

# **Enterprise Manager™: Working with Templates and Changesets**

Version 3.1.1





# Table of Contents

**Legal Notices.....5**

**Acknowledgments.....7**

  

**Chapter 1: Introduction to Templates and Changsets.....11**

    Overview: Templates and changesets.....12

        Network objects and dependencies for templates and changesets.....12

  

**Chapter 2: Template Management.....15**

    About templates.....16

        System-provided templates.....16

    About managing templates.....16

        Creating a template based on an existing device's configuration.....16

        Creating a template from an existing template.....18

        Creating a template using text.....18

        Importing a template from DevCentral.....19

        Exporting an ASM security policy.....20

        Reviewing template variable text.....20

        Publishing a template.....22



# Legal Notices

---

## Publication Date

This document was published on May 15, 2013.

## Publication Number

MAN-0427-01

## Copyright

Copyright © 2012-2013, F5 Networks, Inc. All rights reserved.

F5 Networks, Inc. (F5) believes the information it furnishes to be accurate and reliable. However, F5 assumes no responsibility for the use of this information, nor any infringement of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent, copyright, or other intellectual property right of F5 except as specifically described by applicable user licenses. F5 reserves the right to change specifications at any time without notice.

## Trademarks

Access Policy Manager, Advanced Client Authentication, Advanced Routing, APM, Application Security Manager, ARX, AskF5, ASM, BIG-IP, BIG-IQ, Cloud Extender, CloudFucious, Cloud Manager, Clustered Multiprocessing, CMP, COHESION, Data Manager, DevCentral, DevCentral [DESIGN], DNS Express, DSC, DSI, Edge Client, Edge Gateway, Edge Portal, ELEVATE, EM, Enterprise Manager, ENGAGE, F5, F5 [DESIGN], F5 Management Pack, F5 Networks, F5 World, Fast Application Proxy, Fast Cache, FirePass, Global Traffic Manager, GTM, GUARDIAN, IBR, Intelligent Browser Referencing, Intelligent Compression, IPv6 Gateway, iApps, iControl, iHealth, iQuery, iRules, iRules OnDemand, iSession, L7 Rate Shaping, LC, Link Controller, Local Traffic Manager, LTM, Message Security Manager, MSM, OneConnect, OpenBloX, OpenBloX [DESIGN], Packet Velocity, Policy Enforcement Manager, PEM, Protocol Security Manager, PSM, Real Traffic Policy Builder, Rosetta Diameter Gateway, Scale<sup>N</sup>, Signaling Delivery Controller, SDC, SSL Acceleration, StrongBox, SuperVIP, SYN Check, TCP Express, TDR, TMOS, Traffic Management Operating System, Traffix Diameter Load Balancer, Traffix Systems, Traffix Systems (DESIGN), Transparent Data Reduction, UNITY, VAULT, VIPRION, vCMP, virtual Clustered Multiprocessing, WA, WAN Optimization Manager, WebAccelerator, WOM, and ZoneRunner, are trademarks or service marks of F5 Networks, Inc., in the U.S. and other countries, and may not be used without F5's express written consent.

All other product and company names herein may be trademarks of their respective owners.

## Export Regulation Notice

This product may include cryptographic software. Under the Export Administration Act, the United States government may consider it a criminal offense to export this product from the United States.

## RF Interference Warning

This is a Class A product. In a domestic environment this product may cause radio interference, in which case the user may be required to take adequate measures.

## FCC Compliance

This equipment has been tested and found to comply with the limits for a Class A digital device pursuant to Part 15 of FCC rules. These limits are designed to provide reasonable protection against harmful

interference when the equipment is operated in a commercial environment. This unit generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference, in which case the user, at his own expense, will be required to take whatever measures may be required to correct the interference.

Any modifications to this device, unless expressly approved by the manufacturer, can void the user's authority to operate this equipment under part 15 of the FCC rules.

### **Canadian Regulatory Compliance**

This Class A digital apparatus complies with Canadian ICES-003.

### **Standards Compliance**

This product conforms to the IEC, European Union, ANSI/UL and Canadian CSA standards applicable to Information Technology products at the time of manufacture.

# Acknowledgments

---

This product includes software developed by Bill Paul.

This product includes software developed by Jonathan Stone.

This product includes software developed by Manuel Bouyer.

This product includes software developed by Paul Richards.

