BIG-IP® iApps® Developer's Guide

Version 11.4

Table of Contents

Legal Notices

Publication Date

This document was published on October 15, 2013.

Publication Number

MAN-0469-00

Copyright

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

AAM, Access Policy Manager, Advanced Client Authentication, Advanced Firewall Manager, Advanced Routing, AFM, Alive With F5, APM, Application Acceleration Manager, 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 Certified [DESIGN], F5 Networks, Fast Application Proxy, Fast Cache, FirePass, Global Traffic Manager, GTM, GUARDIAN, iApps, IBR, Intelligent Browser Referencing, Intelligent Compression, IPv6 Gateway, iControl, iHealth, iQuery, iRules, iRules OnDemand, iSession, L7 Rate Shaping, LC, Link Controller, Local Traffic Manager, LTM, LineRate, LineRate Systems [DESIGN], LROS, Message Security Manager, MSM, OneConnect, Packet Velocity, PEM, Policy Enforcement Manager, Protocol Security Manager, PSM, Real Traffic Policy Builder, Scale^N, Signalling Delivery Controller, SDC, SSL Acceleration, StrongBox, SuperVIP, SYN Check, TCP Express, TDR, TMOS, Traffic Management Operating System, Traffix Systems, Traffix Systems (DESIGN), Transparent Data Reduction, UNITY, VAULT, VIPRION, vCMP, VE F5 [DESIGN], 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), which is protected under the GNU Public License.

This product includes software developed by Niels Mueller (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.

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 (eay@cryptsoft.com).

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

This product includes software with glib library utility functions, which is protected under the GNU Public License.

This product includes software with grub2 bootloader functions, which is protected under the GNU Public License.

This product includes software with the Intel Gigabit Linux driver, which is protected under the GNU Public License. Copyright ©1999 - 2012 Intel Corporation.

This product includes software with the Intel 10 Gigabit PCI Express Linux driver, which is protected under the GNU Public License. Copyright ©1999 - 2012 Intel Corporation.

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 software developed by Andrew Tridgell, which is protected under the GNU Public License, copyright ©1992-2000.

This product includes software developed by Jeremy Allison, which is protected under the GNU Public License, copyright ©1998.

This product includes software developed by Guenther Deschner, which is protected under the GNU Public License, copyright ©2008.

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

This product includes gd-libgd library software developed by the following in accordance with the following copyrights:

- Portions copyright ©1994, 1995, 1996, 1997, 1998, 2000, 2001, 2002 by Cold Spring Harbor Laboratory. Funded under Grant P41-RR02188 by the National Institutes of Health.
- Portions copyright ©1996, 1997, 1998, 1999, 2000, 2001, 2002 by Boutell.Com, Inc.
- Portions relating to GD2 format copyright ©1999, 2000, 2001, 2002 Philip Warner.
- Portions relating to PNG copyright ©1999, 2000, 2001, 2002 Greg Roelofs.
- Portions relating to gdttf.c copyright ©1999, 2000, 2001, 2002 John Ellson (ellson@lucent.com).
- Portions relating to gdft.c copyright ©2001, 2002 John Ellson (ellson@lucent.com).
- Portions copyright ©2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007 2008 Pierre-Alain Joye (pierre@libgd.org).
- Portions relating to JPEG and to color quantization copyright ©2000, 2001, 2002, Doug Becker and copyright ©1994, 1995, 1996, 1997, 1998, 1999, 2000, 2001, 2002, Thomas G. Lane. This software is based in part on the work of the Independent JPEG Group.
- Portions relating to WBMP copyright 2000, 2001, 2002 Maurice Szmurlo and Johan Van den Brande. Permission has been granted to copy, distribute and modify gd in any context without fee, including a commercial application, provided that this notice is present in user-accessible supporting documentation.

This product includes software developed by Oracle America, Inc. Copyright ©2012.

1. Java Technology Restrictions. Licensee shall not create, modify, change the behavior of, or authorize licensees of licensee to create, modify, or change the behavior of, classes, interfaces, or subpackages that are in any way identified as "java", "javax", "sun" or similar convention as specified by Oracle in any naming convention designation. In the event that Licensee creates an additional API(s) which: (a) extends the functionality of a Java Environment; and (b) is exposed to third party software developers for the purpose of developing additional software which invokes such additional API, Licensee must promptly publish broadly an accurate specification for such API for free use by all developer.
2. Trademarks and Logos. This License does not authorize an end user licensee to use any Oracle America, Inc. name, trademark, service mark, logo or icon. The end user licensee acknowledges that Oracle owns the Java trademark and all Java-related trademarks, logos and icon including the Coffee Cup and Duke ("Java Marks") and agrees to: (a) comply with the Java Trademark Guidelines at http://www.oraclc.com/html/3party.html; (b) not do anything harmful to or inconsistent with Oracle's rights in the Java Marks; and (c) assist Oracle in protecting those rights, including assigning to Oracle any rights acquired by Licensee in any Java Mark.
3. Source Code. Software may contain source code that, unless expressly licensed for other purposes, is provided solely for reference purposes pursuant to the terms of your license. Source code may not be redistributed unless expressly provided for in the terms of your license.
4. Third Party Code. Additional copyright notices and license terms applicable to portion of the Software are set forth in the THIRDPARTYLICENSEREADME.txt file.
5. Commercial Features. Use of the Commercial Features for any commercial or production purpose requires a separate license from Oracle. "Commercial Features" means those features identified in Table I-I (Commercial Features In Java SE Product Editions) of tile Software documentation accessible at http://www.oracle.com/technetwork/java/javase/documentation/index.html.

