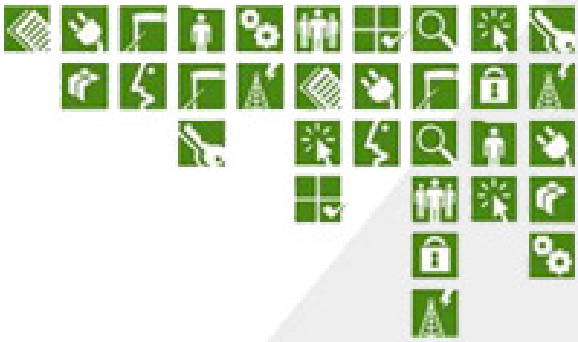


Proximetry AirSync 2.2

Practical User's Guide

Version 1.3.1

PROXIMETRY

Table of Contents

Preface

AirSync is a suite of network and device-management tools designed to manage wireless devices and traffic operating over multi-protocol wireless networks. With powerful support for service provisioning and reporting, AirSync simplifies your ability to manage service flows within the network, while reducing the operational costs associated with customer and server management.

Notice: The use of AirSync software and all other AirSync products is governed by the terms of your agreement(s) with Proximetry, Inc.

The use of Microsoft Virtual Earth software is governed by the terms of your agreement with Proximetry, Inc and Microsoft, Inc. By default this functionality is not enabled in AirSync software.

Intended Audience

This document is primarily intended for system administrators who will use AirSync to manage their wireless network environments. The document assumes that AirSync has already been successfully installed.

Product Version

The document corresponds to the AirSync version 2.2 product release.

Document Revision Level

Revision	Date	Description
Version 1.0.0	July 2008	Initial Release
Version 1.1.0	October 2008	The GUI Mechanics of Implementing QoS chapter updated
Version 1.1.1	November 2008	Using AirSync's Package Management System chapter updated. Appendix C added
Version 1.2.0	December 2008	Initial AirSync System Setup chapter updated. Screen Captures updated due to changes in GUI. Appendix A updated. Appendix D added

Version 1.3.0	January 2009	Appendix E and F added
Version 1.3.1	January 2009	Table 7 updated

Document Roadmap

The document begins with a brief overview of AirSync, then presents the main features of the newly redesigned AirSync version 2.2 graphical user interface. The new user interface is perhaps the biggest differentiator from the version 2.0 product release. The next section describes some initial system setup tasks. The last three sections are task-oriented guides to the three primary functional areas in AirSync: Implementing QoS, Using Package Distribution, and Monitoring the network.

Chapter	Description
Introduction to the AirSync System	Provides an overview of the AirSync system, lists the AirSync features, and describes the AirSync system architecture and configurations.
Exploring the AirSync User Interface	Introduces the user interface and the basic skills needed to use and manipulate the system.
Initial AirSync System Setup	Provides instructions for AirSync to record and use the location for distributed components, for specifying preferences, and for defining confirmation messages and context-sensitive menus for third-party tools, and for registering devices.
Using AirSync to Implement Quality of Service (QoS)	Gives the details of implementing a traffic management system to maximize the QoS with AirSync.
Using AirSync's Package Management System	Gives instructions for system administrators to systematically define and distribute items to managed nodes.
Using AirSync to Monitor the Network	Describes AirSync's variety of network monitoring and mapping functions. The simplest way to invoke them is to right click on a device and select a monitoring function from the context-sensitive menu.
Appendix A. Item Descriptions for Tools – Options	Reference that describes AirSync tools.
Appendix B. Item Descriptions for Tools – System Configuration	Reference that describes system configuration tools.
Appendix C. AirSync Preinstallation Requirements	Reference that describes system preinstallation requirements.

Appendix D - Example AirSync configuration for Wireless ISP scenario	Reference that describes examples AirSync configuration for Wireless ISP scenarios.
Glossary	Defines terms used in AirSync.
Index	Reference to AirSync for finding information.

Document Conventions

This guide uses the following typographic conventions:

Convention	Description
Bold	Text on a window, other than the window title, including menus, menu options, buttons, and labels.
Italic	Variable.
`screen/code`	Text displayed or entered on screen or at the command prompt.
boldface screen font	Information you must enter is in boldface screen font.
< *italic screen* >	Variables appear in italic screen font between angle brackets.
[ ]	Default responses to system prompts are in square brackets.

This guide uses icons to draw your attention to certain information. Warnings are the most critical.

Icon	Meaning	Description
	Note	Notes call attention to important and/or additional information.
	Tip	Tips provide helpful information, guidelines, or suggestions for performing tasks more effectively.
	Caution	Cautions notify the user of adverse conditions and/or consequences (e.g., disruptive operations).
	WARNING	Warnings notify the user of severe conditions and/or consequences (e.g., destructive operations).

Getting Help

If technical support is needed, please gather as much information about the problem as possible, including:

- The circumstances surrounding the error or failure.
- The exact content of any error message(s) displayed.

The Proximetry Customer Service Department can be reached by email at support@proximetry.com Monday through Friday between the hours of 5:30 A.M. and 6:00 P.M. Pacific Time.

Customer Service can receive attachments as well as messages via email.

Introduction to the AirSync System

About AirSync

AirSync consists of a suite of tools that simplify the tasks of managing a complex wireless network and optimizing the traffic flows within the network. AirSync helps reduce the operational costs of customer and server management. In simple terms, AirSync is a tool suite that allows an organization to articulate business rules or policy governing the use of its managed wireless network in a manner that bests suits that organization's unique needs.

AirSync is a Distributed Software Product

AirSync consists of three types of software that work together to bring order and control to wireless networks:

- The front-end or **Graphical User Interface** (GUI). This is the piece system administrators interact with.
- A set of **server components**, responsible for storing the organization's business rules (policy), monitoring the managed network in near-real time, making adjustments based on organizational policy and in response to various trigger events as they occur on the managed network.
- A set of client components or **agents** that run on managed network devices. These agents ensure that the organizational policy is cohesively implemented in the managed network and report status back to the server components.

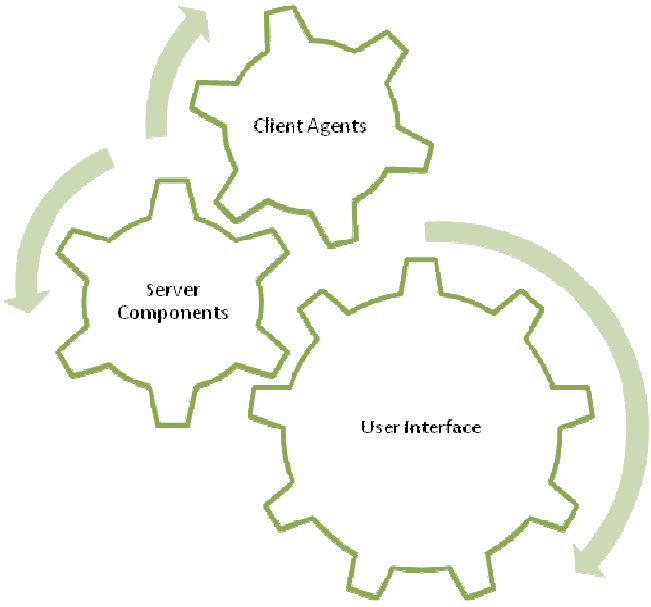


Figure 1. AirSync is composed of three types of software

What Does AirSync Version 2.2 Do?

AirSync has many useful features, but they can be broken down into three main functional groups:

- **QoS or Traffic Shaping**. This means providing differentiated service to users that ensures the right traffic gets through the network at the right time according to the organization's business rules.
- **Package Management**. This is primarily a system management function for uploading new firmware and configuration files to managed devices, but it could be extended as a general content delivery system.
- **Network Monitoring and Feedback**. AirSync also provides visualization and reporting capabilities for showing network devices on a map, generating a logical topology diagram of network connectivity, and charting network statistics (such as signal quality and throughput).

Effective use of AirSync involves the following steps:

1. First, **determine the organization's policy goals**. Which traffic is most important under what circumstances? How should the system arbitrate bandwidth allocation decisions during times of congestion? How should users and traffic flows be prioritized?
2. Next, use the AirSync GUI to **define the organization's usage policy in AirSync**. The policies will be stored and retrieved by the service components and propagated down to the managed devices.
3. Then, use the AirSync GUI tools to **monitor and adjust network behavior**. The reporting and visualization tools can help you verify how well the managed system is implementing the organization's traffic policies and identify adjustments and enhancements to improve network use. Over time, the reporting tools enable you to spot trends - proactively anticipate and solve network issues before they turn into bigger problems.
4. Periodically, use AirSync's package distribution functionality to **upload new firmware or configurations** onto the managed network devices.

A Few Words on the Architecture

At the core of its server components, AirSync uses JBoss, a Java-based, cross-platform application server. The system uses MySQL, a relational database, and Enterprise Java Beans (EJB) to store business logic. AirSync also has a server component, RMServer, that communicates with the software agents resident on managed client devices. While this is probably more detail than needed by most system administrators, a key characteristic is that AirSync uses a web-services-based architecture.

As a result, AirSync is highly customizable and scalable, especially with respect to:

- **The location of server components**. All the server components can run on the same machine. However, the database component can be moved to a separate machine, for example, to improve scalability or security.
- **The relational database used**. By default, MySQL is used, but another relational database product could potentially be used instead.
- **Custom User Interfaces or gateways to third party applications**. While AirSync's redesigned user interface is quite powerful, much of the AirSync functionality can be accessed via web-services-based interfaces to other systems. For example, one Proximetry customer has invested a significant amount of development, training, etc. implementing a custom management system. The web-services-based architecture allowed the existing management system to interface with AirSync.

Other Network Management Tools

AirSync is a sophisticated product that brings order and control to the wireless networks it manages, but there are some tools it does not replace. AirSync is not inherently an:

- Asset management and tracking system
- Incident management (trouble ticket) system, like Remedy
- SNMP network management framework, like HP Openview, although some devices are managed and monitored using SNMP in limited scope.
- Firewall or intrusion detection system
- Network Sniffer or traffic analysis system

However, AirSync's web-services-based architecture gives it flexibility for integrating/interfacing with other third-party network management and control systems. It has been successfully integrated with other systems and it is easy to imagine useful functional pairings.

For example, pairing AirSync with an incident management (trouble ticket) system could be useful. AirSync could periodically report and store signal quality for one or more key network devices. If the signal degraded below a threshold value, AirSync could interface to the incident management system and automatically generate an incident ticket to dispatch a response team to investigate the issue.

If you can think of special interfaces that would be valuable for your organization, contact Proximetry and tell us about your ideas or special needs.

Mentally Decouple the User Interface from the Server Components

AirSync's user interface doesn't necessarily run on the same host machine as the other components. You can load AirConsole.exe on other management workstations as appropriate for organizational needs. In fact, if Linux is used on the platform(s) hosting the AirSync server components, the AirSync UI must run a separate machine, because it must be hosted on a Windows XP® platform running the Microsoft® .NET connection software framework.

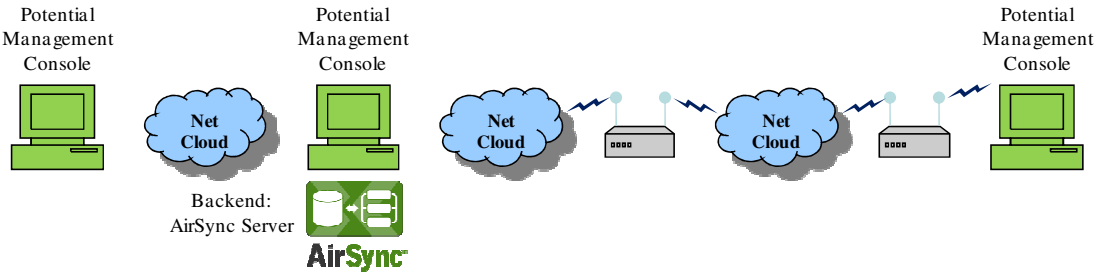


Figure 2. AirSync's Management GUI and Server components can run on different hosts

Develop and Use Consistent Naming Conventions

As you gain familiarity with AirSync, the need for consistent naming conventions will become apparent to you, but it is generally helpful to **implement a consistent naming convention** for objects created in the AirSync GUI. In general, it's easy to rename objects in the AirSync system, if you need to make adjustments.

For example, when registering new devices in a municipal wireless network managed by AirSync, it may be helpful to name all devices mounted on lampposts with an "LP_" prefix followed by the intersection name, and all mobile devices with the prefix "MOB_" followed by the mobile unit number. This will make sorting and searching operations easier. The same goes for naming other objects in AirSync: Groups, Roles, Services and so on. Don't worry, these items will be introduced and discussed in more detail later in the document.

Can AirSync Manage Devices without AirSync Agents?

Yes. AirSync makes software agents that are part of the firmware for multiple wireless network devices, and these agents interact with the server components to implement the organizations business policy. However, the AirSync agents on some of the devices have enough intelligence to manage the characteristics of the other network devices connected to them on the network. For example, it is possible to shape traffic for a laptop PC connected to a network device managed by AirSync, even without an AirSync agent running on the PC host.

However, AirSync can manage devices with AirSync agents more intelligently than those that lack agents, and agents have been developed for a variety of devices ranging from small handheld devices to a variety of radio devices from different vendors.

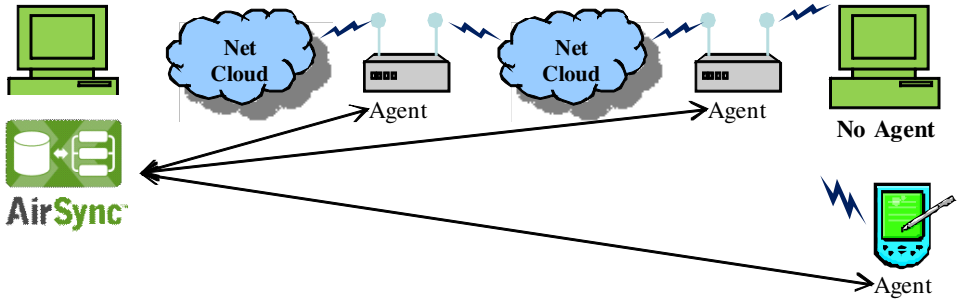


Figure 3. AirSync Manages Devices with agents and devices without agents

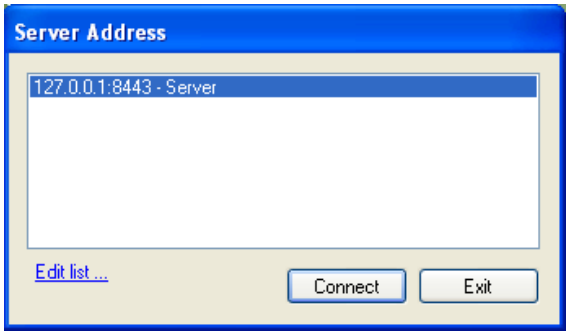
Exploring the AirSync User Interface

The purpose of this section is to introduce AirSync's redesigned user interface and develop the basic skills and techniques needed to use it effectively. The emphasis of this section is understanding how to manipulate the various GUI objects, rather than fully understanding what each particular item means or does.

Associating the AirConsole with the AirSync Server

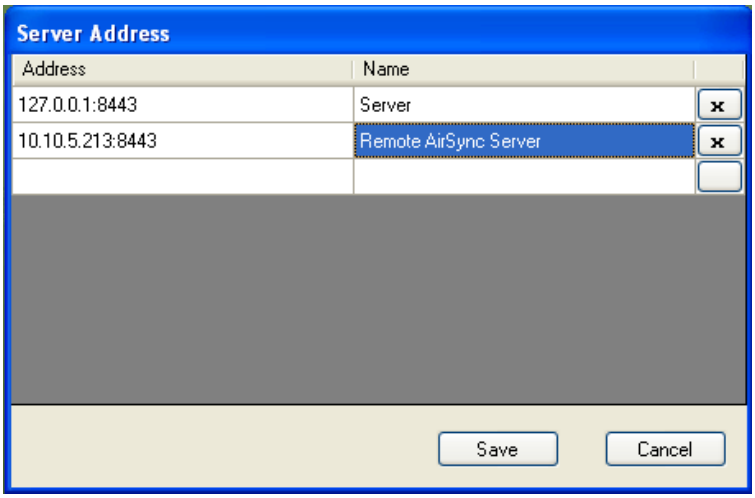
The AirSync GUI program AirConsole.exe will read configuration information from a file named "airconsole.exe.config" located in the same directory as the executable and from "ServerAddresses.xml" file located in "Documents and Settings" directory. The config file contains XML text that specifies configuration parameters for the GUI program and ServerAddresses.xml contains only list of IP addresses of AirSync servers.

After running the AirConsole, it is important to select **the correct AirSync server** it will be used to manage. Default setting is 127.0.0.1:8443 which means that AirConsole will try to connect with AirSync server installed on the same host as it is shown in Screen Capture 1.



Screen Capture 1. Server Address list

To associate the GUI to a specific AirSync server, you must furnish the correct IP address of the AirSync server you intend to manage by appropriately editing the "ServerAddresses.xml" file which is possible after pressing **Edit list...** option.

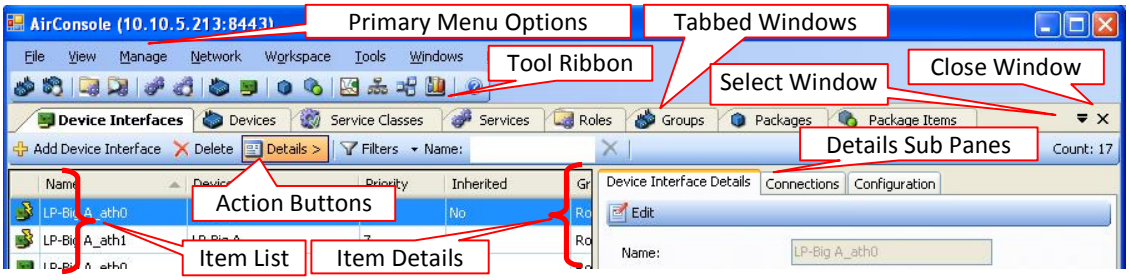


Screen Capture 2. The list of AirSync servers saved by the user

When you set proper values to **Address** and **Name** fields press **Save** button to return to the list of AirSync servers. Notice also that the value of **Address** field include a port number, such as :8443 in the example above. This is the port on which the AirSync application server (JBoss) listens. You should modify the port values appropriately if the JBoss installation on your AirSync platform has been modified to listen on a different port, but for most installations the default value :8443 is correct.

Basic GUI Layout

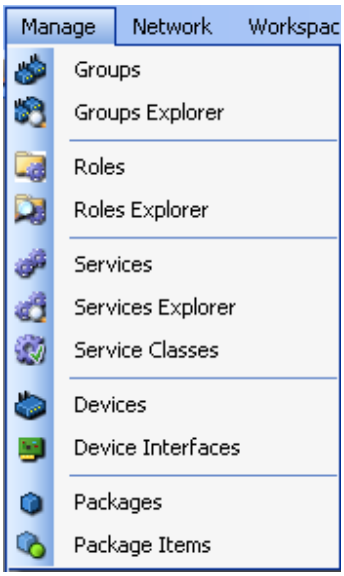
Screen Capture 3 shows the basic AirSync GUI layout.



Screen Capture 3. The Basic AirSync GUI layout

The GUI has eight primary menu items, but administrators will frequently access the items available on the "Manage" submenu shown in Screen Capture 4. Notice the tool ribbon immediately below the primary menu items in Screen Capture 3 contains quick access icons for most of the items also available from the Manage menu, as well as a few items from the Network menu. Only the Service Classes item on the Manage menu do not have quick access icons on the tool ribbon.

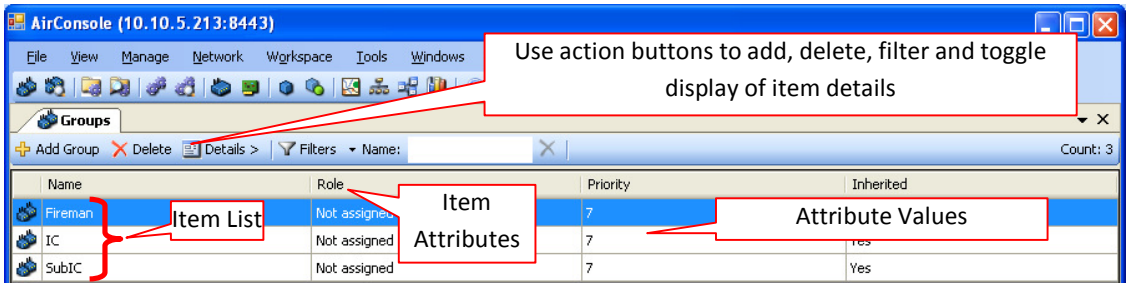
As you can see, there are quite a few items to manipulate and because several items can be manipulated at the same time, each in its own window, the screen can fill up quickly. The next few sections discuss how to manipulate the on-screen items, and how to save and restore user workspaces, once they have been created to suit a user's preferences.



Screen Capture 4. The AirSync Manage menu

Item List, Item Details Metaphor

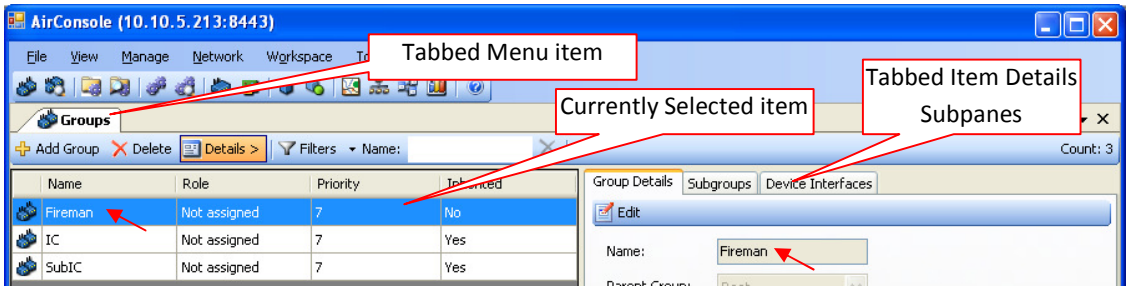
Most of the items from the management menu present a simple list of items when initially opened, as shown for the **Groups** item in Screen Capture 5. Use Action buttons to perform operations on the item list such as adding, deleting and filtering items. The item list begins with a row of column headings that label which attribute values will be displayed for each item in the list. The list displays a row for each item containing a record of the items' attribute values.



Screen Capture 5. Item List for Groups

Clicking the **Details** action button or double-clicking a specific item in the list, in this example the group named "Fireman," toggles the display of a detailed information sub pane (to the right) for the selected item as shown in Screen Capture 6.

Notice as you select different items in the primary item list pane (on the left), the information displayed in the item details sub pane (on the right) is updated to accurately reflect the detailed information for the item selected in the left pane. Double-click an item in the item list pane again to toggle off the display of the item details sub pane.

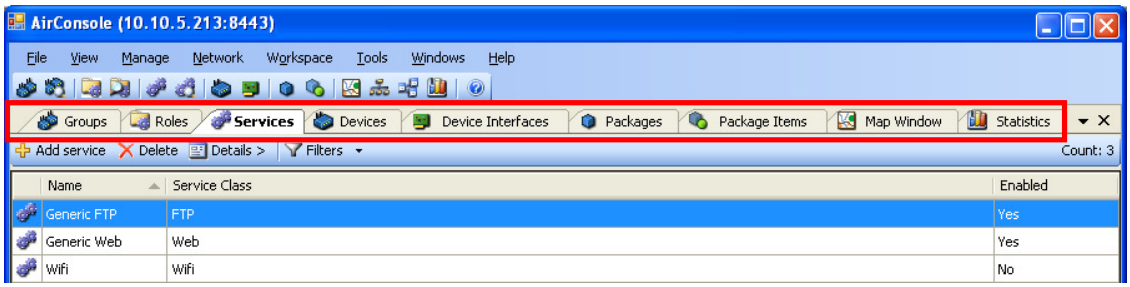


Screen Capture 6. Item details (right pane) for "Fireman" item in "Groups" Item List (left pane)

Tabbed Window Metaphor

Notice also in Screen Capture 6 the use of tabs to save screen real estate but allow users to see that there are more menu items open or more information sub panes available. Tabs also allow users to switch rapidly between different items when multiple windows are open, and between detail sub panes for items that have distinct groups of related details. In the case of tabbed sub panes such as **Group Details**, **Subgroups** and **Device Interfaces** above, each tab will show detailed information related to the item currently selected in the item list pane on the left, in this example, the group named "Fireman."

Managing Multiple Window Regions



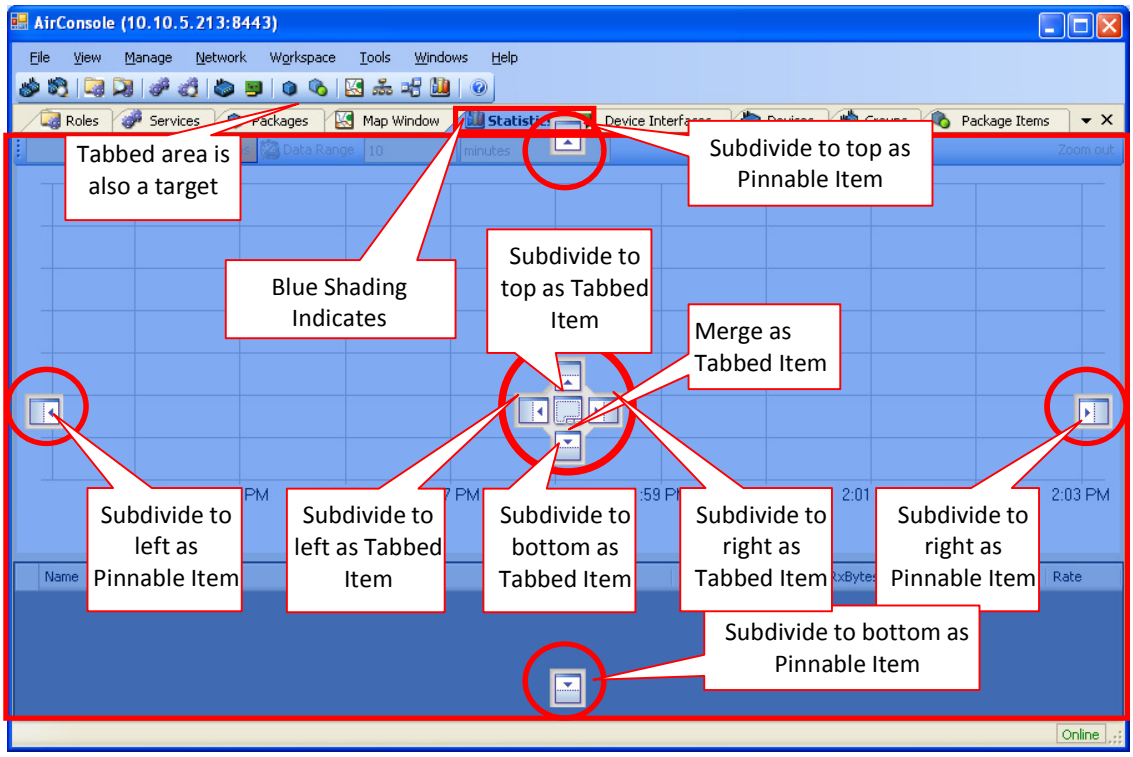
Screen Capture 7. Multiple Tabbed Items are Open

As previously mentioned, and shown in Screen Capture 7, users can open multiple items and switch between them by selecting the tab corresponding to the desired item. It is also possible to subdivide the window into multiple regions (each of which can have multiple tabs), reorder the tabs, undock the tabbed items into floating windows, and re-dock GUI items on the top, right, left, or bottom portion of the window region. The following sections cover these operations.

Moving GUI Objects by Dragging and Observing Visual Cues

GUI objects can be moved around by direct "click and drag" or "drag 'n' drop" manipulation. There are a variety of different behaviors that will be explained below, but the key concept is to manipulate the intended object by dragging it, and observing the visual feedback the GUI provides indicating the state of the operation in progress.

Screen Capture 8 shows the visual feedback cues that occur as a user clicks (and holds) on a tabbed item, before beginning a drag operation. The important cues in the screen capture have been annotated. Learning to recognize and react to the visual feedback is the key to manipulating GUI screen objects effectively.



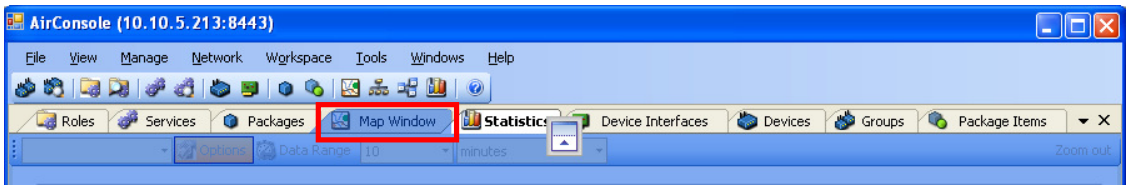
Screen Capture 8. Visual Feedback Cues when Selecting a Tabbed Item

As an item is selected, look for a blue shaded region (annotated above with red rectangle). This indicates the current destination of the operation. In the example above, the shaded region indicates that the tabbed "Statistics" item is docked together with all the other tabbed items.

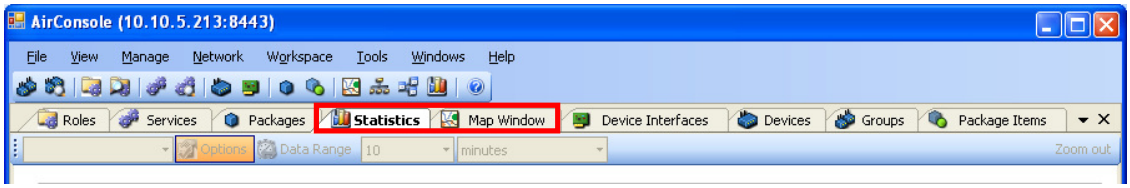
The on-screen controls annotated with red circles represent special targets where the item could be dragged to create a different user interface experience. The group of targets in the center is available for most items (except **Services Explorer**), but the targets on the far top, bottom, left and right only appear when manipulating pin-able items, such as the explorer items (**Groups Explorer**, **Roles Explorer**, **Services Explorer**) and the more graphical items (**Map Window**, **Network Diagram**, **Network Navigator**, **Statistics**). These items will be explained in greater detail later.

Reordering Tabbed Items

Users can reorder the sequence of tabbed items by dragging them. The sequence of Screen Capture 8, Screen Capture 9 and Screen Capture 10 shows the **Statistics** item being moved to the left of the **Map Window** item. Partial images of the final two screen captures have been used to minimize space, emphasize the visual cues (the blue shading moves from the tabbed item **Statistics** in Screen Capture 8 to the tabbed item **Map Window** in Screen Capture 9 indicating the ending destination), and to emphasize the final result (tabbed item **Statistics** is now to the left of tabbed item **Map Window** in Screen Capture 10.



Screen Capture 9. Drag selected item to the left, highlighted tab changes to indicate new position

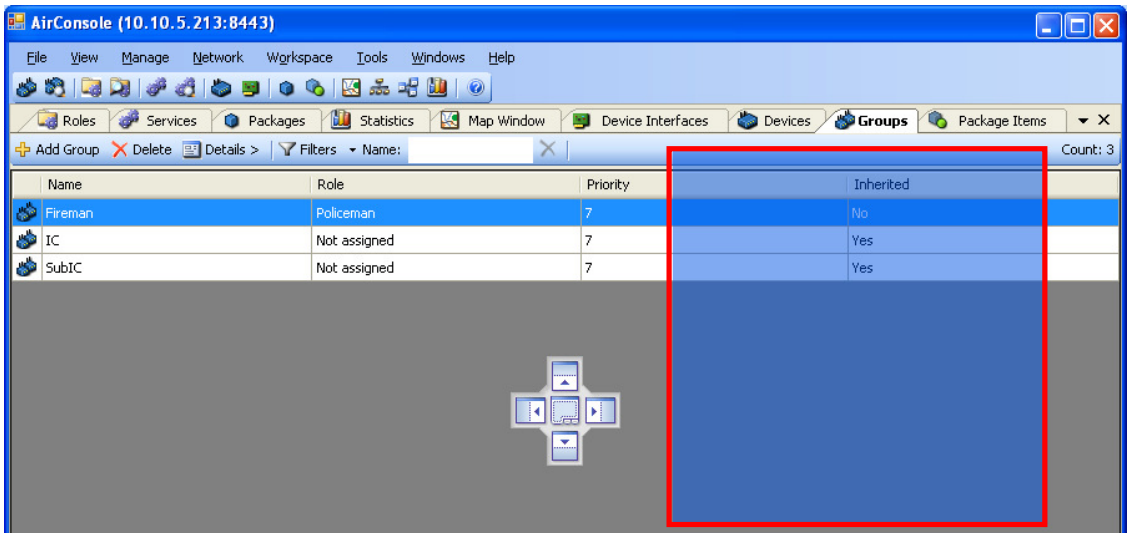


Screen Capture 10. After release, "Map Window" item has been moved to the left of "Statistics" item

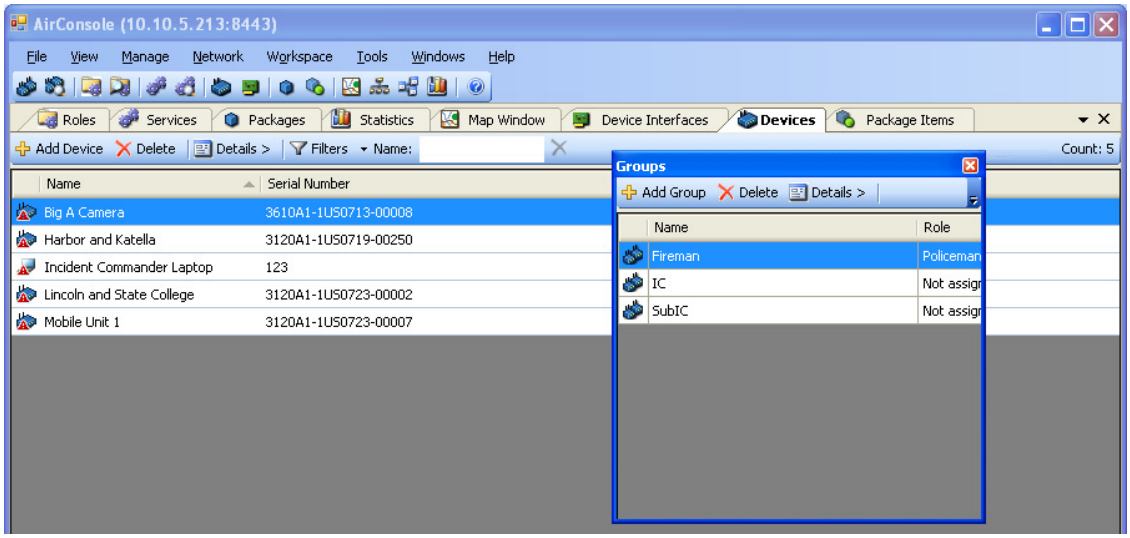
Moving Tabbed Items to Floating Windows

Users can move items to floating windows in a similar fashion. The key is to look for the visual cue indicating a floating window destination. Unlike the visual cues shown in Screen Capture 8 that appear upon clicking and holding the item to be moved, this cue won't appear until after starting the drag operation (moving the mouse) as shown in Screen Capture 11. Screen Capture 12 shows the final result after releasing the drag operation. The floating window can be resized to suit the user's preferences.

Moving an item to a floating window can be useful, for instance on workstations that have multiple monitors available, enabling the user to selectively split AirSync display items between the available monitors. In a network operations center (NOC) setting, the **Map Window**, **Statistics** and **Network Diagram** items are especially well suited for display on separate large screen monitors typically found in NOC environments.

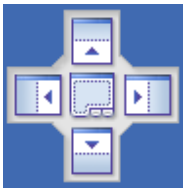


Screen Capture 11. Dragging the "Groups" item to produce a floating window cue



Screen Capture 12. "Groups" item in floating window upon release

Moving Items to Different Window Regions



Users can move items to different window regions and subdivide windows by using the primary AirSync GUI item placement control, shown in Screen Capture 13, as a target for drag 'n' drop operations.

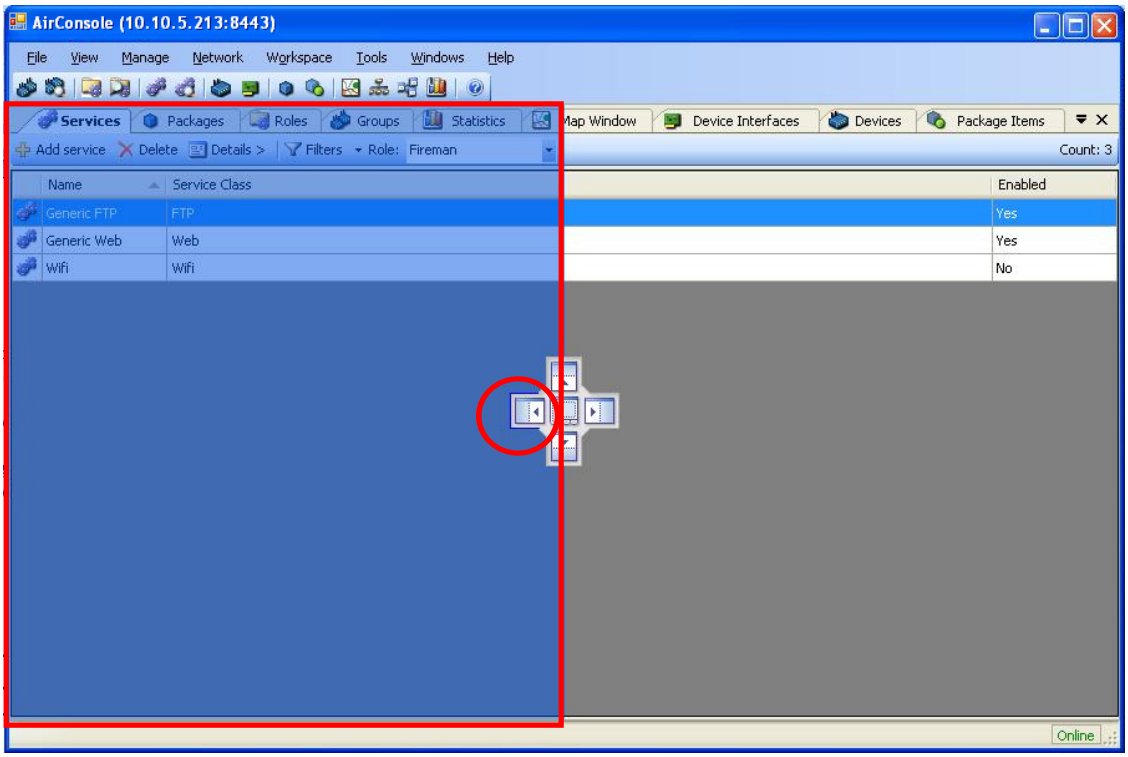
Screen Capture 13. The primary AirSync GUI item placement controls

Items can be moved into a new subdivided window region to the left, right, top or bottom of the original window region by using the left, right, top, or bottom portion of the control as a target.

Items can be merged back to the main tabbed area of another window region by using the central portion of the control as a target.

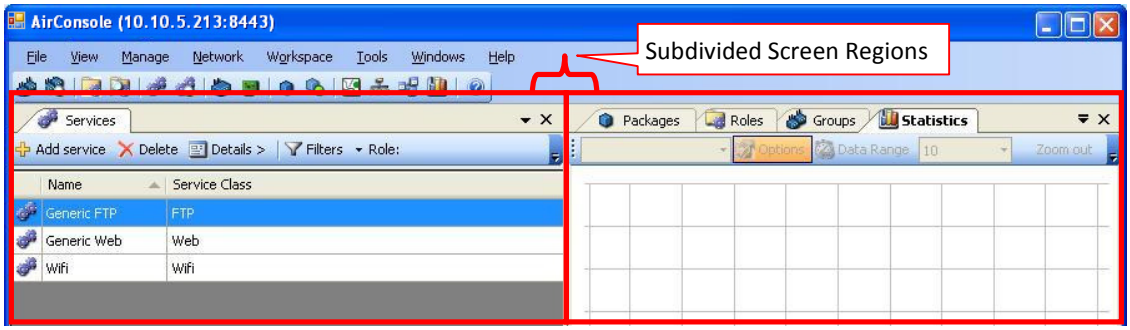
Screen Capture 14. AirSync GUI item placement targets

While dragging the item over the target, look for the appearance of a blue shaded region to indicate the destination screen location for the item. The annotation in Screen Capture 15 indicates a "Subdivide to Left as Tabbed Item" operation.



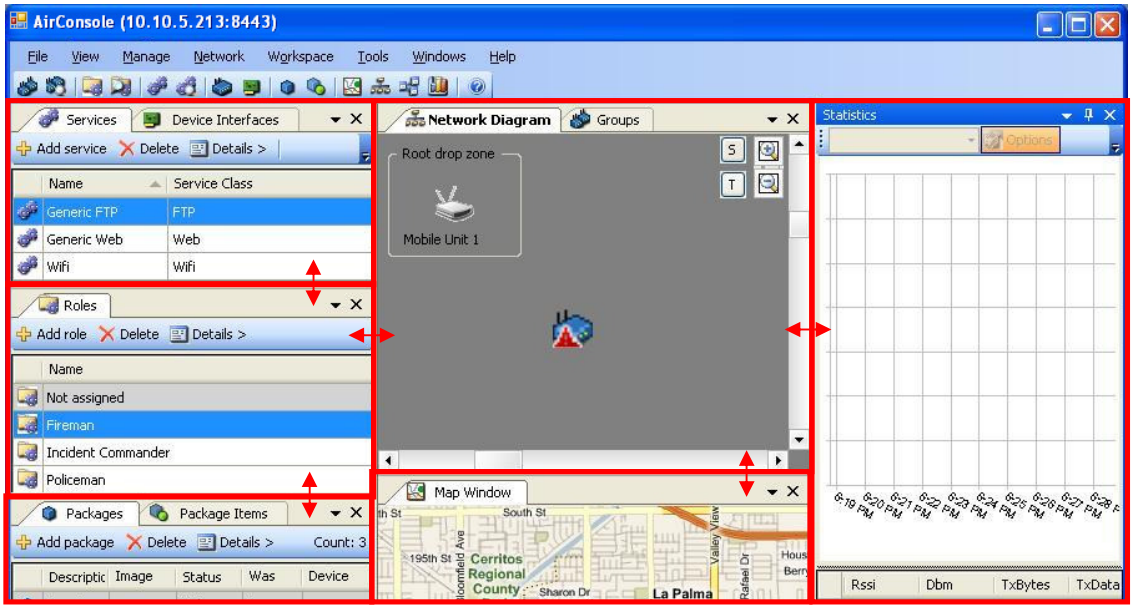
Screen Capture 15. Dragging the "Services" tab over the left portion produces a visual cue

It may seem awkward the first couple of times, but after a few tries it is easy to get the hang of the operation. Screen Capture 16 shows the "Services" item in a subdivided screen region to the left of the original after completing the previous drag operation.