This product includes software developed by the NetBSD Foundation, Inc. and its contributors.

This product includes software developed by the Politecnico di Torino, and its contributors.

This product includes software developed by the Swedish Institute of Computer Science and its contributors.

This product includes software developed by the University of California, Berkeley and its contributors.

This product includes software developed by the Computer Systems Engineering Group at the Lawrence Berkeley Laboratory.

This product includes software developed by Christopher G. Demetriou for the NetBSD Project.

This product includes software developed by Adam Glass.

This product includes software developed by Christian E. Hopps.

This product includes software developed by Dean Huxley.

This product includes software developed by John Kohl.

This product includes software developed by Paul Kranenburg.

This product includes software developed by Terrence R. Lambert.

This product includes software developed by Philip A. Nelson.

This product includes software developed by Herb Peyerl.

This product includes software developed by Jochen Pohl for the NetBSD Project.

This product includes software developed by Chris Provenzano.

This product includes software developed by Theo de Raadt.

This product includes software developed by David Muir Sharnoff.

This product includes software developed by SigmaSoft, Th. Lockert.

This product includes software developed for the NetBSD Project by Jason R. Thorpe.

This product includes software developed by Jason R. Thorpe for And Communications, <http://www.and.com>.

This product includes software developed for the NetBSD Project by Frank Van der Linden.

This product includes software developed for the NetBSD Project by John M. Vinopal.

This product includes software developed by Christos Zoulas.

This product includes software developed by the University of Vermont and State Agricultural College and Garrett A. Wollman.

This product includes software developed by Balazs Scheidler ([bazsi@balabit.hu](mailto:bazsi@balabit.hu)), which is protected under the GNU Public License.

This product includes software developed by Niels Mueller ([nisse@lysator.liu.se](mailto:nisse@lysator.liu.se)), which is protected under the GNU Public License.

In the following statement, *This software* refers to the Mitsumi CD-ROM driver: This software was developed by Holger Veit and Brian Moore for use with 386BSD and similar operating systems. *Similar operating systems* includes mainly non-profit oriented systems for research and education, including but not restricted to NetBSD, FreeBSD, Mach (by CMU).

This product includes software developed by the Apache Group for use in the Apache HTTP server project (<http://www.apache.org/>).

This product includes software licensed from Richard H. Porter under the GNU Library General Public License (© 1998, Red Hat Software), [www.gnu.org/copyleft/lgpl.html](http://www.gnu.org/copyleft/lgpl.html).

This product includes the standard version of Perl software licensed under the Perl Artistic License (© 1997, 1998 Tom Christiansen and Nathan Torkington). All rights reserved. You may find the most current standard version of Perl at <http://www.perl.com>.

This product includes software developed by Jared Minch.

This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (<http://www.openssl.org/>).

This product includes cryptographic software written by Eric Young ([ey@cryptsoft.com](mailto:ey@cryptsoft.com)).

This product contains software based on oprofile, which is protected under the GNU Public License.

This product includes RRDtool software developed by Tobi Oetiker (<http://www.rrdtool.com/index.html>) and licensed under the GNU General Public License.

This product contains software licensed from Dr. Brian Gladman under the GNU General Public License (GPL).

This product includes software developed by the Apache Software Foundation (<http://www.apache.org/>).

This product includes Hypersonic SQL.

This product contains software developed by the Regents of the University of California, Sun Microsystems, Inc., Scriptics Corporation, and others.

This product includes software developed by the Internet Software Consortium.

This product includes software developed by Nominum, Inc. (<http://www.nominum.com>).

This product contains software developed by Broadcom Corporation, which is protected under the GNU Public License.

This product contains software developed by MaxMind LLC, and is protected under the GNU Lesser General Public License, as published by the Free Software Foundation.

This product includes Intel QuickAssist kernel module, library, and headers software licensed under the GNU General Public License (GPL).

This product includes software developed by members of the CentOS Project under the GNU Public License, copyright ©2004-2011 by the CentOS Project.

This product includes software developed by members of the OpenJDK Project under the GNU Public License Version 2, copyright ©2012 by Oracle Corporation.

This product includes software developed by The VMWare Guest Components Team under the GNU Public License Version 2, copyright ©1999-2011 by VMWare, Inc.

This product includes software developed by The Netty Project under the Apache Public License Version 2, copyright ©2008-2012 by The Netty Project.

