Cisco BroadWorks

Xtended Services Platform

Configuration Guide

Release 23.0

Document Version 2

Notification

The BroadSoft BroadWorks has been renamed to Cisco BroadWorks. Beginning in September 2018, you will begin to see the Cisco name and company logo, along with the new product name on the software, documentation, and packaging. During this transition process, you may see both BroadSoft and Cisco brands and former product names. These products meet the same high standards and quality that both BroadSoft and Cisco are known for in the industry.

Copyright Notice

Copyright© 2019 Cisco Systems, Inc. All rights reserved.

Trademarks

Any product names mentioned in this document may be trademarks or registered trademarks of Cisco or their respective companies and are hereby acknowledged.

Document Revision History

Release	Version	Reason for Change	Date	Author
14.0	1	Created document.	March 31, 2008	Martin Bernier
14.0	1	Edited and published document.	April 17, 2008	Andrea Fitzwilliam
14.0	2	Fixed context path in section *6.1 Server Upgrade*.	June 23, 2008	Omar Paul
14.0	2	Edited and published document.	June 23, 2008	Patricia Renaud
14.0	3	Updated for EV 60316.	July 1, 2008	Martin Bernier
14.0	3	Edited and published document.	July 9, 2008	Patricia Renaud
15.0	1	Updated document for Releases 14.sp1 through 14.sp6 and Release 15.0.	July 9, 2008	Patricia Renaud
15.0	2	Clarified the configuration mode usage.	October 30, 2008	Yves Racine
15.0	2	Edited changes and published document.	November 14, 2008	Andrea Fitzwilliam
15.0	3	Indicated in section *8.8.5 Manage SSL Certificates* that the certificate signing request (.csr) is copied in */tmp*.	November 25, 2008	Yves Racine
15.0	3	Edited changes and published document.	December 3, 2008	Andrea Fitzwilliam
15.0	4	Changed section *8.8 HTTP Server/Web Container Configuration* to indicate that a secured HTTP server is automatically configured on a fresh install for EV 87170.	December 22, 2008	Yves Racine
15.0	4	Edited changes and published document.	February 11, 2009	Andrea Fitzwilliam
16.0	1	Introduced document for Release 16.0.	May 25, 2009	Yves Racine, Martin Bernier, Mario Goupil
16.0	1	Updated document following review.	June 24, 2009	Mario Goupil
16.0	1	Edited changes and published document.	August 26, 2009	Andrea Fitzwilliam
17.0	1	Updated document for Release 17.0.	February 21, 2010	Yves Racine
17.0	1	Edited changes and published document.	April 6, 2010	Andrea Fitzwilliam
17.0	2	Updated section *2.1 Deployment Models* to indicate that an Xtended Services Platform model without a Network Server is not supported, for EV 113489.	July 9, 2010	Goska Auerbach
17.0	2	Updated section *6.2 Applications Installation, Upgrade, and Patching* for EVs 113631 and 113491.	July 12, 2010	Goska Auerbach
17.0	2	Edited changes and published document.	July 28, 2010	Andrea Fitzwilliam
18.0	1	Updated document for Release 18.0	October 7, 2011	Yves Racine
18.0	1	Edited changes and published document.	November 7, 2011	Patricia Renaud
19.0	1	Updated document for Release 19.0.	October 3, 2012	Yves Racine
19.0	1	Reflected changes introduced by EV147586 WebContainer Platform Enhancements in Release 19.0.	October 23, 2012	Eric Ross, Martin Bernier

©2019 CISCO SYSTEMS, INC.

Release	Version	Reason for Change	Date	Author
19.0	1	Edited changes and published document.	November 26, 2012	Patricia Renaud
20.0	1	Updated document for Release 20.0.	January 21, 2013	Pierre Drapeau
20.0	1	Updated section *8.8 HTTP Server Configuration* for EV 143355.	January 21, 2013	Pierre Drapeau
20.0	1	Updated section *8.8.5 Manage SSL Certificates* for EV 128419.	January 21, 2013	Pierre Drapeau
20.0	1	Reviewed changes.	January 23, 2013	Goska Auerbach
20.0	1	Added section *6.3 Additional Installation Tips* for EV 186932.	April 10, 2013	Pierre Drapeau
20.0	1	Edited changes and published document.	October 4, 2013	Joan Renaud
21.0	1	Updated document for Release 21.0. Updated section *8.2 Profile Tuning*.	November 20, 2013	Renaud Beaudry-Magnan
21.0	1	Updated document to include all modifications done in Release 21.0.	October 9, 2014	Yves Racine
21.0	1	Edited changes and updated the copyright notice.	October 16, 2014	Joan Renaud
21.0	1	Rebranded and published document.	December 19, 2014	Joan Renaud
22.0	1	Updated document to include all modifications done in Release 22.0.	October 21, 2016	Eric Ross
22.0	1	Edited changes and published document.	December 16, 2016	Joan Renaud
23.0	1	Updated document to include all modifications done in Release 23.0.	September 20, 2018	Gulen Buyukbayram
23.0	1	Rebranded document for Cisco. Edited changes and published document.	September 27, 2018	Joan Renaud
23.0	2	Rebranded product name for Cisco and published document.	March 14, 2019	Joan Renaud

©2019 CISCO SYSTEMS, INC.

Table of Contents

1 Summary of Changes 1

1.1 Changes for Release 23.0, Document Version 2 1

1.2 Changes for Release 23.0, Document Version 1 1

1.3 Changes for Release 22.0, Document Version 1 1

1.4 Changes for Release 21.0, Document Version 1 1

1.5 Changes for Release 20.0, Document Version 1 1

1.6 Changes for Release 19.0, Document Version 1 1

1.7 Changes for Release 18.0, Document Version 1 2

1.8 Changes for Release 17.0, Document Version 2 2

1.9 Changes for Release 17.0, Document Version 1 2

1.10 Changes for Release 16.0, Document Version 1 2

1.11 Changes for Release 15.0, Document Version 4 2

1.12 Changes for Release 15.0, Document Version 3 2

1.13 Changes for Release 15.0, Document Version 2 2

1.14 Changes for Release 15.0, Document Version 1 2

1.15 Changes for Release 14.0, Document Version 3 2

1.16 Changes for Release 14.0, Document Version 2 2

1.17 Changes for Release 14.0, Document Version 1 3

2 Introduction 4

2.1 Deployment Models 4

2.2 Optional Front End 7

3 Getting Started 8

4 Software Distribution 9

5 Licensing 10

6 Installation, Upgrade, and Patching 11

6.1 Server Upgrade 11

6.2 Applications Installation, Upgrade, and Patching 12

6.3 Additional Installation Tips 14

7 Xtended Services Platform Components 15

7.1 Core Daemons 15

7.1.1 Software Manager 15

7.1.2 SNMP Agent 15

7.1.3 License Manager 16

7.1.4 Configuration Agent 17

7.1.5 Platform Processes 18

7.2 Controllable Applications 19

7.2.1 WebContainer 19

8 Configuration and Management 21

©2019 CISCO SYSTEMS, INC.

8.1 Configuration Modes 21
8.2 Profile Tuning 21
8.2.1 Profile Characterization 21
8.2.2 CLI Level 22
8.3 WebApplication Threading 23
8.3.1 General Concept 23
8.3.2 Typical Configurations 24
8.4 Configuration of BWCommunicationUtility 25
8.4.1 Integration Mode 25
8.4.2 Provision to Secondary 26
8.4.3 Communication Settings 26
8.4.4 Overflow Protection Settings 26
8.4.5 Executors 26
8.4.6 External Authentication Support 26
8.4.7 Name Service 27
8.5 Configuration of WebContainer 27
8.5.1 General Settings 28
8.5.2 Logging 28
8.5.3 Executors 29
8.5.4 Process Metrics 30
8.5.5 Overload Protection 31
8.5.6 Session Management 33
8.5.7 Thresholds 34
8.6 Management of WebContainer 36
8.6.1 Metric Groups 36
8.6.2 Sequence Diagrams 36
8.6.3 Process Metrics 41
8.6.4 Thresholds 42
8.7 Application Management 43
8.7.1 Installation 43
8.7.2 Activation 44
8.7.3 Deployment 45
8.7.4 Undeployment 51
8.7.5 Deactivation 52
8.7.6 Uninstallation 53
8.7.7 Upgrade 53
8.8 HTTP Server Configuration 54
8.8.1 Public Host Name 55
8.8.2 Secure HTTP Server 55
8.8.3 Create HTTP Server 55
8.8.4 Delete HTTP Server 56
8.8.5 Manage SSL Certificates 56

©2019 CISCO SYSTEMS, INC.

8.8.6 HTTP Aliases 60
8.8.7 HTTP Bindings 60
8.9 Network Server Configuration 60
9 Network Configuration Changes 61
9.1 Change IP or Host Name of Xtended Services Platform 61
9.1.1 Update HTTP Server Configuration 61
9.2 Change IP, Host Name or FQDN of Network Server 61
9.3 Change IP or Host Name of Application Server 62
10 Troubleshooting 63
10.1 Logs 63
10.1.1 HTTP Access Logs 63
10.1.2 Tomcat Logs 63
10.1.3 Web Application Logs 63
10.1.4 Cisco BroadWorks Application Logs 63
10.1.5 Configuration Agent Logs 63
10.1.6 Other Cisco BroadWorks Logs 63
10.2 Common Problems 64
10.2.1 Authentication Failure for Valid User 64
10.2.2 OCI-C Connectivity Problems 64
11 Xtended Services Platform Web Applications Requirements 66
11.1 Application Packaging Requirements 66
11.1.1 War File Name 66
11.1.2 War File Structure and Contents 66
11.2 Web Application Descriptor File (web.xml) 67
11.3 Web Application Naming 68
11.4 Web Application Versioning 68
11.5 Web Application External Configuration 68
11.5.1 BwCommunicationUtility Overrides 69
11.6 Web Application Context Path 69
References 70

©2019 CISCO SYSTEMS, INC.

Table of Figures

Figure 1 Xtended Services Platform Deployment Models 5
Figure 2 Components Overview 6
Figure 3 Load Balancer Deployment Example 7
Figure 4 Web Container Assembly 19
Figure 5 HTTP Blocking Sequence Diagram (HttpNiO) 37
Figure 6 HTTP Streaming Sequence Diagram (HttpNio) 38
Figure 7 CTI Sequence Diagram 38

©2019 CISCO SYSTEMS, INC.

1 Summary of Changes

This section describes the changes to this document for each release and document version.

1.1 Changes for Release 23.0, Document Version 2

This version of the document includes the following change:

- Rebranded product name for Cisco.

1.2 Changes for Release 23.0, Document Version 1

This version of the document includes the following change:

- Updated document for Release 23.0.

1.3 Changes for Release 22.0, Document Version 1

This version of the document includes the following changes

- Added configuration support for HTTP Strict Transport Security headers.
- Removed Apache and generalization for the use of HTTP Nio connector.
- Added support for HTTPS from BWCommunicationUtility.
- Added configuration enhancements for Cipher and SSL protocol.
- Introduced the BroadWorks Load Balancer application.

1.4 Changes for Release 21.0, Document Version 1

This version of the document includes the following changes:

- Updated document for Release 21.0.
- Updated section *8.2 Profile Tuning*.

1.5 Changes for Release 20.0, Document Version 1

This version of the document includes the following changes:

- Updated section *8.8 HTTP Server Configuration* for EV 143355.
- Updated section *8.8.5 Manage SSL Certificates* for EV 128419.
- Added section *6.3 Additional Installation Tips* for EV 186932.

1.6 Changes for Release 19.0, Document Version 1

This version of the document includes the following changes:

- Updated the list of applications available on the Xtended Services Platform.
- Updates to reflect feature EV147586 *WebContainer Platform Enhancements* with the following modifications:

  Added the following sections:

  - *7.2.1.1 WebContainer Software Components*
  - *7.2.1.2 WebContainer Metric Groups*
  - *8.2 Profile Tuning*

©2019 CISCO SYSTEMS, INC.

- *8.4.5 Executors*
- *8.4.7 Name Service*
- *8.5 Configuration of WebContainer*
- *8.6 Management of WebContainer*

Updated the following sections:

- *8.4.4 Overflow Protection Settings*
- *8.8.7 HTTP Bindings*

1.7 Changes for Release 18.0, Document Version 1

There were no changes to this document for Release 18.0.

1.8 Changes for Release 17.0, Document Version 2

This version of the document includes the following changes:

- Updated section *2.1 Deployment Models* for EV 113489. Specified that an Xtended Services Platform model without a Network Server is not supported.
- Updated section *6.2 Applications Installation, Upgrade, and Patching* for EVs 113631 and 113491.

1.9 Changes for Release 17.0, Document Version 1

There were no changes to this document for Release 17.0.

1.10 Changes for Release 16.0, Document Version 1

Updated document with Cisco BroadWorks Application management.

1.11 Changes for Release 15.0, Document Version 4

Changed section *8.8 HTTP Server Configuration* to indicate that a secured HTTP server is automatically configured on a fresh install.

1.12 Changes for Release 15.0, Document Version 3

Indicated in section *8.8.5 Manage SSL Certificates* that the certificate signing request (.csr) is copied in */tmp*.

1.13 Changes for Release 15.0, Document Version 2

Enhanced section *8.4.1 Integration Mode* to clarify the configuration mode usage.

1.14 Changes for Release 15.0, Document Version 1

There were no changes to this document for Release 15.0.

1.15 Changes for Release 14.0, Document Version 3

Updated document for EV 60316 Xtended Services Platform Enhancements.

1.16 Changes for Release 14.0, Document Version 2

Fixed the context path in section *6.1 Server Upgrade*.

©2019 CISCO SYSTEMS, INC.

1.17 Changes for Release 14.0, Document Version 1

The document was created for this release.

©2019 CISCO SYSTEMS, INC.

2 Introduction

The Xtended Services Platform is a general-purpose application-hosting server. It is meant as a controlled platform on which applications that integrate with other Cisco BroadWorks services/servers can run. The Xtended Services Platform is based on two container frameworks, the BroadWorks Container and the Web Container. The BroadWorks Container provides flexibility for the applications, scalability for the servers, and ease to manage the development and deployment of current and future applications. Applications running inside the BroadWorks Container can have their own threading model, garbage collection algorithm, and memory configuration.

The Web Container relies on the industry reference Apache Hypertext Transfer Protocol (HTTP) Server and Apache Tomcat Servlet Container.

The Xtended Services Platform supplies an enhanced Cisco BroadWorks platform that allows installation, activation, and deployment of Cisco BroadWorks and web applications. It also comes with the usual Cisco BroadWorks platform services such as a command line interface (CLI), Simple Network Management Protocol (SNMP) agent, process monitoring, and patching subsystem.

The Xtended Services Platform also includes the BWCommunicationUtility, which provides base functionality for all web applications. It provides integrated services, which allow simple communication to other Cisco BroadWorks servers. It also provides web application building blocks such as security and authentication services and web application overload protection to Cisco BroadWorks users.

The Xtended Services Platform provides a comprehensive CLI interface that allows operators to install new Cisco BroadWorks (and web) applications and use them as required.

2.1 Deployment Models

The Xtended Services Platform can be deployed in different models depending on the services offered. It can be deployed as a stand-alone server or as pools of servers. Each Xtended Services Platform in a pool is independently managed. No replication occurs between Xtended Services Platform Servers.

The Xtended Services Platform can be located at the front end of multiple Application Server clusters, and a Network Server is always required.

When using pools of servers, the recommended approach is to identify multiple pools of segregated applications. In other words, there should be multiple pools of like-servers such as a pool for Device Management (DM) applications, another pool for Xtended Services Interface (Xsi) applications, and so on.

©2019 CISCO SYSTEMS, INC.

Figure 1 shows the deployment models.

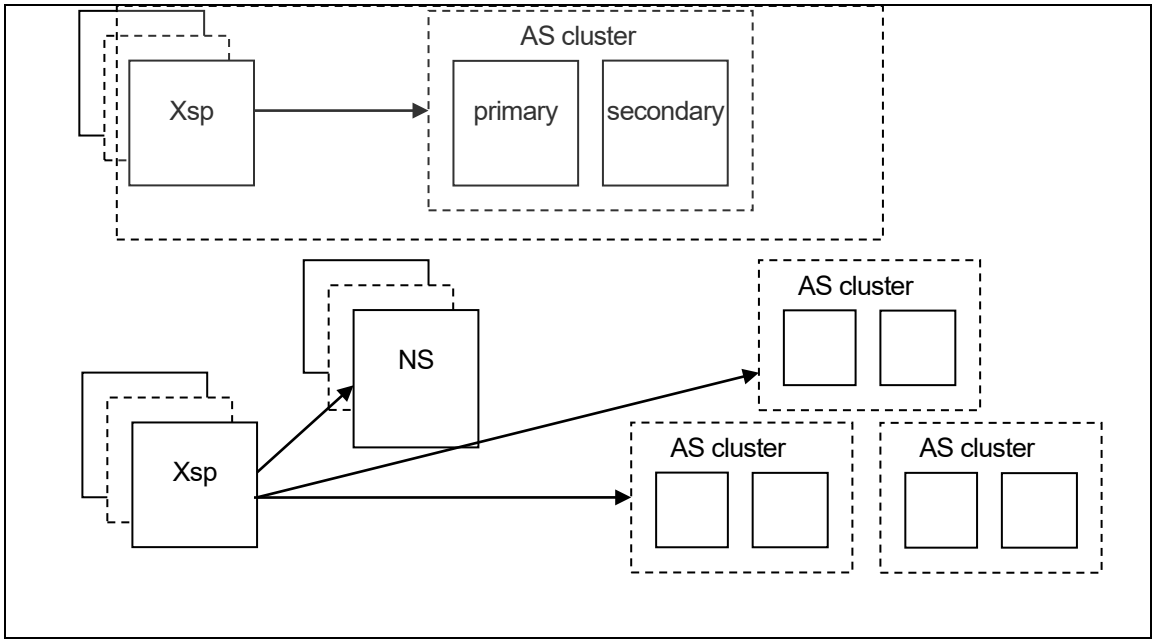

Figure 1 Xtended Services Platform Deployment Models

©2019 CISCO SYSTEMS, INC.

Figure 2 shows the major software components of the Xtended Services Platform and how they relate to each other.