Screen Capture 16. The "Services" item in a subdivided screen region to the left of the original

Screen Capture 17 shows a complex window that has been subdivided many times. The individual regions can be resized by dragging on the borders between the regions.



Screen Capture 17. Resizing regions in a complex, subdivided window

Hints for Manipulating GUI Objects

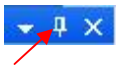
Here are a few hints to note when manipulating GUI objects:

- To drag an item: select it by clicking on its tab or by its title bar if it is a floating window. Otherwise, the item might not appear to be drag-able.
- If you have difficulty getting the target control(s) to appear, try selecting the title bar or tab from a slightly different location.
- Not all target controls will appear for each GUI object.

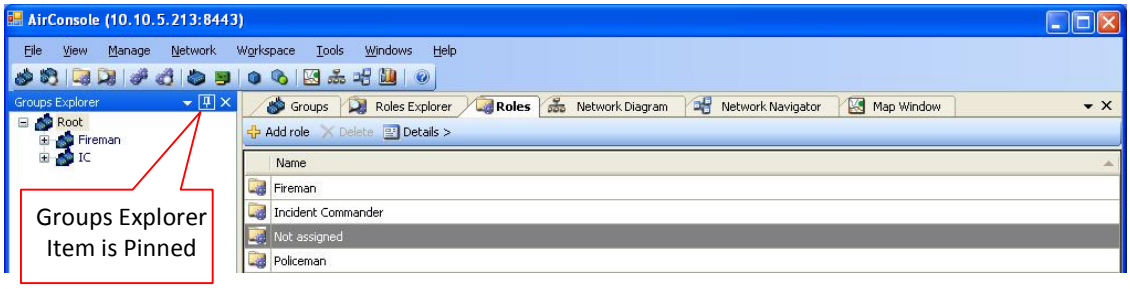
- To move an item back to the tabbed area, use the icon with tabs or just drag the item to a tabbed area. If you drag the item to a tabbed area, you will see a visual indication of the tab position corresponding to destination location.
- You can drag a floating window onto another floating window. If you do so, you may appear to "lose" one of the windows. To find the "lost" item, look for tabs appearing at the bottom of the window.
- If it gets confusing, try closing and reopening some items.
- To move an item that has been unpinned, you must first pin it again.

Pinning and Unpinning Items to Toggle the Auto-hide Feature

The GUI objects corresponding to the explorer items (Groups Explorer, Roles Explorer, Services Explorer) and the more graphical items (**Map Window**, **Network Diagram**, **Network Navigator**, **Statistics**) can be pinned and unpinned. When unpinned, the window for the item will automatically hide near the left, right, top, or bottom edge of a screen region to conserve screen real estate. The corresponding edge will display a visual cue indicating that the hidden window will automatically display by hovering the mouse over or clicking on the cue.

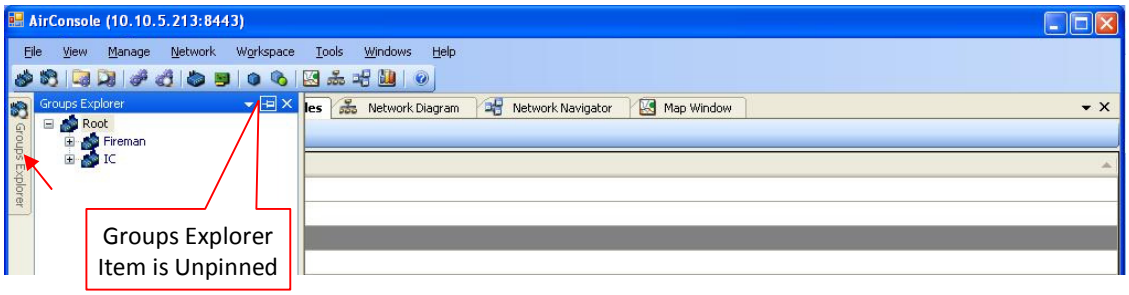


Screen Capture 18 shows the **Groups Explorer** item subdivided as a pinned window to the left. Notice the pin detail in the left screen region. To toggle the pinned status, click the pin icon.



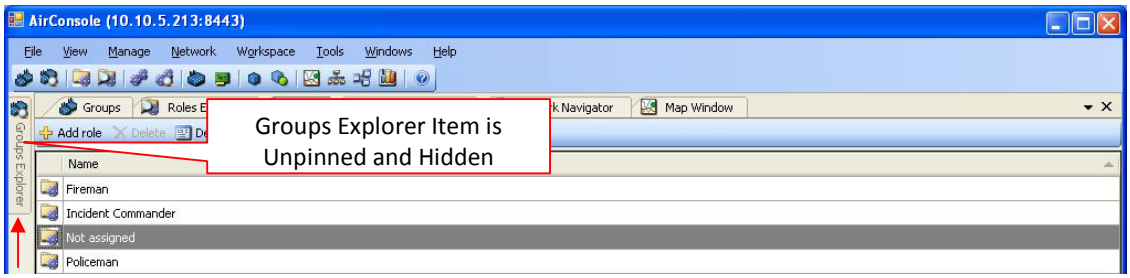
Screen Capture 18. "Groups Explorer" item pinned on the left

Screen Capture 19 shows the **Groups Explorer** item unpinned but still displayed to the left. Notice the pin detail in the left screen region.



Screen Capture 19. "Groups Explorer" Item unpinned, but displayed on the left

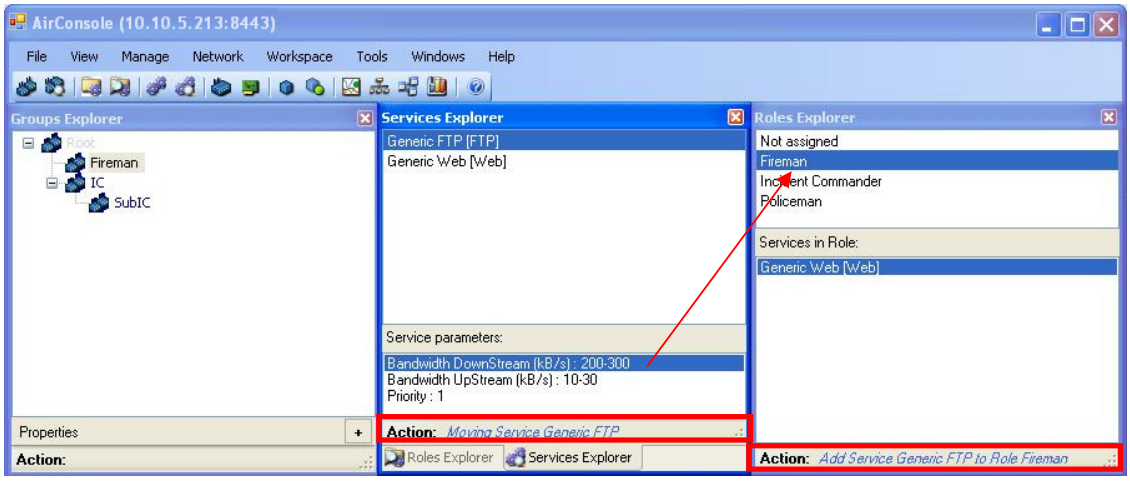
Upon clicking in a different window region, the unpinned **Groups Explorer** item will automatically hide itself as shown in Screen Capture 20. To redisplay the hidden item, hover the mouse over and/or click on the visual cue on the left edge indicating the item is unpinned.



Screen Capture 20. "Groups Explorer" item unpinned and hidden on the left

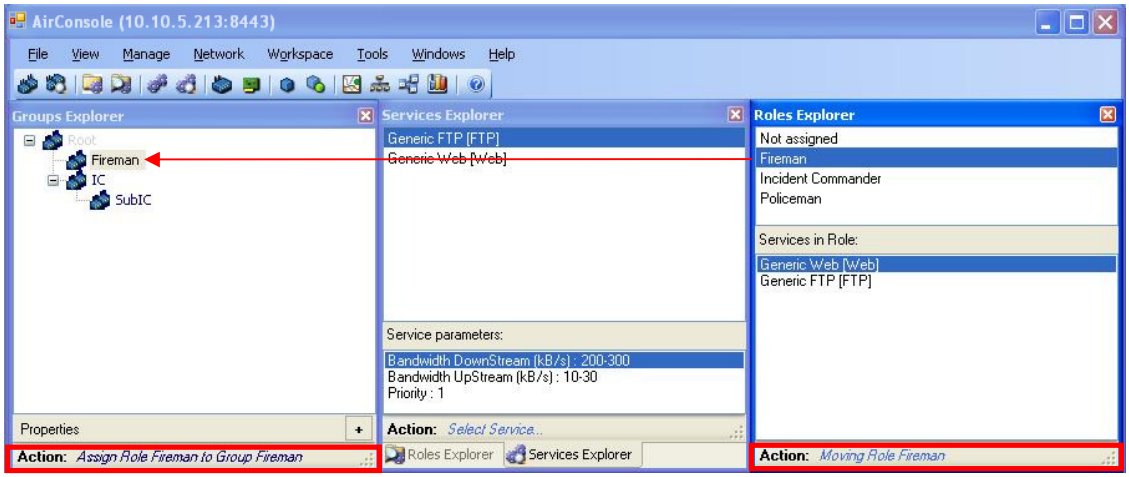
"Drag 'n' Drop" Operations with the Explorer Windows

To facilitate assignment actions, several AirSync GUI objects support drag 'n' drop mouse operations. The explorer items (**Groups Explorer**, **Roles Explorer**, **Services Explorer** support "drag 'n' drop" operations for appropriate items selected from other GUI objects. For example, to quickly assign a service to a role, using the direct, graphical drag 'n' drop object manipulation paradigm, drag a service item from the **Services Explorer** GUI object and drop it on the appropriate role in the **Roles Explorer** GUI object. Screen Capture 21 shows all three of the explorer windows as well as the in-progress result of dragging the **Generic FTP** service item from the **Services Explorer** GUI object to the "Fireman" role item in the **Roles Explorer** GUI object. Upon release, the service will be assigned to the role.



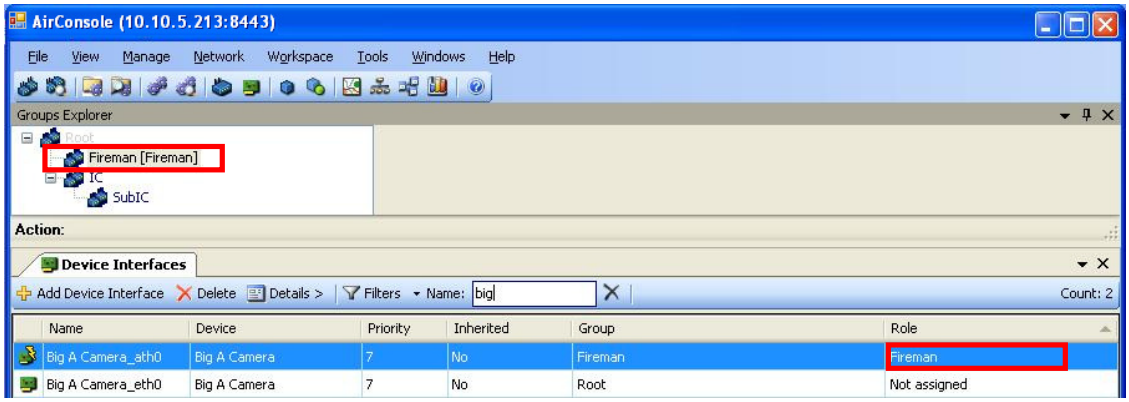
Screen Capture 21. Dragging and Dropping a Service from "Services Explorer" to "Role Explorer"

Screen Capture 22 shows a similar in-progress result of dragging the "Fireman" role item in the **Roles Explorer** GUI object to the "Fireman" group item in the **Groups Explorer** GUI object. Upon release, the role will be assigned to the group.



Screen Capture 22. Dragging and Dropping a Role from "Role Explorer" to "Groups Explorer"

Screen Capture 23 shows the result of dragging the device interface item "Big A Camera_ath0" and dropping it on the group item "Fireman" in the **Groups Explorer** GUI object. Note the text "[Fireman]" in the **Groups Explorer** region immediately to the right of the group item named Fireman. This is a visual cue that the "Fireman" group item has been assigned the role item named "Fireman" (inside the square brackets) as the result of the previous drag 'n' drop operation shown in Screen Capture 22. Notice also the **Device Interfaces** item named "Big A Camera_ath0" now shows a value of "Fireman" for its role attribute indicating the successful completion of the operation shown in Screen Capture 23.

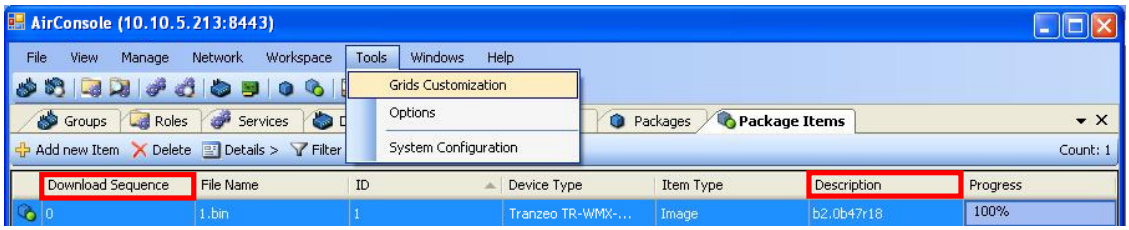


Screen Capture 23. Dragging and Dropping a device interface item to a group item

In general, the explorer windows act as destination targets for the drop operation, but you can also select the source object from non-explorer windows, for example directly from the **Groups** window area, if desired.

Customizing Item List Grids

For objects that display as item lists, it is possible to customize which item attributes are displayed, what order they are displayed in and how the list will be sorted. Screen Capture 24 shows how to invoke the **Grids Customization** item from the **Tools** menu and the original **Package Items** list before customizing the display grid. Notice the sequence of column headings for the **Package Items** list: **Download Sequence**, **File Name**, **ID**, **Device Type**, **Item Type**, **Description**, **Progress**.

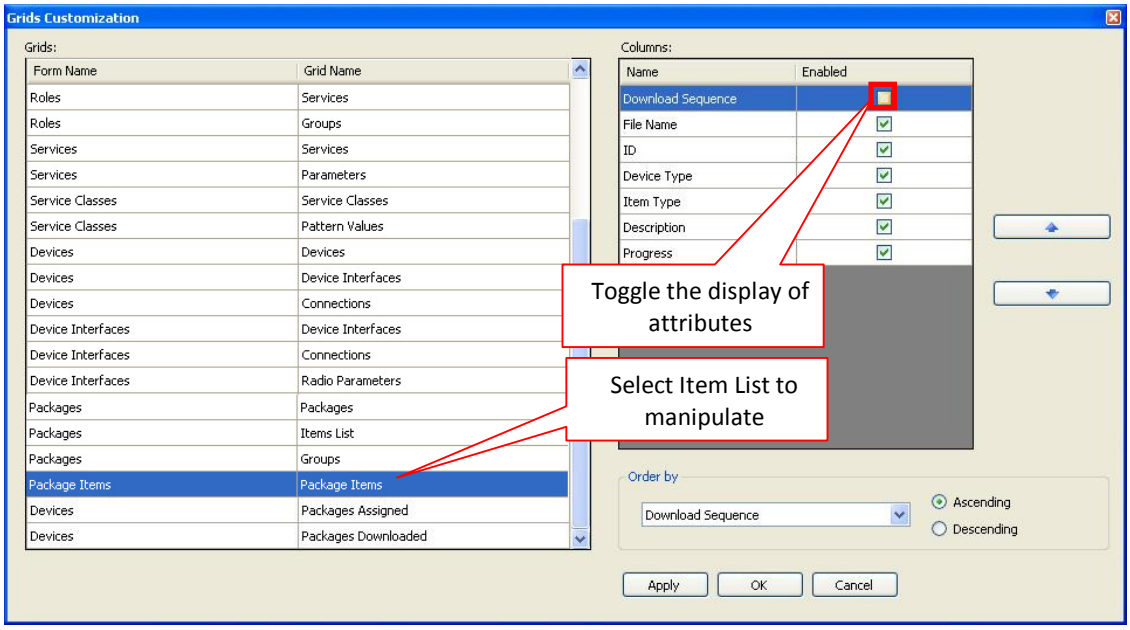


Screen Capture 24. Original "Package Items" List Grid Layout

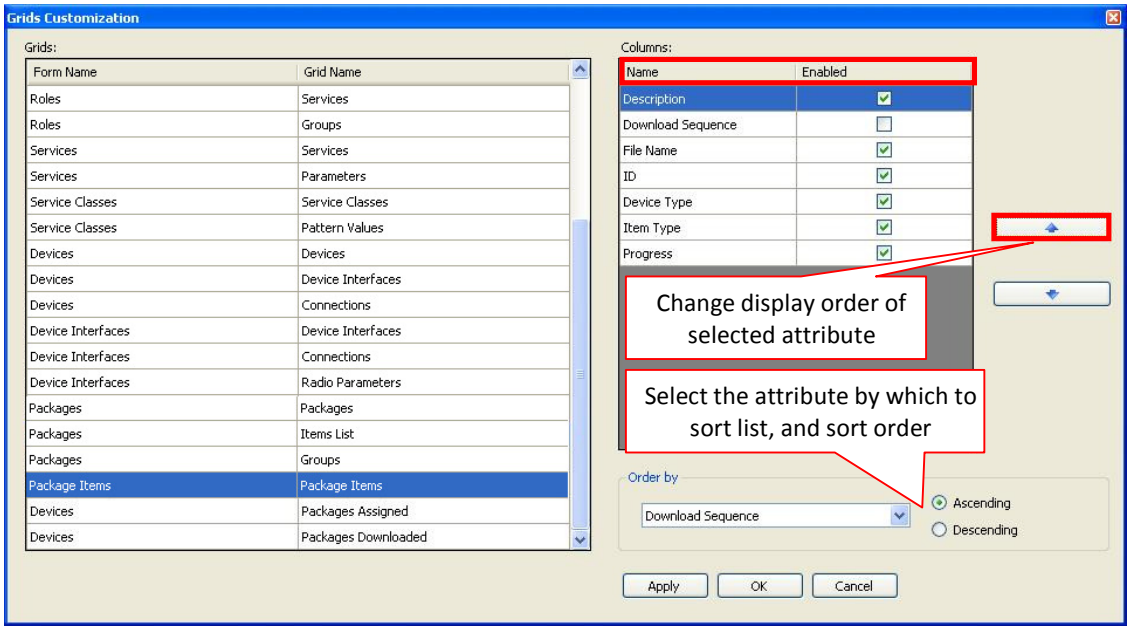
The following example shows how to suppress the display of the **Download Sequence** attribute and reorder the Description attribute to display as the first (leftmost) item. Screen Capture 25 shows the dialog box used to suppress the display of the **Download Sequence** attribute for the **Package Items** list.

Screen Capture 26 shows how to reorder the attribute display columns in the list. In this example the **Description** attribute has been moved from the sixth display column to the first by selecting the item and clicking the up button. The list sort characteristics (sort list by column, ascending or descending order) could also be changed as indicated.

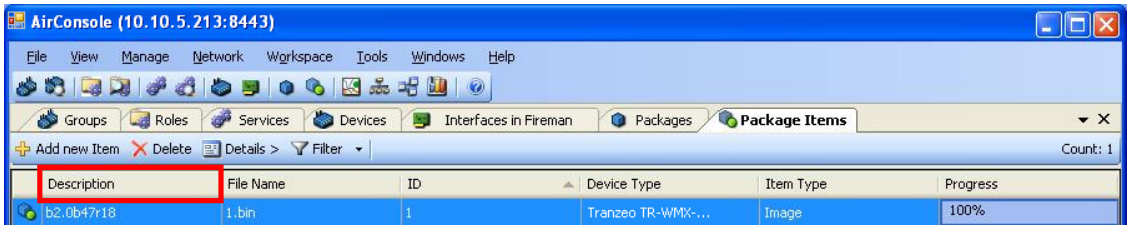
Screen Capture 27 shows the final result. The **Download Sequence** attribute is no longer displayed and **Description** is the left most column in the **Package Items** list display. The display width of each column can be directly manipulated by dragging the divider between any two-column headings.



Screen Capture 25. Toggling off display of the "Download Sequence" attribute for "Package Items"



Screen Capture 26. Moving "Description" item to the left on "Package Items" Display

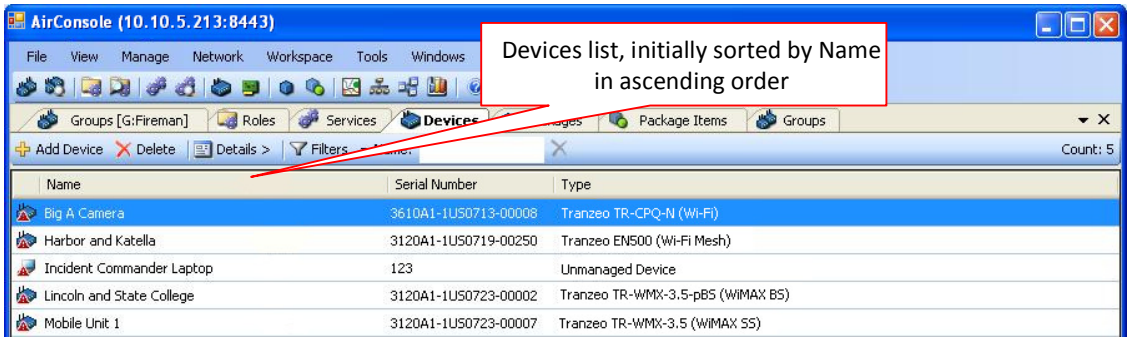


Screen Capture 27. Final result, "Description" Displayed First, "Download Sequence" Suppressed

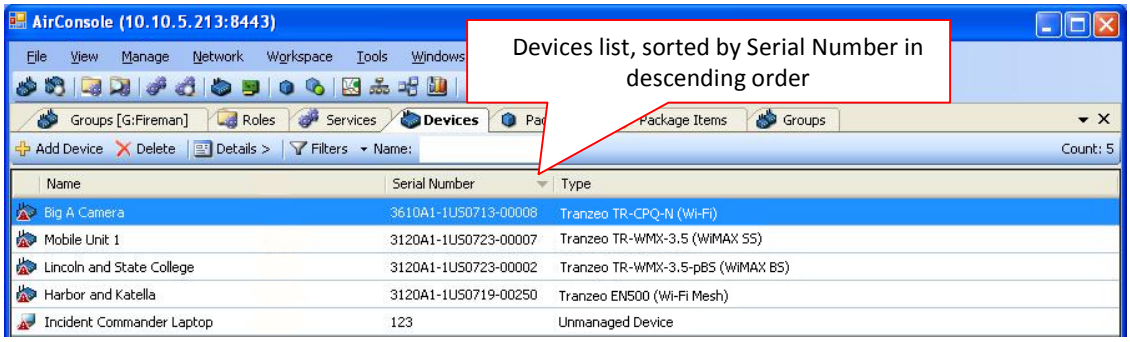
Sorting Item Lists

Screen Capture 26 shows one way to change the default sort order for an item list. On an ad-hoc or temporary basis, you can change the sort order of any item list by clicking on any column heading. As shown in Screen Capture 28, the system will display an up arrow or a down arrow next to one column heading indicating the list is currently sorted by that column in ascending or descending order, respectively.

To sort the list by a different attribute, click on the column header for the desired attribute by which to sort the list. Click on the same column header again to toggle between ascending and descending sort order. Screen Capture 29 shows the same **Devices** list, now sorted in descending order by the Serial **Number** device attribute. This simple technique makes it much easier to find specific items in long item lists.



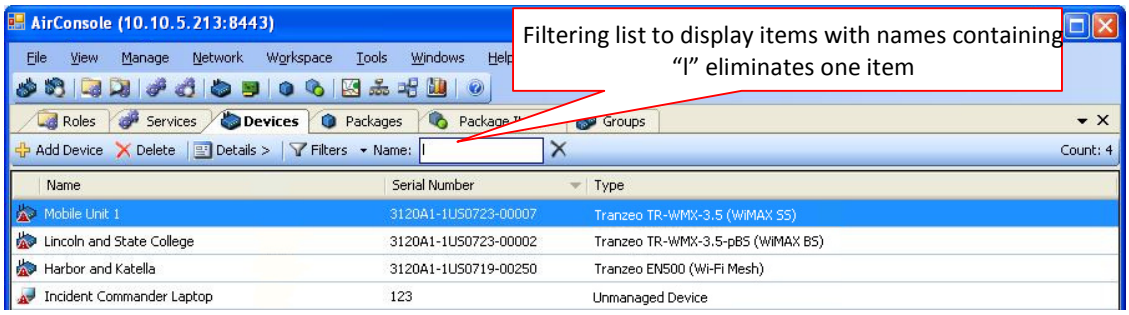
Screen Capture 28. "Devices" list initially sorted by "Name" in ascending order



Screen Capture 29. "Devices" list sorted by "Serial Number" in descending order

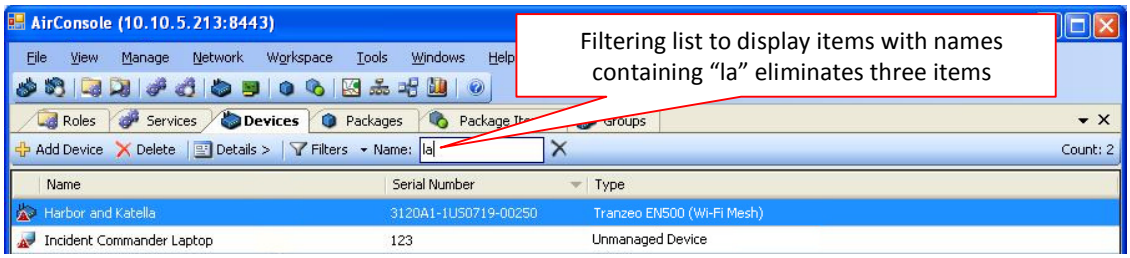
Filtering Item Lists

Various filters can be applied to item lists to narrow the list of items displayed. Filters can be used to quickly search for a set of one or more items from a large list or set of items. Screen Capture 29 shows a small, unfiltered **Devices** list containing five distinct items. Screen Capture 30 shows the list filtered to display only items with names containing "I". This filter eliminates one item from the displayed list of items. Filters provide a handy search mechanism, especially when used in conjunction with a well-designed item naming convention. Some item lists support multiple filters.



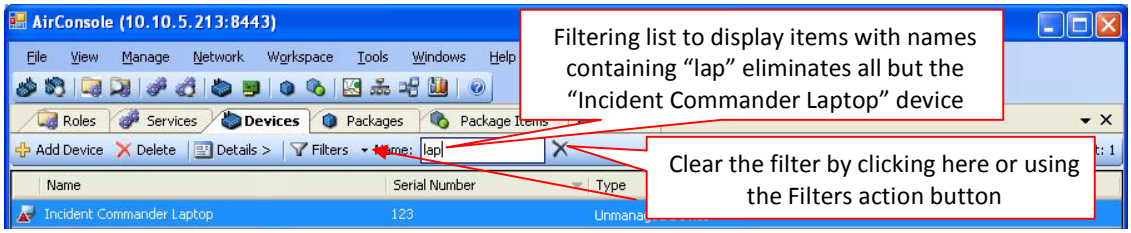
Screen Capture 30. Filtering device list to display only items with names containing "l"

Screen Capture 31 shows the list filtered to display only items with names containing "la". This filter eliminates three items from the displayed list of items



Screen Capture 31. Filtering device list to display only items with names containing "la"

Screen Capture 32 shows the list filtered to display only items with names containing "lap". This filter eliminates all but one item, the Incident Commander's Laptop, from the displayed list of items.



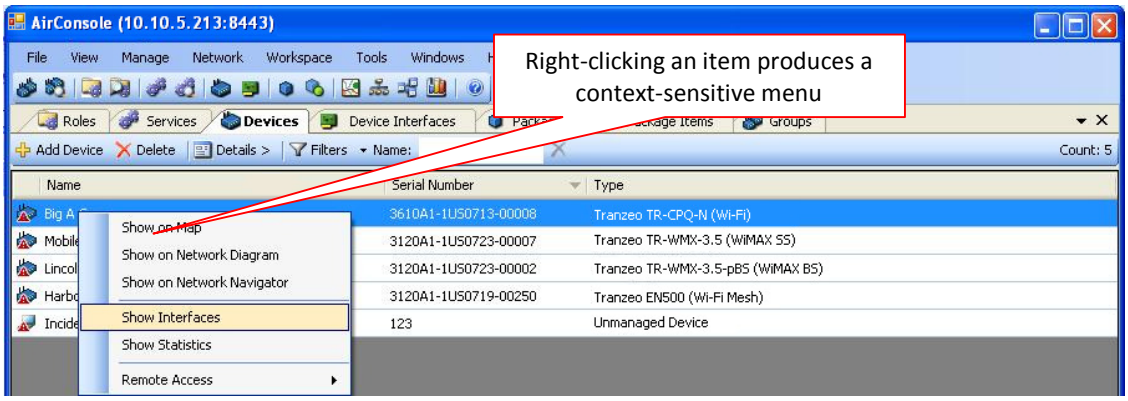
Screen Capture 32. Filtering device list to display only items with names containing "lap"

Context-sensitive Menus

Right-clicking objects in many of the GUI screens will bring up context-sensitive menus whose contents vary depending on the item selected and/or the GUI object or location from which the item was selected. Screen Capture 33 shows a context-sensitive menu that appears when right-clicking on an item in the **Devices** item list.

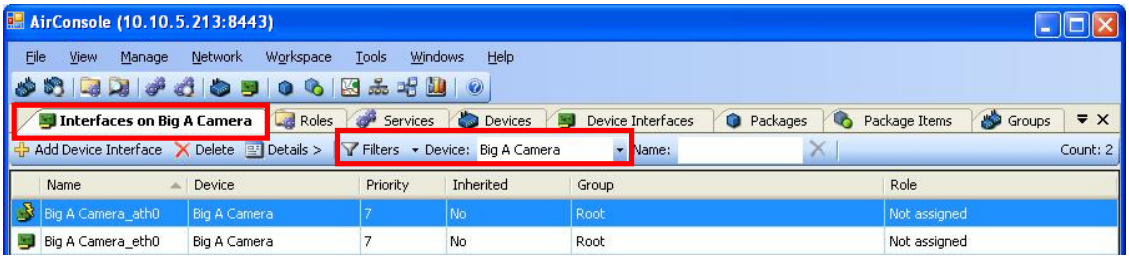
Context-appropriate actions for a selected device include:

- Displaying the device on a map or network diagram
- Listing its interfaces
- Showing statistical information for the selected device
- Launching a remote access connection to manage the device



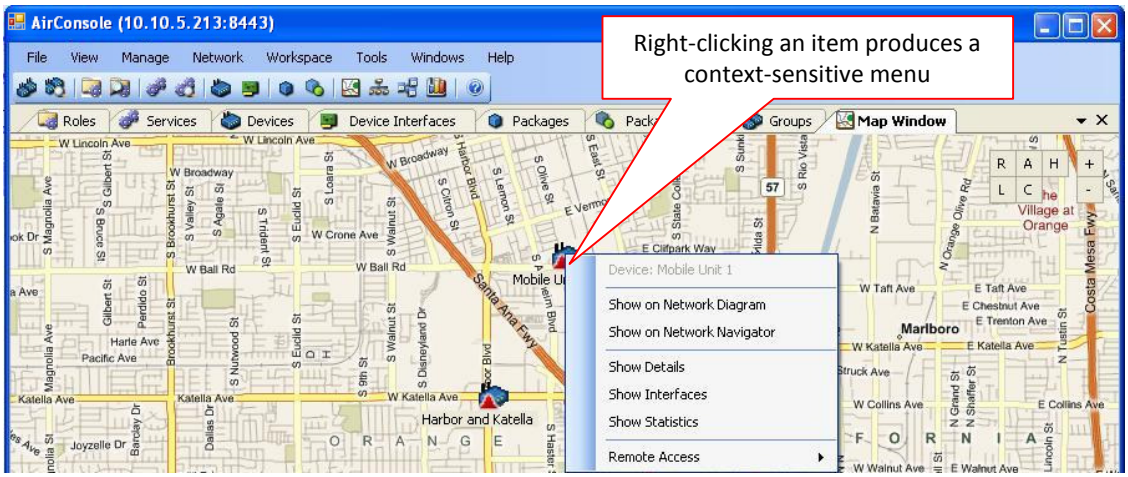
Screen Capture 33. Right clicking on selected item brings up context-sensitive menu

In many cases these context-sensitive menu actions are implemented using filters. Screen Capture 34 shows the result of selecting the **Show Interfaces** context-sensitive menu item by right-clicking on the device named "Big A Camera." The user interface provides two visual cues that a filter has been applied to the display. The tabbed item displays "Interfaces on Big A Camera" to indicate that the display uses filtering to include only Interface items specific to the device named "Big A Camera" in the item list and the **Filters** action button area indicates a device filter is in place for the value "Big A Camera."



Screen Capture 34. "Show Interfaces" context-sensitive menu selection applies a display filter

Screen Capture 35 shows a slightly different context-sensitive menu that appears when right clicking a device from the **Map Window** item.



Screen Capture 35. Different Context-sensitive menu when selecting device from "Map Window"

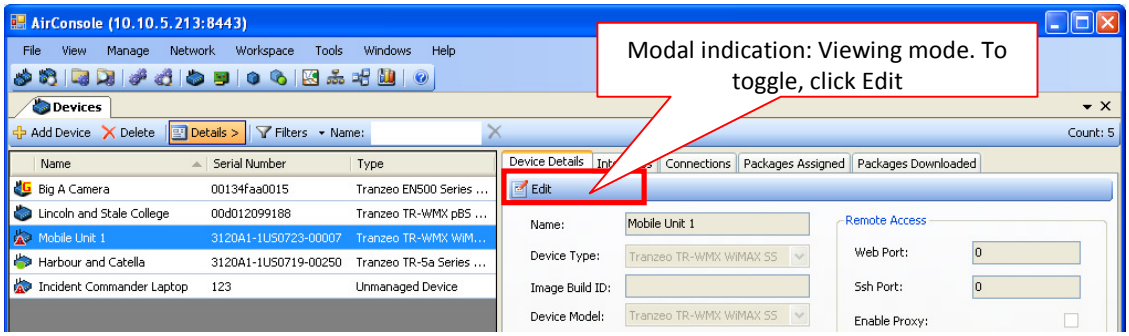
Here are a couple of additional tips and tricks:

- If you don't see an item you expect to see in a particular item list, try checking for filters and disabling them if found.
- The best way to locate an item quickly on the map is to open the **Devices** or **Devices Interfaces** item list (instead of opening the **Map Window** item) and right click on the desired item. From the context-sensitive menu, select **Show on Map** and the **Map Window** item will open with the selected item centered in the map.
- The context-sensitive menu available by right clicking on a device is a handy way to initiate telnet, SSH or HTTP management connections to a device.

Editing Item Attributes

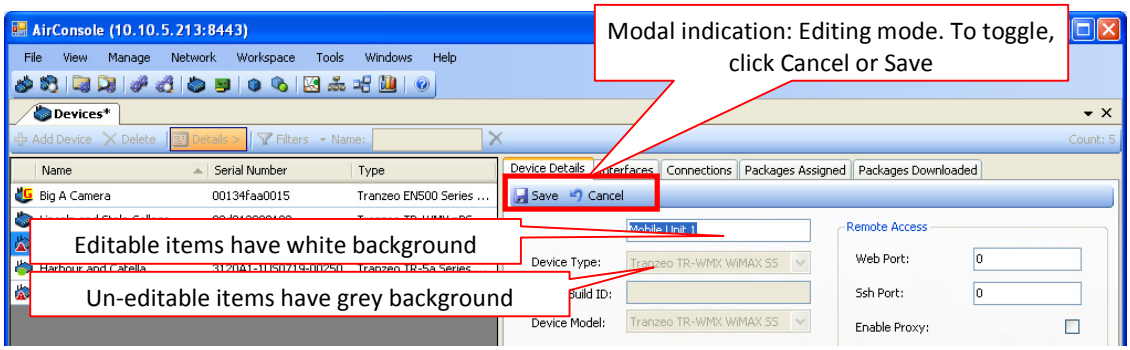
Editing item attributes is a modal operation, in part because editing mode implies a SQL "update" query transaction on the underlying database, which has a different structure than a SQL "insert," "delete," or "select" query. Therefore, before you can edit item attributes, you must leave viewing mode and enter editing mode. The appearance of the edit action button is a modal indication of viewing mode and action button itself provides the mechanism for toggling between modes. To enter editing mode, go to the item details pane and click the **Edit** action button as shown in Screen Capture 36.

Toggling between Edit and View Modes



Screen Capture 36. Use "Edit" button to exit view mode and enter edit mode

Once you enter editing mode, the **Edit** action button disappears, replaced by **Cancel** and **Save** action buttons as shown in Screen Capture 37. The appearance of the **Cancel** and **Save** buttons is a modal indication of editing mode. Click the **Cancel** button to return to viewing mode without saving changes, or click on the **Save** button to save changes before returning back to viewing mode.



Screen Capture 37. Use "Cancel" or "Save" buttons to exit edit mode and return to view mode

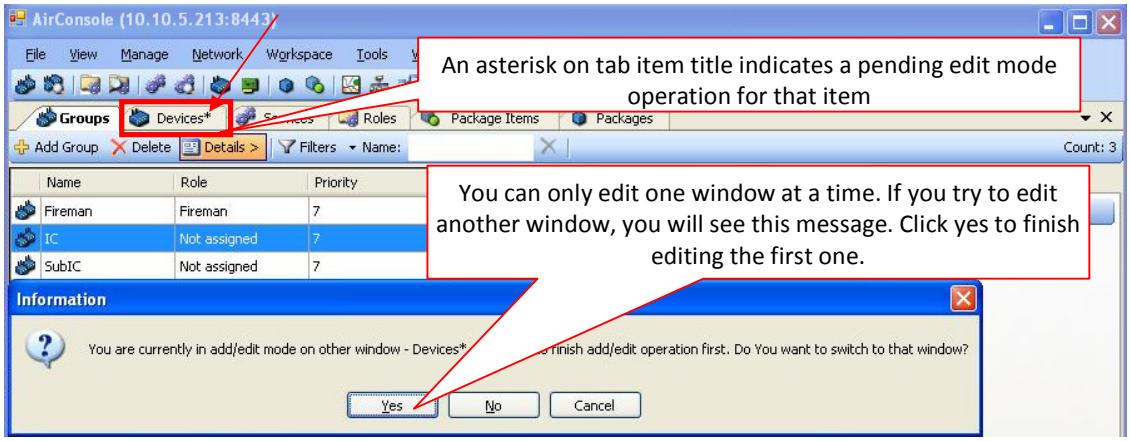
Certain Attributes May Still be Read-Only Even in Edit Mode

Even in editing mode, some attributes may not be editable. Item attributes that are editable appear with a white background. Attributes that appear with a gray background are not editable, often because these attributes serve as primary keys into internal database tables, and editing them could lead to database consistency issues.

In some cases, it may be possible to edit these items from another GUI object. In other cases, the best way to change an item's attribute value(s) may be to delete the entire item and then add it again with the correct attribute value(s). Examples of this include trying to change the MAC address attribute value on a device interface or the values for attributes such as **Image Build ID**, **Serial Number**, or **UUID** on devices. The meaning of these concepts will be explained in greater detail later.

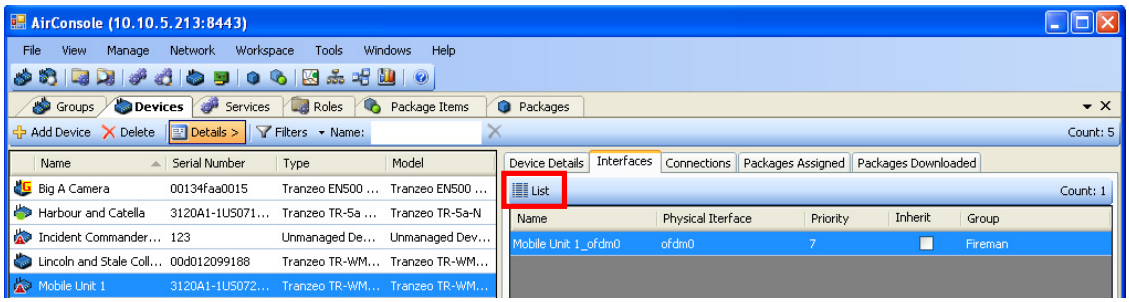
Edit Mode Updates are Transaction-based

You can only edit one set of attributes at a time. If you are in the middle of an edit operation and navigate to another item, you will see the message shown in Screen Capture 38. This message appears, if you try to start another editing operation before finishing the currently pending operation. If you click on the **Yes** button, the system will automatically navigate you to the pending operation where you can click on either the **Cancel** or **Save** action buttons as appropriate to end the pending operation before reattempting the second operation. An asterisk in a tabbed item's title indicates a pending edit mode operation (database transaction) for that item.



