oplon®

OPLON®
Secure Access
Rel. 10.7.0
Reference Guide

OPLON®
Release 10.7.0
Document Rev. 3.0.12
Date 01 March 2023

Property OPLON NETWORKS SRL
www.oplon.net
info@oplon.net

Table of Contents

Product descriptions and versions distributed

Oplon®Secure Access is a set of tools designed to increase the availability of application services ensuring the maximum security obtainable with state-of-the-art technology.
The design choices are based on careful preliminary studies and significant experiences in the field of concurrent processing which began in 1986 which allowed the implementation of a reference model on which the entire system rests. Oplon®Secure Access derives from a long experience gained in numerous mission-critical projects which have allowed the solution to acquire the characteristics of simplicity and reliability typical of this sector. Oplon®Secure Access includes several products and features released in commercial distributions:

1

Configuration & Setup

Notes before installing

The OPLON® products are intended for mission-critical environments therefore only individuals who have successfully taken the course and have passed the examination are therefore authorized to certify the installation and maintenance of the products. All certified individuals are provided with a certificate of proof of participation and successful completion of the courses by OPLON NETWORKS.

Before installing the server component, it is recommended that theOPLON® Management Console be installed to facilitate the configuration.

For the installation of the OPLON® Management Console component refer to the manual: OPLON_ManagementConsole_Installation.pdf available in the storage media supplied with the product or through download from the customer reserved area.

Installation of the OPLON® products requires the necessary licenses. These may be obtained from OPLON NETWORKS through an e-mail addressed to info@oplon.net , or by contacting an authorized distributor.

More information can be found on the web site
www.oplon.net.

Preparing to install: directory tree

The file of the product is a single file in a compressed format with the name composed as follows:

LBLLoadBalancer_aai_999_999_999.zip

Using the underscores as delimeters, the file name represents the following: the first part is the product name, the second part is the distribution contained (eg. Platform, Standard, Enterprise, datawarehouse etc.), and the remaining parts respectively are the release, the dot release and the build.

Once the file has been decompressed the directory structure and the content is similar to the following example.
The location of the directory: "LBLLBLLoadBalancer _aai_999_999_999" identifies the home directory which will be referred to as (LBL_HOME) going forward in the documentation. The environment variable LBL_HOME, which is necessary for the correct operation of the product indicates the directory above.

```
├──LBLLoadBalancer_aai_999_999_999    (LBL_HOME)
├──INTERCEPTORS               (dynamic class for capture and processing of information)
│    ├──dwhInterceptors       (dynamic class for treatment of the database for storage DWH)
│    └──REWRITECLASSES        (dynamic class for treatment and rewriteing of data)
├──legacyBin                  (utilities customized by Operating System)
│    ├──AIX
│    ├──DatabasesScript       (Scripts for networked databases)
│    ├──HP-UX
│    ├──Linux
│    ├──SunOS
│    └──Windows
├──LIB                (Directory of configurations common to all services)
│    ├──CONF           (Configuration file of the MONITOR)
│    ├──confMonitor    (Directory with launch profiles for the processes of the MONITOR)
│    ├──confSchemes    (Directory with the scheme of XSD configurations)
│    ├──extLib         (Directory for external libraries, mail, dbclient etc. )
│    ├──LOGS           (logs of THE MONITOR)
│    ├──notificationDir       (Directory of OutOfOrder notification services or geolocation file)
│    ├──PLUGIN                (default plugin directory)
│    │   └──LBLPLUGIN
│    ├──WEBROO                (root directory for LBL®Monitor WebConsole services)
│    │   ├──WEBAPPS           (home directory of the web Monitor)
│    │   ├──WEBAPPSCON        (configuration file wfor eb applications of the Monitor)
│    │   └──WEBSECURITY       (default digital certificates for management of HTTPS Web servers)
│    │       └──CERTIFICATE
│    ├──WEBROOT_REMOTEBATCH          (home directory for Remote Batch HTTP server)
│    ├──WEBROOT_STATISTICBROKERWEB   (home directory for Statist Web Cache HTTP server)
│    ├──WEBROOT_SURFACECLUSTERDE   (home directory for Surface Cluster Decision Engine HTTP server)
│    ├──WEBROOT_SURFACECLUSTERWF     (home directory for Surface Cluster Work Flow HTTP server)
│    └──WEBROOT_SYSCOMMAND           (home directory for SysCommand  HTTP server)
├──procsProfiles                    (home directory of processes that are managed by the MONITOR)
│    ├──A00_LBLGoSysCommand                  (home directory for Sys Command)
│    ├──A01_LBLIPGeolocalizationDownloader   (home for Geo Localization downloader)
│    ├──A03_LBLGoIPNetworkCardsRedundancy    (home for IP Network Card Redundancy)
│    ├──A03_LBLGoSurfaceClusterDE            (home for Surface Cluster Decision Engine)
│    ├──A03_LBLGoSurfaceClusterDESplitBrainAssassin (home for Surface Cluster Split Brain Assassin)
│    ├──A03_LBLGoSurfaceClusterWF            (home for Surface Cluster Work Flow)
│    ├──A03_LBLGoSurfaceClusterWFRemoteBatch (home for Surface Cluster Remote Batch)
│    ├──A05_LBLGoDNSManager                  (home for DNS Manager)
│    ├──A05_LBLGoStatisticsWebCache          (home for Statistics Broker Web Cache
│    ├──A10_LBLGo                            (home for LoadBalancer)
│    ├──Z99_LBLPluginHTTPCheckWithCmdStart   (home for proactive HTTP Health Check Plugins)
 |    └──Z99_LBLPluginNetworkCheckWithCmdStart      (home for proactive TCP Health Check plugins)
├──RESOURCES
└──SECURITY
    └──CERTIFICATE
```

At the first start other directories will be automatically initialized. For a complete overview of the directories see Oplon®Secure Access Reference Guide

NOTE The contents of the procProfiles file varies depending on the distribution

Environment Variables: file lblsetenv

To execute OPLON®Application Delivery Controller 2 environment variables are necessary:

LBL_HOME := the environment variable that is the location of the OPLON®Application Delivery Controller home directory.

LBL_JAVA_HOME := the environment variable that is the location of the Java home directory.

Before executing any command every shellscript checks for the existence of the environment variables.
In addition the scripts check in LBL_HOME for the existence of:
(LBL_HOME) /lib/LBLLoadBalancer .jar files, while they check in LBL_JAVA_HOME, for the existence of the file (LBL_JAVA_HOME) /java (Unix/Linux) and (LBL_JAVA_HOME) /java.exe (MS Windows).

In order to easily perform the first start, and as soon as the product file is unzipped there are 2 files for Unix/Linux and MS Wndows operating systemsoperating system, lblsetenv.bat and lblsetenv.sh, respectively.

Edit the file of the host operating system, remove the comment from the last 2 rows and insert the absolute locations of the home directories for OPLON®Traffic Mometizer and JAVA.

Ex. Unix/Linux FROM:

```
#!/bin/sh
#                        OPLON(r)ADC
#
#                    This is a commercial software
#You shall not disclose such Confidential Information and shall use
#   it only in accordance with the terms of the license agreement
#
#                         www.oplon.net
#
#                     mailto:info@oplon.net
#
#  OPLON(r)ADC is built on TCOProject(tm) SoftwareLibrary
#LBL and TCOProject are trademarks of F.Pieretti. All rights reserved.
#LBL_HOME=/TCOProject/bin/LBLLoadBalancer; export LBL_HOME
#LBL_JAVA_HOME=/TCOProject/bin/Java/jdk1.6.0_22; export LBL_JAVA_HOME


# os optimizations
LBL_OS=`uname -s`
LBL_NET_DEV="eth0 eth1 eth2 eth3"
LBL_MULTICAST_DEV="eth0 eth1 eth2 eth3"
. $LBL_HOME/lbloptenv.sh
```

TO:

```
#!/bin/sh
#                        OPLON(r)ADC
#
#                    This is a commercial software
#You shall not disclose such Confidential Information and shall use
#   it only in accordance with the terms of the license agreement
#
#                         www.oplon.net
#
#                     mailto:info@oplon.net
#
#  OPLON(r)ADC is built on TCOProject(tm) SoftwareLibrary
#LBL and TCOProject are trademarks of F.Pieretti. All rights reserved.
LBL_HOME=/TCOProject/bin/LBLLoadBalancer_datawarehouse_009_000_000; export
LBL_HOME
LBL_JAVA_HOME=/TCOProject/bin/java; export LBL_JAVA_HOME


# os optimizations
LBL_OS=`uname -s`
LBL_NET_DEV="eth0 eth1 eth2 eth3"
LBL_MULTICAST_DEV="eth0 eth1 eth2 eth3"
. $LBL_HOME/lbloptenv.sh
```

NOTE : The portion of "os optimization" sets, in systems Unix-Linux , the parameters required for the use of OPLON®Application Delivery Controller.
The variable LBL_NET_DEV lists the network interfaces used by LBL while LBL_MULTICAST_DEV lists the network interfaces on which multicast (only necessary in the heart-beat interfaces of Standard and Enterprise versions)

Ex. MS Windows FROM:

```
@ECHO OFF
:                        OPLON(r)ADC
:
:                    This is a commercial software
:You shall not disclose such Confidential Information and shall use
:   it only in accordance with the terms of the license agreement
:
:                         www.oplon.net
:
:                     mailto:info@oplon.net
:
:  OPLON(r)ADC is built on TCOProject(tm) SoftwareLibrary
:LBL and TCOProject are trademarks of F.Pieretti. All rights reserved.
REM set LBL_HOME=C:\TCOProject\bin\LBLLoadBalancer
REM set LBL_JAVA_HOME=C:\TCOProject\bin\Java\jdk1.6.0_22
```

TO:

```
@ECHO OFF
:                          OPLON(r)ADC
:
:                     This is a commercial software
:You shall not disclose such Confidential Information and shall use
:   it only in accordance with the terms of the license agreement
:
:                           www.oplon.net
:
:                       mailto:info@oplon.net
:
:  OPLON(r)ADC is built on TCOProject(tm) SoftwareLibrary
:LBL and TCOProject are trademarks of F.Pieretti. All rights reserved.
set LBL_HOME=C:\TCOProject\bin\LBLLoadBalancer_datawarehouse_009_000_000
set LBL_JAVA_HOME=C:\TCOProject\bin\Java\jdk1.7.0_21
```

To set the environment variables in the current working session in Unix/Linux run:

```
# . ./lblsetenv.sh
# echo $LBL_HOME
/TCOProject/bin/LBLTraining/LBLLoadBalancer_datawarehouse_009_000_000
# echo $LBL_JAVA_HOME
/TCOProject/bin/java
```

For MS Windows, run:

```
C:\> lblsetenv
C:\> echo %LBL_HOME%
C:\TCOProject\bin\LBLtraining\LBLLoadBalancer_datawarehouse_009_000_000
C:\> echo %LBL_JAVA_HOME%
C:\TCOProject\bin\Java\jdk1.7.0_21
```

ATTENTION : This setting of variables is local to the working session.
To make the settings permanent take appropriate action on the operating system files or configurations or on the automatic launch scripts.

Internet protocol: IPv4 , IPv6

Oplon®Secure Access has been made from its inception to use either IPv4 and IPv6.

The use of IPv4 addresses and IPv6 is very simple and is based on W3C recommendation.

IPv4 addresses should be used with the classic decimal notation ex.:

192.168.43.100

IPv6 addresses should be used with the hexadecimal notation separated by : (colon) and between square brackets e.g. :

[fdd4:3c3f:aaaa::99]

All valid representation definitions of IPv6 are covered.

Setting Up java E-Email library

Upon detection of anomalies LBL®Monitor has the ability to send an e-mail notification. For this reason it is necessary to download the API at the following address:

https://java.net/projects/javamail/pages/Home/

And move the library mail.jar in :

cp (TMP_Unzip)/mail.jar (LBL_HOME)/lib/extLib

On startup LBL®Application Delivery Controller verifies the existence of the library and if it is not detected the execution terminates with an error message.

