
# Wavelink Avalanche MC
# Extended Device Support
# Reference Guide

amc-rg-eds-20081030

*Revised 10/30/08*

Wavelink Corporation
6985 South Union Park Avenue, Suite 335
Midvale, Utah 84047
Telephone: (801) 316-9000
Fax: (801) 316-9099
Email: customerservice@wavelink.com
Website: http:\\www.wavelink.com

Email: sales@wavelink.com

# Table of Contents

# Chapter 1: Introduction

This document provides information about using Extended Device Support.

This section provides the following information:

- Document Assumptions
- Document Conventions
- About Extended Device Support

## Document Assumptions

This document assumes that the reader has the following:

- Knowledge of wireless networks and wireless networking protocols.
- Knowledge of TCP/IP, including IP addressing, subnet masks, routing, BootP/DHCP, WINS, and DNS.
- Knowledge of Wavelink Avalanche MC.
- Knowledge of infrastructure devices. (See the documentation included with your devices for more information.)

## Document Conventions

This document uses the following typographical conventions:

| | |
|---|---|
| `Courier New` | Any time you type specific information into a text box (such as a file name), that option appears in the `Courier New` text style. This text style is also used for any keyboard commands that you might need to press.<br><br>Examples:<br><br>Type `Enter` to continue.<br><br>Press `CTRL+ALT+DELETE`. |
| **Bold** | Any time you interact with an option (such as a button or descriptions of different options in a dialog box), that option appears in the **Bold** text style.<br><br>Examples:<br><br>Click **Open** from the **File** Menu.<br><br>Select the **Update** option. |
| *Italics* | Any time this document refers to another section within the document, that section appears in the *Italics* text style. This style is also used to refer to the titles of dialog boxes.<br><br>Examples:<br><br>See *Building the Support File* on page 7 for more information.<br><br>The *Extended Device Support* dialog box appears. |

## About Extended Device Support

Avalanche MC Extended Device Support allows you to quickly and easily add a new infrastructure device or custom infrastructure device properties to your Avalanche MC installation.

Extended Device Support uses a scripting language to automate the process of adding infrastructure devices and properties to Avalanche MC. The script can be created in a standard text editor, such as Notepad, or a more advanced development environment such as Eclipse. Scripts are then stored in `.xml` file format.

---

**NOTE** Extended Device Support is included in Avalanche MC 4.8 and later versions.

---

## Overview of the Extended Device Support Process

The following steps outline the process of adding support for a new infrastructure device or custom infrastructure device property to your Avalanche MC installation:

1. **Create a support file.** Use a text editor to create a script-based device support file in `.xml` format.

   -Or-

   **Create a custom setting file.** Use a text editor to create a script-based custom setting file in `.ext.xml` format.

2. **Import the support file.** Import the support file into Avalanche MC.

3. **Import the firmware file.** You can also import the device's firmware file into Avalanche MC.

4. **Create a firmware update package.** Create a firmware update package that contains the new support and firmware files.

5. **Deploy.** Use the Task Scheduler to deploy the changes to your dServers.

## Supported Settings

Extended Device Support provides support for most infrastructure device settings, including:

- IP Address
- Subnet mask
- Gateway

- Security settings
- Radio
- LAN/WAN
- VLAN/WLAN

# Chapter 2: Creating Support Files

This chapter provides detailed information about creating Extended Device Support files. Support files contain all of the information necessary for Avalanche MC to support new infrastructure devices and custom device settings. The following information is provided:

- Creating a Device Support File
- Creating a Custom Setting File

## Creating a Device Support File

Creating a support file for a new infrastructure device involves the following steps:

- Obtaining Infrastructure Device Information
- Building the Support File

### Obtaining Infrastructure Device Information

Before you can build a device support file, you must obtain information about the settings supported by the new infrastructure device. You can view infrastructure device settings by entering the device's IP address into a web browser and logging onto the device. Use an SNMP MIB browser to obtain details about each setting, such as the OID value.

---

**NOTE** The MG-SOFT MIB Browser can be used to retrieve infrastructure device properties. Refer to the MIB Browser's documentation for more information.

---

### Building the Support File

After you have obtained information about the infrastructure device settings, you can create a script-based support file containing the device settings that will be managed. Avalanche MC will read the support file to recognize, display, query, and configure the settings.

**To create a support file:**

1 Open Notepad or another text editor.

2 Create a new file starting with `as_` and ending in `.xml`. The name of the file should correspond to the hardware and firmware of the infrastructure device.

For example: `as_Symbol4131_0395-04.xml` for a Symbol 4131 with version 3.95-04 firmware.

3 Paste the following text into the `.xml` file:

```
<?xml version="1.0" encoding="ISO-8859-1"
standalone="no"?>
<ScriptingProperties>
  <Hardware>SymbolAP4131</Hardware>
  <FirmwareVersion>03.95-04</FirmwareVersion>
  <FirmwareRangeOfSupport>Version Equal To=3.95-
  04</FirmwareRangeOfSupport/>
  <DeviceRole>AccessPoint</DeviceRole>
  <ManufacturerName></ManufacturerName>
  <ModelName>AP-4131</ModelName>
  <FirmwareFileName>dsap_fw39504.bin
  </FirmwareFileName>
  <FirmwareHTMLFileName>dsap_htm39504.bin
  </FirmwareHTMLFileName>
  <Profileable>true</Profileable>
  <Legacy>false</Legacy>
  <UniqueID></UniqueID>
  <Author>Wavelink</Author>
  <SmallIcon>symbol_AP4131_16x16.gif</SmallIcon>
  <LargeIcon> symbol_AP4131_48x48.gif</LargeIcon>
  <SiteConsoleProperties>
  <DeviceTypeName>Symbol AP 4131</DeviceTypeName>
  <MIBKey>SymbolAP4131</MIBKey>
  <SupportsDefaultProfile>false
  </SupportsDefaultProfile>
  <SupportsAssignIP>false</SupportsAssignIP>
  <SupportsACL>false</SupportsACL>
  <SupportsNetID>false</SupportsNetID>
  <DefaultSSID>Edit ESS IDs in Advanced Dialog
  </DefaultSSID>
  </SiteConsoleProperties>
```

```
<AmcScripts>
<ScriptingProperties>
```

**NOTE** The preceding text uses a Symbol 4131 with version 3.95-04 firmware as an example.

4 Create an `<AmcScripts>` tag inside the `<ScrptingProperties>` tag and add a general script to the `.xml` file before the closing `</ScriptingProperties>` tag.

For example:

```
<AmcScripts>
 <Script name="OnGetVersion">
    Script(OnGetVersion)
    String(strResult)
      strResult=
      SNMP_Get_String("1.3.6.1.4.1.388.1.8.1.1.5")
      Return(strResult)
  </Script>
</AmcScripts>
```

**NOTE** For a list of scripting variables and actions, refer to *Appendix A: Variables, Actions, and Tags* on page 21. For script examples, refer to *Appendix B: Example Support Files* on page 45.