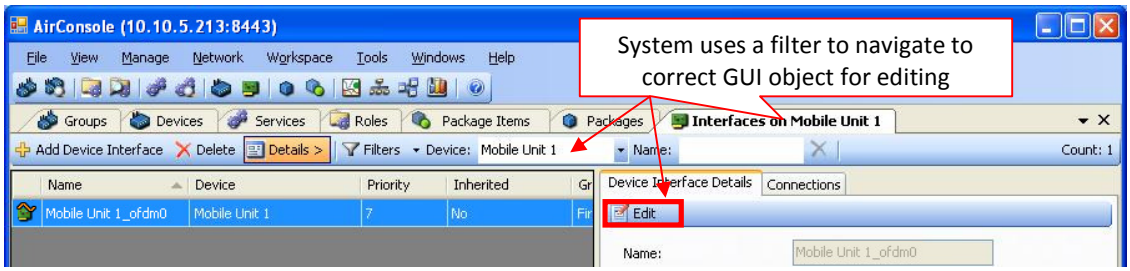
Screen Capture 38. You can only edit one window at a time

As shown in Screen Capture 39, you may at times notice a **List** action button instead of an **Edit** action button. This is generally an indication that you are in the wrong place to edit the displayed attribute values.



Screen Capture 39. The "List" action button navigates to a new location where changes can be made

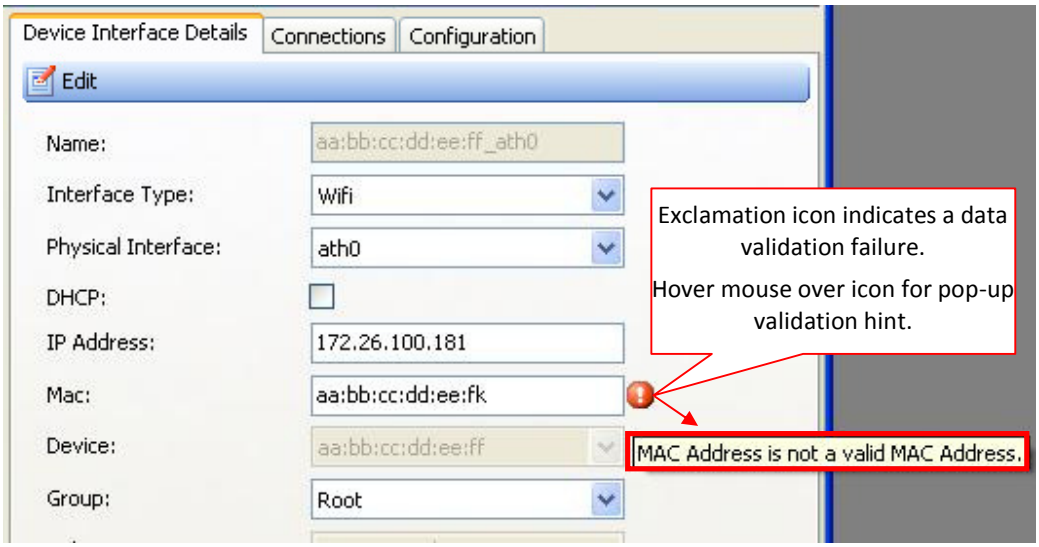
However, upon clicking on the list icon (see Screen Capture 39), the system will generally navigate a newly opened item the correct location for changing the attribute values as shown in Screen Capture 40.



Screen Capture 40. "List" button invokes filter to navigate to correct GUI object for edit operations

AirSync Data Validation

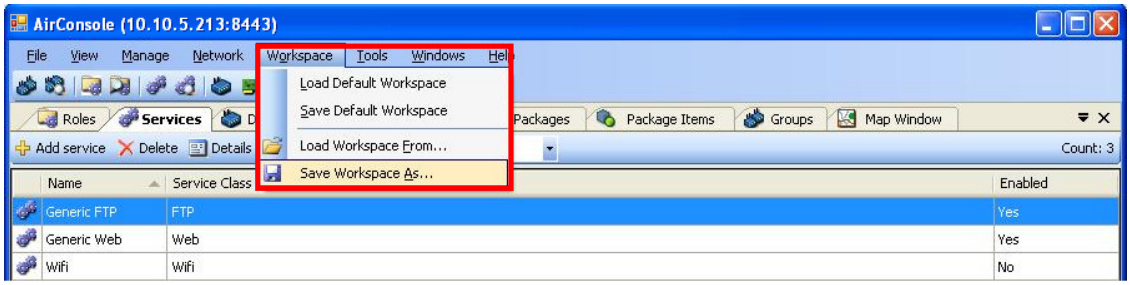
AirSync validates attribute values as they are edited. The system will display a red exclamation icon indicating a data validation failure. You will get a pop-up hint about why the validation failed if you hover the mouse over the exclamation icon. In most cases, the system will force you to adjust the value before you can save it successfully.



Screen Capture 41. Graphical indication of data validation failure during edit mode operation

Loading and Saving Workspaces

After appropriately tailoring the AirSync GUI to best suit user preferences, users can save the tailored workspace environment to a file and subsequently reload the workspace from the saved file to restore the workspace back to a desired state. Screen Capture 42 shows the workspace management menu used for saving and reloading workspace configurations.



Screen Capture 42. Load and restore workspaces from the Workspace management menu

Users may find it helpful to create and save several task-oriented workspaces, for example:

- **Create a workspace for Provisioning QoS**. This workspace might include the **Groups**, **Roles**, **Services**, **Service Classes** items as well as one or more of the explorer items (**Groups Explorer**, **Roles Explorer**, **Services Explorer**) and perhaps the **Device Interfaces** item for assigning device interfaces to appropriate QoS groups. Implementing QoS will be discussed in more detail later

- **Create a workspace for Package Management Functions**. This workspace might include the **Groups**, **Packages**, **Package Items** as well as one or more of the explorer items (**Groups Explorer**, **Roles Explorer**, **Services Explorer**) and perhaps the **Devices** item for assigning device interfaces to appropriate package distribution groups.
- **Create a workspace for other monitoring, visualization and management functions**. This workspace might include the **Devices**, Network **Diagram**, **Network Navigator**, **Map Window**, and **Statistics** items.

Initial AirSync System Setup

Before using AirSync to manage a wireless network, a few items must be tailored to appropriate site-specific values. This is largely a matter of making sure AirSync knows the correct location for its various distributed software components. Initial system setup also involves setting a few user preferences, such as governing the degree to which the system will present confirmation messages, as well as setting up some of AirSync's context-sensitive menu items to work with third-party software tools. The final steps involve registering devices in the system.

The following bullets summarize the initial setup steps, each of which is discussed in greater detail below:

- Assure the system requirements specified by AirSync (in "readme.txt" and in "Proximetry_AirSync_Quick_Install_Guide") are met as well as preinstallation requirements described in Appendix C.
- Set System Configuration parameters
- Set Options for
    - System confirmation messages,
    - The use of third-party tools such as terminal emulators for establishing remote access sessions with managed devices,
    - Internal timers,
    - Windowing options
    - Charting options
- Register Devices

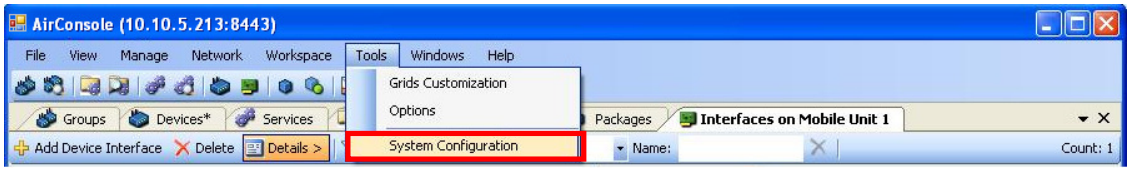
Setting System Configuration Parameters

AirSync exposes several parameters in the user interface with which an administrator can tailor an AirSync installation. Generally speaking, most of the parameters will be set appropriately during system installation, but it may be necessary to adjust a few of the values, especially if IP addresses get manipulated after the installation.

During the initial system setup, only a few of the parameters will require verification and/or adjustment:

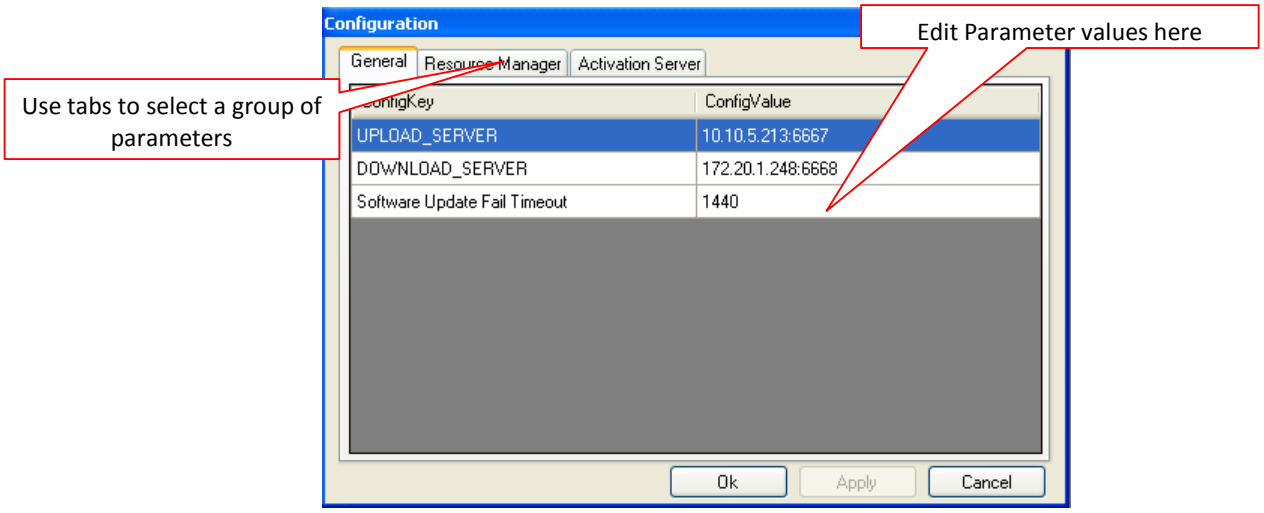
- Under the **General** tab
  - Verify/Adjust the **UPLOAD_SERVER** parameter value. It should contain the IP Address of the AirSync Server followed by a colon ":" followed by the port corresponding to the NFTP upload server which is generally port 6667.
  - Verify/Adjust the **DOWNLOAD_SERVER** parameter value. It should contain the IP Address of the AirSync Server followed by a colon ":" followed by the port corresponding to the NFTP download server which is generally port 6668.
  - Verify/Adjust the **Software Update Fail Timeout** parameter value. It should contain time (in seconds) that is designated as the maximum allowed time span for a requested software update to complete
- Under the **Activation Server** tab
  - Verify/Adjust the **Host** parameter value. It should contain the IP Address of the AirSync Server. This value will be transmitted to clients by the Activation Server
  - Verify/Adjust the **Port** parameter value. It should contain the TCP port of the AirSync Server, which is usually port 8080. This value will be transmitted to clients by the Activation Server
  - Verify/Adjust the **RMServer** parameter value. It should contain the IP Address of the AirSync Server.

The system configuration items can be accessed by clicking the **Tools** menu item as shown in Screen Capture 43.



Screen Capture 43. Accessing System Configuration items from the "Tools" menu

Screen Capture 44 shows the **General** tab of the resulting system configuration dialog box. Use the tabs to switch between groups of related parameters. The middle part of the dialog box will display the parameter names (ConfigKey) and their values (ConfigValue). Use the set of controls located immediately below the tabs to select a specific parameter for modification in the area at the bottom of the dialog box. The **Configuration** dialog box is modal. Users can't navigate to any other AirSync GUI object before closing it.



Screen Capture 44. The system configuration dialog box

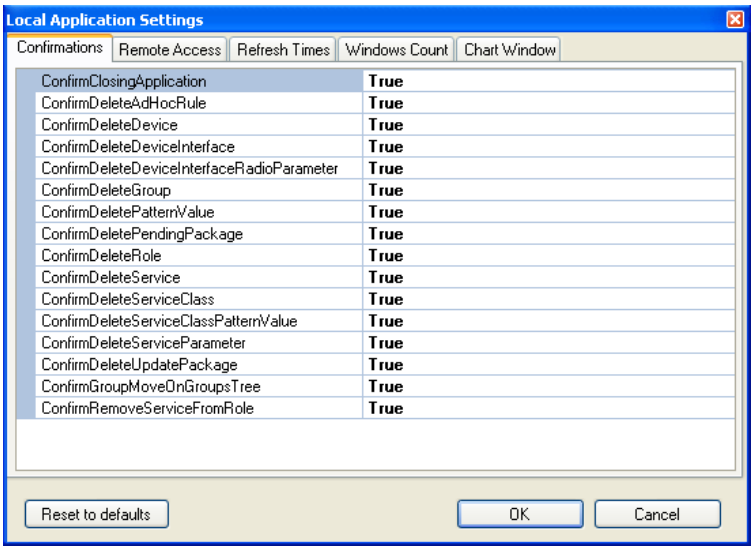
Setting Options

AirSync allows administrators to set a wide variety of options to tailor the product to a user's preferences. In general, AirSync should work fine without setting these options, but setting them can improve the user experience. For example, users can adjust these options to control the way rolling averages are computed for the **Statistics** item, or control which user actions will generate confirmation messages.

The user settable options can be accessed by clicking the **Tools** menu item as shown in Screen Capture 45.

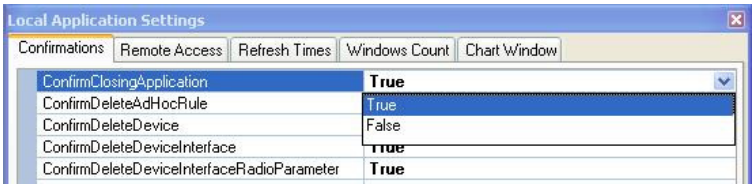
Screen Capture 45. Accessing the Options item from the Tools menu

Screen Capture 46 shows the resulting Options dialog box, opened to the **Confirmations** tab. The **Options** dialog box is modal. Users can't navigate to any other AirSync GUI object before closing it.



Screen Capture 46. Confirmations tab of Options dialog box

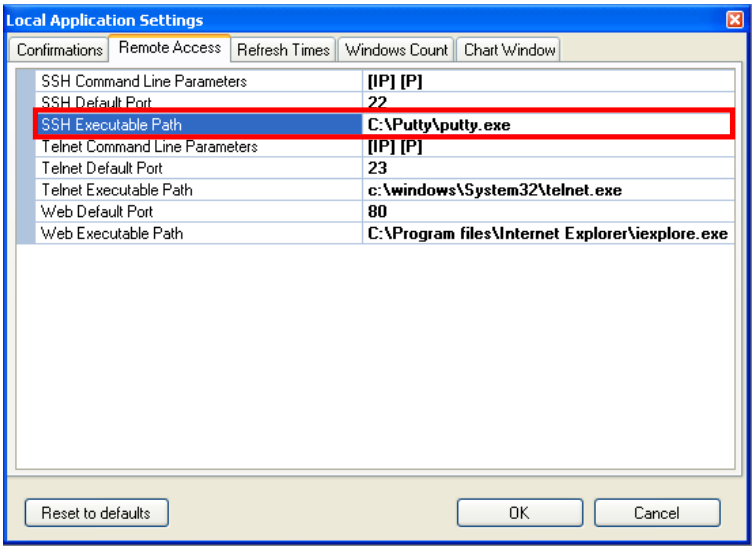
Screen Capture 47 shows the use of a drop down control to toggle a confirmation preference.



Screen Capture 47. Toggle confirmation option preferences as desired

Setting up Third-party Remote Access Tools

During initial setup, users can configure AirSync to use specific third-party software products for remote access operations. For example, to configure AirSync to use a third-party program named PuTTY for SSH access to managed devices, furnish the correct path to the executable for the **SSH Path** item as shown in Screen Capture 48. Adjust any other path items or port items appropriately.



Screen Capture 48. Configuring AirSync to use a third-party remote access tool

You may use Command Line Parameters to set some special configuration for some remote access tools. For example to use third-party program named SecureCRT set **\[IP]** parameter in **SSH Command Line Parameters field**.

Registering Devices in the AirSync System

Devices must be registered in AirSync, i.e., populated in the **Devices** list in order to benefit from AirSync's management capabilities. Devices can be registered in AirSync in an automated fashion or in an entirely manual fashion.

Automatic Device registration

To use the automatic registration facility, start the AirSync activation server and ensure the device has network connectivity. The automatic device discovery/registration process depends on the following factors:

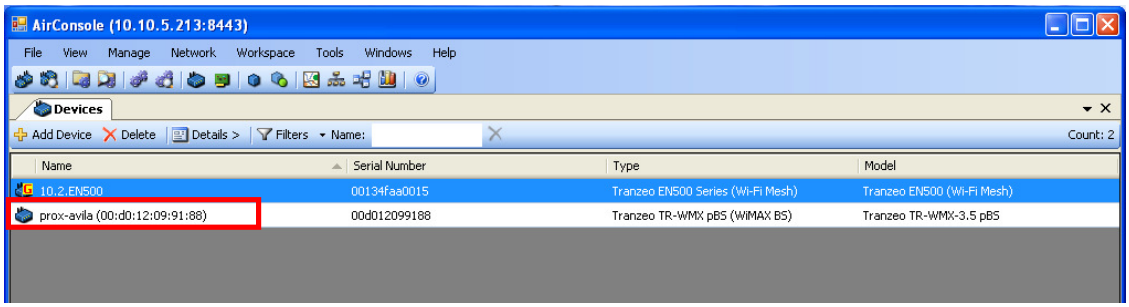
- The device must be powered on and have network access.
- The IP addressing scheme must be appropriate. This involves some platform-specific device configuration, for example, the IP addressing details and some details about the AirSync server must be configured.
- The system configuration parameters on the **Activation Server** tab must contain appropriate values.
- The activation server process must be running.
- The **Device Types** list and Device Model list must contain an appropriate entry.

Automatically Registered Devices Appear with Special Names

After a brief moment, the newly discovered/registered device(s) should appear in the **Devices** list. Newly discovered devices will be apparent in the list by observing the value of the **Name** attribute in the **Devices** list. The **Name** attribute for newly discovered devices depends on device type and may contain a MAC address value of an interface on the device as shown in Screen Capture 49.

Verify/Adjust Attribute Values for Automatically Registered Devices

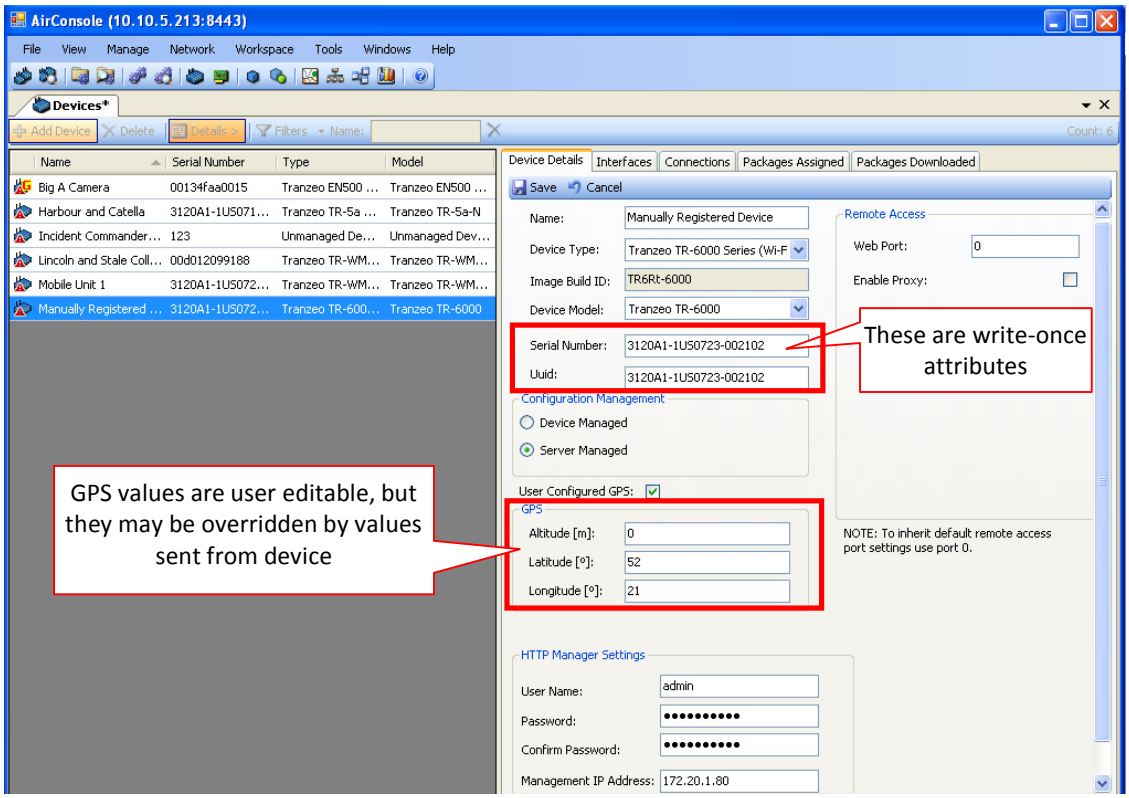
You should edit the name to a value consistent with the organizational naming convention as discussed on page 9. You should also verify/adjust/add interface details for all relevant device interfaces, especially the IP addressing details.



Screen Capture 49. The value of Name attribute may contain MAC address of new device

Manual Device Registration

To manually register a new device use the **Add Device** action button on the **Devices** list tab, also visible in Screen Capture 49. Fill in the item details for the new device as shown in Screen Capture 50.



Screen Capture 50. Manually adding/registering a new device

Device Type and Device Model

Some of device types may have different models (for example Tranzeo Pico BaseStation has three models). To enable selecting models for a device type there is a list of **Device Models** which contain only available models for selected **Device Type**.

"Write-Once" Attributes

When adding a device or a device interface, many attributes cannot be modified after the device or device interface is initially saved. **Serial Number**, and **UUID** are examples of such write-once attributes on the **Device Details** tab.

Correcting by Deleting and Adding Again

Hint: If you make a mistake entering the values for any of these attributes, correct the mistake by deleting the entire item and re-adding it with the correct attribute values.

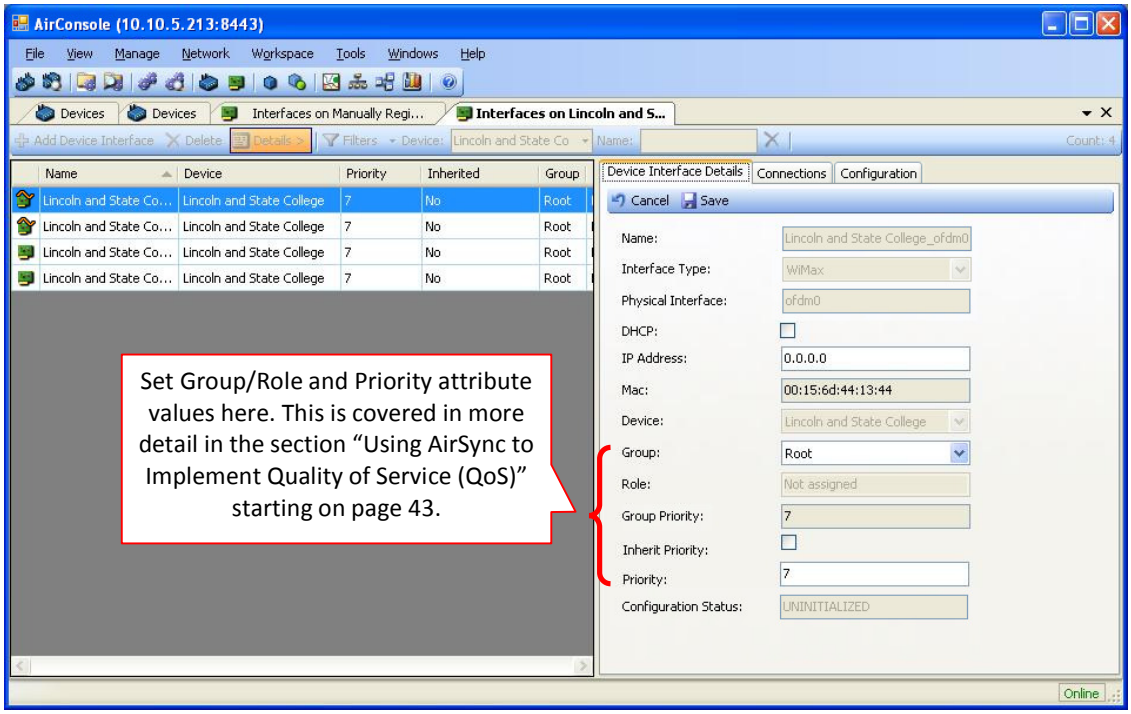
Using Multiple Tabs

Hint: consider opening a second **Devices** tab or a console session to the new device or a similar device for an easy way to cut 'n' paste values for write-once attributes. These values are important to get right and usually similar, if not identical, for devices of the same class (i.e., for devices that share a common value for their **Device Type** attribute).

After saving the **Device Details** for the new manually-added device, the system will prompt you to enter the details for the first interface of the new device as shown in Screen Capture 51.

Screen Capture 51. AirSync prompts you to enter device interface details for the newly added device

Screen Capture 52 shows the **Device Interface Details** tab after previously saving the interface information. Notice the gray background in the **Interface Type**, **Physical Interface**, and **Mac** fields indicates that these are write-once attributes. In fact, only the IP address and attributes related to QoS (notice the white backgrounds) may be subsequently modified on this tab.



Screen Capture 52. The "Device Interface Details" tab

Devices and Device Interfaces may have multiple IP Addresses

Notice that there are IP address fields on both **Device Details** tab (Screen Capture 50) and the **Device Interface Details** tab (Screen Capture 52). The address on the **Device Details** tab is a proxy address for the purpose of establishing management sessions to the device. Each device may have multiple interface-specific IP addresses, too.

GPS Values Set on Device May Override those Set on Server

It is possible to enter values for the **Altitude**, **Latitude,** and **Longitude** GPS-related attributes found on the **Device Details** tab (Screen Capture 50) of the AirSync server. However, depending upon the vendor platform, some managed devices in AirSync have client agents that automatically send updated GPS information on a periodic interval. This allows AirSync to track the device location and update its display on the **Map Window** for example, in response to the movement of mobile devices. In case you want to receive GPS attribute values from the device agent check off **User Configured GPS** field.

Using AirSync to Implement Quality of Service (QoS)

The ability to systematically define and overlay a structured traffic-shaping or QoS scheme onto a complex wireless network environment is probably the most beneficial feature of AirSync. This section discusses the details of implementing QoS with AirSync.

The following features, which will be discussed in greater detail below, summarize AirSync's traffic management features that have the ability to:

- Recognize and differentiate multiple distinct traffic flows.
- Treat multiple instances of the same recognized flows differently based on different organizational usage *roles* of end users using the traffic flows.
- Shape traffic both in the upstream and downstream directions.
- Shape traffic at multiple points in the network.
- Dynamically monitor and detect changes in network bandwidth capacities and automatically make appropriate QoS adjustments.
- Dynamically move QoS "rules" or *Service Level Agreements* (*SLAs*) around the network in response to the movement of mobile devices. As devices move around the network, the rules that govern traffic behavior for associated users move appropriately.
- Systematically arbitrate the allocation of excess bandwidth during periods of network under-subscription.
- Systematically arbitrate the allocation of bandwidth during periods of network over-subscription. This feature, called *Service Level Degradation* (*SLD*) allows network performance to degrade in a graceful, rules-based fashion when conditions make it impossible to meet all defined SLAs.
- Dynamically change QoS parameters "on-the-fly", but without requiring any user intervention, in response to various network trigger events, whenever they may occur. Since it's unknown when or if these events will occur, these systematically defined changes are called *AdHoc Rules*. They may occur, for instance, in response to changes in network topology (the number of stations associated to a device) or changes in signal quality (improvement or degradation of modulation scheme or bit rate).
- Support Wireless Multimedia Extensions (WME), a set of special queuing disciplines well-suited for managing latency-sensitive traffic flows.

Theoretical Building Blocks

Different Flows Have Different Network Characteristics

In the absence of AirSync's systematic QoS scheme, all network traffic gets identical best-effort service, competing for network bandwidth resources on an unmanaged "first-come-first-served" basis. Network performance and user satisfaction quickly degrade even before the network reaches total saturation in terms of raw bandwidth capacity.

Performance may suffer for a variety of reasons because different traffic flows have fundamentally different characteristics:

- **Some traffic, such as voice and video, is *delay-sensitive***. If voice or video packets don't make it through the network on a real-time or near-real-time basis, the perceived voice or video stream becomes garbled. Together with effective receive-side buffering techniques, the human ear and human eye can interpolate to smooth out perceived quality if some of the packets get lost or delayed in-transit. However, after a certain amount of delay, it's better to simply drop video or voice packets rather than transmitting and processing them if they would arrive so late as to merely waste bandwidth and garble the received signal stream. On a limited basis, this traffic could be classified as **somewhat *loss-tolerant***.

- Real-time or near-real-time traffic flows such as **voice and video are also *jitter-sensitive***. Jitter refers not to an absolute magnitude of delay, but rather to a variable rate of delay which can also garble perceptual quality.

- FTP file transfers, for example, and other traditional **data flows are *loss-intolerant* but somewhat *delay-tolerant* and *jitter-tolerant*.** By delay-tolerant and jitter-tolerant, it's meant and understood that users may meet frustration and eventually cancel the operation after a certain threshold of delay, but the transmitted data will still be usable, even if it experiences occasional brief delays and some jitter. Loss-intolerant is described as unlike a small drop or loss rate for voice or video packets that can be tolerated, every packet must be successfully received or the data file will not be usable. Of course, normal TCP and lower-level networking mechanisms will handle error detection, correction and retransmission when necessary, but every packet must be correctly received.

- Some traffic is more *bursty* in nature while other flows have a more *constant bit rate* (CBR) quality.

So even if there's still adequate link capacity, an FTP flow could disrupt a video flow in an unmanaged network, for example by creating excess jitter. Fortunately, AirSync supports multiple queuing disciplines, including special low-latency queues for voice and video flows.

Understanding the AirSync QoS Processes

The following bullets summarize the QoS implementation processes in AirSync:

- **System Administrators define/model organizational network usage policy**. The policy is articulated both in terms of traffic type and user type. To achieve this step, administrators manipulate various AirSync objects including: *patterns*, *service classes*, *services*, *roles*, and *groups*. These concepts will be defined in more detail later.

- As a result, the **AirSync server stores a set of business rules or template trees** that best reflect organizational policy. These templates will be used later to generate instructions that agents on managed devices will eventually use to build the actual packet filters and provisioned-queue structures that ultimately implement the organizational policy. AirSync supports a variety of rules including rules for controlling basic upstream and downstream traffic rates, how to allocate excess bandwidth when it is available, and how to resolve/arbitrate conflict if the network becomes oversubscribed.

- AirSync server components communicate with agents on the managed devices**.** The **AirSync Server continuously monitors the network for performance statistics and significant network events**. Some example events include a managed Wi-Fi station breaking or forming an association with a managed Wi-Fi access point, the usage role of a managed device changing, or an improvement or degradation of signal quality and/or bandwidth capacity on a managed device.

- The **network events trigger the AirSync server to determine and generate the best set of QoS instructions** for the managed devices, based on user roles and current network conditions, by consulting the rules, templates and network topology conditions stored its internal database.

- **The AirSync server sends an appropriate set of QoS instructions to the agents on the managed devices**.

- The **agents construct and activate a set of packet filters and queues** on the managed devices that implement the situationally-appropriate usage policy.

- The **agents report performance statistics and events back to the AirSync server**.

- **The Server monitors information from the agents, reacts to it and provides feedback to system administrators**.

- **System administrators periodically monitor performance and adjust policy**.

Conceptually, there are two distinct cyclical processes involved:

- The human process used by system administrators to define network policy and monitor performance:

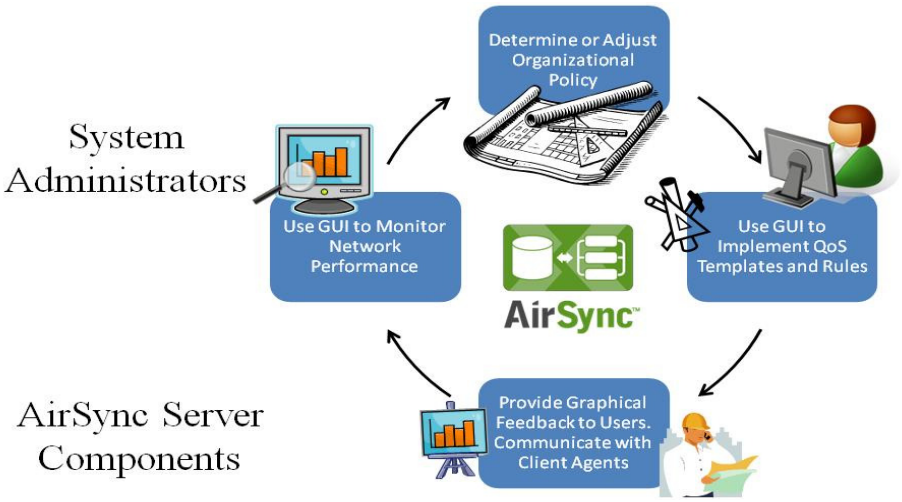


Figure 1. The AirSync QoS System Administration Process

- The near real-time machine process of interaction between the AirSync server components and AirSync agents running on managed client devices that monitor events and implement QOS:

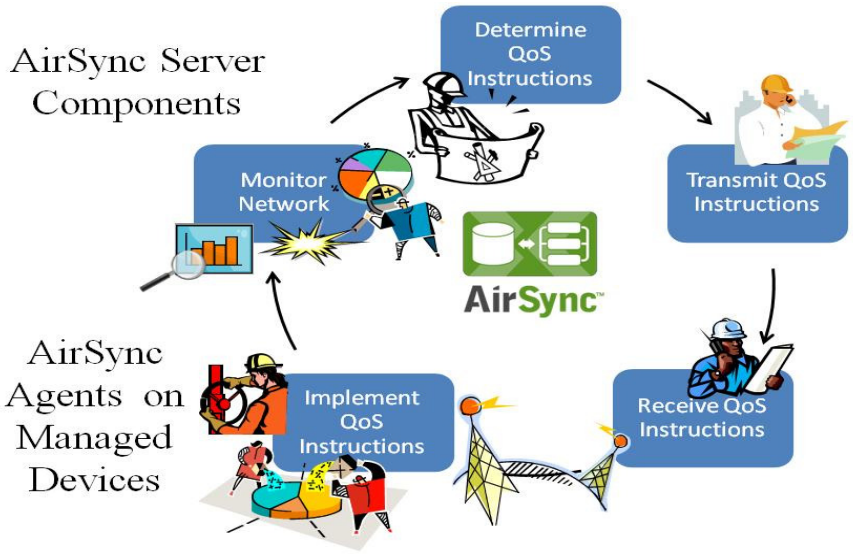
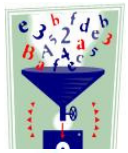


Figure 2. The AirSync Server/Agent QoS Implementation Process

Understanding How the Pieces and the Processes Fit Together

Step 1: Define QoS Goals, Organizational Policy



Implementing QoS starts with understanding what application traffic flows will traverse the managed network:

- What kinds of data will transit the network?
- What are the protocols?
- What are the bandwidth characteristics of each stream, for example, true-color full-screen video at thirty frames-per-second compared to a single VoIP phone call using G.729 vocoding/compression and silence suppression? Is the traffic mostly upstream, mostly downstream, or balanced?

- When are given flows more likely to occur? Will there be surges of email activity at the beginning of the day, at lunch, and at the end of the day? What about batch/bulk file-transfer operations during off-peak hours?
- Is the traffic client-server oriented or more peer-to-peer in nature?
- Between which network end-points will a given flow stream?
- Where are the servers? Where are the clients?

In addition to understanding the application traffic to be used over the network, the implementation of QoS involves understanding differences in users' roles and circumstances:

- Are some jobs more important than others under different circumstances? Are there times when one job function or role should get priority above or below others? Consider a municipal network. During normal operations the network may carry a significantly different traffic flow than during special events (such as city parade, large convention) or during public safety emergencies (such as natural disaster, act of terrorism).

- What jobs do different users perform over the network? During normal operations and even during special operations it may be possible to define different user roles for the network. For example, during a public safety emergency, Firefighters could use the network to download building blueprints or check weather forecasts. Police might use the network to monitor or transmit special camera feeds to or from city hall or remotely control traffic devices. Medical personnel may need to access healthcare information or perhaps send or receive real-time bio-metric information from special devices worn by injured parties. If the network gets congested, how should the traffic get prioritized?

Once traffic characteristics and user needs have been contemplated, you have enough information to define basic service level agreements (SLA) that articulate the organizational usage policy for the users of your managed network.

For example, during normal operations city maintenance engineers should get between 100-300 KBps for web traffic to the internal web server at city hall. General web service to all external web sites will be allocated from 50-150 KBps. The system will allocate between 200-400 KBps for connecting to a special city street maintenance database application. Other users could get vastly different bandwidth allocation profiles based on their differing job roles within the organization.

Step 2: Define *Service Classes* to Recognize Distinct Traffic Flows

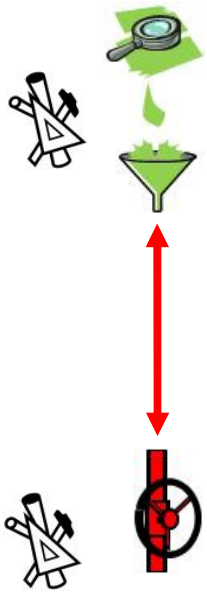
After gaining an understanding of the traffic patterns, priorities, and intended user network usage roles, the information can be modeled as organizational policy in AirSync causing AirSync to generate appropriate QoS instructions and distribute them to the correct managed devices. **But before distinct traffic can be managed with differentiated service level agreements, there must be a way to recognize the various distinct traffic streams** so that AirSync can provide differentiated service to them.

Service Classes give AirSync a mechanism to recognize and sort (classify) packets from different traffic flows as shown in Figure 3. Think of a service class as a template for generating packet filters. The filters will eventually be built and used on managed devices to sort packets into specific "buckets" or "funnels" that buffer and guide packets to specific queues. The queues will empty at independently-provisioned rates to provide differentiated service to different types of traffic.

Figure 3. *Service classes* are templates for building packet filters to sort network traffic flows

Note that **a service class doesn't define how the packets in the class will get treated** (i.e., how much bandwidth should be provisioned, etc). It merely defines a template for recognizing a given traffic flow. **Each service class contains one or more *patterns* that will classify (include) matching packets** based on items such as source IP or network address, transport protocol (TCP/UDP) and port number. In the case of generic web traffic, there would most likely be two patterns, one for HTTP (on TCP port 80), and one for HTTPS (on port 443). The actual QoS parameters governing bandwidth allocation are provisioned elsewhere.

Step 3: Define *Services* that Provision QoS Parameters



Don't confuse the names! Recall that *service classes*, discussed above, are templates for constructing filters that sort flows of packets into specific buckets or funnels called traffic classes. On the other hand, ***services* define templates for provisioning QoS parameters for a traffic class**. AirSync uses *services* to construct the basic SLAs that implement organizational usage policy.

Each *service* references a *service class* and associates it with a specific SLA that establishes how to treat the traffic type for a class of user. **The *service* template defines rate and priority characteristics** that will be factored into the instructions that client agents will eventually use to build provisioned queues for a given class of user on a given device interface. So when you think about services, you can visualize valves on queues for specific traffic flows provisioned to flow at a specific rate for specific users. This constitutes a basic form of an SLA.

Figure 4. *Services* are templates for provisioning basic Service Level Agreements

So why make a distinction between a service class and a service? Reusability is one reason. The same packet classifier (service class) can be reused in multiple services, each provisioned to appropriately reflect different user needs for the same type of traffic.

For example, you could define a single service class called "Generic Web – Any Source" that classifies all HTTP and HTTPS packets to/from any web server. Then you could define distinct services such as "Web – Gold Users," "Web – Silver Users," or "Web – Bronze Users" that could provision different SLAs for three different classes of web users. All three services could be based on the same service class because the web traffic is *recognized* the same way for all three types of users, but defining three distinct services allows the traffic to be *treated* three different ways depending upon the role of the traffic user.

Step 4: Assign *Services* to *Roles*

A *service* defines the way a specific class of traffic gets treated from a QoS perspective for a given class of user. But most users generate and consume more than one type of traffic. A user's traffic usage pattern varies according to the user's tasking or *role* within the organization.

As examples, a call center operator would generate VoIP traffic, probably some email and a fair degree of web traffic from a specific web site, for instance to fulfill orders from a catalog. A doctor might generate or consume a large volume of high-resolution x-ray images or high-definition video. A user in a coffee shop might consume low-resolution streaming videos, read some email and visit a broad spectrum of web sites. As shown in Figure 5, different organizations may have vastly different sets of user roles.

Figure 5. User roles vary between and within organizations

In AirSync, *roles* provide a template to specify a set of provisioned services that reflect the traffic usage policy for a given class of user (or device) within an organization. For example, a fireman may generate FTP, VoIP and web traffic on the job. Figure 6 depicts an AirSync role defining a template for provisioning three services according to the organizational network usage policy for the "Fireman" class of users.