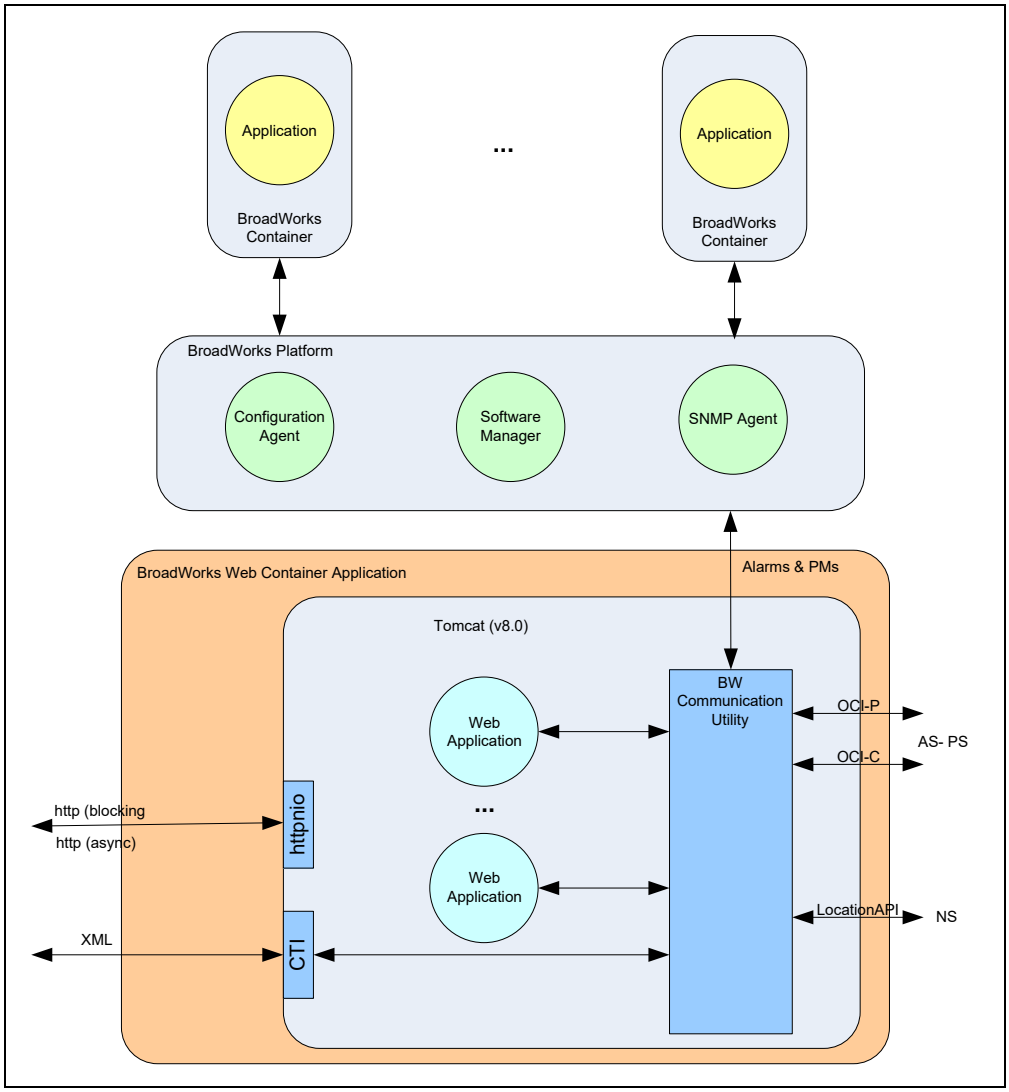

Figure 2 Components Overview

This guide provides general Xtended Services Platform product description and deployment information. It explains the Cisco BroadWorks and web applications management features of the Xtended Services Platform and provides some guidelines on web application development.

©2019 CISCO SYSTEMS, INC.

2.2 Optional Front End

The Load Balancer is an HTTP reverse proxy that can be deployed on an Xtended Services Platform in lieu of the Web Container application. As a gateway in front of other Xtended Services Platforms, it allows for the quick reconfiguration of proxied server farms hosting HTTP services.

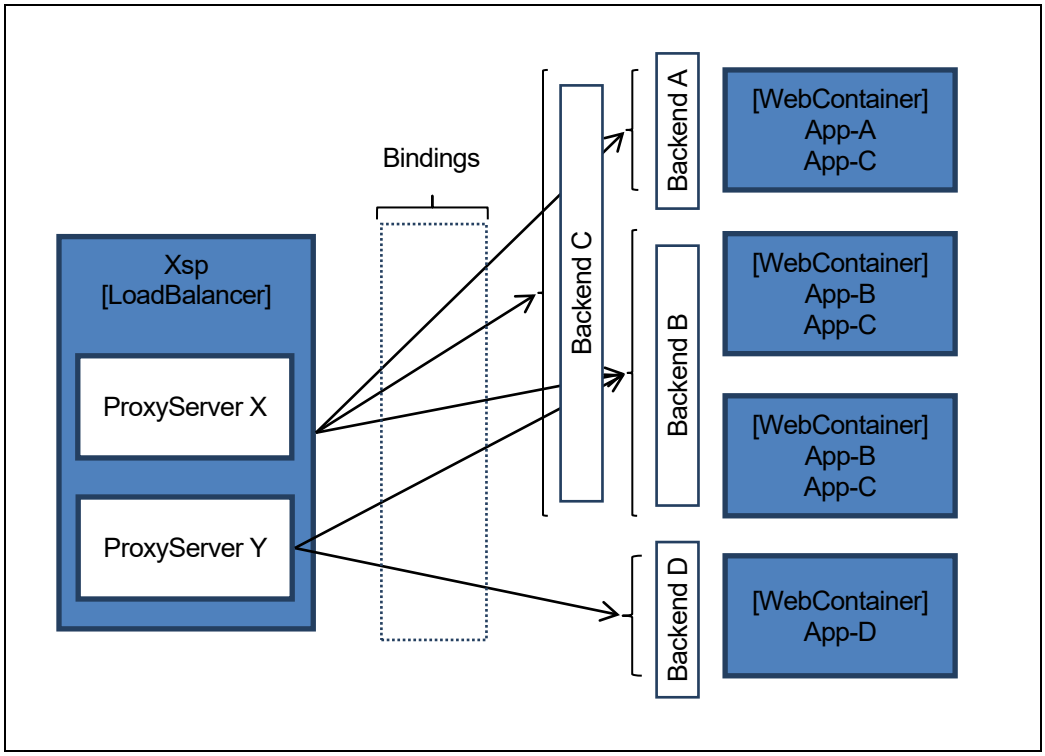

Figure 3 Load Balancer Deployment Example

For more information about the Load Balancer application, see the *Xtended Services Platform Load Balancing Feature Description* [6].

©2019 CISCO SYSTEMS, INC.

3 Getting Started

The following steps should be executed to start an Xtended Services Platform server.

1) Obtain a license from Cisco. For more information, see section *5 Licensing*.
2) Install the Xtended Services Platform from the binary image. For information, see the *Cisco BroadWorks Software Management Guide* [1].
3) Configure the *BWCommunicationUtility* on the Xtended Services Platform. For information, see section *8.4 Configuration of BWCommunicationUtility*.
4) Configure the Application Server cluster to allow Xtended Services Platform connectivity.
5) Start the Xtended Services Platform.

©2019 CISCO SYSTEMS, INC.

4 Software Distribution

The Xtended Services Platform is distributed using binary images for Linux. The image files are named using the usual format:

- XSP_Rel_23.0_1.yyy.Linux-x86_64.bin

Where yyy is a unique identifier for the Xtended Services Platform Release 23.0.

©2019 CISCO SYSTEMS, INC.

5 Licensing

The Xtended Services Platform has its own license. This license is such that it can contain multiple licenses to allow the running of Cisco BroadWorks applications.

The Xtended Services Platform does not start if it is not properly licensed.

©2019 CISCO SYSTEMS, INC.

6 Installation, Upgrade, and Patching

The installation procedure for the Xtended Services Platform is similar to the one used for existing Cisco BroadWorks servers. For detailed information and procedures on installing the Xtended Services Platform, see the *Cisco BroadWorks Software Management Guide* [1].

The upgrade and patching process for the Xtended Services Platform works the same way as for other Cisco BroadWorks servers. However, the Cisco BroadWorks and web applications hosted by the Xtended Services Platform are handled differently from the Xtended Services Platform itself. The following subsection provides more information.

6.1 Server Upgrade

The upgrade of the Xtended Services Platform is similar to the other Cisco BroadWorks servers except for the activation. Compared to the other servers where the activation is done in two steps (setting the target version and resetting), activating an Xtended Services Platform is done in a single step, that is, setting the active software version. As soon as the set command is executed and accepted, the server is stopped and its version is switched to the specified version. Finally, the server is started with the specified version. The following example shows the set command.

```
XSP_CLI/Maintenance/ManagedObjects> help set
This command is used to modify the Managed Object (MO)-related
attributes.

Parameters description:
forceOption          : This parameter determines the force option. Use
the
                       "force" option to apply the change immediately.
                       Otherwise, a restart of the application is
required for
                       the changes to apply.
option               : This parameter determines the options.
activeSoftwareVersion: This parameter specifies the active software
version.
type                 : This nested parameter determines the version of
the
                       server. Use this option when a different version
of the
                       server needs to be activated. The software
automatically
                       determines if it needs to upgrade or rollback.
server               : This parameter specifies server-related
information.
identity             : This nested parameter specifies the name of the
server
                       identity  to set to "active". It is usually the
short
                       name of a BroadWorks server (for example, Profile
                       Server, Network Server, and so on).
version              : This nested parameter specifies the version of the
                       software to activate.
revert               : This parameter provides the ability to revert to a
                       previous release. Use the "revert" option to
perform a
                       revert to the specified previous release.
Otherwise, a
```


©2019 CISCO SYSTEMS, INC.

```
                        rollback to the specified previous release is
performed.
backupLocation       : This nested parameter specifies the full path of
the
                        database backup file to use when performing a
revert.

=========================================================================
set
    [<forceOption>, Choice = {force}]
    <option>, Choice = {activeSoftwareVersion}
        activeSoftwareVersion:
            <type>, Choice = {server}
                server:
                    <identity>, String {1 to 80 characters}
                    <version>, String {1 to 80 characters}
                    [<revert>, Choice = {revert}]
                        revert:
                            [<backupLocation>, String {0 to 80
characters}]


XSP_CLI/Maintenance/ManagedObjects> set activeSoftwareVersion server XSP
23.0_1.1015

+++ WARNING +++ WARNING +++ WARNING +++
This command will change the active software version of XSP to
23.0_1.1015. NOTE that this action will cause downtime.
 Continue?

Please confirm (Yes, Y, No, N): y
BroadWorks SW Manager checking XSP server version 23.0_1.1015...
```

NOTE: Server upgrade shall be performed during maintenance window.

6.2 Applications Installation, Upgrade, and Patching

Although the Xtended Services Platform is an open platform for which applications can be deployed as required, the Xtended Services Platform installation does include/embed some Cisco BroadWorks and web applications. For more information on web applications, see the BroadWorks Xtended program at http://www.broadsoft.com/xtended/index.htm and the BroadWorks Xtended Developers Program at http://developer.broadsoft.com. All applications are automatically installed during the Xtended Services Platform installation process. Other applications can be installed by the operator via the CLI. For more information, see section *8.7 Application Management*.

©2019 CISCO SYSTEMS, INC.

The following table provides information regarding packaged applications.

Name	Type	Version	Upgrade Mode	Default State
AuthenticationService	BW	23.0	Automatic	Installed
BWCallCenter	Web	23.0.16	Manual	Installed
BWReceptionist	Web	23.0.16	Manual	Installed
BroadworksDms	BW	23.0	Automatic	Installed
BwCmProxy	BW	23.0	Automatic	Installed
CommPilot	BW	23.0	Automatic	Installed
CommPilot-XS-TAS	BW	23.0	Automatic	Installed
CustomMediaFilesRetrieval	BW	23.0	Automatic	Installed
DeviceManagementTFTP	BW	23.0	Automatic	Installed
LoadBalancer	BW	23.0	Automatic	Installed
LogServer	BW	23.0	Automatic	Installed
ModeratorClientApp	Web	23.0.3	Manual	Installed
NotificationPushServer	Web	23.0	Automatic	Installed
OCIFiles	BW	23.0	Automatic	Installed
OCIOverSoap	BW	23.0	Automatic	Installed
OpenClientServer	BW	23.0	Automatic	Installed
PublicECLQuery	BW	23.0	Automatic	Installed
PublicReporting	BW	23.0	Automatic	Installed
RatingFunction	BW	23.0	Automatic	Installed
SIPLocation	BW	23.0	Automatic	Installed
UCAPI	BW	23.0	Automatic	Installed
UserLocation	BW	23.0	Automatic	Installed
WebContainer	BW	23.0	Automatic	Deployed
webrtcFileServer	Web	20.0.585	Manual	Installed
Xsi-Actions	BW	22.0	Automatic	Installed
Xsi-Actions-XS-TAS	BW	22.0	Automatic	Installed
Xsi-Events	BW	22.0	Automatic	Installed
Xsi-Events-XS-TAS	BW	22.0	Automatic	Installed
Xsi-MMTel	BW	22.0	Automatic	Installed
Xsi-MMTel-XS-TAS	BW	22.0	Automatic	Installed
Xsi-VTR	BW	22.0	Automatic	Installed

©2019 CISCO SYSTEMS, INC.

The Xtended Services Platform supports two upgrade modes for application, automatic and manual.

- Automatic upgrade is used for applications that are completely controlled by Cisco BroadWorks. These applications are only embedded in Cisco BroadWorks installation packages or patches. They cannot be manually installed or uninstalled via *bwar/war* files. They are automatically upgraded by the Cisco BroadWorks upgrade process. The only non-automated task is the decision to activate/deactivate or deploy/undeploy the application, which is still made by the operator.
- Manual upgrade is used for all other applications that are individually distributed as *bwar/war* files by Cisco or through the Xtended Developer's program. These applications can be freely installed, activate, deployed, undeployed, deactivate and uninstalled by the operator. These applications are not automatically managed by the upgrade process, that is, upgrading Cisco BroadWorks does not affect the installed/deployed application.

The application modes cannot be modified through the CLI. Only Cisco can provide applications in automatic mode.

Two application types are also supported: Cisco BroadWorks (BW) application and web application.

- Cisco BroadWorks applications are packaged as a BroadWorks ARchive (*.bwar*) file. These applications are always packaged with the Xtended Services Platform and are always automatically installed on the server. Their upgrade mode is set to "Automatic" as these applications are upgraded at the same time as the Xtended Services Platform is upgraded.
- Web applications are packaged as Web ARchive (*.war*) file. Some of these applications are packaged with the Xtended Services Platform and these are installed by default. Other web applications are available through the Xtended Developer's program.

6.3 Additional Installation Tips

Best practices dictate that an operator should provision a firewall between the public Internet and the Xtended Services Platforms to facilitate tracking of resources, such as the number of connections per IP, long connections to Apache, and so on.

©2019 CISCO SYSTEMS, INC.

7 Xtended Services Platform Components

The Xtended Services Platform hosts several active components that provide internal and external services.

7.1 Core Daemons

The components described in this section are Cisco BroadWorks server daemons that are started by the operating system service initialization script when the server boots. They are only stopped when the operating system stops. As such, these daemons are started when the operating system enters run level 3, 4, or 5 (that is, normal operation). They are terminated when the operating system leaves these levels (for example, shutdown).

Manual control of the daemon's life cycle is neither required nor expected but can be accomplished using the commands specified in the following sections.

7.1.1 Software Manager

The Software Manager is part of the core components of the Xtended Services Platform. It is responsible for the patching of the Cisco BroadWorks applications and platform. It is also responsible for the installation, activation, and deployment of Cisco BroadWorks applications and web applications. The Software Manager is a critical component that must run for an operator to perform management and maintenance of the Cisco BroadWorks server.

7.1.1.1 Life Cycle Control

While it is not recommended during normal operation, it is possible to restart the Software Manager during the maintenance window. To stop and start it, the following commands can be used.

```
bwadmin@mtl64lin12 $ stopswman.pl

Stopping SW Manager version: 751259
bwadmin@mtl64lin12 $ startswman.pl

Starting SW Manager version: 751259
bwadmin@mtl64lin12 $
```

7.1.1.2 Logs

The Software Manager logs are located in directory */var/broadworks/logs/swmanager*.

7.1.2 SNMP Agent

The SNMP Agent is a core component of the Xtended Services Platform. It is not started and stopped with Cisco BroadWorks. It is running all the time. The SNMP Agent exposes the application counters and gauges to the management system. When a new application (Cisco BroadWorks or web) is installed on the Xtended Service Platform, its counters and gauges are automatically added to those exposed by the SNMP Agent. On the other end, when an application is uninstalled, its counters and gauges are removed.

©2019 CISCO SYSTEMS, INC.

7.1.2.1 Life Cycle Control

While it is not recommended during normal operation, it is possible to restart the SNMP Agent during the maintenance window. The life cycle of the SNMP Agent is controlled with the snmpdctl script located in */usr/local/broadworks/bw_base/bin*. The script supports the following options:

- start: Starts the SNMP Agent
- stop: Stops the SNMP Agent
- status: Shows the current SNMP Agent's state (*inService* or *dead*)
- restart: Stops and restarts the SNMP Agent

Following is an example of the usage.

```
bwadmin@mtl64lin12 $ snmpdctl
Usage: snmpdctl {start|stop|restart|status}
bwadmin@mtl64lin12 $ snmpdctl stop
Shutting down bwsnmpd
bwadmin@mtl64lin12 $ snmpdctl start
Executing transform...   [ok]

Starting bwsnmpd:        [ok]
bwadmin@mtl64lin12 $
bwadmin@mtl64lin12 $ snmpdctl status
bwsnmpd: inService
```

NOTE: The statistics collected by the SNMP Agent are lost when the SNMP Agent is stopped.

7.1.2.2 Logs

The SNMP Agent logs are located in the */var/broadworks/logs/snmp* directory.

7.1.3 License Manager

The License Manager (LicenseManager) is a core component of the Xtended Services Platform. It runs all the time, regardless of whether Cisco BroadWorks is started or stopped. The License Manager is started initially during the installation process but more generally, the UNIX *initd* starts and stops it at boot-time and shutdown.

The LicenseManager is responsible for managing the license files locally, getting the license files from the Network Function Manager (NFM) server, and requesting license permission usage to the Network Function Manager.

The License Manager must be running for the applications to start. If the License Manager is down or stopped and you attempt to start Cisco BroadWorks (startbw), the application does not start.

©2019 CISCO SYSTEMS, INC.

7.1.3.1 Life Cycle Control

While it is not recommended during normal operation, it is possible to restart the License Manager during the maintenance window. The life cycle of the License Manager is controlled with the lmdctl script located in */usr/local/broadworks/bw_base/bin*. The script supports the following options:

- start: Starts the License Manager.
- stop: Stops the License Manager.
- status: Shows the current License Manager's state (*inService* or *dead*).
- restart: Stops and restarts the License Manager.

Following is an example of the usage.

```
bwadmin@mtl64lin12 $ lmdctl
Usage: lmdctl {start|stop|restart|status}
bwadmin@mtl64lin12 $ lmdctl stop
Shutting down lmd...            [ok]
bwadmin@mtl64lin12 $ lmdctl start
Executing transform...   [ok]

Starting lmd...                 [ok]
bwadmin@mtl64lin12 $
bwadmin@mtl64lin12 $ lmdctl status
lmd: inService
```

7.1.3.2 Logs

The License Manager logs are located in the */var/broadworks/logs/lmd* directory.

7.1.4 Configuration Agent

The Configuration Agent is responsible for managing the server's configuration. It hosts the configuration data and is interfaced by other Cisco BroadWorks components that need to read and/or write configuration data. The Configuration Agent is a critical component that must run for other components to operate properly. It provides local services to other components of the server and external services accessed by the Cisco BroadWorks EMS.

The Configuration Agent communicates using the Network Configuration (NETCONF) protocol. This interface complies with *RFC 6241 Network Configuration Protocol (NETCONF)*. For more information, see the *Cisco BroadWorks Configuration System Interface Specification* [6].

7.1.4.1 Life Cycle Control

It is not recommended to stop the Configuration Agent during normal operation. The Configuration Agent is used by many platform components and applications and when stopped, these components and applications may misbehave. When required, it can be stopped during the maintenance window using the configdctl script provided in */usr/local/broadworks/bw_base/bin*. The script supports the following options:

- start: Starts the Configuration Agent.
- stop: Stops the Configuration Agent if Cisco BroadWorks applications are not running.
- stopnow: Stops the Configuration agent even if Cisco BroadWorks applications are running.