Setting up JavaDB libraries

Oplon® Application Delivery Controller stores statistics in a Relational Database. The default installation uses JavaDB distributed directly in the JDK. If not otherwise specified Oplon® Application Delivery Controller uses JavaDB in embedded mode. In any case Oplon® Application Delivery Controller already during the installation phase requires the following libraries in order to be executed:

- derby.jar
- derbyclient.jar
- derbytools.jar

These libraries are distributed directly inside of the JDK, and after installation can be found in:

Unix/Linux:
(LBL_JAVA_HOME)/db/lib

Windows:
%LBL_JAVA_HOME%\db\lib

To make these libraries visible to Oplon® Application Delivery Controller and simply copy them into (LBL_HOME) /lib/extLib

Unix/Linux:
cp (LBL_JAVA_HOME)/db/lib/derby.jar (LBL_HOME)/lib/extLib
cp (LBL_JAVA_HOME)/db/lib/derbyclient.jar (LBL_HOME)/lib/extLib
cp (LBL_JAVA_HOME)/db/lib/derbytools.jar (LBL_HOME)/lib/extLib

Windows:
copy %LBL_JAVA_HOME%\db\lib\derby.jar %LBL_HOME%\lib\extLib

```
copy %LBL_JAVA_HOME%\db\lib\derbyclient.jar  %LBL_HOME%\lib\extLib
copy %LBL_JAVA_HOME%\db\lib\derbytools.jar  %LBL_HOME%\lib\extLib
```

On startup Oplon® Application Delivery Controller checks for the existence of these libraries. If they are not detected startup terminates with an error message.

ATTENTION

In the 64 bit windows JDK distribution JavaDB might not be present. As JavaDB is fully Java, it is sufficient to take the required libraries from 32 bit JDK distributions.

Start parameters

It is possible to run the start of OPLON®Application Delivery Controller using the parameter "-u" (URL) to indicate the location of the parameter files. This value is set automatically in the presence of the parameter confDir in the <process> section. It is possible to copy only some of the required files to a startup load position while leaving the others to be loaded from the startup common area or within the library.

The following are some examples of start (the result of the setting of the confDir parameter in the <process> section:

```
java loadbalancer.starter.LBLServerStarterApp -u “http://localhost:8080/myParams”
or
java loadbalancer.starter.LBLServerStarterApp -u “file:/C:/myParams”
or
java loadbalancer.starter.LBLServerStarterApp -u “C:\myParams”
or
java loadbalancer.starter.LBLServerStarterApp -u “/myParams”
```

This feature is very important because different configuration releases can be created in different places providing the ability to try new configurations and eventually return to a previous one if need be.

Three parameters have been added at the start of the processes associated with the management service e.g. :

-m identifies the unique service e.g.: -m https://localhost:5900/1_1313t197_A10_LBLGo
-n is the name of the process e.g. : -n A10_LBLGo
-pm is the URL of the OPLON®Monitor process that launches the process e.g. : pm https://localhost:54443

```
<process enable="true"
        description="LBL(r) LoadBalancer Starndard HA Edition"
        start="automatic"
        numberTryStartOnFailure="-1"
```

```
                waitBeforeKill="10000"
                runLevel="2">
            <start osName="Windows">
                <env>CLASSPATH=lib;lib\LBLLoadBalancer.jar</env>
                <workingDir></workingDir>
                <exec>java -Xrs -server -XX:-UseGCOverheadLimit -Xss256k
                        #LBL_GLOBAL_START_HEAP_LOADBALANCER#

#LBL_GLOBAL_GARBAGE_COLLECTOR_LOADBALANCER#
                        %LBL_EXEC_DEFINES%
                        -DLBL_INTERACTIVE_CMD=true
                        loadbalancer.starter.LBLServerStarterApp</exec>
    </start>
```

Release check and updates

At the start of its processes Oplon®Secure Access verifies the release at www.tcoproject.com. The data sent to the site does not contain sensitive data but only reports:
Oplon®Secure Access
rel=99.99.99 ; license:127123163 ***** ; IP= 99999999; RL=9

- rel: the release and the version of the product
- license: the distinctive part of the license in use
- IP: process for license check
- RL:the LBL Run Level

The release check can be deactivated through the parameter " -ncu" during startup of the processes. This can done using the launch profile as shown below. If this functionality is disabled or the associated message cannot reach www.tcoproject.com, TCOGROUP SRL will not be able to provide proactive patching or urgent security related notifications.

```
Ex.:
<process enable="true"
        description="LBL(r) LoadBalancer Platform Edition"
        start="automatic"
        numberTryStartOnFailure="-1"
        waitBeforeKill="10000"
        runLevel="1">
<start osName="Windows">
        <env>CLASSPATH=lib;lib\LBLLoadBalancer.jar</env>
        <workingDir></workingDir>
        <exec>java -Xrs -server -XX:-UseGCOverheadLimit -Xss256k
                #LBL_GLOBAL_START_HEAP_LOADBALANCER#
                #LBL_GLOBAL_GARBAGE_COLLECTOR_LOADBALANCER#
                %LBL_EXEC_DEFINES%
                -DLBL_INTERACTIVE_CMD=true
                loadbalancer.starter.LBLServerStarterApp -ncu</exec>
</start>
```

To disable the verification of the release at the start of LBL Monitor set the parameter -ncu (NoCheckUpdate) at the initial start (batch or service).

Ex.: go.bat :

```
PATH="%LBL_JAVA_HOME%\bin";%PATH%
cd /d "%LBL_HOME%"

set WHAT=loadbalancer.starter.LBLServerStarterApp
set CLASSPATH=lib;lib\LBLLoadBalancer.jar;lib\extLib\mail.jar
java -server -XX:-UseGCOverheadLimit -Xms256m -Xmx256m -DLBL_RUNLEVEL=0 -
DLBL_MONITOR=true -DLBL_INTERACTIVE_CMD=true %WHAT% -ncu
```

OPLON®Monitor start definitions

OPLON®Application Delivery Controller is equipped with a monitoring system for the balancing processes and/or connected to the resources checks. The "Monitor" is therefore the first process to start and then is responsible for launching the other program(s) with the specific service(s).

The parameters for launching the "Monitor" program are:

java \
-server \
-DLBL_RUNLEVEL=0 \
-DLBL_MONITOR=true \
-DLBL_INTERACTIVE_CMD=true \
loadbalancer.starter.LBLServerStarterApp

In debug mode they are:

java \
-server \
-DLBL_RUNLEVEL=0 \
-DLBL_MONITOR=true \
-DLBL_INTERACTIVE_CMD=true \
-DDEBUG=debug \
-DTCO_DEBUG_PROCESS=true \
loadbalancer.starter.LBLServerStarterApp

The parameter -DLBL_INTERACTIVE_CMD=true tells the Monitor to start with an interactive console.

The script/batch to launch the initial (Monitor) program is defined in (LBL_HOME) and can be run in interactive mode with command line:

Unix/Linux:
go.sh

Windows:
go.bat

Or in batch mode:

Unix/Linux:
go.sh false

Windows:
go.bat false

This feature is exploited during the start of OPLON®Application Delivery Controller in Daemon mode (Unix/Linux) and Services (MSWindows) at the start of the Operating System.

OPLON®Secure Access Definitions optional start

It is possible to create several definitions at the start of services *OPLON®* . These definitions are used by the program to change some behaviors such as the log level of the operations. An example can be found in the initial comments in (LBL_HOME) /lib/confMonitor/LBLGo .xml. In chapter ***confmonitor*** individual sections that describe the start of a process are explained.

```
<start osName="Windows">
        <env>CLASSPATH=lib;lib\LBLLoadBalancer.jar</env>
        <workingDir></workingDir>
        <exec>java java -Xss64k -Xrs -server
                -DLBL_INTERACTIVE_CMD=true
              loadbalancer.starter.LBLServerStarterApp</exec>
        <logDirFiles>lib\logs</logDirFiles>
</start>
<start osName="Linux">
        <env>CLASSPATH=lib;lib/LBLLoadBalancer.jar</env>
        <workingDir></workingDir>
        <exec>java -Xss64k -server
                -DLBL_INTERACTIVE_CMD=true
               loadbalancer.starter.LBLServerStarterApp</exec>
        <logDirFiles>lib/logs</logDirFiles>
</start>
<start osName="SunOS">
        <env>CLASSPATH=lib;lib/LBLLoadBalancer.jar</env>
        <workingDir></workingDir>
        <exec>java -Xss64k -server
                -DLBL_INTERACTIVE_CMD=true
              loadbalancer.starter.LBLServerStarterApp</exec>
        <logDirFiles>lib/logs</logDirFiles>
</start>
```

Table of definitions for debug/trace, enabling/disabling basic functionality:

Definition	Description
DEBUG	debug functionality activation-deactivation
LBL_CHECK_REL_JVM	Verifica minima rel JVM supportata. default= true
LBL_DELAY_SWITCHOFF	delay switch off in seconds during OPLON®ADC shotdown. For OPLON®Standard Edition this value could not be less to 40”
LBL_DEBUG_PORT_REWRITING	port rewriting conditions debug
LBL_DEBUG_FLOW	flow debugging
LBL_DEBUG_HEADER	trace HTTP loadbalanced HEADER
LBL_DEBUG_BODY	message body debugging
LBL_DEBUG_HTTPV	show http version
LBL_DEBUG_ENDPOINT	endpoint traking
LBL_DEBUG_SESSION	session traking
LBL_DEBUG_ACTIVESESSIONS	Log delle sessioni attive
LBL_DEBUG_MONITOR	monitor
LBL_DEBUG_MONITOR_HEADER	for http header monitor debugging
LBL_DEBUG_REWRITING	enables the rewriting trace during condition tests and variable loading
LBL_DEBUG_ROW_REWRITING	enable the trace of the rewriting of the stream fragments before and after the changes ATTENTION: this flag if enabled executes a very voluminous log, enable in the debug phase the rules
LBL_DEBUG_ENDPOINT_STATISTICS	system property for debug endpoint staistics
LBL_DEBUG_STATISTIC_CACHE	system property for first and second level of statistics
LBL_DEBUG_HTTP_BAD_REQUEST	HTTP HEADER trace in case of 400 bad header
javaagent:lib\LBLLoadBalancer.jar	To be used with activation of the definitions [A]
LBL_DEBUG_STATISTIC_CACHE_SIZE	[A] statistics occupation log in the first level cache
LBL_DEBUG_IPROXY_SESSIONPOOL_SIZE	[A] occupation log in bytes of the session pool #
LBL_DEBUG_STATISTIC_CACHE_HEADER	stat http header debugging
LBL_DEBUG_LEGACY_COMMAND_HEADER	legacy command http header debugging
LBL_DEBUG_IPROXY_ACTIVESESSIONS_IN_TABLE_SIZE	[A] occupation of active session objects in the session table
TCO_DEBUG_HEADER	Trace SYSTEM internal traffic HTTP HEADER
TCO_DEBUG_IO	IO
TCO_DEBUG_HTTP_SERVER	debug http server and services
TCO_DEBUG_RAWIO	IO (byte by byte)
TCO_DEBUG_AUTHENTICATION	debug internal authentication
TCO_DEBUG_EDITRAWIO	edited IO
TCO_DEBUG_SESSION	expiration of the application session library components TCOProject® (Embedded Application Server)
TCO_DEBUG_STAYALERT	Stay alert debug
LBL_DEBUG_UDP	Debug connettore UDP
LBL_DEBUG_UDP_REWRITING	Debug rewriting connettore UDP
LBL_DEBUG_SERVICE_MANAGEMENT_SERVER	Debug management service
LBL_DEBUG_SERVICE_MANAGEMENT_MESSAGES	Debug messages management service
LBL_PERMIT_UNCOMMITTED_DATA_FORWARDING	Enable forwarding of uncompleted HTTP BODY parts. Default value to false.
LBL_KEEP_ALIVE_EVALUATION	Enables the assessment of the Keep-Alive entity for maintaining connections. Default value to false.
LBL_SURVIVOR_TUNNELS_PERC	a) the number of tunnels must be> = 100 b) there must be at least two group / domain / uripath

Definition	Description
	c) this definition must be> 0
DB_UNDEF	If true sets the values to null with value "undef" Default value to true.
LBL_WHEN_WANT_FORWARD_CLIENT_CERTIFICATES_ANYWAY	if true and there is a request for a client certificate and the parameter is a want to forward the certificate anyway, the default is false
VXLAN	Default: vxlan, docker, vethe

OPLON®Monitor IP address configuration

To use the management console an IP address must be set on which the Monitor process will create a listening instance for incoming requests from the management console.