5 Add infrastructure device properties by inserting an `<OIDValues>` tag into the `.xml` file after the `</AmcScripts>` end tag. Each property requires an `<OID></OID>` entry. For examples, refer to *Appendix B: Example Support Files* on page 45.

**NOTE** The MG-SOFT MIB Browser can be used to retrieve infrastructure device properties. Refer to the MIB Browser's documentation for more information.

6 Add tags used to get and set the property. For a list of tags, refer to *Extended Device Support Tags* on page 42.

7 Specify the advanced properties user interface layout by inserting a `<UserInterface>` tag after the `<OIDValues>` tag. Each section of the user interface requires a `<Label></Label>` entry. For examples, refer to *`Appendix B: Example Support Files`* on page 45.

8 Ensure that the `.xml` file contains all required entries and save your changes. For a list of required entries, refer to the following section, *Required .xml File Entries*.

9 Create a zip file that contains the device support file and the two icon files detailed in *Device Icons* on page 13.

## Required .xml File Entries

This section lists the entries that must be present in the `.xml` file. These entries are required for the support file to function correctly in Avalanche MC.The information is divided into the following sections:

- Scripting Properties
- Avalanche MC Scripts
- OID Values
- Label Values
- Device Icons

### Scripting Properties

The following entries are needed by the Infrastructure dServer and the eServer:

| | |
|---|---|
| `DeviceRole` | Identifies the hardware as either an access point or wireless switch. Allowable values are `Access Point`, `Switch`, and `AccessPort`. |
| `Hardware` | Identifies the hardware supported by the file; appears as an entry in the hardware drop-down menu during profile creation. |

| | |
|---|---|
| `FirmwareVersion` | Identifies the firmware supported by the file; appears as an entry in the firmware drop-down menu during profile creation. |
| `ManufacturerName` | Identifies the manufacturer. |
| `FirmwareRangeOfSupport` | Identifies the range of firmware supported by this property file. |
| `ModelName` | Identifies the model. |
| `FirmwareFileName` | The name of the firmware file associated with this device version. Specifying this information in the `.xml` file eliminates the need for an entry in the `agentsvc.cfg` file. |
| `Profileable` | Indicates whether a profile may be created with this version, true or false. Including this name in the `.xml` file helps the eServer to automatically add support for this firmware.<br><br>**NOTE** This field is required; however, the value can be blank. |
| `Legacy` | Indicates whether the Infrastructure dServer has support for composite profiles for this device. Including this name in the `.xml` file helps the eServer to automatically add support for this firmware.<br><br>**NOTE** This field is required; however, the value can be blank. |
| `UniqueId` | This value is reserved and is not used. |
| `Author` | The author of the scripting file. |

| | |
|---|---|
| `SmallIcon` | A small icon which will be displayed in the infrastructure inventory in the Avalanche MC Console.<br><br>**NOTE** This field is required; however, the value can be blank. |
| `LargeIcon` | A large icon which will be displayed when viewing device details in the Avalanche MC Console.<br><br>**NOTE** This field is required; however, the value can be blank. |
| `SiteConsoleProperties` | Describes the values used in the Siteconsole tool when displaying the device. |
| `AmcScripts` | A section providing scripts that interact with the device. These scripts are not specific to a property, rather they access device information and are used primarily to discover the device. Refer to the following section for a list of supported scripts. |

### Avalanche MC Scripts

Avalanche MC identifies devices by running scripts. The following are the required Avalanche MC scripts:

| | |
|---|---|
| `OnGetVersion` | Gets the firmware version of the device. |
| `OnSetVersion` | Formats the version string into a format used by the Infrastructure dServer. |
| `OnGetDeviceName` | Returns the current name of the device. |
| `OnGetSubnetMask` | Returns the ethernet subnet mask for the device. |

| | |
|---|---|
| `OnGetLanMACAddress` | Returns the ethernet MAC address. |
| `OnGetModelName` | Gets the model of the device. |
| `OnGetFirmwareVersion` | Gets the firmware version of the device. |
| `OnGetManufacturer` | Returns the manufacturer of the device. |

#### OID Values

Any setting supported by the Infrastructure dServer requires an OID. The settings supported for the device each contain an OID entry in the property file. This entry contains the information to query, set, and render the setting.

#### Label Values

To organize OID settings in the user interface, create Label entries in the property file. Labels may be nested in a hierarchy. The OID's `<UISection>` tag contains the Label where the OID appears in the user interface.

#### Device Icons

The Avalanche MC Console uses icons when it displays information about a device. Create two `.gif` icons to represent the infrastructure device in Avalanche MC. One icon should be 16x16 pixels, and the other should be 48x48 pixels.

These icons are listed in the `.xml` file for flexibility. A reference to the icon name is required, and is converted to hex format prior to being sent to the eServer.

## Creating a Custom Setting File

Creating a support file for custom infrastructure device settings involves the following steps:

- Obtaining Settings Information
- Building the Custom Setting File

### Obtaining Settings Information

Before you can build a custom setting file, you must obtain information about the settings supported by the infrastructure device. You can view infrastructure device settings by entering the device's IP address into a web

browser and logging onto the device. Use an SNMP MIB browser to obtain details about each setting, such as the OID value.

---

**NOTE** The MG-SOFT MIB Browser can be used to retrieve infrastructure device properties. Refer to the MIB Browser's documentation for more information.

---

## Building the Custom Setting File

After you have obtained information about the infrastructure device settings, you can create a script-based file containing the custom device settings you want to manage. Avalanche MC will read the file to recognize, display, query, and configure the settings.

**To create a support file:**

1. Open Notepad or another text editor.
2. Create a new file starting with `as_` and ending in `.ext.xml`. The name of the file should correspond to the hardware and firmware of the infrastructure device.

   For example: `as_Symbol4131_0395-04.ext.xml` for a Symbol 4131 with version 3.95-04 firmware.
3. Paste the following text into the `.ext.xml` file:

```
<?xml version="1.0" encoding="UTF-8" ?>
<ScriptingProperties>
<OIDValues>
</OIDValues>
</ScriptingProperties>
```

4. Add a custom infrastructure device property by inserting an `<OID></OID>` entry into the `.ext.xml` file after the `<OIDValues>` tag. If desired, you can add multiple entries to support additional custom settings in one file. For examples, refer to *Appendix B: Example Support Files* on page 45.
5. Ensure that the `.xml` file contains all required entries and save your changes.

# Chapter 3: Adding Device Support

This chapter provides information about the following:

- Adding a New Infrastructure Device
- Adding Custom Properties
- Packaging and Deploying New Files

## Adding a New Infrastructure Device

This section provides information about adding support for a new infrastructure device to Avalanche MC, including the following:

- Importing a Support File
- Importing a Firmware File

### Importing a Support File

After you have created a device support file, import the file into Avalanche MC. Avalanche MC will use the support file to update the infrastructure device information.

