

Creating AOS Configuration Templates for n-Command MSP

This reference guide describes the creation of ADTRAN Operating System (AOS) configuration templates used in conjunction with n-Command Managed Service Provider (MSP). AOS configuration templates can be customized using a text editor of your choosing to generate AOS configuration settings. Once generated, the templates can be used to push the configuration to AOS products managed by n-Command MSP. This guide provides an overview of how the AOS configuration template is formatted, specifics about the element data supported, and an example AOS configuration template. Additional information for uploading and managing the AOS configuration templates on the n-Command MSP server is available from the online help.

The following topics are covered in this guide:

- *Overview AOS Device Configuration Management on page 2*
- *Hardware and Software Requirements and Limitations on page 2*
- *Template Elements on page 2*
- *AOS Configuration Template Example on page 10*
- *Additional Resources on page 11*

Overview AOS Device Configuration Management

As the number of network devices managed by your n-Command MSP server increases, the need for customizing AOS configuration files and applying them to managed AOS devices becomes a time-consuming task. Deploying a large number of devices with similar configurations requires redundant effort which can be resolved using an AOS configuration template. An AOS configuration template provides a streamlined method for customizing configuration files with similar input data and applying the configuration settings to specific AOS devices.

Using a text editor to compose the AOS configuration template allows you to create the template with the information required for your specific network design. Elements within the AOS configuration template allow small changes to customize the configuration for a particular scenario. For example, entering the hostname or selecting the time zone from a drop-down menu. The specific template elements are discussed in this reference guide to provide you with the parameters necessary for composing your template.

Once composed, the template is uploaded to n-Command MSP through the graphical user interface (GUI). It is made available to the installer to enter the customized configuration parameters. The template form input is validated as the requested information is entered through the n-command MSP GUI. Upon validation, the AOS configuration settings are generated and pushed out to the applicable AOS devices on the network.



AOS configuration and command line syntax is not validated by n-Command MSP. Proper use of AOS commands and syntax is the responsibility of the template creator. The only validation that occurs against the AOS configuration template is the validation of the input which the installer enters when using the configuration template.

Hardware and Software Requirements and Limitations

The AOS configuration template can be initiated from the n-Command MSP GUI. This tool is available in n-Command MSP servers running software version 6.1.5 or later.

The AOS configuration templates must have the **.txt** extension. The content of the file must conform to the XML version 1.0 specification and use UTF-8 encoding.

Template Elements

The AOS configuration template consists of the **root**, **form**, and **config** element data, and allows an optional **filters** element. Each element, and the data it contains, are explained in detail in the following sections.

Example configuration templates are provided in [AOS Configuration Template Example on page 10](#).

Root Element Data

The **root** element is defined first and contains all the other elements. It is identified by **aos-config-template** in the template file. The other elements defined within the **aos-config-template** element, which begins with `<aos-config-template>` and ends with `</aos-config-template>`, consist of an optional **filters** element, one or more **form** elements, and a **config** element.

The **filters** element contains filter tags to limit the pool of devices to which to apply the configuration template. The **form** element contains character data (CDATA) that defines an input form used to create an AOS configuration. The **config** element contains CDATA that defines the AOS configuration with reference(s) to form input. An optional version attribute can be defined for the **aos-config-template** element. This attribute specifies the template version for processing (only version 1.0 is currently implemented). The default value is the latest version.

For example, the **root** element is identified in template as follows:

```
<aos-config-template>
</aos-config-template>
```

Filters Element Data

The **filters** element is optional and contains one or more filter tags. The tags provide a device property to limit the number of devices on which the configuration template may be used.

The basic definition of a filter tag is as follows:

```
<filter field="FIELD" type="COMPARE-TYPE" value="VALUE" />
```

where **FIELD** indicates the device property to filter (either **devicegroup**, **devicetype**, or **firmwareversion**), **COMPARE-TYPE** is the type of comparison to perform (either **is** or **is not**), and **VALUE** is the comparison value.



When using multiple filter statements, an implicit logical AND joins the statements. There is not an operator to join multiple statements with a logical OR.