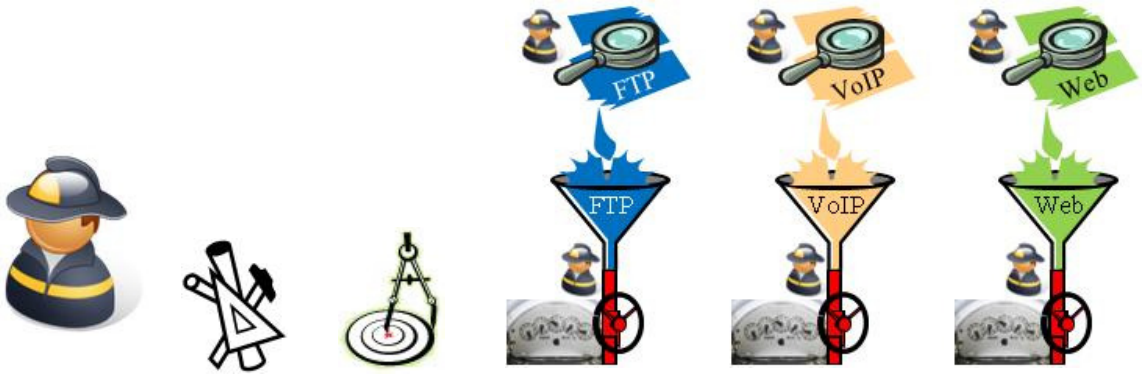


Figure 6. The Fireman role template (on the AirSync Server) defines provisioning for three services

The set of all defined service classes, services and roles creates a "template tree" indexed by user type or role. As described in the process depicted by Figure 2. The AirSync Server/Agent QoS Implementation Process, when AirSync detects users associated with managed devices, a server component consults this template tree to generate QoS instructions appropriate for the user role(s) assigned to the associated devices. The AirSync Server sends the instructions to AirSync agents on the managed devices for implementing the actual packet filters and queues to control traffic as depicted in Figure 7.

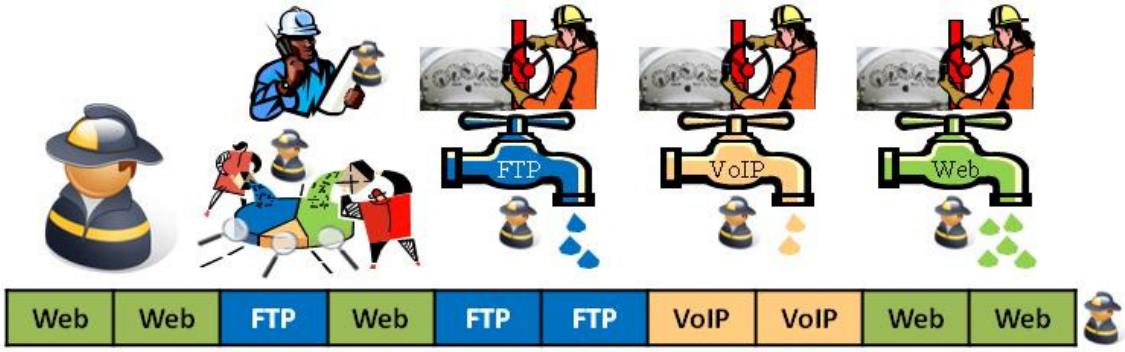


Figure 7. AirSync agents implement packet filters and output queues on managed devices

In the following series of template tree diagrams, notice the pointer relationship between the various objects.

Figure 8 shows a more detailed look at the "Fireman" role showing how all the related parts fit together. Looking from the bottom up on the template tree structure, notice that the *role* references *services* which reference *service classes* which reference *patterns*.

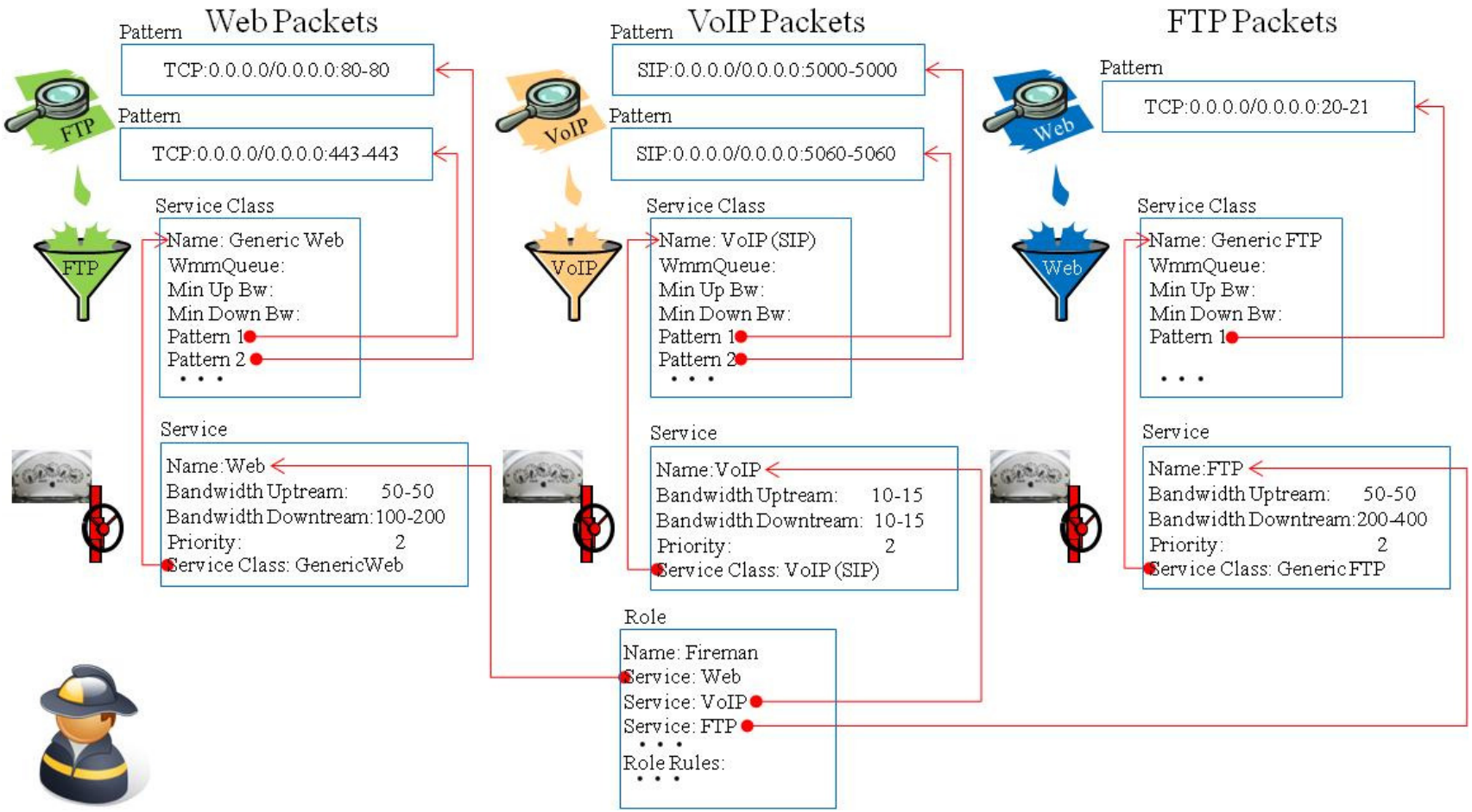


Figure 8. A detailed look at the Fireman Role

An organization could define distinct "Fireman" and "Policeman" roles which may (or may not) rely on common service classes, for example "Generic Web Traffic." The organization could define one set of provisioned services, say, Web, VoIP, and FTP for the "Fireman" role and another set, such as Web, Video, and Email for the "Policeman" role. If the organizational policy dictates that the "Fireman" and "Policeman" user classes should get identical bandwidth allocations (think *service*) for generic web traffic, which is the one type of traffic (think *service class*) both roles have in common, a common "Web – Public Safety (50K-100K)" *service* could be used in both roles as depicted by the template tree shown in Figure 9.

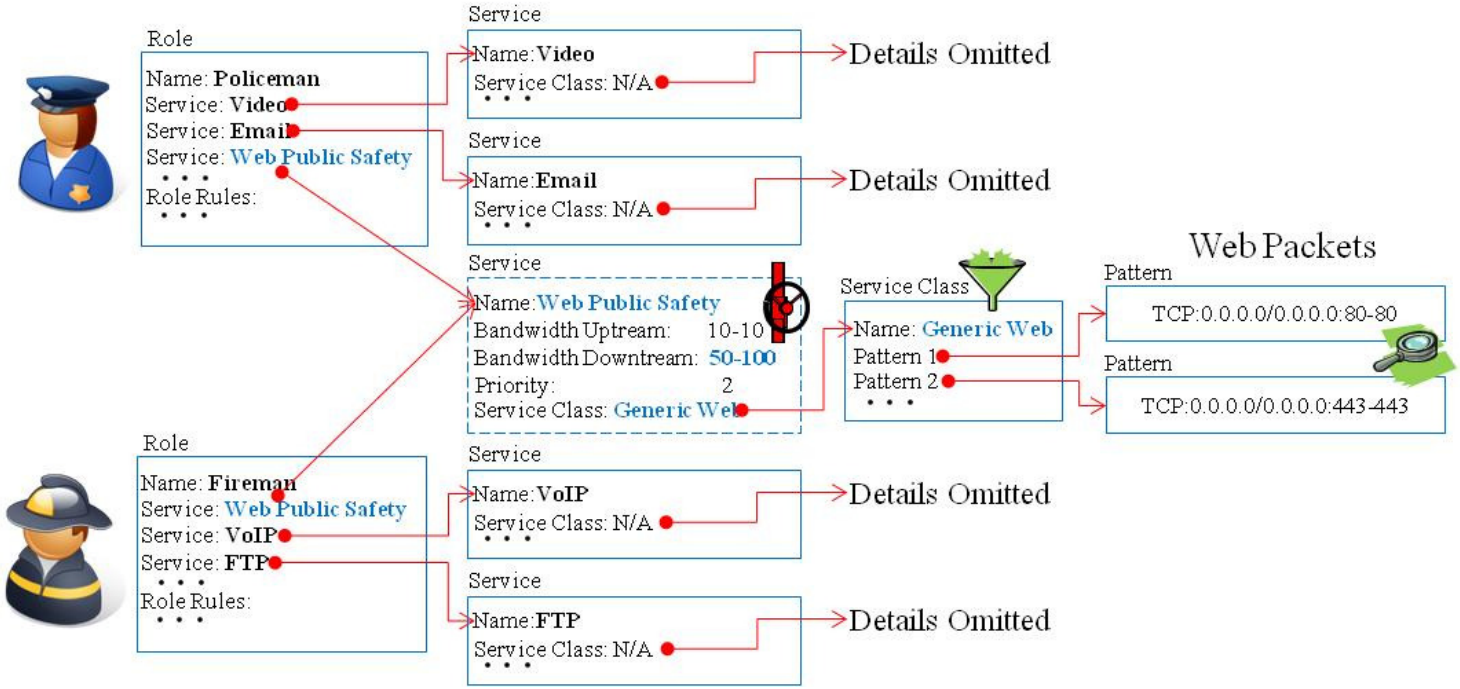


Figure 9. A template tree where distinct roles share an identically provisioned web service

If the organization wanted to provision different web traffic SLAs for firemen and policemen, it could define two distinct services, say "Web – Fire Dept (100K-200K)" and "Web – Police Dept (250K-350K)" and refer to the different web services in the respective user roles as depicted by the template tree shown in Figure 10. In either case, as long as the rule for recognizing web traffic was the same, all the web services could reference the same service class, "Generic Web Traffic – Any Source."

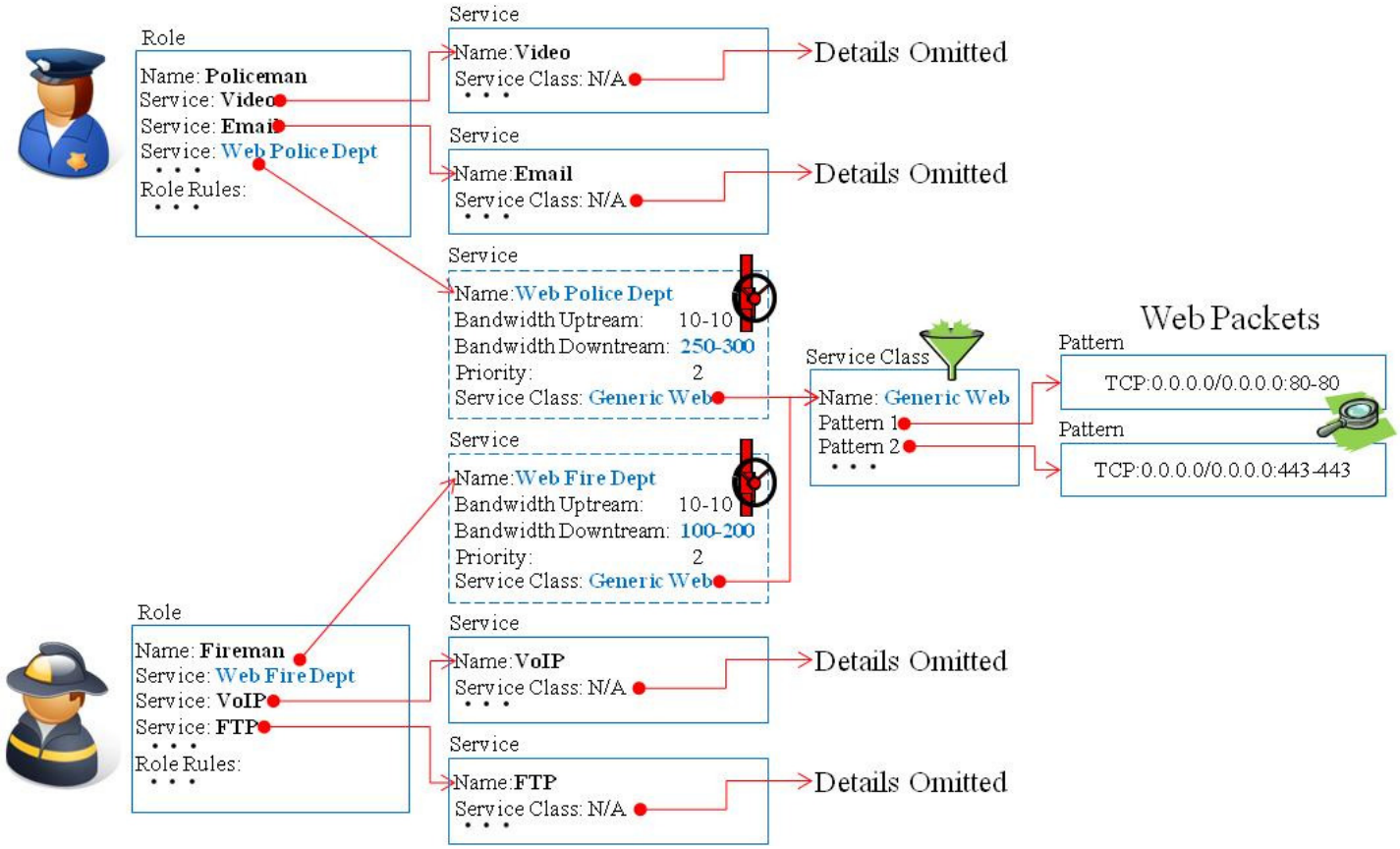


Figure 10. A template tree where distinct roles each have individually provisioned services for identical web traffic flows

If one of the user classes, say "Policeman" needed specially-provisioned access to a specific departmental server, however, it may be more appropriate to use the two distinct web services and also have them reference distinct service classes, too. In this case the "Web – Fire Dept (100K-200K)" service might still reference the "Generic Web Traffic – Any Source" service class, but the "Web – Police Dept (250K-350K)" service might reference a new "Web – Police Server" service class that matches only web traffic to the departmental server as depicted by the template tree shown in Figure 11.

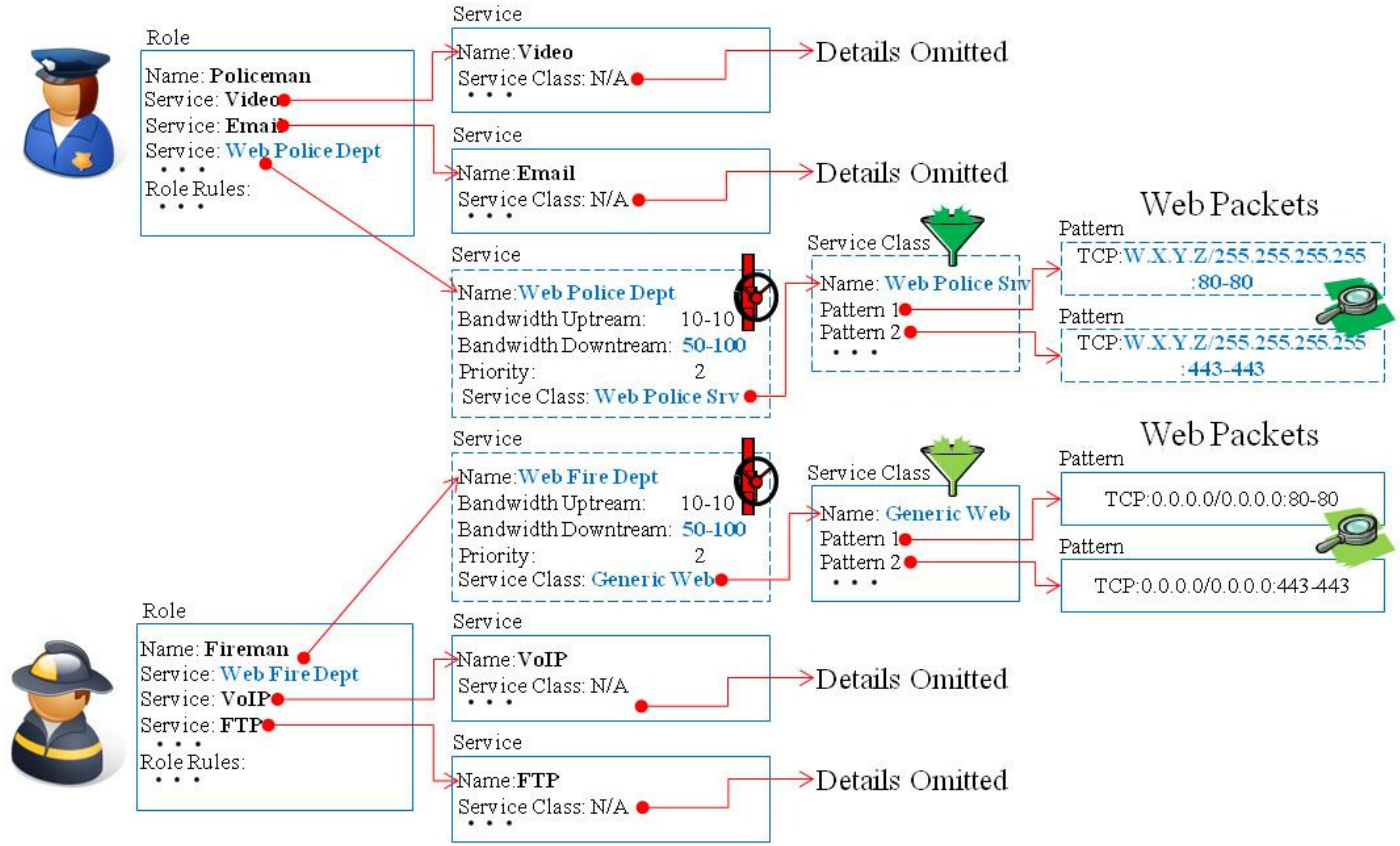


Figure 11. A template tree where distinct roles each have individually provisioned services for different web traffic flows

You could even define two different web services in the "Policeman" role if you wanted to provision, for the policeman user class, internal web traffic to/from the departmental web server independently from other web traffic to/from all other sources as depicted by the template tree shown in Figure 12. There are many possible combinations. The key point is that **AirSync uses service classes, services, and roles to give you reusability when you want it, but flexibility when you need it**.

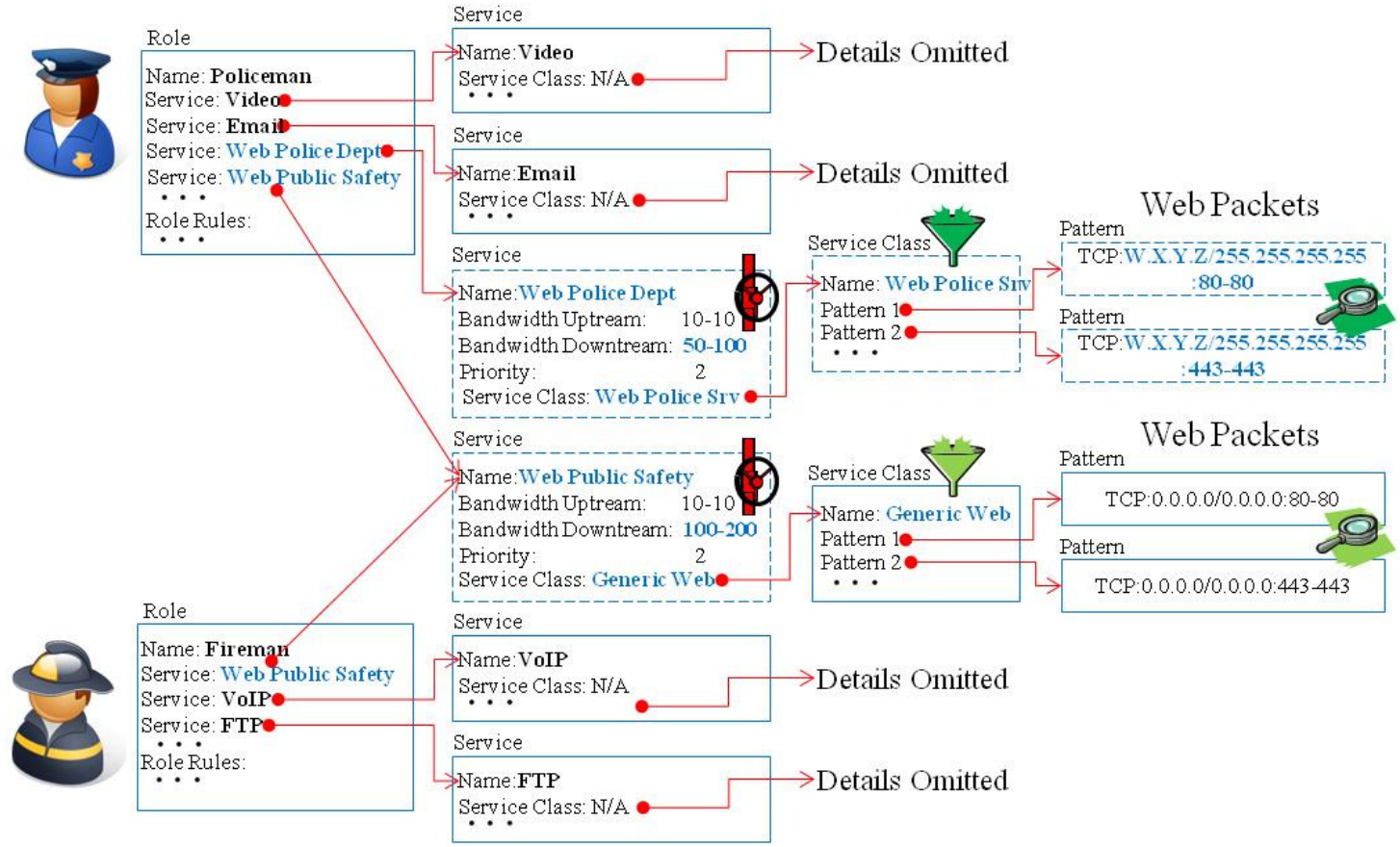


Figure 12. A template tree where one role has multiple individually provisioned services for different web traffic flow

What's in a Naming Convention?

Hint: Notice that the naming convention for services in the examples above, for example, "Web – Policeman (250K-350K)," included traffic type (Web), user class (Policeman) and provisioning information (250K-350K). This can be good and bad. It makes it easy to understand what the services are for, but if you often adjust provisioned bandwidth parameters, you create a maintenance burden of adjusting the service name. If you fail to keep up with the maintenance, the names become misleading.

Devices Personified

Hint: It is perfectly appropriate to **define one or more roles, services and/or service classes for devices as well as users**. For example, a municipal network may include several traffic camera devices that generate traffic. In this case, each camera device functions as a network user (generates traffic).

Step 5: Define *Groups* for Associating *Device Interfaces* with a *Role*

Within AirSync, groups are containers with which you can relate other AirSync objects, such as device interfaces and roles. Groups function as an association or assignment operator and provide a mechanism for efficiently managing a large number of items. In simple terms, **groups associate device interfaces with roles**.

In practice, groups are often created and in a one-to-one relationship to the roles with which they will be associated, for example, a group named "Fireman" that is always associated with a role named "Fireman" - there may also be other naming conventions that make more sense for a given organization. If you assigned several device interfaces to a group called "Mobile Units," you could provision appropriate QoS parameters for the whole set of units in a single operation by assigning the role "Policeman" to the "Mobile Units" group. If you subsequently reassign a different role, say "Fireman" to the group, AirSync will generate new QoS instructions (based on the template tree indexed by the "Fireman" role) and send them to the appropriate units to be implemented.

With respect to provisioning QoS, a group can be referenced or "be pointed to" by zero one or more device interfaces. A group can reference or "point to" zero or one role. The key point for this section is to understand that **a group associates zero or one user role with zero or more device interfaces**. Groups have a few other interesting attributes that will be discussed in more detail later.

Note: you may perform Step 6 and Step 7 below in any order.

Step 6: Assign *Device Interfaces* to *Groups*

A device interface can only be a member of one group, but **a group can have several different device interfaces associated with it**. If a device interface is assigned to the "Fireman" group, that same device interface can't be simultaneously associated with any other group, but the Fireman group can be associated with many device interfaces.

Step 7: Assign *Groups* to *Roles*

A given group can only be associated with one role, but **a role may have more than one group associated with it**. If a group is assigned to the "Fireman" role, that same group can't be simultaneously associated with any other role, but the "Fireman" role can be associated with many groups.

After completing both Step 6 and Step 7, you will then have a template tree structure similar to the one shown in Figure 13. Device Interface "Ath0" on device Mobile 1 (upper left portion of the tree) has been assigned to the "Fireman" group. Notice that device "Mobile 1" has multiple interfaces. Device interface "Ath1" of device "Mobile 1" (upper middle portion of the tree) could be independently assigned to a different group.

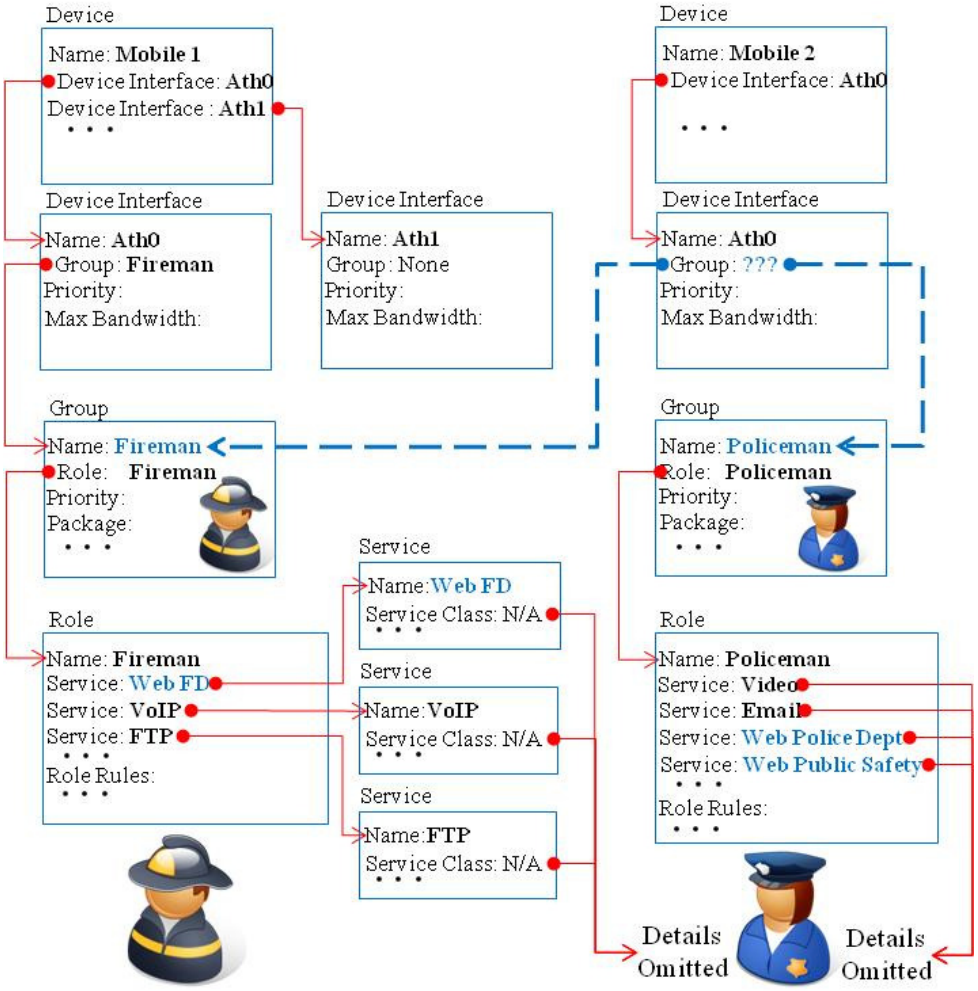


Figure 13. Template Tree structure shows relationship between Device Interfaces, Groups and Roles

Still referring to Figure 13, notice that Device Interface "Ath0" on device Mobile 2 (upper right portion of the tree) has been assigned to group "???", with blue dashed arrows pointing to either the "Fireman" group or the "Policeman" group. This device interface can only point to one group. The arrows represent the one-to-many relationships. There can be only one source (tail-end) for every arrow, but multiple arrows (head-end) can point to a common target object.

Try to visualize how AirSync's behavior would differ if the device interface pointed to one group instead of the other. If this device interface pointed to the Fireman group, AirSync would generate instructions to create three distinct queues (per traffic direction) for this interface - one queue for each of the services defined in the "Fireman" role - whenever it establishes an association with another managed wireless device, for example a radio mounted on a city lamppost. If this device interface pointed to the Policeman group, AirSync would generate instructions to create four distinct queues (per traffic direction) for this interface

Step 8: Use AirSync to Monitor and Adjust Policy Compliance.

The final step involves using AirSync's statistical visualization tools to periodically monitor performance and adjust policy as needed.

Summary

In summary, the human administration process involves:

- Gaining an understanding of network traffic flows and characteristics.
- Defining user needs and organizational priorities.
- Modeling them as business rules (building template trees) in AirSync.
- Monitoring network performance and making adjustments as necessary.

Figure 14 shows the high-level, end-to-end relationship between the AirSync objects used to implement QoS.

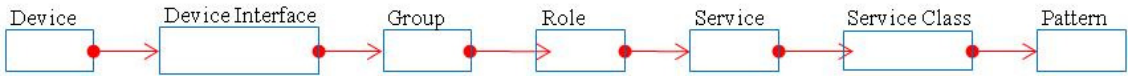


Figure 14. The High-Level Relationship between AirSync objects

An End-to-End QoS Example

The following example assumes that the QoS template tree has been created in accordance with Figure 13. A multi-interface Wi-Fi bridge device has been installed on a municipal lamppost at the corner of Broadway St. and Main St. It has three interfaces. Ath0 is not currently in use, but could function in the future as a wireless backhaul link to city hall. Interface Eth0 is a wired backhaul link to city hall. Interface Ath1 runs in Wi-Fi "access-point" mode, as shown in Figure 15.

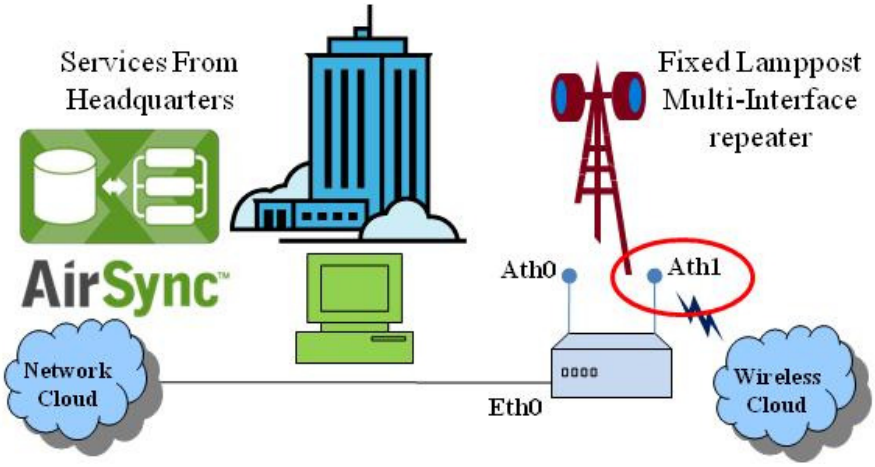


Figure 15. A Municipal Lamppost on Broadway and Main with no Devices Associated

Device "Mobile 1" is mounted in a fire engine. It has three device interfaces: Interface Ath0 runs in Wi-Fi station mode ready to associate with any Wi-Fi access point(s) in range. Interface Ath1 runs in Access Point mode to support any wireless devices (handhelds, or PCs) near the fire engine. Interface Eth0 supports PC devices wired into the fire engine. Interface Ath0 has been assigned to the "Fireman" group which in turn has been assigned to the "Fireman" role as shown in Figure 16.

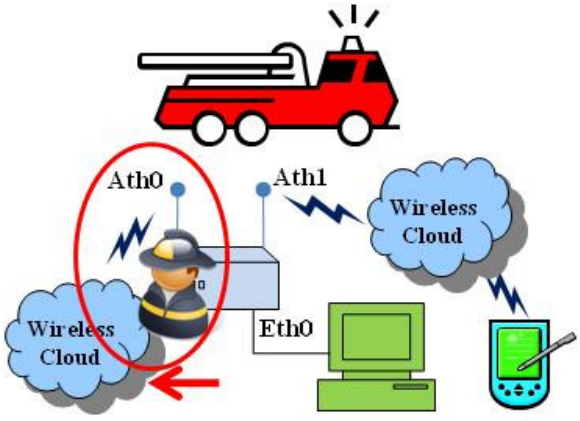


Figure 16. Device Mobile 1 has three interfaces. Ath0 has been assigned to the Fireman Group/Role

Both the lamppost radio at Broadway and Main and the radio mobile 1 mounted in the fire engine have been registered as "managed devices" in AirSync. Assume at first that the fire engine is parked in the station with the radio powered off and that no other units have associated with the lamppost radio. At his point in time, there are no QoS structures (filters, queues) present on either radio device. The AirSync system is monitoring the network waiting for a significant network event to occur.

Now the fire engine leaves the station and proceeds in response to an event near Broadway and Main. The fire engine gets close enough to associate to the lamppost radio at Broadway and Main as shown in Figure 17.

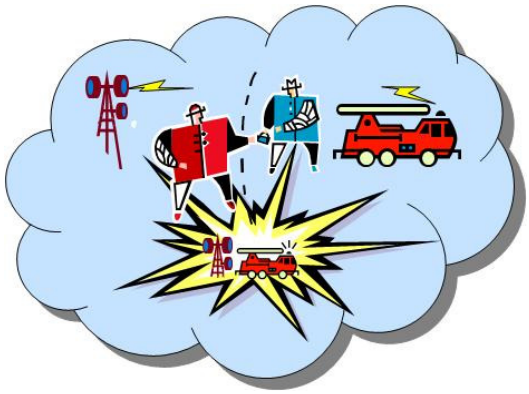


Figure 17. Mobile Device Association with Lamppost Triggers AirSync QOS Mechanisms

Soon after the association event, AirSync detects and reports it as shown in **Figure 18**. It recognizes that the lamppost radio now has a registered device associated with it. It determines by MAC address that device interface "Ath0" on device "Mobile 1" is now associated with device interface "Ath1" on (lamppost) device "Broadway and Main".

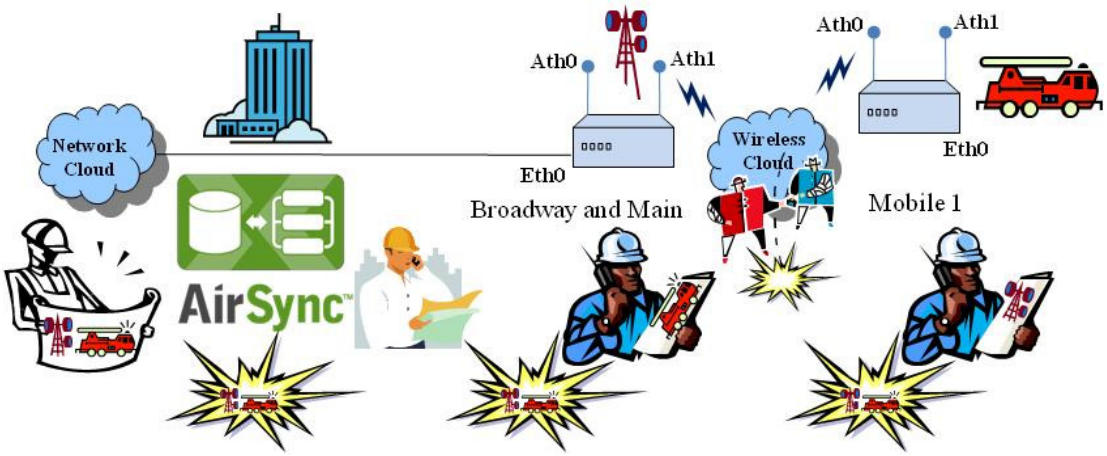


Figure 18. AirSync Monitors Detects and Reports Network Events such as Device Associations

AirSync consults its template tree and determines that interface "Ath0" in device "Mobile 1" is associated with group "Fireman" and that group "Fireman" is associated with role "Fireman." **Prior to generating any QoS instructions, AirSync evaluates the current network conditions**. AirSync considers many factors including whether any other devices are associated to the lamppost at Broadway and Main, what roles are assigned to other associated devices, link quality, modulation rate. AirSync computes bandwidth allocation parameters **starting from the values stored in the templates, but modified as appropriate due to current network conditions**.

AirSync generates a set of QoS instructions for building queues, based on the QoS parameters retrieved from the Web, VoIP, and FTP services defined for the role "Fireman," but adjusted for the current state of the network. AirSync also includes instructions for building the appropriate packet filters (patterns) for recognizing each type of traffic flow (service class), as shown in **Figure 19**.

Figure 19. AirSync generates instructions based on templates, network conditions, roles of devices

The AirSync server sends instructions to agents on the lamppost radio, as well as to agents on the mobile unit as shown in **Figure 20**.

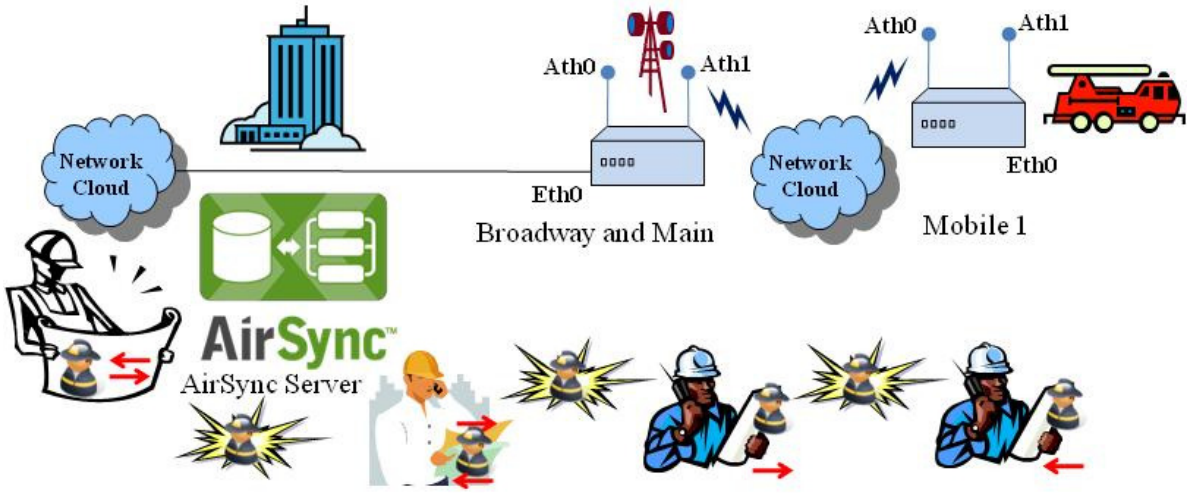


Figure 20. The AirSync server communicates QoS instructions to agents on managed devices

Agents on the managed devices receive the instructions and implement appropriate packet filters and queues for upstream and downstream interfaces, based on the roles of the associated devices and the network conditions at the time of association.

As shown in **Figure 21**, the agents on **the lamppost device will build downstream shaping rules** for egress (outgoing traffic) interface Ath1 based on the downstream QoS parameters defined for each service referenced by the role "Fireman" (Web, VoIP, FTP), and the patterns in the service classes referenced by each service (one queue and one set of packet filters for each service). The agents on **the mobile device will build upstream shaping rules** (packet filters and queues) for egress interface Ath0 based on the upstream QoS parameters and the same pattern definitions.

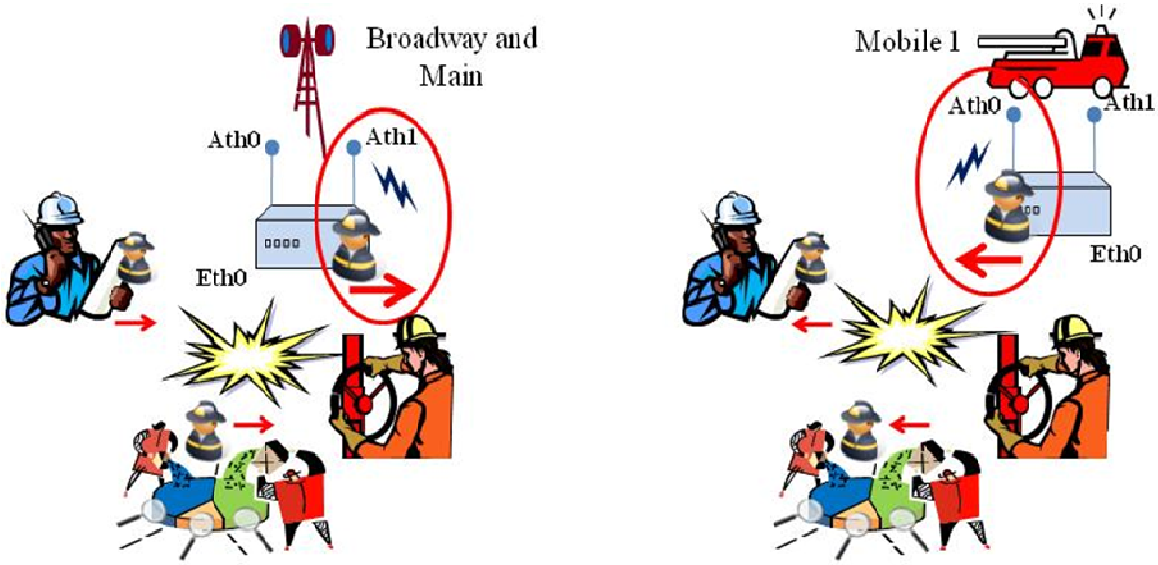


Figure 21. AirSync client agents implement packet filters and queues according to instructions

As other devices establish and tear down radio associations with the lamppost device, AirSync will again follow the bandwidth allocation/arbitration process shown previously in **Figure 19. AirSync generates instructions based on templates, network conditions, roles of devices**, the stored business rules (SLAs) but may make adjustments if necessary, for instance if the network is undersubscribed (there is excess uncommitted bandwidth after all the minimum SLAs have been satisfied) or oversubscribed (too many devices have associated and AirSync cannot meet the minimum SLAs).