The configuration files of the MONITOR MANAGEMENT process are contained in (LBL_HOME)/lib/conf

The single file to be modified is: (LBL_HOME)/lib/conf/procsProperties.xml. Change the "LBL_GLOBAL_ADDRESS_MANAGEMENT" parameter which is normally set to "localhost" as shown.

```
<variables>
      <var name="LBL_GLOBAL_ADDRESS_MANAGEMENT"
            value="localhost"/>
      <var name="LBL_GLOBAL_ADDRESS_BROKERWEBCACHE"
            value="localhost"/>
      <var name="LBL_GLOBAL_URL_BROKERWEBCACHE"
            value="http://#LBL_GLOBAL_ADDRESS_BROKERWEBCACHE#:5993"/>
      <var name="LBL_GLOBAL_GARBAGE_COLLECTOR_LOADBALANCER"
            value="-XX:+UseParallelGC -XX:+UseParallelOldGC"/>
      <var name="LBL_GLOBAL_GARBAGE_COLLECTOR_WEB_CACHE_DWH"
            value="-XX:+UseParallelGC -XX:+UseParallelOldGC"/>
<!-- Please NOTE: G1 garbage collector only with jdk 1.7.0_xx. You can find xx
in LBL's compatibility matrix -->
<!--        <var name="LBL_GLOBAL_GARBAGE_COLLECTOR_LOADBALANCER"value="-XX:
+UseG1GC -XX:InitiatingHeapOccupancyPercent=45"/> -->
<!--        <var name="LBL_GLOBAL_GARBAGE_COLLECTOR_WEB_CACHE_DWH"     value="-
XX:+UseG1GC -XX:InitiatingHeapOccupancyPercent=45"/> -->

      <var name="LBL_GLOBAL_START_HEAP_LOADBALANCER"
            value="-Xms1g -Xmx1g"/>
<!-- DWH VARIABLES -->
      <var name="LBL_GLOBAL_DWH_DBNAME"
            value="thin:@__ipAddress__:1521:__instanceName__"/>
      <var name="LBL_GLOBAL_DWH_DBLOGIN"
            value="LBLDWH"/>
      <var name="LBL_GLOBAL_DWH_DBPASSWORD"
            value="lbldwh"/>
</variables>
```

Set the value of the variable to the IP address selected in the network management.

Once the IP address is set, the settings for these variables can be managed via the Management Console.

Definitions of Client SSL connections

SSL protocols

```
java -DSSLPROTOCOLS=SSLv2Hello;SSLv3;TLSv1;TLSv1.1;TLSv1.2;TLSv1.3
```

Definitions of SSL Server connections (listeners)

Listeners and endpoints can have different SSL protocols and different ciphersuite configurations

SUPPORTED CIPHERSUITES AND SSL PROTOCOLS

```
============================
=   OPLON SECURE ACCESS     =
=       ADC WAF GATEWAY     =
=             REL 10.5.x    =
============================

000 SUPPORTED CIPHER SUITES
============================
TLS_AES_256_GCM_SHA384
TLS_AES_128_GCM_SHA256
TLS_CHACHA20_POLY1305_SHA256
TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256
TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256
TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
TLS_DHE_RSA_WITH_AES_256_GCM_SHA384
TLS_DHE_RSA_WITH_CHACHA20_POLY1305_SHA256
TLS_DHE_DSS_WITH_AES_256_GCM_SHA384
TLS_DHE_RSA_WITH_AES_128_GCM_SHA256
TLS_DHE_DSS_WITH_AES_128_GCM_SHA256
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256
TLS_DHE_RSA_WITH_AES_256_CBC_SHA256
TLS_DHE_DSS_WITH_AES_256_CBC_SHA256
TLS_DHE_RSA_WITH_AES_128_CBC_SHA256
TLS_DHE_DSS_WITH_AES_128_CBC_SHA256
```

TLS_ECDH_ECDSA_WITH_AES_256_GCM_SHA384
TLS_ECDH_RSA_WITH_AES_256_GCM_SHA384
TLS_ECDH_ECDSA_WITH_AES_128_GCM_SHA256
TLS_ECDH_RSA_WITH_AES_128_GCM_SHA256
TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA384
TLS_ECDH_RSA_WITH_AES_256_CBC_SHA384
TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA256
TLS_ECDH_RSA_WITH_AES_128_CBC_SHA256
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA
TLS_DHE_RSA_WITH_AES_256_CBC_SHA
TLS_DHE_DSS_WITH_AES_256_CBC_SHA
TLS_DHE_RSA_WITH_AES_128_CBC_SHA
TLS_DHE_DSS_WITH_AES_128_CBC_SHA
TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA
TLS_ECDH_RSA_WITH_AES_256_CBC_SHA
TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA
TLS_ECDH_RSA_WITH_AES_128_CBC_SHA
TLS_RSA_WITH_AES_256_GCM_SHA384
TLS_RSA_WITH_AES_128_GCM_SHA256
TLS_RSA_WITH_AES_256_CBC_SHA256
TLS_RSA_WITH_AES_128_CBC_SHA256
TLS_RSA_WITH_AES_256_CBC_SHA
TLS_RSA_WITH_AES_128_CBC_SHA
TLS_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA
TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA
SSL_DHE_RSA_WITH_3DES_EDE_CBC_SHA
SSL_DHE_DSS_WITH_3DES_EDE_CBC_SHA
TLS_ECDH_ECDSA_WITH_3DES_EDE_CBC_SHA
TLS_ECDH_RSA_WITH_3DES_EDE_CBC_SHA
SSL_RSA_WITH_3DES_EDE_CBC_SHA
TLS_EMPTY_RENEGOTIATION_INFO_SCSV
TLS_DH_anon_WITH_AES_256_GCM_SHA384
TLS_DH_anon_WITH_AES_128_GCM_SHA256
TLS_DH_anon_WITH_AES_256_CBC_SHA256
TLS_ECDH_anon_WITH_AES_256_CBC_SHA
TLS_DH_anon_WITH_AES_256_CBC_SHA
TLS_DH_anon_WITH_AES_128_CBC_SHA256
TLS_ECDH_anon_WITH_AES_128_CBC_SHA
TLS_DH_anon_WITH_AES_128_CBC_SHA
TLS_ECDH_anon_WITH_3DES_EDE_CBC_SHA
SSL_DH_anon_WITH_3DES_EDE_CBC_SHA
TLS_ECDHE_ECDSA_WITH_RC4_128_SHA
TLS_ECDHE_RSA_WITH_RC4_128_SHA
SSL_RSA_WITH_RC4_128_SHA
TLS_ECDH_ECDSA_WITH_RC4_128_SHA

TLS_ECDH_RSA_WITH_RC4_128_SHA
SSL_RSA_WITH_RC4_128_MD5
TLS_ECDH_anon_WITH_RC4_128_SHA
SSL_DH_anon_WITH_RC4_128_MD5
SSL_RSA_WITH_DES_CBC_SHA
SSL_DHE_RSA_WITH_DES_CBC_SHA
SSL_DHE_DSS_WITH_DES_CBC_SHA
SSL_DH_anon_WITH_DES_CBC_SHA
SSL_RSA_EXPORT_WITH_DES40_CBC_SHA
SSL_DHE_RSA_EXPORT_WITH_DES40_CBC_SHA
SSL_DHE_DSS_EXPORT_WITH_DES40_CBC_SHA
SSL_DH_anon_EXPORT_WITH_DES40_CBC_SHA
SSL_RSA_EXPORT_WITH_RC4_40_MD5
SSL_DH_anon_EXPORT_WITH_RC4_40_MD5
TLS_RSA_WITH_NULL_SHA256
TLS_ECDHE_ECDSA_WITH_NULL_SHA
TLS_ECDHE_RSA_WITH_NULL_SHA
SSL_RSA_WITH_NULL_SHA
TLS_ECDH_ECDSA_WITH_NULL_SHA
TLS_ECDH_RSA_WITH_NULL_SHA
TLS_ECDH_anon_WITH_NULL_SHA
SSL_RSA_WITH_NULL_MD5

000 ENABLED CIPHER SUITES
==========
TLS_AES_256_GCM_SHA384
TLS_AES_128_GCM_SHA256
TLS_CHACHA20_POLY1305_SHA256
TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256
TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256
TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
TLS_DHE_RSA_WITH_AES_256_GCM_SHA384
TLS_DHE_RSA_WITH_CHACHA20_POLY1305_SHA256
TLS_DHE_DSS_WITH_AES_256_GCM_SHA384
TLS_DHE_RSA_WITH_AES_128_GCM_SHA256
TLS_DHE_DSS_WITH_AES_128_GCM_SHA256
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256
TLS_DHE_RSA_WITH_AES_256_CBC_SHA256
TLS_DHE_DSS_WITH_AES_256_CBC_SHA256
TLS_DHE_RSA_WITH_AES_128_CBC_SHA256
TLS_DHE_DSS_WITH_AES_128_CBC_SHA256
TLS_ECDH_ECDSA_WITH_AES_256_GCM_SHA384

TLS_ECDH_RSA_WITH_AES_256_GCM_SHA384
TLS_ECDH_ECDSA_WITH_AES_128_GCM_SHA256
TLS_ECDH_RSA_WITH_AES_128_GCM_SHA256
TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA384
TLS_ECDH_RSA_WITH_AES_256_CBC_SHA384
TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA256
TLS_ECDH_RSA_WITH_AES_128_CBC_SHA256
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA
TLS_DHE_RSA_WITH_AES_256_CBC_SHA
TLS_DHE_DSS_WITH_AES_256_CBC_SHA
TLS_DHE_RSA_WITH_AES_128_CBC_SHA
TLS_DHE_DSS_WITH_AES_128_CBC_SHA
TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA
TLS_ECDH_RSA_WITH_AES_256_CBC_SHA
TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA
TLS_ECDH_RSA_WITH_AES_128_CBC_SHA
TLS_RSA_WITH_AES_256_GCM_SHA384
TLS_RSA_WITH_AES_128_GCM_SHA256
TLS_RSA_WITH_AES_256_CBC_SHA256
TLS_RSA_WITH_AES_128_CBC_SHA256
TLS_RSA_WITH_AES_256_CBC_SHA
TLS_RSA_WITH_AES_128_CBC_SHA
TLS_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA
TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA
SSL_DHE_RSA_WITH_3DES_EDE_CBC_SHA
SSL_DHE_DSS_WITH_3DES_EDE_CBC_SHA
TLS_ECDH_ECDSA_WITH_3DES_EDE_CBC_SHA
TLS_ECDH_RSA_WITH_3DES_EDE_CBC_SHA
SSL_RSA_WITH_3DES_EDE_CBC_SHA
TLS_EMPTY_RENEGOTIATION_INFO_SCSV

000 SUPPORTED PROTOCOLS
==========
TLSv1.3
TLSv1.2
TLSv1.1
TLSv1
SSLv3
SSLv2Hello

000 ENABLED PROTOCOLS
==========
TLSv1.3
TLSv1.2
TLSv1.1

TLSv1

001 FORCED ENABLE PROTOCOLS
==========
SSLv3
SSLv2Hello
TLSv1
TLSv1.1
TLSv1.2
TLSv1.3

OPLON®Network Tools

To run network tests LBL ® S.A.A.I. provides a set of basic tools. The testing tools are available and ready to use with the libraries of the distributions once they are unzipped - both LBL ® server side services and LBL ® Management Console.