The following example filters devices where the device type is **7100** and the firmware version is not **R10.7.0E**:

```
<filters>
  <filter field="devicetype" type="is" value="7100"/>
  <filter field="firmwareversion" type="is not" value="R10.7.0.E"/>
</filters>
```

Each filter field corresponds to a device property which has a slightly different heading in the n-Command MSP graphical user interface (GUI). The following table provides a cross reference for each device property, its corresponding filter field, and where it is located in the GUI.

Device Property	Filter Field	Location in n-Command MSP GUI
Device Group	devicegroup	Under Firmware
Device Type	devicetype	Under Devices
Running F/W Version	firmwareversion	Under Devices
<i>Device Property is also the heading name in n-Command MSP GUI.</i>		

Form Element Data

Each text line in the **form** element data is equivalent to a line in the form displayed by n-Command MSP. User input fields have a special notation very similar to Hypertext Markup Language (HTML) form input field elements. This notation cannot be broken up over multiple lines. The following user input fields are supported:

- Drop-down text selection
- Radio button
- Text input

An additional form field notation is available for constant values that cannot be modified or seen by a user (i.e., hidden values). All of the supported **form** fields are described in the following section.

Drop-Down Text Selection

The basic definition of the drop-down text selection input field is as follows:

```
<select id="FIELD-ID"><option>CHOICE #1</option><option>CHOICE #2</option></select>
```

where **FIELD-ID** is the unique (case-insensitive) identification for the field and **CHOICE #1** and **CHOICE #2** are the selectable text options.

Multiple option elements can be added. The following attributes are valid for the **select** element:

Select Attribute	Description
id	<p>Required. Specifies the unique field identification and must conform to the following format:</p> <ul style="list-style-type: none"> • at least two characters in length • begin with an alphabetic character • only contain alphanumeric, dash (-), or underscore (_) characters • end with an alphanumeric character • cannot match (case-insensitive) the identification of another element in the same form

The following attributes are valid for the **option** element:

Option Attribute	Description
selected	Optional. Indicates whether the option is initially selected. The default value is true for the first option without this attribute and with no other options defining a true attribute value. The default value for all other options is false .

Radio Button

The basic definition of the **radio** button field is as follows:

```
<input id="FIELD-ID" name="GROUP-ID" type="radio">TEXT</input>
```

where **FIELD-ID** is the unique (case-insensitive) identification for the field, **GROUP-ID** is the unique (case-insensitive) identification for the associated group, and **TEXT** is the field value.

The following attributes are valid for the radio input element:

Radio Input Attribute	Description
checked	Optional. Indicates whether the field is initially checked. The default value is true for the first radio input associated with a name attribute and without a checked attribute as long as no other radio inputs associated with the name have a true checked value. The default value for all other radio inputs is false .
id	Required. Specifies the unique (case-insensitive) field identification. This identification must conform to the following format: <ul style="list-style-type: none"> • at least two characters in length • begin with an alphabetic character • only contain alphanumeric, dash (-), or underscore (_) characters • end with an alphanumeric character • cannot match (case-insensitive) the identification of another element in the same form
name	Required. Specifies the group identification to which this field belongs. This identification must conform to the following format: <ul style="list-style-type: none"> • be at least two characters in length • begin with an alphabetic character • only contain alphanumeric, dash (-), or underscore (_) characters • end with an alphanumeric character
value	Optional. Specifies the initial value of the field. The default value is "". This attribute cannot be defined if the element has content defined (i.e., text between the element tags).

Text Input

The basic definition of the radio button field is as follows:

```
<input id="FIELD-ID" type="text">VALUE</input>
```

where **FIELD-ID** is the unique (case-insensitive) identification for the field and **VALUE** is the initial field value.