This product includes utilities developed by Linus Torvalds for inspecting devices connected to a USB bus.

This product includes perl-PHP-Serialization software, developed by Jesse Brown, copyright ©2003, and distributed under the Perl Development Artistic License (http://dev.perl.org/licenses/artistic.html).

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.

6. Redistributions of any form whatsoever must retain the following acknowledgment: "This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (http://www.openssl.org/)"

THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT ``AS IS'' AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE OpenSSL PROJECT OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This product includes software licensed from William Ferrell, Selene Scriven and many other contributors under the GNU General Public License, copyright ©1998 - 2006.

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 includes software developed by the Computer Systems Engineering Group at Lawrence Berkeley Laboratory. Copyright ©1990-1994 Regents of the University of California. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software must display the following acknowledgment: This product includes software developed by the Computer Systems Engineering Group at Lawrence Berkeley Laboratory.
4. Neither the name of the University nor of the Laboratory may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY

This product includes software developed by Sony Computer Science Laboratories Inc. Copyright © 1997-2003 Sony Computer Science Laboratories Inc. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY SONY CSL AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL SONY CSL OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

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 product includes the ixgbevf Intel Gigabit Linux driver, Copyright © 1999 - 2012 Intel Corporation, and distributed under the GPLv2 license, as published by the Free Software Foundation.

Chapter 1

iApps Overview

What is iApps?

iApps® is the BIG-IP® system framework for deploying services-based, template-driven configurations on BIG-IP systems running TMOS® 11.0.0 and later. iApps allows creation of application-centric configuration interfaces on BIG-IP systems, reducing configuration time and increasing accuracy of complex traffic management configurations.

Important: *Because objects that are initially configured by an iApps template can very difficult to change, there are configuration objects that are good to include in your iApps templates, and others that are good practice to exclude. In general, avoid elements that you configure using the BIG-IP System options. Examples of configuration objects that are good to include in iApps templates are:*

- pool members
- virtual servers
- health monitors
- profiles

About version 10 template incompatibility

BIG-IP version 10 introduced F5 Networks-supplied templates to perform common application-specific configuration tasks. iApps has replaced the version 10 templates with an expanded set of iApps templates that can be copied, customized, and shared but are not compatible.

Important: *If you are upgrading from a 10.x version of BIG-IP to version 11, you will have to reenter the 10.x template data in a similar template on the version 11 iApp template.*

Chapter 2

The Parts of iApps

iApps components

The iApps® framework consists of two main components, application services and templates.

Application services

iApps *application services* use templates to guide users through configuring new BIG-IP® system configurations. An application service lets an authorized user easily and consistently deploy complex BIG-IP® system configurations just by completing the information required by the associated template. Every application service is attached to a specific configuration and cannot be copied the way that iApps templates can.

Templates

iApps *templates* create configuration-specific forms used by application services to guide authorized users through complex system configurations. The templates provide programmatic, visual layout and help information. Each new application service uses one of the templates to create a screen with fields and help that guide the user through the configuration process and creates the configuration when finished.

iApps templates allow users to customize by either modifying an existing template or creating one from scratch. Users can create scratch-built templates using either the iApps Templates screen or any text-editing software.

Note: *iApps templates are used to create application services and cannot be used by themselves. This is a fundamental change from previous TMOS® versions in the way templates are used.*

About iApps templates

iApps® application services generate configurations based on application-centric user entries dictated by the template in use.

The BIG-IP® system comes with several system-supplied templates that you can use as-is to create your application services. You can also use the system-supplied templates as a starting point for your own templates, or you can write templates from scratch using, variously, `tmsh` and Tcl for the back-end template implementation section, the iApps application programming language (APL) for the presentation section, and HTML for the help section.

Template sections

Templates have four sections: macro, presentation, implementation, and help.

- The *macro section* provides template developers with the ability to develop macros that can create objects.
- The *presentation section* collects user entries.
- The *implementation section* uses user entries to build a configuration that will control traffic.
- The *help section* documents the template and its presentation to users when creating an application service.