Performing tests is possible from the command line by using the libraries of the Management Console as well as the libraries from any OPLON®Application Delivery Controller distribution.

OPLON®Management Console:

```
(LBL_MCHOME)/java -classpath LBLManagementConsole_lib.jar
managementconsole.starter.LBLNetworkTest
usage: managementconsole.starter.LBLNetworkTest mail|multicastReceiver|
multicastSender|ping|tcpClient|tcpServer|udpReceiver|udpSender
```

example:

```
managementconsole.starter.LBLNetworkTest multicastReceiver 228.5.6.7
192.168.43.100 6789 6790 6791
```

OPLON®S.A.A.I. server side distributions (OPLON®Application Delivery Controller, OPLON®Surface Cluster... etc):

```
(LBL_HOME)/lib/java -classpath LBLLoadBalancer.jar
loadbalancer.starter.LBLNetworkTest
usage: loadbalancer.starter.LBLNetworkTest mail|multicastReceiver|
multicastSender|ping|tcpClient|tcpServer|udpReceiver|udpSender
```

example:

```
loadbalancer.starter.LBLNetworkTest multicastReceiver 228.5.6.7
192.168.43.100 6789 6790 6791
```

The commands and execution are the same whether using the LBLManagementConsole_lib.jar library or using the LBLLoadBalancer.jar library. For this reason, the following text will use the LBLLoadBalancer.jar title as an example.

The available commands, viewable through the execution without parameters, are as follows:
mail (mail.jar in classpath required)
multicastReceiver
multicastSender
ping
tcpClient
tcpServer
udpReceiver
udpSender

Each command has its own help as in the example below:

java -classpath LBLLoadBalancer.jar loadbalancer.starter.LBLNetworkTest multicastReceiver
LBLNetworkTest multicastReceiver
usage: address interfaceAddress port [...]
example: 228.5.6.7 192.168.43.100 6789 6790 6791

In this case with the instances OPLON®Application Delivery Controller Standard or Enterprise edition running the following or similar would result:

```
(LBL_HOME)/lib/java -classpath LBLLoadBalancer.jar
loadbalancer.starter.LBLNetworkTest multicastReceiver 228.5.6.7
192.168.44.141 6789

LBLNetworkTest multicastReceiver 228.5.6.7 192.168.44.141 6789
MULTICAST RECEIVER 228.5.6.7:6789 on interface: 192.168.44.141
/192.168.44.141 started!
port: 6789 num: 1 hostAddress: legendoneprivate: #LBL(r)LoadBalancer
#Tue Jan 31 12:33:57 CET 2012
healthCheckPublicNetworkHost=legendonegrid
port=5991
sign=1328862510464
groupWeight=100
osVersion=5.0
osName=Windows 2008
outOfOrder=false
subGroup=LBLSubGroup
dist=3
type=1
direction=0
group=LBLEnterpriseGroup
host=legendoneprivate

port: 6789 num: 2 hostAddress: legendtwoprivate: #LBL(r)LoadBalancer
#Tue Jan 31 12:34:00 CET 2012
healthCheckPublicNetworkHost=legendtwogrid
port=5991
sign=1328863884130
groupWeight=100
osVersion=5.0
osName=Windows 2008
outOfOrder=false
subGroup=LBLSubGroup
dist=3
type=1
direction=0
group=LBLEnterpriseGroup
host=legendtwoprivate
```

The information displayed is the result of the exchange of information between two OPLON®Application Delivery Controller Enterprise instances. In this case, given the multicast protocol, it can be noted that all the heart-beat messages from different nodes with their characteristics are “captured”.

Leaving the program running in multicastReceiver the user can move to another node and generate multicast messages with the multicastSender tool:

```
(LBL_HOME)/lib/java -classpath LBLLoadBalancer.jar
loadbalancer.starter.LBLNetworkTest multicastSender 228.5.6.7 6789 1 10
LBLNetworkTest multicastSender 228.5.6.7 6789 1 10
Packets sent: 0
Packets sent: 1
Packets sent: 2
Packets sent: 3
Packets sent: 4
Packets sent: 5
Packets sent: 6
Packets sent: 7
Packets sent: 8
Packets sent: 9
```

If multicast is enabled correctly on the receiving node the following output should result:

```
host=legendoneprivate

port: 6789 num: 53 hostAddress: legendtwobackend: TCOGROUP SRL0
port: 6789 num: 54 hostAddress: legendtwobackend: TCOGROUP SRL1
port: 6789 num: 55 hostAddress: legendtwobackend: TCOGROUP SRL2
port: 6789 num: 56 hostAddress: legendtwobackend: TCOGROUP SRL3
port: 6789 num: 57 hostAddress: legendtwobackend: TCOGROUP SRL4
port: 6789 num: 58 hostAddress: legendtwobackend: TCOGROUP SRL5
port: 6789 num: 59 hostAddress: legendtwobackend: TCOGROUP SRL6
port: 6789 num: 60 hostAddress: legendtwobackend: TCOGROUP SRL7
port: 6789 num: 61 hostAddress: legendtwobackend: TCOGROUP SRL8
port: 6789 num: 62 hostAddress: legendtwobackend: TCOGROUP SRL9
port: 6789 num: 63 hostAddress: legendtwoprivate: #LBL(r)LoadBalancer
#Tue Jan 31 12:40:41 CET 2012
healthCheckPublicNetworkHost=legendtwogrid
port=5991
sign=1328863884130
groupWeight=100
osVersion=5.0
osName=Windows 2008
outOfOrder=false
subGroup=LBLSubGroup
dist=3
type=1
direction=0
group=LBLEnterpriseGroup
host=legendtwoprivate
```

As you can be seen the messages appear along with the output from the LBL ® nodes. This does not involve any malfunction of the nodes that receive test messages since each message contains a fingerprint that, if not validated, is discarded.

Another example of use can be a network performance test:

```
(LBL_HOME)/lib/java -classpath LBLLoadBalancer.jar
loadbalancer.starter.LBLNetworkTest tcpServer
usage: address port maxThreads
example: 192.168.43.100 8086 2000
```

```
(LBL_HOME)/lib/java -classpath LBLLoadBalancer.jar
loadbalancer.starter.LBLNetworkTest tcpServer 192.168.43.141 8086 200
TCP Server listen 192.168.43.141:8086
```

In another node it is possible to execute client function:

```
(LBL_HOME)/lib/java -classpath LBLLoadBalancer.jar
loadbalancer.starter.LBLNetworkTest tcpClient
usage: address port maxThreads createConnTimeout tcpTimeout fragmentLength
waitTimeInterWrite
example: 192.168.43.100 8086 2000 3000 30000 65535 100

(LBL_HOME)/lib/java -classpath LBLLoadBalancer.jar
loadbalancer.starter..LBLNetworkTest tcpClient 192.168.43.141 8086 100 4000
10000 1024 1
 len read=1029 Buffer to forward=1024|wk-
44LLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLL...
 len read=1029 Buffer to forward=1024|wk-
38LLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLL...
 len read=1029 Buffer to forward=1024|wk-
33LLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLL...
 len read=1029 Buffer to forward=1024|wk-
16LLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLL...
 len read=1029 Buffer to forward=1024|wk-
39LLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLL...
 len read=1029 Buffer to forward=1024|wk-
37LLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLL...
```

The result on the tcpServer side is as follows:

```
len read=1029 Buffer to forward=1024|wk-
8LLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLL ....
len read=1029 Buffer to forward=1024|wk-
76LLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLL ...
len read=1029 Buffer to forward=1024|wk-
50LLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLL ...
len read=1029 Buffer to forward=1024|wk-
83LLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLL ...
len read=1029 Buffer to forward=1024|wk-
74LLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLL ...
len read=1029 Buffer to forward=1024|wk-
90LLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLL ...
14.
```

OPLON®Monitor procProperties.xml

(LBL_HOME)/lib/conf/procProperties

In OPLON® Monitor the procProperties file defines names with a value to use in the configuration files of all managed processes.

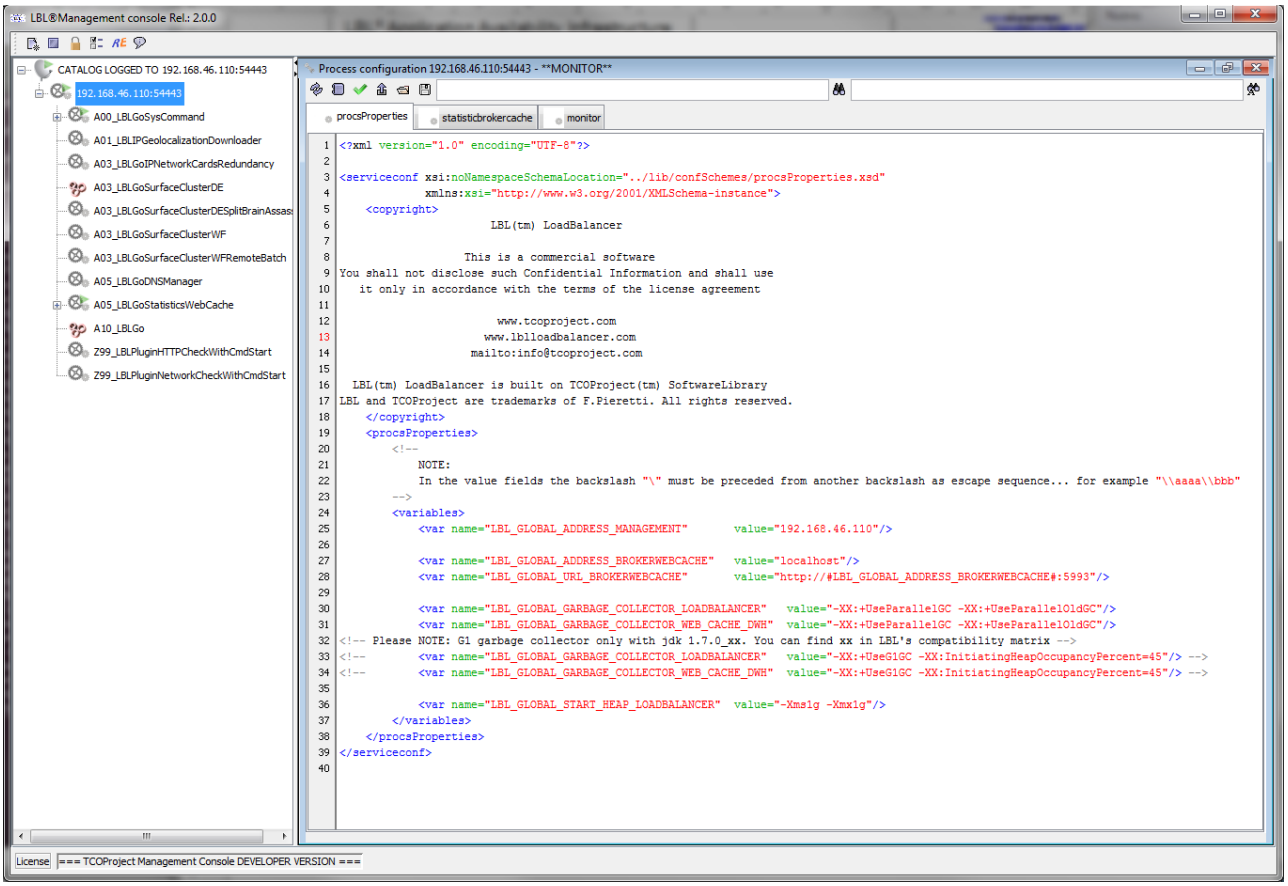


Illustration 1: OPLON®Monitor procProperties file as seen in the management console

The structure is very simple as all that is required is to assign a name and a value in the <variables> section. It is also possible to use previously set variables within the value, so as to create new variables. The variables identified in the <var> section, can be used in all configuration files of the processes. In this case, procProperties at the OPLON® Monitor level, the variables can also be used at the level of the launch profile process (eg.: A10_LBLGo).