**To import:**

1. Ensure you have created a device support file and saved it in a zip file with the device icons. You should know the file location on your system.
2. From the Avalanche MC Console, select **File > Import > Extended Device Support**.

   The *Extended Device Support* dialog box appears.

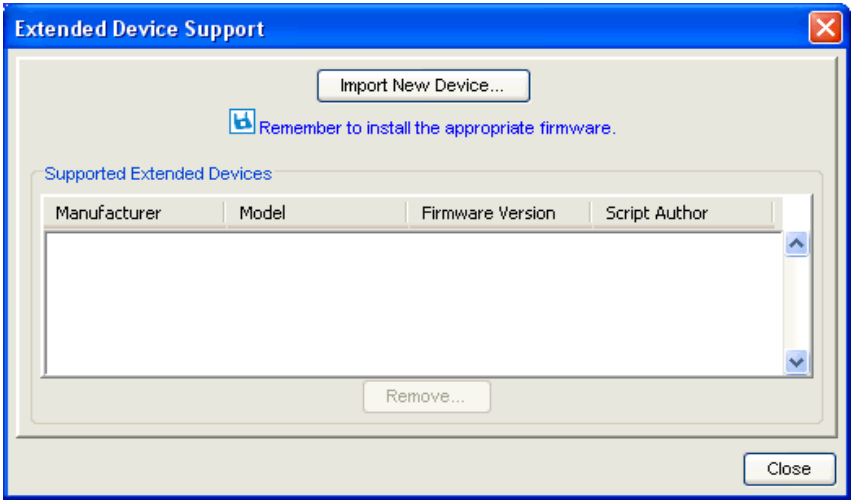


**Figure 3-1.** *Extended Device Support Dialog Box*

**3** Click the **Import New Device** button.

The *Select Support File* dialog box appears.

**4** Navigate to and select the `.zip` support file, then click **Select File**.

The *Select Support File* dialog box closes and a new dialog box appears, indicating whether the support file import was successful.

---

**NOTE** If the import was unsuccessful, the dialog box will indicate the reason the import failed. You can use this information to revise the support file as necessary.

---

**5** Click **OK**.

If the file import was successful, the device information appears in the **Supported Extended Devices** list.

**NOTE** The support file does not include any device firmware. The firmware file must be imported separately. You can click on the **Remember to install the appropriate firmware** link, or import the firmware manually. See the following section, *Importing a Firmware File*, for more information.

6 If you want to remove a support file, select the appropriate file from the **Supported Extended Devices** list, and click **Remove**.

7 Click **Close** to exit the *Extended Device Support* dialog box and return to the Avalanche MC Console.

## Importing a Firmware File

For Avalanche MC to fully support a new infrastructure device, you must import the device's firmware file in addition to the support file. You must have downloaded the firmware files from either the device manufacturer or from Wavelink.

**To import:**

1 Ensure you have downloaded the firmware files and know their location on your system.

2 From the Avalanche MC Console, select **File > Import > Firmware Files**.

The *Manage Infrastructure Firmware* dialog box appears. This dialog box displays the manufacturer, model, version, and indicates whether the firmware has been installed.

3 If desired, narrow the list using the **Manufacturer** and **Model** drop-down menus.

4 Select the firmware from the list and click **Install**.

The *Select Source Folder* dialog box appears, displaying the firmware file name in the **File of type** text box.

5 Navigate to and select the folder that contains the firmware file, then click **Select Folder**.

A success message appears when the transmit completes. The firmware will appear in the applicable dialog box when you add an infrastructure

profile. The new firmware is also available to deploy to Infrastructure dServers.

**NOTE** For further information about Avalanche MC firmware support, refer to the *Wavelink Avalanche MC User Guide*.

## Adding Custom Properties

Extended Device Support enables you to add custom infrastructure device properties to new or existing devices.

**To add properties:**

1 Ensure you have created a custom property support file and know its location on your system.

2 From the Avalanche MC Console, select **File > Import > Custom Advanced Settings**.

The *Custom Advanced Settings* dialog box appears.

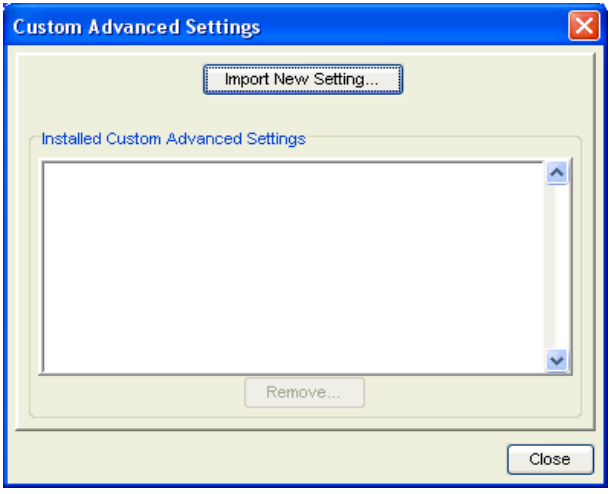


**Figure 3-2.** *Custom Advanced Settings Dialog Box*

3 Click **Import New Setting**.

The *Select Custom Advanced Settings File* dialog box appears.

4 Navigate to and select the custom settings file and click **Select File**.

The *Select Custom Advanced Settings File* dialog box closes and the advanced settings information appears in the **Installed Custom Advanced Settings** list.

5 If you want to remove a custom settings file, select the appropriate file from the Installed Custom Advanced Settings list and click **Remove.**

6 Click **Close** to exit the *Custom Advanced Settings* dialog box and return to the Avalanche MC Console.

## Packaging and Deploying New Files

Once you have finished importing your support and firmware files, you need to build a firmware package with the new files. The firmware package should then be deployed to your dServers. For detailed instructions about creating and deploying packages, refer to *Wavelink Avalanche MC User Guide.*

# Appendix A: Variables, Actions, and Tags

This appendix lists the variables, actions, and tags used to create Extended Device Support script-based files.