The following attributes are valid for the text input element:

Text Input Attribute	Description
allowblank	Optional. Indicates whether blank is a valid value for the field. The default value is false .
id	Required. Specifies the unique (case-insensitive) field identification. This identification must conform to the following format: <ul style="list-style-type: none"> at least two characters in length begin with an alphabetic character only contain alphanumeric, dash (-), or underscore (_) characters end with an alphanumeric character cannot match (case-insensitive) the identification of another element in the same form
maxlength	Optional. Specifies the maximum number of characters the field can contain. The default value is 0 (no maximum).
regexp	Optional. Specifies a regular expression pattern (without the enclosing //) which the field value must match. The pattern should match Adobe ActionScript 3.0 regular expression syntax (without flags). Refer to http://help.adobe.com/en_US/ActionScript/3.0_ProgrammingAS3/WS5b3ccc516d4fbf351e63e3d118a9b90204-7ea9.html for more information. The default value is "" (no pattern to match).
regexpflags	Optional. Specifies the regular expression flags for the specified pattern. The default value is "" (no flags). Refer to http://help.adobe.com/en_US/ActionScript/3.0_ProgrammingAS3/WS5b3ccc516d4fbf351e63e3d118a9b90204-7ea9.html for more information.
restrict	Optional. Specifies the characters that the field can accept. An empty string "" will restrict all input. Ranges for digits and letters can be specified using a hyphen (-) character. Including an uppercase letter in the value will result in the automatic capitalization of the lowercase letter (if entered by the user) and vice-versa. Starting the value with a caret (^) character prevents all the subsequent characters from being accepted. A backslash (\) character must precede the characters -, ^, and \ in order to include them in the list of characters. The default value is no restriction.
trim	Optional. Indicates whether surrounding whitespace (such as horizontal spaces, tabs, etc.) should be trimmed from the field value. The default value is true .
value	Optional. Specifies the initial value of the field. The default value is "". This attribute cannot be defined if the element has content defined (i.e., text between the element tags).

The following example displays a form requesting text input using the **regexp** (regular expression pattern) attribute:

```
<form><![CDATA[
```

```
CONFIGURATION FORM HEADING...
```

```
Please enter MAC address:<input id="mac" type="text" maxlength="17" restrict=":A-F0-9\-"
  regexp="^[A-F0-9]{2}[:\-]{5}[A-F0-9]{2}$"/>
]]></form>
```

Constant Value

The basic definition of the constant value field is as follows:

```
<input id="FIELD-ID" type="hidden">VALUE</input>
```

where **FIELD-ID** is the unique (case-insensitive) identification for the field and **VALUE** is the constant field value.

The following attributes are valid for the hidden input element:

Hidden Input Attribute	Description
id	<p>Required. Specifies the unique (case-insensitive) field identification. This identification must conform to the following format:</p> <ul style="list-style-type: none"> • be at least two characters in length • begin with an alphabetic character • only contain alphanumeric, dash (-), or underscore (_) characters • end with an alphanumeric character • cannot match (case-insensitive) the identification of another element in the same form
value	<p>Optional. Specifies the initial value of the field. The default value is "". This attribute cannot be defined if the element has content defined (i.e., text between the element tags).</p>

The following example uses a constant value for the hidden input attribute:

```
<input id="user" type="hidden">default</input>
```

Config Element Data

The **config** element data will appear very familiar to someone with knowledge of AOS configurations. With the exception of references to form field values, the data should match the AOS configuration to be generated by processing a defined form element. References to form field values should be prefixed with a \$ (dollar sign) and optionally surrounded by {} (brackets). (To include an actual \$ character in the configuration, it must be prefixed with a \$, so that \$\$ generates a single \$ in the final configuration.) When a form is processed, the field references are replaced with the user-entered field values.

The simplest reference only includes the field identification or group name following the \$ or within the {}, such as:

\$FIELD-ID or \${FIELD-ID}

The surrounding {} are required if the reference is not surrounded by whitespace. For example, **\$FIELD-ID.com** references an invalid field **FIELD-ID.com** while, **\${FIELD-ID}.com** references field **FIELD-ID** followed by the text **.com**. Radio button input can only be referenced by the associated group name, not by field identification.

Functions

Functions that manipulate the referenced field value can also be called from within the **config** element. The function invocation is proceeded by a \$ and surrounded by {} like a normal field reference. The arguments to the function are specified within parentheses and separated by commas.