©2019 CISCO SYSTEMS, INC.

- status: Shows the current Configuration Agent's state (*inService* or *dead*).
- restart: Stops and restarts the Configuration Agent.
- validate: Checks if the config.xml file is valid.

Following is an example of the usage.

```
bwadmin@mtl64lin12 $ configdctl
Usage: configdctl {start | stop | stopnow | restart | status | validate}
bwadmin@mtl64lin12 $ configdctl stop
Shutting down configd...     [ok]
bwadmin@mtl64lin12 $ configdctl stopnow
Shutting down configd...     [ok]
bwadmin@mtl64lin12 $ configdctl start
Setting system info values
Patching configuration...       [ok]

Executing transform...   [ok]

Starting configd...          [ok]
bwadmin@mtl64lin12 $ configdctl status
configd: inService
bwadmin@mtl64lin12 $ configdctl validate
Validating ...  [ok]
```

7.1.4.2 Logs

The Configuration Agent issues logs to the */var/broadworks/logs/config* directory.

7.1.5 Platform Processes

On the Xtended Services Platform, the showrun command always displays two categories of processes: the application and the platform processes. Following is an example of the showrun command output.

```
bwadmin@mtl64lin12.mtl.broadsoft.com$ showrun

Currently running BroadWorks Application processes:

  tomcat process monitor (pid=3333)
  tomcat (pid=3498)

Currently running BroadWorks Platform processes:

  BroadWorks Configuration Agent process monitor (pid=1610)
  BroadWorks Configuration Agent (pid=1633)
  BroadWorks License Manager process monitor (pid=31971)
  BroadWorks License Manager (pid=31994)
  BroadWorks SNMP Agent process monitor (pid=31534)
  BroadWorks SNMP Agent (pid=31557)
  BroadWorks Software Manager (pid=30685)
```

The platform processes must always be running for the Cisco BroadWorks applications to run properly. The healthmon command monitors these processes and reports any missing platform processes.

©2019 CISCO SYSTEMS, INC.

7.2 Controllable Applications

The components described in this section are Cisco BroadWorks services that can be started and stopped by the operator. Deployed applications typically start when Cisco BroadWorks starts, and stop when Cisco BroadWorks stops.

7.2.1 WebContainer

The Xtended Services Platform deploys a WebContainer application to host the web applications. The WebContainer is built around the Tomcat application server.

The WebContainer application is like any other automatic Cisco BroadWorks applications, that is, it can be activated, deployed, undeployed, and deactivated.

7.2.1.1 WebContainer Software Components

The following figure fragments the Web Container into software components pertinent for the metrics it provides. This section provides only a summary in the form of metric groups. This representation is reused in sequence diagrams that explain the key metrics in detail. For a complete description, see section *8.6.1 Metric Groups*.

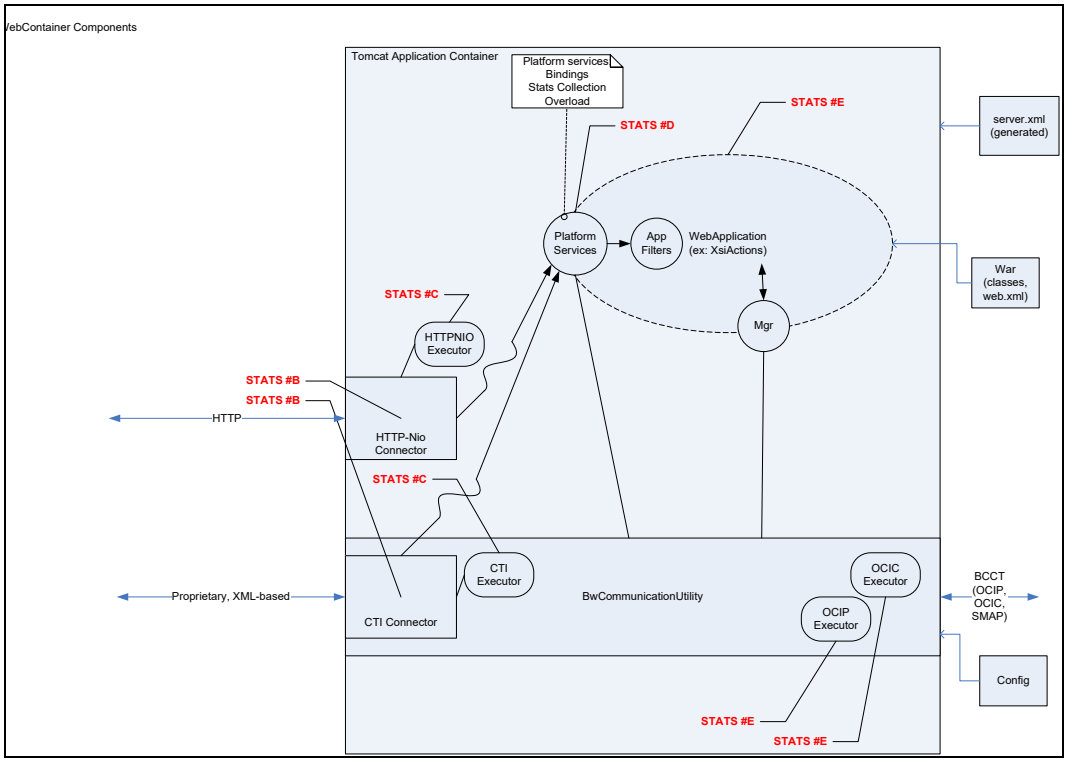

Figure 4 Web Container Assembly

The BwCommunicationUtility provides common platform-level services that webapps can use to interact with Cisco BroadWorks services. It provides a simplified API that abstracts details of the underlying communication infrastructure. Each webapp uses a dedicated instance of the BwCommunicationMgr. The webapp can also instantiate and configure request processing filters that are provided with the BwCommunicationUtility.

All incoming HTTP traffic is processed by Tomcat's specialized HttpNio connector. The connector uses the associated executor (queue/threadPool) to process all requests. Applications can use blocking or non-blocking processing, whichever is better suited to

©2019 CISCO SYSTEMS, INC.

their needs. The blocking processing consumes a thread for the duration of the request processing, up until the response is transmitted. The non-blocking processing uses threads sporadically as needed up until the response is transmitted.

The CTIConnector is a proprietary connector that allows use of the webapps via an XML-based protocol used for Computer Telephony Integration (CTI) integration with third-party systems.

7.2.1.2 WebContainer Metric Groups

As indicated in the previous figure, performance metrics are collected at various points within the WebContainer.

1) Connector processing metrics (Stats #B): These metrics are reported per connector type and provide total processing time (cumulative) and max processing time (single request).
2) Executor processing metrics (Stats #C and #E): These metrics are reported per executor type and provide measurements about queue delay and task processing time.
3) Platform Services metrics (Stats #D): These filters provide global and webapp-level metrics about overload protection, overhead processing time, and bindings.

7.2.1.3 Logs

The Tomcat application container used by the Xtended Services Platform issues logs in the */var/broadworks/logs/tomcat* directory. This log contains the container internal logs and possibly the exceptions it encounters while executing the various web applications.

The general settings for Tomcat allow configuring the log level. For complete details, see section *8.5.1.2 Tomcat*.

©2019 CISCO SYSTEMS, INC.

8 Configuration and Management

8.1 Configuration Modes

The Xtended Services Platform supports configuration locally through the CLI.

It is not all the configurations of a node that have been integrated in the Configuration Agent. Some platform components still rely on configuration files. The following are always managed locally:

- Maintenance task configuration
- Software Manager configuration

8.2 Profile Tuning

The profile tuning provides a list of predefined profile definitions based on a set of identified deployment models. This flexibility eases the tailoring of a server to meet its expected usage.

The Xtended Services Platform renders a CLI level for the profile tuning under the *XSP_CLI/System/ProfileTuning/GeneralSettings* level.

The profile tuning has the following characteristics.

- This functionality is optional. That is, a server functions normally without an assigned profile.
- The configuration action for a selected profile is immediate: when an administrator selects a profile, the system sets immediately the parameters with the values fetched from the profile.
- Only one profile is active at any given time. That is, the profiles are not stackable.
- The selected profile can be changed at any time. This action is immediate and supersedes all previously tailored values.
- The selected profile is reloadable at any time. Such action may be used to restore a server to the initial profile values (thus eliminating manual customization for the specific parameters included in the reloaded profile).
- When a profile includes webapp centric data, the profile values apply upon deploying a webapp even if the webapp was not present at the time of selecting a profile.
- This is the only mechanism available for tuning the Java Garbage Collection (GC) parameters.
- It is not possible to revert or undo a profile. The CLI only indicates the profile that is currently applied. To go back to a known profile, clear the existing profile and re-apply the desired profile.

8.2.1 Profile Characterization

The Tomcat HttpNio executors are the primary terminations for all incoming HTTP connections that include both blocking and streaming requests. The number of processing threads allocated to Tomcat encompasses both traffic types and reflects the expected usage for each.

The following table captures the targeted characteristics when identifying the needs for a profile. In an effort to ease the documentation, the table includes only groups of parameters and targeted characteristics rather than a list of specific parameters and values.

©2019 CISCO SYSTEMS, INC.

Profile Type			Non Real-Time	Real-Time
Server Type	**Parameter Group**	**Input parameters**	**Targeted Content**	
XSP	Garbage Collection (tomcat)	▪ Memory ▪ CPU Counts	▪ Disable concurrent GC. ▪ Use values similar to existing defaults.	▪ Enables concurrent GC (parallel GC threads @ 100% of CPU Counts) ▪ Same value for heap min and heap max ▪ Fixed new size generation @ 100% overall heap size
	Tomcat	None	Use values similar to existing defaults.	Targeted parameters: ▪ Global session timeout ▪ Per webapp session timeout ▪ Max threads per connector type

Based on these characteristics, the following profiles are available:

- Xsp non real-time
- Xsp real-time
- A default profile that corresponds to the factory config settings is also available

8.2.2 CLI Level

The Profile Tuning CLI level provides the typical get and apply commands. In addition, it provides a command named *describe* that prints the list of changes expected from applying a specific profile. When the describe command is used, there is no configuration change applied to the system. This is the key difference between the describe and apply commands.

Following is an example of describing the real-time profile tuning.

```
XSP_CLI/System/ProfileTuning/GeneralSettings> help describe
This command is used to print the configuration changes performed by a
profile. The printed output lists the difference between the current
values stored within the configuration and the new values that would be
set by applying the profile. No configuration changes are applied.

Parameters description:
profileTuningName: This parameter specifies the name of the selected
profile tuning.

========================================================================
describe
    <profileTuningName>, Choice = {default, realTime, nonrealtime, xsi}

XSP_CLI/System/ProfileTuning/GeneralSettings> describe realTime
This is a description of the realTime profile. The configuration is not
changed by this command.
/Applications/WebContainer/Tomcat/Executors/CTI/ThreadPool
Modifies,   min: 200 to 400
Modifies,   max: 200 to 400
/Applications/WebContainer/Tomcat/Executors/HTTPNio/ThreadPool
Preserves,  min: 166
Preserves,  max: 1666
/Applications/WebContainer/Tomcat/SessionManagement/Server
Modifies,   sessionTimeout: 1800 to 600
/Applications/WebContainer/Tomcat/SessionManagement/Webapps
Modifies,   sessionTimeout[Xsi-Actions]: <nil> to 600
/Interface/Http/GeneralSettings
Modifies,   keepAliveTimeout: 5 to 4
```


©2019 CISCO SYSTEMS, INC.

```
Preserves,  maxConnections: 1666
Preserves,  maxQueuedConnections: 833
Preserves,  pollerThreadCount: 2

The following values are not exposed through the CLI
Modifies,   /Applications/WebContainer/Tomcat GC: Parallel to Concurrent
Modifies,   /Applications/WebContainer/Tomcat heap size: Variable to
Fixed

No configuration changes have actually been performed.

XSP_CLI/System/ProfileTuning/GeneralSettings>
```

8.3 WebApplication Threading

Processing of HTTP requests within web applications is performed on threads provided by the underlying platform components (WebContainer and BWCommunicationUtility). Web Applications can have internal threads as needed but the main processing threads are those provided by the platform. The threading service is provided by the various "Executors" provided at the platform level and illustrated in the preceding Web Container Assembly figure.

A distinct executor is available at each input/output location. Each executor can be individually configured for the needs of the interface it serves. The WebContainer manages the following external interfaces: HTTPNIO, and CTI whereas the BWCommunicationUtility manages the Open Client Interface-Provisioning (OCI-P) and Open Client Interface-Call Control (OCI-C) interfaces. Refer to the following subsections for specific details about these executors.

8.3.1 General Concept

An executor is a means of binding and managing resources consumed when executing a collection of tasks. It is composed of a thread pool used to execute required tasks and a queue where tasks wait to be executed. To address the level of flexibility expected from the Xtended Services Platform, the following configuration parameters are supported for each executor:

- **minPoolSize**: defined as the minimum number of live threads (busy or idle) kept in the pool at any time (excluding startup transients).
- **maxPoolSize**: defined as the maximum number of live threads (busy and idle) allowed in the pool.
- **keepAliveTime**: defined as the amount of idle time before an excess thread is terminated.
- **queueCapacity**: defined as the maximum number of tasks that can be queued while waiting for other tasks to complete their execution. The capacity can be unlimited (left unspecified), disabled (0 capacity,) or fixed.

Based on the above parameters, the following approach is used when processing tasks (for example, processing an incoming http request):

- The actual pool size is automatically adjusted based on the **minPoolSize** and the **maxPoolSize**. When a new task is submitted and fewer than **minPoolSize** threads are running, the system allocates a new thread to handle the request, even if other worker threads are idle. If there are more than **minPoolSize** but less than **maxPoolSize** threads running, the system allocates a new thread only if the queue is full or disabled (as expressed by the actual queue size).

©2019 CISCO SYSTEMS, INC.

- Furthermore, if the pool currently has more than **minPoolSize** threads, excess threads will be terminated if they have been idle for more than the **keepAliveTime**. This provides a means of reducing resource consumption when the pool is not being actively used. If the pool becomes more active later, new threads will be allocated. Note that the *keep-alive* policy applies only when there are more than **minPoolSize** threads.

As a result of this behavior, the system rejects a request when the queue is full and the actual pool size equals **maxPoolSize**.

The queue capacity can be left unspecified, meaning unlimited queuing. In that case, the queuing capacity is implicitly limited by the total memory capacity available to the WebContainer. The queue capacity can be set to 0, effectively disabling the queuing function (the queue can be thought of as being always full). In that case, new tasks are immediately rejected if no thread is available in the pool and the pool size has already reached its **maxPoolSize**.

8.3.2 Typical Configurations

Although providing a great deal of configuration flexibility, executors are typically configured following two operational models:

- **Fixed threading**: The thread pool is fixed for optimum processing throughput; waiting is acceptable for momentary excess demand. [minPoolSize=maxPoolSize, queueCapacity > 1].
- **Adjustable threading**: The number of threads is adjusted with demand, waiting is not acceptable. [minPoolSize < maxPoolSize, queueCapacity=0].

8.3.2.1 Configuration through the CLI

Following is an example of configuring an OCI-C executor (for both the queue and thread pool).

```
XSP_CLI/System/CommunicationUtility/DefaultSettings/Executors/OCIC/Queue>
help set
The command is used to modify Queue settings.

Parameters description:
attribute: The name of an attribute to modify.
capacity : This parameter specifies the maximum queue capacity. Tasks are
           queued while waiting for an available thread. When the value is
           not set, queuing is unlimited and tasks are queued as long as
           there is enough memory available.

======================================================================
set
    <attribute>, Multiple Choice = {capacity}
        <capacity>, Integer {1 to 2147483647}

XSP_CLI/System/CommunicationUtility/DefaultSettings/Executors/OCIC/Queue> set
capacity 25000
*** Warning: Broadworks needs to be restarted for the changes to take effect
***

XSP_CLI/System/CommunicationUtility/DefaultSettings/Executors/OCIC/Queue> get
  capacity = 25000

XSP_CLI/System/CommunicationUtility/DefaultSettings/Executors/OCIC/Queue>
```

```
XSP_CLI/System/CommunicationUtility/DefaultSettings/Executors/OCIC/ThreadPool>
help set
This command is used to modify the thread pool settings.
```


©2019 CISCO SYSTEMS, INC.

```
Parameters description:
attribute    : The name of an attribute to modify.
min          : This parameter specifies the minimum number of threads to keep
               in steady state. Lower thread counts are possible during
               initialization.
max          : This parameter specifies the maximum number of threads this
pool
               can contain.
keepALiveTime: This parameter specifies the amount of idle time before an
               excess thread is terminated.

=========================================================================
set
    <attribute>, Multiple Choice = {min, max, keepALiveTime}
        <min>, Integer {1 to 10000}
        <max>, Integer {1 to 10000}
        <keepALiveTime>, Integer {1 to 3600}

XSP_CLI/System/CommunicationUtility/DefaultSettings/Executors/OCIC/ThreadPool> set
min 5 max 5 keepALiveTime 30
*** Warning: Broadworks needs to be restarted for the changes to take effect
***

XSP_CLI/System/CommunicationUtility/DefaultSettings/Executors/OCIC/ThreadPool> get
  min = 5
  max = 5
  keepALiveTime = 30

XSP_CLI/System/CommunicationUtility/DefaultSettings/Executors/OCIC/ThreadPool>
```

8.4 Configuration of BWCommunicationUtility

The BWCommunicationUtility provides the framework on which web applications can be built. It can be configured with default settings that are inherited by all web applications. However, web applications are not forced to use the default settings. Instead, they can override them with values that are more appropriate to their needs. The BWCommunicationUtility default settings are configurable through the CLI, under the *XSP_CLI/System/CommunicationUtility/DefaultSettings* level.

8.4.1 Integration Mode

The integration mode used to communicate with other Cisco BroadWorks servers is controlled by the *mode* attribute. The mode can be set to either “NS” or “AS”, where NS should be used when a Network Server is used to front end one or multiple Application Server clusters, and AS should be used when connecting to a single Application Server cluster without a Network Server.

In NS mode, the Network Server cluster address must be provided as well as the OCI-P/OCI-C ports to use. Note that in NS mode, all Application Server clusters must be using the same OCI-P port.

In AS mode, the primary and secondary Application Server addresses are configurable, as well as the Open Client Interface ports to use for OCI-P.

NOTE: OCI-C is not supported in this mode.

©2019 CISCO SYSTEMS, INC.

8.4.2 Provision to Secondary

The Xtended Services Platform can be configured to give preference to the secondary Application Server when sending OCI-P messages to an Application Server cluster. The *provisionToSecondary* attribute controls this behavior.

8.4.3 Communication Settings

The following settings affect the communication channels to other Cisco BroadWorks servers:

- *reconnectionTimerSecs* – The number of seconds to wait between reconnection attempts to a Cisco BroadWorks server.
- *responseTimeoutSecs* – The number of seconds to wait for a response from a Cisco BroadWorks Server after issuing a message.
- *useSecureBCCT* – Only secure BCCT connections are used for OCI-P and OCI-C.

8.4.4 Overflow Protection Settings

The BWCommunicationUtility allows rate protection per web application for user HTTP/HTTPS requests.

The rate is controlled by first setting the calculation period in seconds using the *transactionLimitPeriodSecs* attribute. Then, the number of requests allowed per user during the same period is determined by the *userTransactionLimit*.