This product includes software developed by Stephen Colebourne under the Apache Public License Version 2, copyright ©2001-2011 Joda.org.

This product includes software developed by the GlassFish Community under the GNU Public License Version 2 with classpath exception, copyright ©2012 Oracle Corporation.



This product includes software developed by the Mort Bay Consulting under the Apache Public License Version 2, copyright ©1995-2012 Mort Bay Consulting.

This product contains software developed by members of the Jackson Project under the GNU Lesser General Public License Version 2.1, ©2007 – 2012 by the Jackson Project”.

This product contains software developed by QOS.ch under the MIT License, ©2004 – 2011 by QOS.ch.

This product includes software licensed from Gerald Combs (gerald@wireshark.org) under the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or any later version. Copyright ©1998 Gerald Combs.

This product includes software developed by Thomas Williams and Colin Kelley. Copyright ©1986 - 1993, 1998, 2004, 2007

Permission to use, copy, and distribute this software and its documentation for any purpose with or without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation. Permission to modify the software is granted, but not the right to distribute the complete modified source code. Modifications are to be distributed as patches to the released version. Permission to distribute binaries produced by compiling modified sources is granted, provided you

1. distribute the corresponding source modifications from the released version in the form of a patch file along with the binaries,
2. add special version identification to distinguish your version in addition to the base release version number,
3. provide your name and address as the primary contact for the support of your modified version, and
4. retain our contact information in regard to use of the base software.

Permission to distribute the released version of the source code along with corresponding source modifications in the form of a patch file is granted with same provisions 2 through 4 for binary distributions. This software is provided "as is" without express or implied warranty to the extent permitted by applicable law.

This product contains software developed by Google, Inc. Copyright ©2011 Google, Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

This software incorporates JFreeChart, ©2000-2007 by Object Refinery Limited and Contributors, which is protected under the GNU Lesser General Public License (LGPL).

This product contains software developed by the Mojarra project. Source code for the Mojarra software may be obtained at <https://jaserverfaces.dev.java.net/>.



---

# Chapter 1

---

## Introduction to Templates and Changsets

---

- *Overview: Templates and changesets*
-

### Overview: Templates and changesets

---

There are two versatile options you can use to manage device configurations: templates and changesets.

A *template* is a configuration model on which you base new configurations. When you create a new template, you change only those variables that are unique to the target device, so the template is customized for that device. For example, if you manage devices in multiple data centers that reside in multiple time zones, you can create a template to set the time zone variable for a specific device.

Conversely, a *changeset* is a collection of specific user-defined configuration data that you create and save from any managed device in your network, to distribute to other managed devices. For example, when you initially configure a device, you typically specify certain profiles, monitors, and iRules®. To configure systems individually, you must keep track of each setting, and manually enter those values for every new device that you add to the network. However, using changesets, you can deploy the same profiles, monitors, and iRules configurations from one device to as many devices as needed.

You can also use these options together to manage device configurations. Because templates offer you the ability to set variables for different devices in the network, you can apply them in conjunction with changesets to help manage common network configuration tasks.

The flexibility of these options helps you to efficiently manage these tasks:

- Deploy common configurations to new devices
- Manage configuration continuity across devices
- Roll out new applications to multiple devices
- Audit configuration changes
- Manage network dependencies for devices

### Network objects and dependencies for templates and changesets

Before you use can successfully use templates and changesets, you must understand and honor the network object's dependencies on the target system. A *dependency* is additional network object data or resources required for the primary object to function correctly.

For example, when you configure a virtual server, you usually need to also define dependent objects or resources, such as pools, nodes, or profiles. If you were to manually copy configuration files from one system to another, you would need to know each of the dependencies for every object or system setting that you plan to copy. However, when you use templates and changesets, these dependencies are automatically managed.

The presence of dependencies adds a level of complexity to the process of storing and copying device configurations. If you were to manually copy configuration files from one system to another, you would need to know each of the dependencies for every object or system setting that you plan to copy. However, when you use templates and changesets, the system automatically manages these dependencies.

### Network object elements for templates and changesets

You typically create a template or changeset using the Template or Changeset wizard. These wizards prompt you for the information required and automatically generate the proper network object syntax, which consists of these elements.