AirSync's QoS / bandwidth allocation algorithms have some interesting implications:

- Note that the templates stored on the server do not consume or allocate any bandwidth so **no bandwidth is statically allocated or "nailed-up"**. The templates are just blueprints representing the organization's network usage / QoS policy.

- Bandwidth is allocated dynamically, based on user role, and current network conditions, in near real-time but only after a device association occurs.
- When the device association ends, the bandwidth allocation commitments are returned to the pool of available bandwidth and AirSync reconsiders how to allocate it, and any other available bandwidth, between devices that are still associated with the lamppost radio.

- As a result, AirSync provides built in QoS support for roaming mobile devices. QoS rules or instructions will follow a mobile device around the network as it changes its associations between various intermediate access point devices.

The AirSync Bandwidth Allocation Process

You can think of a service level agreement (SLA) as a commitment or a promise to deliver at least the specified minimum amount of bandwidth to the AirSync *service class* object associated with the *service* object that defines the particular SLA.

Note that the SLA commitment applies to the entire *service class* in the aggregate (a set of traffic flows), not to each individual traffic flow within it. For example, consider the "Web Public Safety" service provisioned with 100-200 KBps of downstream bandwidth in Figure 12 on page 58. A user could start zero, one or many simultaneous web sessions that match the patterns of the associated service class. The SLA applies to the service class as a whole, not to each individual web session.

At any point in time, depending on instantaneous user load, the network either has enough bandwidth available to satisfy the sum of promised bandwidth allocations in the set of all relevant SLAs (*services*) or it doesn't. This section explains how AirSync intelligently manages bandwidth allocation in either case, and how to control AirSync's bandwidth allocation mechanism in accordance with an organization's network usage policy.

Understanding the Bandwidth Allocation Range

Services are provisioned with a range of bandwidth such as 100-200 KBps, in each direction and a priority for the service. To simplify this discussion we will consider only the downstream direction for now. Also since AirSync provisions bandwidth in terms of **Kilobytes per second (KBps)**, assume KBps as the unit of measure whenever units are omitted.

The first part of the bandwidth range determines SLA compliance. In the example above, 100 represents a *minimum* bandwidth guarantee. The system meets the SLA when it can successfully allocate this minimum bandwidth level to the traffic in the *service class* associated with the *service* defining the SLA. The system fails to meet the SLA when it fails to allocate at least this coefficient of bandwidth

The second part of the bandwidth range controls the allocation of excess bandwidth. This enables the traffic in a service class to burst a controlled amount above its provisioned minimum bandwidth allocation. In the example above, 200 represents a maximum boundary on bandwidth allocation for the service. When excess bandwidth is available, it may be allocated to traffic in the service class allowing it to burst only up to this specified value. **This value serves as a cap that limits the excess bandwidth allocated to a service** such that if excess bandwidth still remains after meeting this cap, it will be made available to other services, in effect controlling the relative degree that traffic flows in different service classes will be able to burst above the minimum SLA values.

Consider the total guaranteed downstream bandwidth, for a given access point or base station device interface, to be the sum of all minimum bandwidth commitments in the relevant SLAs for all connected subscriber station (or CPE device) interfaces. More specifically, it's the sum of the minimum bandwidth values defined in each *service* template defined in each *role* template (if any) assigned to each remotely connected subscriber/CPE device interface.

The Three Bandwidth Allocation Cases

There are three possible cases when attempting to allocate bandwidth. In two of the cases it is possible to meet SLAs, in one of them it is not:

- **Demand is less than capacity**. It is possible to satisfy all SLAs. The network is undersubscribed. There is excess bandwidth available to allocate to traffic flows allowing them to burst above their minimum provisioned SLAs. Bandwidth can be freely dispensed, according to the user-defined rules.
- **Demand is equal to capacity**. It is possible to satisfy all SLAs. The network is fully subscribed. There is no additional bandwidth available to allow traffic flows to burst above their minimum provisioned SLAs. This is a rare case and not very interesting in terms of intelligent bandwidth allocation.
- **Demand is greater than capacity**. It is impossible to satisfy all SLAs. The network is oversubscribed. How should the insufficient bandwidth capacity be allocated? **Which SLAs should be honored? Which should not?** This state is called Service Level Degradation (SLD), because AirSync must resolve which of the SLAs it will meet and which it will not. AirSync has an SLD algorithm that performs this arbitration on a priority basis. When it's impossible to meet all relevant SLAs, AirSync's algorithm ensures that service degrades in an orderly rules-based fashion consistent with organizationally defined priorities.

Understanding Link Capacity

AirSync must know the capacity of a given link before it can determine which of the three bandwidth allocation cases above applies. AirSync determines how to allocate bandwidth based on the relationship between current link capacity and current demand. AirSync has an embedded bandwidth estimator that will attempt to dynamically discover the approximate bandwidth capacity of a link, or alternatively, a system administrator can statically set the maximum bandwidth capacity for the link.

Understanding How Priorities affect Bandwidth Allocation

AirSync uses a priority system to allocate excess bandwidth when the network is either undersubscribed or over-subscribed. Priorities range in value between 1 (most preferred) and 7 (least preferred). In the AirSync GUI, priorities can be set in multiple objects:

- Each service can have a priority assigned to it. **The priority values set for *services* govern how excess bandwidth is allocated** during periods of network under-subscription.
- Each group can have a priority assigned to it. **The priority values set for *groups* govern how insufficient bandwidth resources get allocated** during periods of network over-subscription.

Service Level Degradation (SLD) – Unable to Honor Minimum SLAs

In this example scenario, we will examine how AirSync allocates bandwidth in situations when there is not enough bandwidth to satisfy all SLA for all the relevant services provisioned for a particular device interface. Consider the topology in Figure 22 and the data in Table 1. The link has a capacity of 1000 KBps.

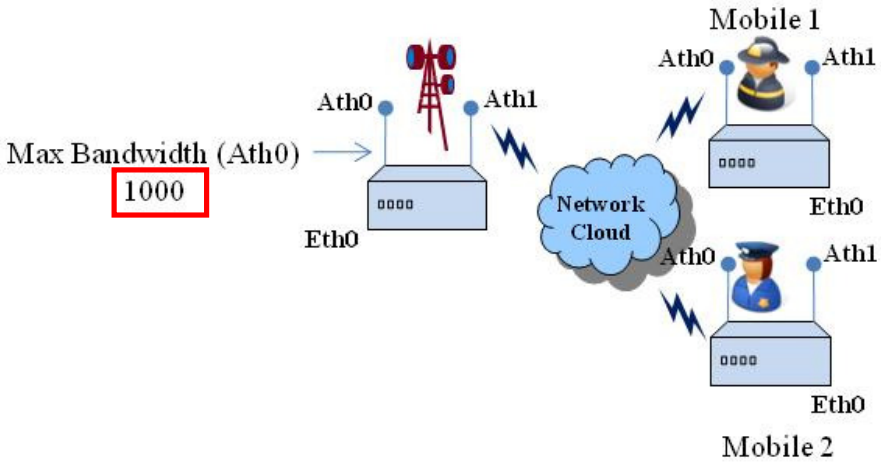


Figure 22. A bandwidth allocation example with SLD

Minimum Link Bandwidth 1000 KBps						
Group (ath0)	Group Priority	Role (ath0)	Service	Service Priority	Min BW	Max BW
1	1	1	App A	2	200	300
			App B	3	300	500
			App C	5	400	450
2	2	2	App D	1	500	800
			App E	4	100	300
			App F	6	50	250

Table 1. Bandwidth characteristics for SLD example

A police department unit and a fire department unit have associated with the lamppost's ath1 interface. Reviewing Table 1, notice that each mobile unit's ath0 interface (which have associated with the lamppost's ath1 interface) has been assigned to a distinct group with a distinct role. Each group has a distinct group priority and each role has three distinctly provisioned services. Each service has a distinct service priority.

AirSync consults its templates and retrieves the QoS parameters in an attempt to generate QoS instructions. By summing the minimum bandwidth requested for each instance of each service, AirSync determines that demand exceeds capacity. As summarized in Table 2, there will not be enough bandwidth to meet all SLAs. AirSync must invoke its SLD algorithm to arbitrate bandwidth allocation.

Service	Prio-g	Prio-s	Min	Max	Spread
App A	1	2	200	300	100
App B	1	3	300	500	200
App C	1	5	400	450	50
App D	2	1	500	800	300
App E	2	4	100	300	200
App F	2	6	50	250	200
Totals			1550	2600	1050

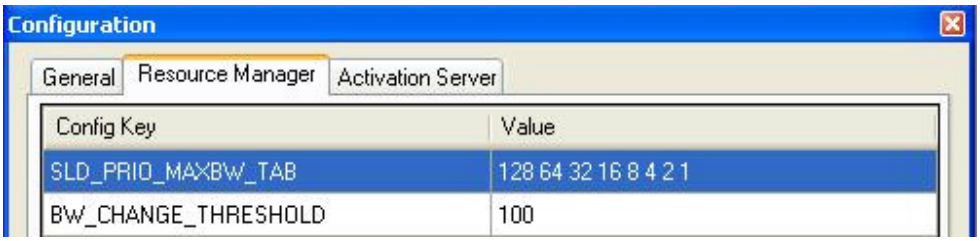
1000	Max BW
0	Excess BW
550	Shortage BW

Table 2. Summary of Bandwidth Allocation Characteristics

Note that by design, the AirSync SLD algorithm is not a strict priority-based queuing algorithm. In strict priority-based algorithms, no lower priority services (for example, those with group priority 2) would get any bandwidth allocated until all the higher priority services (those with group priority 1) have been allocated their minimum bandwidth allocations. In these schemes, less important flows may be subject to queue starvation and receive zero bandwidth.

In contrast, the AirSync algorithm protects lower priority services from queue starvation, but heavily weights the allocation in favor of the higher priority services. The net result is that the high priority services get the lion's share of what they need to meet SLAs, but lower priority services still get a measure of bandwidth, as well.

AirSync retrieves the parameter SLD_PRIO_MAXBW_TAB from the database. (This value is a use-adjustable configuration item available from the **Resource Manager** tab of the **System Configuration** item available from the **Tools** menu as shown in Screen Capture 53.) This string represents the relative weighting assigned to each group priority level by the SLD algorithm. We recommend that you leave it at its default value, 128 64 32 16 8 4 2 1, until you become familiar with SLD behavior.



Screen Capture 53. The SLD priority weighting table

Without getting into all of the details of the SLD algorithm, suffice it to say that AirSync relies on group priority to arbitrate bandwidth allocation during periods of network oversubscription. It orders the relevant services by group priority and multiplies each minimum allocation value (retrieved from the templates in the database) by the appropriate weighting factor for that group's priority. After a few minor adjustments, it performs a linear scaling operation to finalize the QoS instructions that it generates and sends to the affected device(s). The results are shown in Table 3 and charted in Figure 23.

Service	Prio-g	Prio-s	Min	Max	Spread	Multiplier	SLA pct	Burst	adm 4
App A	1	2	200	300	100	64	83.00%	0	166
App B	1	3	300	500	200	64	81.33%	0	244
App C	1	5	400	450	50	64	81.50%	0	326
App D	2	1	500	800	300	32	40.80%	0	204
App E	2	4	100	300	200	32	40.00%	0	40
App F	2	6	50	250	200	32	40.00%	0	20
Totals			1550	2600	1050		64.52%		1000

Table 3. Bandwidth Allocation after SLD algorithm

Note in Table 3 and Figure 23 that none of the services had its SLA totally satisfied, but that it has been satisfied between 81.5 and 83 percent for the services with a group priority 1 and between 40 - and it satisfied 40.8 percent for services with group priority 2. Note that true to the weighting values (64 and 32), services in group priority 1 got about twice the percentage of their SLAs met (~80, ~40).

AirSync service classes have attributes called Min Up BW and Min Down BW described on page 80 and shown in **Screen Capture 55**. By setting values for these attributes, AirSync will avoid allocating any bandwidth whatsoever if the value specified here cannot be allocated. This allows higher priority services to pass bandwidth to other services if this specified value cannot be achieved during SLD.

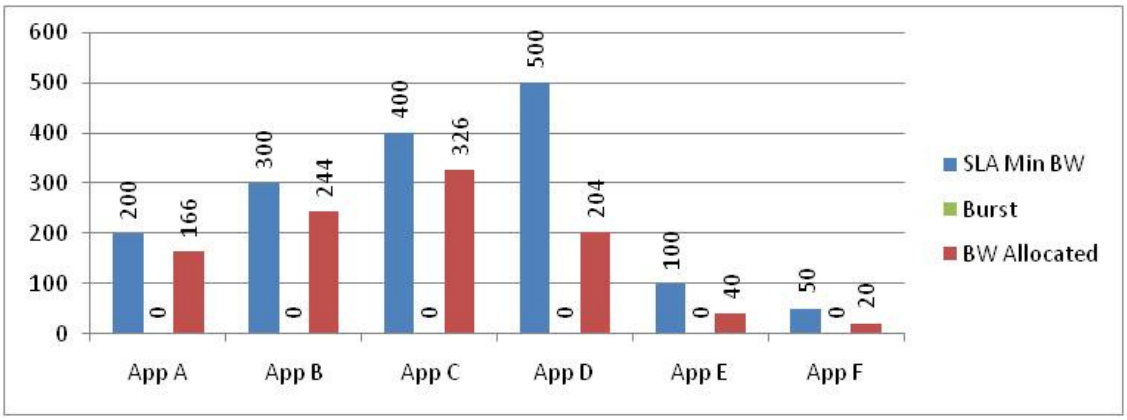


Figure 23. Final Arbitration of Bandwidth Allocation after SLD algorithm

Allocation of Extra Bandwidth After Honoring Minimum SLAs

In the next example scenario, we will examine how AirSync allocates bandwidth in situations when extra bandwidth remains available after satisfying all minimum SLAs for all the relevant services provisioned for a particular device interface. Consider the topology in Figure 24 and the data in Table 4. Link capacity has doubled to 2000 KBps.

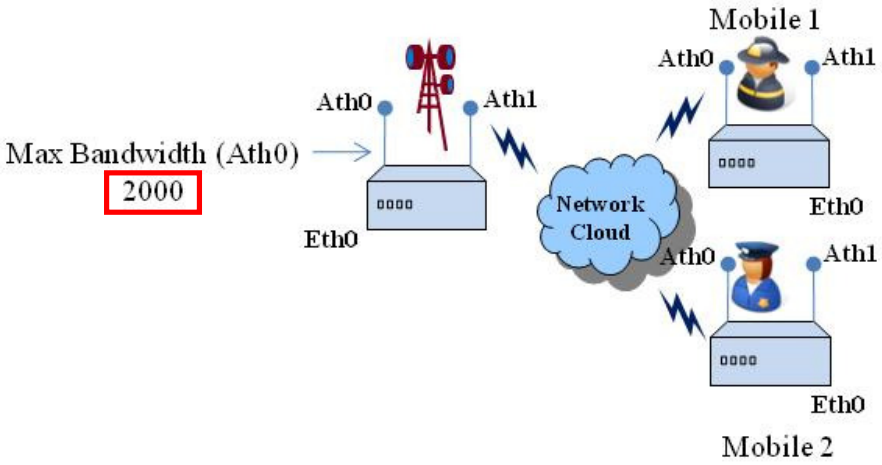


Figure 24. A bandwidth allocation example with excess bandwidth available for bursting

Minimum Link Bandwidth 2000 KBps						
Group (ath0)	Group Priority	Role (ath0)	Service	Service Priority	Min BW	Max BW
1	1	 1	App A	2	200	300
			App B	3	300	500
			App C	5	400	450
2	2	 2	App D	1	500	800
			App E	4	100	300
			App F	6	50	250

Table 4. Bandwidth characteristics for SLD example

AirSync consults its templates and retrieves the QoS parameters in an attempt to generate QoS instructions. By summing the minimum bandwidth requested for each instance of each service, AirSync determines that capacity exceeds demand. There will be extra bandwidth available to allow some services to get an additional burst of bandwidth capacity.

AirSync uses the service priority to determine how to allocate this excess bandwidth. The services are sorted by their service priorities and then extra bandwidth is allocated to each service in order until all the excess bandwidth has been allocated or the maximum bandwidth value for the service (the higher number in the bandwidth range) has been reached. Table 5 summarizes the results charted in Figure 25.

Service	Prio-g	Prio-s	Min	Max	Spread	Multiplier	SLA pct	Burst	adm 4
App D	2	1	500	800	300	32	100.00%	300	800
App A	1	2	200	300	100	64	100.00%	100	300
App B	1	3	300	500	200	64	100.00%	50	350
App E	2	4	100	300	200	32	100.00%	0	100
App C	1	5	400	450	50	64	100.00%	0	400
App F	2	6	50	250	200	32	100.00%	0	50
Totals			1550	2600	1050		100.00%		2000

Table 5. Bandwidth allocation with excess available bandwidth for bursting

Notice that App D, and App A (service priority 1 and 2) were allowed to burst up to their maximum configured values. App B (service priority 3) was allowed to burst, but only by 50 KBps because that exhausted all excess bandwidth. None of the apps with lower service priorities (App E, priority 4; App C, priority 5; App F, priority 6) got any extra burst capacity.

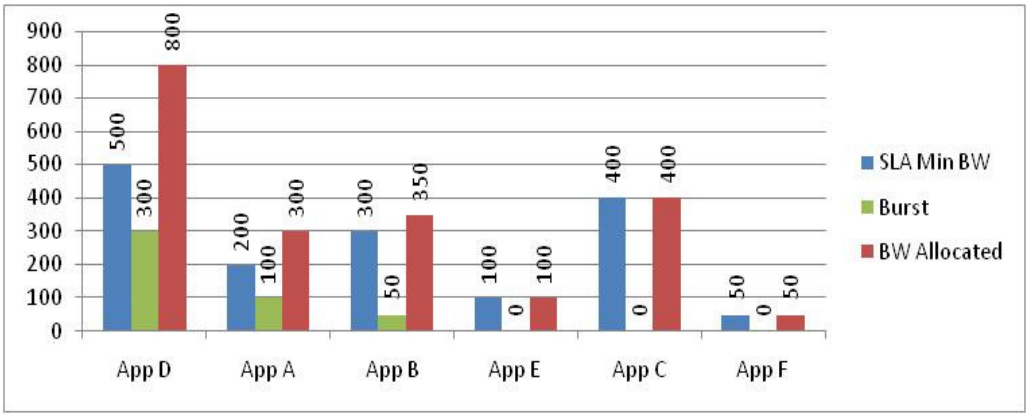


Figure 25. Final Bandwidth allocation with excess available bandwidth for bursting

How does AirSync handle resolution between items with identical priority

AirSync uses an internal heuristic to order service items, for example by order of MAC address (of the associated interface) or by the order in which connections were established. For practical purposes, this tie breaking mechanism should be considered non-deterministic. If it's really important, try to manipulate the priority assignments such that there could never be a tie.

The Default Queue and the Management Queue

What happens to traffic flows that do not match any specific pattern and therefore do not get any specifically provisioned bandwidth? AirSync maintains a default queue for all traffic flows that don't match any service class. These flows can still traverse the network, but without any specifically provisioned bandwidth guarantees. AirSync also maintains a special queue for its own internal management traffic so system administrators won't need to worry about the provisioning details for AirSync management traffic between nodes.

Understanding AdHoc Rules

AdHoc Rules allow an administrator to specify certain "trigger conditions" that will cause AirSync to manipulate QoS settings on the fly, based on network events such as topology changes and signal changes. In near real time, and without human intervention, they conditionally modify the basic SLAs defined in services. These rules are defined as part of the AirSync **Role** object. The user interface details for working with AdHoc Rules are discussed in the section beginning on page 89.

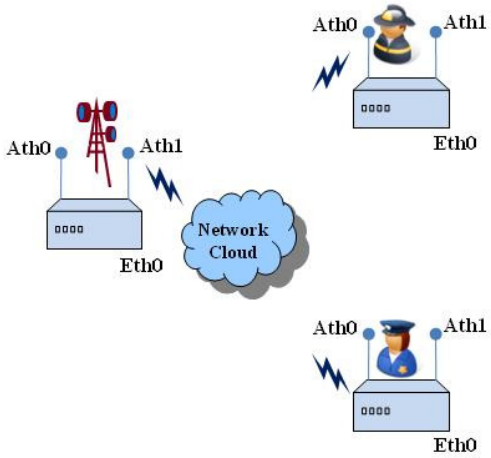
- AdHoc Rules can automatically be triggered in response to the following stimulus events:

- Network Topology changes as indicated by changes in the number of stations associated with a device.
- Signal degradation or improvement as indicated by changes in the bit rate or modulation scheme used over a link.

- AdHoc Rules can select which services to modify based on:
  - The role assigned to one or more device interfaces
  - The bit rate (modulation scheme) reported by one or more devices

- AdHoc Rules can modify the parameters of one or more services by:
  - Disabling one or more services while the trigger condition is true
  - Increasing or decreasing the bandwidth allocation in either direction for one or more services
  - Increasing or decreasing the priority for one or more services

Consider the following example. Initially two mobile devices have associated to a fixed lamppost access point unit. One unit has been assigned the Fireman role, the other has been assigned the Policeman role. The Fireman and Policeman Roles each have three services provisioned as indicated in Figure 26.

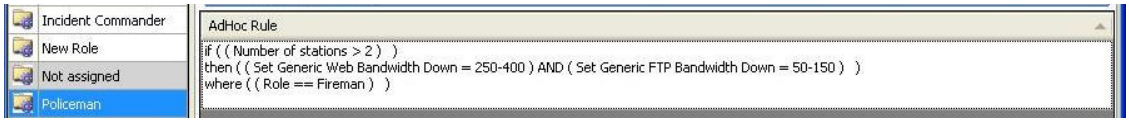


Service	Web	VoIP	FTP
BW Up	50-80	20-20	10-30
BW Down	500-800	20-20	100-300
Priority	1	4	6

Service	Email	Video	Email
BW Up	40-45	30-50	20-30
BW Down	400-450	300-500	200-300
Priority	5	3	2

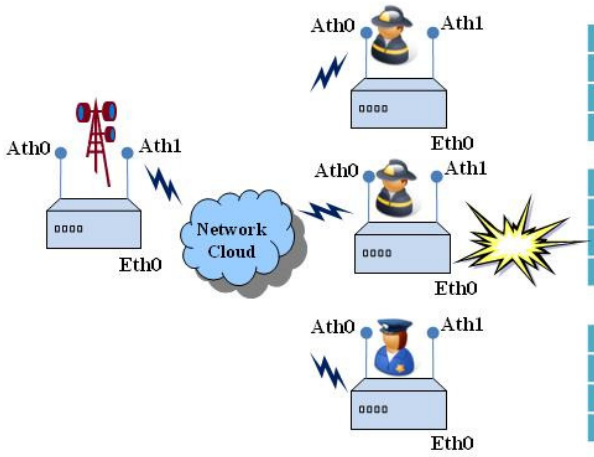
Figure 26. A simple scenario before an event triggers AdHoc Rule

A simple AdHoc Rule has been created for the Policeman role as shown in Screen Capture 54. The rule states that whenever there are more than two stations associated to the lamppost, and at least one of the connected stations has the Policeman role (implied because this rule is part of Policeman role), adjust the parameters for stations in the Fireman role. More specifically, cut the downstream rate for web from 500-800 to 250-400, change the priority from 1 to 7, and cut the downstream rate for web from 100-300 to 50-150.



Screen Capture 54. A Simple AdHoc Rule for the Policeman role affects services for the Fireman role

Now a third mobile device, which has also been assigned the Fireman role, forms an association to the same lamppost access point. This association event triggers AirSync to regenerate new QoS instructions. The association event triggers the AdHoc condition (number of stations > 2) for the rule on the Policeman. The QoS instructions are modified as described above to give a bias to the Policeman unit, consistent with organizational policy. The Fireman units effectively split bandwidth between themselves without affecting the Policeman unit as shown in Figure 27.



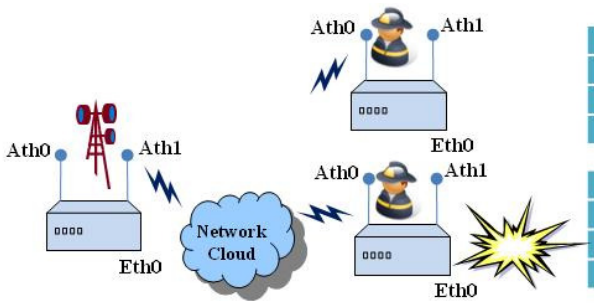
Service	Web	VoIP	FTP
BW Up	50-80	20-20	10-30
BW Down	250-400	20-20	50-150
Priority	7	4	6

Service	Web	VoIP	FTP
BW Up	50-80	20-20	10-30
BW Down	250-400	20-20	50-150
Priority	7	4	6

Service	Email	Video	Email
BW Up	40-45	30-50	20-30
BW Down	400-450	300-500	200-300
Priority	5	3	2

Figure 27. Event triggers AdHoc Rule on Policeman role to change services for Fireman Role

When the Policeman unit disassociates, the two firemen get their original provisioned values as shown in Figure 28.



Service	Web	VoIP	FTP
BW Up	50-80	20-20	10-30
BW Down	**500-800**	20-20	**100-300**
Priority	**1**	4	6

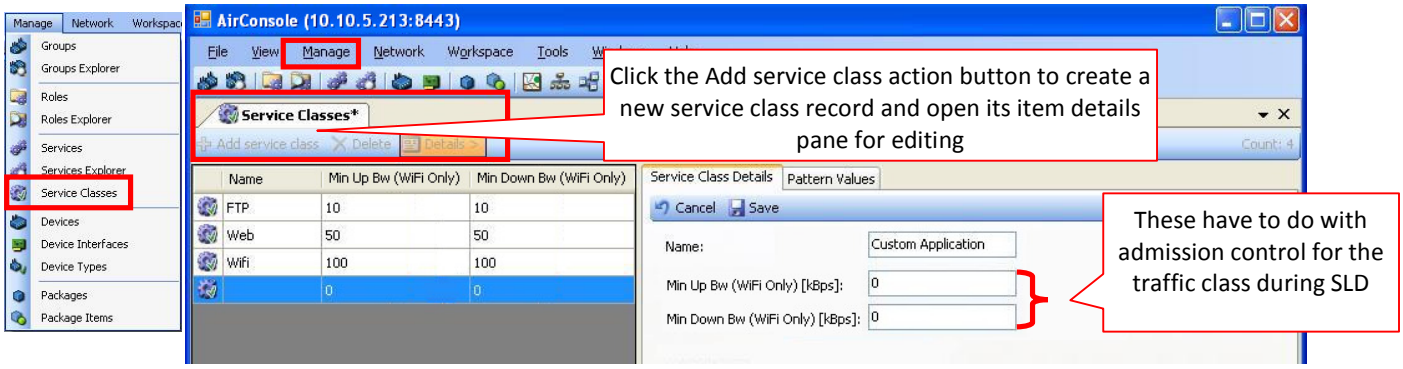
Service	Web	VoIP	FTP
BW Up	50-80	20-20	10-30
BW Down	**500-800**	20-20	**100-300**
Priority	**1**	4	6

Figure 28. Police unit disassociates, AdHoc Rule no longer applies, Fireman services restored

The GUI Mechanics of Implementing QoS

Working with Service Classes

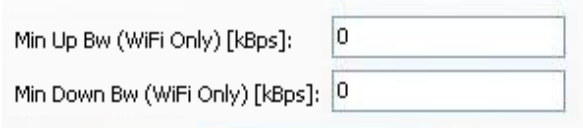
Service classes are discussed starting on page 50. To add, edit or delete a service class open the **Service Classes** item from the **Manage** menu. Clicking the **Add service class** action button creates a new service class record and opens the **Service Class Details** pane for editing the new record as shown in **Screen Capture 55**. An example of adding a service class for a custom application follows.



Screen Capture 55. Adding a Service Class for an Application

The Min Up BW and Min Down BW attributes

For each service class, you can specify optional values for the Min Up BW and/or the Min Down BW attributes. **These values provide a type of traffic admission control or conditional bandwidth allocation functionality** for the traffic class based on minimum bandwidth requirements, if specified, for either direction.



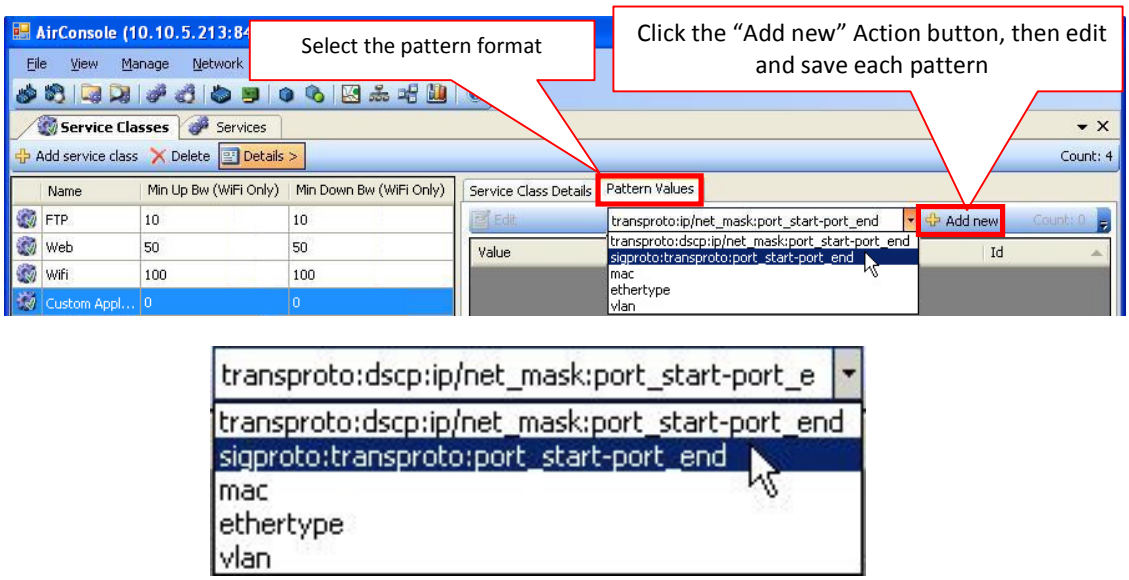
For example, assume that you have a special application that needs at least 250 Kbps in both directions to run effectively. Then, application performance begins to degrade to the point where users no longer consider using the application if it can't get at least this minimum amount of bandwidth in both directions. **Setting these values causes AirSync to conditionally generate QoS instructions** such that anytime it can't meet the minimum specified values (250 KBps downstream, 250 KBps upstream), it won't allocate any bandwidth for the class, even a compromised amount. Instead, the bandwidth will be made available for allocation to other traffic classes that can use it effectively, rather than wasting the bandwidth by allocating an insufficient amount that would only result in unacceptable application performance for the original application.

Assume that due to currently-congested network conditions, AirSync only has 175 Kbps of uncommitted bandwidth left to allocate. Instead of allocating an insufficient amount of bandwidth (only 175 KBps of the 250 KBps required), the system will refrain from allocating any bandwidth to an instance of this service class and instead keep it for allocation to instances of other traffic classes. When the congestion conditions improve to the point where AirSync could allocate the minimum specified values, it would allocate the bandwidth as normal.

Note that although these *service class* attributes are related to bandwidth allocation, they are not the primary controls for specifying bandwidth allocation parameters. The primary bandwidth allocation controls are the attributes available for each *service* (review page 51).

Adding Classification Patterns

Recall from the discussion on page 50 that service classes use patterns to classify distinct traffic flows. After saving the basic service class record, add one or more packet classification patterns to it by using **the Pattern Values** sub pane. The process involves selecting the appropriate pattern format, clicking the **Add new** action button, then editing and saving each pattern as shown below in Screen Capture 56.



Screen Capture 56. Adding Pattern Values to the Service Class

Choosing A Pattern Format for Packet Classification

There are three distinct pattern formats for classifying matching packets:

- The *pattern* format transproto:dscp:ip/net_mask:port_start-port_end is appropriate for many applications.
  - Transproto field can be equal to "TCP', "UDP" or "ANY" value. It matches packets by 4th layer transport protocol.
  - DSCP field is a 6 bits number covered in IP packet and is used by application with QoS support. It can be equal to decimal number from range 0 – 63 (0 means lowest and 63 highest priority) or "ANY".
  - IP address and netmask field with port range field determine service source socket.

See Table 6. Pattern example for more details.

Pattern example	Description
TCP:0:172.20.1.1/255.255.255.255:80-80	HTTP traffic between 172.20.1.1 host and subscriber
UDP:28:10.10.1.0/255.255.255.0:1-65535	any UDP traffic, with DSCP=28, between 10.10.1.0/24 network and subscriber
ANY:ANY:0.0.0.0/0.0.0.0:1-65535	any traffic

Table 6. Pattern example

- The *sigproto*:transproto:port_start-port_end pattern is appropriate for protocols that operate at higher stack layers such as SIP-based VoIP. Transproto and port range field have same role like in first pattern. For example, SIP:UDP:5060-5061 matches UDP, SIP traffic (service is available on any host, on ports 5060 and 5061).
- The *mac* pattern format classifies packets based on layer 2 (mac) addresses. For example, 00:00:12:ac:23:21 matches traffic between subscriber and host with 00:00:12:ac:23:21 MAC address.
- The *ethertype* pattern matches traffic due ethertype field in ethernet frame. This field can be equal to 4-chars, hexadecimal number (without 0x prefix) For example: 0800 pattern matches IPv4 traffic. For more details see Ethernet II (DIX Ethernet) specification.
- The *vlan* pattern matches traffic across specified vlans. Available range is 0 – 4095. For more information check IEEE 802.1Q specification.

Here are some useful pattern specification hints:

Hint: Don't forget the punctuation marks (":", "/","-")

Hint: When AirSync generates QoS instructions for downstream traffic, the patterns reference source protocols, ports and addresses. AirSync inverts the source and destination when it generates instructions for the upstream direction, so the patterns reference destination protocols, ports and addresses in the upstream direction.

Hint: The pattern format is a Write-Once attribute for the pattern. If you need to modify a pattern's format, delete the pattern and add it again.

Hint: To specify a specific IP host type its address and use a full length network mask such as 192.168.10.100/255.255.255.255. To specify any host use 0.0.0.0/0.0.0.0. To specify all hosts on the 192.168.10.0 / 24 network, use "192.168.10.0/255.255.255.0"

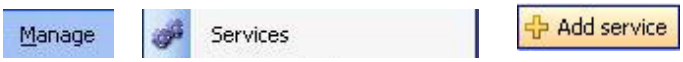
Hint: You must use dotted decimal notation when specifying a mask. You can't use CIDR notation such as 192.168.10.0 / 24

Hint: You can use ANY as a wildcard for transport protocols. For example, Domain Name Service (DNS) traffic can use TCP or UDP at layer 4, but generally runs on port 53, so ANY:172.16.1.100/255.255.255.255:53-53 matches all DNS traffic from the specific host 172.16.1.100 regardless of whether it's TCP or UDP as a transport protocol.

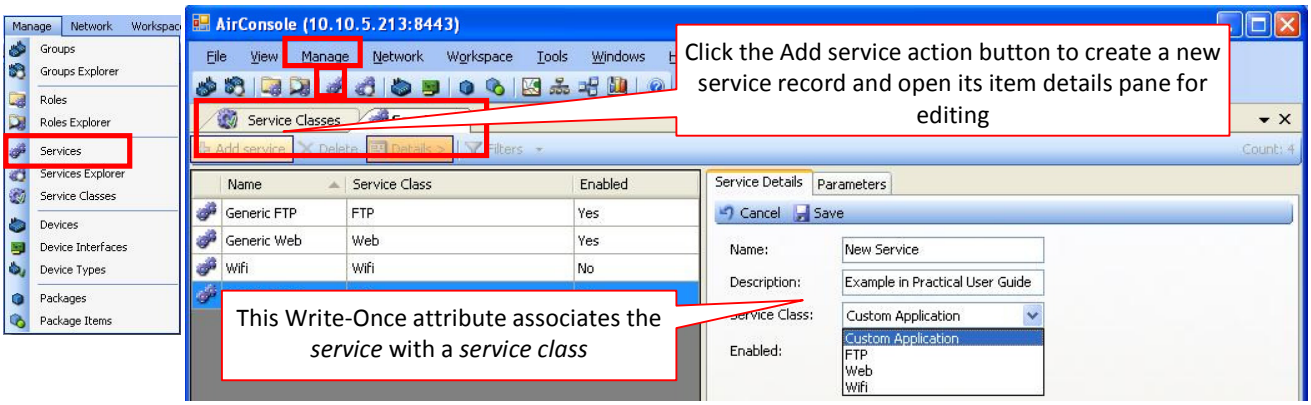
Hint: To specify a wildcard for ports, use a range like 0-0 or 0-65535.

Hint: To match traffic that uses a non-contiguous port range such as web traffic including HTTP on port 80 and HTTPS on port 443, simply specify two distinct patterns such as shown in Figure 12 on page 58 poniżej.

Working with Services



To add, edit or delete a service, open the **Services** item from the **Manage** menu or click it from the tool ribbon. Clicking the **Add service** action button creates a new service record and opens the **Service Details** pane for editing the new record as shown in Screen Capture 57.



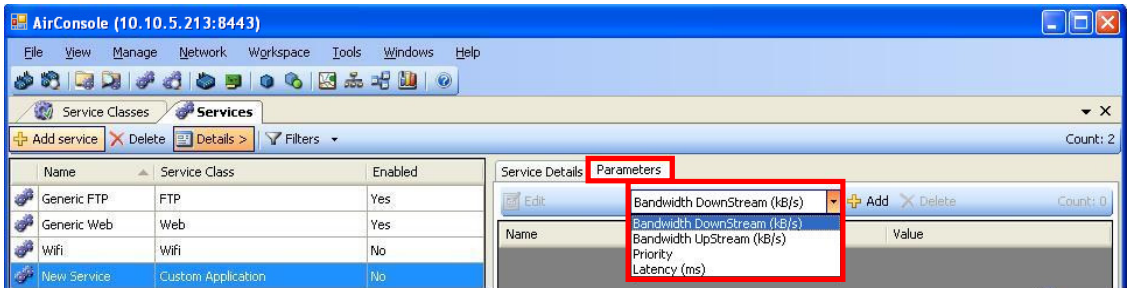
Screen Capture 57. Adding a Service for an Application

Associating a Service with a Service Class

The **Service Class** attribute can be assigned by selecting an existing service class from a drop-down list box. This Write-Once attribute determines the *service class* to which this *service* will be linked. This implies that you must create the service class you need before creating a service that relies on it. If you subsequently want to use another value, delete this service and re-add it using the desired value for its **Service Class** attribute.

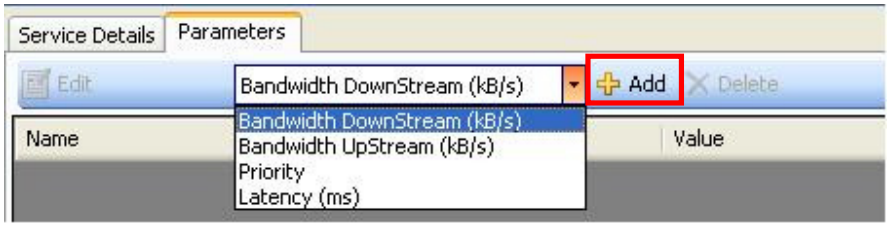
Provisioning the Service Parameters

To provision QoS parameters for the service, click the **Parameters** tabbed item on the **Service details** sub pane then select the parameter you wish to provision from the drop down list box as shown in Screen Capture 58.



Screen Capture 58. Selecting a QOS Parameter to provision for the service

Next click the **Add** action button as shown in Screen Capture 59.

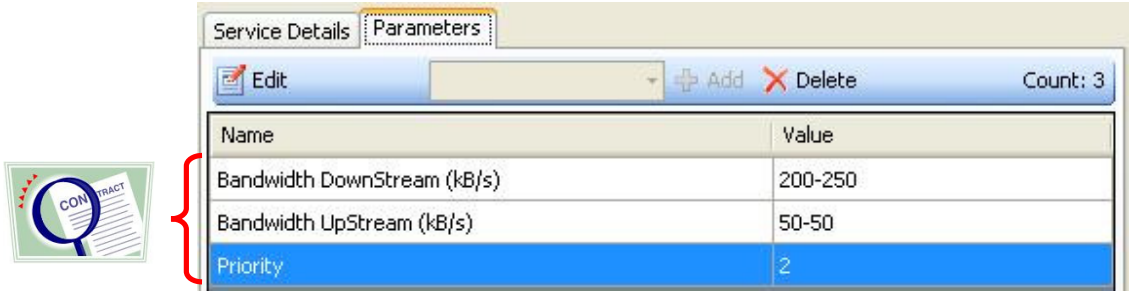


Screen Capture 59. Adding a QoS provisioning parameter to a service

In the lower right-hand portion of the Service Parameters Pane you will have an opportunity to enter a bandwidth allocation range. Specify a minimum value followed by a dash "-" followed by a maximum value as shown in Screen Capture 60. The meaning of this bandwidth range specification is discussed starting on page 68.