For example:

- transactionLimitPeriodSecs = 10 seconds
- userTransactionLimit = 5
- user rate = 5 requests every 10 seconds

The rates are computed per web application.

Note that the section *8.5.4 Overload Protection* describes another mechanism that provides overload controls before authentication takes place.

8.4.5 Executors

Individual executors are available for OCI-C and OCI-P processing. Each Web App that requires an executor gets a distinct instance. However, all instances are configured using the same settings.

The recommended operating model for these executors is the fixed threading configuration: minPoolSize=maxPoolSize, with queueCapacity > 1.

These executors are configurable from the *XSP_CLI/System/CommunicationUtility/DefaultSettings/Executors* level.

8.4.6 External Authentication Support

Cisco BroadWorks user authentication can be delegated to an external entity, and the administrator has the choice of either Web-based Authentication Server (WAS), Kerberos, LDAP or RADIUS as the selected external authentication mechanism. The administrator has also the alternative of using an embedded agent.

For more information, see the *Cisco BroadWorks External Portal Integration Guide*.

©2019 CISCO SYSTEMS, INC.

8.4.7 Name Service

The Xtended Services Platform integrates the BroadWorks Name Service into the BWCommunicationUtility. Webapps and the Open Client Server (OCS) application also benefit from this mechanism since they go through the BroadWorks Name Service for name resolution.

8.4.7.1 NS Location URL

The NS location URL uses the following syntax.

```
scheme://hostname:[port]
```

When the NS location URL specifies HTTPS as the scheme, the Xtended Services Platform initiates a secured HTTP connection (using Transport Layer Security [TLS]/Secure Sockets Layer [SSL]) to the server. Otherwise, the Xtended Services Platform initiates an unsecured HTTP connection.

SRV lookup for the nslocation service is performed when the port is omitted from the configured locationServiceURL. When using this option, the locationServiceURL's hostname is taken as the domain name for an SRV lookup on the following service. The result from the lookup considers the weight and cost associated with the service.

```
_nslocation._tcp.<domain>
```

If the port is omitted from the configured locationServiceURL but no SRV records are found, the URL is used as-is with the default HTTP port (80) or HTTPS port (443), as determined by the URL scheme to perform an A/AAAA lookup.

Alternatively, to immediately perform an A/AAAA lookup and, consequently, prevent a Service Locator (SRV) lookup, the port needs to be specified explicitly in the URL.

The following example shows a URL that is resolved using SRV lookups.

```
https://nsdomain.com
```

When using this syntax, a lookup on the following SRV record occurs.

```
_nslocation._tcp.nsdomain.com SRV 1 10 1234 nsdomain1.broadsoft.com
_nslocation._tcp. nsdomain.com SRV 2 10 567 nsdomain2.broadsoft.com
```

These records result in the following URLs that reflect the weight and cost associated with the service.

```
https://nsdomain1.broadsoft.com:1234
https://nsdomain2.broadsoft.com:567
```

If no SRV records are found, the following URL is used instead to perform an A/AAAA lookup.

```
https://nsdomain.com:443
```

8.5 Configuration of WebContainer

The WebContainer CLI level allows for configuration of Tomcat. The configuration spans several distinct levels, which are documented in the following subsections.

©2019 CISCO SYSTEMS, INC.

8.5.1 General Settings

The WebContainer general setting is configurable through the CLI, under the level *XSP_CLI/Applications/WebContainer/Tomcat/GeneralSettings*. In particular, it allows configuring the following parameters:

- **uriEncoding**: This parameter specifies the character encoding for URI definition.
- **authenticationEncoding**: This parameter specifies the character encoding used to decode the username and password coming from the authenticate header when using basic authentication.
- **statisticsRefreshPeriod**: This parameter specifies the frequency (in seconds) at which Cisco BroadWorks fetches Performance Measurements (PMs) within Tomcat.
- **maxHttpHeaderSize**: This parameter specifies the limit (in bytes) on the allowed size of an HTTP request header field.
- **xFrameOptionsSameOrigin**: This parameter enables the addition of the X-Frame-Options HTTP Header (defined by *RFC 7034*) to all HTTP responses, with a value of "SAMEORIGIN". This is used to improve the protection of web applications against clickjacking.
- **jvmRoute**: This parameter specifies the name of this server instance that will be appended to the JSESSIONID in HTTP responses.

8.5.1.1 HTTP Strict Transport Security Headers

The WebContainer can be configured to add the HTTP Strict Transport Security (HSTS) header to its responses. This allows compatible user agents to enforce the use of secure connections, as specified in *RFC 6797*. The HSTS settings can be configured in the CLI under the level *XSP_CLI/Applications/WebContainer/Tomcat/GeneralSettings/HSTS*. For earlier releases, the same settings are available as container options. The following parameters are available:

- **enabled**: This parameter controls whether the HTTP Strict-Transport-Security header will be included in all HTTP responses.
- **maxAgeInSeconds**: This parameter specifies the amount of time in seconds the HTTP Strict-Transport-Security directive will be heeded by the connecting clients.
- **includeSubdomains**: This parameter controls whether the HTTP Strict-Transport-Security directive should be applied to all subdomains of the current domain by the connecting clients.

8.5.2 Logging

The log settings for the Tomcat Application container are configurable through the CLI under the *XSP_CLI/Applications/WebContainer/Tomcat/Logging* level. It uses the common Cisco BroadWorks logging mechanism for input and output channels.

The Tomcat input channel includes the following names:

- Generic
- NameService
- TomcatCore
- CtiConnector
- GcLog
- SMAP

©2019 CISCO SYSTEMS, INC.

- ExternalAuthenticator

By default, the Tomcat application container used by the Xtended Services Platform issues logs in the */var/broadworks/logs/tomcat* directory. This log contains the container internal logs and possibly the exceptions it encounters while executing the various web applications.

The Tomcat output channel follows the typical Cisco BroadWorks output channel pattern.

Following is an example of setting the TomcatCore log level to "FieldDebug".

```
XSP_CLI/Applications/WebContainer/Tomcat/Logging/InputChannels> h set
This command is used to modify input channels-related attributes.

Parameters description:
name     : This parameter specifies the name of the logging input channel.
attribute: The name of an attribute to modify.
enabled  : This parameter turns the logging to the logging input channel on
and
           off.
severity : This parameter specifies the severity level of the logging input
           channel.

=============================================================================
set
    <name>, Choice = {Generic, NameService, TomcatCore,
ExternalAuthenticator, CtiConnector, GcLog, SMAP}
    <attribute>, Multiple Choice = {enabled, severity}
        <enabled>, Choice = {false, true}
        <severity>, Choice = {Debug, FieldDebug, Info, Notice, Warn}


XSP_CLI/Applications/WebContainer/Tomcat/Logging/InputChannels> set
TomcatCore enabled true severity FieldDebug
*** Warning: BroadWorks needs to be restarted for the changes to take
effect ***

XSP_CLI/Applications/WebContainer/Tomcat/Logging/InputChannels> get
                    Name  Enabled    Severity
=============================================
                 Generic     true
             NameService     true        Info
              TomcatCore     true  FieldDebug
  ExternalAuthenticator     true        Info
           CtiConnector     true        Info
                  GcLog     true        Info
                   SMAP    false

7 entries found.

XSP_CLI/Applications/WebContainer/Tomcat/Logging/InputChannels>
```

8.5.2.1 HTTP Access Logs

The configuration related to access logs can be changed from the CLI level *XSP_CLI/Applications/WebContainer/Tomcat/AccessLogs*.

8.5.3 Executors

Individual executors are available for each access point:

- HTTPNIO: Blocking HTTP requests and NonBlocking HTTP requests used for streaming.

©2019 CISCO SYSTEMS, INC.

- CTI: CTI interface.

The recommended operating model for the HTTPNIO and CTI executors is the fixed threading configuration described in section *8.3.2 Typical Configurations.* (minPoolSize=maxPoolSize, with queueCapacity > 1).

These executors are configurable from the *XSP_CLI/Applications/WebContainer/Tomcat/Executors* level.

8.5.4 Process Metrics

The process metrics targets the Java Virtual Machine (JVM) metrics that is used by many Cisco BroadWorks servers.

The Java Virtual Machine (JVM) metrics provide several built-in statistics (using JVM MBeans) to report on key performance factors.

The Xtended Services Platform renders a CLI level for process metrics under the *XSP_CLI/Applications/WebContainer/Tomcat/JVMStatsCollector/GeneralSettings* level.

Data collection is controlled by setting the *enabled* attribute and the data frequency is controlled by setting the period in seconds using the *statisticsRefreshPeriod* attribute.

- **enabled**: This parameter enables or disables data collection for JVM statistics.
- **statisticsRefreshPeriod**: The frequency (expressed in seconds) at which Cisco BroadWorks fetches PMs from the JVM MBeans. The default value is 5 seconds. In addition to the regular polling, the system refreshes the PMs after every full garbage collection[1].

8.5.4.1 CLI Level

Following is an example of setting the refresh period for the JVM Stats Collector.

```
XSP_CLI/Applications/WebContainer/Tomcat/JVMStatsCollector/GeneralSettings>
help set
The command is used to modify JVM statistics collector general settings.

Parameters description:
attribute              : The name of an attribute to modify.
enabled                : This parameter enables or disables data collection
for
                         JVM statistics
statisticsRefreshPeriod: This parameter specifies the frequency at which
                         BroadWorks fetches PMs from JVM Mbeans

=========================================================================
set
    <attribute>, Multiple Choice = {enabled, statisticsRefreshPeriod}
        <enabled>, Choice = {false, true}
        <statisticsRefreshPeriod>, Integer {1 to 86400}

XSP_CLI/Applications/WebContainer/Tomcat/JVMStatsCollector/GeneralSettings>
XSP_CLI/Applications/WebContainer/Tomcat/JVMStatsCollector/GeneralSettings>
set enabled true statisticsRefreshPeriod 10
*** Warning: Broadworks needs to be restarted for the changes to take effect
***

XSP_CLI/Applications/WebContainer/Tomcat/JVMStatsCollector/GeneralSettings>
get
  enabled = true
  statisticsRefreshPeriod = 10
```

[1] There is a notification sent upon completion of a full garbage collection.

©2019 CISCO SYSTEMS, INC.

```
XSP_CLI/Applications/WebContainer/Tomcat/JVMStatsCollector/GeneralSettings>
```

8.5.5 Overload Protection

The BWCommunicationUtility allows rate protection at the Web Container level. It is registered globally and allows overload controls before authentication takes place. This mechanism provides both global and webapp-level protection.

The Xtended Services Platform renders a CLI level for the overload protection under the *XSP_CLI/Applications/WebContainer/Tomcat/OverloadProtection/Server* and *XSP_CLI/Applications/WebContainer/Tomcat/OverloadProtection/Webapps* levels.

The *Server* level controls the number of requests allowed globally and the *Webapps* level controls the number of requests allowed per webapp.

Each rate is controlled by first setting the calculation period in seconds using the p*eriod* attribute. Then the total number of requests allowed during this period is determined by the *limit* attribute.

For example:

- server.period = 10 seconds
- server.limit = 1000
- webapp.name = PublicReporting
- webapp.period = 10 seconds
- webapp.limit = 200
- global rate = 1000 requests every 10 seconds
- PublicReporting webapp rate = 200 requests every 10 seconds

8.5.5.1 CLI Level

Following is an example of limiting the number of request for the Xsi-Actions webapp.

```
1)  If the web application does not exist, add by:

XSP_CLI/Applications/WebContainer/Tomcat/OverloadProtection/Webapps> help
add
This command is used to add the web application overload protection
entry.

Parameters description:
name   : This parameter specifies the name of the web application.
period: This parameter specifies the duration expressed in seconds during
which
        the total number of transactions is limited for the overall
server.
limit : This parameter specifies the maximum number of requests to be
allowed
        during the specified period duration.

=========================================================================
add
```


©2019 CISCO SYSTEMS, INC.

```
    <name>, Choice = {AuthenticationService, BWCallCenter, BWOCTabs,
BWPresencePluginUpdate, BWReceptionist, BWSametime,
BWSametimeConnectUpdate, Bria-Webapp, BroadworksDms,
BusinessCommunicator, BwCmProxy, CommPilot, CommPilot-XS-TAS,
CustomMediaFilesRetrieval, ModeratorClientApp, NotificationPushServer,
OCIFiles, OCIOverSoap, PXSAastra, PhoneXtension, PublicECLQuery,
PublicReporting, UCAPI, UserLocation, Xsi-Actions, Xsi-Actions-XS-TAS,
Xsi-Events, Xsi-Events-XS-TAS, Xsi-MMTel, Xsi-MMTel-XS-TAS, Xsi-VTR,
webrtcFileServer}
    <period>, Integer {1 to 300}
    <limit>, Integer {1 to 65535}

XSP_CLI/Applications/WebContainer/Tomcat/OverloadProtection/Webapps> add
Xsi-Events 1 50

2)  If the web application exists, modify it by:

XSP_CLI/Applications/WebContainer/Tomcat/OverloadProtection/Webapps> help set
The command is used to modify per web application overload protection
settings.

Parameters description:
name      : This parameter specifies the name of the web application.
attribute: The name of an attribute to modify.
period    : This parameter specifies the duration expressed in seconds during
            which the total number of transactions is limited for the overall
            server.
limit     : This parameter specifies the maximum number of requests to be
            allowed during the specified period duration.

======================================================================
set
    <name>, Choice = {AuthenticationService, BWCallCenter, BWOCTabs,
BWPresencePluginUpdate, BWReceptionist, BWSametime, BWSametimeConnectUpdate,
Bria-Webapp, BroadworksDms, BusinessCommunicator, BwCmProxy, CommPilot,
CommPilot-XS-TAS, CustomMediaFilesRetrieval, ModeratorClientApp,
NotificationPushServer, OCIFiles, OCIOverSoap, PXSAastra, PhoneXtension,
PublicECLQuery, PublicReporting, UCAPI, UserLocation, Xsi-Actions, Xsi-
Actions-XS-TAS, Xsi-Events, Xsi-Events-XS-TAS, Xsi-MMTel, Xsi-MMTel-XS-TAS,
Xsi-VTR, webrtcFileServer}
    <attribute>, Multiple Choice = {period, limit}
        <period>, Integer {1 to 300}
        <limit>, Integer {1 to 65535}


XSP_CLI/Applications/WebContainer/Tomcat/OverloadProtection/Webapps>
XSP_CLI/Applications/WebContainer/Tomcat/OverloadProtection/Webapps> set Xsi-
Events period 1 limit 50
*** Warning: Broadworks needs to be restarted for the changes to take effect
***

XSP_CLI/Applications/WebContainer/Tomcat/OverloadProtection/Webapps> get
             Name  Period  Limit
================================
      Xsi-Actions       1    100
       Xsi-Events       1     50
        Xsi-MMTel       1    100

3 entries found.
```


©2019 CISCO SYSTEMS, INC.

8.5.6 Session Management

Session management allows specifying session timeouts globally and per web app (optionally).

The Xtended Services Platform renders a CLI level for the session management under the *XSP_CLI/Applications/WebContainer/Tomcat/ SessionManagement/Server* and *XSP_CLI/ Applications/WebContainer/Tomcat/ SessionManagement/Webapps* levels.

Data collection is controlled by setting the *sessionTimeout* attribute.

- **sessionTimeout**: a duration expressed in seconds after which an inactive session times out. The default is 30.

The parameter is configurable per individual webapp (identified by the display name). When set, the value supersedes the value specified globally for the webapp. When there are multiple instances of a given webapp, they all share the same settings. That is, the same session timeout setting applies to all instances.

- **sessionTimeout (webapp)**: a webapp-specific duration expressed in seconds after which an inactive session times out. By default, this parameter is empty for all configured webapps.

8.5.6.1 CLI Level

Following is an example of setting the Xsi-Actions timeout to “1800”.

```
1)  If the web application does not exist, add by:

XSP_CLI/Applications/WebContainer/Tomcat/SessionManagement/Webapps> help add
This command is used to add the web application session management entry.

Parameters description:
name           : This parameter specifies the name of the web application.
attribute      : Name of the webapp.
sessionTimeout: This parameter specifies a web application specific duration
                expressed in seconds after which an inactive session times
out.

======================================================================
add
    <name>, Choice = {AuthenticationService, BWCallCenter, BWOCTabs,
BWPresencePluginUpdate, BWReceptionist, BWSametime, BWSametimeConnectUpdate,
Bria-Webapp, BroadworksDms, BusinessCommunicator, BwCmProxy, CommPilot,
CommPilot-XS-TAS, CustomMediaFilesRetrieval, ModeratorClientApp,
NotificationPushServer, OCIFiles, OCIOverSoap, PXSAastra, PhoneXtension,
PublicECLQuery, PublicReporting, UCAPI, UserLocation, Xsi-Actions, Xsi-
Actions-XS-TAS, Xsi-Events, Xsi-Events-XS-TAS, Xsi-MMTel, Xsi-MMTel-XS-TAS,
Xsi-VTR, webrtcFileServer}
    [<attribute>, Multiple Choice = {sessionTimeout}]
        <sessionTimeout>, Integer {1 to 86400}

XSP_CLI/Applications/WebContainer/Tomcat/SessionManagement/Webapps> add Xsi-
Actions sessionTimeout 1200
*** Warning: BroadWorks needs to be restarted for the changes to take effect
***

2)  If the web application exists, modify it by:

XSP_CLI/Applications/WebContainer/Tomcat/SessionManagement/Webapps> help set
The command is used to modify per web application session management
settings.

Parameters description:
```


©2019 CISCO SYSTEMS, INC.

```
name          : This parameter specifies the name of the web application.
attribute     : The name of an attribute to modify.
sessionTimeout: This parameter specifies a web application specific duration
                expressed in seconds after which an inactive session times
out.

======================================================================
set
    <name>, Choice = {AuthenticationService, BWCallCenter, BWOCTabs,
BWPresencePluginUpdate, BWReceptionist, BWSametime, BWSametimeConnectUpdate,
Bria-Webapp, BroadworksDms, BusinessCommunicator, BwCmProxy, CommPilot,
CommPilot-XS-TAS, CustomMediaFilesRetrieval, ModeratorClientApp,
NotificationPushServer, OCIFiles, OCIOverSoap, PXSAastra, PhoneXtension,
PublicECLQuery, PublicReporting, UCAPI, UserLocation, Xsi-Actions, Xsi-
Actions-XS-TAS, Xsi-Events, Xsi-Events-XS-TAS, Xsi-MMTel, Xsi-MMTel-XS-TAS,
Xsi-VTR, webrtcFileServer}
    <attribute>, Multiple Choice = {sessionTimeout}
        <sessionTimeout>, Integer {1 to 86400}

XSP_CLI/Applications/WebContainer/Tomcat/SessionManagement/Webapps> set Xsi-
Actions sessionTimeout 1200
*** Warning: Broadworks needs to be restarted for the changes to take effect
***

XSP_CLI/Applications/WebContainer/Tomcat/SessionManagement/Webapps> get
         Name  Session Timeout
==============================
  Xsi-Actions             1200

1 entry found.
```

8.5.7 Thresholds

This section describes the list of configurable thresholds. For the list of SNMP traps generated upon a threshold crossing, see section *8.6.4 Thresholds*.

8.5.7.1 Executors

For each of the executor name (CTI and HTTPNIO), the Xtended Services Platform provides the ability to set the following thresholds.

Executor Queue Usage Ratio

The executor queue usage ratio is defined as a percentage using the following formula: EventQueueCount / queueCapacity * 100. By default, this threshold defines the arm level to be 90 and the clear level to be 81. This threshold is enabled.