Element	Example syntax	Description
Object class	#F5[Local Traffic / Pool]	An <i>object class</i> is the category in which a network object belongs. You can include any type of available object class in a template or changeset. The available object classes vary depending on the device's licensed features. Every network object in a template or changeset must have a class directive. This example syntax informs the system that the object configuration that follows the text refers to local traffic objects, specifically pools. When you deploy this template or changeset, the system uses the <code>bigpipe</code> utility to add this configuration information as a pool configuration on the target device.
System class	#F5[System / DNS]	A <i>system class</i> is associated with a specific system setting. All system settings contained in a configuration must specify a system class directive. This example syntax informs the system that the configuration data that follows the text refers to system objects, specifically DNS settings. When you deploy this template or changeset, the system uses the <code>bigpipe</code> utility, or other utilities, to add the DNS settings to the appropriate configuration file on the target device.
Unnamed objects	#F5[Local Traffic / SSL Certificate / sample.crt] #F5[Local Traffic / SSL Key / sample.key]	Certain objects that you include in a changeset or template require additional information. For example, the SSL certificate system class requires that you include the SSL certificates and SSL keys, and specify the name of the target files. Enterprise Manager™ copies the object data with these directives to the <code>sample.crt</code> and <code>sample.key</code> files on the target device, respectively, as in this example.
Administrative partitions	#F5[\$target_partition\$]	If a managed device supports administrative partitions, the system includes object partition information in the template or changeset text. If you include an object targeted to a specific partition, the system precedes the object class directive with this text, where <i>target_partition</i> is the name of the partition on the target device. This text directs the system to generate a <code>shell write partition bigpipe</code> command using the partition name you specified when the system verifies or deploys the template or changeset.

Element	Example syntax	Description
Object settings	<pre>#F5[Local Traffic / Virtual Server] shell write partition Common virtual MyVIP { pool MyPool destination 10.20.10.10:http ip protocol tcp }</pre>	<p>If a managed device supports administrative partitions, the system includes object partition information in the template or changeset text. If you include an object targeted to a specific partition, the system precedes the object class directive with this text, where <code>target_partition</code> is the name of the partition on the target device. This text directs the system to generate a <code>shell write partition bigpipe</code> command using the partition name you specified when the system verifies or deploys the template or changeset. When you deploy the template or changeset as defined for this example, the target devices you selected contain the local traffic objects MyVIP and MyPool.</p>

---

# Chapter 2

---

## Template Management

---

- *About templates*
  - *About managing templates*
-

## About templates

A *template* is as a model of a configuration on which you base a new configuration for another target device. *Template variables* are unique values, or settings, that are specific to a managed device such as virtual server names, node addresses, port information, and so forth. Some network objects (for example, nodes or pools) automatically generate variable information when you add them to a template. Other network objects (such as system settings) require that you manually add variable information, like the variable name and default value.

## System-provided templates

The system ships with several templates that you can use for common configuration tasks.

System-provided template	Use to
<code>ltm_create_simple_http_vip_and_pool</code>	Create a basic HTTP virtual server and pool.
<code>ltm_pool_member_disable</code>	Disable a local traffic pool member (allows established sessions).
<code>ltm_pool_member_enable</code>	Enable a local traffic pool member.
<code>ltm_pool_member_down</code>	Set a local traffic pool member's status to down (allowing no new connections).
<code>ltm_node_enable</code>	Enable a local traffic server address (all pools).
<code>ltm_node_disable</code>	Disable a local traffic server address (for all pools, only allowing established connections).
<code>ltm_node_down</code>	Set a local traffic server address to down (for all pools, allowing no new connections).

## About managing templates

There are three different options for creating a new template: using an existing device as a model, using an existing template as a model, or creating the template by adding pre-set object class paths to populate the template text field. You can also import (as well as export) templates on DevCentral™.

After you create a template, you can review and modify the template variable text as required. When you want to make the template available for others to use, you can publish it.

## Creating a template based on an existing device's configuration

Before you use templates, be sure that you understand how to honor the network object's dependencies on the target system.

You can use an existing device configuration as a model for a new template.

1. On the Main tab, click **Enterprise Management > Tasks > Task List > New Task**.
2. For the **Configurations** setting, select **Create Template**, and click the **Next** button.