Viewing a template's contents

You can view the contents of any iApps® template on a BIG-IP® system by clicking the template name in the Template List.

1. On the Main tab, click **iApp** > **Templates**.
   The Templates screen displays a list of available templates.
2. Click the name of the template that you wish to view.

The **Template Properties** screen opens so you can view the contents of the template.

Note: *You may edit only non-system-supplied templates from this view. System-supplied templates must be copied and renamed before they can be edited.*

Copying a template

Before modifying any template, you must license the BIG-IP® modules used by that template.

Ensure that you have administrator privileges on the BIG-IP system you are using to modify a copied template.

Important: *Once a template is modified, it can be used on any BIG-IP system running TMOS® 11.0 or later, as long as the user has sufficient privileges on that system for each called action.*

The easiest way to make a new iApps® template is to copy and then modify a system-supplied template. You can modify the copied template and include it in existing or new application services.

1. On the Main tab, click **iApp** > **Templates**.
   The Templates screen displays a list of available templates.
2. Click the name of a write-protected system-supplied template.
   The **Template Properties** screen for the template will open.
3. At the bottom of the screen click **Copy**.
   A copy of the template replaces the original template.
4. In the **Template Name** field, give the template copy a new name.

   You can edit the template now or come back to edit it later.
5. At the bottom of the screen click **Finished** to save your changes.

The system saves a copy of the template and makes it visible on the **Template List screen**. You can open the new template for editing by clicking the template name.

Signing a template

Before you can sign a template, you need to have a usable signing key available.

Signatures are one means of determining a template's authenticity. F5-authored and signed templates that have a valid signature display the F5 Verified logo in the Verification column of the Template list. Templates you author and sign that have a valid signature display the Signature Verified logo. The BIG-IP system can use unsigned templates, but if a template is signed and the signature is invalid, the template will fail to load. Checksum verification is the other method used to establish a template's integrity. You cannot use both a checksum and a signature.

You can sign a template that you create so that the BIG-IP® system will validate the signature before it loads the template.

1. On the Main tab, click **iApp** > **Templates**.
   The Templates screen displays a list of available templates.
2. Select the check box for the user-defined template that you want to sign.
3. Click **Add Signature**.
4. From the **Signing Key** list, select a signing key.
5. Click **Add Signature**.

The template is signed with a usable signing key.

Adding a checksum to a template

You can use a template with no checksum, but a template with an invalid checksum will not load.

You can add a checksum to an iApps® template that you create so that when someone downloads it, they can be confident in the integrity of the template.

1. On the Main tab, click **iApp** > **Templates**.
   The Templates screen displays a list of available templates.
2. Select the check box for the user-defined template that you want to add a checksum.
3. Click **Add Checksum**.

The checksum for the template is added.

iControl manipulation of iApp templates

Besides working with iApps® templates directly, you can also use iControl® for creating and manipulating templates. iControl methods available to use with iApps templates are listed here. For the most up-to-date list of iControl methods for iApps templates, refer to F5 DevCentral™ (`http://devcentral.f5.com`).

iControl method	Description
create	Creates a set of application templates. Each template automatically is created with a default empty "definition" action.
delete_all_templates	Deletes all application templates.
delete_application_template	Deletes a set of application templates.
get_action_implementation	Gets the implementation script text for the specified application template actions.
get_action_list	Gets the names of the actions in the application templates.
get_action_presentation	Gets the presentation text for the specified application template actions.
get_action_presentation_help	Gets the HTML help text for the specified application template actions.
get_description	Gets the descriptions for the specified application templates.

iControl method	Description
get_list	Gets the names of all application templates.
get_version	Gets the version information for this interface.
set_action_implementation	Sets the implementation script text for a set of actions in an application templates. The implementation is a TMSH script that can use the answers to the questions in the presentation, provided as variables, to create the configuration objects that make up the application service. This script is executed whenever the application is created or modified.
set_action_presentation	Sets the presentation text for a set of actions in application templates. The presentation contains Application Presentation Language (APL) text that describes what input is needed and how it should be displayed in the UI.
set_action_presentation_help	Sets the HTML help text for a set of actions in application templates.
set_description	Sets the descriptions for the specified application templates.

About application services

An iApps® application service lets you create and deploy new configurations by filling out screens with forms generated by the underlying information in the iApps template. Within the BIG-IP® Configuration utility, iApps users can manage, reconfigure, and view statistics about an application service once it is created.

About application services deployment requirements

Before deploying any iApps® template you should consider the following requirements:

- You must license and provision a template's required BIG-IP® modules on all BIG-IP systems that will be using the template.
- If you are modifying a template, ensure that you have sufficient privileges on the BIG-IP system you are using so you can test the template.

