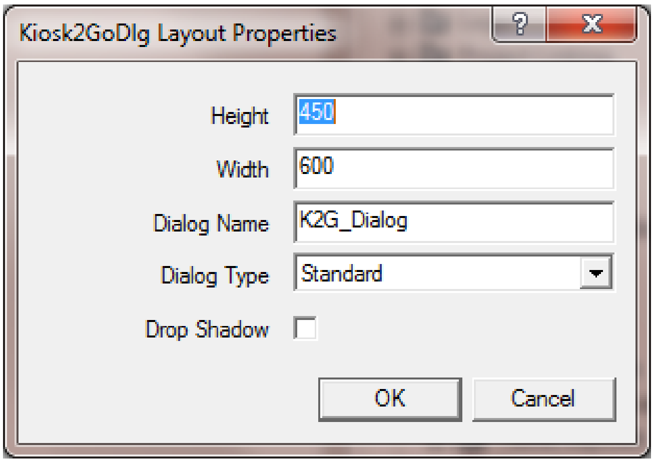
The dialogs also support help via the help button in the same way as pages on a Kiosk2Go device. For more information, see *Adding context sensitive help to a Kiosk2Go control* (on page 250).

Note: Currently, you cannot drag a dialog to reposition it.

⏩ *To add a dialog to your design*

1. In the device tree, expand the **Graphic Elements & Blocks / Kiosk2Go Layouts** folder.
2. Drag a **K2G Dialog** device over to the design page.

The **Kiosk2GoDlg Layout Properties** dialog box is displayed.



Kiosk2GoDlg Layout Properties

Height: 450

Width: 600

Dialog Name: K2G_Dialog

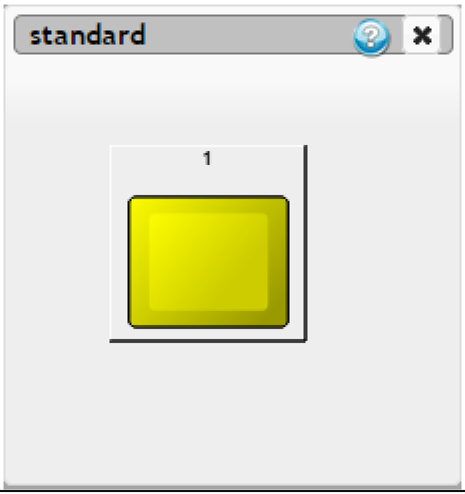
Dialog Type: Standard

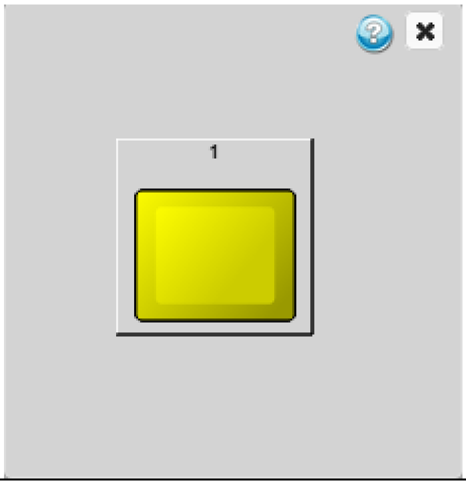
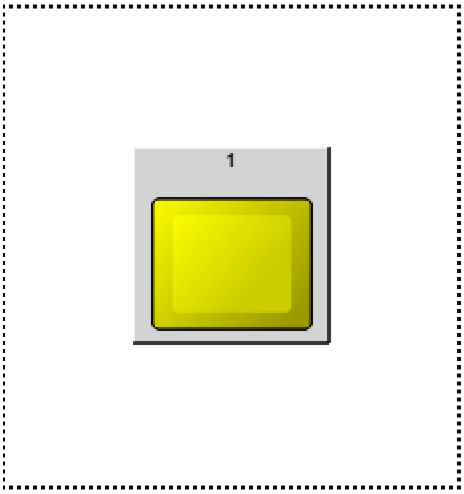
Drop Shadow: □

3. If you want to change the dimensions of the dialog, type the new values into the **Height** and **Width** boxes.
4. If you want to change the name of the dialog, type a new name in the **Dialog Name** box.

Note: Each dialog must have a unique name.

5. In the **Dialog Type** box, click an option to specify the visual style of the dialog.

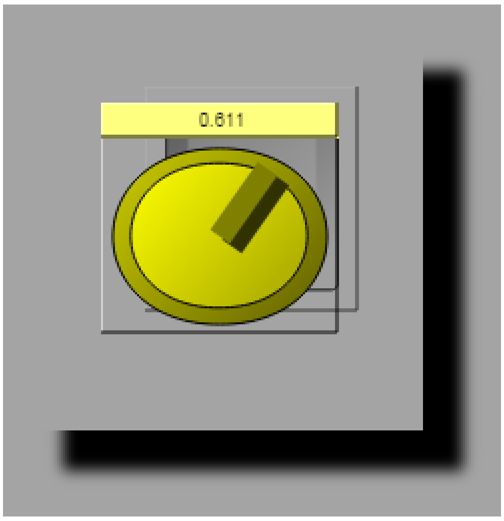
Standard	Dialog with titlebar, help button and close button. 
No Titlebar	Dialog with help button and close button, but no title bar.

	
Empty Frame	Dialog with controls only. No titlebar, help button, close button or border are shown. (A dotted border is shown in the example to illustrate that the dialog size has not changed.)  **Notes:** ▪ If a background image is added to the dialog in NWare, it will be displayed in Kiosk2Go. ▪ If you want to add a help button or close button to an *empty frame* dialog, you must add the button to the dialog manually in NWare, then assign a special alias to it on the Wiring tab: *helpbutton* or *closebutton*.

6. If you want to add a drop shadow to the dialog, select the **Drop Shadow** check box.

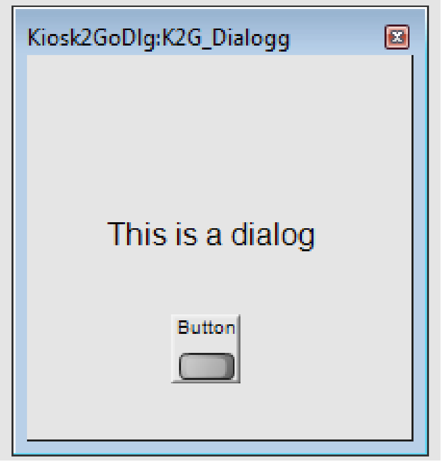
Note: This setting is only valid for the *Empty Frame* dialog type.

Here is an example.



7. Click **OK**.
8. Double-click the Kiosk2GoDlg block to open it.
9. Add Generic Controls to your design page, and then copy and paste them over to the dialog.

   In the example below, a label and a button have been added. For more information, see *Adding controls to the Kiosk2Go Layout device* (on page 235).



⏩ *To open the dialog from a button*

1. Add a boolean generic control to your design page.
2. Open a Kiosk2Go device and copy and paste the button from inside the generic control onto one of the pages.
3. Display the button control properties.
4. On the **Type** tab, under **Popup Dialog**, select the **Enable Dialog Popup** check box.
5. In the **Dialog** list, click the name of the dialog you want to link to the button.
6. Click **OK**.

   When the user taps the button in a browser, the dialog will be displayed.