The Xtended Services Platform renders a CLI level for this threshold under the *XSP_CLI/Applications/WebContainer/Tomcat/Executors/<executorName>/Queue/SizeThreshold* level.

The following is an example of setting the arm value to “88” and the reset value to “78” for the HTTPNio executor.

```
XSP_CLI/Applications/WebContainer/Tomcat/Executors/HTTPNio/Queue/SizeThreshold>
help set
This command is used to modify the size threshold settings.

Parameters description:
attribute : The name of an attribute to modify.
enabled   : This parameter determines whether a threshold is active.
armValue  : This parameter specifies a value for enabling a threshold crossing.
resetValue: This parameter specifies a value for clearing a threshold crossing.
```


©2019 CISCO SYSTEMS, INC.

```
=====================================================================
set
    <attribute>, Multiple Choice = {enabled, armValue, resetValue}
        <enabled>, Choice = {false, true}
        <armValue>, Integer {0 to 100}
        <resetValue>, Integer {0 to 100}

XSP_CLI/Applications/WebContainer/Tomcat/Executors/HTTPNio/Queue/SizeThreshold> set
armValue 88 resetValue 78
*** Warning: BroadWorks needs to be restarted for the changes to take effect ***

XSP_CLI/Applications/WebContainer/Tomcat/Executors/HTTPNio/Queue/SizeThreshold> get
  enabled = true
  armValue = 88
  resetValue = 78

XSP_CLI/Applications/WebContainer/Tomcat/Executors/HTTPNio/Queue/SizeThreshold>
```

Executor Queue Latency

The executor queue latency is defined as an absolute value and represents the number of requests that are currently queued while waiting for other tasks to complete during execution. By default, this threshold defines the arm level to be 2500 and the clear level to be 2000. This threshold is enabled.

The Xtended Services Platform renders a CLI level for this threshold under the *XSP_CLI/Applications/WebContainer/Tomcat/Executors/<executorName>/Queue/LatencyThreshold* level.

Executor Thread Pool Processing Time

The executor thread pool processing time is defined as an absolute value and represents the execution time taken by a task to complete. By default, this threshold defines the arm level to be 2500 and the clear level to be 2000. This threshold is enabled.

The Xtended Services Platform renders a CLI level for this threshold under the *XSP_CLI/Applications/WebContainer/Tomcat/Executors/<executorName>/ThreadPool/ProcessingTimeThreshold/* level.

Executor Thread Pool Busy Ratio

The executor thread pool busy ratio is defined as a percentage using the following formula: threadsBusy / maxPoolSize * 100. By default, this threshold defines the arm level to be 95 and the clear level to be 85. This threshold is enabled.

The Xtended Services Platform renders a CLI level for this threshold under the *XSP_CLI/Applications/WebContainer/Tomcat/Executors/<executorName>/ThreadPool/UsageThreshold/* level.

8.5.7.2 Process Metrics

Heap and Nonheap Memory Usage Ratios

The Xtended Services Platform provides the ability to set a threshold for the heap and nonheap memory usage ratios. By default, these thresholds define the arm level to be 90 and the clear level to be 81. These thresholds are enabled.

The heap memory usage ratio is defined as a percentage and represents the heap memory usage after a full garbage collection.

The nonheap memory usage ratio is defined as a percentage and represents the nonheap memory usage after a full garbage collection.

©2019 CISCO SYSTEMS, INC.

The Xtended Services Platform renders a CLI level for this threshold under the *XSP_CLI/Applications/WebContainer/Tomcat/JVMStatsCollector/HeapUsageThreshold* level.

The following is an example of setting the arm value to “88” and the reset value to “78” for the heap memory usage threshold.

```
XSP_CLI/Applications/WebContainer/Tomcat/JVMStatsCollector/HeapUsageThreshold> help
set
This command is used to modify the heap usage threshold settings.

Parameters description:
attribute : The name of an attribute to modify.
enabled   : This parameter determines whether a threshold is active.
armValue  : This parameter specifies a value for enabling a threshold crossing.
resetValue: This parameter specifies a value for clearing a threshold crossing.

=======================================================================
set
    <attribute>, Multiple Choice = {enabled, armValue, resetValue}
        <enabled>, Choice = {false, true}
        <armValue>, Integer {0 to 100}
        <resetValue>, Integer {0 to 100}

XSP_CLI/Applications/WebContainer/Tomcat/JVMStatsCollector/HeapUsageThreshold> set
armValue 88 resetValue 78
*** Warning: Broadworks needs to be restarted for the changes to take effect ***

XSP_CLI/Applications/WebContainer/Tomcat/JVMStatsCollector/HeapUsageThreshold> get
  enabled = true
  armValue = 88
  resetValue = 78

XSP_CLI/Applications/WebContainer/Tomcat/JVMStatsCollector/HeapUsageThreshold>
```

8.6 Management of WebContainer

The combination of flexible configurations and strong reporting capabilities allows for configuring specific thresholds with predefined values. Upon threshold crossing, the system generates SNMP traps to report this state to the network management layers.

The Xtended Services Platform allows enabling (or disabling) data collection and the frequency of these metrics as well as customizing thresholds.

From a network management layer, it is possible to collect the Performance Measurements (PMs) and receive threshold crossing notifications.

8.6.1 Metric Groups

For a review of the metric groups, see section *7.2.1.2 WebContainer Metric Groups*.

8.6.2 Sequence Diagrams

Using sequence diagrams, this section highlights key processing time metrics. They are organized by connector and executor types.

©2019 CISCO SYSTEMS, INC.

The HttpNio connector may expose processing time that varies widely. The key difference is the types of processing that the HttpNio connector handles since it supports both non-streaming web application and persistent HTTP connection with two-way HTTP streaming. For blocking request, the processing blocks until it transmits a response whereas for streaming request, the processing uses threads sporadically as needed up until the response is transmitted. For the CTI connector, the processing returns immediately after an event was sent since all the requests are non-blocking. The sequence diagrams also capture the implications of the OCI executor that is involved when receiving responses from other Cisco BroadWorks servers.

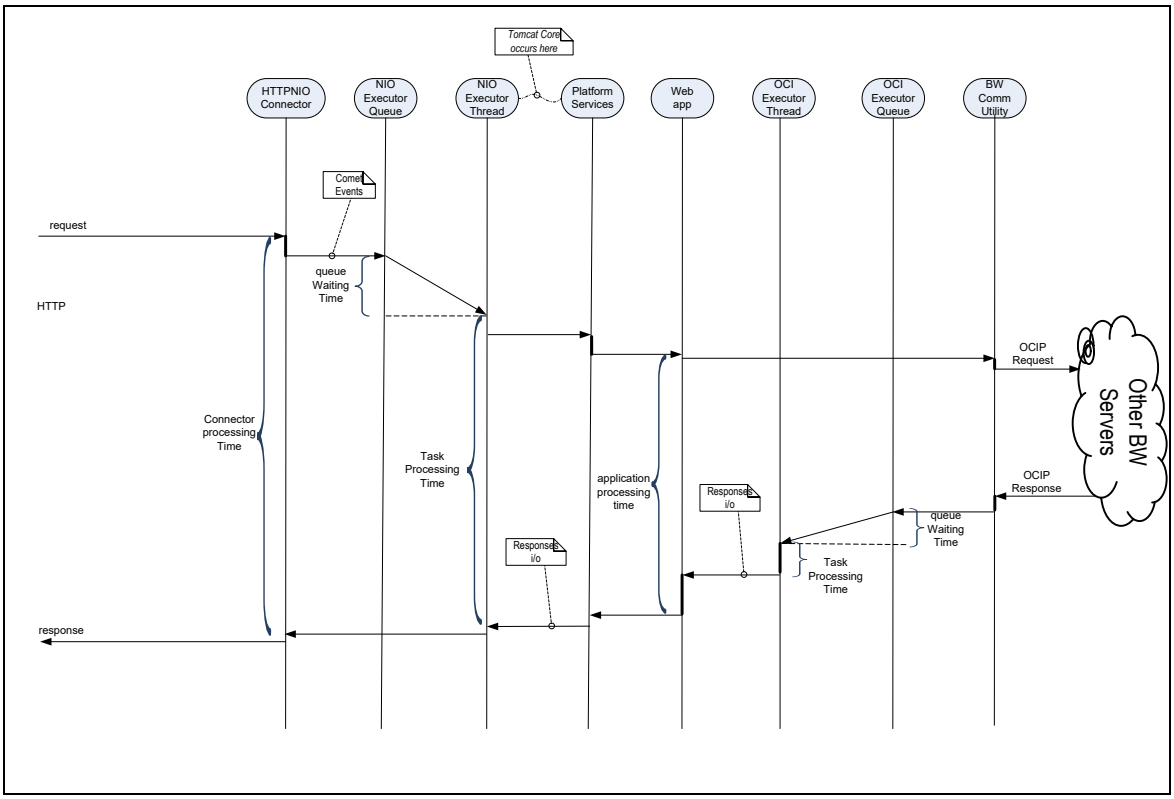

Figure 5 HTTP Blocking Sequence Diagram (HttpNiO)

©2019 CISCO SYSTEMS, INC.

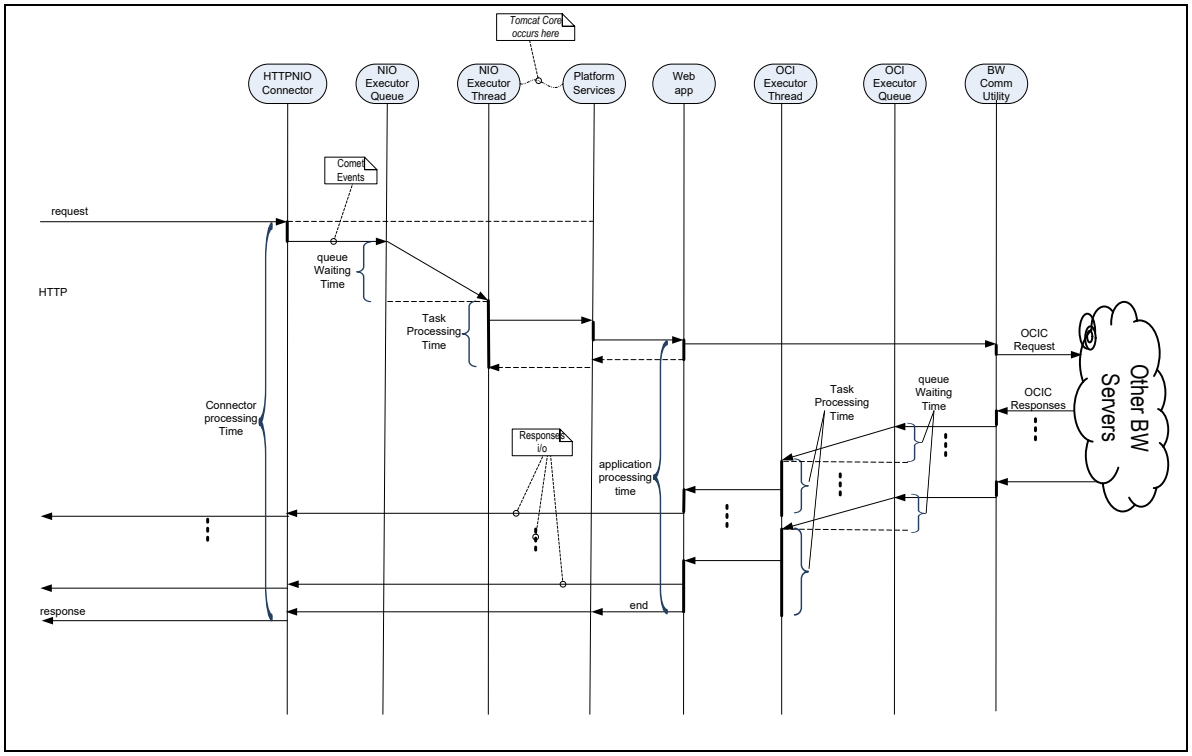

Figure 6 HTTP Streaming Sequence Diagram (HttpNio)

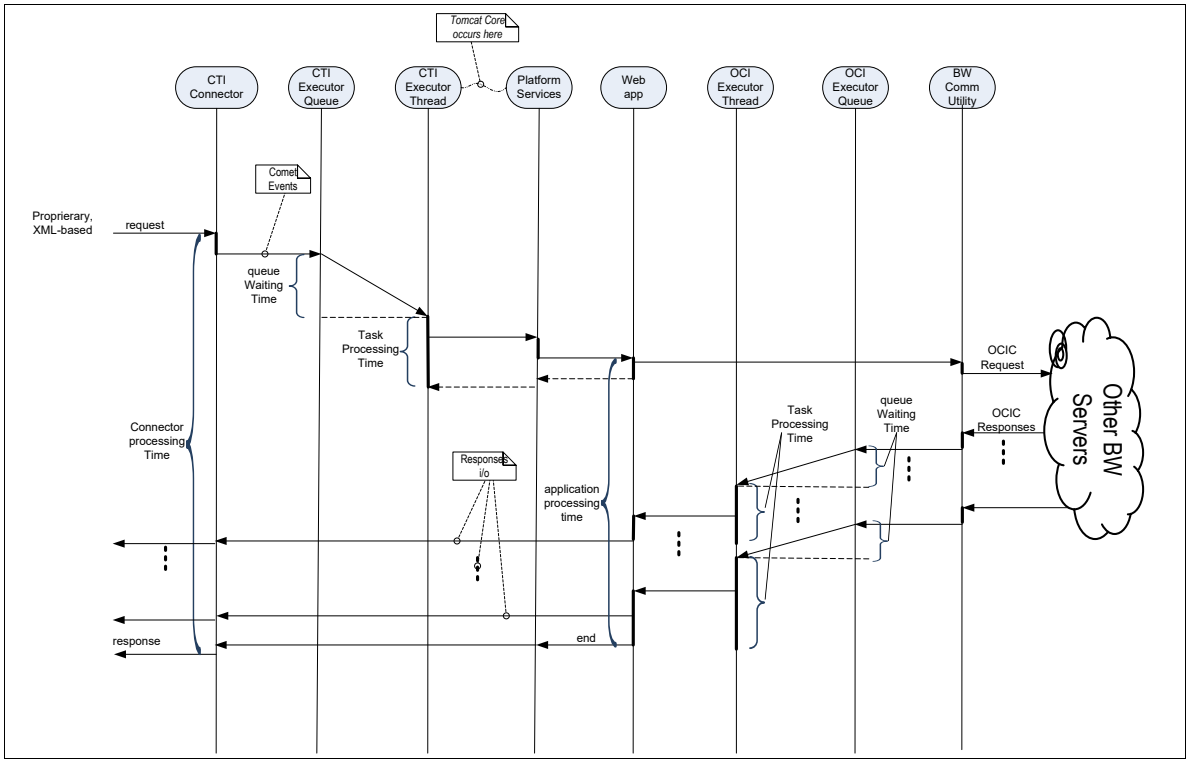

Figure 7 CTI Sequence Diagram

©2019 CISCO SYSTEMS, INC.

These diagrams highlight the following metrics. Note that only the object identifiers (OIDs) pertinent to the key metrics are presented here. For complete details, see the *Cisco BroadWorks Performance Measurements Interface Specification* [2].

8.6.2.1 Connector Processing Time

This statistic measures the processing time spent in the connector. The system does not report an instant value. Instead, it reports the cumulative time and the maximum value for an individual request. The *BW-WebContainer* MIB reports this statistic with the following OIDs:

- bwWebContainer.connectors.bwConnectorTotalProcessingTime
- bwWebContainer.connectors.bwConnectorMaxProcessingTime

These OIDs are available from the following CLI level.

```
XSP_CLI/Monitoring/PM/WebContainer> cd connectors;get
bwWebContainer/connectors/
------------------------------------------------------------------------------
bwWebContainer/connectors/
------------------------------------------------------------------------------
bwConnectorTable:

  (1) bwConnectorIndex
  (2) bwConnectorName
  (3) bwConnectorExecutorName
  (4) bwConnectorRequestCount
  (5) bwConnectorBytesRx
  (6) bwConnectorBytesTx
  (7) bwConnectorTotalProcessingTime
  (8) bwConnectorMaxProcessingTime

    (1)                          (2)       (3)     (4)     (5)      (6)     (7)     (8)
      1  [localhost/127.0.0.1]:8010  httpnio       0       0        0       0       0

XSP_CLI/Monitoring/PM/WebContainer>
```

8.6.2.2 Executors

The Queue Waiting Time and Task Processing Time are part of the executor metrics.

The Queue Waiting Time statistic reports the time spent waiting in the executor queue. A delay occurs when the executor threads are all busy processing other events. This metric is reported in a MIB table identified by the executor type. The *BW-WebContainer* MIB reports this statistic with the following OID.

- bwWebContainer.executors.executorTable.bwExecutorQueueWaitingTime

The Task Processing time statistic reports the processing time spent in the executor thread. This metric is reported in a MIB table identified by the executor type. The system does not report an instant value. Instead, it tracks the minimum, average, and maximum values. The *BW-WebContainer* MIB reports this statistic with the following OIDs:

- bwWebContainer.executors.executorTable.bwExecutorTaskProcessingTimeMin
- bwWebContainer.executors.executorTable.bwExecutorTaskProcessingTimeAvg
- bwWebContainer.executors.executorTable.bwExecutorTaskProcessingTimeMax

These OIDs are available from the following CLI level.

```
XSP_CLI/Monitoring/PM/WebContainer> cd executors;get
bwWebContainer/executors/
------------------------------------------------------------------------------
```


©2019 CISCO SYSTEMS, INC.

```
bwWebContainer/executors/
-------------------------------------------------------------------------------
    *bwExecutorReset           0
bwExecutorTable:

  (1) bwExecutorIndex
  (2) bwExecutorName
  (3) bwExecutorQueueCapacity
  (4) bwExecutorQueueSize
  (5) bwExecutorQueueWaitingTimeAvg
  (6) bwExecutorQueueWaitingTimeMin
  (7) bwExecutorQueueWaitingTimeMax
  (8) bwExecutorQueueWaitingTimeMaxTimestampMSB
  (9) bwExecutorQueueWaitingTimeMaxTimestampLSB
 (10) bwExecutorMinPoolSize
 (11) bwExecutorMaxPoolSize
 (12) bwExecutorThreadsBusy
 (13) bwExecutorCompletedTaskCount
 (14) bwExecutorThreadsAlive
 (15) bwExecutorThreadsAliveMax
 (16) bwExecutorTaskProcessingTimeAvg
 (17) bwExecutorTaskProcessingTimeMin
 (18) bwExecutorTaskProcessingTimeMax
 (19) bwExecutorTaskProcessingTimeMaxTimestampMSB
 (20) bwExecutorTaskProcessingTimeMaxTimestampLSB

(1)     (2)    (3) (4) (5) (6) (7) (8)         (9) (10) (11) (12) (13) (14) (15) (16) (17) (18) (19) (20)
  1 httpnio 20000   0   0   0   0   0          0  200  200    0    0    0    0    0    0    0    0    0

XSP_CLI/Monitoring/PM/WebContainer>
```

8.6.2.3 Application Processing Time

This statistic reports the processing time spent in the web application. The system does not report an instant value. Instead, it reports only the cumulative time. The *BW-WebContainer* MIB reports this statistic with the following OID:

- bwWebContainer.applications.applicationResources.bwApplicationResourceTotalProcessingTime

This OID is available from the following CLI level.

```
XSP_CLI/Monitoring/PM/WebContainer> cd applications/applicationResources;get
bwWebContainer/applications/applicationResources/
-------------------------------------------------------------------------------
bwWebContainer/applications/applicationResources/
-------------------------------------------------------------------------------
bwApplicationResourcesTable:

  (1) bwApplicationResourceIndex
  (2) bwApplicationResourceWebAppName
  (3) bwApplicationResourceRequestCount
  (4) bwApplicationResourceTotalProcessingTime

    (1)                          (2)     (3)     (4)
      1                         /dms       0       0
      2         /com.broadsoft.ccpri       0       0
      3   /com.broadsoft.xsi-events        0       0
      4                  /webservice       0       0
      5  /com.broadsoft.xsi-actions        4      41
      6                   /CommPilot    1672   52648
      7                    /ocifiles       0       0

XSP_CLI/Monitoring/PM/WebContainer> >
```


©2019 CISCO SYSTEMS, INC.