It is recommended to declare the variables at this level with names that identifiy their position, which is to say that at the OPLON® Monitor level to use, for example, the suffix "LBL_GLOBAL ..." so as not to overwrite them with the variable declarations at the process level, which as a matter of fact is possible should it be necessary.

The settings hierarchy is then:

1) set variables inside of OPLON® Monitor procProperties

2) set variables within the procProperties of individual processes managed by OPLON® Monitor

The procProperties file is not propagated to other nodes and is typical of the OPLON® Monitor instance or the single process. "procProperties" is in fact used to make similar configuration files for different nodes where the variables declared typify the configuration fo the single-node.

NOTE: The variables used within the processes profile files (e.g. A10_LBLGo.xml) will be used during the process characteristics loading by OPLON® Monitor. If a variable is changed at this level, the process launch profile must be re-uploaded and consequently the process must be restarted with the new definitions.

The file is sub-divided into the following sections:

```
<processconf>
  <copyright>
  </copyright>
  <procsProperties>
    <variables>
      <var/>
      <var/>
      <var/>
        ...
    </variables>
  </procsProperties>
</processconf>
```

```
<procsProperties>
                < variables>
                        < var
```

name=: default value=””
The name of the variable to declare. The name must not contain #. If an already existing variable name is declared, it is overwritten based on the hierarchy of settings.

value=: default value=””
The value that the variable will take on. The value can also be composed of previously created variables eg.

```
<variables>
  <var name="LBL_GLOBAL_ADDRESS_MANAGEMENT"
       value="192.168.46.110"/>
  <var name="LBL_GLOBAL_ADDRESS_BROKERWEBCACHE"
       value="localhost"/>
  <var name="LBL_GLOBAL_URL_BROKERWEBCACHE"
       value="http://#LBL_GLOBAL_ADDRESS_BROKERWEBCACHE#:5993"/>
</variables>
```

In the configuration files, it is therefore possible to replace all or part of the value of a

property.
eg. iproxy.xml:

```
<bind listenType="NAT" address="#LBL_ADDRESS_IPV4_PUBLIC#" port="5050" enable="true"/>
```

OPLON®S.A.A.I. Process specific procProperties

The same considerations apply to this section as those made for the OPLON® Monitor procProperties. These variables are accessed through the properties of the single process.

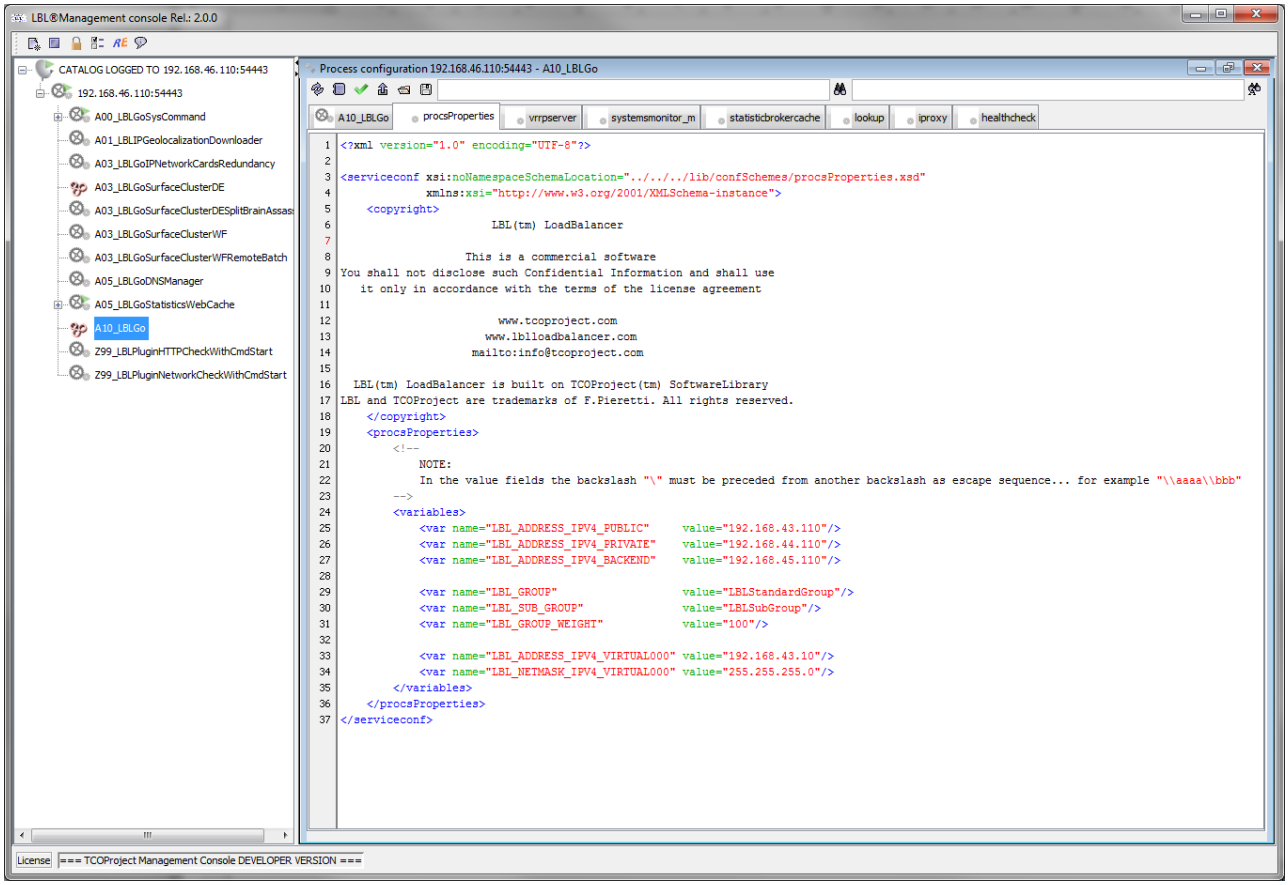


Illustration 2: process variables of (A10 LBLGO)

These variables will be available, unlike variables declared at the OPLON® Monitor level, only for the process to which they belong. The configurations can be updated with the new values through the hot reinit services within the processes.

The file structure is identical to the structure of the OPLON® Monitor global procProperties file.

Logfile

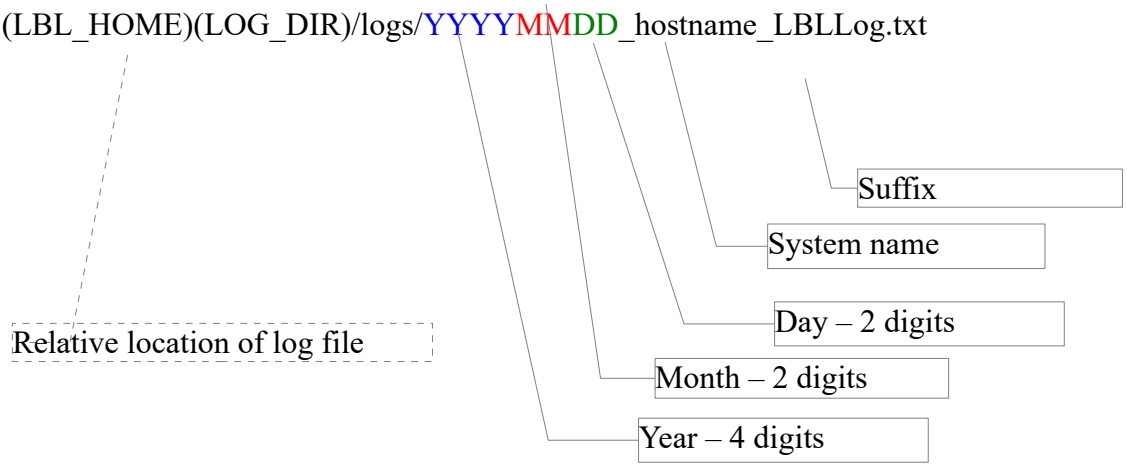
OPLON® S.A.A.I. has a sophisticated system of logging that allows operators, or monitoring systems, to easily identify malfunctions/anomolies. Logging is asynchronous and respects the sequential timeline of the operations. The characteristic of "log-rotation" keeps the history of events for a predefined number of days erasing older files. This will reduce maintenance times of the sytems and maintain consistency in the resources of systems used for storage. Moreover, all log messages are stored in the SYSLOG_EVENT database table, to enable centralized the management of events.

Each LBL A.A.I. process stores its logs in its own relative (LBL_HOME)/procsProfiles/XXX_procName/logs.

The monitor stores its log in :(LBL_HOME)/lib/logs.

The log file names structure is as follows:

(LBL_HOME)(LOG_DIR)/logs/YYYYMMDD_hostname_LBLLog.txt



The log file is in text format with elements separated by a pipe (|). The outline is as follows:

- Type of message:
  DEBUG=debug message
  WARNING= Warning message is not an error
  ERROR= Error message, potential problem to be analyzed especially if persistent
  FATAL= Very serious error: forced stoppage of a thread or unexpected shutdown of the instance
- JVM version
- Name of entity that generated the message
- Name of host that generated the message
- Date expressed in milliseconds from midnight of January 1970 UTC
- Date in readable form
- Message
- Number of consecutive repetitions of the same message

NOTE: The "message" (element 7) may contain newlines and is always terminated by a pipe (|). To protect the pipe character "|" within the elements, use the backslashes "\" as the escape characters/sequence.

example:

```
|WARNING|1.6.0_07!UserService.healthcheck|wilelblonemg|1199522061093|Jan 5, 2008 9:34:21 AM|HTTP
Embedded Server listen on localhost:5991|||
```

2 OPLON®Monitor processName.xml

(LBL_HOME) /lib/confMonitor/PROCESS _NAME.xml

Oplon®Secure Access has a monitoring system for the processes linked to the capabilities of the individual processes/services. The monitoring sysetm starts first and manages the entire life cycle of other processes. From the Monitor one can start, stop and view the log files.
At startup the Monitor verifies the contents of the directory
(LBL_HOME) /lib/confMonitor
and reads the content of all files with .xml extension. If the content has a <processconf> section, the Monitor interprets it and, should the content provide for it, the process is executed.

The file is divided into the following sections:

```
<processconf>
  <copyright>
  </copyright>
  <properties name=”filename”>
    <process>
      <start>
        <env></env>
        <workingDir></workingDir>
        <exec></exec>
        <execStop></execStop>
      </start>
      <warningMessages>
        <message/>
      </warningMessages>
      <errorMessages>
        <message/>
      </errorMessages>
      <restartMessages>
        <message/>
      </restartMessages>
      <alwaysNotifyMessages>
        <message/>
```

```
      </alwaysNotifyMessages>
    </process>
  </properties>
</processconf>
```

Below are brief definitions of the sections and following that is a section by section deep dive on every aspect of configuration.