Important: *When creating an application service, users are prompted to license the modules required by the selected template if the modules are not licensed, and this may require permission that the user does not have.*

iApps user role permissions

The standard BIG-IP® user roles have built-in permission settings for iApps®. Only certain system user roles can create, edit, or delete templates or application services. All user roles with system access can see the contents of iApps templates and application services.

BIG-IP role	Manage application services?	Edit templates?
Administrator	Yes	Yes
Application Editor	Yes	No
Certificate Manager	No	No
Guest	No	No
Manager	Yes	No
Operator	No	No
Resource Administrator	Yes	No
User Manager	No	No

tmsh user role name cross-reference

The user roles displayed in the BIG-IP® Configuration utility are identified by slightly different names in `tmsh`. The cross-reference table below shows the Configuration utility user role name and the corresponding `tmsh` name.

Configuration utility user role name	`tmsh` value	Associated BIG-IP® module
Administrator	admin	all
Application Editor	application-editor	all
Certificate Manager	certificate-manager	all
Guest	guest	all
Manager	manager	all
No Access	no-access	all
Operator	operator	all
Resource Administrator	resource-admin	all
User Manager	user-manager	all
Web Application Security Administrator	web-application-security-administrator	ASM™
Web Application Security Editor	web-application-security-editor	ASM™

About the strict updates setting

When you are working in the Application Services properties screen, and select the **Advanced** view, the **Strict Updates** field is shown. Selecting **Strict Updates** protects against accidental changes to an application service's configuration. The **Strict Updates** setting is on by default when an application service is created.

Note: *Unless you have a specific reason to turn off strict updates, F5 recommends that you leave the setting on. If you do turn strict updates off, we do not recommend using Template reentrancy (discussed in the iApps Template Authoring chapter).*

When the strict updates setting is enabled, users can control only objects that are exposed through the templates.

***Note:** Even with strict updates enabled, it is possible to enable and disable some objects using interfaces (such as* `tmsh` *or the Configuration utility) other than the reentrant template. These objects include:*

- nodes
- pool members
- virtual addresses
- virtual servers

Deploying an application service

The following procedure covers the minimum steps needed to deploy a configuration using an iApps® application service.

1. On the Main tab, expand **iApp**, and click **Application Services**.
2. Click **Create**.
3. In the **Name** field, type the name for your application service.
4. From the **Template List** menu, select a template for your application, and wait for the screen to automatically refresh.
5. Configure remaining settings as needed.
6. At the bottom of the screen click **Finished** to save your changes.
7. Wait for the application properties to load.
8. (Optional) In the **Description** field, enter information to describe this application service and click **Update**.

Your application service is now deployed on the BIG-IP system.

Modifying an application service

The following procedure tells how to modify an existing application service.

1. On the Main tab, expand **iApp**, and click **Application Services**.
2. From the Application Service List, select an application service to view.
3. Click the Reconfigure tab.
   The screen displays the settings for the application service.
4. Click the Components tab and use the components tree to view the components that belong to the application service.
5. Edit the fields that require modification and then click **Finished** to save your changes.

The system saves the application service modifications and they are ready to use.

Chapter 3

iApps Template Authoring

About the iApps template framework

An iApps® template has four sections; macro, implementation, presentation, and help. You can view and edit each section with the system's Template Properties screen or with a text editor. If you choose to use an editor, you might find it helpful to see the structure of an empty template as shown here.

```
sys application template f5.template_name {
    actions {
        definition {
            html-help {
                <!-- insert html help text -->
            }
            implementation {
                tmsh::create cli alias shared f5.template_name command list
            }
            macro {
               <% #insert text and macro syntax here %>
            }
            presentation {
                # insert apl script
            }
            role-acl none
            run-as none
        }
    }
```

Help section

The help section, written in HTML, can provide users with more details about the template.

Macro section

The macro section provides template developers with the ability to develop macros that create objects, and then associate them with iRules®.

Implementation section

The implementation section contains the `tmsh` and Tcl code that performs the actions of the template on the BIG-IP® system.

Presentation section

The presentation section contains the application presentation language (APL) that is used by the BIG-IP system to create the template's graphical user interface (GUI).

About template design considerations

Well-designed, easy-to-use templates should be the goal of all iApps® template authors.

A complex presentation section can frustrate users and generate more support questions and comments for you, the template author. Here are some basic design considerations your template should follow:

- Use a question and answer format.

- Be simple.
- Be clear.
- Be complete.
- Be understandable.
- Follow established conventions.

About application reentrancy

Application reentrancy, describes the process of making revisions to an application service by revising the design template entries that you used to create it. When you select an application service and then click **Reconfigure**, you can change the entries on the design template that defines that application service. If you make changes to an application service that results in a new configuration, a mark-and-sweep process transforms the old configuration to this new configuration.

iApps service object

Stores the values associated with template inputs

Mark-and-sweep

On reentrancy, `tmsh create` is changed to `tmsh modify`. Untouched objects are deleted in the sweep.