## Extended Device Support Variables

The following are variables supported by Extended Device Support:

| | |
|---|---|
| Boolean | Can be True or False.<br>**Default Value:** False |
| Number | Can have a positive or negative value.<br>**Default Value: 0** |
| String | Contains values such as “WS5000 Switch.”<br>**Default Value:** Empty string |
| Bytes | Contains binary information.<br>**Default Value:** 0 |
| Vector_String | Contains many string variables.<br>**Default Value:** Empty list |
| Vector_Number | Contains many number variables.<br>**Default Value:** Empty list |

# Extended Device Support Actions

The following are actions supported by Extended Device Support:

| Action | Description | Return Type/ Parameters |
|---|---|---|
| `Blank_Line` | Proceeds to the next instruction without taking any action. | Void. |
| `Comment`<br>`//`<br>`--` | Proceeds to the next instruction without taking any action. You can use the word `Comment`, or use either of the `//` and `--` symbols. | Returns void.<br>Takes a string parameter that is the comment. |
| `Goto` | Jumps to the supplied label. | Returns void.<br>Takes a string parameter that is the label. |
| `Label` | Label to which a `Goto` can jump. | Returns void.<br>Takes a string parameter that is the label. |
| `Return` | Exits the script normally. | Void. |
| `If` | If the test is TRUE, continues executing until the next `Else` or `End_If` statement. Otherwise, only executes actions (if any) between the next `Else` and `End_If` statements. | Returns void.<br>Takes a boolean parameter that gets tested. |
| `If_Not` | If the test is FALSE, continues executing until the next `Else` or `End_If` statement. Otherwise, only executes actions (if any) between the next `Else` and `End_If` statements. | Returns void.<br>Takes a boolean parameter that gets tested. |

| Action | Description | Return Type/ Parameters |
|---|---|---|
| `Else` | Start of statements to be executed if an `If` test fails. This command is only valid inside of an `If` block. | Void. |
| `End_If` | End of statements to be executed for an `If` test. | Void. |
| `While` | If the Test is TRUE, the statements after `While` and before the next `EndWhile` statement are executed and the `While` statement will be executed again. Otherwise, execution will proceed to the next `EndWhile` statement. The `While` loop will continue to execute until the test fails, a `Break` command is executed, or the script exits. | Returns void.<br>Takes a boolean parameter that gets tested. |
| `While_Not` | If the Test is FALSE, the statements after `While` and before the next `EndWhile` statement are executed and the `While` statement will be executed again. Otherwise, execution proceeds to the next `EndWhile` statement. The `While` loop continues to execute until the test succeeds, a `Break` command is executed, or the script exits. | Returns void.<br>Takes a boolean parameter that gets tested. |
| `End_While` | End of statements to be executed for a `While` test. | Void. |

| Action | Description | Return Type/ Parameters |
|---|---|---|
| `Continue` | Jumps back to the last `While` statement and re-tests the test value. This command is only valid inside of a `While` loop. | Void. |
| `Break` | Jumps to the first statement following the next `End_While` statement (exiting the loop). This command is only valid inside of a `While` loop. | Void. |
| `Delay` | Suspends the current script until the specified time has passed. The time is in milliseconds, so a value of 1000 would be 1 second. | Returns void.<br>Takes an integer parameter that is the number of milliseconds. |
| `Boolean_Set` | Returns TRUE if the test is TRUE, FALSE otherwise. | Returns boolean.<br>Takes a boolean parameter that gets tested. |
| `Boolean_Not` | Returns FALSE if the test if TRUE, TRUE otherwise. | Returns boolean.<br>Takes a boolean parameter that gets tested. |
| `Boolean_And` | Returns TRUE if all test values are TRUE. Returns FALSE otherwise. All tests will be evaluated each time this action is taken. | Returns boolean.<br>Takes two to five boolean parameters that get tested. |
| `Boolean_Or` | Returns TRUE if one or more test values are TRUE. Returns FALSE otherwise. All tests will be evaluated each time this action is taken. | Returns boolean.<br>Takes two to five boolean parameters that get tested. |

| Action | Description | Return Type/ Parameters |
|---|---|---|
| `Boolean_Equal` | Returns TRUE if both Test1 and Test2 are TRUE, or both Test1 and Test2 are FALSE. Returns FALSE otherwise. | Returns boolean.<br>Takes two boolean parameters that get tested. |
| `Boolean_Not_Equal` | Returns FALSE if both Test1 and Test2 are TRUE, or both Test1 and Test2 are FALSE. Returns TRUE otherwise. | Returns boolean.<br>Takes two boolean parameters that get tested. |
| `Logging_On` | Creates a log file that records all subsequent script execution activity. If `Overwrite Previous` is TRUE, a previous log file will be overwritten. Otherwise, the new information will be appended to the existing file.Logging is only turned on for the current script. Scripts called by this script will not have logging enabled. | Returns void.<br>Takes a string parameter that is the file path and a boolean parameter that determines if the previous file gets overwritten. |
| `Logging_Off` | Turns of logging for the script. | Void. |
| `Bytes_Set_Byte_At` | Sets one byte in a byte array, at the zero-based index. | Returns void.<br>Takes a bytes parameter that gets a byte set, and integer parameter that is the index into the bytes and a byte parameter that is the value to set. |
| `Bytes_Set_Length` | Sets the length of a byte array. | Returns void.<br>Takes a bytes parameter that gets its length set and an integer parameter that is the length. |

| Action | Description | Return Type/ Parameters |
|---|---|---|
| `Bytes_Equal` | Returns TRUE if Test1 is the same as Test2, FALSE otherwise. | Returns boolean.<br>Takes a bytes parameter Test1 for comparing with the bytes parameter Test2. |
| `Bytes_Not_Equal` | Returns FALSE if Test1 is the same as Test2, TRUE otherwise. | Returns Boolean.<br>Takes a bytes parameter Test1 for comparing with the bytes parameter Test2 |
| `Bytes_To_String` | Converts a byte array into a string. | Returns string.<br>Takes a bytes parameter that gets changed into a string. |
| `Bytes_Get_Byte_At` | Gets one byte from a byte array, at the zero-based index. | Returns integer.<br>Takes a bytes parameter and an integer index that is the location of the byte to return. |
| `Bytes_Get_Length` | Gets the number of bytes in a byte array. | Returns integer.<br>Takes a bytes parameter. |
| `Bytes_Set` | Sets a byte array. | Returns bytes.<br>Takes a bytes parameter. |
| `Bytes_To_IP_Address_String` | Converts a byte array into an IP address string. | Returns string.<br>Takes a bytes parameter that is an IP address. |
| `String_To_Bytes` | Converts a string to a byte array. | Returns bytes.<br>Takes a bytes parameter to convert to a string. |

| Action | Description | Return Type/ Parameters |
|---|---|---|
| `String_Set_Character_At` | Sets one string character of a string, at the zero-based index. | Returns void.<br>Takes a string parameter that gets a character set, an integer parameter that is the index into the string, and a one-character string parameter that is the value to set. |
| `String_Get_Character_At` | Gets one character of a string, at the zero-based index. | Returns string.<br>Takes a string parameter and an index parameter that is the location of the character to get. |
| `String_Empty` | Returns TRUE if the string is 0 characters in length, FALSE otherwise. | Returns boolean.<br>Takes a string parameter that is checked. |
| `String_Less_Than` | Returns TRUE if Test1 precedes Test2 in alphabetical ordering, FALSE otherwise. If the `Maximum Length` value is greater than 0, any characters after the specified number of characters are ignored. If `Ignore Case` is TRUE, then upper-case and lower-case letters are considered to be equal. | Returns boolean.<br>Takes a string parameter that is checked. |