3. In the **Name** and **Description** fields, type a name and description for this template.
4. From the **Source** list, select **Device**.
5. From the **Device** and **Partitions** lists, select the device and partition from which you want to copy available network object elements.
6. Click the **Next** button.  
The Object Type Selection screen opens.
7. For the **Object Type List** setting, click the name of an object in the **Available** list and click the Move (<<) button to move it to the **Selected** list. Repeat this step to add additional objects.
8. Click the **button**.  
The Template Variable Properties screen opens.
9. To specify the default value for a variable, in the **Default Value** field, type a value.
10. If you want users to be able to edit the variable, select the **Editable** check box.
11. If you want users to be able to see the variable, but not edit it, select only the **Visible** check box.
12. Click the **Finished** button.  
The template you created displays in the template list.
13. Click the **Next** button.  
The Template Summary screen opens.
14. From the **Dependency Handling** list, select an option.

<b>Option</b>	<b>Description</b>
<b>Include resource objects</b>	When you select this option, the system automatically includes dependent objects for the selected object classes.
<b>Skip resource objects</b>	When you select this option, you must manually type the dependent objects for the selected object classes on the target device, when you deploy the changeset based on this template.
15. To view object details, click the name of a **User Selected Object**.  
Details about the selected object display below the list in several fields, some of which you can edit.
16. To view any associated resource object details, click the name of a **Resource Object**.  
Details about the selected object display below the list in several fields, some of which you can edit.
17. When you are finished viewing or modifying objects, click **Next**.  
The Template Properties screen opens where you can review and modify template properties.
18. If you want the system to verify the staged changeset based on this template prior to deployment, select the check box next to **Require verification of staged changesets**.  
The system uses the `bigpipe verify` command to verify configuration changes prior to deployment.
19. If you want to retain the changeset based on this template for future deployment, select the check box next to **Allow staged changesets to persist**.  
This changeset remains in the staged changeset list until you delete it.
20. To add an additional object type, select a type from the **Select Object Type** list and click the **Add Type** button.
21. To search for an object type and replace it with another, type the current and replacement object in the **Search for** and **Replace with** fields, and click the **Search and Replace** button.
22. When you are finished reviewing and modifying the template, click **Next**.  
The Template Variable Properties screen opens.
23. To specify the default value for a variable, in the **Default Value** field for a variable, type a value.
24. To specify a description for the variable, in the **Description** field for a variable, type a description.  
This description appears when you stage a changeset based on this template, and provides an essential description of the variable for the user who stages the changeset.
25. To allow others to modify values for this variable, select the **Editable** check box.

26. To allow others to view this variable, select the **Visible** check box.
27. To modify the allowed values for this variable, click the **Edit Allowed Values** button.  
The Allowed Values screen opens where you can modify allowed values.
28. Click the **Finished** button when you are done making changes.  
The Template List screen opens, displaying the new template you created.

### Creating a template from an existing template

Before you use templates, be sure that you understand how to honor the network object's dependencies on the target system.

You can use an existing template as a model for a new template.

1. On the Main tab, click **Enterprise Management > Tasks > Task List > New Task**.
2. For the **Configurations** setting, select **Create Template**, and click the **Next** button.
3. In the **Name** and **Description** fields, type a name and description for this template.
4. From the **Source** list, select **Existing Template**.  
The screen refreshes, displaying existing templates.
5. Select the existing template on which you want to base this new template, and click the **Next** button.  
The Template Properties screen opens.
6. If you want the system to verify the staged changeset based on this template prior to deployment, select the check box next to **Require verification of staged changesets**.  
The system uses the `bigpipe verify` command to verify configuration changes prior to deployment.
7. If you want to retain the changeset based on this template for future deployment, select the check box next to **Allow staged changesets to persist**.  
This changeset remains in the staged changeset list until you delete it.
8. To add an additional object type, select a type from the **Select Object Type** list and click the **Add Type** button.
9. To search for an object type and replace it with another, type the current and replacement object in the **Search for** and **Replace with** fields, and click the **Search and Replace** button.
10. When you are finished reviewing and modifying the template, click **Next**.  
The Template Variable Properties screen opens.
11. To specify the default value for a variable, in the **Default Value** field for a variable, type a value.
12. To specify a description for the variable, in the **Description** field for a variable, type a description.  
This description appears when you stage a changeset based on this template, and provides an essential description of the variable for the user who stages the changeset.
13. To allow others to modify values for this variable, select the **Editable** check box.
14. To allow others to view this variable, select the **Visible** check box.
15. To modify the allowed values for this variable, click the **Edit Allowed Values** button.  
The Allowed Values screen opens where you can modify allowed values.
16. Click the **Finished** button when you are done making changes.  
The Template List screen opens, displaying the new template you created.