About the macro section

The *macro section* of an iApps® template allows template developers to create macro templates. These templates produce generic text that can be used as iRules®. Using macros in this fashion makes it very straightforward to produce iRules that your users can easily customize with their unique input while maintaining syntactical accuracy and consistency.

When macros are used with iApps templates, the iApps user customizes the iRule simply by responding to simple questions. These responses are then used to populate the macro variables using a `tmsh` command (`TMSH::Macro_expand`).

The iApps developer can use the presentation section of the iApps template to make it very straightforward for end users unfamiliar with Tcl code to use and customize iRules. The graphical user interface (GUI) displays questions related to the customizable sections of the iRule that the developer specified in the presentation section of the iApps template. Users can then make simple modifications to create high performance, custom iRules. The resulting iRules are less error prone, and easier to deploy and provide the benefit of fully re-entrant iApps.

Built into the macro syntax is the ability to include implicit or explicit debug statements in the iRules text. These statements can provide iRules developers an excellent mechanism for adding troubleshooting support to their iRules without incurring a runtime cost when the iRules are in production.

For example, consider a template that requests responses for IP addresses and pool names. You create a macro file that contains a list of addresses and pool names, and distribute that to your users; the values will populate the template when the iApps template runs. Users can revise the macro file with entries appropriate for their topology.

Syntax Example

This sample iRule includes macro parameters set up to be populated at runtime.

```
when HTTP_REQUEST {
  if { [HTTP::uri] starts_with "/foobar" } {
    switch -glob [HTTP::uri] {
      "*[0-9].jpg" { pool numbers_pool }
      default {
        if { [string length [substr [HTTP::uri] 0 "?"]] > 0 } {
          HTTP::respond 200 content
           "<html><head><title>Where's  the
number?</title></head></body><h1>Where's the number?</h1></body></html>"
        }
      }
    }
  }
}
when HTTP_RESPONSE {
  if { [HTTP::header Content-Length] > 100 } {
    log local0. "too much data requested."
    drop
  }
}
```

This sample illustrates the same iRule, after the parameters have been populated using the macro syntax.

```
    when HTTP_REQUEST {
  if { [HTTP::uri] starts_with "<%=$url_prefix%>" } {
    switch -glob [HTTP::uri] {
      "<%=$glob_pattern%>" { pool <%=$matched_pool%> }
      default {
        if { [string length [substr [HTTP::uri] 0 "?"]] > 0 } {
          HTTP::respond 200 content

"<html><head><title><%=$html_title%></title></head></body><h1><%=html_message%></h1></body></html>"

        }
      }
    }
  }
}
when HTTP_RESPONSE {
  if { [HTTP::header Content-Length] > <%=$max_content_length%> } {
    log local0. "too much data requested."
    drop
  }
}
```

Macro delimiter syntax

The delimiters for macro parameters are defined in the table.

Delimiter/Command	Name	Description
%<	Code to Evaluate	The beginning of an expansion code block.
`<%=`	Evaluate and Output	The beginning of an expansion code block. Spool the output after evaluating.
`<%D[0-9][0-9]`	Debugging/Logging Code	The beginning of a debug/logging code block with the debug threshold set to 0 through 99.
`%>`	Close Code/Output Block	The end of the current block (applies to all delimiter types).
`<%D[0-9][0-9]=`	Debugging/Logging Output	The beginning of an debug expansion code block. Spool the output after evaluating.

About the implementation section

The *implementation section* of an iApps® template contains the executable code that processes the elements of the presentation section. Template authors must be familiar with Tcl and `tmsh` scripting in order to write implementations.

About value elements

Value elements are primitives in the implementation section that represent the actual user interface components. Displayed as part of the form, value elements are associated with a form variable such as `<type>`, `<name>`, or `<properties>`.

About the presentation section

iApps® templates use the application presentation language to create application-specific user interfaces in the BIG-IP® Configuration utility. Application presentation language code is parsed when a template is selected for use in an application service. When the template is updated, the system also checks the syntax of the code.

About the iApps application presentation language

The iApps® *application presentation language* defines the user interface in the presentation section of an iApps template. The language describes what questions to ask, how the questions are presented (input field verses a pull-down list, for example), and the names of the variables used to store the values of the user entries. It consists of a set of primitive form elements, a set of grouping and organization constructs, methods for hiding or displaying portions of the form based on the values of other portions (optional), and a method to associate various form elements (text) with text that users can read.

About template localization

Text strings in an iApp template can be localized by including localized text stings identified with ISO 639-1 language codes. The text string for a supported locale is selected at run time based on the requesting browser's locale preference. If a string for that locale is not included, the default text string is used.