| Action | Description | Return Type/ Parameters |
|---|---|---|
| `String_Less_Than_Or_Equal` | Returns TRUE if Test1 precedes Test2 in alphabetical ordering or they are the same string, FALSE otherwise. If the `Maximum Length` value is greater than 0, any characters after the specified number of characters are ignored. If `Ignore Case` is TRUE then upper-case and lower-case letters are considered to be equal. | Returns boolean.<br>Takes two string parameters that are compared, an integer maximum-length parameter, and a boolean parameter that specifies if the check should treat upper and lower case as equal. |
| `String_Equal` | Returns TRUE if Test1 and Test2 are the same string, FALSE otherwise. If the `Maximum Length` value is greater than 0, any characters after the specified number of characters are ignored. If `Ignore Case` is TRUE, then upper-case and lower-case letters are considered to be equal. | Returns boolean.<br>Takes two string parameters that are compared, an integer maximum-length parameter, and a boolean parameter that specifies if the check should treat upper and lower case as equal. |
| `String_Greater_Than_Or_Equal` | Returns TRUE if Test1 follows Test2 in alphabetical ordering or they are the same string, FALSE otherwise. If the `Maximum Length` value is greater than 0, any characters after the specified number of characters are ignored. If `Ignore Case` is TRUE then upper-case and lower-case letters are considered to be equal. | Returns boolean.<br>Takes two string parameters that are compared, an integer maximum-length parameter, and a boolean parameter that specifies if the check should treat upper and lower case as equal. |

| Action | Description | Return Type/ Parameters |
|---|---|---|
| `String_Greater_Than` | Returns TRUE if Test1 follows Test2 in alphabetical ordering, FALSE otherwise. If the `Maximum Length` value is greater than 0, any characters after the specified number of characters are ignored. If `Ignore Case` is TRUE then upper-case and lower-case letters are considered to be equal. | Returns boolean.<br>Takes two string parameters that are compared, an integer maximum-length parameter, and a boolean parameter that specifies if the check should treat upper and lower case as equal. |
| `String_Not_Equal` | Returns FALSE if Test1 and Test2 are the same string, TRUE otherwise. If the `Maximum Length` value is greater than 0, any characters after the specified number of characters are ignored. If `Ignore Case` is TRUE then upper-case and lower-case letters are considered to be equal. | Returns boolean.<br>Takes two string parameters that are compared, an integer maximum-length parameter, and a boolean parameter that specifies if the check should treat upper and lower case as equal. |
| `String_Set` | Returns the value of the string. | Returns string.<br>Takes a string parameter that gets returned. |
| `String_Combine` | Returns the value of string1 combined with string2. | Returns string.<br>Takes two string parameters where the second string gets concatenated onto the first string. |
| `String_Left` | Returns a string with just the first *n* characters of the input string. If the input string is less than *n* characters, the entire string is returned. | Returns string.<br>Takes a string parameter and an integer number of characters parameter. |

| Action | Description | Return Type/ Parameters |
|---|---|---|
| `String_Right` | Returns a string with just the last *n* characters of the input string. If the input string is less than *n* characters, the entire string is returned. | Returns string.<br>Takes a string parameter and an integer number of characters parameter. |
| `String_Middle` | Returns a string with just the middle *n* characters of the input string. The string parsing starts at the position specified, with 0 being the left-most character, so a position value of 0 is the same as `String_Left`.If the input string is less than *n* characters, the entire string is returned. | Returns string.<br>Takes a string parameter and an integer starting number parameter and an integer number of characters parameter. |
| `String_Upper` | Returns a string with all characters converted to uppercase. | Returns string.<br>Takes a string parameter that gets converted to uppercase. |
| `String_Lower` | Returns a string with all characters converted to lowercase. | Returns string.<br>Takes a string parameter that gets converted to lowercase. |
| `String_Replace` | Returns a string where all instances of `Substring to Replace` have been replaced with `Replacement Substring`. If `Ignore Case` is TRUE then upper-case and lower-case letters are considered to be equal. | Returns string.<br>Takes a string parameter that is the base string, a string parameter that is the substring to replace, a string parameter that replaces the substring and a boolean parameter that specifies if upper and lower case are treated as equal. |

| Action | Description | Return Type/ Parameters |
|---|---|---|
| `String_Only_Characters` | Returns a string where all characters in `String to Parse` that are not in `Characters to Keep` have been deleted. If `Ignore Case` is TRUE then upper-case and lower-case letters are considered to be equal. | Returns a string.<br>Takes a string parameter that is the base string, a string parameter that is the set of characters to keep in the base string, and a boolean parameter that specifies if upper and lower case are treated as equal. |
| `String_Strip_Characters` | Returns a string where all characters in `String to Parse` that are in `Characters to Strip` have been deleted. If `Ignore Case` is TRUE then upper-case and lower-case letters are considered to be equal. | Returns String.<br>Takes a string parameter that is the base string, a string parameter that is the set of characters to delete in the base string and a boolean parameter that specifies if upper and lower case are treated as equal. |
| `String_Trim_Spaces_Start` | Returns a string where all spaces and tabs at the start of the string have been deleted. | Returns String.<br>Takes a string parameter that gets trimmed. |
| `String_Trim_Spaces_End` | Returns a string where all spaces and tabs at the end of the string have been deleted. | Returns String.<br>Takes a string parameter that gets trimmed. |
| `String_Length` | Returns the number of characters in the string. Returns 0 if the string is empty (has no characters). | Returns integer.<br>Takes a string parameter. |

| Action | Description | Return Type/ Parameters |
|---|---|---|
| `String_Find_First` | Finds the first instance of the substring inside the string, and returns the position where that substring starts. The left-most position is 0, so a value of 0 would be returned if the string started with the substring. A value of -1 is returned if no instances of the substring are in the string. | Returns Integer.<br>Takes a string parameter that is the base string, a string parameter that is the substring to find, and a boolean parameter that specifies if upper and lower case are treated as equal. |
| `String_Find_Last` | Finds the last instance of the substring inside the string, and returns the position where that substring starts. The left-most position is 0, so a value of 0 would be returned if the only substring found was at the beginning of the string. A value of -1 is returned if no instances of the substring are in the string. | Returns Integer.<br>Takes a string parameter that is the base string, a string parameter that is the substring to find, and a boolean parameter that specifies if upper and lower case are treated as equal. |
| `String_To_Number_Binary` | Returns the binary (base-2) number represented by the string. Parsing the string continues until a character other than a 0 or 1 is reached. If the string does not represent a binary number, a 0 is returned. | Returns Integer.<br>Takes a string parameter that gets converted. |