- <**processconf** > Identifies a process configuration of the Monitor.
- <**copyright >** Section that contains the trademarks and legal rights. All the files, documents and programs are protected by trademark and may not be removed.
- <**properties** > This section identifies the unique name of the process within the process table of the Monitor. Its parameter name should be the file name without the extension, e.g. : <properties name= "A10_LBLGo"> in case the file is called A10_LBLGo.xml
- <p**rocess** > The section that contains the start parameters for the process. Inside this section more starters can be described – for as many operating systems for which this process was designated (see parameters <start> section)
- <**warningMessages** > This section enables filtering of the messages from the standard output and standard error of the running process. For each message indicated the Monitor will log with priority " |WARNING| ".
- <**errorMessages** >This section enables filtering of the messages from the standard output and standard error of the running process. For each message indicated the Monitor will log with priority " |ERROR| ".
- <r**estartMessages** > This section enables filtering of the messages from the standard output and standard error of the running process. For each message indicated the Monitor will log with priority " |ERROR| "and will execute a restart of the process. This section is particularly useful to reinitialize the processes that are still active but that for some reason are not operating as for example in conjunction with "OutOfMemoryError".
- <**alwaysNotifyMessages**>This section enables filtering of the messages from the standard output and standard error of the running process. For each message indicated the Monitor will send a notification via e-mail or via HTTP post. Messages from successful restores, for example a READY AGAIN process, are normally filtered in this section.

NOTE It is important to bear in mind that, being the name of the process is unique to each instance of a Monitor, the name of the xml file must always contain the parameter name of section XML <properties>.
If for example, the file is called "LBLFileName.xml " the first chapter, containing the start parameters must be called " <properties name= "LBLFileName" >"

<processconf>

Contains the configurations of thep rocess managed by the Monitor

<properties>

```
<processconf>
    <properties name= "filename">
```

name = "Process Name"
The name of the process must match the name of the xml file that describes it.

<process>

```
<processconf>
        <properties name= "filename">
                <Process
```

The section process contains the parameters for general behavior of the process.

enable =: default value= "false"
If true indicates that the file descriptor of the process is enabled to start automatically

description =: default value=""
The description of the process.

start =: default value= "automatic"
If "automatic" the process will start automatically at the start of the Monitor. If manual startup is delegated to the operator.

numberTrystartOnFailure=: default value= " 3"
The maximum number of restarts if the job terminates unexpectedly or a restart is executed by the Monitor due to failure message as indicated in the section <restartMessages>. If set to -1 the Monitor performs the restart infinite times. This feature is very useful for performing shellscript control that must be relaunched cyclically. The Monitor attempts the restart of the process every 10" unless otherwise indicated.

waitBeforeKill =: default value= " 5000"
The maximum time to wait after the shutdown command controlled by a process. When this limit is exceeded, if the process is not terminated regularly, the Monitor performs a "kill" (signal 15 on Unix /Linux/BSD). The restart is regulated by the parameters "numberTryStartOnFailure" and "start".

waitBeforeKillOnFailure =: default value= " 5000"
The maximum time to wait after an event of failure. When this limit is exceeded, if the process is not terminated regularly, the Monitor performs a "kill" (signal 15 on Unix /Linux/BSD). The restart is regulated by the parameters "numberTryStartOnFailure" and "start".

managementService =: default value= "true"
This parameter indicates whether the process has the management services at its disposal. Normally if set to true identifies a process LBL A. A. I. and at the start the dynamic port the management service for will be communicated.

confDir =: default value=""
The location of the directory of the configuration file of an LBL process. This value is passed to the application via the parameter -u (see section Start parameters of this manual).

runLevel =: default value= " -1"
The runLevel of an LBL process. If -1 the process is not of type LBL. This value is substituted with %LBL_EXEC_DEFINES% indicated in section <exec>

waitBeforeRestart =: default value= " 3000"
The time to wait before a new reboot (restart) of the service. The restart is controlled by the parameters "numberTryStartOnFailure" and "start".

sysCommand =: default value= "false"
If true characterizes the process as a process for executing system commands.

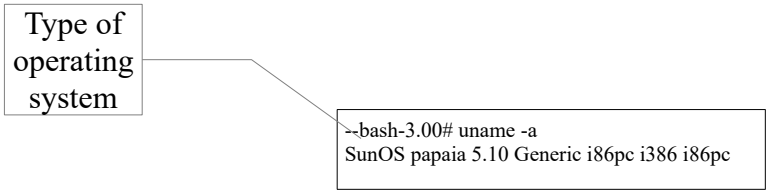
waitBeforeKillexecStop =: default value= " 10000"
The maximum time to wait after the start of the program that performs the stop of the command executed in section <exec> . When this limit is exceeded, if the process is not terminated regularly, the Monitor performs a "kill" (Signal 9 on Unix /Linux/BSD).

sendShutdownCmd =: default value= "true"
The parameter that enables the dispatch of the shutdown command via standard output described in parameter shutdownCmd.

<start>

```
<processconf>
        <properties>
                <process>
                        <start
```

This section identifies, by type of operating system, the parameters of the process to manage. <start> has a single parameter:
osName = : *default value= ""*
Identifier of the operating system. With the command uname -a it is possible to obtain the type of operating system.



For MS Windows, the name is "Windows".

osSoftKill = : *default value* = "false"
Indicates if Java executes an irrevocable command to kill or signals to the process, with reasonable signal, the intention to stop the process. This indication is assessed by the process manager during the stop stage of a process in the event of failure and subsequent restart. The value is also assessed at the end of a programmed process termination. The parameter "osSoftKill" has a default setting of "false". Currently "Linux" and SunOS (Solaris/OpenSolaris) operating systems perform a SoftKill (kill -15) while Microsoft Windows systems perform an irrevocable kill.

=: Default value= "shutdown"
The command written to the standard input of the running process. This command can then be captured by the running program for controlling a graceful (elegant) shutdown of the process.

<env>

```
<processconf>
        <properties>
                <process>
                        <start osName= "XX">
                                <\env>
```

Default value=""
This section may be repeated more than once and identifies a parameter of the environment (environment variable) to set or vary with respect to the parent process (the Monitor).
For example, to make a change or setting to the CLASSPATH ($CLASSPATH on Unix/Linux, %CLASSPATH% on MS Windows) its contents should be:
"CLASSPATH=lib;lib\LBLLoadBalancer.jar " in MS Windows environment and
"CLASSPATH=lib:lib/LBLLoadBalancer.jar " in Unix/Linux environments.
The process being managed inherits the other environment variables from the Monitor.

Ex:

```
<env>CLASSPATH=lib;lib\LBLLoadBalancer.jar;lib\extLib\ojdbc6.jar</env>
```

<workingDir> < /workingdir>

```
        <processconf>
                <properties>
                        <process>
                                <start osName=”XX”>
                                        <workingDir>..</workingDir>
```

default value=""
This section identifies the working directory of the process. If the value is empty the working directory is the same as that of the Monitor.

<exec> < /exec>

```
        <processconf>
                <properties>
                        <process>
                                <start osName=”XX”>
                                        <exec>..</exec>
```

default value=""
The name of the process that must be managed by the Monitor.
Examples:

```
<exec>
        java
        -Xms512m -Xmx1024m -DLBL_INTERACTIVE_CMD=true
        -DLBL_RUNLEVEL=2
        oadbalancer.starter.LBLServerStarterApp”
</ exec>
```

or

```
<exec>
        java lblplugin.LBLNetworkCheckPlugin roadtrbackend 8181
        /notificationDir/outOfOrder.apple 3 10000 lblpluginlogs/logfile.apple false
</ exec>
```

or

```
<exec>T:/papaia/LBLScriptTemplate.bat </ exec>
```

or

```
<exec>/papaia/LBLScriptTemplate.sh</ exec>
```

or

```
<exec> java -jar derbyrun.jar server start</ exec>
```

<execStop> < /execStop>

```
<processconf>
        <properties>
                <process>
                        <start osName=”XX”>
                                <execStop></execStop>
```

default value=""
The name of the process that must perform a graceful shutdown of the process launched in section <exec>. If the section has not specified a process to launch or is empty, no stop commands are executed. In this case, if provided ofr, a soft|Kill is performed and at the end of the allotted time a kill process (signal 9 for Linux/Unix) is executed.

Example of shutdown command :

```
<execStop> java -jar derbyrun.jar server shutdown”</execStop>
```

<warningMessages>

```
<processconf>
        <properties>
                <process>
                        <warningMessages>
```

This section enables filtering of the messages from the standard output and standard error of the running process. For each message indicated the Monitor will log with priority " | WARNING| ".

<message> </message>

```
<processconf>
        <properties>
                <process>
                        <warningMessages>
                                <message
```

message =: default value=""
The value identifies the content of the message to filter. The value is case-sensitive.

<errorMessages>

This section enables filtering of the messages from the standard output and standard error of the running process. For each message indicated the Monitor will log with priority " | ERROR| ".

<message> </message>

```
<processconf>
        <properties>
                <process>
                        <errorMessages>
                                <message
```

message =: default value=""
The value identifies the content of the message to filter. The value is case-sensitive.

<restartMessages>

This section enables filtering of the messages from the standard output and standard error of the running process. For each message indicated the Monitor will log with priority " | ERROR| "and will execute a restart of the process. This section is particularly useful to reinitialize the processes that are still active but that for some reason are not operating as for example in conjunction with "OutOfMemoryError".

<message> </message>

```
<processconf>
        <properties>
                <process>
                        <restartMessages>
                                <message
```

message =: default value=""
The value identifies the content of the message to filter. The value is case-sensitive.

<alwaysNotifyMessages>

This section enables filtering of the messages from the standard output and standard error of the running process. For each message indicated the Monitor will send a notification via e-mail or via HTTP post. Messages from successful restores, for example a READY AGAIN process, are normally filtered in this section.

<message> </message>

```
<processconf>
        <properties>
                <process>
                        <alwaysNotifyMessages>
                                <message
```

message =: default value=""
The value identifies the content of the message to filter. The value is case-sensitive.

1 OPLON®Catalog catalog.xml

The catalog.xml file lists and allows centralized management of OPLON®S.A.A.I. installations within the Datacenter. In order to be used, one must have the appropriate Catalog license.

The file contains two categories of information:

- LBL®S.A.A.I nodes Catalogue.
- Catalog of declared Clusters.

The file structure is as follows:

```
<serviceconf>
    <copyright>
    </copyright>
    <catalog>
      <monitors>
        <monitorID>
        <monitorID>
        <monitorID>
        ...
      </monitors>
      <clusters>
        <clusterGroup>
          <monitorID>
          <monitorID>
          ...
        </clusterGroup>
        <clusterGroup>
          <monitorID>
          <monitorID>
          ...
        </clusterGroup>
          ...
      </clusters>
    </catalog>
```

</serviceconf>

<processconf> This is the section that identifies a process Monitor configuration.

<copyright> The section contains the trademarks and legal rights. All files, documents and programs are protected by trademark and may not be removed.

<catalog> Main section of OPLON®Catalog

<monitors> Section that contains the list of nodes in OPLON®S.A.A.I. managed by OPLON®Catalog

<monitorID> Identifies an OPLON®S.A.A.I. node

<clusters> Section tha defines clusters

<clusterGroup> Section that defines the nodes pertaining to a

<monitorID> Identifies the process petaining to he cluster

<catalog>

Main section of OPLON®Catalog

description=: default value=””
General description of the catalog.

<monitors>

Section that contains the list of nodes in OPLON®S.A.A.I. managed by OPLON®Catalog

< monitorID>

```
<serviceconf>
    <catalog>
        <monitors>
            < monitorID
```

Identifies an OPLON®A.A.I node

address=: default value=”” UM=host:port
Is the address and port of the OPLON®Monitor instance to manage

eg.: address="192.168.46.109:54443"

description=: default value=””
Description of the instance.

<clusters>

Section that defines the clusters.

<clusterGroup>

```
<serviceconf>
        <catalog>
                <clusters>
                        < clusterGroup
```

Within the section are identified the processes tha are part of a cluster.

description=: default value=””
Description of the cluster.