Tip: *Let your template users know that localized text is available either with a note in help or with a notice at the top of the template.*

```
text
{
    loc.example “default text”
}

text "de_AT" {
    loc.example "Text für österreichische locale (text for German in Austrian
 locales)"
}

text "es_MX" {
    loc.example "texto predeterminado (text for Spanish in Mexico)"
}
```

About template double-byte character support

For international iApps® template audiences, the application presentation language `text` and `html` commands support Unicode double-byte characters. Using double-byte characters with any other Application Presentation Language (APL) commands might produce unpredictable results.

About application presentation language elements

Application presentation language elements are the graphics building blocks of the presentation section. The iApps® application presentation language uses the following elements to create the user interface for template users.

- string
- password
- choice
- editchoice
- multichoice
- table
- row
- message
- section
- optional
- text definitions
- include
- define

string

The `string` element is the most basic element where a user may enter a single value.

Properties

Property	Description
default	The default value presented if no value was previously entered. This can be a string or a Tcl expression.
display	Gives a hint to the renderer about how this element should be displayed. The values can be `small`, `medium`, `large`, `xlarge`, or `xxlarge`.
required	If this property is displayed, the user must provide a valid value.
validator	Tells the renderer what type of data is valid. If input is not valid, the user is returned to the template screen that has a red error message next to the field in question. Validators include `FQDN`, `IpOrFqdn`, `IpAddress`, `NonNegativeNumber`, `Number`, and `PortNumber`

Syntax Example

```
string user_name required
string port default "80" display "small" validator "PortNumber"
```

password

The `password` element is similar to a `string` element, except the contents are obscured to protect the data.

Properties

Property	Description
display	Gives a hint to the renderer about how this element should be displayed. The values can be `small`, `medium`, `large`, `xlarge`, or `xxlarge`.
required	If this property is displayed, the user must provide a valid value.

Syntax Example

```
password passwd required
```

choice

The `choice` element presents a list of options where only one item can be selected.

Properties

Property	Description
choices	Required. This lists the available items that the user can select. This is a list of strings that can be defined manually or populated with a TCL expression.
default	The default value presented if no value was previously entered. This can be a string or a Tcl expression.
display	Gives a hint to the renderer about how this element should be displayed. The values can be `small`, `medium`, `large`, `xlarge`, or `xxlarge`.

Syntax Examples

```
        choice pools tcl { tmsh::run_proc utils:get_items ltm pool }
```

```
        choice yesno default "no" {"yes", "no"}
```

```
        choice load_balance_method { "Round Robin" => "round-robin", "Least
Connections" => "least-connections-memeber", "Ratio" => "ratio-member" }
```

Note: *In the choices family of APL elements, the => attribute maps the actual value provided to the implementation script to what is displayed in the Configuration utility.*

editchoice

The `editchoice` element presents multiple choices that users can select, and allows the user to enter a new value if the default choices are not acceptable.

Properties

Property	Description
choices	Required. This lists the available items that the user can select. This is a list of strings that can be defined manually or populated with a TCL expression.

Property	Description
default	The default value presented if no value was previously entered. This can be a string or a Tcl expression.
display	Gives a hint to the renderer about how this element should be displayed. The values can be `small`, `medium`, `large`, `xlarge`, or `xxlarge`.

Syntax Example

```
editchoice port default "80" {"* All Services" => "*", "HTTP" => "80", "HTTPS"
 =>"443"}
```

Note: *In the choices family of APL elements, the => attribute maps the actual value provided to the implementation script to what is displayed in the Configuration utility.*

multichoice

Use the `multichoice` element to allow the user to select multiple items from the available choices.

Properties

Property	Description
choices	Required. This lists the available items that the user can select. This is a list of strings that can be defined manually or populated with a TCL expression.
default	The default value presented if no value was previously entered. This can be a string or a Tcl expression.
display	Gives a hint to the renderer about how this element should be displayed. The values can be `small`, `medium`, `large`, `xlarge`, or `xxlarge`.

Syntax Example

```
multichoice your_string default {"Choice1", "Choice3"} {"Choice1", "Choice2",
 "Choice3"}
```

Note: *In the choices family of APL elements, the => attribute maps the actual value provided to the implementation script to what is displayed in the Configuration utility.*

table

The `table` element can have items added to it, with each item containing multiple pieces of information. Each element added as a child of the `table` element defines the columns of the table. While the `string` and `choice` elements give the ability to set scalar types, `table` provides the ability to set list types.

Syntax Example

```
table servers {
   string address
   choice port {"80", "443"}
}
```

row

You can also think of the `row` element as a single-row table, and each element that is added as a child of the `table` element defines the columns of the row.

Syntax Example

```
your_row seven { string bar1 string bar2}
```

message

The `message` element adds an extra text message within a section.