### Creating a template using text

Before you use templates, be sure that you understand how to honor the network object's dependencies on the target system.

You have the option to create a template by adding pre-set object class path text directly into the text field.

1. On the Main tab, click **Enterprise Management > Tasks > Task List > New Task**.
2. For the **Configurations** setting, select **Create Template**, and click the **Next** button.
3. In the **Name** and **Description** fields, type a name and description for this template.
4. From the **Source** list, select **Text** and click **Next**.  
The Template Properties screen opens.
5. If you want the system to verify the staged changeset based on this template prior to deployment, select the check box next to **Require verification of staged changesets**.  
The system uses the `bigpipe verify` command to verify configuration changes prior to deployment.
6. If you want to retain the changeset based on this template for future deployment, select the check box next to **Allow staged changesets to persist**.  
This changeset remains in the staged changeset list until you delete it.
7. To add an additional object type, select a type from the **Select Object Type** list and click the **Add Type** button.
8. To search for an object type and replace it with another, type the current and replacement object in the **Search for** and **Replace with** fields, and click the **Search and Replace** button.
9. When you are finished reviewing and modifying the template, click **Next**.  
The Template Variable Properties screen opens.
10. To specify the default value for a variable, in the **Default Value** field for a variable, type a value.
11. To specify a description for the variable, in the **Description** field for a variable, type a description.  
This description appears when you stage a changeset based on this template, and provides an essential description of the variable for the user who stages the changeset.
12. To allow others to modify values for this variable, select the **Editable** check box.
13. To allow others to view this variable, select the **Visible** check box.
14. To modify the allowed values for this variable, click the **Edit Allowed Values** button.  
The Allowed Values screen opens where you can modify allowed values.
15. Click the **Finished** button when you are done making changes.  
The Template List screen opens, displaying the new template you created.

## Importing a template from DevCentral

To access the DevCentral™ site, you must have a user name and password.

In addition to using templates from one of your devices, you can also use shared templates from the F5 developer community, DevCentral.

---

**Note:** DevCentral is an online community featuring tools, technology, and collaboration for F5 products. After registering for free, you can access resources such as discussion forums, documentation wikis, sample applications, and templates.

---

1. Log on to the DevCentral site, <http://devcentral.f5.com>.
2. Navigate to the Advanced Design & Config wiki.
3. Locate the Enterprise Manager Application Deployment Templates section.
4. Click the name of the template you want to view.
5. Place the cursor in the right corner of the template text and select the view source icon.  
A pop-up screen opens displaying the formatted template text.
6. Select and copy all of the template text, and navigate back to the Enterprise Manager™ system.

7. On the Main tab, click **Enterprise Management > Tasks > Task List > New Task**.
8. For the **Configurations** setting, select **Create Template**, and click the **Next** button.
9. In the **Name** and **Description** fields, type a name and description for this template.
10. From the **Source** list, select **Text** and click **Next**.  
The Template Properties screen opens.
11. If you want the system to verify the staged changeset based on this template prior to deployment, select the check box next to **Require verification of staged changesets**.  
The system uses the `bigpipe verify` command to verify configuration changes prior to deployment.
12. If you want to retain the changeset based on this template for future deployment, select the check box next to **Allow staged changesets to persist**.  
This changeset remains in the staged changeset list until you delete it.
13. In the **Text** setting text box, paste the template text that you copied from the DevCentral site.
14. Review and modify the template text as required.
15. When you are finished reviewing and modifying the template, click **Next**.  
The Template Variable Properties screen opens.
16. To allow others to modify values for this variable, select the **Editable** check box.
17. To allow others to view this variable, select the **Visible** check box.
18. Click the **Finished** button when you are done making changes.  
The Template List screen opens, displaying the new template you created.

## Exporting an ASM security policy

You must import a security policy from a managed BIG-IP® Application Security Manager™ device to Enterprise Manager™, before you can export it.

You can export a security policy from one web application to use it as a baseline for a new web application. You can also export a security policy to archive it on a remote system before upgrading the system software, or to create a backup copy.

---

***Note:** This procedure is only for BIG-IP Application Security Manager devices running version 11.3.0 or later.*

---

1. On the Main tab, click **Security > Application Security > Policies**.
2. Click the select button next to the security policy that you want to export, and click the **Export** button.  
A dialog box opens.
3. Click the **Save** button.
4. Browse to the location that you want to export the security policy to, and click the **Save** button.