| Action | Description | Return Type/ Parameters |
|---|---|---|
| `String_To_Number_Octal` | Returns the octal (base-8) number represented by the string. Parsing the string continues until a character other than a 0 through 7 is reached. If the string does not represent an octal number, a zero is returned. | Returns Integer.<br>Takes a string parameter that gets converted. |
| `String_To_Number_Decimal` | Returns the decimal (base-10) number represented by the string. Parsing the string continues until a character other than a 0 through 9 is reached. If the string does not represent a decimal number, a 0 is returned. | Returns Integer.<br>Takes a string parameter that gets converted. |
| `String_To_Number_Hexadecimal` | Returns the hexadecimal (base-16) number represented by the string. Parsing the string continues until a character other than a 0 through 9, a through f, or A through F is reached. If the string does not represent a hexadecimal number, a 0 is returned. | Returns Integer.<br>Takes a string parameter that gets converted. |
| `Number_Less_Than` | Returns TRUE if Test1 is smaller than Test2, FALSE otherwise. | Returns Boolean.<br>Takes two integer parameters that are compared. |
| `Number_Less_Than_Or_Equal` | Returns TRUE if Test1 is no greater than Test2, FALSE otherwise. | Returns Boolean.<br>Takes two integer parameters that are compared. |

| Action | Description | Return Type/ Parameters |
|---|---|---|
| Number_Equal | Returns TRUE if Test1 is the same as Test2, FALSE otherwise. | Returns Boolean.<br>Takes two integer parameters that are compared. |
| Number_Greater_Than_Or_Equal | Returns TRUE if Test1 is no smaller than Test2, FALSE otherwise. | Returns Boolean.<br>Takes two integer parameters that are compared. |
| Number_Greater_Than | Returns TRUE if Test1 is larger than Test2, FALSE otherwise. | Returns Boolean.<br>Takes two integer parameters that are compared. |
| Number_Not_Equal | Returns FALSE if Test1 is the same as Test2, TRUE otherwise. | Returns Boolean.<br>Takes two integer parameters that are compared. |
| Number_To_String_Binary | Returns a string with the binary (base 2) representation of the number. | Returns String.<br>Takes an integer parameter that gets converted. |
| Number_To_String_Octal | Returns a string with the octal (base 8) representation of the number. | Returns String.<br>Takes an integer parameter that gets converted. |
| Number_To_String_Decimal | Returns a string with the decimal (base 10) representation of the number. | Returns String.<br>Takes an integer parameter that gets converted. |
| Number_To_String_Hexadecimal_Lowercase | Returns a string with the hexadecimal (base 16) representation of the number using lowercase characters. | Returns String.<br>Takes an integer parameter that gets converted. |
| Number_To_String_Hexadecimal_Uppercase | Returns a string with the hexadecimal (base 16) representation of the number using uppercase characters. | Returns String.<br>Takes an integer parameter that gets converted. |

| Action | Description | Return Type/ Parameters |
|---|---|---|
| `Number_To_Character` | Returns a string one character in length, where the value for that character is the supplied number. For example, a number value of 87 would return a string consisting of a `W` -- the ASCII character for value 87. | Returns String.<br>Takes a number parameter that is converted into a one-character string. |
| `Number_Set` | Returns the value of the number. | Returns Integer.<br>Takes an integer parameter that gets returned. |
| `Number_Plus` | Returns the sum of the two numbers. | Returns Integer<br>Takes two integer parameter that get added. |
| `Number_Minus` | Returns the value when Number2 is subtracted from Number1. | Returns Integer<br>Takes a base integer parameter and an integer parameter that gets subtracted from the base. |
| `Number_Multiply` | Returns the product of the two numbers. | Returns Integer.<br>Takes two integer parameter that get multiplied. |
| `Number_Divide` | Returns the value when Number1 is divided by Number2. Because the numbers are integers, the remainder is ignored. For example, 7 divided by 3 would return 2. | Returns Integer<br>Takes a base integer parameter and an integer parameter that gets divided into the base. |
| `Number_Divide_Remainder` | Returns the remainder when Number1 is divided by Number2. For example, 7 divided by 3 would return a remainder of 1. | Returns Integer<br>Takes a base integer parameter and an integer parameter that gets divided into the base. |

| Action | Description | Return Type/ Parameters |
|---|---|---|
| `Character_To_Number` | Converts the character at position Index in the string into the number value for that character. An index of 0 indicates the left-most character in the string. If the Index does not point to a character, a value of 0 is returned. | Returns Integer.<br>Takes a string parameter and an integer parameter that is the index of the character that gets converted. |
| `Vector_String_Append` | Pushes a value onto the vector list. | Returns void.<br>Takes a vector-string parameter and a string parameter that gets appended to the end of the vector-string. |
| `Vector_String_Clear` | Clears the entire vector list. | Returns void.<br>Takes a vector-string parameter that gets cleared. |
| `Vector_String_Insert_At` | Inserts a string in the vector list. | Returns void.<br>Takes a vector-string parameter, a zero-based integer parameter that is the position of the insertion, and a string parameter that gets inserted into the vector-string. |
| `Vector_String_Erase_At` | Erases a string in the vector list. | Returns void.<br>Takes a vector-string parameter and a zero-based integer parameter that is the position of the string that gets erased in the vector-string. |

| Action | Description | Return Type/ Parameters |
|---|---|---|
| `Vector_String_Set_At` | Sets a string in the vector list. | Takes a vector-string parameters, a zero-based integer parameter that is the position of the string to set, and a string parameter that gets set in the vector-string. |
| `Vector_Number_Append` | Pushes a value onto the vector list. | Returns void.<br>Takes a vector-integer parameter and a integer parameter that gets appended to the end of the vector-integer. |
| `Vector_Number_Clear` | Clears the entire vector list. | Returns void.<br>Takes a vector- integer parameter that gets cleared. |
| `Vector_Number_Insert_At` | Inserts a number in the vector list. | Returns void.<br>Takes a vector-integer parameter, a zero-based integer parameter that is the position of the insertion, and an integer-parameter that gets inserted into the vector-integer. |
| `Vector_Number_Erase_At` | Erases a number in the vector list. | Returns void.<br>Takes a vector-integer parameter and a zero-based integer parameter that is the position of the integer that gets erased in the vector-integer. |

| Action | Description | Return Type/ Parameters |
|---|---|---|
| Vector_Number_Set_At | Sets a number in the vector list. | Takes a vector-integer parameter, a zero-based integer parameter that is the position of the integer to set, and an integer parameter that gets set in the vector-integer. |
| Vector_String_Get_At | Gets a string in the vector list. | Returns string.<br>Takes a vector-string parameter and an integer parameter that is the index of the string to get from the vector-string. |
| Vector_String_Get_Length | Gets the number of elements in a string vector. | Returns integer.<br>Takes a vector-string parameter. |
| Vector_Number_Get_Length | Gets the number of elements in a number vector. | Returns integer.<br>Takes a vector-number parameter. |
| Vector_Number_Get_At | Gets a number in the vector list. | Returns integer.<br>Takes a vector-number parameter and an integer index that is the location of the number to return. |
| Vector_String_Set | Sets a vector-string array. | Returns vector-string.<br>Takes a vector-string parameter. |
| Vector_Number_Set | Sets a vector-number array. | Returns vector-number.<br>Takes a vector-number parameter. |
| Get_Time | Returns the number of seconds that have elapsed since January 1, 2000. | Integer. |