Screen Capture 60. Specifying a range value for the "Bandwidth DownStream" parameter

Add parameter values for the **Bandwidth Upstream**, **Bandwidth Downstream** and **Priority** parameters to the service. As you add the parameter values you will see them listed on the **Parameters** sub pane for the service as shown in Screen Capture 61. If at any time you wish to change a provisioned value (modify the terms of the SLA), you can select the parameter value and edit it.



Screen Capture 61. Specify Bandwidth Upstream, Bandwidth Downstream, and Priority

WiMAX QoS classes and WLAN WMM access categories, both depend of Bandwidth Upstream, Bandwidth Downstream and Latency parameters. For details see Table 7. WiMAX QoS classes definition as AirSync Service parameters and Table 8. **WiFi WMM access categories definition as AirSync Service parameters.**

WiMAX QoS classes:	AirSync Service Parameters:
UGS (Unsolicited Grant Service)	Latency defined and maximal bandwidth equal to minimal bandwidth.
RTPS (Real Time Polling Service)	Latency defined and maximal bandwidth higher than minimal bandwidth.
NRTPS (Non Real Time Polling Service)	No latency defined and minimal bandwidth higher than zero.
BE (Best Effort)	No latency defined and minimal bandwidth equal to zero.

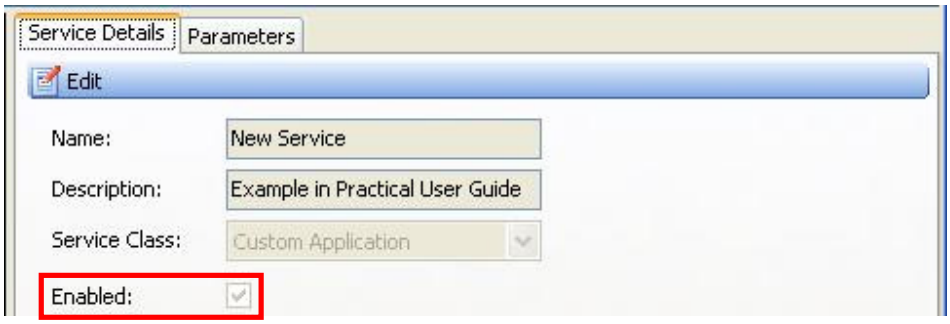
Table 7. WiMAX QoS classes definition as AirSync Service parameters

WiFi QoS access categories:	AirSync Service Parameters:
VI (voice)	Latency defined and maximal bandwidth equal to minimal bandwidth.
VO (video)	Latency defined and maximal bandwidth higher than minimal bandwidth.
BE (Best Effort)	No latency defined and minimal bandwidth higher than zero.

BG (Background)	No latency defined and minimal bandwidth equal to zero.

Table 8. WiFi WMM access categories definition as AirSync Service parameters

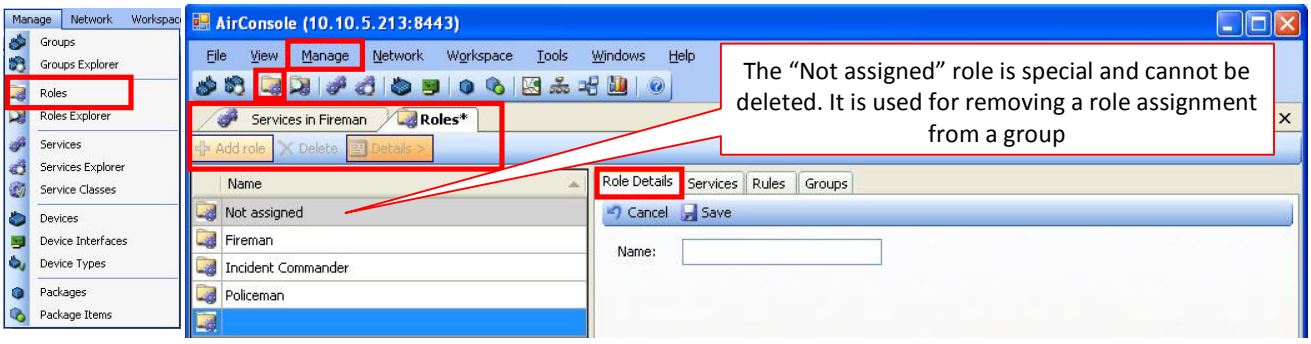
Note that the **Enabled** attribute on the **Service Details** sub pane will show as checked or true as shown in Screen Capture 62 if and only if at least 2 (Priority and one of Bandwidth) of four QoS parameters have been provisioned for this service. Specifying the set of three QoS parameters defines a basic SLA for service.



Screen Capture 62. Service will show as enabled if and only if all three QoS parameters are defined

Working with Roles

To add, edit or delete a role, open the **Roles** item from the **Manage** menu or click it from the tool ribbon. Clicking the **Add role** action button creates a new role record and opens the **Role Details** pane for editing the new record as shown in Screen Capture 63. An example of adding a role for a custom application follows.



Screen Capture 63. Adding a New Role

Note that the **Not Assigned** role is special and can't be deleted. It is used for removing a role assignment from a group. There isn't a lot to manipulate on the **Role Details** sub pane as shown in Screen Capture 64, but you can change the role name there if desired.

Screen Capture 64. Role Name can be edited from "Role Details" sub pane

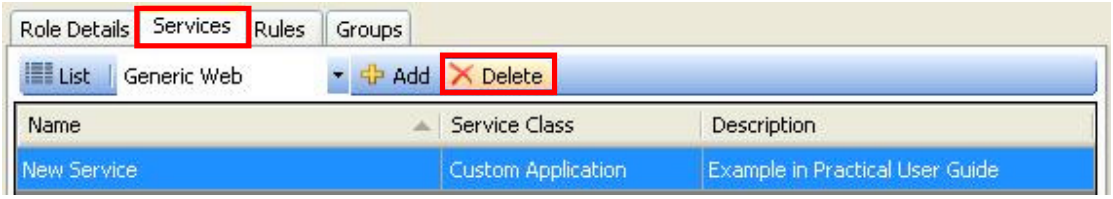
Associating services with a role

To associate one or more services with a role, click the **Services** sub pane item for the selected role and select the desired service from the drop-down list. Click on the **Add** action button as shown in Screen Capture 65. Recall from the discussion starting on page 52 that a role can be associated with zero or more services.

Alternatively, you can drag and drop a service object (from **Service Explorer**) onto a defined role object in the "Roles Explorer" window area, if open. The section titled "Drag 'n' Drop" Operations with the Explorer Windows" beginning on page 22 shows examples of drag 'n' drop operations. See also Screen Capture 21. Dragging and Dropping a Service from "Services Explorer" to "Role Explorer" on page 22.

Screen Capture 65. Selecting and associating a service with a role from the services sub pane

To disassociate a service from a role, select the service item from the sub pane and click the **Delete** action button as shown in Screen Capture 66. You cannot disassociate a service attribute value from a role by dragging it from the **Roles Explorer** window area.



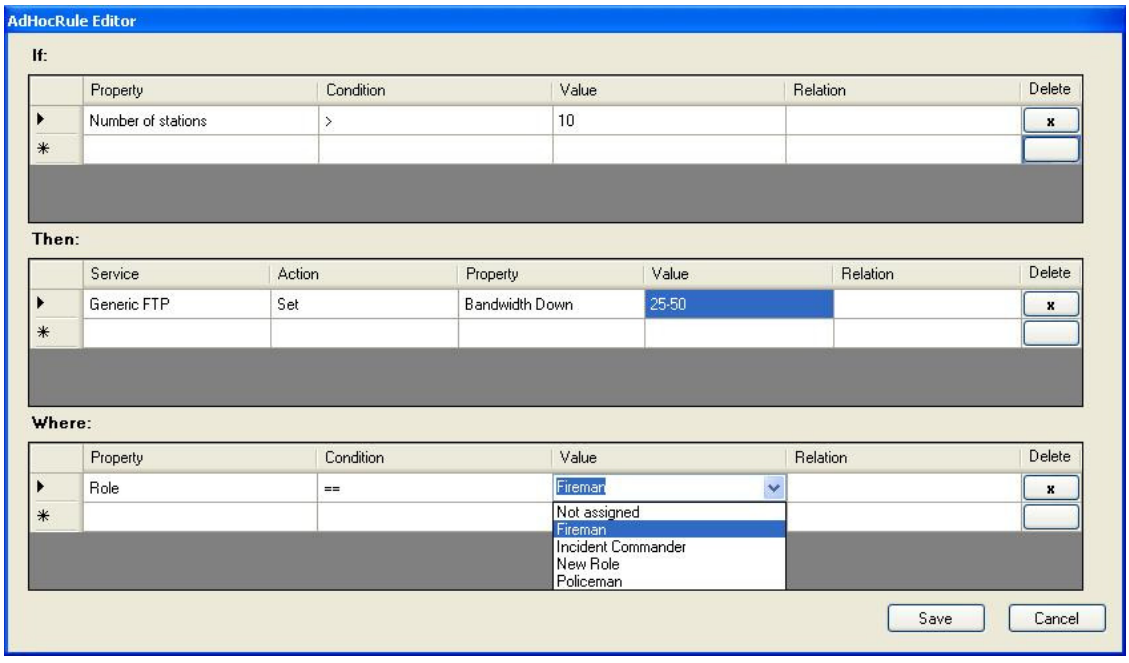
Screen Capture 66. Deleting a service from a role

Working with AdHoc Rules

Implementing AdHoc Rules is an optional way to conditionally modify SLAs. AdHoc Rules are discussed in the section starting on page 76. In summary, AdHoc Rules allow an administrator to set-up certain "trigger conditions" that cause AirSync to manipulate QoS settings on the fly based on network events such as topology changes and signal changes. To add an AdHoc Rule to a role, click on the Rules sub pane item for the selected role and click on the **Add new rule** action button, as shown in Screen Capture 67.

Screen Capture 67. Adding an AdHoc rule to a Role

AirSync will then present its AdHoc Rule editor, which allows administrators to generate rules governing QOS behavior based on certain trigger events. The rule editor is shown in Screen Capture 68. It facilitates the creation of sophisticated, condition-based QoS rules without requiring administrators to memorize any special language. Instead, administrators point, click and select (or furnish) values to GUI objects that encapsulate the syntax details from the administrator.



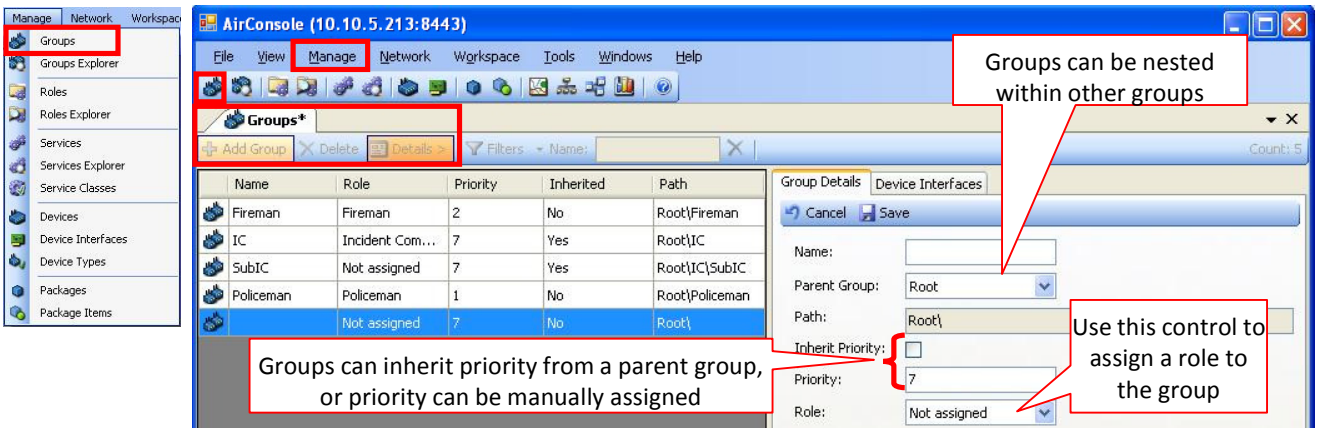
Screen Capture 68. AirSync's AdHoc rule editor

Screen Capture 69 shows the **Rules** sub pane for the new role after adding an AdHoc rule to it. Multiple AdHoc rules can be created for a role, if desired. Adding, editing and deleting them are straightforward operations.

Screen Capture 69. The "Rules" sub pane after adding an AdHoc rule to the role "New Role"

Working with Groups

To add, edit or delete a group, open the **Groups** item from the **Manage** menu or click it from the Tool ribbon. Clicking the **Add Group** action button creates a new group record and opens the **Group Details** pane for editing the new record as shown in Screen Capture 70. An example of adding a new group follows. Fill in the name field for the new group.



Screen Capture 70. Adding a new group

Groups can be nested within other groups

Groups can be arranged in a hierarchical fashion, if desired. To do so, select a value for the "parent Group" attribute from its associated drop-down list box as shown in Screen Capture 70 and with more detail in Screen Capture 71. To remove the hierarchical relationship, select the **Root** value from the list. The **Root** value is a special value representing the top level of the group hierarchy and cannot be deleted from the list. The **Path** attribute is a display-only attribute that shows the hierarchical relationship. Groups can inherit a value for the **Priority** attribute from the parent group, but cannot inherit a role from the parent group.

Screen Capture 71. Groups can be nested within groups

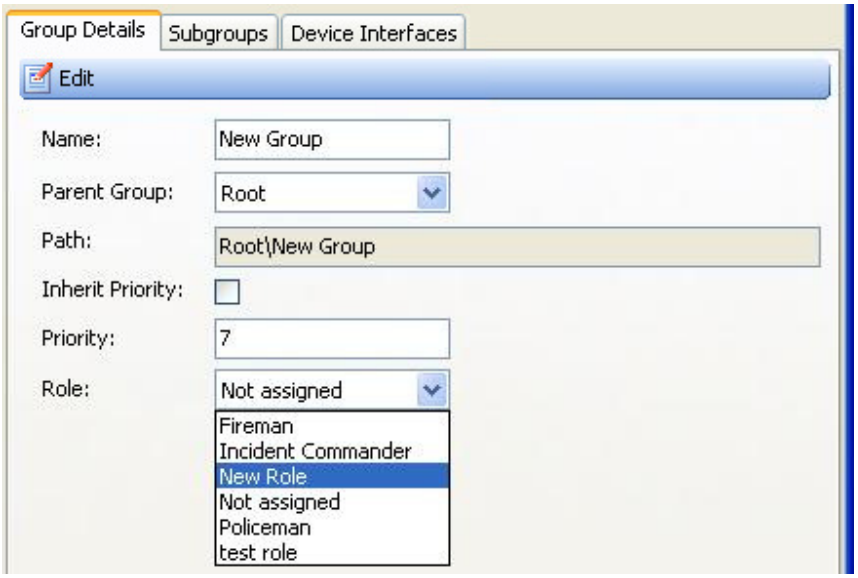
Assigning a Priority to a Group

You can choose to have a group inherit a priority value from its parent group by selecting the **Inherit Priority** attribute check box. Alternatively, you can leave this box unchecked and directly assign a priority value. See Screen Capture 70 and also Screen Capture 72.

Assigning a Role to a Group

To assign a role to a group, select the desired value from the drop down list box for the **Role** attribute as shown in Screen Capture 70 and with more detail in Screen Capture 72. Alternatively, you can drag and drop a role value onto a defined group object in the **Groups Explorer** window area, if open. If the **Roles Explorer** window item is open, you can achieve the same thing by dragging and dropping a group value onto a role in the **Roles Explorer** window. The section titled "Drag 'n' Drop" Operations with the Explorer Windows" on page 22 shows examples of "drag 'n' drop" operations.

To clear the role, select the **Not assigned** value. This is a special value and cannot be deleted. Unlike the **Priority** attribute value which can be inherited from a parent group, the **Role** attribute value must be explicitly assigned to a group. The default value is **Not assigned**.



Screen Capture 72. Assigning priority and role attribute values to a group

Working with Devices and Device Interfaces

The section titled "Registering Devices in the AirSync System" starting on page 40 discusses the basic details for adding devices and device interfaces. From a QoS perspective, the key point is to set the **Max Bandwidth kbps**, **Priority**, and **Group** attribute values appropriately for the device interface. Set these attribute values on the **Device Interface Details** tab as shown in Screen Capture 73.

Assigning the Group/Role Device Interface Attribute Value

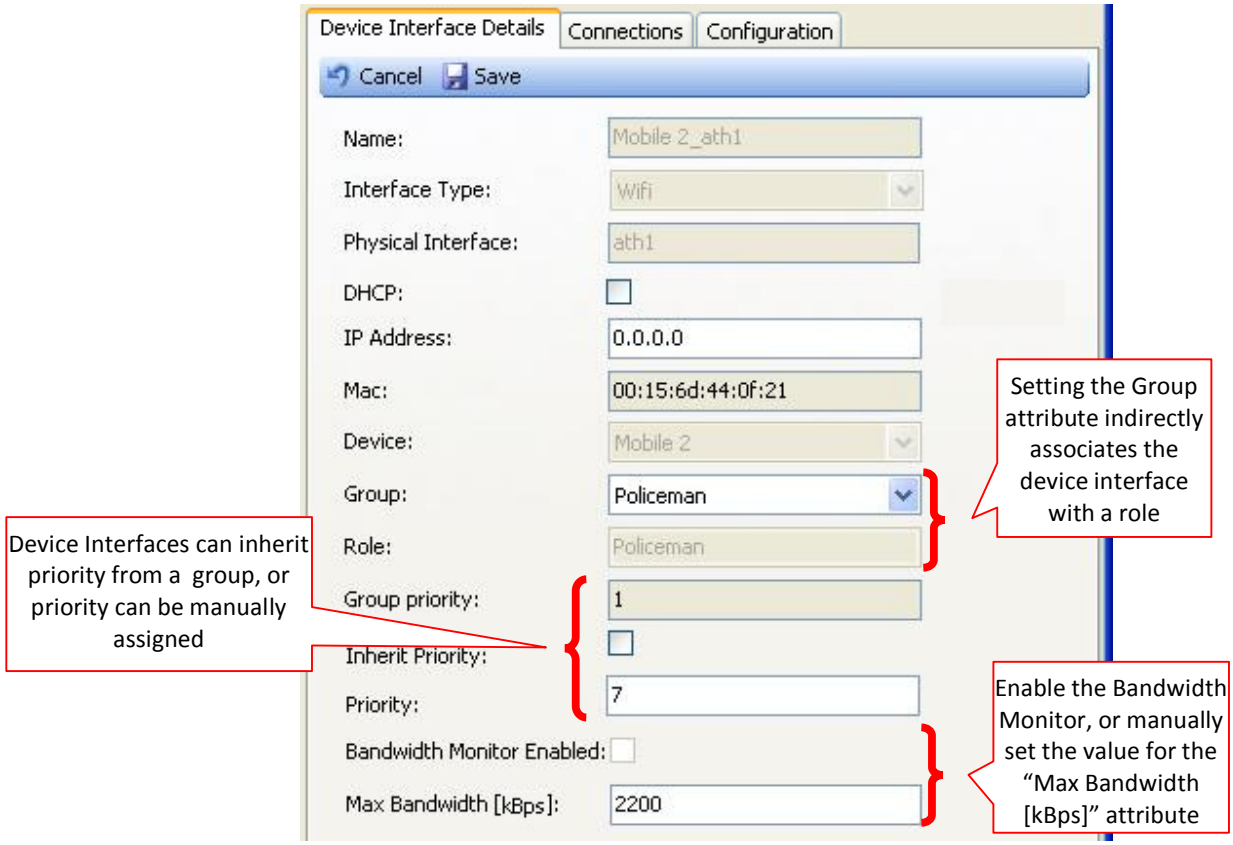
Remember from the discussion beginning on page 60 that assigning a value to the **Group** attribute of a device interface indirectly associates a role with it, too. On the **Device Interface Details** sub pane, the **Role** attribute is a read-only display indicator of the role, if any, that has been assigned to the Group item selected. To modify the role value associated with the group in question, navigate instead to the **Groups** item list, select the group in question and modify the **Role** attribute value from the **Group Details** sub pane.

Assigning the "Priority" Device Interface Attribute Value

Device Interfaces can inherit the priority attribute from an assigned group, or priority can be manually assigned. The **Group Priority** attribute is a read-only indicator of the value assigned to the priority attribute of the group that this device interface is a member. To modify this value, navigate instead to the **Groups** item list, select the group in question and modify the priority attribute value from the **Group Details** sub pane.

Assigning the "Max Bandwidth" Device Interface Attribute Value

With respect to setting the **Max Bandwidth kbps** attribute value, you can either select the **Bandwidth Monitor Enabled** attribute value checkbox to have AirSync automatically estimate the link bandwidth for you, or clear this check box and statically set the value for the **Max Bandwidth kbps** attribute. The section titled "The AirSync Bandwidth Allocation Process" beginning on page 68 discusses the semantics of these items.



Screen Capture 73. Setting the Group, Priority, and Max Bandwidth attributes for a device interface

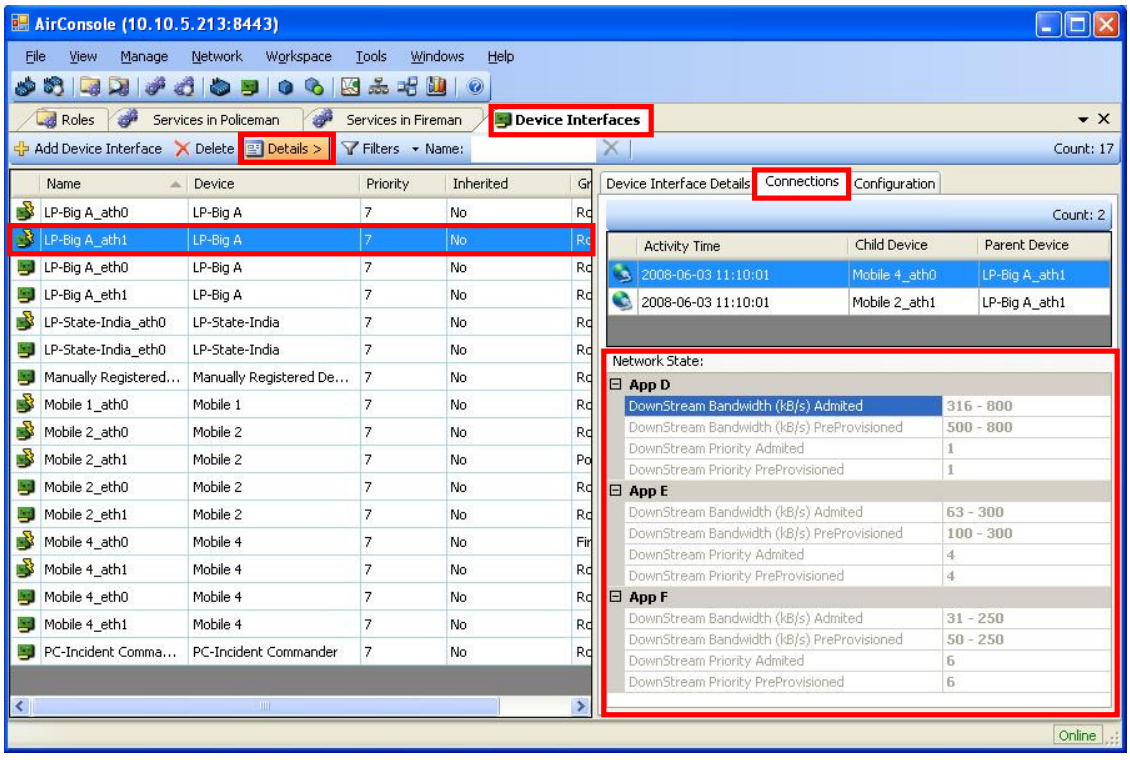
Monitoring the Results

There are three basic methods for verifying the results of an AirSync QOS implementation:

- Inspect the **Network State** item on the **Device Interface Details**, **Connections** sub pane for the device(s) in question.
- Show statistics such as number of bytes transmitted and received for the device interface(s) in question.
- Establish a remote access session with the device(s) in question and then use device platform specific tools to verify that the QoS rules have propagated all the way down to the device(s). A discussion of platform specific device tools is outside the scope of this document.

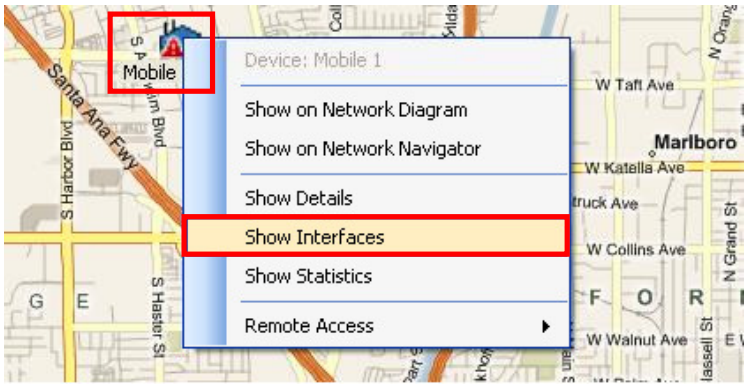
Inspecting the "Network State" for a device interface

To inspect the network state for a device interface, navigate to the Device Interface Details screen for the device(s) in question. There are several ways to navigate to the Device Interface Details screen. Screen Capture 74 shows the navigation path from the main **Device Interfaces** item list.



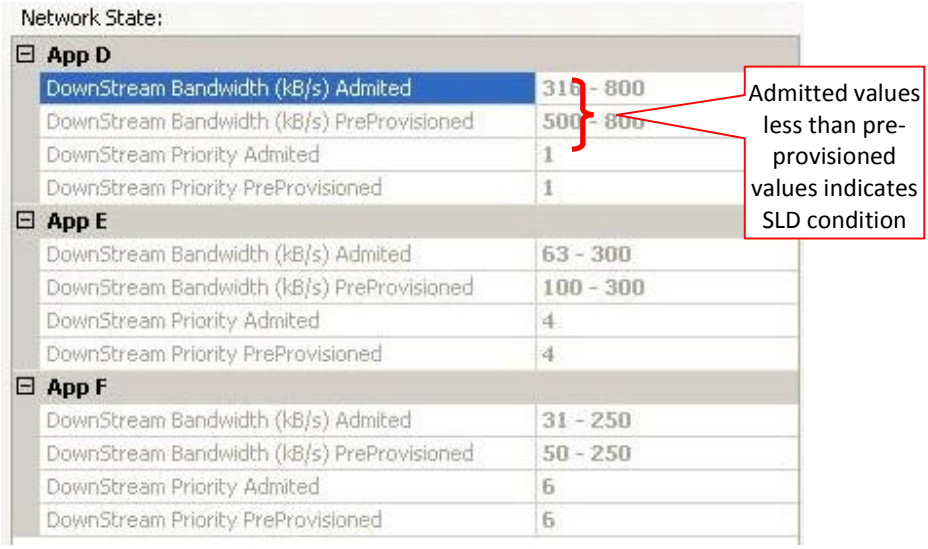
Screen Capture 74. Navigating to the Connections sub pane for a device interface

Screen Capture 75 shows how to navigate to the Device Interface Details connections sub pane by right clicking a device on the map or network diagram window and selecting **Show Interfaces** from the context-sensitive menu.



Screen Capture 75. Navigating to device interfaces from a context-sensitive menu

Screen Capture 76 shows a close-up view of the **Network State** information for a particular device interface. This display shows the queueing structures (QoS Instructions) that AirSync has generated and sent to agents for implementation on the managed device. When this display shows Admited values less than the Pre-Provisioned values, the system is in an SLD condition.



Screen Capture 76. Close-up of "Network State" for a device interface

Charting Statistics

Screen Capture 77 shows a chart of statistical information for a device interface. By viewing the aggregate I/O rates in and out of the interface (upper graph), you can get an idea of how well the QoS instructions sent to the device implement the organizational usage policy. Large dips or increases may indicate a change in topology that results in the application of a new set of QoS instructions on the device interface. These changes may indicate a new role or the application of AdHoc Rules due to dynamically occurring network conditions or events.

It can be useful to chart statistics related to signal quality such as SNR, RSSI and Rate (lower graph) to correlate with the application of AdHoc Rules. Note that the rate value is not an indication of actual link throughput capacity, so much as an indication of the current modulation scheme (QAM, QPSK, BPSK, etc.) used on the link. The modulation scheme determines the theoretical maximum bandwidth available on the link. Further, the framing efficiency of different modulation schemes varies inversely with the amount of forward error correction (FEC) used by the modulation scheme.

Screen Capture 77. Charting statistical throughput on a device interface

Remote Access

For radio device platforms that allow telnet (or similar) access to system administrators, it can be helpful to start a remote-access session to verify that the device properly received and implemented the QoS instructions from the AirSync server. The commands to verify the implementation status vary by vendor platform and are beyond the scope of this document. However, AirSync does provide the ability to establish remote access connections to devices.

Using AirSync's Package Management System

AirSync has a package management system that allows system administrators to systematically define and distribute items to managed nodes.

Theoretical Building Blocks

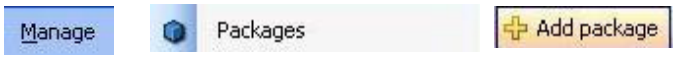
A package is a container containing content that is addressed for delivery to a specific set of devices and instructions for how to process the content after it is received by the targeted device(s). The package can contain more than one item if desired. Currently, the package distribution system is primarily designed to deliver firmware upgrades and configuration files to managed radios, but it could easily be expanded to other uses. Proximetry has a related product, GateSync, which is a more sophisticated package management system.

The basic steps for package management are:

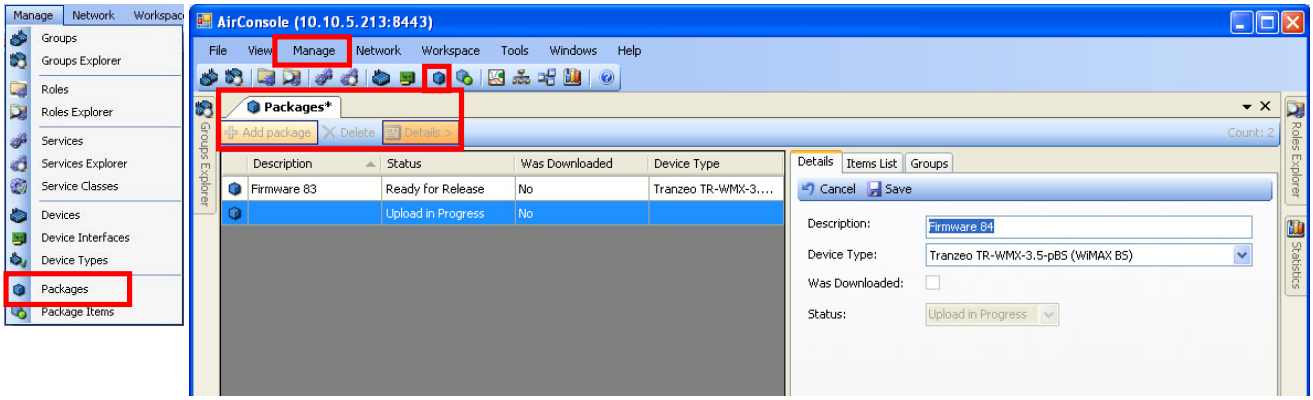
1. Define a new package.
2. Upload one or more package items for storage on the AirSync server.
3. Assign appropriate device interfaces to a group used for package management.

   Note, because configuration files are device-specific, you should have only one device interface assigned to any group used for distributing configuration files. For configuration files, create one group for each device.
4. Assign the package to the appropriate group(s).
5. Change the status of the package from **Staged** to **Ready for Release**.

Working with Packages



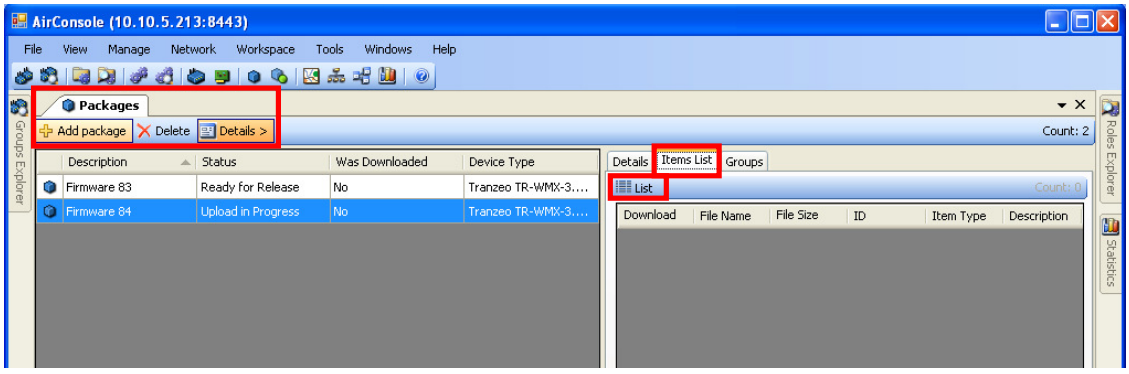
To add, edit or delete a package, open the **Packages** item from the **Manage** menu or click it from the tool ribbon. Clicking the **Add package** action button creates a new package record and opens the **Package Details** pane for editing the new record as shown in Screen Capture 78.



Screen Capture 78. Adding a new package

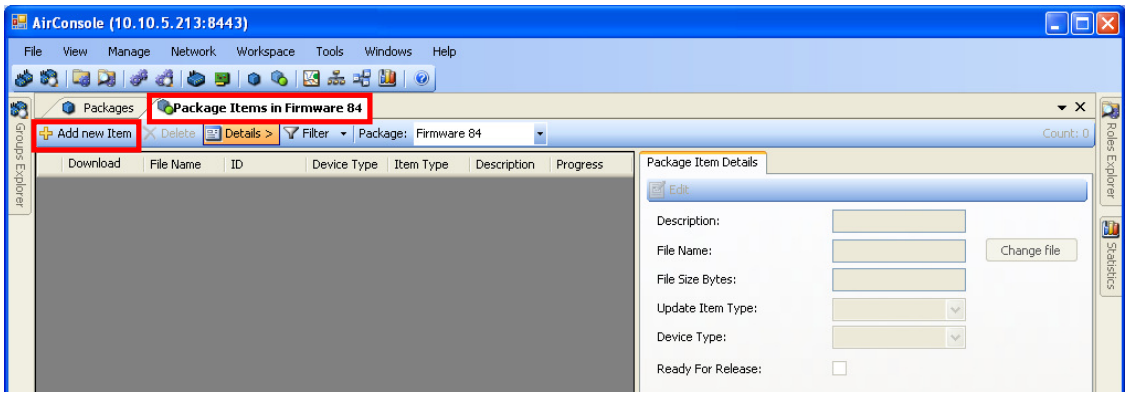
Working with Package Items

After defining the basic package you can add one or more items to the package. To work with **Package Items** you must navigate to the **Package Items** list. The quickest way to do this, however is to click the **List** action button from the **Items List** sub pane as shown in Screen Capture 79.



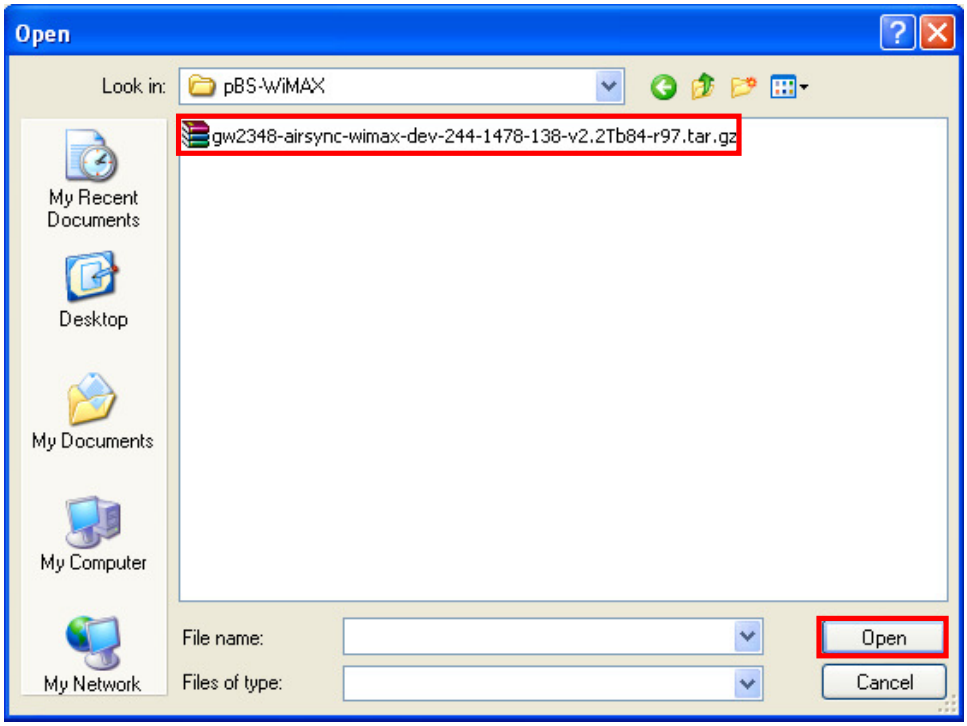
Screen Capture 79. Click the "List" Action button in the "Items List" sub pane to add items

Clicking the **List** action button will open a new **Package Items in** … tab. Click the **Add new Item** action button as shown in Screen Capture 80 to begin adding items to the package.



Screen Capture 80. Next click the "Add new Item" action button in the "Package Items in …" pane

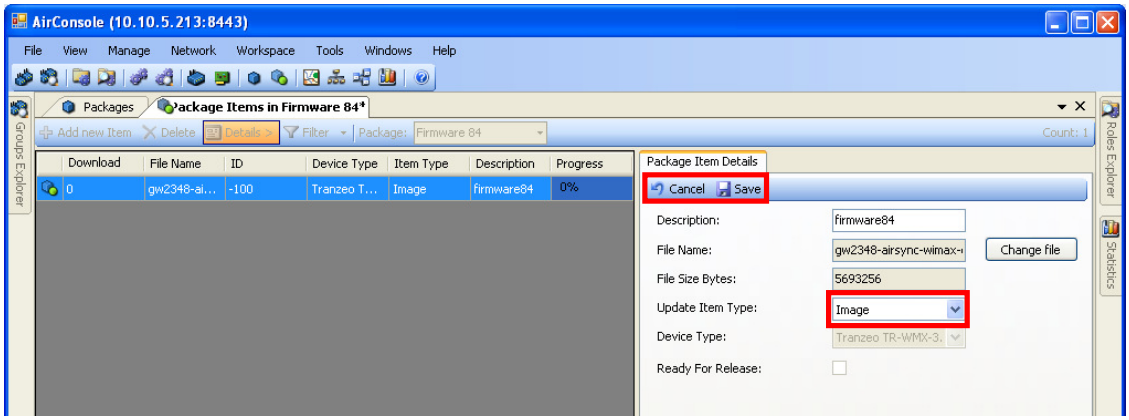
Each time you click the **Add new Item** action button, a windows explorer browser window opens allowing you to find and select the source file for your package item as shown in Screen Capture 81.



Screen Capture 81. Finding and Selecting the source file for a package item in windows

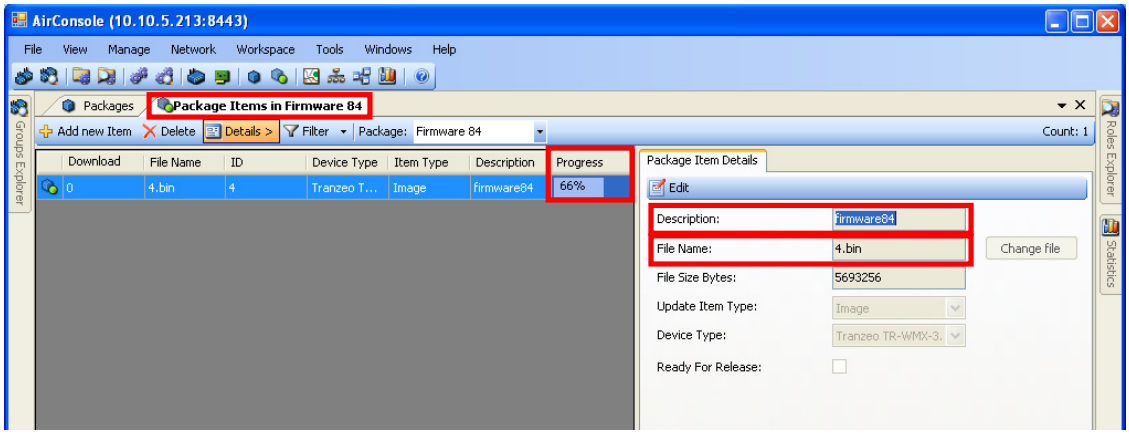
After browsing to find and select a file to include as a package item, Select the value for the **Update Item Type** attribute as shown in Screen Capture 82. Select **Image** for firmware updates and **Image Delta** for configuration file updates. Selecting the **Update Item Type** generates an appropriate set of instructions that tells the device how to process the package item after it has been received.

Notice: **Update Item Type** for the first package item has to be set to **Image**.



Screen Capture 82. Choosing the "Update Item Type" attribute value for the package item

Click the **Save** action button and the file will be uploaded to the AirSync Server. You can watch the progress indicator as shown in Screen Capture 83 as the file is uploaded to the AirSync server.



Screen Capture 83. Watching the Progress Indicator as the file is uploaded to the AirSync server

Where and How are the files stored?

The files are upload to the ./services/nftp/files subdirectory of the AirSync installation directory as shown in Screen Capture 84. Note the **File Name** and **File Size Bytes** attributes shown in Screen Capture 83 and also in Screen Capture 84 differs. The file size reported in the AirSync UI in is 5693256 which is smaller than the value 5693428 initially reported in Screen Capture 84.