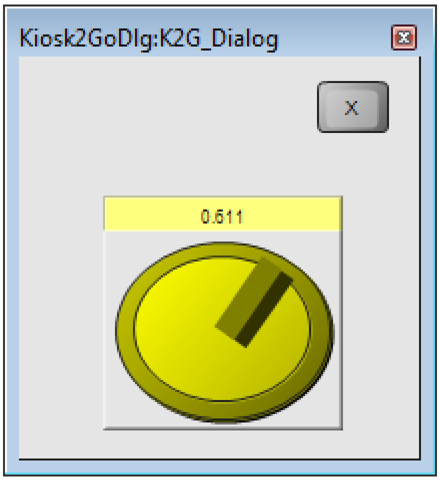
Using custom help and close buttons on dialogs

You can add your own buttons to dialogs to call the help for a control, or close the dialog.

Tip: This feature is designed to be used with the *Empty Frame* dialog style option. If you use it with one of the other style options, the default help and close buttons will also be visible to the user.

1. Add a boolean generic control (button) to your design page.
2. Open the dialog block and the generic control block, then copy and paste the button onto the control surface.

   In the example below, we have created a button that will be used to close the dialog.



3. Right-click the button, and then click **Control Properties**.
4. On the **Type** tab, under **Options**, click **Toggle**.

   Note: The *Momentary* and *String* options are not supported for custom help and close buttons.

5. Click the **Wiring** tab.
6. If you want the button to call the help, type `helpbutton` in the **Control Alias** box.

   If you want the button to close the dialog, type `closebutton` in the **Control Alias** box.
7. Click **OK**.

Linking Kiosk2Go devices together

You can add a button to a Kiosk2Go device that will open another Kiosk2Go device when the user taps it.

⏩ *To link Kiosk2Go devices together*

1. Add two Kiosk2Go devices to your design.
2. Add a boolean generic control to your design page.
3. Open the first Kiosk2Go device, and then copy and paste the button from inside the generic control onto one of the pages.
4. Display the button control properties.
5. On the **Style** tab, under **Jump to K2G device**, select the **Enable Jump to K2G device** check box.
6. In the **Device** list, click the name of the Kiosk2Go device you want to link to the button.
7. Click **OK**.

When the user taps the button in a browser, the first page from the linked Kiosk2Go device will be displayed.

Creating a browser bookmark to automatically log on to Kiosk2Go

You can create a browser bookmark that will start Kiosk2Go and automatically log on. The url for the bookmark can include a Kiosk2Go layout device name and a user name, but no password. If the user does not have a password, they will be logged on automatically when the bookmark is opened.

⏩ *To create a browser bookmark to automatically log on to Kiosk2Go*

1. Open a browser and type in the url to connect to Kiosk2Go.

   For example: `http://10.20.0.99:7681`

2. Create a bookmark, and then edit the bookmark to include *userName* and *deviceName* settings:

   ```
   http://<IP address of node>:7681/kiosk2Go.html?userName=<user
   name>&devName=<Kiosk2Go device layout device name>
   ```

   where *userName* is the NWare user name used to log on to Kiosk2Go, and *devName* is the name of the Kiosk2Go Layout device block.

   For example:

   ```
   10.20.0.99:7681/kiosk2Go.html?userName=elliot&devName=iPad
   ```

Creating a shortcut to start Kiosk2Go full screen

When you open a URL to start Kiosk2Go in a browser, tabs and a location bar are typically shown at the top of the window. These extra UI features take up screen space, which may mean some of your controls are pushed off the screen at the bottom of the window. If you create a web shortcut to a Kiosk2Go page, when it is opened, Kiosk2Go is started full screen, allowing the maximum available space for your controls.

Tip: This section uses Chrome as the example browser on Android. The procedure for other browsers will be similar.

⏩ *To create a shortcut in Chrome on Android*

1. Navigate to the Kiosk2Go start page.
2. Tap the menu icon ⋮.
3. Tap **Add to home screen**.
4. Specify a name for the shortcut.
5. Tap **Add**.

⏩ *To create a shortcut in Safari on iOS*

1. Navigate to the Kiosk2Go start page.
2. Tap ⇪.
3. Tap **Add to Home Screen**.
4. Specify a name for the shortcut.
5. Tap **Add**.

Chapter 10

Administering NWare

In This Chapter

Choosing a PC network card

If the PC running NWare is fitted with more than one network card, you can choose which card to use when communicating with the network. This is useful if, for example, your PC is connected to two separate networks via two separate network cards or via a network card and a wireless network interface.

When you start NWare, you will be asked which card you want to use. Once you choose a card, the setting is used each time you start the program. You will only be asked to choose a card at a later time if the IP address of any of the cards has changed.

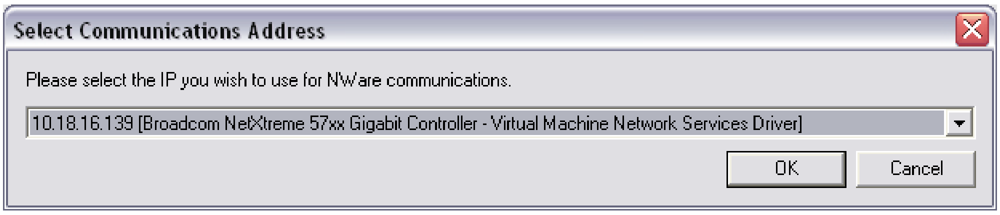
Notes:

- If you start using a new IP address to connect to a network, or disconnect and then reconnect to the network, you may need to restart Pandad. For more information, refer to the section *Recovering Pandad after it has stopped functioning* in the document *Pandad Administrator Guide*.
- You can use a wireless network with NWare and Kiosk, but we recommend that you use a wired network whenever possible, especially when deploying projects. Wi-fi connections are generally not as reliable as wired connections for transferring large amounts of data.

⏩ *To choose a PC network card*

1. If the **Select Communications Address** dialog box is not displayed, on the **Tools** menu, click **Communication Preferences**.

   The **Select Communications Address** dialog box is displayed.

   Select Communications Address

   Please select the IP you wish to use for NWare communications.

   10.18.16.139 [Broadcom NetXtreme 57xx Gigabit Controller - Virtual Machine Network Services Driver]