<monitorID>

```
<serviceconf>
        <catalog>
                <clusters>
                        < clusterGroup>
                                <monitorID
```

address=: default value=”” UM=host:port
Is the address and port of the OPLON®Monitor instance to manage

eg.: address="192.168.46.109:54443"

Address can be either addresses or hostnames. If hostnames are used (declared), the names must be resolved by either DNS or "hosts" file.

processName=: default value=””

It is the name of the process that makes up the cluster. It is normally the launch profile name such as: A10_LBLGo or A03_LBLGoSurfaceClusterDE.

There are also private namespaces that identify specific processes as:
****CATALOG****
****MONITOR****
These two names are used to declare catalog and monitors instances as a cluster in order to keep the configurations aligned and redundant.

3 OPLON®Monitor monitor.x ml

This parameter file describes the monitoring service for OPLON®Application Delivery Controller.
The file is composed of the sections:

- <params>
- <surfaceClusterDecisionEngines>
- <surfaceClusterWorkFlows>
- <notifications>

While the <params> section describes the parameters of the embedded application server the <notifications> section describes the parameters which are necessary for the monitor to send email or perform http posts in the event of anomalous events or in the case an operator wants to notify the the control center using notes via the WebConsole. The <serviceconf>, <copyright> and < sysobserver> sections have the same functions in all the configuration files and thus refer to (LBL_HOME)/lib/conf/iproxy.xml for their explanation.

```
<serviceconf>
   <copyright>
   </copyright>
   <monitor>
        <params>
        </params>
        <surfaceClusterDecisionEngines>
                <instance>
                </instance>
        </surfaceClusterDecisionEngines>
        <surfaceClusterWorkFlows>
                <instance>
                </instance>
        </surfaceClusterWorkFlows>
        <notifications>
                <email>
                </email>
                <http>
                </http>
        </notifications>
```

```
                <sysobserver>
                    <service>
                    </service>
                </sysobserver>
            </monitor>
        </serviceconf>
```

<monitor>

The following parameters set the behavior of the monitoring service.

<params>

```
<serviceconf>
    <monitor>
        <params
```

frequency =: default value= " 10000" UM=Milliseconds
The frequency by which to verify status changes in processes or parameters.

monitorConfDir =: default value= "lib/confMonitor"
The directory of the application launch profile files.

managementPortRangeMin =: default value= "5850"
The beginning of the range of ports to dynamically assign the processes for their management.

managementPortRangeMax=: default value= "5990"
The end of the range of ports to dynamically assign the processes for their management.

address =: default value= "localhost"
Bind internal application server web console
The value, which is set by default to localhost, is used for services of the management network. In the absence of a management network it is recommended that this be set to the network at the backend.

addressToConnect=: valore di default=”value in address”
This value, if different from "address", is the address to which the management console should refer in order to access the LBL management services. This is especially useful in those cases where the address "address" on the Monitor which LBL binds and listening is not directly accessible. e.g.: If LBL Monitor was installed on AWS Amazon EC2, to gain access from the outside you must set address = "0.0.0.0" and addressToConnect with the value of public address. In this way you can access to management services from your own desktop.

port =: default value= " 54443"
The port on which the service responds.

backlog =: default value= "20"
The maximum number of incoming connections at the socket-level where the operating system allows this setting.

OPLON®Application Delivery Controller its own connection requests management system in case the operating system does not allow taking advantage of this feature.

reuseAddress =: default value= "true"
The corresponding socket parameter SO_REUSEADDR.

concurrentWorkers =: default value= " 20"
The initial number of requests simultaneously handled from the WebConsole.

maxConcurrentWorkers =: default value= " 100"
The maximum number of requests simultaneously handled from the WebConsole.

healthCheckContextPath =: default value= " /HealthCheck"
The path to the activities healthcheck. This value is usually never changed unless it is already present in other applications.

webAppsDir =: value of default=" lib/webroot/webapps"
Home directory of web applications

webAppsConfDir =: value of default="lib/webroot/webappsconf"
Configuration directory of web applications

webSecurityDir =: value of default=" lib/webroot/websecurity "
Configuration directory for security of web applications

certificateURL =: value of default=" /certificate/serverkeys "
If set indicates the http address from which to get the certificate.

keyStore=: default value= "JKS"
Indicates the type of SSL keystore from which to get the certificate. Normally if the JVM kestore is used it must be set to "JKS" if an OpenSSL keystore is used, it must be set "PKCS12 ".

keyStorePassword =: default value= "defaultpwd"
Password to access the keystore.

alias =: default value= "lblcert"
The identifier of the certificate in the keystore.

aliasPassword =: default value= "defaultpwd"
The password to gain access to the certificate contained in the keystore.

keyManagerFactory =: default value= "SunX509"
This indicates the method of interpretation of the certificate. Normally set to "SunX509".

SSLContextVersion =: default value= "SSLv3"
Indicates the version of the SSL. Normally set to "SSLv3" for the JVM keystore or "TLS" for OpenSSL.

Below are the parameters for the system commands
< !-- System command parameters -->

sysCommandTimeOut =: default value= " 10000" UM=Milliseconds
Indicates the time required to declare time-out on a system command. If the command exceeds this limit an abort command is executed, and subsequently the control is released to the application.

sysCommandCheckRate =: default value= " 300" UM=Milliseconds
The frequency by which to check on the state of the system command.

sysCommandRemoteURL =: default= "https://localhost:5992/sysCommand"
The URL of the service to run system commands

Statistic broker web cache parameters →

statisticBrokerWebCacheURL=: default= "http://localhost:5993"
The URL of the service to request the statistics

The following parameters are for the exchange of information with services →

maxSizeInBuffer =: default= " 10485760" UM=Bytes
The maximum size of the input buffer for exchange of information with services.

DateFormat =: default=”dd/MM/yyyy HH:mm:ss:SSSS”
The format of the date of the statistical data from the Statistic Broker Web Cache and other services

dateFormatDate =:default=”dd/MM/yyyy”
The format of the date that expresses the values day, month, year used for the display

dateFormatTime =: default= "HH:mm:ss"
The format of the date that expresses the values hours, minutes, seconds used for the display

delimiter =: default= " |" (pipe)
The character that delimits the fields during the exchange of information.

<surfaceClusterDecisionEngines>

```
<serviceconf>
    <monitor>
        <surfaceClusterDecisionEngines>
```

This section describes the characteristics of the connection to the OPLON®Surface Cluster Decision Engine services. The parameters can be used to establish the connection to the services and then be able to act centrally for both the verifications as well as execution of actions.

<instance>

```
<serviceconf>
    <monitor>
        <surfaceClusterDecisionEngines>
            <instance
```

The <instance> section describes the connection parameters of an OPLON®Surface Cluster Decision Engine instance .

enable =: default= "false" UM=boolean
Enables or disables the assessment of this section of the instance.

surfaceClusterURL =: default= "https://localhost:54445"
URL to connect to Decision Engine services

description =: default= "Surface Cluster Decision Engine"
Description of the Decision Engine instance

healthCheckUriPath =: default= " /HealthCheck"
URIPath for HealthCheck service

commandUriPath =: default= " /SCDECommand"
URIPath for web service command services

<surfaceClusterWorkFlows>

```
<serviceconf>
    <monitor>
        <surfaceClusterWorkFlows>
```

This section describes the characteristics of the connection to the OPLON®Surface Cluster Work Flow services. The parameters can be used to establish the connection to the services and then be able to act centrally for both the verifications as well as execution of actions.

<instance>

```
<serviceconf>
    <monitor>
        <surfaceClusterWorkFlows>
            <instance
```

The <instance> section describes the connection parameters of an OPLON®Surface Cluster Work Flow instance.

enable =: default= "false" UM=boolean
Enables or disables the assessment of this section of the instance.

surfaceClusterURL =: default= "https://localhost:54444"
URL to connect to Work Flow services.

description =: default= "Surface Cluster Work Flow"

Description of the Work Flow instance.

healthCheckUriPath =: default= " /HealthCheck"
URIPath service of HealthCheck

commandUriPath =: default= " /SCWFCommand"
URIPath for web service command services

<notifications>

```
<serviceconf>
    <monitor>
        <notifications
```

The section which describes the procedures for notification of events from OPLON®Application Delivery Controller outwards. At the moment there are two types of notifications: by e-mail and by HTTP post.

<email>

```
<serviceconf>
    <monitor>
        <notifications>
            <email
```

enable =: default value= "false"
Enables or disables the notification

interval =: default value= " 600000" UM=Milliseconds
The interval of time between one notification and the other. Once this value is exceeded, if there is still a problem, Monitor performs a notification with the first new message from the process being monitoring. All error or restart messages filtered from the profiles of launch (LBL_HOME)/lib/confMonitor/*.xml can cause a notification.

from =: default value="" UM=email address
It is the email address that identifies the site or the control center. If subscribed to the support with notification with TCOProject® verify the clauses of service activation.

fromLogin =: default value="" UM=email address
It is the value of the login of the SMTP server. Often coincides with the "from"

fromPassword =: default value=""
This is the value of the password for the SMTP server if required.

to =: default value="" UM=email address(es)
Identifies the address(es) for the destination(s) of the notification. To enter more than one destination address simply separate the addresses with one or more spaces.

cc =: default value="" UM=email address(es)
Identifies the address(es) for the destination(s) to be placed on copy of the notification. To enter more than one destination address simply separate the addresses with one or more

spaces.

bcc =: default value="" UM=email address/es
Identifies the address(es) for the destination(s) to be placed on blind copy of the notification. To enter more than one destination address simply separate the addresses with one or more spaces.

comment =: default value=""
An alphanumeric free form comment.

<property>

```
<serviceconf>
    <monitor>
        <notifications>
            <email>
                <property
```

This section, which can be repeated for as many parameters are required, is the exact expression of the property used to set parameters in the JavaMail library leveraged by OPLON®Application Delivery Controller and which is necessary to download and install prior to use as described in the installation manual of OPLON®Application Delivery Controller.

name =: default value=""
The name of the property to assess.

Value =: default value=""
The value of the property.

So as to minimize the effort of configuration, OPLON®Application Delivery Controller is distributed with some pre-set parameters and usually only two of them need to be modified.

<property name="mail.smtp.auth" value="true"/>

This property must have a value of true if the SMTP server requires a password to be able to perform the sending of the email.

<property name="mail.transport.protocol" value="smtp"/>

This property is normally set to smtp as the protocol to be able to send the email.

<property name="mail.smtp.host" value="__smtpserver__"/>

This property identifies the name of the server on which the SMTP service runs

<http>

```
<serviceconf>
    <monitor>
        <notifications>
```

<http

Sending notification via HTTP

enable =: default value= "false"
Enables or disables the notification

interval =: default value= " 600000" UM=Milliseconds
The interval of time between one notification and the other. Once this value is exceeded, if there is still a problem, Monitor performs a notification with the first new message from the process being monitoring. All error or restart messages filtered from the profiles of launch (LBL_HOME)/lib/confMonitor/*.xml can cause a notification.

postURL =: default value="" UM=URL
It is the URL to direct all notifications via the function HTTP POST.

comment =: default value=""
An alphanumeric free form comment.

4 OPLON®SysCommand syscommand.xml

The service described in this profile is designated to run system commands. The command is encrypted for security reasons and sent to this service for execution. As of version 7.0 this service is more secure and protected both in the application protocol, with calculation of a digest, as well as with Authentication and Authorization and SSL transmission.

```
<serviceconf>
  <copyright>
  </copyright>
  <syscommand>
     <params>
     </params>
     <sysobserver>
        <service>
        </service>
     </sysobserver>
  </syscommand>
</serviceconf>
```

This file is located in
(LBL_HOME)/procsProfiles/A00_LBLGoSysCommand/conf/syscommand.xml

<syscommand>

Container for parameter setting of the service.