| Action | Description | Return Type/ Parameters |
|---|---|---|
| `Get_Time_Since_Reset` | Returns the number of milliseconds that the computer has been non-suspended since the last reboot. | Integer. |
| `Get_Oid_Value_String` | Looks up and returns the current value of the specified property. | Returns string.<br>Takes a string parameter that is the OID and a string parameter that is `Get` for using the `Get Store` or `Set` for using the `Set Store`. |
| `Get_Oid_Value_Integer` | Looks up and returns the current value of the specified property. | Returns integer.<br>Takes a string parameter that is the OID and a string parameter that is `Get` for using the `Get Store` or `Set` for using the `Set Store`. |
| `Get_Oid_Value_Bytes` | Looks up and returns the current value of the specified property. | Returns bytes.<br>Takes a string parameter that is the OID and a string parameter that t is `Get` for using the `Get Store` or `Set` for using the `Set Store`. |
| `Get_Oid_Table_Length` | Gets the number of rows in the table. | Returns integer.<br>Takes a string parameter that is the OID of the table. |
| `Set_Oid_Table_Changed` | Sets a flag that the OID Table has changed. | Returns void.<br>Takes a string parameter that is the OID of the table that changed. |
| `Get_Oid_Table_Changed` | Returns TRUE if the flag is set that the OID Table has changed. | Returns boolean.<br>Takes a string parameter that is the OID of the table. |

| Action | Description | Return Type/ Parameters |
|---|---|---|
| `Bytes_To_MAC_Address_String` | Converts a byte array into a MAC address string (I.e. 00:01:02:43:B1:22). | Returns string.<br>Takes a bytes parameter that is a MAC address. |
| `SNMP_Get_String` | Gets the string value of an OID. | Returns a string.<br>Takes a string parameter for the OID. |
| `SNMP_Get_Integer` | Gets the integer value of an OID. | Returns an integer.<br>Takes a string parameter for the OID. |
| `SNMP_Set_String` | Sets the string value of an OID. | Returns void.<br>Takes a string parameter for the OID and a string parameter for the new value. |
| `SNMP_Set_Bytes` | Sets the integer value of an OID. | Returns void.<br>Takes a string parameter for the OID and an integer parameter for the new value. |
| `SNMP_Get_Integer_IP_Index` | Gets the integer value of an OID. | Returns an integer.<br>Takes a string parameter for the OID. This function appends the IP address of the device onto the parameter. |
| `SNMP_Get_String_IP_Index` | Gets the string value of an OID. | Returns void.<br>Takes a string parameter for the OID and a string parameter for the new value. This function appends the IP address of the device onto the parameter. |

| Action | Description | Return Type/ Parameters |
|---|---|---|
| SNMP_Get_Bytes_IP_Index | Gets the bytes value of an OID. | Returns void.<br>Takes a string parameter for the OID and an integer parameter for the new value. This function appends the IP address of the device onto the parameter. |
| SNMP_Get_Bytes | Gets the bytes value of an OID. | Returns bytes.<br>Takes a string parameter for the OID. |
| SNMP_Set_Integer | Sets the bytes value of an OID. | Returns void.<br>Takes a string parameter for the OID and a bytes parameter for the new value. |
| Telnet_Send | Sends a string over Telnet, returning TRUE if successful. | Returns boolean.<br>Takes a string parameter to send over Telnet and an integer wait time in seconds in case a Telnet login is needed. |
| Telnet_Receive | Receives Telnet data, returns TRUE if successful. | Returns boolean.<br>Takes a string parameter that gets set to the received data and an integer wait time in seconds. |
| Telnet_Send_And_Receive | Sends a string over Telnet, then waits for the Telnet response. Returns TRUE if successful. | Returns boolean.<br>Takes a string parameter to send over Telnet, a string parameter that gets set to the results and an integer wait time in seconds. |

| Action | Description | Return Type/ Parameters |
|---|---|---|
| `Telnet_Warning_For_Response` | Logs a warning if the Telnet response matches this string. Use `*` to cause a warning for any non-empty response. | Returns void.<br>Takes a string parameter that is the response to look for, or empty to not look, and takes a boolean parameter (TRUE) to replace all other responses to look for. |
| `Telnet_Set_Expected_Prompt` | Sets the response string that is expected to be received after the next call to `Telnet_Send`. | Returns void.<br>Takes a string parameter that is the expected response. |

# Extended Device Support Tags

The following are tags used to get and set values on infrastructure devices.

**NOTE** Because these are `.xml` tags, you must add a closing tag after the value.

| Tag | Type | Description |
|---|---|---|
| `<FriendlyName>` | String | The name of the property as viewed by the user. |
| `<DottedName>` | String | The dotted name or OID used by this property to get and set its value. |
| `<Protocol>` | String | The method used to get and set this property's value. Acceptable values are **Telnet** or **SNMP**. |
| `<OnGet>` | Script | Script used to obtain the current value of the property. |
| `<OnAfterGet>` | Script | Script used to translate the value received into one that can be displayed to the user. |

| Tag | Type | Description |
|---|---|---|
| <OnChangeSNMPTypeForGet> | Script | Script used to change the type of SNMP setting received into another type. |
| <OnSet> | Script | Script used to set the property with a new value. |
| <OnBeforeSet> | Script | Script used to translate the value to send into one that is acceptable to the device. |
| <OnChangeSNMPTypeSet> | Script | Script used to change the type of SNMP setting into another before sending it to the device. |
| <SNMPType> | Integer | The executed SNMP type of the value from the device. |
| <Type> | Integer | The type used to enter and display the value in the UI. |
| <DefaultValue> | String or integer | The default value used if the device cannot be reached. |
| <Profileable> | Integer | If this setting can be placed in a profile this value is 1, otherwise 0. |
| <Range> | String or integer | The range of acceptable settings separated by semicolons. |
| <Default> | String or integer | The initial value used in the UI when displaying the setting for a profile. |
| <RWPermission> | String | Specify R if this setting is read only or R/W if it can be both read and written for a profile. |
| <Description> | String | The description of this setting in the UI. |
| <UISection> | String | The label path separated by forward slashes to indicate which section in the UI displays this setting. |
| <UIPosition> | Integer | The relative order, beginning at position 1 of this setting in its section. |
| <Query> | Integer | Instructs the Avalanche MC Infrastructure dServer to include this setting in the list of queried values. |

# Appendix B: Example Support Files

This appendix provides example support files. Use these examples to customize your own support files as desired.

## Example 1: New Device Support File

This is an example of an `As*.xml` device support file that adds support for a Symbol 4131 Access Point with version 3.95-04 firmware.