2. In the list, click the details of the network card you want to use.
3. Click **OK**.

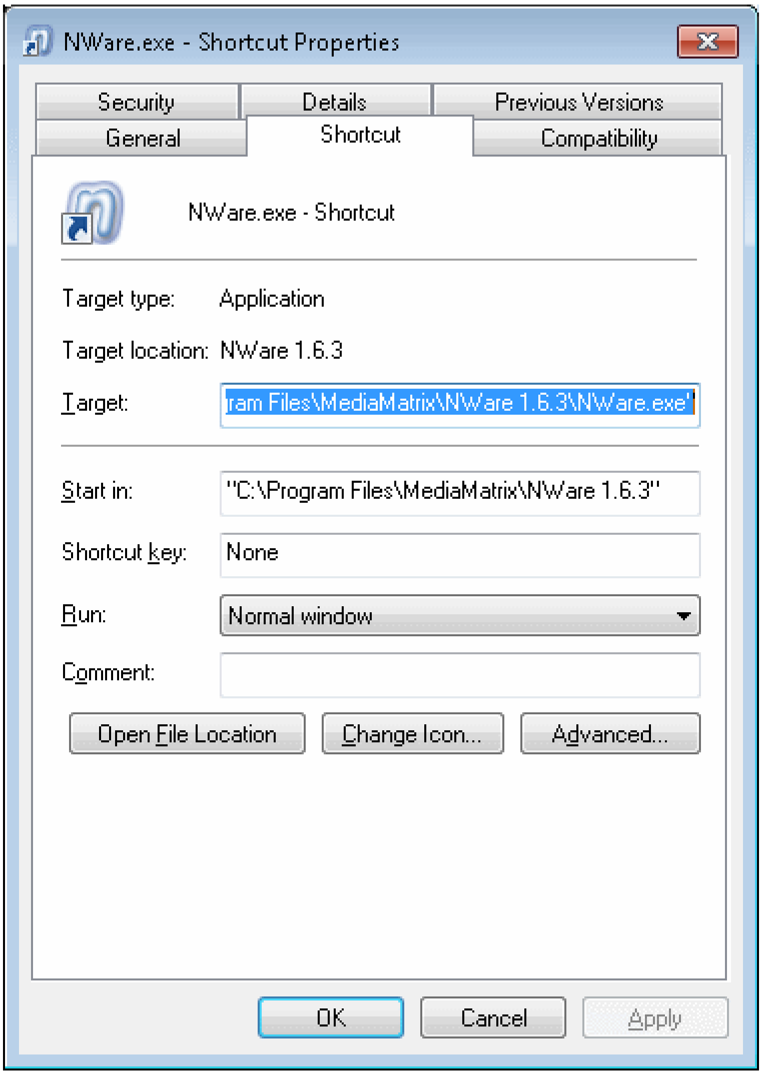
Starting NWare in full screen mode

Normally, when you start NWare, the size of the application window is the same as it was when you last started the application. However, you can force it to be maximized at start up.

⏩ *To start NWare in full screen mode*

1. Open Windows Explorer and navigate to the *NWare <version>* installation folder.
2. Right-click *NWare.exe*, and then click **Create Shortcut**.
3. Right-click the shortcut, and then click **Properties**.

The **Shortcut to NWare.exe Properties** dialog box is displayed.



Target type: Application

Target location: NWare 1.6.3

Target: ram Files\MediaMatrix\NWare 1.6.3\NWare.exe"

Start in: "C:\Program Files\MediaMatrix\NWare 1.6.3"

Shortcut key: None

Run: Normal window

Comment:

4. Add `/fullscreen=1` after `\NWare.exe"` in the **Target** box.

   Here is an example:

   ```
   "C:\Program Files\MediaMatrix\NWare 1.6.3\NWare.exe" /fullscreen=1
   ```

5. Click **OK**.
6. Double-click the shortcut.

Showing and hiding tooltips (hover popups) on controls

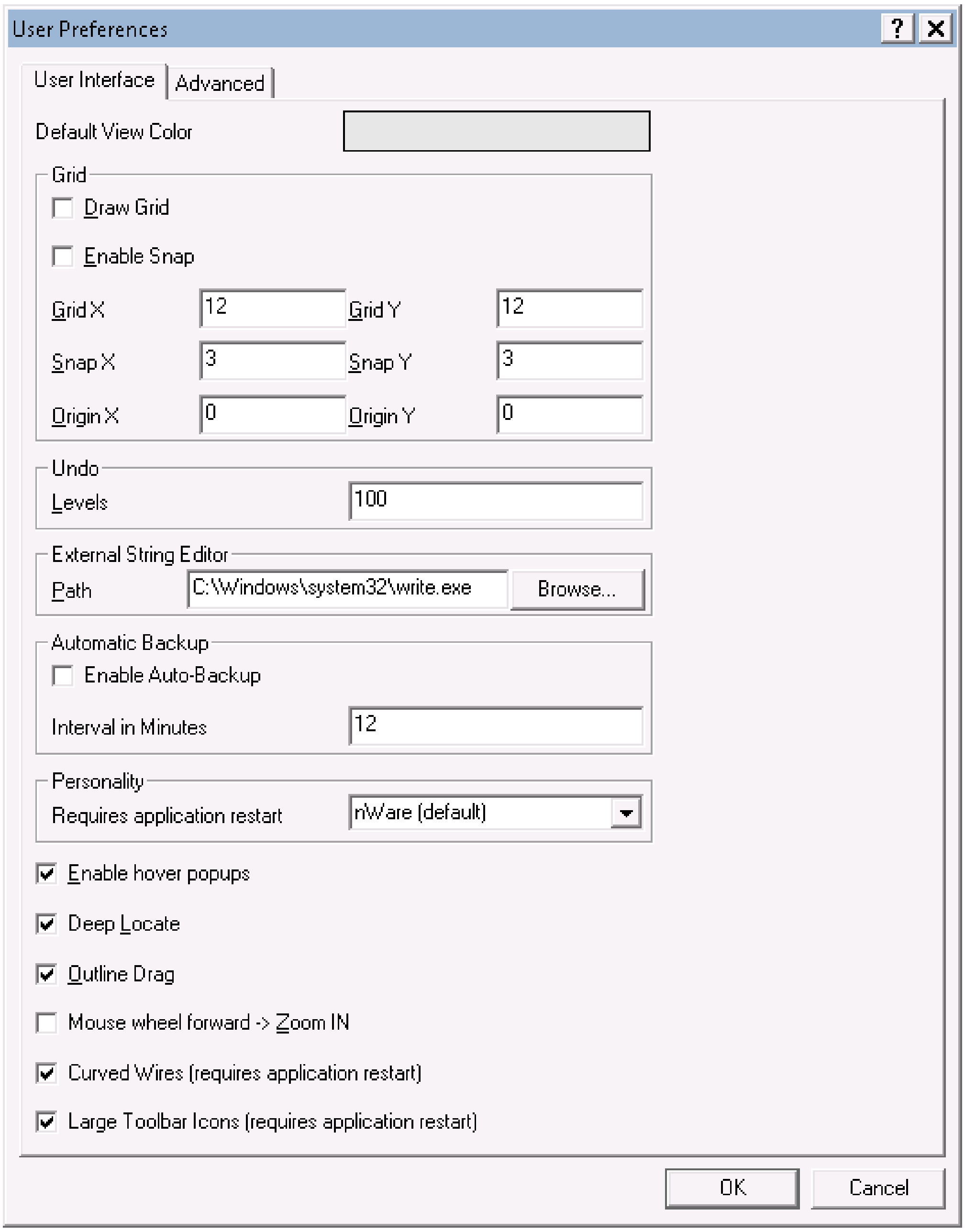
When you *emulate* (on page 167) or deploy a project and then connect to it using NWare, by default, tooltips are displayed for the different controls. These can be switched on and off from the *User Preferences* dialog box.

Tip: The display of tooltips in Kiosk is controlled using a preferences file. For more information, see *Showing or hiding tooltips (hover popups)* (on page 222).

⏩ *To show or hide tooltips (hover popups) on controls*

1. On the **Tools** menu, click **User Preferences**.

The **User Preferences** dialog box is displayed.