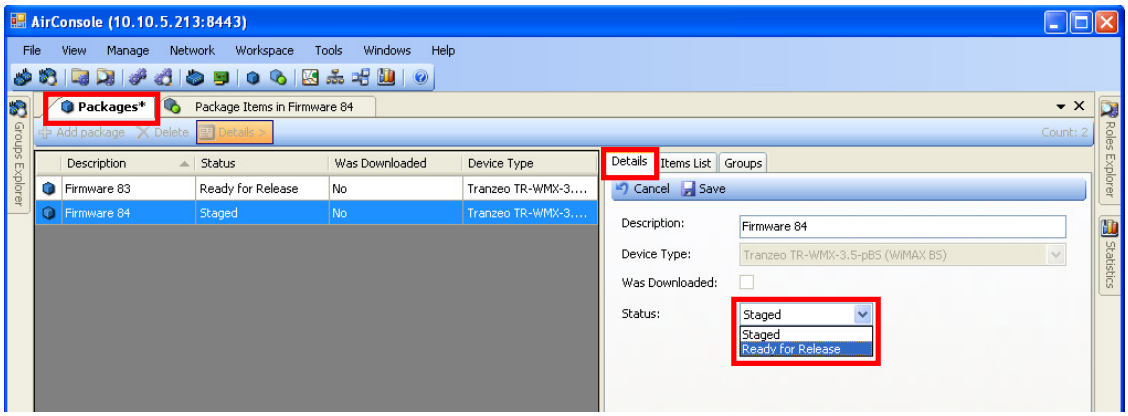
This is because NFTP encapsulates the original source file within an additional header. The smaller size value, 5693256, represents the original file size on the source computer before encapsulating the contents within an NFTP header. A subsequent inspection of the stored file, "4.bin" shows that the NFTP header has the original source file size encoded within it.

```
SD-AirSync:~# ls -l /home/airsync/services/nftp/files
total 22288
-rw-r--r-- 1 root root 5692941 2008-11-06 15:50 1.bin
-rw-r--r-- 1 root root 5693428 2008-11-06 15:51 2.bin
-rw-r--r-- 1 root root 5693428 2008-11-06 16:02 3.bin
-rw-r--r-- 1 root root 5693428 2008-11-06 16:02 4.bin
SD-AirSync:~#

SD-AirSync:/home/airsync/services/nftp/files# more 4.bin
 <FileContent><Build_id>gw2348-airsync-wimax-</Build_id><File name="gw2348-airsync-
wimax-dev-244-1478-138-v2.2Tb84-r97.tar.gz" size
="5693256"/></FileContent>
```

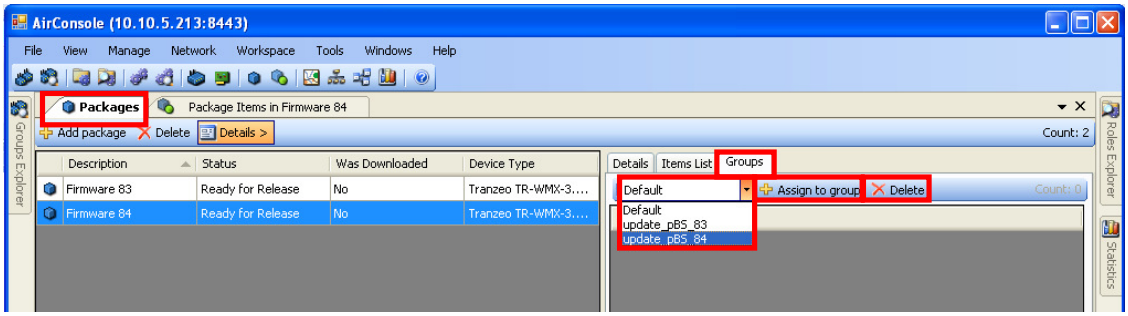
Screen Capture 84. Inspecting the package files stored on the server

Before the package can be downloaded to any devices, its status must be changed to ready for release as shown in Screen Capture 85.

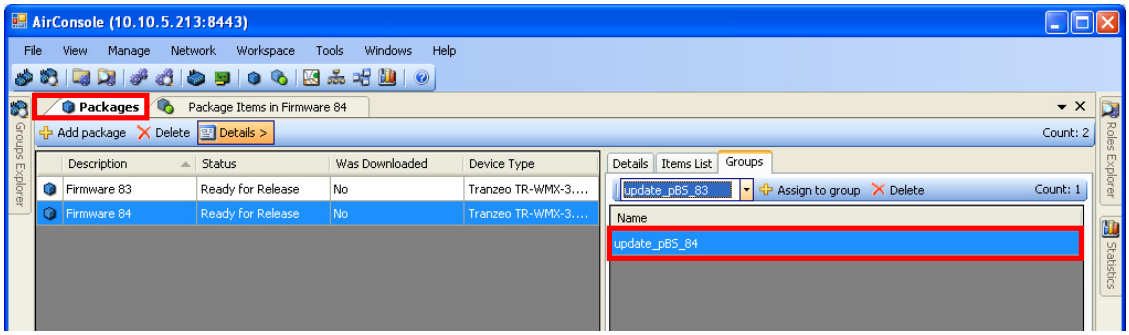


Screen Capture 85. Changing the status from "Staged" to "Ready for Release"

The next step is to assign the package to one or more groups as shown in Screen Capture 86. To do so, navigate to the **Groups** sub pane and select a target group for the package from the drop down list control. After selecting a group, click the "Assign to group" action button. You will see the list of groups the package has been assigned to as shown in Screen Capture 87.



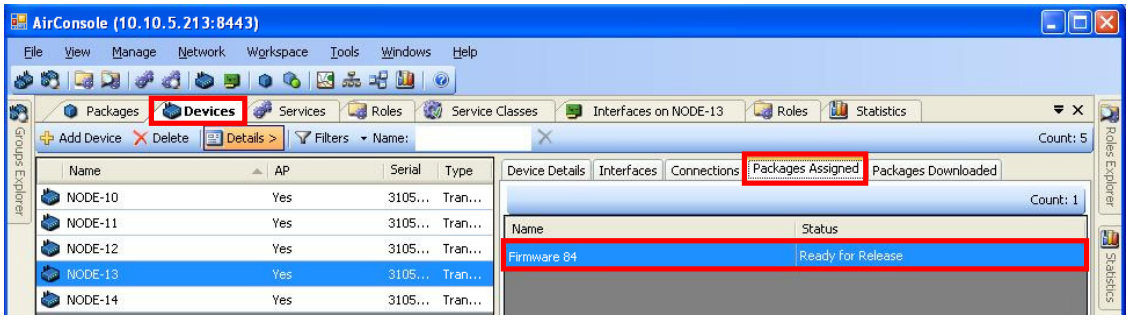
Screen Capture 86. Picking and assigning or deleting a target group for the package



Screen Capture 87. When you are done you will see the group(s) the package has been assigned to

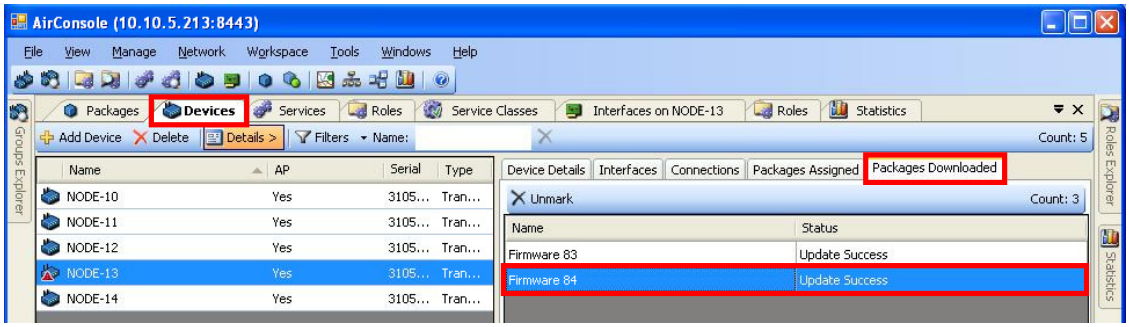
In order for the package to be processed by a device, one of the device's interfaces must be a member of one of the groups that has been assigned the package. Assuming this has been done, you can monitor the package management status by navigating to the **Devices** item list.

The **Packages Assigned** sub pane as shown in Screen Capture 88 will show all the packages currently assigned to a given device. In order to show up as assigned, the package must be assigned to an appropriate group or set of groups and the device must have one of its interfaces as a member of one of the groups for which the package has been targeted. To clear the package from the device, either remove the device's interface(s) from the targeted group(s) for the package, or remove the package assignment from the group.



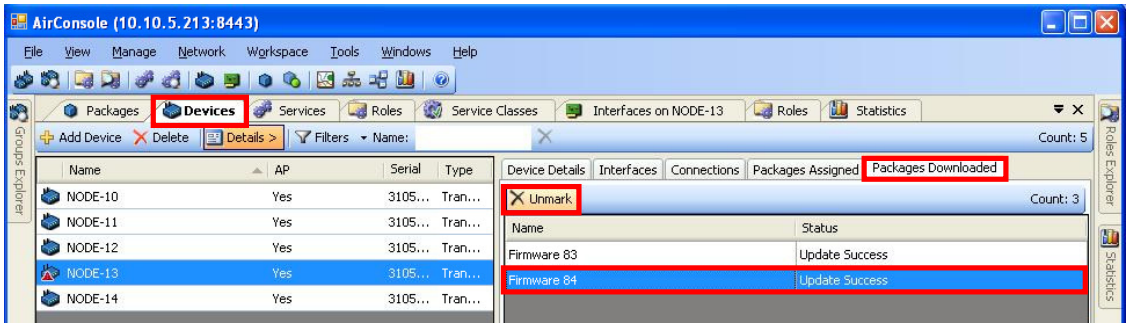
Screen Capture 88. Checking the assignment status on the "Packages Assigned" Device sub pane

Navigate to the **Packages Downloaded** sub pane as shown in Screen Capture 89. You will see a list of the packages that have been downloaded and processed by the device in question. Items will show up in this list, even if they are not currently assigned to the device.



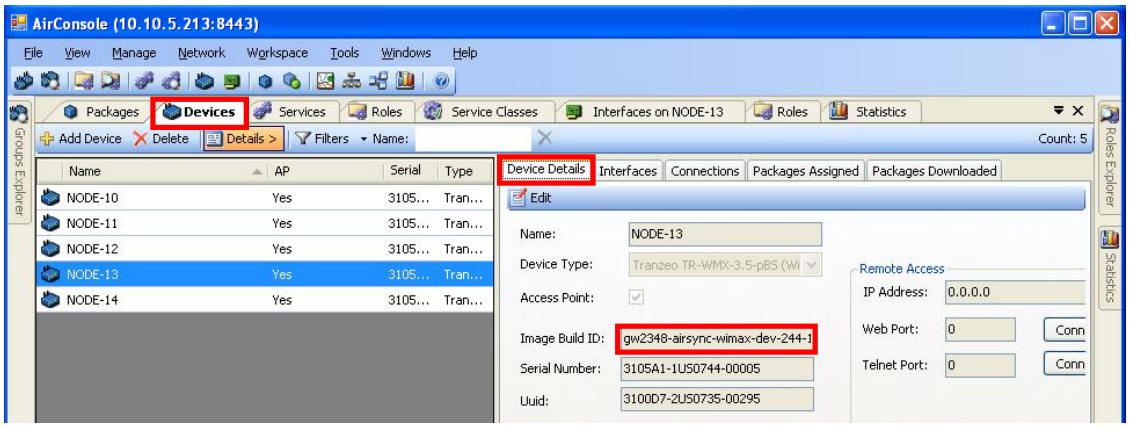
Screen Capture 89. Checking download status on the "Packages Downloaded" Device sub pane

If you want to re-process a package that is assigned to a device and has previously been processed, select the package in question from the **Packages Downloaded** item list for the device in question and click the **Unmark** action button as shown in Screen Capture 90. The selected package will disappear from the list. If it is still assigned to the device, it will be re-processed. This provides a convenient mechanism for rolling firmware back to previous versions.



Screen Capture 90. Using the "Unmark" action button to reapply the package update

After a firmware or a configuration update package is received and unpacked successfully by the device, it will reboot to finish the process. Notice that AirSync will automatically adjust the value of the device's **Image Build ID** attribute as shown in Screen Capture 91. If you recall from a previous session, the **Image Build ID** is a read-only attribute, but AirSync's package management functionality keep this value appropriately updated.



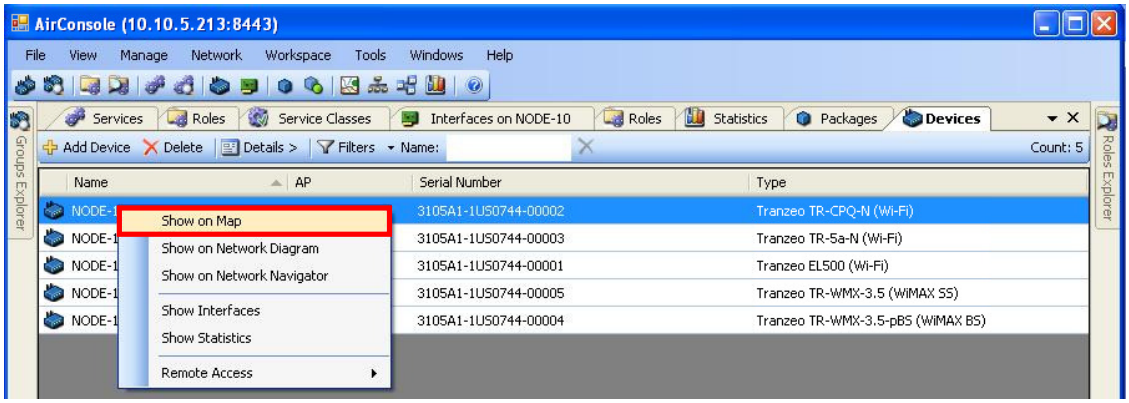
Screen Capture 91. The Image Build ID attribute value for the device changes after update

Deleting Packages

Before you can delete a package, you must remove all targeted groups from it (select the package, then the targeted group item then click the **Delete** action button from the **Groups** sub pane as shown in Screen Capture 86).

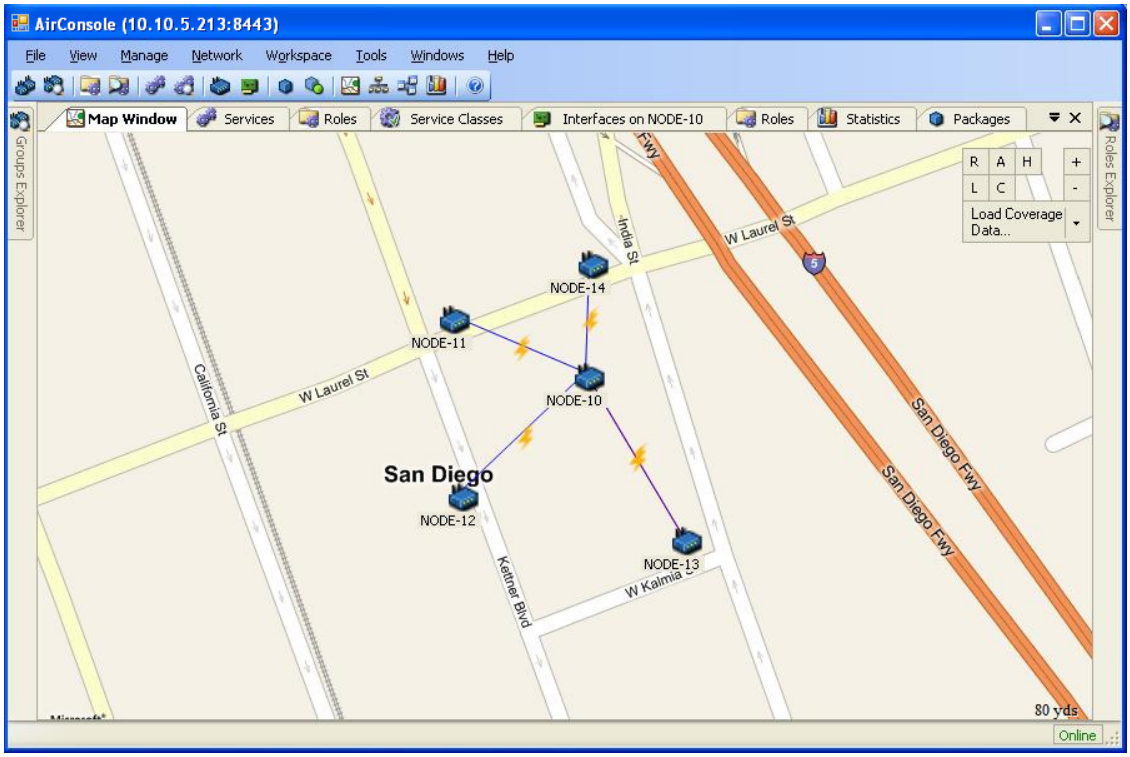
Using AirSync to Monitor the Network

AirSync provides a variety of network monitoring and mapping functions. Perhaps the simplest way to invoke them is to right click on a device and select a monitoring function from the context-sensitive menu. Screen Capture 92 shows a context-sensitive menu for displaying a device on a map.



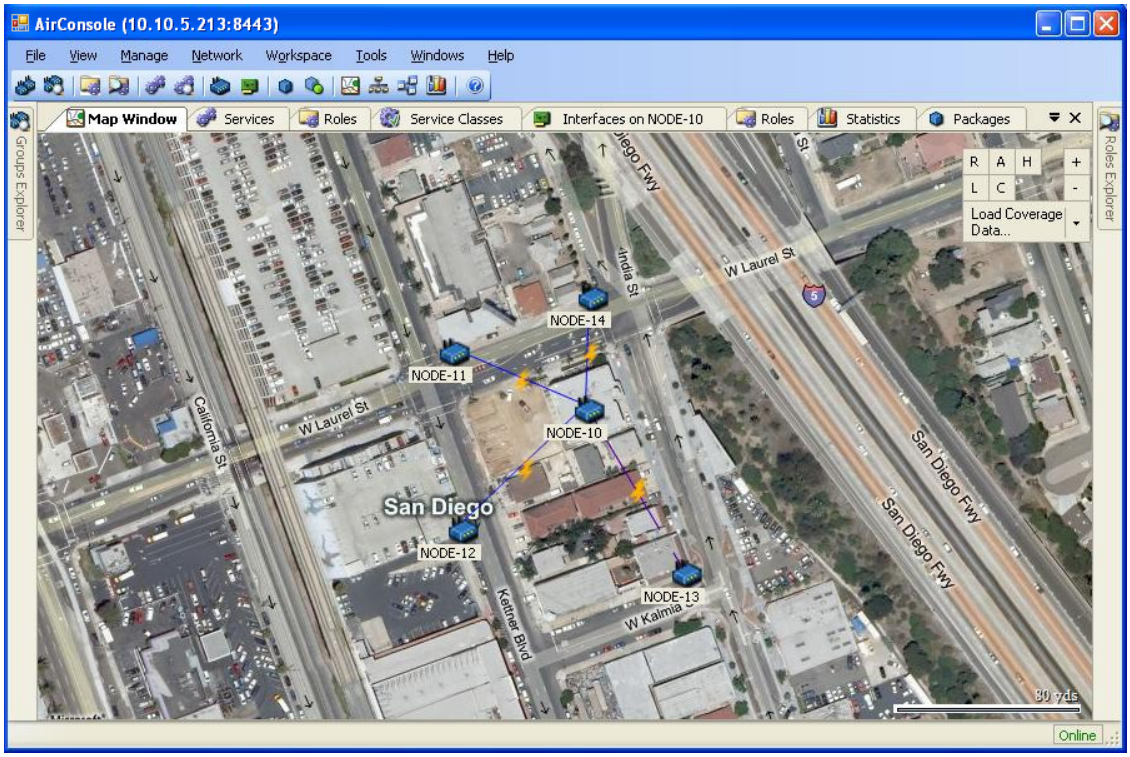
Screen Capture 92. Using a context sensitive menu to open the Map Window

Screen Capture 93 shows the **Map Window** open in streets mode. Notice the controls in the upper right hand part of the window. You can change the map mode to display an aerial or satellite view (click the A) or a Hybrid view showing both Street names and the satellite view. You can zoom in and zoom out with the "+" and "-" buttons. You can also hide or display links and connections by toggling the "L" and the "C".



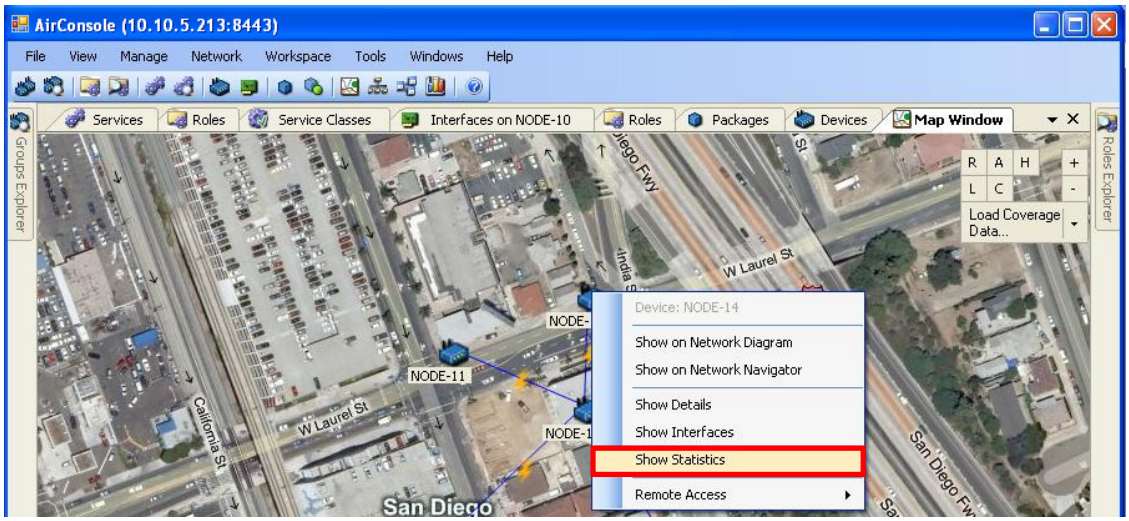
Screen Capture 93. The Map window in Streets mode

Screen Capture 94 shows the map window after it has been switched into hybrid mode.



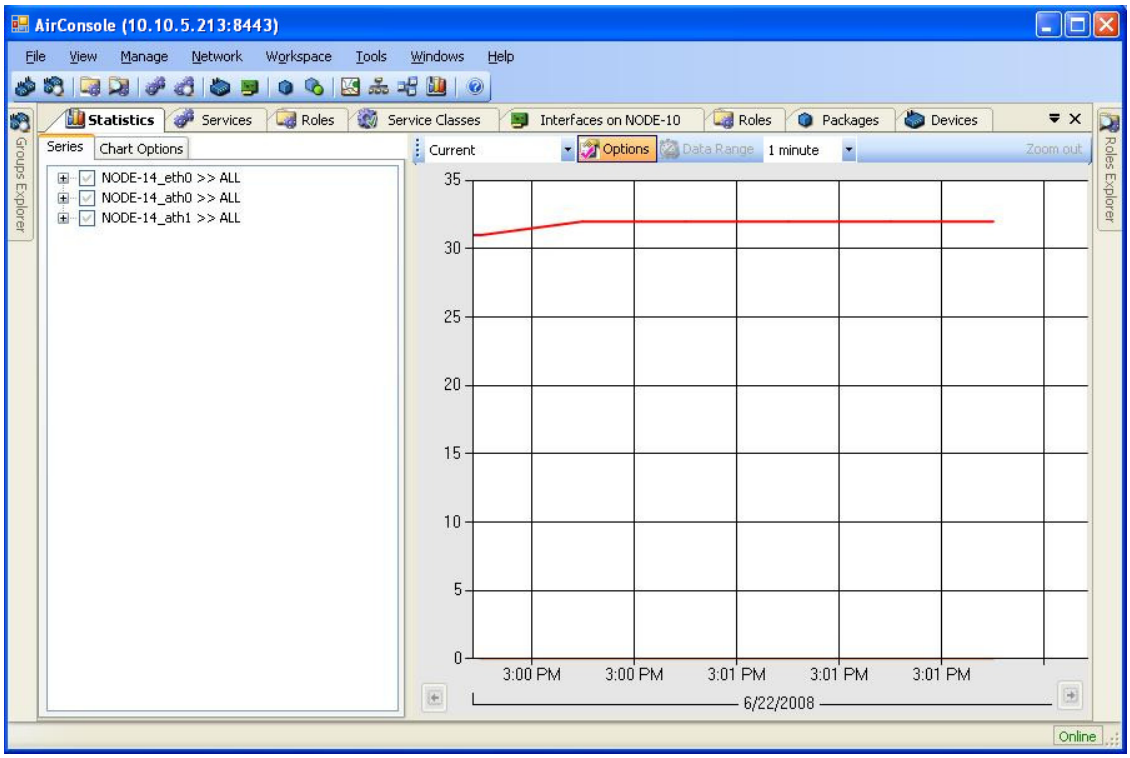
Screen Capture 94. Switching the Map window into Hybrid mode

Notice that you can right click on links, connections and nodes to bring up context sensitive menus from the **Map** window as shown in Screen Capture 95.



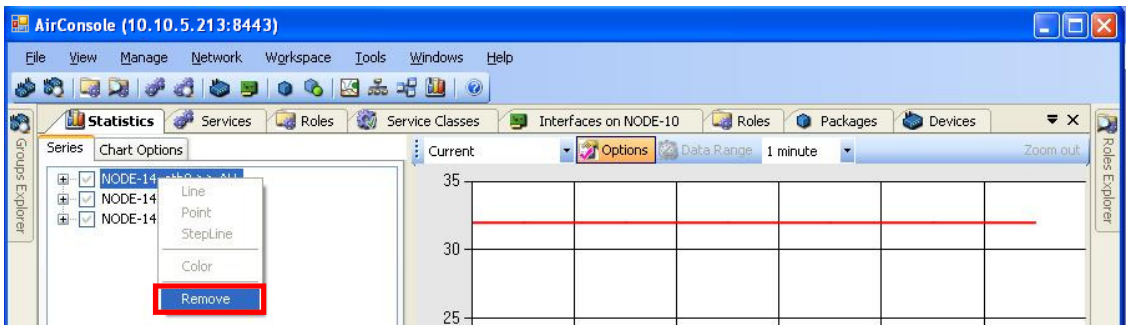
Screen Capture 95. Using a context menu to open the Statistics menu for an item

shows the statistics window. You can also drag and drop devices or device interfaces into this window to chart them.



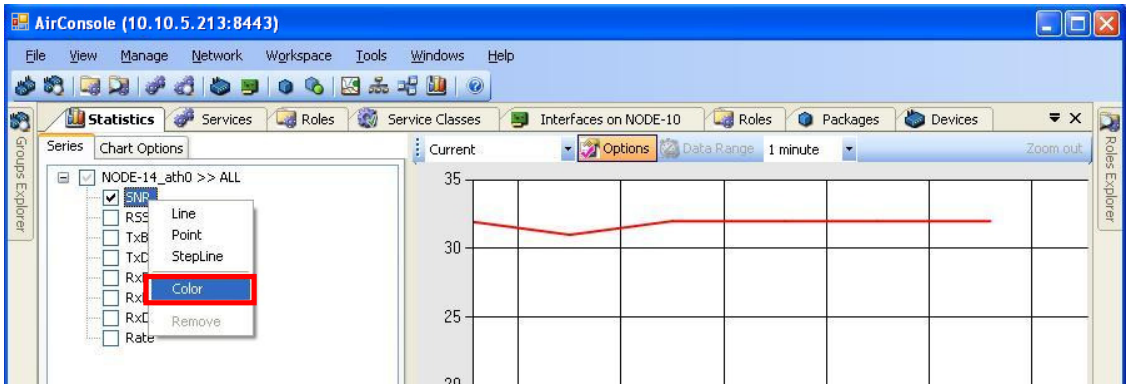
Screen Capture 96. The statistics window

To remove an item, right click it and choose **Remove** from the context sensitive menu as shown in Screen Capture 97.



Screen Capture 97. Right click an item and choose "Remove" to delete it from the chart

If you expand an item in the **Series** pane, you can choose which options to chart including various items showing signal quality and throughput as shown in Screen Capture 98. You can also change the chart type and the color of the series plot.

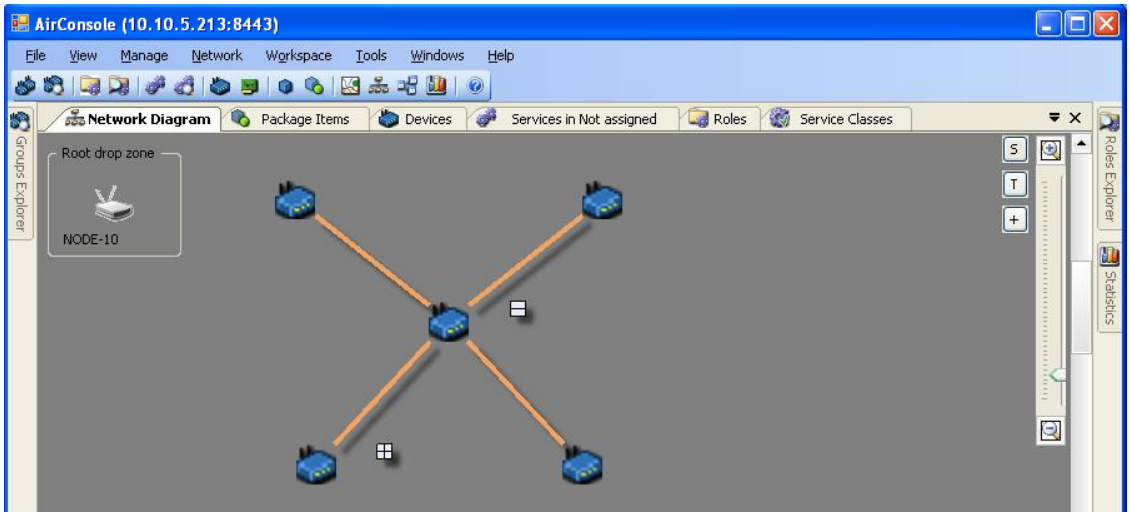


Screen Capture 98. Changing the color for a chart item from the context-sensitive menu

shows the color selection tool window.

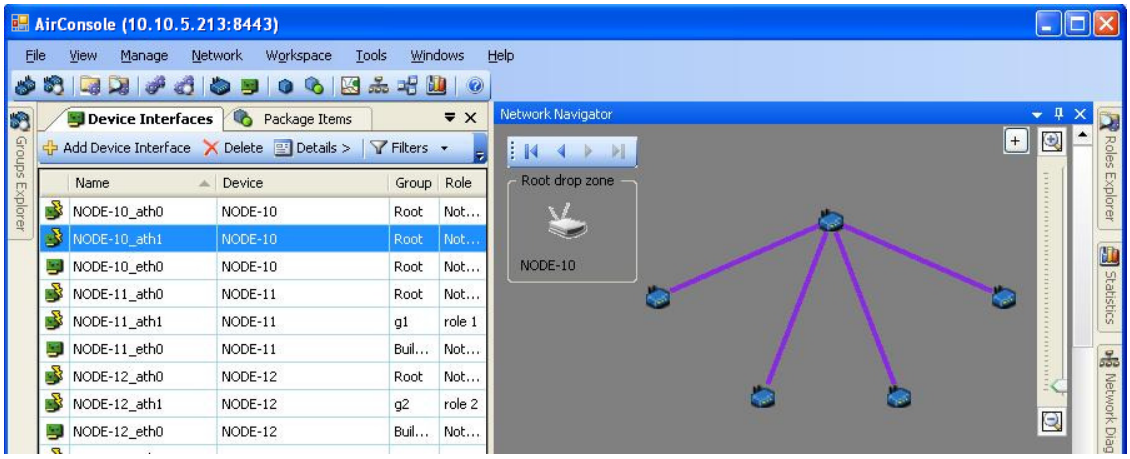
Screen Capture 99. The Color selection tool window

If you prefer to see a logical view of network topology, right click a device and choose the **Show on Network Diagram** or **Show on Network Navigator** option from the context-sensitive menu. Screen Capture 100 shows the **Network Diagram** window.



Screen Capture 100. The "Network Diagram" window

Screen Capture 101 shows the **Network Navigator** window. It is very similar to the **Network Diagram** window, but it also keeps a history of nodes visited and supports enhanced drag 'n' drop capabilities. Also notice the navigator control that lets you step through the history of nodes you have visited while exploring the network topology.



Screen Capture 101. The "Network Navigator" window

Appendix A. Item Descriptions for Tools – Options

Confirmations Tab

All values can be set to **True** or **False**

ConfirmClosingApplication

Determines whether or not the system will display a confirmation dialog box when the user attempts to close an application.

ConfirmDeleteAdHocRule

Determines whether or not the system will display a confirmation dialog box when the user attempts to delete an AdHoc Rule.

ConfirmDeleteDevice

Determines whether or not the system will display a confirmation dialog box when the user attempts to delete a device from the AirSync system database.

ConfirmDeleteDeviceInterface

Determines whether or not the system will display a confirmation dialog box when the user attempts to delete an interface from a device in the AirSync database.

ConfirmDeleteDeviceInterfaceRadioParameter

Determines whether or not the system will display a confirmation dialog box when the user attempts to delete a device interface radio parameter.

ConfirmDeleteGroup

Determines whether or not the system will display a confirmation dialog box when the user attempts to delete a group from AirSync's database.

ConfirmDeletePatternValue

Determines whether or not the system will display a confirmation dialog box when the user attempts to delete a pattern value from AirSync's database.

ConfirmDeletePendingPackage

Determines whether or not the system will display a confirmation dialog box when the user attempts to delete a pending package from AirSync's database queue.

ConfirmDeleteRole

Determines whether or not the system will display a confirmation dialog box when the user attempts to delete a role from AirSync's database.

ConfirmDeleteService

Determines whether or not the system will display a confirmation dialog box when the user attempts to delete a service from AirSync's database.

ConfirmDeleteServiceClass

Determines whether or not the system will display a confirmation dialog box when the user attempts to delete a service class from AirSync's database.

ConfirmDeleteServiceClassPatternValue

Determines whether or not the system will display a confirmation dialog box when the user attempts to delete a Service Class Pattern Value from AirSync's database.

ConfirmDeleteServiceParameter

Determines whether or not the system will display a confirmation dialog box when the user attempts to delete a service parameter from AirSync's database.

ConfirmDeleteUpdatePackage

Determines whether or not the system will display a confirmation dialog box when the user attempts to delete a package update from AirSync's database.

ConfirmGroupMoveOnGroupsTree

Determines whether or not the system will display a confirmation dialog box when the user attempts to move a group on the Groups Tree.

ConfirmRemoveServiceFromRole

Determines whether or not the system will display a confirmation dialog box when the user attempts to a service from a role in AirSync's database.

Remote Access Tab

These settings allow users to control the way AirSync establishes remote access connections to managed devices, for instance by right-clicking on a device and selecting "Remote Access" from the context-sensitive menu. All values are strings

SSH_DefaultPort

Default value: 22

Meaning: Specify the port number to use for establishing SSH connections. This should normally be set to the default value, 22, but should be changed if managed devices expect SSH connections on a different port

SSH_Path

Default value:

Meaning: Specify the full path to the third-party executable program to be invoked for establishing remote access SSH connections to a managed device.

Telnet_DefaultPort

Default value: 23

Meaning: Specify the port number to use for establishing telnet connections. This should normally be set to the default value, 23, but should be changed if managed devices expect telnet connections on a different port

Telnet_Path

Default value: C:\Windows\System32\telnet.exe

Meaning: Specify the full path to the third-party executable program to be invoked for establishing remote access telnet connections to a managed device.

Web_DefaultPort

Default value: 80

Meaning: Specify the port number to use for establishing web (HTTP) connections. This should normally be set to the default value, 80, but should be changed if managed devices expect web (HTTP) connections on a different port

Web_Path

Default value: C:\Program Files\Internet Explorer\iexplore.exe

Meaning: Specify the full path to the third-party executable program to be invoked for establishing remote access telnet connections to a managed device.

Refresh Times Tab

These values determine the responsiveness of the user interface to a variety of events that can cause display information to change. All values are in seconds. Setting the values lower increases responsiveness at the expense of greater polling/CPU overhead.

Config_Items_Refresh_Time

Default value: 40

Meaning: How often configuration settings are refreshed from the server.

Device_Interface_Connections_Refresh_Time

Default value: 5

Meaning: How often Connection information is refreshed from the server.

Device_Interface_Types_Refresh_Time

Default value: 600

Meaning: How often the interface type dictionary list is refreshed from the server.

Device_Interfaces_Refresh_Time

Default value: 5

Meaning: How often the interface list is refreshed from the server.

Device_Station_Network_State_Refresh_Time

Default value: 30

Meaning: How often the network state information(below connection child list for both devices and interfaces) is refreshed from the server.

Devices_Refresh_Time

Default value: 15

Meaning: How often the device list is refreshed from the server.

Groups_Refresh_Time

Default value: 15

Meaning: How often the group list is refreshed from the server.

Pattern_Values_In_Service_Class_Refresh_Time

Default value: 16

Meaning: How often the pattern values child list for service classes is refreshed from the server.

Roles_Refresh_Time

Default value: 12

Meaning: How often the roles list is refreshed from the server.

Rules_In_Roles_Refresh_Time

Default value: 19

Meaning: How often the AdHoc Rule data is refreshed from the server.

Service_Classes_Refresh_Time

Default value: 60

Meaning: How often the service class list is refreshed from the server.

Service_Parameter_Types_Refresh_Time

Default value: 60

Meaning: How often the service parameter type dictionary list is refreshed from the server.

Service_Parameters_Refresh_Time

Default value: 16

Meaning: How often the service parameter list is refreshed from the server.

Services_In_Roles_Refresh_Time

Default value: 19

Meaning: How often the services child list for roles is refreshed from the server.

Services_Refresh_Time

Default value: 20

Meaning: How often the services list is refreshed from the server.

Update_Items_Refresh_Time

Default value: 18

Meaning: How often the update item list is refreshed from the server.

Update_Packages_Refresh_Time

Default value: 20

Meaning: How often the update package list is refreshed from the server.

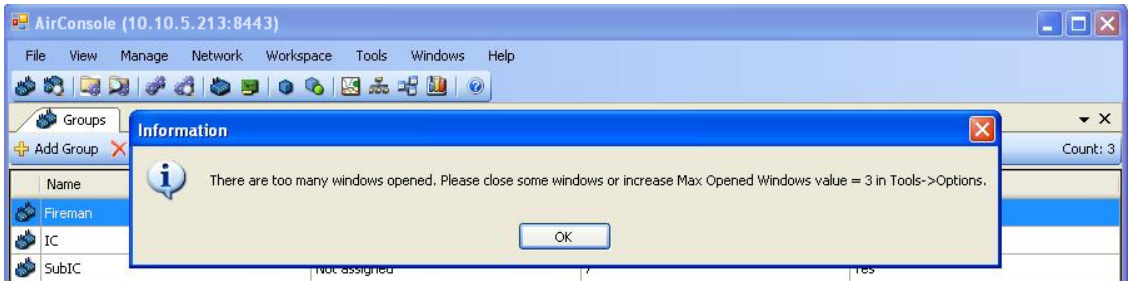
Windows Count Tab

MaxOpenedWindows

Default value: 30

Meaning: Controls how many user interface windows (excluding explorer windows) can be open at one time. If a user tries to open more windows than allowed by this value, the system will present a message (see Screen Capture 102) telling the user to close one or more windows before attempting to open another one. Setting this value to a lower value may help conserve system resources on the management workstation platform such as memory and CPU cycles.

The main application window counts as two open windows and each tabbed or floating item also counts as an open window, so setting this value to three would allow the user to keep one GUI item opened at a time. This parameter does not apply to explorer windows



Screen Capture 102. Informational message when MaxOpenedWoindows value is exceeded

MaxOpenExplorerWindows

Default value: 5

Meaning: Controls how many user interface explorer windows can be open at one time. If a user tries to open more explorer windows than allowed by this value, the system will present a message telling the user to close one or more explorer windows before attempting to open another one. Setting this value to a lower value may help conserve system resources on the management workstation platform such as memory and CPU cycles.

Chart Window Tab

ValuesToAverageForCounterStat

Default value: 5 samples

Meaning: The AirSync system keeps statistical information for charting. For each statistical item, AirSync charts values based on a "rolling average." This setting controls the granularity of the rolling average. For example, at the default value of 5 samples every point plotted represents an average value computed over a sliding window of the last 5 values received. Setting the value to one would turn off the averaging function and chart each individual sample received.

Appendix B. Item Descriptions for Tools – System Configuration

General Configuration Tab

UPLOAD_SERVER

The UPLOAD_SERVER configuration item specifies connectivity parameters for transferring files to the AirSync server. For instance, AirSync's Package Distribution functionality uses this facility for staging packages on the AirSync Server for subsequent redistribution to managed client devices. The Package Distribution feature will be discussed in more detail elsewhere in this document.

Specifically, this configuration item should be set to an IP address followed by a ":" and a TCP port number:

<IP Address>:<TCP Port>

The IP address and port number specify an endpoint where an NFTP server process is configured to listen for requests. The product installation process sets the initial value for the UPLOAD_SERVER configuration item based on the IP address furnished to the installation script and a standard default port value.

This parameter will not require modification for most installations, but it may be appropriate to modify it, for example to distribute or offload server components to multiple machines, or if the default port (6667) is not available on the host machine.

DOWNLOAD_SERVER

Indicates, in <IP Address>:<TCP Port> format, the address of the server from which package updates may be accessed and downloaded.

SOFTWARE UPDATE FAIL TIMEOUT

Time (in seconds) that is designated as the maximum allowed time span for a requested software update to complete.

Resource Manager Configuration Tab

SLD_PRIO_MAXBW_TAB

This parameter governs AirSync's bandwidth allocation algorithm during periods of contention. AirSync invokes its Service Level Degradation (SLD) algorithms to arbitrate bandwidth allocation between competing traffic flows when there is not enough bandwidth to satisfy all requests. This is discussed in greater detail elsewhere in the document.

You should generally never need to adjust this parameter value and doing so could lead to unexpected results.

BW_CHANGE_THRESHOLD

This parameter also governs AirSync's bandwidth allocation algorithms.

You should generally never need to adjust this parameter value and doing so could lead to unexpected results.

DATAGRAM_TIMEOUT

This parameter defines timeout for UDP packet waiting in the queue, after this timeout queue is being flushed.

You should generally never need to adjust this parameter value and doing so could lead to unexpected results.