The following functions are available for use:

Function	Parameter	Entry Type	Description
lowercase	value	Quoted text or a function call	Converts value to lowercase characters.
replace	value	Quoted text or a function call	Replaces pattern (regular expression or text) in value with repl .
	pattern	Regular expression or text	
	repl	Quoted text	
substr	value	Quoted text or a function call	Returns len characters in value from position startIndex (the first position is 0).
	startIndex	Integer value	
	len*	Integer value	
trim	value	Quoted text or a function call	Removes surrounding white space from value .
uppercase	value	Quoted text or a function call	Converts value to uppercase characters.
<i>*Optional parameter.</i>			

- The **value** parameter is entered as quoted text for field identification or as another function invocation.
- The **pattern** parameter is entered as a regular expression (i.e., /[0-9]+/g) or quoted text (either single or double quotes).
- The **repl** parameter is entered as quoted text (either single or double quotes).
- The **startIndex** and **len** parameter are entered as integer values.

The following table displays the results of some example function calls for the input string “**Press Button 01 Now**” and **x** as the variable name:

Function	Result
lowercase(x)	"press button 01 now "
replace(x, /test/, "")	"Press Button 01 Now "
replace(x, /[A-Za-z]/, "")	"ress Button 01 Now "
replace(x, /o/g, "*")	"Press Butt*n 01 N*w "
replace(x, /[A-Za-z]/g, "")	" 01 "
replace(x, "o", "#")	"Press Butt#n 01 Now "
replace(x, "Now", "")	"Press Button 01 "
substr(x, 0)	"Press Button 01 Now "
substr(x, 6)	"Button 01 Now "
substr(x, 20)	""
substr(x, 0, 5)	"Press"
substr(x, 13, 2)	"01"
substr(x, 13, 20)	"01 Now "
trim(x)	"Press Button 01 Now"
uppercase(x)	"PRESS BUTTON 01 NOW "

In the following example, the **config** element uses the replace function:

```
<config><![CDATA[
hostname ${replace(mac, /[:\-]/g, "")}
!
enable password $enable-password
!
username "${user}" password "${password}"
]]></config>
```

AOS Configuration Template Example

The following is an example of an AOS configuration file template that uses only the required data elements; root, form, and config element data:

```
<?xml version="1.0" encoding="utf-8"?>
<aos-config-template>
<form id="user-defined-auth"><![CDATA[

CONFIGURATION FORM HEADING...
Please enter MAC address:<input id="mac" type="text" maxlength="17"
  regexp="^[A-F0-9]{2}[:\-]{5}[A-F0-9]{2}$"/>

AUTHENTICATION...
Please enter user:<input id="user" type="text" maxlength="15" restrict="A-Z"/>
Please enter password:<input id="password" type="text" maxlength="15" restrict="A-Za-z0-9!$#\_"/>
<input id="enable-password" type="hidden">en_Passw0rd</input>

]]></form>
<form id="mac-only"><![CDATA[

CONFIGURATION FORM HEADING...
Please enter MAC address:<input id="mac" type="text" maxlength="17" restrict=":A-F0-9\-"
  regexp="^[A-F0-9]{2}[:\-]{5}[A-F0-9]{2}$"/>
<input id="user" type="hidden">default</input>
<input id="password" type="hidden">Passw0rd</input>
<input id="enable-password" type="hidden">en_Passw0rd</input>

]]></form>
<config><![CDATA[

hostname ${replace(mac, /[:\-]/g, "")}
!
enable password $enable-password
!
username "${user}" password "${password}"

]]></config>
</aos-config-template>
```

The following is an example of an AOS configuration file template that uses all data elements, including the optional filters element:

```
<?xml version="1.0" encoding="utf-8"?>
<aos-config-template version="1.00">
<filters>
<filter field="devicetype" type="is" value="7100"/>
<filter field="firmwareversion" type="is not" value="R10.7.0.E"/>
</filters>
<form><![CDATA[
Please enter hostname:<input id="hostname" type="text" maxlength="20" trim="true"/>
]]></form>
<config><![CDATA[
hostname ${hostname}
]]></config>
</aos-config-template>
```

Additional Resources

Additional resources are available to assist in using n-Command MSP server. The documents listed below are available online at ADTRAN's Support Forum at <https://supportforums.adtran.com>.

- *Configuring Auto-Link in AOS for n-Command MSP*
- *n-Command MSP Quick Start Guide*
- *n-Command MSP Online Help*