User Preferences

User Interface | Advanced

Default View Color

Grid
- □ Draw Grid
- □ Enable Snap

Grid X: 12 | Grid Y: 12

Snap X: 3 | Snap Y: 3

Origin X: 0 | Origin Y: 0

Undo
Levels: 100

External String Editor
Path: C:\Windows\system32\write.exe

Automatic Backup
- □ Enable Auto-Backup

Interval in Minutes: 12

Personality
Requires application restart: nWare (default)

- ☑ Enable hover popups
- ☑ Deep Locate
- ☑ Outline Drag
- □ Mouse wheel forward -> Zoom IN
- ☑ Curved Wires (requires application restart)
- ☑ Large Toolbar Icons (requires application restart)

2. Select or clear the **Enable hover popups** check box.
3. Click **OK**.

Dealing with project compatibility issues

Normally, when you open a project created in a previous version of NWare, it continues to function without any problems. However, when a new version of NWare is released, sometimes devices that existed in previous versions are updated or even removed from the device tree; this in turn means you may need to perform some maintenance tasks before your project will run.

When the settings for a device are updated, because a new check box has been added to the device properties, for example, the old version of the device in your project is referred to as an *out of date* device. It is quite common for out of date devices to exist, as we often improve algorithms and the devices that control them when we release a new version of software.

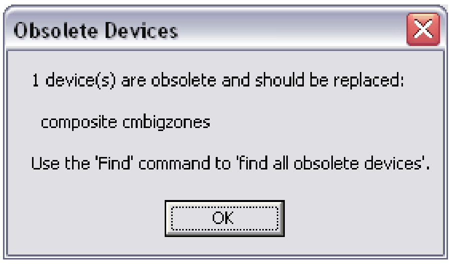
Tip: We recommend that you check for out of date devices each time you upgrade NWare, and also if you choose to downgrade NWare to an earlier version.

Sometimes, devices are removed from the device tree altogether. This is usually because the devices are *beta* devices created for a particular project. In a later release, the functionality from the devices is reworked and transferred to other devices, then the original devices are no longer required. These devices are quite rare, and a referred to as *obsolete* devices.

Finding out of date and obsolete devices

NWare will not warn you that your project contains out of date devices, but if you suspect that it does, you will need to select a menu option to locate the out of date devices and update them.

When MediaMatrix has removed a device from the device tree because it is *obsolete*, and your project contains this device, NWare will display a warning when the project is opened.



Obsolete Devices

1 device(s) are obsolete and should be replaced:

composite cmbigzones

Use the 'Find' command to 'find all obsolete devices'.

You will need to replace the device with an alternative from the device tree. There is no automatic replacement feature.

Note: If you create a project that contains a device new to the current version of NWare, then open the project in a previous version of NWare, an error will be displayed explaining that items of unrecognized data have been discarded. Further errors will be displayed if you attempt to emulate or deploy the project. To ensure the project works correctly, you should only open it in the current version of NWare.

Tip: When you save a project file, details about the version of NWare used are saved with the project. When you subsequently load the project file, the version details are displayed on the *Log* tab.

Preparing the project files

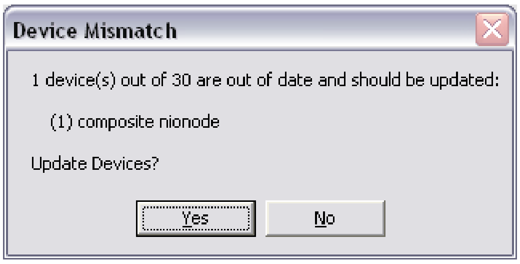
1. Make a copy of the original project created with the old version of NWare.
2. Rename the copy to indicate that it will be the new version, compatible with the latest version of NWare.

   In the procedures below, the original project will be referred to as the *old project*, and the new version will be referred to as the *new project*.

Updating all the out of date devices

1. Open the new project in the latest version of NWare.
2. On the **Tools** menu, click **Update All Out of Date Devices**.

   The **Device Mismatch** dialog box is displayed showing the number of out of date devices that have been detected.



Device Mismatch

1 device(s) out of 30 are out of date and should be updated:

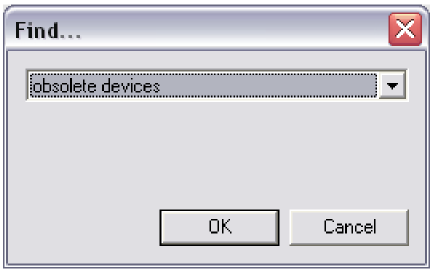
(1) composite nionode

Update Devices?

3. Click **Yes** to update the devices.
4. Save the project.

Replacing the obsolete devices

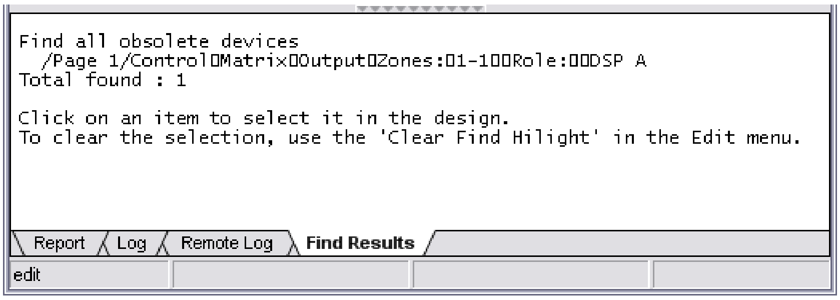
1. Press **CTRL+F** to display the **Find** dialog box.
2. In the list, click **obsolete devices**.



Find...

obsolete devices

3. Click **OK**.
4. Click the **Find Results** tab to view the obsolete devices in the project.



```
Find all obsolete devices
  /Page 1/Control□Matrix□Output□Zones:01-10□Role:□□DSP A
Total found : 1

Click on an item to select it in the design.
To clear the selection, use the 'Clear Find Hilight' in the Edit menu.
```

5. Click the path link to the first device to locate it.
6. Drag a new device of the same type from the device tree over to the page.

Note: This device will replace the obsolete device, so it is important to choose it carefully to ensure it will provide the same functionality as the old device. If you are unsure about choosing an alternative device, contact *MediaMatrix Technical Support* (*mailto:mmtechsupport@peavey.com*).

7. If you are prompted to specify a name for the device, specify a name that is different to the original device name.

8. Display the device properties of both the obsolete device and the replacement device.
9. Select settings on the replacement device to match the settings used for the obsolete device.
10. Wire the replacement device to the same devices the obsolete device is wired to.
11. Delete the obsolete device.

Note: It is important to add the new device before deleting the old one. This is not just to allow easier duplication of the settings, but because it ensures that NWare will not allocate the same device ID to the new device as was allocated to the old device. Reusing the device ID can mean that corrupt settings from the old device are copied over to the new device.