SLD_LIMITING_MS_INTERVAL

This parameter defines interval between invocations of Rules Enforcement algorithm.

For WiFi Rules Enforcement algorithm includes SLD algorithm, thus its processing time might be increased.

Activation Server Configuration Tab

The Activation Server runs on the AirSync server. Its primary function is to automatically detect manageable devices and register them with the AirSync system. The activation server sends XML-based multicast messages informing client devices about the parameters they should use to register themselves with an AirSync server. The most important parameters to check and set are the **Host**, **Port**, and **RMServer** parameters.

CompanyId

This parameter is used internally.

You should generally never need to adjust this parameter value and doing so could lead to unexpected results.

GroupId

This parameter is used internally.

You should generally never need to adjust this parameter value and doing so could lead to unexpected results.

Host

Set this value to the IP Address of the AirSync (JBoss) server.

Login

This parameter stores the user name with which the AirSync (JBoss) server will authenticate to the internal database server component, by default a MySQL database server. Paired together with the associated password (Pass), **this value pair must match a valid username/password credential pair in the database installation**.

If you adjust SQL credential(s) from the defaults to enhance security, you should modify this value accordingly.

Pass

This parameter stores the password with which the AirSync (JBoss) server will authenticate to the internal database server component, by default a MySQL database server. Paired together with the associated username (Login), **this value pair must match a valid username/password credential pair in the database installation**.

If you adjust SQL credential(s) from the defaults to enhance security, you should modify this value accordingly.

UpdateTimeSpan

This parameter value should be configured via *Management Consol. It is* interval between checking for updates (in case when server is available), configured in *connection.ini, line 4*. Default value 600.

Port

This parameter value should match the TCP port number on which the AirSync (JBoss) server listens, port 8080 by default.

You should generally never need to adjust this parameter value and doing so could lead to unexpected results, unless you tailor the JBoss server component to listen on a different port, in which case you should set this to the matching port value.

RegisterTimeSpan

This parameter is configured via *Management Console. It is* interval between starting registration process. It is configured in *connection.ini, line 3*.Default value 300.

RMServer

AirSync server process that communicates with AirSync software agents on managed client devices.

StatisticTimeSpan

This parameter is configured via *Management Console*. It is interval between sending statistics. Statistics are also used as a device heartbeat. It is configured in *connection.ini, line 5* or by passing command line argument *–si*. Default value 10.

Appendix C. AirSync Preinstallation Requirements

This document describes AirSync preinstallation requirements related to network configuration for AirSync ability to operate. Figure 29 shows diagram of general AirSync usage.

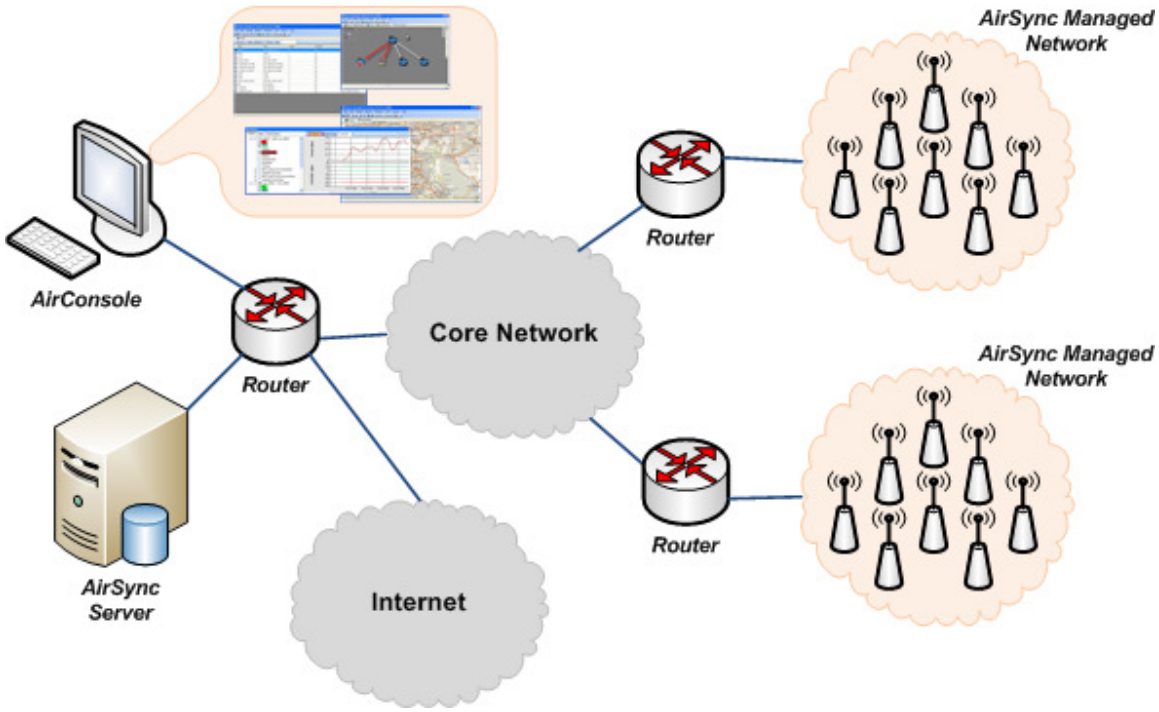


Figure 29: General AirSync usage diagram

Requirements Related to communication between AirSync Server and Managed Networks

All routers/firewalls operating between AirSync Server and AirSync Managed Networks must be configured to pass through following network traffic:

- multicast communication between AirSync enabled network devices and AirSync

Server,

- udp communication on 5000 port
- udp communication on 6666 port
- tcp communication on 6668 port

tcp communication on 80 port for devices managed via AirSync HTTP Manager (or other port which is used by device's web configuration tool)

Requirements Related to communication between AirConsole and AirSync Server

All routers/firewalls operating between AirSync Server and AirConsole must be configured to pass trough following network traffic:

- tcp communication on 8443 port
- tcp communication on 6667 port

All routers/firewalls operating between AirConsole and Internet must be configured to pass trough following network traffic:

- tcp communication on 80 port

Requirements Related to Network Time Synchronization

AirSync managed devices and AirSync Server machine must have synchronized time in UTC (e.g. via NTP service).

Appendix D - Example AirSync configuration for Wireless ISP scenario

Following chapter describes simple AirSync configuration for Wireless ISP scenario. The scenario assumes that network operator provides for his customers three services: VoIP, Internet Access with various service levels and VLAN dedicated connection. Figure 1 shows general network diagram for this scenario.

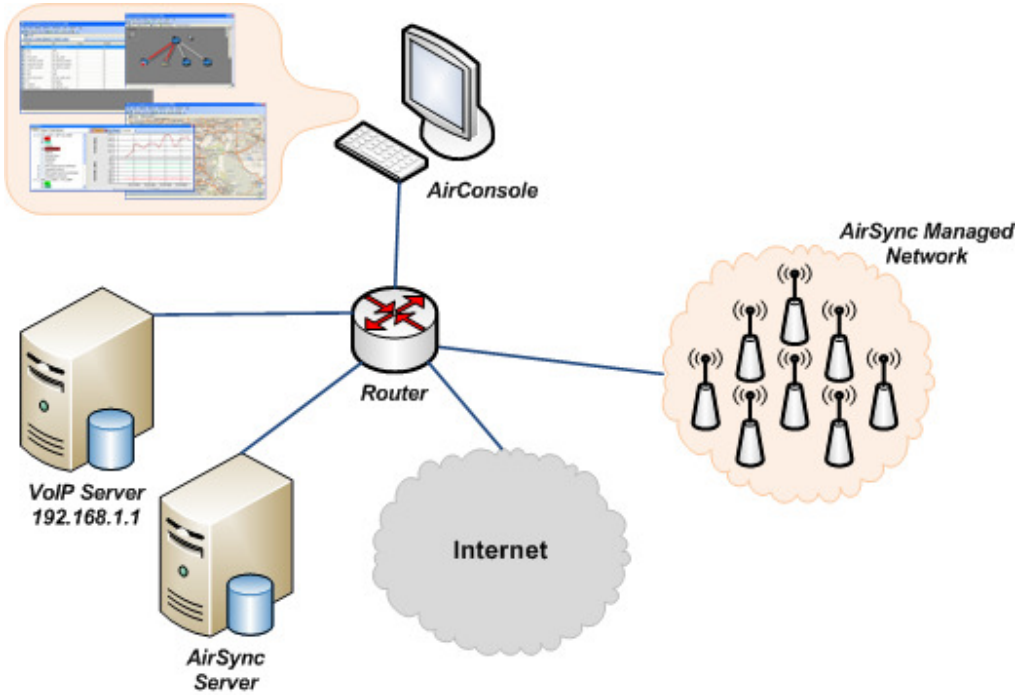


Figure 30. General network diagram for Wireless ISP scenario

Wireless ISP service description

1. VoIP

Voice over IP service runs on server with IP address 192.168.1.1 which uses UDP 5060 port for signaling protocol and UDP ports 10001 – 20000 for RTP streams.

Service parameters: downstream and upstream bandwidth 320 – 400 kbps, latency up to 40 ms, priority 1.

2. Internet Access

Internet Access service provides connectivity to all IP addresses over ICMP and TCP protocols and to all IP addresses over UDP protocol on all ports except 5060 & 10001-20000.

Internet Access service has two levels:

- Basic – upstream bandwidth up to 128kbps, downstream bandwidth up to 256kbps, priority 6,
- Premium – upstream bandwidth up to 512kbps, downstream bandwidth up to 1024kbps, priority 6.

3. VLAN dedicated connection

VLAN dedicated connection service provides VLAN based connectivity. In this example VLAN ID 50 is used. Service parameters: downstream and upstream 2048kbps, priority 2.

AirSync Service Classes configuration

Following tables shows AirSync Service Classes configuration (see 'Working with Service Classes' chapter for details about creating Service Classes).

Name	WISP – VoIP Service Class
Min Up Bw	20
Min Down Bw	20
Patterns	Pattern type: transproto:dscp:ip/net:port_start-port_end Pattern value: udp:any:192.168.1.1/255.255.255.255:10001-20000 Pattern type: transproto:dscp:ip/net:port_start-port_end

	Pattern value: udp:any:192.168.1.1/255.255.255.255:5060-5060

Table 9. WISP – VoIP Service Class configuration

Name	WISP – Internet Access Service Class
Min Up Bw	0
Min Down Bw	0
Patterns	Pattern type: transproto:dscp:ip/net:port_start-port_end Pattern value: icmp:any:0.0.0.0/0.0.0.0:0-65535 Pattern type: transproto:dscp:ip/net:port_start-port_end Pattern value: udp:any:0.0.0.0/0.0.0.0:0-5059 Pattern type: transproto:dscp:ip/net:port_start-port_end Pattern value: udp:any:0.0.0.0/0.0.0.0:5061-10000 Pattern type: transproto:dscp:ip/net:port_start-port_end Pattern value: udp:any:0.0.0.0/0.0.0.0:20001-65535 Pattern type:

	transproto:dscp:ip/net:port_start-port_end Pattern value: tcp:any:0.0.0.0/0.0.0.0:0-65535

Table 10. WISP – Internet Access Service Class configuration

Name	WISP – VLAN_50 Service Class
Min Up Bw	0
Min Down Bw	0
Patterns	Pattern type: vlan Pattern value: 50

Table 11. WISP – VLAN_50 Service Class configuration

AirSync Services configuration

Following tables shows AirSync Services configuration (see 'Working with Services' chapter for details about creating Services).

Name	WISP – VoIP Service
Description	Voice over IP Network Traffic
Service Class	WISP – VoIP Service Class
Parameters	Bandwidth DownStream: 40-50 Bandwidth UpStream: 40-50 Priority: 1 Latency: 40

Table 12. WISP – VoIP Service configuration

Name	WISP – Internet Access Service - 128/256kbps

Description	Internet Traffic
Service Class	WISP - Internet Access Service Class
Parameters	Bandwidth DownStream: 0-32 Bandwidth UpStream: 0-16 Priority: 6

Table 13. WISP – Internet Access Service - 128/256kbps configuration

Name	WISP – Internet Access Service - 512/1024kbps
Description	Internet Traffic
Service Class	WISP - Internet Access Service Class
Parameters	Bandwidth DownStream: 0-32 Bandwidth UpStream: 0-16 Priority: 6

Table 14. WISP – Internet Access Service - 512/1024kbps configuration

Name	WISP - VLAN_50 Service
Description	VLAN 50 Network Traffic
Service Class	WISP – VLAN_50 Service Class
Parameters	Bandwidth DownStream: 256-256 Bandwidth UpStream: 256-256 Priority: 2

Table 15. WISP - VLAN_50 Service configuration

AirSync Roles configuration

Following tables shows AirSync Roles configuration (see 'Working with Roles' chapter for details about creating Roles).

Name	WISP – Basic Internet Access
Services	WISP – Internet Access Service - 128/256kbps

Table 16. WISP – Basic Internet Access Role configuration

Name	WISP – Premium Internet Access
Services	WISP – Internet Access Service - 512/1024kbps

Table 17. WISP – Premium Internet Access Role configuration

Name	WISP – Basic Internet Access + VoIP
Services	WISP – Internet Access Service – 512/1024kbps WISP – VoIP Service

Table 18. WISP – Basic Internet Access + VoIP Role configuration

Name	WISP – Premium Internet Access + VoIP
Services	WISP – Internet Access Service – 512/1024kbps WISP – VoIP Service

Table 19. WISP – Premium Internet Access + VoIP Role configuration

Name	WISP – VLAN_50 Network
Services	WISP - VLAN_50 Service

Table 20. WISP – VLAN_50 Network Role configuration

AirSync Groups configuration

Following tables shows AirSync Groups configuration (see 'Working with Groups' chapter for details about creating Groups).

Name	Internet Basic
Role	WISP – Basic Internet Access

Table 21. WISP – Internet Basic Group configuration

Name	Internet Premium
Role	WISP – Premium Internet Access

Table 22. WISP – Internet Premium Group configuration

Name	Internet Basic + VoIP
Role	WISP – Basic Internet Access + VoIP

Table 23. WISP – Internet Basic + VoIP Group configuration

Name	Internet Premium + VoIP
Role	WISP – Premium Internet Access + VoIP

Table 24. WISP – Internet Premium + VoIP Group configuration

Name	VLAN 50 Network
Role	WISP – VLAN_50 Network

Table 25. WISP – VLAN 50 Network configuration

Appendix E. AirSync tuning

AirSync system after installation is configured to some default parameters. As networks can vary in size (including infrastructure characteristic like number of managed devices vs unmanaged clients), services definitions and especially activity characteristics (frequency and type of events generated which have to be serviced by Server) all parameters which can be used to tune the system regardless if available from AirConsole or require changing startup scripts or modifying configuration files are described.

Parameters

Parameters used by AirSync Agent:

Parameter	Agent Process	Description	Default Value
UpdateTimeSpan (s)	DMUpdate	configured via *Management Console* interval between checking for updates (in case when server is available), configured in *connection.ini, line 4*	600
RegisterTimeSpan (s)	DMRegister	configured via *Management Console* interval between starting registration process, configured in *connection.ini, line 3*	300
StatisticTimeSpan (s)	RMAgent	configured via *Management Console* interval between sending statistics, statistics are also used as a device heartbeat, configured in *connection.ini, line 5* or by passing command line argument *–si*	10
ACK Timeout (s)	RMAgent	timeout for receiving acknowledge message from server, after this timeout RMAgent retries sending message, configured in	5

		connection.ini, line 6 or by passing command line argument *–t*	
Retry Count	RMAgent	number of retries of sending message when ACK Timeout occurs, after this counted is reached, RMAgent is being resetted, configured in *connection.ini, line 7* or by passing command line argument *–r*	5
Monitoring Interval (ms)	RMAgent	interval for polling driver for system events, configured by passing command line parameter *–mi*	5000

Parameters used by AirSync Server:

Parameter	Description	Value
Polling Interval (ms)	interval for polling RM State Tables, configured by passing command line argument *–i*	10000
ACK Timeout (ms)	timeout for receiving acknowledge message from agent, after this timeout message is being resent, configured by passing command line argument *–t*	5000
Dead Timeout (ms)	timeout for receiving acknowledge message from agent, after this timeout agent is being asked to reset itself, configured by passing command line argument *–td*	16000
Device Status Timeout (ms)	timeout for monitoring if device is alive (if sends heartbeats), after this timeout Discovery Procedure is being initiated, configured by passing command line argument *–dst*	60000
Message Buffer Size (B)	size of queue for messages received from socket, configured by passing command line argument *–b*	20000000
Full Queue Timeout (s) / DATAGRAM_TIMEOUT (s)	timeout for UDP packet waiting in the queue, after this timeout queue is being flushed, configured via	5000

	Management Console	

Business:

Parameter	Description	Value
Rules Enforcement Interval (ms) / SLD_LIMITING_MS_INTERVAL (ms)	interval between invocations of Rules Enforcement algorithm, configured via *Management Console*	1000[1]
Bandwidth Estimator Threshold (kB)	threshold value for difference between last estimated bandwidth and new estimated bandwidth, if exceed SLD is being invoked, configured via *Management Console*	100

Parameters Dependency

Server – Device communication

Communication protocol between server and device is based on UDP. In order to increase its reliability acknowledges mechanism is used. In order to assure proper communication between devices in the network and AirSync server, parameters of this mechanism has to be tuned to existing network conditions. In communication between Server and Device, only Statistics message is not acknowledged. From the AirSync Agent perspective parameters used are ACK Timeout and Retry Count, while from RMServer ACK Timeout and Dead Timeout.

Parameters ACK Timeout (on both sides of the communication) has to be tuned to existing network conditions. Of course value of this parameter on RMServer side has to be tuned to the worst node (AirSync enabled device), while on the Agent side it can be treated individually. ACK Timeout parameter takes into account time of sending and receiving UDP packet, therefore its value cannot be less than ping time. This value should be tuned in case there are succeeding restarts of AirSync Agent. If the value is too small, it can result in unneeded retried of sent packets, on the other hand if the value is too large, communication protocol becomes unreliable; packet loss is increased as well as system response time.

Parameters Retry Count (AirSync Agent) and Dead Timeout (RMServer) are used to stop the communication that is not reliable. From the Agent perspective, after ACK Timeout is being reached, last message is being resent. If number of resends exceeds Retry Count, communication is being stopped, i.e. AirSync Agent goes into fault mode of operation. If the value of this parameter is too small, Agent will most of the time be in fault mode, if it's too

[1] For WiFi Rules Enforcement algorithm includes SLD algorithm, thus its processing time might be increased

large network traffic generated by AirSync may increase. On the server side parameter Dead Timeout is analogous to Retry Count, however it works differently. If value of this parameter is exceeded, server starts to treat Agent as working in fault mode and asks it to reset itself. It is very important that value of Dead Timeout is greater than value of ACK Timeout parameter, otherwise acknowledges mechanism on server side will be disabled, e.g. if ACK Timeout = 5s and Dead Timeout = 15s not acknowledged message will be resent after 5s, but after three retries, device is asked to reset itself. In case of setting Dead Timeout smaller than ACK Timeout, after exceeding value of the first, device is already treated as dead and none message is being resent.

In case of very dynamic network, or network startup when very large number of network events is sent from managed devices to server, it may happen that server will not be able to process events with sufficient time and it'll be blocked after a while. In order to prevent from such a situation, server posses a mechanism for flushing events queue in case of processing time being too large. This mechanism together with reliability advantages of communication protocol (acknowledges and retries) allows to handle the situation presented. The parameter that controls when the queue will be flushed is called Full Queue Timeout. The parameter value says that if first packet that is supposed to be processed stays in the queue longer than Full Queue Timeout, the queue should be flushed.

Proper settings of all parameters described above are crucial for the operation of AirSync system. In case of too aggressive (small) values, system might work unstable, in case to too relaxed (large) values, system will become unreliable, and managed network could be overloaded by AirSync.

Device status/heartbeat

In current AirSync implementation role of heartbeat is filled by statistics sent from the device. Statistics sending interval is defined by the parameter StatisticTimeSpan. Because statistics are the largest messages send from agent, if this parameter is badly tuned, system might be very unstable. If statistics are sent too often, it may result in overloading AirSync server.

In case when statistics are sent too rarely system may recognize device as being down and start Device Discovery Procedure. The parameter tuned on the server side in order to prevent from such a situation is called Device Status Timeout. Such a situation results in a fact that any increase of StatisticTimeSpan parameter on the Agent should be followed by increase of Device Status Timeout on the RMServer. Also, one should take into account that as RMServer serves all devices, Device Status Timeout should be higher than the highest StatisticTimeSpan in all Agents. From the other hand, if the parameter is too large, recognition of dead devices will be slower.

Other bad results of too rarely sent statistics are delayed visualization (in comparison to reality) and slower system response on network changes (Bandwidth Estimator algorithm will be invoked rarely).

The last bad effect of too rarely sent statistics is NAT Traversal mechanism built into AirSync communication protocol. In case when on the path between AirSync Agent and AirSync Server (RMServer) NATting device exists, this mechanism is used by AirSync Server to communicate to the device. NAT Traversal mechanism is based on the assumption that each packet send from inside network establishes session on the NATting device and so response

to this packet can be sent. However, in case of TCP default session timeout is about 3 days, while for UDP (protocol used by AirSync) this value is about 30s (in Linux). Therefore, as statistics in AirSync are treated as heartbeat, if they are sent more rarely than UDP session timeout in NATting device, communication from AirSync Server to AirSync Agent might become impossible.

QoS rules propagation

Process of providing QoS rules to the devices is triggered by any of the following:

1. Association/Disassociation of client/subscriber device.
2. Satisfaction (or revocation) of AdHoc rule IF statement.
3. Bandwidth change calculated by Bandwidth Estimator (for WiFi only).
4. Provisioning plan modification by system operator (not tunable).

First and second types of events have their source in changes in network environment, therefore there tuning is performed on Agent side by the parameter Monitoring Interval. Setting this interval to small value increase faster system reaction on any of above, however increase both network traffic (multiple messages in short time slot) and server load. For example if Monitoring Interval is set to 1s and any of association/disassociation/modulation change/snr change happens each second, than each second a message from a device is being sent to server, however if in the same case Monitoring Interval is set to 5s, then only network state from fifth second is being sent. In case of oscillating networks it might be useful to set this interval to larger value in order to stabilize system reaction (do not react on small changes). Other important thing to remember is that results of monitoring are periodically sent as statistics, therefore Monitoring Interval should be less than StatisticTimeSpan. Otherwise, wrong statistics data will be propagated to the server, e.g. if Monitoring Interval is 20s, while StatisticTimeSpan is 10s, than each second statistics interval, same values of statistical parameters will be sent.

Third type of event is controlled by two parameters, namely StatisticTimeSpan (described earlier) and Bandwidth Estimator Threshold. First one was already described above. Second parameter allows tuning system reaction for small changes. As Bandwidth Estimator algorithm is very sensitive to both environment (radio) and topology changes, in case of varying network system may end up with permanent device reconfiguration, and finally in network instability. In case of small value of this parameter, above situation may occur. However, in case of too large value, system may not react on bandwidth changes of the radio links. Bandwidth Estimation is available only for WiFi devices, thus this type of event is WiFi specific.

Rules propagation mechanism is driven by events described above, but can be also tuned in the second stage of it. Two parameters used for this tuning are Rules Enforcement Interval[2] and Polling Interval. First parameter defines how often service parameters (and resulting QoS rules) are scheduled for distribution to the network. In case of WiFi, before the distribution occurs an adjusting of the parameters to existing network conditions is being performed. In case when QoS rules are not changed, they are not propagated to the devices, even when any of above events occurs. Therefore this is the next place in rules

[2] For WiFi Rules Enforcement algorithm includes SLD algorithm, thus its processing time might be increased.

propagation process that may slow down system reaction, or decrease system oscillations (and in result increase network stability) and in opposite. Second parameter, Polling Interval, is used by RMServer to poll database for any changes in QoS rules. This is the last point where slowing down of the rules propagation process can be performed. In case when this parameters is too large, system response for any network event increases.

Device registration

This process is used for two purposes, automatic registration of devices and automatic refresh of registration. In case when any of two fails, remaining AirSync Agent processes are stopped until successful registration/refresh. Parameter that controls this process is RegisterTimeSpan. This parameter defines how often registration process will be invoked. Generally registration process should be started at system startup and then only used as safety check (to verify if someone deleted device or in case of some server failure). Good practice is to set this parameter to high value, e.g. 604800 (once a week). When this value is too small, this may significantly load the server.

Device update

Update process is very crucial both from system, network and device perspective. From system side, when poll for updates from network nodes are very often, it may increase the load on the server. Similarly network also will be overloaded because of additional transfer generated by updating application. And at the end, most software updates result in device reboot, so these are crucial for network access. In order to parameterize Device update process, one can use UpdateTimeSpan parameter. This parameter defines how often device polls server for update. Taking into account how crucial the process is, value of the parameter should rather be defined in hours. Other fact to be said for it is that in real life network device updates do not happen very often, therefore system reaction on the update can be delayed. It is claimed as a good practice to try to tune value of this parameter for all network devices in such a way, that they do not poll for update in the same time. This can both decrease server and network load.

Example Configurations

Below are some example configurations for 2 kinds of network with different activity characteristics. The settings are especially valid to hardware platform like: Intel Core2Duo 2,4GHz CPU, 2GB RAM, 250GB HDD, 100Gb Ethernet with Debian GNU/Linux.

Below are described characteristic of some networks, proposed configurations and expected network startup time in case all devices try to be serviced by system at one time (e.g. after network breakdown).

Network 1:

- 10 WiMAX BS with 10 clients for every BS

- 5 bidirectional service flows defined and assigned to every client device
- Maximum planned clients' activity (like associations or modulation changing) is not more often generated than every 10 seconds by an individual client device.

Parameter	Value
UpdateTimeSpan (s)	*3600*
RegisterTimeSpan (s)	*900*
StatisticTimeSpan (s)	*10*
Device Status Timeout (ms)	*60000 (default)*
SLD_LIMITING_MS_INTERVAL (ms)	*5000*

In such a configured network startup time (excluding device startup time) is about 15 seconds.

Network 2:

- 100 WiMAX BS with 10 clients for every BS
- 5 bidirectional service flows defined and assigned to every client device

A) Maximum planned clients' activity (like associations or modulation changing) is not more often generated than every 2 minutes by an individual client device.

Parameter	Value
UpdateTimeSpan (s)	*43200 (10h)*
RegisterTimeSpan (s)	*3600 (1h)*
StatisticTimeSpan (s)	*10*
Device Status Timeout (ms)	*60000*

SLD_LIMITING_MS_INTERVAL (ms)	*30000*
Monitoring Interval (ms)	*5000*

In such a configured network startup time (excluding device startup time) is about 2 minutes.

B) Maximum planned clients' activity (like associations or modulation changing) is not more often generated than every 1 minute by an individual client device.

Parameter	**Value**
UpdateTimeSpan (s)	*43200 (10h)*
RegisterTimeSpan (s)	*3600 (1h)*
StatisticTimeSpan (s)	*30*
Device Status Timeout (ms)	*120000*
SLD_LIMITING_MS_INTERVAL (ms)	*30000*
Monitoring Interval (ms)	*5000*

In such a configured network startup time (excluding device startup time) is about 2 minutes.

Appendix F. Setting AirSync Server Logging Options

Setting AirSync's JBoss server logging options

Locating the configuration file

The main configuration file for log4j can be found under:

.\airsync_dir\servers\jboss_dir\server\default\conf\jboss-log4j.xml (AirSync 2.2 windows installation).

or

/home/airsync_dir/services/jboss_dir/server/default/conf/jboss-log4j.xml (AirSync 2.2 linux installation).

Logging behavior

By default Activation logs the messages to:

.\ airsync_dir\Servers\ jboss-4.2.2.GA\server\default\log\server.log (AirSync 2.2 windows installation)

or

/home/airsync_dir/services/jboss/server/default/log/server.log (AirSync 2.2 linux installation) with the log level set in **jboss-log4j.xml**.

By default, logging levels for all categories (i.e. source packages from which the log messages are incoming) are set to **ERROR**. This setting is supposed to fit the production environments, meaning that the log will contain only the messages with priorities of **ERROR** or higher. Log4j's logging levels hierarchy can be found at http://logging.apache.org/log4j/1.2/manual.html. For a quick reference, AirSync uses the following log levels:

DEBUG < WARN < INFO < ERROR

There are several categories in the configuration file that inherit their settings (logging levels, appenders) from the root category. They can be set to an individual log level, as well as an appender.

Log level for file can be easily changed by editing the **jboss-log4j.xml** file. Edit this line:

```
<param name="Threshold" value="ERROR"/>
```

in section *Preserve messages in a local file* for file. For example changing log level to a debug for a file:

```
<!-- ================================= -->
   <!-- Preserve messages in a local file -->
   <!-- ================================= -->
   <!-- A size based file rolling appender-->
   <appender name="FILE"
class="org.jboss.logging.appender.RollingFileAppender">
     <errorHandler class="org.jboss.logging.util.OnlyOnceErrorHandler"/>
     <param name="File" value="${jboss.server.home.dir}/log/server.log"/>
     <param name="Append" value="false"/>
     <param name="MaxFileSize" value="500MB"/>
     <param name="MaxBackupIndex" value="2"/>
    <param name="Threshold" value="DEBUG"/>
     <layout class="org.apache.log4j.PatternLayout">
       <param name="ConversionPattern" value="%d %-5p [%c] %m%n"/>
     </layout>
   </appender>
```

JBoss server.log is a cyclic buffer managed by log4j. The default settings are to keep 2 backups of **server.log** file, each sized 500MB max. That means you need at least triple more free space for those log files. It can be easily changed by editing **log4j.properties** file. Edit this two lines:

```
<param name="MaxFileSize" value="500MB"/>
<param name="MaxBackupIndex" value="2"/>
```

Remember that logging level has impact on log file size.

Setting AirSync's Activation logging options

Locating the configuration file

The main configuration file for log4j can be found under:

.\airsync_dir\Servers\Activation\log4j.properties (AirSync 2.2 windows installation)

or

/home/airsync_dir/services/activation/log4j.properties (AirSync 2.2 linux installation).

Logging behavior

By default Activation logs the messages to:

.\ airsync_dir\Servers\Activation\activation.log (AirSync 2.2 windows installation)

or

/home/airsync_dir/services/activation/ activation.log (AirSync 2.2 linux installation) with the log level set in **log4j.properties**.

Log level for file appenders can be easily changed by editing the **log4j.properties** file. Edit the first two lines:

```
log4j.logger.com.proximetry=error, R
```

The available log levels are: **debug <warn <info <error** (these are log4j's priorities). This means that if you want to discard debug and warning messages, you should set the log level to **info** (at least).

Activation log is a cyclic buffer managed by log4j. The default settings are to keep 2 backups of **activation.log** file, each sized 100MB max. That means you need at least triple more free space for those log files. It can be easily changed by editing **log4j.properties** file. Edit this two lines:

```
log4j.appender.R.MaxFileSize=100MB
log4j.appender.R.MaxBackupIndex=2
```

Remember that logging level has impact on log file size.

Setting AirSync's RMServer logging options

Locating the configuration file

The main configuration file for log4j can be found under:

.\airsync_dir\servers\RMServer\log4j.properties (AirSync 2.2 windows installation)

or

/home/airsync_dir/services/remserver/log4j.properties (AirSync 2.2 linux installation).

Logging behavior

By default RMServer logs the messages to:

.\airsync_dir\servers\RMServer\rmserver.log (AirSync 2.2 windows installation)

or

/home/airsync_dir/logs/ rmserver.log (AirSync 2.2 linux installation) with the log level set in **log4j.properties**.

Log level for file appenders can be easily changed by editing the **log4j.properties** file. Edit the first two lines:

```
log4j.logger.fileLogger=error, R
```

The available log levels are: **debug <warn <info <error** (these are log4j's priorities). This means that if you want to discard debug and warning messages, you should set the log level to **info** (at least).

RMServer log is a cyclic buffer managed by log4j. The default settings are to keep 2 backups of **rmserver.log** file, each sized 50MB max. That means you need at least triple more free space for those log files. It can be easily changed by editing **log4j.properties** file. Edit this two lines:

```
log4j.appender.R.MaxFileSize=50MB
log4j.appender.R.MaxBackupIndex=2
```

Remember that logging level has impact on log file size.

Setting AirSync's NFTP Servers logging options

Locating the configuration file

The main configuration file for log4j can be found under:

.\airsync_dir\servers\NFTP\log4j-down.properties (AirSync 2.2 windows installation)

.\airsync_dir\servers\NFTP\log4j-up.properties (AirSync 2.2 windows installation)

or

/home/airsync_dir/services/nftp/log4j-down.properties (AirSync 2.2 linux installation)

/home/airsync_dir/services/nftp/log4j-up.properties (AirSync 2.2 linux installation).

Logging behavior

By default NFTP servers logs the messages to:

.\airsync_dir\servers\NFTP\nftp-down.log and **nftp-up.log** (AirSync 2.2 windows installation)

or

/home/airsync_dir/services/nftp/nftp-down.log and **nftp-up.log** (AirSync 2.2 linux installation) with the log level set in **log4j.properties**.

Log level for file appenders can be easily changed by editing the **log4j-down.properties** and/or **log4j-up.properties** files. Edit this line:

```
log4j.rootLogger=error, stdout, R
```

The available log levels are: **debug <warn <info <error** (these are log4j's priorities). This means that if you want to discard debug and warning messages, you should set the log level to **info** (at least).

NFTP logs are cyclic buffers managed by log4j. The default settings are to keep 1 backup of **nftp-down.log** and **nftp-up.log** files, each sized 50MB max. That means you need at least twice more free space for those log files. It can be easily changed by editing **log4j.properties** file. Edit this two lines:

```
log4j.appender.R.MaxFileSize=50000KB
log4j.appender.R.MaxBackupIndex=1
```

Remember that logging level has impact on log file size.

Setting AirSync's HTTPManager logging options

Locating the configuration file

The main configuration file for log4j can be found under:

.\airsync_dir\servers\HTTPManager\log4j.properties (AirSync 2.2 windows installation)

or

/home/airsync_dir/services/httpmanager/log4j.properties (AirSync 2.2 linux installation).

Logging behavior

By default HTTPManager logs the messages to:

.\airsync_dir\servers\HTTPManager\httpmanager.log (AirSync 2.2 windows installation)

or

/home/airsync_dir/services/httpmanager/ httpmanager.log (AirSync 2.2 linux installation) with the log level set in **log4j.properties**.

Log level for file appender can be easily changed by editing the **log4j.properties** file. Edit this two lines:

```
log4j.logger.fileLogger=error, R
```

The available log levels are: **debug <warn <info <error** (these are log4j's priorities). This means that if you want to discard debug and warning messages, you should set the log level to **info** (at least).

HTTPManager log is a cyclic buffer managed by log4j. The default settings are to keep 1 backup of **httpmanager.log** file, each sized 50MB max. That means you need at least twice more free space for those log files. It can be easily changed by editing **log4j.properties** file. Edit this two lines:

```
log4j.appender.R.MaxFileSize=50MB
log4j.appender.R.MaxBackupIndex=1
```

Remember that logging level has impact on log file size.

Glossary

Activation Server

A process that runs periodically. The activation server makes sure that nodes managed by AirSync know with which AirSync server to communicate. Among other things, the activation server helps during the device registration process. It makes it possible for devices to be automatically "discovered" by the AirSync server.

AdHoc Rules

AdHoc Rules are custom rules defined by a role that influence the role. AdHoc Rules can also influence traffic of other network devices according to the device priority, throughput defined by particular services for a given device, and available network resources.

Bandwidth downstream

A service parameter that defines the interval for the throughput to the end-user device for a service. This parameter is defined as the range of two integers representing lower and upper limits, in kB/s. For example [100-250] means Bandwidth DownStream should be in the range from 100kB/s to 250 kB/s. The range you select should be sufficient for the particular service type.

Bandwidth upstream

A service parameter that defines the interval for the throughput from the end-user device for a service. This parameter is defined as the range of two integers representing lower and upper limits, in kB/s. For example [100-250] means Bandwidth UpStream should be in the range from 100kB/s to 250 kB/s. The range you select should be sufficient for the particular service type.

Device

A device is an item to be managed in AirSync. Devices are generally radios but other types of devices such as PCs can be registered and managed in AirSync. For radios, think of devices as a collection of interfaces. While traffic shaping and package distribution both work in conjunction with device interfaces, certain attributes apply to the entire device, such as the configuration file and the firmware.

Device Interface

Device interfaces are the focal point for traffic shaping, package distribution and statistics collection. To implement traffic shaping and/or package distribution, include the appropriate device interface in a group and then assign a role to the group (traffic shaping) or assign a package to the group (package distribution).

Group

A group links end-user device interfaces to a role. Groups have a priority that can either be explicitly defined for the group or inherited by a device. If inherited, the priority can influence the Service Level Degradation algorithm and affect network traffic-shaping parameters.

AirSync uses groups as containers for associating roles (for traffic shaping) and packages (for package distribution) with device interfaces.

NFTP

Network File Transfer Protocol. This is a special protocol for efficiently transferring files over wireless networks.

NIC

Network Interface Card, a PC card or expansion board inserted into a device to connect the device to a network.

Package

A package is used for delivering and installing files such as configuration files or new firmware versions on devices. Packages contain one or more items and some instructions telling the receiving device how to process the items received in the package.

Package Item

A package item is a specific file to be included in a package. Although many packages may have only one item, it is possible to define a package with multiple items for instance a firmware version and some corresponding patches.

Pattern

Patterns are an important component of *Service Classes.* Patterns are used to construct packet classifiers used for implementing traffic shaping. There are three types of patterns: Those based on MAC addresses, those based on traditional TCP/IP or UDP/IP socket connections, and those based on higher level application characteristics such as SIP

Priority

AirSync uses the term priority in two different ways. When there is not enough bandwidth available to satisfy all SLAs defined for an interface (i.e., the system is experiencing SLD), AirSync uses the priority associated with a device interface (which can be inherited from group membership or explicitly defined for the device interface) to arbitrate the bandwidth allocation compromises that must be made while the network is oversubscribed.

AirSync uses the priority associated with services to arbitrate the bandwidth allocation process when there is surplus bandwidth available, that is to say, after all the minimum SLAs for an interface have been satisfied. AirSync allocates additional "burst" bandwidth capacity (up to the maximum value in the bandwidth range specified for a service) for traffic flows based on the priority level associated with each service.

Rate

On RF links you may see a reference to rate with values such as 54 48 36 24 18 12 9 6 (depending on frequency). These numbers are really an indication of the modulation scheme being used which defines the maximum theoretical rate of traffic over the link. The rate parameter may be used in traffic shaping, for example as the basis for an ad-hoc shaping rule.

Resource Management

Resource Management lets you manage wireless network resources using AirSync. Management is performed using services, service parameters, groups, group priorities, roles, and AdHoc Rules, all of which are defined in AirSync. The Resource Manager administers Quality of Service and Throttling in wireless networks. Using AirSync Resource Management lets you attain the level of service described in Service Level Agreement.

Role

A role is used in traffic shaping. A single role can be assigned to a group which effectively associates it with all the device interfaces in the group. A role can have a set of provisioned services each representing an SLA for a given type of traffic for a given class of user.

RSSI

Received Signal Strength Indication. This is an indication of the signal quality of an RF link. For some chipsets this is really just the difference between the signal level and the noise level on the link.

Service

A service represents a provisioned SLA for a given type of traffic for a given type of traffic. Each service references a service class containing one or more patterns that match the traffic that will be provisioned according to the SLA defined by the service.

Service parameters define the upstream and downstream bandwidth and priority for a service in the system. Using AdHoc Rules and device priority, you can modify a device's service parameters, based on the results of the Service Level Degradation (SLD) algorithm (which adjusts the network based on a particular Service Level Agreement and current network load).

Service Class

Service Classes are related to services but don't carry the provisioning information that defines an SLA. Service classes reference patterns that define a set of packets that will be treated the same way from a traffic shaping perspective. It is the service, not the service class that defines the SLA. Each service references a single service class. A single service class can reference multiple patterns.

Signal Quality

Signal Quality affects the throughput available on a link and therefore affects the implementation of traffic shaping for a link.

SLA

Service Level Agreement. In AirSync you can define minimum throughput rates for services. If the system is able to deliver the minimum rate of traffic, it is meeting the SLA. At times, however, the system does not have enough bandwidth to meet the SLA.

SLD

During periods of network oversubscription when the system cannot meet the set of SLAs for an interface the system experiences Service Level Degradation, or SLD. AirSync implements an intelligent SLD algorithm that allows the system to degrade traffic in a systematic fashion based on a seven level priority scheme. The algorithm includes a tunable weighting coefficient for traffic at each priority level. The algorithm works in such a way that each priority level will get better service than worse priority levels (1 is the most preferred, 7 is the least preferred).

The algorithm ensures that the most bandwidth is allocated to the most important traffic, unlike strict priority queuing schemes, the AirSync SLD algorithm eliminates queue starvation for the lowest priority traffic. Even the lowest priority levels will get a little bit of bandwidth during times of congestion.

The Service Level Degradation algorithm considers:

- The current network load on wireless network devices.
- Service-level agreement parameters defined in the AirSync system.
- Ad-hoc rules used to adjust these parameters to current network conditions.
- Group/device priorities.
- Parameters defined in the AirSync configuration.

Based on these factors, the algorithm reduces the minimum defined bandwidth or denies access for particular users or user services until there are sufficient network resources for them to operate correctly while allowing mission-critical data to pass.

A high load on an access point may prevent the Service Level Degradation from achieving the provisioned minimum bandwidth. If this occurs, you can change the provisioned rules or add network devices to increase the available resources.

Universal Datagram Protocol (UDP)

A connectionless protocol that, like TCP, runs on top of IP networks. UDP offers a direct way to send and receive datagrams over an IP network.

VoIP

Voice over Internet Protocol, a category of hardware and software that enables the Internet as the transmission medium for telephone calls by sending voice data in packets using IP rather than by traditional circuit transmissions of the PSTN.

Index

Proximetry, Inc.

Corporate Headquarters
909 West Laurel Street, Suite 200
San Diego, CA 92101
U.S.A
Phone: 1 619 704 0020
www.proximetry.com

PROXIMETRY