The security policy is now available to import to another managed device.

## Reviewing template variable text

When you create a template using the Template Wizard, you are prompted for information, and the system automatically generates the appropriate syntax. After you have created a template, you can review the template variable syntax and add additional variables, as required.

1. On the Main tab, click **Enterprise Management > Configurations > Template List**.

2. Click the name of the template that you want to view.  
The Properties screen for that template opens.
3. View and modify the template as required.

## About template variable syntax

When you create a template using the Template Wizard, you are prompted for information and the system automatically generates the appropriate syntax. After you have created a template, you can review the template variable syntax and add additional variables, as required.

The template variable syntax is `@define <variable_name>` where `<variable_name>` is the name of the variable in the network object. For example, the following entry in the **Text** field of the Template Variable Properties screen disables a node.

```
@define @node_ip
@F5[Local Traffic / Node]
#F5[$target_partition$]
node @node_ip {
    session disable
}
```

**Note:** Although the leading at (@) symbol is not required for variables names, the system uses it to distinguish a variable from static configuration information. This symbol can also help you easily identify variables when you read the configuration text.

The variables in this example template are defined as follows:

Variable	Description
<code>@define</code>	Flags the line as a variable and prompts the system to replace the variable for <code>@node_ip</code> that you specified for the template.
<code>@F5[Local Traffic / Node]</code>	Indicates the target object class and instance.
<code>@f5[\$target_partition\$]</code>	Indicates the partition on the target.
<code>node @node_ip {</code>	Starts the command that disables the node indicated by the variable <code>@node_ip</code> .

## Template variable descriptions

Template variables are defined here.

Template Variable	Description
<i>Variable Name</i>	Typically assigned by the system, this name appears in the template configuration text and staged changeset if you do not specify a variable description.
<i>Default Value</i>	The system uses the default value for a variable when you deploy a changeset based on the template you create.
<i>Description</i>	A variable description is an important tool for properly identifying an existing template.

Template Variable	Description
<i>Editable</i>	Used by the Administrator-level user to specify whether users can change a variable when they use this template to stage a changeset.
<i>Visible</i>	Used by the Administrator-level user to hide the variable setting from restricted users when they use the template to stage a changeset.
<i>sEdit Allowed Value</i>	Used by the Administrator-level user to specify which values a user can select from when staging a changeset with this template.

### Publishing a template

Before you publish a template, you must first create it.

When you create a custom template, it is available only for you to deploy and use. To make the template available for others, you must publish it. This adds an additional layer of control to device configuration management.

1. On the Main tab, click **Enterprise Management > Configurations > Template List**.
2. Click the name of the template you want to publish.  
The general Properties screen for that template opens.
3. Select the **Published** check box.

This template is now available for others to use as a source.

# Index

## A

Application Security Manager security policies  
exporting [20](#)

## D

default value  
defined for template variables [21](#)  
description  
defined for template variables [21](#)  
DevCentral  
using customer-provided template text [19](#)  
device  
using to create a new template [16](#)

## E

editable  
defined for template variables [21](#)  
edit allowed values  
defined for template variables [21](#)

## H

HTTP virtual server  
creating with a standard template [16](#)

## L

local traffic server  
enabling and disabling with a standard template [16](#)

## N

network objects  
and dependencies for templates and changesets [12](#)  
and elements for templates and changesets [12](#)

## P

pool members  
enabling and disabling with a standard template [16](#)

## S

security policies  
exporting [20](#)  
standard templates  
shipped with system [16](#)

## T

template  
using to create a new template [18](#)  
templates  
about [16](#)  
about managing [16](#)  
creating [19](#)  
creating from existing device [16](#)  
creating from existing template [18](#)  
creating with text [18](#)  
example [21](#)  
importing from DevCentral [19](#)  
publishing [22](#)  
reviewing variable text [20](#)  
templates and changesets  
and dependencies for network objects [12](#)  
and network object elements [12](#)  
overview [12](#)  
template variables  
about [16](#)  
defined [21](#)  
example [21](#)  
reviewing [20](#)  
text  
using to create template [18–19](#)

## V

variable name  
defined for template variables [21](#)  
variables  
for templates [16](#)  
variable text  
reviewing for templates [20](#)  
visible  
defined for template variables [21](#)