12. Repeat steps 5-11 for the other obsolete devices.
13. Save the project.

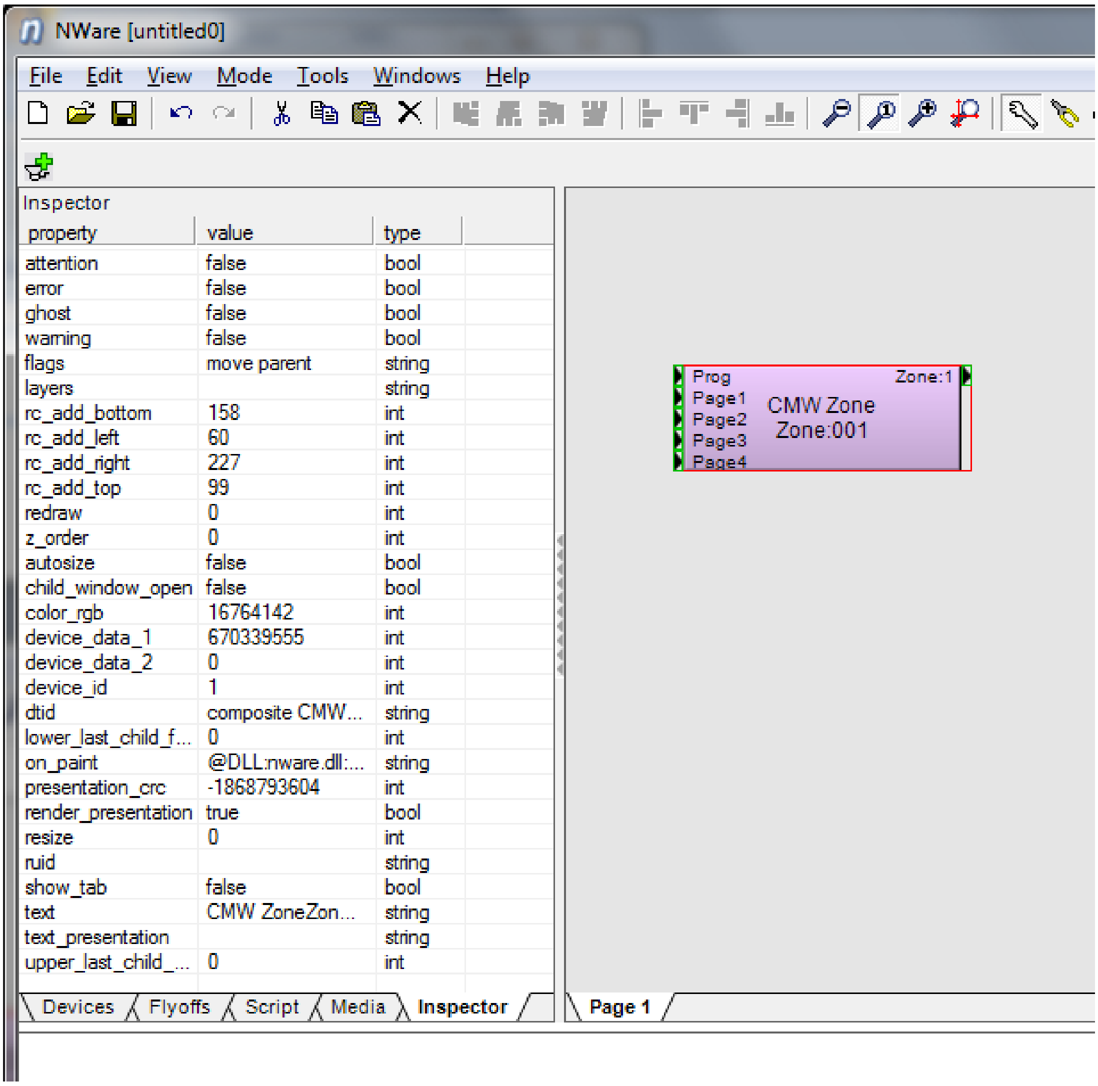
Copying the control surface settings between the old and new devices

1. Open the old version of NWare and load the old project.
2. If not already open, open the new version of NWare and load the new project.
3. Arrange the old version of NWare and the new version of NWare on the screen side-by-side.
4. Emulate the old project.
5. Emulate the new project.
6. In the old project, open the first device that you replaced in the new project.
7. Open the same device in the new project.
8. Copy the settings from the old project device over to the new project device:
   a) With the old project active, press CTRL+A, and then CTRL+C to select all the settings on the device page and copy them to the clipboard.
   b) Switch to the new project, and then press CTRL+V.
   c) Repeat steps a and b for the other pages in the device.
   d) Step through the pages on the devices in both versions of NWare to check the settings match.
9. Repeat steps 6 - 8 for the other replaced devices.
10. Save the new project.
11. Deploy the new project and check that it is functioning correctly.

Displaying object values using the Inspector

The NWare Inspector tab displays settings for objects on the design page. In the example below, the Inspector shows the settings for a Zone Paging Block.

Tip: The Inspector displays the RUID (Relatively Unique ID) for a control. This can then be used to reference the control using RATC RAW.



property	value	type
attention	false	bool
error	false	bool
ghost	false	bool
warning	false	bool
flags	move parent	string
layers		string
rc_add_bottom	158	int
rc_add_left	60	int
rc_add_right	227	int
rc_add_top	99	int
redraw	0	int
z_order	0	int
autosize	false	bool
child_window_open	false	bool
color_rgb	16764142	int
device_data_1	670339555	int
device_data_2	0	int
device_id	1	int
dtid	composite CMW...	string
lower_last_child_f...	0	int
on_paint	@DLL:nware.dll:...	string
presentation_crc	-1868793604	int
render_presentation	true	bool
resize	0	int
ruid		string
show_tab	false	bool
text	CMW ZoneZon...	string
text_presentation		string
upper_last_child_...	0	int

⏩ *To switch the Inspector on or off*

1. On the **Tools** menu, click **User Preferences**.
2. Click the **Advanced** tab.
3. Select or clear the **Show Inspector** check box.
4. Click **OK**.
5. Restart NWare.

Chapter 11

Administering nodes

In This Chapter

Updating firmware on MediaMatrix devices

You can update the firmware on several types of MediaMatrix devices, including NioNodes and nControl nodes, using NWare.

Notes:

- During the update process, the device will restart. If you are updating a node that is outputting audio, output will stop until the unit has restarted.
- To reduce the risk of compatibility problems, we recommend that all the hardware devices (NioNodes, nControl nodes etc.) on the network run the same firmware version, and that the version matches the version of NWare you are running.
- If you are using a Dante system, and you need to upgrade the Dante firmware, we recommend that you do this *before* you upgrade the NioNode firmware.
- Firmware on nTouch 60 nodes is managed from the control surface of each nTouch 60 device. For more information, see *Updating firmware on nTouch 60 nodes* (on page 284).

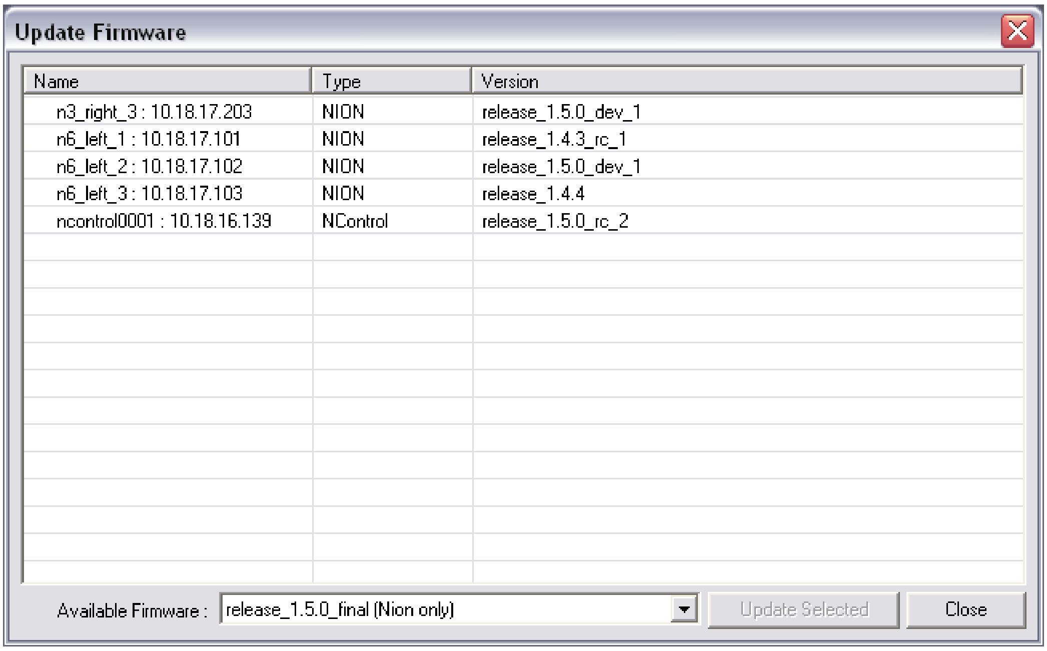
Each version of NWare includes firmware for all the MediaMatrix hardware devices it supports.