8.6.3 Process Metrics

The process metrics targets the Java Virtual Machine (JVM) metrics that is used by many Cisco BroadWorks servers.

The JVM metrics provide several built-in statistics (using JVM MBeans) to report performance on key performance factors.

The *BW-WebContainer* MIB reports this statistic with the following OID:

- bwWebContainer.processMetrics

8.6.3.1 Memory

The OIDs for heap and nonheap memories are available from the following CLI levels.

```
XSP_CLI/Monitoring/PM/WebContainer> cd processMetrics/memory/heap;get
bwWebContainer/processMetrics/memory/heap/
------------------------------------------------------------------------------
bwWebContainer/processMetrics/memory/heap/
------------------------------------------------------------------------------
                   bwProcessMetricsHeapInitSize   226492416
                    bwProcessMetricsHeapMaxSize   226492416
                       bwProcessMetricsHeapUsed    60406032
                    bwProcessMetricsHeapUsedMax    92134800
                  bwProcessMetricsHeapCommitted   226492416
     bwProcessMetricsHeapLastPostCollectionSize    60496664

XSP_CLI/Monitoring/PM/WebContainer>
```

```
XSP_CLI/Monitoring/PM/WebContainer> cd memory/nonheap;get
bwWebContainer/processMetrics/memory/nonheap/
------------------------------------------------------------------------------
bwWebContainer/processMetrics/memory/nonheap/
------------------------------------------------------------------------------
                   bwProcessMetricsNonHeapInitSize    24313856
                    bwProcessMetricsNonHeapMaxSize   318767104
                       bwProcessMetricsNonHeapUsed    34920144
                    bwProcessMetricsNonHeapUsedMax    38417936
                  bwProcessMetricsNonHeapCommitted    35192832
     bwProcessMetricsNonHeapLastPostCollectionSize           0

XSP_CLI/Monitoring/PM/WebContainer>
```

8.6.3.2 Thread

The OIDs for threads are available from the following CLI level.

```
XSP_CLI/Monitoring/PM/WebContainer> cd processMetrics/threads;get
bwWebContainer/processMetrics/threads/
------------------------------------------------------------------------------
bwWebContainer/processMetrics/threads/
------------------------------------------------------------------------------
        bwProcessMetricsThreadsAlive          33
     bwProcessMetricsThreadsAliveMax          34
      bwProcessMetricsThreadsStarted       37911


XSP_CLI/Monitoring/PM/WebContainer>
```


©2019 CISCO SYSTEMS, INC.

8.6.4 Thresholds

This section describes the list of SNMP traps generated upon a threshold crossing. For the list of thresholds whose default values are configurable, see section *8.5.6 Thresholds*.

8.6.4.1 Executors

For each of the executor name (CTI, and HTTPNIO), the Xtended Services Platform provides the ability to generate the following traps:

Executor Queue Usage Ratio

The executor queue usage ratio is defined as a percentage using the following formula: EventQueueCount / queueCapacity * 100. Upon threshold crossing, the system generates the following trap:

- Trap **bwExecutorQueueUsageExceeded** (of severity high): the usage ratio of the executor queue exceeded the threshold level.

Executor Queue Latency

The executor queue latency is defined as an absolute value and represents the number of requests that are currently queued while waiting for other tasks to complete during execution. Upon threshold crossing, the system generates the following trap:

- Trap **bwExecutorQueueLatencyExceeded** (of severity high): The executor queue latency exceeded the threshold level.

Executor Thread Pool Processing Time

The executor thread pool processing time is defined as an absolute value and represents the execution time taken by a task to complete. Upon threshold crossing, the system generates the following trap:

- Trap **bwExecutorThreadPoolProcessingTimeExceeded** (of severity high): the executor thread pool processing time exceeded the threshold level.

Executor Thread Pool Busy Ratio

The executor thread pool busy ratio is defined as a percentage using the following formula: threadsBusy / maxPoolSize * 100. Upon threshold crossing, the system generates the following trap:

- Trap **bwExecutorThreadPoolBusyExceeded** (of severity high): The executor thread pool busy ratio exceeded the threshold level.

8.6.4.2 Process Metrics

Heap and Nonheap Memory Usage Ratios

The Xtended Services Platform provides the ability to set a threshold for the heap and nonheap memory usage ratios.

The heap memory usage ratio is defined as a percentage and represents the heap memory usage after a full garbage collection.

The nonheap memory usage ratio is defined as a percentage and represents the nonheap memory usage after a full garbage collection.

Upon a threshold crossing, the system generates the following traps:

- Trap **bwHeapMemoryUsageExceeded** (of severity high): the memory usage ratio after a full garbage collection (in %) exceeded the threshold level.

©2019 CISCO SYSTEMS, INC.

- Trap **bwNonheapMemoryUsageExceeded** (of severity high): the nonheap memory usage ratio after a full garbage collection (in %) exceeded the threshold level.

8.6.4.3 Overload Protection

The Xtended Services Platform raises an SNMP trap when the system starts rejecting requests. For these traps, the threshold setting is embedded in the definition of the protection mechanism. Hence, there is no additional threshold parameter required. These traps are notifications and do not clear automatically.

- Trap **bwServerTransactionLimitExceeded**: during the last overload protection period, the number of incoming requests exceeded the prescribed limit for the whole server.
- Trap **bwWebAppTransactionLimitExceeded**: during the last overload protection period, the number of incoming requests exceeded the prescribed limit for a specific web app (as identified by its name).
- Trap **bwUserTransactionLimitExceeded**: during the last overload protection period, the number of incoming requests exceeded the prescribed limit for a specific user.

8.7 Application Management

Operators can install, activate, and deploy new applications on the Xtended Services Platform. The Xtended Services Platform provides application management functionality through the CLI, under the *Maintenance/ManagedObjects* level. Applications can be installed, uninstalled, activated, deactivated, deployed, and undeployed. All operations are CLI-driven. No manual operations are required.

Applications must be packaged using two possible formats, the standard web application archive (*.war*) format and the Cisco BroadWorks Application format (*.bwar*). The web application must follow specific guidelines documented in section *11 Xtended Services Platform Web Applications Requirements* so that they can be installed on the Xtended Services Platform. The Cisco BroadWorks applications are created and released by Cisco/BroadSoft.

The full life cycle of an application is:

Installation → activation → Deployment → Undeployment → deactivation → Uninstallation

Individual CLI operations are described in the following subsections.

Unless otherwise mentioned, no restart is necessary for any operation. All changes are effective immediately.

8.7.1 Installation

The first step in the application life cycle is the installation. For the installation of applications that are not pre-packaged with the Xtended Services Platform, the application file (*.war* or *.bwar*) must be copied to a local temporary directory on the server, for example, */tmp*. Using the CLI, in the *Maintenance/ManagedObjects* level, invoke the install command, providing the full path name of the application file. Note that the name of the file is irrelevant to the Xtended Services Platform. All the descriptive and operational data used by the Xtended Services Platform are inside the file. By default, pre-packaged applications are located in the */usr/local/broadworks/apps* directory.

©2019 CISCO SYSTEMS, INC.

An application is made unique by its name and its version, both of which are defined in the internal deployment descriptor (in the *.war/.bwar*). For web application, the deployment descriptor is the *web.xml* file. On the other hand, the Cisco BroadWorks application deployment descriptor is in the *broadworks.xml* file. The Xtended Services Platform enforces uniqueness of applications and does not allow installation of two applications with the same name/version.

Following is an example of an installation of an application.

```
XSP_CLI/Maintenance/ManagedObjects> install application /tmp/Example-
_1.0.war
Broadworks SW Manager installing Example_1.0.war...
   - Performing basic validation
      . Valid application structure
      . Name       : Example
      . Description: This is an example Web Application
      . Version    : 1.0
...Done

XSP_CLI/Maintenance/ManagedObjects>
```

During the installation process, the application file is validated and, if valid, it is internally copied and installed. Once the application is installed, the file used to install the application is no longer required and can be deleted.

After being installed, the application can either be activated to make it configurable or it can be uninstalled.

NOTE: Applications with an automatic upgrade mode are installed by default when the Xtended Services Platform is installed. Such applications cannot be uninstalled.

8.7.2 Activation

To be able to configure the application, it must be activated. Activation is performed through the CLI with the activate command. The following examples illustrate the activation of applications.

```
XSP_CLI/Maintenance/ManagedObjects> activate ?
    <type>, Choice = {application}
        application:
            <name>, String {1 to 80 characters}
            <version>, String {1 to 80 characters}
            [<contextPath>, String {1 to 80 characters}]

XSP_CLI/Maintenance/ManagedObjects> activate application Example 1.0
/example
Broadworks SW Manager activating...Example version 1.0
...Done

XSP_CLI/Maintenance/ManagedObjects> activate application SIPLocation
23.0_1.1015
BroadWorks SW Manager activating...SIPLocation version 23.0_1.1015
...Done

XSP_CLI/Maintenance/ManagedObjects>
```

When activating an application, the name and version are mandatory. The *contextPath* parameter is only required when activating a web application. It must not be specified when activating a Cisco BroadWorks application.

©2019 CISCO SYSTEMS, INC.

Multiple instances of the same web application can be activated using different context paths. However, a single activation of a Cisco BroadWorks application is allowed.

Once activated, applications can be configured. Configuration is done through the CLI from the application specific context. This context can be found under the application level as illustrated in the following example.

```
XSP_CLI/Applications> ?
    0)  SIPLocation: go to level SIPLocation
    1)  Example : go to level Example_1.0

  h (help),  e (exit),  q (quit),  r (read),  w (write),  t (tree),
  c (config),  cd (cd),  a (alias),  hi (history),  p (pause),  re
(repeat)

XSP_CLI/Applications> 1

XSP_CLI/Applications/Example> ?
Configure the Example_1.0 web application.

Commands:
    0)               get : show webapp-related attributes
    1)               set : modify webapp-related attributes
    2)             clear : clear webapp-related attributes

  h (help),  e (exit),  q (quit),  r (read),  w (write),  t (tree),
  c (config),  cd (cd),  a (alias),  hi (history),  p (pause),  re
(repeat)

XSP_CLI/Applications/Example>
```

As shown in the example, the two activated applications are listed as sub context. Note however, that not all applications support configuration.

8.7.3 Deployment

Applications activated on the Xtended Services Platform can be selectively deployed using the CLI deploy command. Following are examples of application deployments.

```
XSP_CLI/Maintenance/ManagedObjects> deploy ?
    <type>, Choice = {application}
        application:
            <nameOrContextPath>, String {1 to 255 characters}

XSP_CLI/Maintenance/ManagedObjects> deploy application /example
Broadworks SW Manager deploying /example...
...Done

XSP_CLI/Maintenance/ManagedObjects> deploy application SIPLocation
Broadworks SW Manager deploying SIPLocation...
WARNING -> You need to restart the server prior to start this application
due to memory settings change.
...Done

XSP_CLI/Maintenance/ManagedObjects>
```

As shown in the previous example, deploying a web application and a Cisco BroadWorks application require different parameters. A web application is deployed using the context path specified during the activation command. On the other hand, deploying a Cisco BroadWorks application is done using the application name.

©2019 CISCO SYSTEMS, INC.

The following subsections present the deployment differences between web and Cisco BroadWorks application.

8.7.3.1 Web Application

When activating and deploying a web application, the HTTP uniform resource locator (URL) path that leads to it must be specified; this is the “context path”. The Xtended Services Platform can host many web applications and each one must have a unique context path.

Web applications are deployed to the application container (BroadWorks WebContainer application), creating a running instance of the application. A given web application (name, version) can be deployed many times using different context paths, creating as many independently running instances.

Deployed web applications remain deployed until they are undeployed. However, it is not because a web application is deployed that it automatically has a running instance. There is a relationship between web applications and BroadWorks WebContainer application such that the WebContainer application must be deployed and running for the web application instances to be running. When the BroadWorks WebContainer application is stopped, the running instances of the web applications are automatically stopped.

The state of the web applications is preserved through Xtended Services Platform restarts and Xtended Services Platform upgrades.

Note that the context path must always start with a leading “/” and cannot contain any other “/” characters. For example “/xsi-actions_1.0” is a valid context path.

8.7.3.2 Cisco BroadWorks Application

Compared to web applications, Cisco BroadWorks applications are made of one or more components called containers. Each container is implemented as a process on the server. Deploying the Cisco BroadWorks application is not enough to get the application’s functionality; it must be started. Cisco BroadWorks applications are started and stopped individually or all at the same time using either the CLI start or stop command or the startbw or stopbw script. The syntax is similar in both cases. When the optional application name is specified, only the specified application is started/stopped. Following are some examples of application start and stop.

```
XSP_CLI/Maintenance/ManagedObjects> start application SIPLocation
--> Starting application SIPLocation <--

Broadcast message from bworks@mtl64lin12 (Tue Sep 18 15:04:42 2018):

===== BROADWORKS CONTROL --- START INITIATED ON MTL64LIN12 (SIPLocat
ion) =====
    Starting                          [done]


XSP_CLI/Maintenance/ManagedObjects>


bwadmin@mtl64lin12 $ showrun

Currently running BroadWorks Application processes:

  sipLocation process monitor (pid=29261)
  sipLocation (pid=29301)
  tomcat process monitor (pid=3333)
  tomcat (pid=3498)
```


©2019 CISCO SYSTEMS, INC.

```
Currently running BroadWorks Platform processes:

  BroadWorks Configuration Agent process monitor (pid=1610)
  BroadWorks Configuration Agent (pid=1633)
  BroadWorks License Manager process monitor (pid=31971)
  BroadWorks License Manager (pid=31994)
  BroadWorks SNMP Agent process monitor (pid=31534)
  BroadWorks SNMP Agent (pid=31557)
  BroadWorks Software Manager (pid=30685)

bwadmin@mtl64lin12 $ stopbw SIPLocation
BroadWorks control script version 23.0
Stopping SIPLocation...

Broadcast message from bworks@mtl64lin12 (Tue Sep 18 15:06:07 2018):

===== BROADWORKS CONTROL --- STOP INITIATED ON MTL64LIN12 (SIPLocati
on) =====
Stopping sipLocation...
Waiting for core processes to terminate.....
... Done.

Currently running BroadWorks Application processes:

  tomcat process monitor (pid=3333)
  tomcat (pid=3498)

Currently running BroadWorks Platform processes:

  BroadWorks Configuration Agent process monitor (pid=1610)
  BroadWorks Configuration Agent (pid=1633)
  BroadWorks License Manager process monitor (pid=31971)
  BroadWorks License Manager (pid=31994)
  BroadWorks SNMP Agent process monitor (pid=31534)
  BroadWorks SNMP Agent (pid=31557)
  BroadWorks Software Manager (pid=30685)

bwadmin@mtl64lin12 $
```

As soon as an application is deployed, healthmon starts monitoring it and reports any abnormal condition. Following is an example of healthmon output where both UC-Connect and WebContainer applications are deployed but not running.

```
bwadmin@mtl64lin12 $ healthmon -l
Partition monitoring  [.....................]
BroadWorks monitoring [.....................]
Resources monitoring  [.....................]
--------------------------------
System Health Report Page
   BroadWorks Server Name : mtl64lin12
   Date and time          : Wed Sep 19 13:58:34 EDT 2018
   Report severity        : CRITICAL
   Upgrade check severity : CRITICAL
   Server type            : XtendedServicesPlatform
   Server state           : Unlock
--------------------------------

BroadWorks XtendedServicesPlatform processes in trouble:

tomcat not running
tomcat process monitor not running
sipLocation not running
```


©2019 CISCO SYSTEMS, INC.

```
sipLocation process monitor not running


--------------------------------

Recommendations
---------------

The WebContainer application needs to be restarted.
The SIPLocation application needs to be restarted.

--------------------------------
bwadmin@mtl64lin12 $
```

8.7.3.2.1 Memory Management

Each Cisco BroadWorks application has its own memory requirement (minimum and maximum values) defined by Cisco based on the application deployment requirements. When deploying a Cisco BroadWorks application, the Software Manager calculates the current system memory usage to determine whether the system has enough memory available for the application being deployed. In cases where there is not enough memory available, the Software Manager recalculates the allocated memory and may decide to reduce the memory allocated to already deployed applications to give some memory to the application being deployed. In such a case, a warning is displayed to indicate a restart of the applications is required for the applications to start with their readjusted memory configuration. Following is an example of such a warning.

```
XSP_CLI/Maintenance/ManagedObjects> deploy application
DeviceManagementTFTP
Broadworks SW Manager deploying DeviceManagementTFTP...
WARNING -> You need to restart the server prior to start this application
due to memory settings change.
...Done

XSP_CLI/Maintenance/ManagedObjects>
```

In cases where the amount of physical memory is changed on the server, the application's memory configuration is automatically recalculated by the Software Manager at boot time.

8.7.3.2.2 State Management

Each Cisco BroadWorks application implements its own state. The application state is presented as an administrative state and an effective state. Both states can be viewed from the CLI *Maintenance/ManagedObjects* level using the command `get broadworks full`. Only the administrative state can be controlled by an operator; the effective state simply reflects the actual state of the application. The administrative state is controlled through the CLI *Maintenance/ManagedObjects* level using the lock and unlock commands. As the commands suggests, the possible administrative states are:

- Locked
- Unlocked

When an application administrative state is set to one of these states, it means that the application tries to transition to that state (assuming the application is running).

©2019 CISCO SYSTEMS, INC.

On the other hand, the effective state reflects the actual state of the application and can have the following values.

- *Locked*: The application is locked. Depending on the application's implementation, it may mean that it provides limited to no functionality at all.
- *Locking*: The application is in the process of being locked. This is a transitional state.
- *Unlocked*: The application is unlocked; all its functionality is available.
- *Unlocking*: The application is in the process of being unlocked. This is a transitional state.
- *Stopping*: The application is in the process of being stopped. This is a transitional state.
- *Stopped*: The application is stopped; none of its functionality is available.
- *InTrouble*: The application is not functional; one or more (but not all) of its containers are dead. The application must be restarted. The application also enters this state when one or more (but not all) of its containers is restarted by the process monitor.
- *Dead*: The application is not functional; all of its containers are dead. The application must be restarted.