### Example Code

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="no" ?>
<ScriptingProperties>
<Hardware>SymbolAP4131</Hardware>
<FirmwareVersion>03.95-04</FirmwareVersion>
<FirmwareRangeOfSupport>Version Equal To=3.95-04
</FirmwareRangeOfSupport>
<DeviceType>AccessPoint</DeviceType>
<ManufacturerName>Symbol</ManufacturerName>
<ModelName>AP-4131</ModelName>
<FimrwareFileName>dsap_fw39504.bin</FimrwareFileName>
<FimrwareHTMLFileName>dsap_htm39504.bin
</FimrwareHTMLFileName>
<Profileable>true</Profileable>
<Legacy>false</Legacy>
<UniqueID />
<AmcScripts>
<Script name=
"OnGetVersion">Script(OnGetVersion)String(strResult)
strResult=SNMP_Get_String("1.3.6.1.4.1.388.1.8.1.1.5")
Return(strResult)
</Script>
<Script Name=
"OnSetVersion">Script(OnSetVersion)Parameter(String,
strVersion) Parameter(String,strMajor) Parameter(Number,
nMinor) String(strResult)
Comment ----------------------------------------
Comment This function needs to take a version string
and parse Comment it into 3 values - Major, Minor, and
version string Comment ------------------------------
---------- strResult = strVersion strMajor = "3.95" nMinor = 4
Return(strResult)</Script>
<ScriptName="OnGetDeviceName">
Script(OnGetDeviceName) String(strDeviceName)strDeviceName =
```

```
SNMP_Get_String("1.3.6.1.4.1.388.1.8.1.2.1")Return(strDeviceNam
e)</Script>
<ScriptName="OnGetSubnetMask">Script(OnGetSubnetMask)
String(strSubnetMask) strSubnetMask =
SNMP_Get_String("1.3.6.1.4.1.388.1.8.1.2.3")
Return(strSubnetMask)</Script>
<Script Name="OnGetLanMACAddress">
Script(OnGetLanMACAddress) String(strResult)
strResult =SNMP_Get_String("1.3.6.1.4.1.388.1.8.1.1.4")
Return(strResult)</Script>
<Script Name="OnGetModelName">Script(OnGetModelName)
String(strResult) strResult =
SNMP_Get_String("1.3.6.1.4.1.388.1.8.1.1.1")
Return(strResult)</Script>
<Script Name="OnGetFirmwareVersion">
Script(OnGetFirmwareVersion) String(strResult)
strResult =SNMP_Get_String("1.3.6.1.4.1.388.1.8.1.1.5")
Return(strResult)</Script>
</AmcScripts>
<OIDValues>
<OID>
<DottedName>1.3.6.1.4.1.388.1.8.1.2.1.0</DottedName>
<Script name="OnGet">Script(OnGet) String(strResult)
strResult =SNMP_Get_String("1.3.6.1.4.1.388.1.8.1.2.1.0")
Return(strResult)</Script>
<Script name="OnSet">Script(OnSet)
String(strResult)String(strVersion) Number(nResultLength)
Number(nVersionPosition) strResult = "OnSet"
Return(strResult)</Script>
<FriendlyName>Unit Name</FriendlyName>
<Type>String</Type>
<Range>0..127</Range>
<RWPermission>R/W</RWPermission>
<Description>An ASCII string used to identify this access point.
This string can be up to 127 characters.
</Description>
<UISection>LABEL_SystemConfig</UISection>
<UIPosition>1</UIPosition>
</OID>
<OID>
<DottedName>1.3.6.1.4.1.388.1.8.1.2.5.0</DottedName>
<Script name="OnGet">Script(OnGet) Number(nResult)
nResult = SNMP_Get_Integer("1.3.6.1.4.1.388.1.8.1.2.5.0")
Return(nResult)</Script>
<Script name="OnSet">Script(OnSet) Parameter(Number,
nInterface)
```

```
SNMP_Set_Integer("1.3.6.1.4.1.388.1.8.1.2.5.0", nInterface)</
Script>
<FriendlyName>Default Interface</FriendlyName>
<Type>ComboBox</Type>
<Range>ethernet(1);wlap(3)</Range>
<Default>1</Default>
<RWPermission>R/W</RWPermission>
<Description>Specifies the default interface (Ethernet or WLAP)
for the access point to forward an unknown packet. Default is
Ethernet (1).</Description>
<UISection>LABEL_SystemConfig</UISection>
<UIPosition>2</UIPosition>
</OID>
<OID>
<DottedName>1.3.6.1.4.1.388.1.8.1.2.18.0</DottedName>
<Script name="OnGet">Script(OnGet) String(strResult)
strResult = SNMP_Get_String("1.3.6.1.4.1.388.1.8.1.2.18.0")
Return(strResult)</Script>
<Script name="OnSet">Script(OnSet) Parameter(String,
strMHMD5Key)
SNMP_Set_String("1.3.6.1.4.1.388.1.8.1.2.18.0",strMHMD5Key)</
Script>
<Type>Password</Type>
<Range>0..13</Range>
<Default>Symbol-MD5Key</Default>
<RWPermission>R/W</RWPermission>
<Description>The MD5 key is a secret key shared between home
agent and the mobile device. This key is used to generate a MD5
checksum, which is used for packet authentication. Length of
this variable should not exceed 13 characters.</Description>
<UISection>LABEL_SystemConfig</UISection>
<UIPosition>3</UIPosition>
</OID>
</OIDValues>
<OtherProperties>
<VLANEnableProperties>
<Property Oid="1.3.6.1.4.1.388.1.8.1.12.1.0" />
</VLANEnableProperties>
</OtherProperties>
<SupportedFeatures>
<Feature>VLANEnable</Feature>
<Feature>StatAlertSupport</Feature>
</SupportedFeatures>
<UserInterface>
<UILabels>
<Label Oid="LABEL_SystemConfig" Name="SystemConfig"
```

```
dbMapped="false" />
</UILabels>
</UserInterface>
</ScriptingProperties>
```

## Example 2: Custom Setting File

This is an example of an `As*.ext.xml` custom setting file that adds support for an Airbeam server file to be downloaded to a Symbol 4131 with version 3.95-04 firmware.

### Example Code

```
<?xml version="1.0" encoding="UTF-8" ?>
<ScriptingProperties>
<OIDValues>
<OID>
<FriendlyName>Airbeam File Name</FriendlyName>
<DottedName>1.3.6.1.4.1.388.1.8.1.2.33.0</DottedName>
<Query>1</Query>
<SNMPType>4</SNMPType>
</OID>
</OIDValues>
</ScriptingProperties>
```

# Appendix C: Wavelink Contact Information

If you have comments or questions regarding this product, please contact Wavelink Customer Service via e-mail or telephone.

**Email:** customerservice@wavelink.com

**Phone:** 1-888-699-WAVE (9283)

# Index

## P



## R



## S



## T



## V



## W