You can download the latest software and earlier versions of software from *http://www.peaveycommercialaudio.com/products.cfm/Software/NWare-Software/NWare-Software* (*http://www.peaveycommercialaudio.com/products.cfm/Software/NWare-Software/NWare-Software*).

To update firmware on MediaMatrix devices

1. In NWare, on the **Tools** menu, click **Update Target Firmware**.

   The **Update Firmware** dialog box is displayed.



Name	Type	Version
n3_right_3 : 10.18.17.203	NION	release_1.5.0_dev_1
n6_left_1 : 10.18.17.101	NION	release_1.4.3_rc_1
n6_left_2 : 10.18.17.102	NION	release_1.5.0_dev_1
n6_left_3 : 10.18.17.103	NION	release_1.4.4
ncontrol0001 : 10.18.16.139	NControl	release_1.5.0_rc_2

Available Firmware : release_1.5.0_final (Nion only)

The dialog box shows the devices NWare has discovered on the network and the versions of software they are currently running.

The **Available Firmware** list shows the latest version of firmware and older versions that are available on the NWare PC. The latest version of firmware will be selected automatically.

2. Click the first device you want to update.
3. Click **Select for update**.
4. Repeat steps 2 and 3 for the remaining devices.

   If you want to cancel the update process for a particular device, click the device and then click **Cancel Update**.
5. Click **Update Selected**.

   A warning dialog box is displayed.
6. Click **Yes** to continue with the update.

   The progress of the update will be shown as the firmware is downloaded to the units. This may take several minutes to complete.

   When the update has finished for a particular device, the word **success** will be shown next to it in the **Version** column and the device will be restarted. During the restart process, the device will not be listed in NWare.
7. Click **Close**.

Using NDP to discover network devices and assign network settings

In the NWare installation folder, there is a program called *ndp.exe*. It can be used to list devices on the network and assign them IP addresses. NDP uses *Neighborhood Discovery Protocol* (*http://en.wikipedia.org/wiki/Neighbor_Discovery_Protocol*), which is available with IPv6.

NDP can be used, for example, when the network settings assigned to a device do not match the scheme used with your network, and you need to assign new settings, or you do not know the IP address of a device and it does not have a front panel that displays the IP address.

When you use NDP, changes are first made locally (using the *setconfig* command), and then sent to the actual device (using the *commit* command). This allows you to preview the changes before they are made permanent.

Note: The current version of NWare does not include all the required components to run NDP. Before you use it, you must install a program called WinPcap, which is freely available from *http://www.winpcap.org/* (*http://www.winpcap.org/*).

⏩ *To use NDP to discover network devices and assign IP addresses*

1. Open Windows Explorer.
2. Navigate to the NWare installation folder.
3. Hold down SHIFT, right-click the NWare installation folder, and then click **Open command window here**.

   A Command Prompt window will be displayed.
4. Type `ndp`

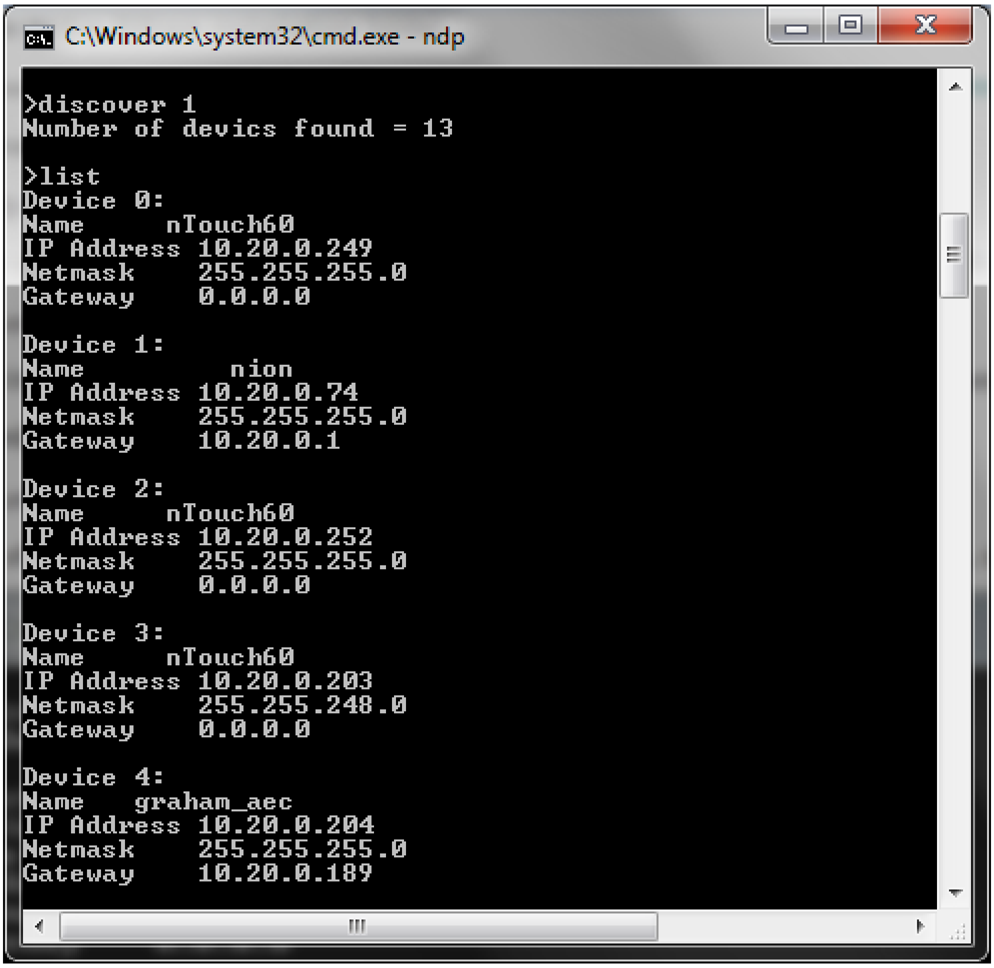
   The NDP program will start and a list of available network interfaces and commands will be displayed.
5. Type `discover <interface number>`

   where *<interface number>* is the number of the interface used for the control network.

The number of discovered devices will be displayed.

6. Type `list`

   Details of each of the devices will be shown, including the name, IP address, subnet mask and gateway.