Following is an example of the `get broadworks full` command, where three Cisco BroadWorks applications are deployed but only two are running.

```
XSP_CLI/Maintenance/ManagedObjects> get broadworks full
BroadWorks Managed Objects
==========================

* Server:
  Identity.............: XSP
  Version..............: Rel_23.0_1.1015
  Administrative State..: Unlocked

* Applications:
                  Name      Version  Deployed  Administrative State
Effective State
================================================================================
===========
  DeviceManagementTFTP  23.0_1.1015      true      Unlocked
Stopped
           SIPLocation  23.0_1.1015      true      Unlocked
Unlocked
          WebContainer  23.0_1.1015      true              Unlocked
Unlocked

3 entries found.

* Hosted Applications:
                  Name      Version  Context Path  Deployed
Administrative State  Effective State
================================================================================
=========================
  BusinessCommunicator          2.1         /test     false
Unlocked          Stopped
             CommPilot  23.0_1.1015    /CommPilot      true
Unlocked         Unlocked

2 entries found.
```


©2019 CISCO SYSTEMS, INC.

```
Currently running BroadWorks Application processes:

  sipLocation process monitor (pid=12728)
  sipLocation (pid=12818)
  tomcat process monitor (pid=12676)
  tomcat (pid=12917)

Currently running BroadWorks Platform processes:

  BroadWorks Configuration Agent process monitor (pid=1610)
  BroadWorks Configuration Agent (pid=1633)
  BroadWorks License Manager process monitor (pid=31971)
  BroadWorks License Manager (pid=31994)
  BroadWorks SNMP Agent process monitor (pid=31534)
  BroadWorks SNMP Agent (pid=31557)
  BroadWorks Software Manager (pid=30685)


XSP_CLI/Maintenance/ManagedObjects>
```

The server administrative state, as shown in the top of the above output as *Unlocked*, also has an impact on the application's effective state. For instance, if an operator locks the server, all application effective states transition to *Locked* regardless of the application's administrative state. This is because the *Locked* server's administrative state supersedes the application's administrative state. However, when the server is in *Unlocked* state, an operator can freely change an application administrative state, and the application effective state transitions to the requested state. Following is an example where the server administrative state is set to "Locked" and the application's administrative state is "Unlocked" but the effective state of the running application is "Locked".

```
XSP_CLI/Maintenance/ManagedObjects> qbw
BroadWorks Managed Objects
==========================

* Server:
  Identity.............: XSP
  Version..............: Rel_23.0_1.1015
  Administrative State..: Locked

* Applications:
                  Name      Version  Deployed  Administrative State
Effective State
================================================================================
===========
  DeviceManagementTFTP  23.0_1.1015      true              Unlocked
Stopped
           SIPLocation  23.0_1.1015      true              Unlocked
Locked
          WebContainer  23.0_1.1015      true              Unlocked
Locked

3 entries found.

* Hosted Applications:
                  Name      Version           Context Path  Deployed
Administrative State  Effective State
================================================================================
==================================
  BusinessCommunicator          2.1  /BusinessCommunicator      true
Unlocked           Locked
```


©2019 CISCO SYSTEMS, INC.

```
             CommPilot  23.0_1.1015             /CommPilot      true
Unlocked           Locked

2 entries found.


Currently running BroadWorks Application processes:

  sipLocation process monitor (pid=12728)
  sipLocation (pid=12818)
  tomcat process monitor (pid=12676)
  tomcat (pid=12917)

Currently running BroadWorks Platform processes:

  BroadWorks Configuration Agent process monitor (pid=1610)
  BroadWorks Configuration Agent (pid=1633)
  BroadWorks License Manager process monitor (pid=31971)
  BroadWorks License Manager (pid=31994)
  BroadWorks SNMP Agent process monitor (pid=31534)
  BroadWorks SNMP Agent (pid=31557)
  BroadWorks Software Manager (pid=30685)

XSP_CLI/Maintenance/ManagedObjects>
```

8.7.4 Undeployment

Applications can be undeployed when they are no longer required. Web applications can be undeployed without restriction; however, before being undeployed, Cisco BroadWorks applications must be stopped. Undeploying an application is done with the command undeploy invoked from the CLI level *Maintenance/ManagedObjects*. For web applications, the path must be specified and for Cisco BroadWorks applications, the application name must be used. Following are some examples.

```
XSP_CLI/Maintenance/ManagedObjects> undeploy application
DeviceManagementTFTP
Broadworks SW Manager un-deploying DeviceManagementTFTP...
...Done

XSP_CLI/Maintenance/ManagedObjects> undeploy application
/BusinessCommunicator
Broadworks SW Manager un-deploying /BusinessCommunicator...
...Done

XSP_CLI/Maintenance/ManagedObjects> get broadworks full
BroadWorks Managed Objects
==========================

* Server:
  Identity.............: XSP
  Version..............: Rel_23.0_1.1015
  Administrative State..: Locked

* Applications:
                  Name      Version  Deployed  Administrative State
Effective State
================================================================================
===========
  DeviceManagementTFTP  23.0_1.1015     false              Unlocked
Stopped
```


©2019 CISCO SYSTEMS, INC.

```
           SIPLocation  23.0_1.1015      true              Unlocked
Locked
          WebContainer  23.0_1.1015      true              Unlocked
Locked

3 entries found.

* Hosted Applications:
                  Name      Version           Context Path  Deployed
Administrative State  Effective State
================================================================================
==================================
  BusinessCommunicator          2.1  /BusinessCommunicator     false
Unlocked          Stopped
             CommPilot  23.0_1.1015             /CommPilot      true
Unlocked           Locked

2 entries found.


Currently running BroadWorks Application processes:

  sipLocation process monitor (pid=20220)
  sipLocation (pid=20348)
  tomcat process monitor (pid=20168)
  tomcat (pid=20406)

Currently running BroadWorks Platform processes:

  BroadWorks Configuration Agent process monitor (pid=1610)
  BroadWorks Configuration Agent (pid=1633)
  BroadWorks License Manager process monitor (pid=31971)
  BroadWorks License Manager (pid=31994)
  BroadWorks SNMP Agent process monitor (pid=31534)
  BroadWorks SNMP Agent (pid=31557)
  BroadWorks Software Manager (pid=30685)


XSP_CLI/Maintenance/ManagedObjects>
```

Once a web application is undeployed, no instance of this application's context path is running within the WebContainer. To reuse the context path, the web application must be deactivated.

When a Cisco BroadWorks application is undeployed, it can no longer be started.

Applications can still be configured once undeployed.

> **NOTE**: Undeploying the BroadWorks Web Container Application affects all web applications. The web applications are no longer running.

8.7.5 Deactivation

Undeploy applications can be deactivated using the deactivate CLI command. Deactivated applications can no longer be configured. As such, their associated CLI level under the application context is removed. For web applications, the context path is released and can be reused. Following are some examples of application deactivation. Note that deactivating a Cisco BroadWorks application is done using the application name, whereas deactivating a web application is done using the context path.

©2019 CISCO SYSTEMS, INC.

```
XSP_CLI/Maintenance/ManagedObjects> deactivate application
DeviceManagementTFTP
Broadworks SW Manager deactivating...DeviceManagementTFTP version
23.0_1.1015
...Done

XSP_CLI/Maintenance/ManagedObjects> deactivate application
/BusinessCommunicator
BroadWorks SW Manager deactivating...BusinessCommunicator version 2.1
...Done

XSP_CLI/Maintenance/ManagedObjects>
```

8.7.6 Uninstallation

Installed applications that are no longer required can be uninstalled. This removes all files related to this application from the Xtended Services Platform. Only applications in *Active* state can be uninstalled. Furthermore, only applications having their upgrade mode set to "Manual" can be uninstalled. Those with an Automatic upgrade mode cannot be uninstalled as they are part of the Xtended Services Platform and they are upgraded with the platform. An application is uninstalled using the uninstall command form the CLI, in the *Maintenance/ManagedObjects* level. The uninstall command required an application and a version. Following is an example of an uninstallation of a web application.

```
XSP_CLI/Maintenance/ManagedObjects> uninstall application
BusinessCommunicator 2.1
BroadWorks SW Manager un-installing BusinessCommunicator version 2.1...
...Done

XSP_CLI/Maintenance/ManagedObjects>
```

8.7.7 Upgrade

Applications that are using the Manual upgrade mode must be manually upgraded through the CLI. Applications with an Automatic upgrade mode are automatically upgraded when the Xtended Services Platform is upgraded. Currently all Cisco BroadWorks applications have an Automatic upgrade mode; therefore, these applications cannot be manually upgraded. The following procedure describes the steps required to upgrade a running web application:

1) Copy the new application *.war* file to a local directory (*/tmp/first.war*).
2) Install the new web application using the CLI.
3) [Optional] Test the new web application:
   - Deploy a test instance using the CLI.
   - Test the application.
   - Undeploy the test instance using the CLI.
4) Undeploy the old version of the web application using the CLI.
5) Deploy the new version using the CLI.

©2019 CISCO SYSTEMS, INC.

8.8 HTTP Server Configuration

Typically, a Web Server processes HTTP requests on port 80 and HTTPS requests on port 443 of its public Internet Protocol (IP) interface. The Xtended Services Platform requires the additional capacity to serve HTTP/HTTPS on different ports and to support multiple IP interfaces. For this reason, the Xtended Services Platform allows complete operator control over which IP interfaces and ports are used to accept server requests. The CLI provides comprehensive management functions to configure and set up HTTP servers. All commands are located under the *XSP_CLI/Interfaces/Http/HttpServer* level.

The Xtended Services Platform defines an HTTP server as a point of service for HTTP or HTTPS requests. An HTTP server is uniquely defined by an IP address of a local interface and a port (for example, 192.168.12.12:80). Each HTTP server is defined with a public host name and can be secured with a secure sockets layer (SSL).

During the Xtended Services Platform installation, a default secured HTTP server is created using the first IP address corresponding to the local host name and port 443. If no interface matches the local host name configuration, the first IP address is used.

Requests received by all Web Servers are dispatched to the proper web application according to the web application's context path. There are no restrictions applied to the dispatching of requests based on the HTTP server that received them.

Following is an example of how the HTTP server can be properly defined.

```
XSP_CLI/Interface/Http/HttpServer> get
     Interface Port           Name  Secure  Client Auth Req  Cluster FQDN
=========================================================================
192.168.13.186  80  192.168.13.186   false            false

192.168.13.186 443  192.168.13.186    true            false
```

In the example, *web.broadsoft.com* represents the FQDN used by all HTTP servers (namely, XSP1 and XSP2), and *xsp1.broadsoft.com* represents the local name of the server. Naming can be important for single SSL certificate generation.

The HTTP servers generate self-signed certificates by default and use the default SSL/TLS protocol and cipher provided by Java. These defaults may be configured on a per end-point basis using the following CLI levels.

```
XSP_CLI/Interface/Http/HttpServer/SSLSettings> h
This level is used to configure the Secure Sockets Layer (SSL)/Transport
Layer Security (TLS) parameters.

Commands:
    0)  Certificates : go to level Certificates
    1)       Ciphers : go to level Ciphers
    2)     Protocols : go to level Protocols

  h (help),  e (exit),  q (quit),  r (read),  w (write),  t (tree),
  c (config),  cd (cd),  a (alias),  hi (history),  p (pause),  re
(repeat),
  k (keyboardHelp)

XSP_CLI/Interface/Http/HttpServer/SSLSettings>
```

The protocols and ciphers may also be configured at less specific contexts at the interface and server level. These less specific contexts are referred to as *SSLCommonSettings*.

For more information, see the *Cisco BroadWorks SSL Support Options Guide* [4].

©2019 CISCO SYSTEMS, INC.

8.8.1 Public Host Name

Each HTTP server is configured with a public host name. The HTTP servers issue a redirect to *http://<hostname>:port/...* for each request it receives in which the host name part of the URL does not match its configured host name.

For example, a server configured with a host name of "myhost1.com" would redirect a request from *http://<ip>/some* to *http://myhost1.com/some*.

Clients must be able to resolve the host name to an IP address. The Xtended Services Platform verifies that the host name is locally defined in */etc/host*.

Requests to the following applications will not be redirected:

- BroadworksDMS
- OCIFiles

For all other web application, exceptions to this behavior can be established using aliases. For more information, see section *8.8.6 HTTP Aliases*.

8.8.2 Secure HTTP Server

The HTTP servers can be configured as secured or not secured. A secured HTTP server only serves HTTPS requests.

Mutual TLS/SSL authentication or certificate-based mutual authentication refers to two parties authenticating each other through verifying the provided digital certificate so that both parties are assured of the other's identity. In Cisco BroadWorks terms, it refers to a client authenticating themselves to an Xtended Services Platform and the Xtended Services Platform also authenticating itself to the client through verifying each other's certificate. The Xtended Services Platform supports one-way authentication, authenticating itself to the client similar to any other web server and, optionally, supports mutual TLS/SSL authentication.

One-way or two-way authentication requires secure sockets layer (SSL) certificates signed by a recognized authority. On the server side, the Xtended Services Platform initially generates self-signed certificates and provides CLI commands to manage replacement of these self-signed certificates with properly signed ones.

The use of self-signed certificates allows for immediate use of the HTTP server for HTTPS; however, they should be replaced with signed certificates for production systems. On the client side, there are commands to generate trust anchors.

8.8.3 Create HTTP Server

Additional HTTP servers can be added to the Xtended Services Platform by using the add command. The IP must be an IP that is locally valid. The host name is the name that clients are redirected to and shows up in URLs.

For example:

```
XSP_CLI/Interface/Http/HttpServer> add 192.168.12.12 8080 myhost
false false
```


©2019 CISCO SYSTEMS, INC.

8.8.4 Delete HTTP Server

HTTP servers that are no longer required can be deleted using the delete command. The IP and port of the server to delete must be provided.

For example:

```
XSP_CLI/Interface/Http/HttpServer> delete 192.168.12.12 8080
```

8.8.5 Manage SSL Certificates

8.8.5.1 Server Certificates

When using a secured HTTP server, the Xtended Services Platform automatically generates self-signed certificates using the host specified for that server. To sign the certificate, the certificate signing request must be extracted from the Xtended Services Platform. Using the *sslGenKey*, the certificate signing request file is generated to the */var/broadworks/tmp* directory under the name <host>.csr. This file is required for the official signed procedures.

Once signed, the certificate file must be locally copied to the Xtended Services Platform, for example, in */tmp*. The sslUpdate command can be used to reload the signed certificate (and optionally, the certificate chain file), replacing the old self-signed certificates. If a certificate chain file is no longer required, it can be removed using the sslRemove command.

The sslExport command can also be used to export the certificate file, the SSL key, and the certificate chain file.

The Xtended Services Platform supports single-host, multi-domain (SAN) and wildcard certificates. Single-host certificates are required for device management when devices do not support wildcard certificates. SAN certificates are required when multiple domains are required for a single certificate. The following subsection shows examples of using each.

8.8.5.2 Single Wildcard SSL Certificate

The certificate must be named properly in accordance with the network naming convention. The following example shows the best-practice approach to single certificate naming.

Assume two (or more) Xtended Services Platform HTTP servers are defined as follows.

On XSP1:

```
XSP_CLI/Interface/Http/HttpServer> get
  Interface Port              Name Secure  Client Auth Req        Cluster FQDN
================================================================================
192.168.8.8  443 xsp1.broadsoft.com   true            false   xsp.broadsoft.com
192.168.8.8   80 xsp1.broadsoft.com  false            false   xsp.broadsoft.com
```

On XSP2:

```
XSP_CLI/Interface/Http/HttpServer> get
  Interface Port              Name Secure  Client Auth Req        Cluster FQDN
================================================================================
192.168.8.9  443 xsp2.broadsoft.com   true            false   xsp.broadsoft.com
192.168.8.9   80 xsp2.broadsoft.com  false            false   xsp.broadsoft.com
```


©2019 CISCO SYSTEMS, INC.

Because the names of the HTTP servers follow a proper pattern for the server name, a single certificate with a common name using wildcard characters, such as *xsp*.broadsoft.com* or **.broadsoft.com*, can be issued. One could decide to have two distinct certificates here, for example, *xsp1.broadsoft.com* and *xsp2.broadsoft.com*, although this would not be practical in an environment where both Xtended Services Platforms serve the same software clients in a farm.

Note that software clients accessing the HTTP server using a different alias name, allowed as defined under the *XSP_CLI/Interface/Http/HttpAlias* level, are still handled properly by the same single certificate name. For more information, see section *8.8.6 HTTP Aliases*.

If the names of the HTTP servers do not follow a common pattern (for instance, if the servers are named xsp1.broadsoft.com and web2.bsoft.com), a single wildcard certificate cannot be used for the Xtended Services Platform farm. However, a single multi-domain (SAN) certificate may address that requirement. For details, see section *8.8.5.3 Single Multi-Domain (SAN) Certificate*.

One or more aliases defined for this server do not affect or influence in any way how the certificate is managed. The certificate is always derived from the HTTP server name defined under the CLI and is uniquely bound to an interface and port, whether or not aliases are defined.

This type of certificate will address the need for additional servers without the need for signing again. For example, the existing certificate whose common name is *.broadsoft.com supports xsp3.broadsoft.com.

8.8.5.3 Single Multi-Domain (SAN) Certificate

The certificate must be named properly in accordance with the network naming convention. The following example shows the best-practice approach to SAN certificate naming.

Assume two Xtended Services Platform HTTP servers are defined as follows.

On XSP1:

```
XSP_CLI/Interface/Http/HttpServer> get
  Interface Port               Name Secure  Client Auth Req       Cluster FQDN
================================================================================
192.168.8.8  443 xsp1.broadsoft.com   true            false  xsp.broadsoft.com
192.168.8.8   80 xsp1.broadsoft.com  false            false  xsp.broadsoft.com
```

On XSP2:

```
XSP_CLI/Interface/Http/HttpServer> get
  Interface Port               Name Secure  Client Auth Req       Cluster FQDN
================================================================================
192.168.8.9  443      web2.bsft.com   true            false       web.bsft.com
192.168.8.9   80      web2.bsft.com  false            false       web.bsft.com
```

By creating a single certificate that includes the subjectAlternativeName extension with the value “xsp1.broadsoft.com web2.bsft.com”, a single certificate may be shared between the 2 servers.

It is important to understand that the presence of the subjectAlternativeName in a certificate is mutually exclusive with the certificate common name. That is, when the subjectAlternativeName is present, the SSL handshake does not use the common name. As a result, the common name should be added to the subjectAlternativeName extension when required.

©2019 CISCO SYSTEMS, INC.

Note that software clients accessing the HTTP server using a different alias name, allowed as defined under the *XSP_CLI/Interface/Http/HttpAlias* level, are still handled properly by the same single multi-domain (SAN) certificate name.

If the need for an additional server arise (for example, web3.bsft.com), a SAN certificate will need to be signed again with the updated field subjectAlternativeName ("xsp1.broadsoft.com web2.bsft.com web3.bsft.com").

8.8.5.4 Multiple Single-host SSL Certificates

There are cases where more than one SSL certificate is needed.

For example, when a single Xtended Services Platform is configured to serve two or more customers on the same interface, it is important to remember that only one certificate can be issued for a single HTTP server interface per port for this Xtended Services Platform.

Assume that the server is defined as follows.

```
XSP_CLI/Interface/Http/HttpServer> get
  Interface Port              Name Secure  Client Auth Req  Cluster FQDN
=============================================================================
192.168.8.8  443       companyA.com   true            false
192.168.8.8   80       companyA.com  false            false
```

```
XSP_CLI/Interface/Http/HttpAlias> get

  Interface                      Alias                Cluster FQDN
=================================================================
192.168.8.8               companyB.com
```

The URL *companyB.com* would appear in the HTTP header of a browser because it is defined here as an alias. However, the certificate can only be issued against *companyA.com* with the interface address 192.168.8.8 and port 443. A user accessing the server with *companyB.com* and fetching information about the SSL certificate would realize it was issued against *companyA.com*, which is undesirable. An alternative is to define additional HTTP servers with the internet address 192.168.8.8 but with different ports (for example, 442 and 444), and with some kind of a dummy and generic name applicable to both companies. Although this is not necessarily desirable either.