Syntax Example

```
message mymsg "This is my message to you."
```

section

The `section` element is the topmost hierarchical layout element. The primary function of this element is to group related sub-elements together.

Syntax Example

```
section ssl_questions {
    choice ssl_enabled { "true", "false" }
}
```

define

The `define` element allows the creation of user-defined types created out of existing types. The defined type can then be used multiple times independently. This is especially useful in conjunction with the `include` element because types can be defined in the included application presentation language script, and then used where necessary in the template.

Syntax Example

```
define choice boolean_question {"True", "False"}

section basic {
    boolean use_ssl
}
```

define group type

The built-in `group`type for `define` allow the you to group multiple elements together into a single defined type so that it can be reused, but will not visually group the elements like "section" does. .

Usage considerations	Details
Restricted use	The `group` type can only be used as part of a `define`.
Transparent for parent/child type checking	You cannot have a top-level string; it must be inside of a section. When the traversal type checks are made, the group element is ignored and the parent is compared with the top-level elements in the group.
No text association	The `group` element is not allowed to have text associated with it.

Syntax Example

```
sys application apl-script f5.apl_common {
    script {
        define group snatpool {
            noyes use_snat
            optional ( use_snat == "No" ) {
                noyes need_snatpool
                optional ( need_snatpool == "Yes" ) {
                    table snatpool_members {
                        string addr required validator "IpAddress"
                    }
                }
            }
        }
    }
}

sys application template f5.http {
    presentation {
        section basic {
            ...
            snatpool snat
        }
        text {
            basic.snat.use_snat "Do the HTTP servers have a route back to
application clients via this BIG-IP system?"
          basic.snat.need_snatpool "Will you have more than 64,000 connections
 at one time? If so, you will need to enter at least one IP address for each
 64,000 connections. "
            basic.snat.snatpool_members "Enter IP addresses that can be used
 for a SNAT pool. Enter one IP address for each 64,000 connections "
            basic.snat.snatpool_members.addr "Address: "
        }
    }
}
```

optional

The `optional` element allows application variables to be hidden or shown based on the state of other application variable. The syntax for the `optional` element is: `optional (<expr>) { <contents...> }`. Visibility of `optional` elements is re-evaluated any time the value of an element changes. With `optional`, you can use only elements that have a value like `string` or `choice`.

Syntax Example

In this example, the `optional_question` string should be shown only if *`use_ssl`* is set to `"True"`

Important: *Only the == and `!=` operators are valid in the optional statement. This may require a little extra thought on your part if you are used to creating conditional statements using a broader range of operators.*

```
choice use_ssl {"True", "False"}
optional (use_ssl == "True") {
    string optional_question
}
```

include

The `include` element copies the referenced content of the application presentation language script into the current document.

Properties

None.

Syntax Example

```
include "com.f5.apl.common"
```

text

The `text` element lets you define the text labels and questions for sections, elements, table, or row sub-elements. The syntax for the `text` element is: `text { <section, element, table, or row>.<string name> <"string contents in parenthesis"> }`.

Syntax Example

In this example, the text strings contain the template's "edit" section's questions.

```
section edit
{
 string addr
 string port
}
Text
{
 edit “The EDIT section”
 edit.addr “What IP address do you want for the EDIT virtual server?”
 edit.port “What port should the EDIT virtual server answer on?”
}
```

Dynamic data binding using Tcl expressions

You can use Tcl expressions to dynamically generate default values or an available-choices list at run time. Tcl expressions can retrieve information from the mcpd service and process the data as needed.

After a Tcl script is run, the returned data can be processed into discrete items separated by a new line (`\n`). Next, split the items by tabs (`\t`) into display and value segments that represent what the user sees, and what is saved by the *mcpd*, respectively. Without a tab to split the items, the display is the same as the value.

For example, if a Tcl expression for a `choice` element returns `disp1\t val1\n disp2\t val2`, then the list of items presented to the user would be `disp1` and `disp2`.

The following example creates a `choice` that presents a list of LTM®pool names. Note how the `tmsh` commands are used to retrieve the data.

```
section my {
    choice pools
        tcl {
            set objs [tmsh::get_config ltm pool]
            foreach obj $objs {
                append results [tmsh::get_name $obj]
                append results "\n"
            }
            return $results
        }
}
```

About disabled Tcl commands

When you use iApps® to create your own templates from scratch, along with `tmsh`, you also use Tool Command Language (Tcl) for most of the basic instructions. Because a subset of Tcl commands is disabled from use with iApps templates, you should avoid using these when creating or editing iApps templates.