C:\Windows\system32\cmd.exe - ndp

```
>discover 1
Number of devics found = 13

>list
Device 0:
Name      nTouch60
IP Address 10.20.0.249
Netmask    255.255.255.0
Gateway    0.0.0.0

Device 1:
Name          nion
IP Address 10.20.0.74
Netmask    255.255.255.0
Gateway    10.20.0.1

Device 2:
Name      nTouch60
IP Address 10.20.0.252
Netmask    255.255.255.0
Gateway    0.0.0.0

Device 3:
Name      nTouch60
IP Address 10.20.0.203
Netmask    255.255.248.0
Gateway    0.0.0.0

Device 4:
Name    graham_aec
IP Address 10.20.0.204
Netmask    255.255.255.0
Gateway    10.20.0.189
```

7. If you want to assign new network settings to a device:

   a) Make a note of the device number for the device you want to configure.

   b) Type `setconfig <device number> <name> <IP address> <Netmask> <Gateway>`

   Examples:

   ```
   setconfig 3 nTouch60 10.20.0.201 255.255.248.0 0.0.0.0
   setconfig 3 - - 255.255.248.0 -
   ```

   Note: You must specify a device number after the *setconfig* command, but you do not need to specify all of the other parameters – you can specify just the subnet mask, for example. You can substitute the hyphen (-) character for the other fields to leave the settings unchanged.

   c) Type `commit <device number>`

   The device will be configured to use the new settings. You may not be able to contact the device using a browser, for example, for a few seconds after the commit command has been specified.

NDP command reference

`commit <index>`	Commit changes specified using *setconfig* permanently to the specified device. **Note:** The device may automatically reboot after this command has been typed.
`configure <index>`	For nTouch 60 devices: Select configuration mode and display the configuration menu on the screen. For other devices: Temporarily apply changes specified using *setconfig* to the specified device. The settings will be used by the device until it is rebooted. After this, it will use the previous settings.
`discover <interface number>`	Discover devices on the network connected to the specified interface. The list of devices is automatically stored. When you first type *ndp*, a list of available interfaces is displayed and each is given a number. **Note:** Only NION and nTouch 60 devices can be discovered. You cannot discover nControls, for example.
`list`	Display all the discovered devices. Each device is given an index number, which is used to reference the device. If you have changed settings using *setconfig*, this command shows the updated settings, even if they have not been made permanent using the *commit* command.
`setconfig <index> <name> <IP address> <Netmask> <Gateway>`	Update the local (temporary) settings for the specified device. Use the hyphen (-) character to leave a setting as is. For example: `setconfig 1 ntouch60 10.20.0.202 - -` This changes the name and IP address, but leaves the subnet mask and gateway unchanged. **Note:** The changes are stored locally and not updated on the actual device until you use the *commit* command.

`showconfig <index>`	Display the local (temporary) settings for a device. You can use this command to check the settings you have specified using a *setconfig* command before committing them using the *commit* command.

Updating firmware on nTouch 60 nodes

When you want to update the firmware on nTouch 60 nodes, you must configure the nodes using the control surface of each corresponding nTouch 60 device in your NWare project. You cannot use the *Update Target Firmware* option on the NWare Tools menu.

You can either manually update the firmware on an nTouch 60 node or select an option that will update the firmware automatically, as soon as the node powers up and connects to the project.

Note: The manual firmware option installs the firmware included with the version of NWare you are running, even if it is earlier than the version currently installed on the nTouch 60. The automatic option will only install the firmware if it is more recent than the version the nTouch 60 is running.

To update the firmware manually

1. Create or load a project containing at least one nTouch 60 device and an nTouch 180 or nControl node.
2. On the toolbar, click the **Deploy** button.
3. Click the **Firmware Update** tab.
4. Click **Update**.

   The firmware included with NWare will be downloaded and installed on the nTouch 60.

To update the firmware automatically

1. Create or load a project containing at least one nTouch 60 device and an nTouch 180 or nControl node.
2. On the toolbar, click the **Deploy** button.
3. Click the **Firmware Update** tab.
4. Click **Automatic update**.

   If the firmware included with NWare will be installed if it is more recent than the version the nTouch 60 is running.

Displaying log messages for a node

MediaMatrix nodes store log messages that you can view by enabling polling of the node from within NWare.

To display log messages for a node

- On the **Remote Log** tab at the bottom of the NWare window, right click the node you want to poll in the **Targets Found** list, and then click **Enable Polling**.

Log messages already stored on the node and newly generated messages will be displayed.

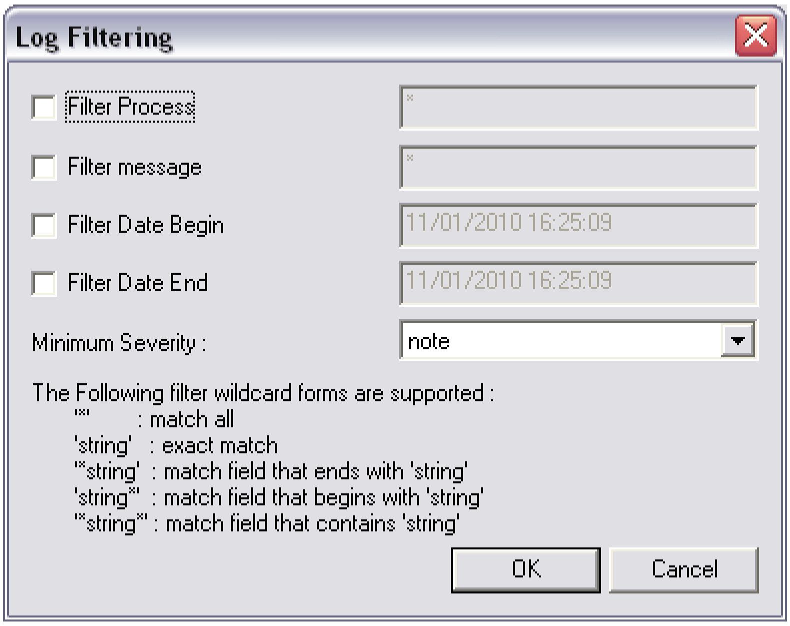
Tips:

- If you need to gather log files from NioNodes or other types of nodes on the network to send to MediaMatrix Technical Support, you can collect them together using a utility call Lumberjack. For more information, see *Gathering log files from MediaMatrix nodes using Lumberjack* (on page 286).
- If you are troubleshooting a problem on a NioNode and you want to view debug messages in the log (and not just the default messages), see *Including debug messages in the log* in the *NION Hardware Manual*.
- If there are nodes on the network that you want to use, but they are not listed on the **Remote Log** tab, there may be a problem with the configuration of Pandad. For more information, refer to the *Pandad Administrator Guide*.

To filter the display

1. On the **Remote Log** tab, right-click the log display, and then click **Filter**.

   The **Log Filtering** dialog box is displayed.



□ Filter Process *

□ Filter message *

□ Filter Date Begin 11/01/2010 16:25:09

□ Filter Date End 11/01/2010 16:25:09

Minimum Severity : note

The Following filter wildcard forms are supported :
'*' : match all
'string' : exact match
'*string' : match field that ends with 'string'
'string*' : match field that begins with 'string'
'*string*' : match field that contains 'string'

2. If you want to filter log entries based on the entry in the **Process** column, select the **Filter Process** check box, and then type a filter wildcard in the box.
3. If you want to filter log entries based on the entry in the **Message** column, select the **Filter Message** check box, and then type a filter wildcard in the box.
4. If you want to display only messages generated after a particular time, select the **Filter Date Begin** check box, and then change the date/time entry in the box.
5. If you want to display only messages generated before a particular time, select the **Filter Date End** check box, and then change the date/time entry in the box.
6. If you want to change the minimum severity level that a message must have in order to be displayed, click a new entry in the **Minimum Severity** list.