If additional network cards are added to the platform, additional HTTP servers can be defined, which can then have their own certificates, as illustrated by the following example.

```
XSP_CLI/Interface/Http/HttpServer> get
  Interface Port              Name Secure  Client Auth Req  Cluster FQDN
=============================================================================
192.168.8.8  443       companyA.com   true            false
192.168.8.8   80       companyA.com  false            false
192.168.8.9  443       companyB.com   true            false
192.168.8.9   80       companyB.com  false            false
```

When the need for an increasing number of distinct SSL certificates arises, it might not be feasible to add as many network cards as required. Virtual IPs can be defined for such a purpose.

- On Linux, each physical interface is represented by a file corresponding to */etc/sysconfig/network-scripts/ifcfg-eth<x>* where <x> represents the unique interface number for that card (for example, the first interface card is represented by *ifcfg-eth0*).

  To create a virtual interface for that physical interface, a file, in the format of *ifcfg-eth0:<y>* where <y>represents the alias number, must be created.

©2019 CISCO SYSTEMS, INC.

For example:

```
ifcfg-eth0
# Intel Corporation 82540EM Gigabit Ethernet Controller
DEVICE=eth0
IPADDR=192.168.8.8
NETMASK=255.255.255.0
NETWORK=192.168.8.0
BROADCAST=192.168.8.255
BOOTPROTO=none
ONBOOT=yes
HWADDR=00:11:25:22:28:c8
```

```
ifcfg-eth0:0
# Intel Corporation 82540EM Gigabit Ethernet Controller
DEVICE=eth0:0
IPADDR=192.168.8.10
NETMASK=255.255.255.0
BOOTPROTO=none
ONPARENT=yes
BROADCAST=192.168.8.255
```

Note that the addition of Xtended Services Platform interface addresses must be properly propagated through the entire network (DNS or /etc/hosts, Network Servers configuration, and so on). All settings required on the network and in the Cisco BroadWorks configuration for the primary IP address of the Xtended Services Platform are also required for the added IP addresses.

The virtual interfaces act just as physical interfaces as far as certificates are concerned.

8.8.5.5 Trust anchors

Configuration of client authentication is centralized under a dedicated CLI context named *ClientAuthentication*. The list of trust anchors applies to all interfaces listed in the *HttpServer* level where the field *clientAuthReq* is set to "true". The following shows where this *HttpServer* level is located.

```
PS_CLI/Interface/Http/SSLCommonSettings/ClientAuthentication>
```

Requiring client authentication means that the server must validate the certificate received from the client during the SSL handshake. The server uses a trust store that contains trust anchors to validate these certificates. Therefore, this CLI level provides a sublevel named *trusts* that defines a set of dedicated commands for handling trust anchors. This sublevel is located as follows for the *HttpServer* level.

```
PS_CLI/Interface/Http/SSLCommonSettings/ClientAuthentication/Trusts>
```

These interactions allow the generation of a trust anchor that can be directly distributed to the client software[2] or can be used as an issuer for generating client certificates. Depending on the network requirement, a Cisco BroadWorks customer can require multiple trust anchors. For example, certain device vendors demand the use of their own certificates. The CLI commands require an administrator to specify an alias name to allow for identifying the trust anchors individually.

[2] A trust anchor is itself a certificate.

©2019 CISCO SYSTEMS, INC.

8.8.6 HTTP Aliases

The Xtended Services Platform allows the use of HTTP aliases for each local interface. Aliases are managed through the *XSP_CLI/Interface/Http/HttpAliases* level in the CLI.

Normally, the HTTP servers receiving requests with a host name different from their public host name redirect the client to use the public host name. For more information, see section *8.8.1 Public Host Name*.

Aliases define additional host names values that the HTTP servers accept without redirections. Aliases can be added and deleted through the CLI.

8.8.6.1 Fully Qualified Domain Name Mapping

The HTTP servers support mapping of known fully qualified domain names (FQDNs) to aliases through redirections. An FQDN value can be provisioned on each HTTP alias. HTTP servers receiving a request with the FQDN as a host name redirect the client to use the corresponding alias.

8.8.7 HTTP Bindings

The HTTP bindings define which interfaces (IP address and port) a given web application is available and allow the binding of all applications to specific interfaces. The name of the web application corresponds to the display name defined in its *web.xml* file. An HTTP server web application binding can be created even if the application is not activated or deployed. By default, if no entry is defined under the HTTP server binding, the web application is available on all interfaces. As soon as one entry is added, it is only allowed on that interface.

8.9 Network Server Configuration

There are no changes required to the Network Server configuration to use the Xtended Services Platform.

©2019 CISCO SYSTEMS, INC.

9 Network Configuration Changes

This section provides information on Xtended Services Platform configuration operations required when the IP or host name of the Xtended Services Platform, Application Server, or Network Server is changed.

9.1 Change IP or Host Name of Xtended Services Platform

When changing the IP address or host name of the Xtended Services Platform, the HTTP server and in some cases the HTTP Alias configuration must be updated.

The overall procedure for updating the IP address or host name of a Cisco BroadWorks server is described in the *Cisco BroadWorks Maintenance Guide*. This section details the operations specific to the Xtended Services Platform.

After the IP address or host name has been updated at the operating system (OS) level, the following must be performed:

1) Update the HTTP server configuration.
2) Update the Application Server network access lists for the open client interface (OCI) and ExtAuth.

9.1.1 Update HTTP Server Configuration

Updating the HTTP server configuration is done using the CLI, under the *XSP_CLI/Interface/Http/HttpServer* LEVEL.

For IP address changes:

1) First, for each HTTP server using the old IP, create a new HTTP server on the new IP with the same attributes. Use the add command (for example, add 192.168.12.1 80 host false).
2) When all new HTTP servers are created, simply delete those using the old IP. Use the delete command (for example, delete 192.168.13.1 80).

For host name changes:

1) If the host name was being used by an HTTP server, update the HTTP server name with the new host name. Use the set command (for example, set 192.168.12.1 80 name=newhost).
2) If using the host name for a secured HTTP server, the Xtended Services Platform generates self-signed certificates. These certificates must be signed. For more information, see section *8.8.5 Manage SSL Certificates*.

9.2 Change IP, Host Name or FQDN of Network Server

The BwCommunicationUtility must be reconfigured with the new IP or host name if the integration mode is set to “NS”. If it is set to “NS”, the IP address or host name of the Network Server must be entered in the locationServiceURL using the CLI, under *XSP_CLI/System/CommunicationUtility/DefaultSettings*. For more information, see section *8.4.1 Integration Mode*.

If using a clusterFQDN to reach the Network Server(s), then this step is only required when the FQDN changes.

©2019 CISCO SYSTEMS, INC.

9.3 Change IP or Host Name of Application Server

The BwCommunicationUtility must be reconfigured with the new IP or host name if the integration mode is set to “AS”. If it is set to “AS”, the IP address or host name of the Application Server must be entered using the CLI, under the *XSP_CLI/System/CommunicationUtility/DefaultSettings.* For more information, see section *8.4.1 Integration Mode*.

Change the *asPrimaryAddress* or the *asSecondaryAddress* as required.

©2019 CISCO SYSTEMS, INC.

10 Troubleshooting

This section provides general information about Xtended Services Platform troubleshooting.

10.1 Logs

Most Xtended Services Platform components issue logs. The following subsections provide information on where logging information can be found on the Xtended Services Platform.

10.1.1 HTTP Access Logs

The access logs associated with HTTP requests are stored in the */var/broadworks/logs/tomcat* directory.

The following shows an example of access logs.

```
bwadmin@xsp:/var/broadworks/logs/tomcat> more access_log.2015.06.19-
14.57.04.txt

127.0.0.1 - - [19/Jun/2015:14:57:04 -0400] "GET
/servlet/UpdateServerVersion?address=brian-XSP&version=XSP_Rel_22.0_1.337
HTTP/1.1" 404 1014 0.253 253 "-" "Java/1.7.0_75"

10.9.33.101 - - [19/Jun/2015:15:12:19 -0400] "GET /XspTest/showrequest
HTTP/1.1" 200 2790 0.039 39 "http://10.9.33.22/XspTest/" "Mozilla/5.0 (Windows
NT 6.1; WOW64; rv:38.0) Gecko/20100101 Firefox/38.0"
```

10.1.2 Tomcat Logs

The Tomcat servlet container used by the Xtended Services Platform issues logs in the */var/broadworks/logs/tomcat* directory. This log contains the container internal logs and possibly the exceptions it encounters while executing the various web applications.

10.1.3 Web Application Logs

Web application logging is under the control of the individual web applications. For information on logging, refer to the related documentation for each web application.

For example, the XtendedServicesInterface, version 1.0, issues logs in the */var/broadworks/logs/xsp* directory, using the *XsiLog*.txt* file name.

10.1.4 Cisco BroadWorks Application Logs

Cisco BroadWorks application logging is under the control of the individual applications. For information on logging, refer to the related documentation for each application.

10.1.5 Configuration Agent Logs

The Configuration Agent responsible for hosting the server's configuration issues logs to the */var/broadworks/logs/config* directory.

10.1.6 Other Cisco BroadWorks Logs

In general, all logs issued by Cisco BroadWorks components are located in the */var/broadworks/logs* directory.

©2019 CISCO SYSTEMS, INC.

10.2 Common Problems

This section describes common problems and potential causes.

10.2.1 Authentication Failure for Valid User

The following subsections describe some of the causes for an authentication failure for a valid user.

10.2.1.1 User Login without Domain Name when Network Server Mode is Used

In "NS" mode, the *BWCommunicationUtility* performs LocationAPI lookups on the Network Server to resolve the Application Server cluster that owns it. The full user name must be user: user@domain because the Xtended Services Platform is not aware of the default domain of the Application Server cluster.

The following log can usually be observed.

```
BwCommunicationMgr: Executing NS user lookup
BwCommunicationMgr: Servers={null,null}
BwCommunicationMgr: No usable OCI-P authenticator for <user>
```

10.2.1.2 Application Server ExtAuth or OCI AccessControlList Not Provisioned

The following log can usually be observed.

```
OCI-P authentication for <user>@<domain>
…
Authentication rejected for <user>@<domain>
```

10.2.2 OCI-C Connectivity Problems

The following subsections describe common errors that occur when using OCI-C.

10.2.2.1 Application Server Does Not Support OCI-C (For Release 14.sp5 or Later)

The following log can usually be observed.

```
Failed to register OCI-C protocol
```

10.2.2.2 Xtended Services Platform is Configured for "AS" Mode Integration

The following log can usually be observed.

```
Operation aborted, OCI-C is not supported in AS integration mode
```

10.2.2.3 The Web Application Does Not Provide an OCI-C Application ID

The following log can usually be observed.

```
Operation aborted, OCI-C application is not configured
```


©2019 CISCO SYSTEMS, INC.

10.2.2.4 OCI-C Application ID used by Web Application is Not Properly Provisioned on Application Server

The following log can usually be observed.

```
OCI-C Connection Registration rejected app <name>
AS responded with <reason>
```


©2019 CISCO SYSTEMS, INC.

11 Xtended Services Platform Web Applications Requirements

The Xtended Services Platform allows installation and deployment of standard web applications packaged as web application archive (war) files. This section defines the expected structure of such web applications and provides guidelines for their implementation.

These guidelines take into account the fact that some of the web applications produced by Cisco BroadWorks can be deployed on third-party platforms.

This section is not a tutorial on how to build web applications. Rather, it focuses on the compliancy with the Xtended Services Platform.

11.1 Application Packaging Requirements

The applications must be packaged as standard war files. The applications must be designed with the mindset that the war is never to be exploded or tampered with to be installed, deployed, and used.

11.1.1 War File Name

The following war file naming convention is used by Cisco/BroadSoft for all applications it releases:

<appname>_<version>.war

… where:

<version> = <major>.<minor>

For example: bwXsi_1.0.war

> **NOTE**: This naming convention has no functional bearing on the web application and is intended only as a convenience for users.

The Xtended Services Platform does not rely on any elements from the naming pattern for management of the web application. It extracts name and version information from the embedded *web.xml*, as described in the following subsections. The Xtended Services Platform therefore, does not expect or enforce this naming convention.

Note that manually changing the *.war* file name can be required on third-party platforms to accommodate deployment and establishment of the context path for the application (notably for Tomcat in some configurations). This is to be expected.

11.1.2 War File Structure and Contents

The *.war* file must have the following directory structure[3]:

- *.html, *.jsp, and so on – The HTML and JSP pages, along with other files that must be visible to the client browser (such as JavaScript, style sheet files, and images).

[3] The standard for building the web application can be obtained from http://tomcat.apache.org/tomcat-6.0-doc/appdev/deployment.html.

©2019 CISCO SYSTEMS, INC.

- /WEB-INF/web.xml – The web application deployment descriptor for the application. Details about this file can be obtained from http://tomcat.apache.org/. All servlet mappings and listener configuration (as defined earlier in this document) must be done by the web application developer. This file only contains static configuration for the application.
- /WEB-INF/classes/ – This directory contains any Java class files (and associated resources) required by the interface, including both servlet and non-servlet classes, which are not combined into JAR files. If some classes are to be organized into Java packages, one must reflect this in the directory hierarchy under */WEB-INF/classes/*. For example, a Java class named *com.mycompany.mypackage.MyServlet* would need to be stored in a file named */WEB-INF/classes/com/mycompany/mypackage/MyServlet.class*.
- /WEB-INF/lib/ – This directory contains JAR files that contain Java class files (and associated resources) required by the application, such as third-party class libraries. The BwCommunicationUtility should not be included in the *.war* file.

Note that the *.war* file can be built using zip or jar utilities. One simply needs to use the *.war* file extension rather than *.zip* or *.jar*.

11.2 Web Application Descriptor File (web.xml)

To deploy web applications on the Xtended Services Platform, the following *web.xml* items are used by the Xtended Services Platform for web application management processes. The Xtended Services Platform validates the *web.xml* of all web applications and enforces the presence of mandatory fields.

Item	Status	XML Tags
Display name	required	<display-name>bwXsi</display-name>
Description	optional	<description> BroadWorks Xtended Services Interface </description>
Webapp version	required	<env-entry> <description>webapp version</description> <env-entry-name>webAppVersion</env-entry-name> <env-entry-type>java.lang.String</env-entry-type> <env-entry-value>1.0</env-entry-value> </env-entry>
Webapp configuration file	optional	<env-entry> <description> Name of the configuration file for this webapp. </description> <env-entry-name>webAppConfigFile</env-entry-name> <env-entry-type>java.lang.String</env-entry-type> <env-entry-value>bwXsiActionsConfig.properties </env-entry-value> </env-entry>

The following provides a sample *web.xml* with the expected items.

```
<web-app>
  <display-name>bwXsi</display-name>
  <description>BroadWorks Xtended Services Interface</description>
```


©2019 CISCO SYSTEMS, INC.

```
  <env-entry>
    <env-entry-name>webAppVersion</env-entry-name>
    <env-entry-type>java.lang.String</env-entry-type>
    <env-entry-value>1.0</env-entry-value>
  </env-entry>

  <env-entry>
    <env-entry-name>webAppConigFile</env-entry-name>
    <env-entry-type>java.lang.String</env-entry-type>
    <env-entry-value>bwXsiActionsConfig.properties</env-entry-value>
  </env-entry>
  …

</web-app>
```

11.3 Web Application Naming

To manage the web application, the display-name must be defined in the *web.xml*. This should be a short token and must not contain spaces.

The name is used together with the version to uniquely identify a web application on the Xtended Services Platform. It is presented to the operator by the Xtended Services Platform's CLI to provide web application management functions.

Note that this name has no bearing on the web application's context path, on the Xtended Services Platform.

11.4 Web Application Versioning

To help manage the web application, the web application developer is required to include the web application version information in the web application descriptor file (web.xml). The version is specified as an environment entry <*env-entry*>. For tag details, see section *11.2 Web Application Descriptor File (web.xml)*.

The version information together with the display-name uniquely identifies a web application on the Xtended Services Platform. It is presented to the operator by the Xtended Services Platform CLI to provide web application management functions. This is the version used by the CLI and not the version embedded in the war file name.

The version should not contain spaces.

11.5 Web Application External Configuration

The *web.xml* and *context.xml* presented in the *.war* file should be used to provide configuration data to the web application implementation instead of the hard-coding values in the Java code. This allows easier customization of web applications by application developers. If the *web.xml* and *context.xml* are too limited for some configuration data, data files can be included under *WEB-INF/lib* and loaded by the web application. However, this is to be considered as static "build time" configuration rather than end-user web application configuration because it relies on files packaged inside the war file. Once packaged, war files should not be required to be exploded and rebuilt to be configured.

Web applications that require external deployment time configuration input must rely on an external properties file to do this. The external properties file is to be made accessible to one of the application container's classloaders. For Tomcat, the file should be made available to the shared classloader by placing it under */usr/local/tomcat/tomcat_base/lib*.

The web application must publish the name for this file (no path is required since it is loaded through the classpath). In addition, it must define the file name as an environment entry in the *web.xml*, under the *webAppConfigFile* name.

©2019 CISCO SYSTEMS, INC.

On the Xtended Services Platform, this file can be managed through the CLI provided the web application includes a Cisco BroadWorks configuration descriptor “bwCongif.xml” in the META-INF directory included in its war file. If no configuration descriptor is provided or if you are running the web application on a third-party container, this file must be managed manually. The Xtended Services Platform renders a CLI level for the web application under the *XSP_CLI/Maintenance/WebApp/Config* level.

11.5.1 BwCommunicationUtility Overrides

The *BwCommunicationUtility* allows “per-webapp” configuration of all its settings. The default values for all web applications can be configured through the *BwCommunicationUtility’s* own configuration file. If a web application wants to use its own values for some of the settings (for example, overload controls), it can either rely on internal static values that it uses to configure the library through its application programming interface (API) or it must read its values from its external configuration file and use these to configure the library through its API.

11.6 Web Application Context Path

The *context.xml* embedded in the war file contains a *path* attribute that can theoretically define the URL path to the web application; however, it is not recognized on most containers.

On the Xtended Services Platform, the web application context paths are specified through the CLI at deployment time.

©2019 CISCO SYSTEMS, INC.

References

[1] Cisco Systems, Inc. 2019. *Cisco BroadWorks Software Management Guide, Release 23.0*. Available from Cisco at xchange.broadsoft.com.

[2] Cisco Systems, Inc. 2019. *Cisco BroadWorks Performance Measurements Interface Specification, Release 23.0*. Available from Cisco at xchange.broadsoft.com.

[3] Cisco Systems, Inc. 2019. *Cisco BroadWorks Xtended Services Platform Fault and Alarm Interface Specification, Release 23.0*. Available from Cisco at xchange.broadsoft.com.

[4] Cisco Systems, Inc. 2019. *Cisco BroadWorks SSL Support Options Guide*, *Release 23.0*. Available from Cisco at xchange.broadsoft.com.

[5] Cisco Systems, Inc. 2016. *Xtended Services Platform Load Balancing Feature Description, Release 22.0.* Available from Cisco at xchange.broadsoft.com

[6] Cisco Systems, Inc. 2019. *Cisco BroadWorks Configuration System Interface Specification*, Release 23.0. Available from Cisco at xchange.broadsoft.com.

©2019 CISCO SYSTEMS, INC.