The Tcl commands that are disabled for iApps template use are:

after	auto_execok	auto_import
auto_load	auto_mkindex	auto_mkindex_old
auto_qualify	auto_reset	bgerror
cd	close	eof
exec	exit	fblocked
fconfigure	fcopy	file
fileevent	filename	flush
gets	glob	http
interp	load	memory
namespace	open	package
pid	pkg::create	pkg_mkIndex
proc	pwd	rename
seek	socket	source
tcl_findLibrary	tell	unknown
update	uplevel	upvar
vwait		

About presentation section usage recommendations and limitations

The iApps® application presentation language has the following usage recommendations and limitations. Follow the recommendations and avoid the limitations to ensure that the system will accept and run your template.

Recommendation or limitation	Detail
Forward definition order	Variables exist only if they have been defined earlier in the template. Specifically, you can create an optional section that displays its contents based on a field above it in the template, but not based on a field that exists below it in the template.
No nested sections or tables	The section and table elements do not function correctly if they are embedded in other APL elements.

Recommendation or limitation	Detail
Optionals inside tables must depend on elements outside the table	iApps cannot locate variables inside a template's table, so they must be located outside the APL table element structure.

About layout elements

Layout elements do not have a value or define a variable. Their primary function is to give hints to the renderer on how the children of the element should be displayed. The layout elements also affect the variable name for the element. When referencing a specific variable within the application presentation language, such as within an expression, then the fully-qualified dotted name is used. For example, if a string called `string` is contained within a section, `my`, then the full name would be `my.string`.

About element validators

Validators are used on certain elements to define what user input is allowable. The BIG-IP® system always has the final say on what is allowed, but if a validator value is given to the Configuration utility, it can present the user with a more useful error message about the location of the error.

Note: *The validator name is not case-sensitive.*

Validator	Accepts
FQDN	Domain name (RFC 1034)
IpOrFqdn	ipv4, ipv6 or domain name (explicitly disallow "*")
IpAddress	"*", ipv4 or ipv6
NonNegativeNumber	Integer >=0
Number	Any integer
PortNumber	"*" or int value 0-65535 (inclusive)

The following is an example of a validator that restricts text box input to a valid port number.

```
string port default "*" validator "portnumber"
```

About the help section

The *help section* of an iApps® template contains HTML help text that template users can access in the BIG-IP® Configuration utility under the Help tab. Template authors can supply as little or as much help content as needed. F5 Networks encourages all template authors to provide at least a minimum amount of help.

Important: *For security purposes, the help section only displays text using a simplified set of HTML commands. The help section does not support images, links, and scripts.*

HTML tags supported in the template help section

An iApps® template's help section supports only the HTML tags listed in the following table. Templates containing HTML tags that are not listed in the table generate an error message when an application service tries to use them.

HTML tag	Function
<b>	Defines bold text
<blockquote>	Defines a long quotation
<br />	Defines a single line break
<code>	Defines computer code text
<dd>	Defines a description of a term in a definition list
<dl>	Defines a definition list
<dt>	Defines a term (an item) in a definition list
<em>	Defines emphasized text
<h1> to <h6>	Defines HTML headings
<i>	Defines italic text
<li>	Defines a list item
<ol>	Defines an ordered list
<p>	Defines a paragraph
<pre>	Defines preformatted text
<small>	Defines small text
<strike>	Defines strikethrough text
<strong>	Defines strong text
<sub>	Defines subscripted text
<sup>	Defines superscripted text
<u>	Defines underlined text
<ul>	Defines an unordered list
<!--...-->	Defines a comment

About the role-acl property

The *`role-acl`* property restricts user access to application services. Besides using the standard user roles to control access to application services, you can further restrict access by setting the `role-acl` property on the template used for an application service. For example, if you want only administrators to manage the application services, then you set the `role-acl` property to just `admin`.

The role-acl property exists at the same hierarchal level in the template as the template sections, as shown below. The role-acl property can only be edited using a text editor because it is not visible in the Configuration utility.

```
   sys application template f5.template_name {
    actions {
        definition {
            html-help {
                                    <HTML help definition goes here>
            }
            implementation {
                                    <Back-end TCL and TMSH code goes here>
            }
            presentation {
                                    <APL GUI code goes here>
            }
            role-acl { admin manager resource-admin }
        }
    }
}
```

The role-acl property is also useful if a template creates objects that the user role is not able to create. For example, the application editor role is not allowed to create pools, so if your template creates pools, then you should set the role-acl property so that the user gets a more meaningful error message.

Chapter 4

Managing iApp Objects with Generic Application Services

About generic application services

You can use the Generic Application Services feature to create empty application services (without using an iApps® application template), which enables you to group new and existing configuration objects. After creating a generic application service, you can apply a template to create an application service.

When you create a new application service, from the Template list, select **None-Do not use a template** to configure a new generic application service. In addition, you can convert an existing application service into a generic application service by reconfiguring the Template setting to use the **None - Do not use a template** setting.

Index