   The default setting is *note*.
7. Click **OK**.

Gathering log files from MediaMatrix nodes using Lumberjack

When you want to gather together the log files from several nodes, including NioNodes and nTouch 180 nodes, for example, you can use a utility called Lumberjack. This is included in each release of NWare.

Lumberjack gathers the log files together from each node on the local area network into a single zip file, which can then be *emailed* (*mailto:mmtechsupport@peavey.com*) to MediaMatrix Technical Support.

⏩ *To gather NioNode log files using Lumberjack*

1. Open a Command Prompt window.
2. Navigate to the NWare installation folder.
3. Navigate to the *Utilities* folder.
4. Type `lumberjack <file name>.zip` and press Enter.

> **Tip:** You can specify a file path before `<file name>.zip` to save the file to a different location.

5. If a **Windows Security Alert** dialog box is displayed, click **Unblock**.

   A number of messages will be displayed as Lumberjack retrieves the log files from the MediaMatrix nodes on the network and compresses them into a zip file. For example:

```
grabbing logs from nion_n1

grabbing logs from nion_n2

grabbing logs from ntouch180_nt1

compressing log files to nodelogs.zip...
```

> **Tip:** If the message *access denied* is displayed for a node, it means that the settings for *defaultuser* are preventing Lumberjack from gathering the log from the node. Connect to the node via the web interface and ensure that the *Log Access* privilege for *defaultuser* is set to *View and clear*. For a NioNode, see *Managing users* in the *NION Hardware Manual*. Settings for other node types are similar.

Using the Carpenter utility to merge NION log files

When you want to merge several individual NION log files into a single log file, use the Carpenter utility. This is located in the *Utilities* folder under the NWare installation folder. In the procedure below, it is assumed that you have already downloaded the log files.

⏩ *To use the Carpenter utility to merge NION log files*

1. Open a Command Prompt window.
2. Navigate to the NWare installation folder.
3. Navigate to the *Utilities* folder.
4. Type `carpenter <path>`, where `<path>` is the path to the NION log files, and press Enter.

   Here is an example:

```
C:\Program Files (x86)\MediaMatrix\NWare 2.0.0\utilities>carpenter
"C:\NION logs"
```

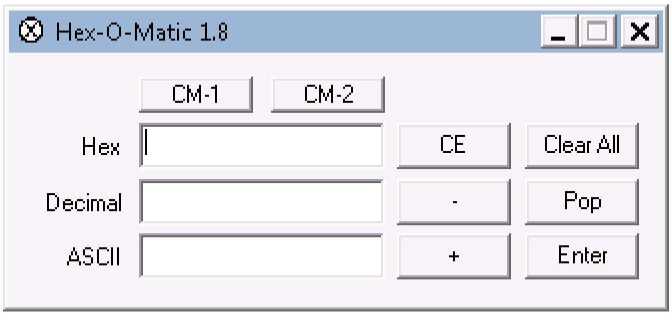
```
Reading 'C:\NION logs\NION n3 Log.log'
Reading 'C:\NION logs\NION n6 Log.log'
Sorting 3,874 entries
Writing merged output file 'C:\NION logs\MergedLogs.csv'
Finished
```

Using Hex-O-Matic to convert values

When you want to convert values between hexadecimal, decimal and ASCII, use the Hex-O-Matic utility. This is located in the *Utilities* folder under the NWare installation folder.

⏩ *To use the Hex-O-Matic utility to convert values*

1. Open Windows Explorer, and then navigate to the NWare installation folder.
2. Navigate to the *Utilities* folder.
3. Open *Hex-O-Matic*.



4. Type a value in the **Hex** or **Decimal** boxes, or type a character in the **ASCII** box.

   Hex-O-Matic converts the value or character, and displays the result in the other boxes. For example, if you type A in the ASCII box, the ASCII value of 65 (decimal) and 41 (hex) are displayed.

Using SNMP Blaster to control MediaMatrix devices using SNMP

SNMP Blaster allows you to read values from, and write values to, CobraNet MediaMatrix devices on the network.

⏩ *To use SNMP Blaster to control MediaMatrix devices using SNMP*

1. Open Windows Explorer, and then navigate to the NWare installation folder.
2. Navigate to the *Utilities* folder.
3. Open *SNMP_Blaster*.

Refer to *SNMP Blaster.pdf* in the *Utilities* folder for more information.

Appendix A

Keyboard accelerators

In This Appendix

File menu

New	Ctrl+N
Open	Ctrl+O
Save	Ctrl+S
Save As	Ctrl+Alt+S
Deploy	F9
Emulate	Ctrl+F9
Compile	Shift+F9

Edit menu

Undo	Ctrl+Z **Tip:** 100 undo levels are supported.
Redo	Ctrl+Shift+Z
Cut	Ctrl+X
Copy	Ctrl+C
Select Position / Paste	Ctrl+V
Paste	Ctrl+Shift+V
Duplicate	Ctrl+D
Duplicate Special...	Ctrl+Shift+D
Delete	Delete / Del
Select All	Ctrl+A
Find...	Ctrl+F

Align

Align Left	Ctrl+Shift+L
Align Top	Ctrl+Shift+T
Align Right	Ctrl+Shift+R
Align Bottom	Ctrl+Shift+B

Pack

Pack Left	Ctrl+L
Pack Top	Ctrl+T
Pack Right	Ctrl+R
Pack Bottom	Ctrl+B

Arrange

Bring To Front	Ctrl+]
Send To Back	Ctrl+[

View menu

Show/Hide Output Frame	F10
Show/Hide Utility Frame	F11
Show/Hide Status Bar	F12
Show/Hide Block Tabs	Ctrl+q

Zoom

Zoom Out	Ctrl+PageUp
Zoom Normal	Ctrl+home
Zoom In	Ctrl+PageDown
Zoom Fit All	Ctrl+0 (zero) **Note:** The zero key on the numeric keypad cannot be used to select this function.

Mode menu

Edit Mode	Ctrl+E
Wire mode	Ctrl+W
Gesture Mode	Ctrl+G
Paint Mode	Ctrl+P

Tools menu

Control Group	Ctrl+I
Control Ungroup	Ctrl+Shift+I
Lock	Ctrl+Alt+L
Unlock All	Ctrl+Alt+U
Replace Text	Ctrl+8
Insert waypoint	Ctrl+Shift+W
Change Flyoff Direction	Tab
Open Object	Enter
Display Object Properties	Alt+Enter
Display Block Properties	Alt+Shift+Enter
Device Placement...	Ctrl+Shift+Enter

Expression labeling

Expression Label Flyoff	Ctrl+Shift+F
Expression Label Name	Ctrl+Shift+E
Expression Label Alias	Ctrl+Shift+X

Control Nodes

Set Master Node for control	Ctrl+Shift+M
Set Slave Node for control	Ctrl+Shift+S
Set Peer Node for control	Ctrl+Shift+P

Windows menu

Next Window	Ctrl+Alt+N
Previous Window	Ctrl+Shift+N
Next Page	Ctrl+Tab
Previous Page	Ctrl+Shift+Tab

MediaMatrix®

A Division of Peavey Electronics Corp.

5022 Hartley Peavey Drive, Meridian Mississippi, 39305, USA

Phone: 866.662.8750

http://www.peaveycommercialaudio.com/products.cfm/mm/

Features & Specifications subject to change without notice