<params>

```
<serviceconf>
        <syscommand>
                <params
```

address =: default value= "localhost"
The value must remain set to localhost for security reasons. Only in special cases can take different values.

port =: default value= " 5992"
The port on which the service responds.

backlog =: default value= "20"
The maximum number of incoming connections at the socket-level where the operating system allows this setting.
OPLON®Application Delivery Controller its own connection requests management system in case the operating system does not allow taking advantage of this feature.

reuseAddress =: default value= "true"
The corresponding socket parameter SO_REUSEADDR.

contextPath =: default value= " /SysCommand"
The context path on which the service is set.

certificateURL =: value of default=" /certificate/serverkeys "
If set indicates the http address from which to get the certificate.

keyStore=: default value= "JKS"
Indicates the type of SSL keystore from which to get the certificate. Normally if the JVM kestore is used it must be set to "JKS" if an OpenSSL keystore is used, it must be set "PKCS12 ".

keyStorePassword =: default value= "defaultpwd"
Password to access the keystore.

alias =: default value= "lblcert"
The identifier of the certificate in the keystore.

aliasPassword =: default value= "defaultpwd"
The password to gain access to the certificate contained in the keystore.

keyManagerFactory =: default value= "SunX509"
This indicates the method of interpretation of the certificate. Normally set to "SunX509".

SSLContextVersion =: default value= "SSLv3"
Indicates the version of the SSL. Normally set to "SSLv3" for the JVM keystore or "TLS" for OpenSSL.

timeOut=: default value= " 1500" UM=thousandths of a second
The vrrpserver service is a HTTP1.0 /1.1 service and this value indicates the connection timeout.

timeOutFactor =: default value= " 300"
The multiplicative factor of time out.

timeOutOpenSocket =: default value= " 5000" UM=Milliseconds
The timeout for the connection of the socket.

timeOutContConnection =: default value= " 30000" UM=Milliseconds

This parameter identifies the timeout during a HTTP 1.1 connection between a consistent reading of one HTTP header and another. This provides the ability to take full advantage of the optimization of the HTTP 1.1 connections/disconnections and adapt to the not always canonical use by the client. A value of -1 disables the feature.

tcpNoDelay =: default value= "true"
Enables/Disables the Nagle algorithm to check the data buffering.

tcpKeepAlive =: default value= "true"
Enable/Disable SO_KEEPALIVE in connections with clients.

concurrentWorkers=: default value= " 10"
The initial number of workers for the resolution of connection requests.

maxConcurrentWorkers=: default value= " 20"
The maximum number of workers for the resolution of connection requests.

webAppsDir =: default value= "lib/webroot_syscommand/webapps"
Home directory of web applications.

webAppsConfDir =: default value= "lib/webroot_syscommand/webappsconf"
Configuration directory of web applications.

webSecurityDir =: default value= "lib/webroot_syscommand/websecurity"
Configuration directory for security of web applications.

5 OPLON®IPGeoLocDownloader iplocalizationdownloader.xml

This file is located at:
(LBL_HOME)/procsProfiles/A10_A01_LBLIPGeolocalizationDownloader/conf/
iplocalizationdownloader.xml

The process of downloading the updated repository for the geolocalization of the ips takes place through this process. Once it is set up and started , it periodically updates from the site tcoproject.com with new associations ip < - > location.

The structure of the file is as follows:

```
<serviceconf>
  <copyright>
  < /copyright>
  <iplocalizationdownloader>
    <params>
    < /params>
    <sysobserver>
      <service>
      < /service>
    < /sysobserver>
  < /iplocalizationdownloader>
< /serviceconf>
```

<iplocalizationdownloader>

Contains the parameters for downloading the database of geolocalization

<params>

downloadDir =: default value= "lib/notificationDir"
The directory in which the repository file will be made available

ipLocalizationFileName =: default value= "LBLGeoIPCountry.geo "

The name of the file once downloaded

downloadURL=: value of default="http://www.tcoproject.com /registration/ipgeolocalization/LBLGeoIPCountry geo.gz ."
The URL from which to execute the download. This parameter must be changed in cases requiring to centralize the download at the level of datacenter.

user =: default value= "empty"
Basic user authentication.

password =: default value= "empty"
Basic password authentication.

fileMaxSize =: default value= " 52428800"
The maximum size of the file that can be loaded into memory. This value exists to protect the memory repository from overly large loads.

proxyAddress =: default value= "empty"
If access to the site tcoproject.com must pass through a proxy this value must be assigned with the address of the proxy.

proxyPort =: default value= " 0"
If access to the site tcoproject.com must pass through a proxy this value must be assigned with the proxy port.

proxyUser =: default value= "empty"
If access to the site tcoproject.com must pass through a proxy with basic authentication this value must be assigned with the user of the proxy.

proxyPassword =: default value= "empty"
If access to the site tcoproject.com must pass through a proxy with basic authentication this value needs to be assigned with the password for the proxy.

expireDays =: default value= " 30" UM=days
The waiting time to download updates of the repository.

keepGzip =: default value= "false"
If true the result of the download is again transformed into .geo.gz to be made available in a centralized manner.

trace =: default value= "false"
If true sets the trace of the downloader.

6 OPLON®Application Delivery Controller iproxy.xml

Load balancing policies settings are located in a single file (LBL_HOME)/procsProfiles/A10_LBLGo/conf/iproxy.xml

The file is subdivided into the following paragraphs:

```
<serviceconf>
  <copyright>
  </copyright>
  <iproxy>
    <listeners>
      <bind>
        <keystoresSNI>
          <keystore>
          </keystore>
        </keystoresSNI>
      </bind>
    </listeners>
    <endPointsGroupingParams>
      <endPointsGrouping>
      </endPointsGrouping>
    </endPointsGroupingParams>
    <params>
    </params>
   <idSessionsManagement>
     <idSessions>
       <id>
       </id>
     </idSessions>
   </idSessionsManagement>
    <dosAddressesQuarantineList>
      <address></address>
      <address></address>
```

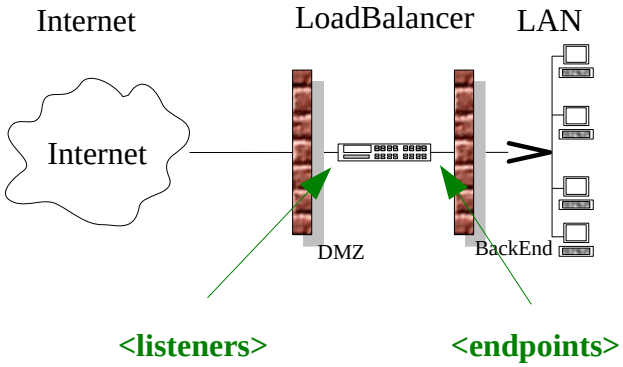
```
        </dosAddressesQuarantineList>
        <cacheControl>
          <cacheControlId>
          </cacheControlId>
        </cacheControl>
        <rewriteManagement>
          <rewriteHeaderRule>
          </rewriteHeaderRule>
          <rewriteBodyRule>
          </rewriteBodyRule>
        </rewriteManagement>
        <endpoints>
          <endPointsGrouping>
            <virtualDomain>
            </virtualDomain>
          </endPointsGrouping>
        </endpoints>
        <sysobserver>
        </sysobserver>
    </iproxy>
    </serviceconf>
```

Following are the definitions of the sections and associated tags used for configuration.

<serviceconf> Since OPLON®Application Delivery Controller is a services based system, this defines the specifics of the configuration.

<copyright> Contains legal rights and marks. All files, documents and programs/code are under commercial trademark protections and may not be removed.

<iproxy> This is the section that is the basis of the load balancer strategy. The load balancing system is defined as a system that receives requests as input and then sorts and distributes them to multiple destinations as shown below:



<listeners> This section will list which network addresses/interfaces will accept connections to be distributed to endpoints in the back end.

<endPointsGroupingParams> Port forwarding to different protocols can make it necessary to customize TCP settings. This section allows you to set the most significant values of the TCP layer 4 treatment (port Forwarding)

<params> Defines the parameters for the balancer such as the size of the interchange data buffer, the number of concurrent threads, the time-outs and how much is used to handle the sending and forwarding of information.

<idSessionsManagement> This is the section used to identify the elements that determine the "session" in HTTP 1.0/1.1 7. In this section you can also instruct OPLON® LoadBalancer to generate and manage the session autonomously (Loadbalancer managed sessions) and manage it in autonomy.

<dosAddressesQuarantineList> This is the section that identifies the ipv6 / ipv4 addresses that should not be evaluated in the function 'DoS Address in Quarantine' of the DoS Attack Prevention module.

<rewriteManagement> Section which describes both the rewriting rules for HTTP HEADER and HTTP BODY contents during the forwarding of information.

<endpoints> Section that indicates the rules for routing of service requests.

<sysobserver> Section mandatory for OPLON® versions Standard and Enterprise Edition LoadBalancer that describes logical physical associations to communicate with other services. Normally it should not be changed except in cases where you want to virtualize more load balancers within the same JVM.

<listeners>

```
<serviceconf>
        <iproxy>
                <listeners>
```

This section will list which network addresses/interfaces will accept connections to be distributed to endpoints in the back end.

<bind>

```
<serviceconf>
        <iproxy>
                <listeners>
                        <bind
```

Subparagraph <bind> specifies the behavior of each individual listener and can contain the following parameters:

description=: default value =”empty”
Listener description.

listenType=: default value =STATIC
May have three different values: STATIC, DAL, NAT.

- **STATIC**: Rigid / hard coded address/network interface.
  This association, already present in version 1.1, provides a fixed combination between listener and IP address. If you were to verify address changes or malfunctions of the network interface OPLON® LoadBalancer ceases to provide the service for that network address. This mode is the DEFAULT if not specified explicitly.

- **NAT** Network Adapter Translation.

Allows OPLON®Application Delivery Controller to adapt the listeners on the same address even if during run-time, this address has been moved from one interface to another.

es. the parameter setting of a NAT listener:

```
... <iproxy>
    <listeners>
      <bind listenType="NAT"
          address="pluto" port="5656" enable="true"/>
      <bind ...
```

- **DAL** Dynamic Address Listen.

The functionality allows OPLON® LoadBalancer to specify a network interface, physical or logical, and the position of an associated IP address. Upon a change to the IP address, OPLON® LoadBalancer will detect that change and create a new bind with the new address and close the previous one. This feature is especially useful in those cases where the network address is reassigned dynamically, DHCP, or to create a temporary "bridge" on ADSL networks by offering a single point of entry to the back-end services.

eg. setting parameters of a DAL listener:

```
... <iproxy>
      <listeners>
         <bind listenType="DAL" monitorTimer="10000"
                netInterface="eth1" subInterface="1"
                port="5454" enable="true"/>
         <bind ...
```

NOTE: do not confuse the Network Address Translation (NAT) with Network Adapter Translation. A (OPLON® LoadBalancer) gateway, by its very nature, hides the end-point addresses under management.

address=: default value =”no default, required value” UM=address list
Specifies the TCP/IP address either as a name or a numeric address. If numeric address indicate:

For Ipv4 eg.: 192.168.43.100
For Ipv6 eg.: [fdd4:3c3f:aaaa::]

It is also possible to specify multiple IP addresses within this parameter. For each IP address a listener will be instantiated with the port(s) indicated in the port parameter(s). eg:

```
<bind listenType="NAT"
          address="localhost [fdd4:3c3f:aaaa::] 192.168.43.10"
          port="80"
          xForwardedFor="true"
          enable="true"/>
```

port=: default value =”no default, required value” UM=integer range
Port on which to accept connections. It is possible to indicate not only a single port but also a range of ports. The syntax for creating a listen on a port range is as follows:

```
port="22,70-1000,8000-9000, 30, 60"
```

This expression indicates to instantiate a listener on port 22, listeners from 70 to 1000 and 8000 to 9000, and listeners on 30 and 60.

portForwarding=: default value =”false”
This indicator serves to enable symmetric forwarding of ports associated with service request. If activated, the forwarding service request will be overturned in the backend to the same port of the request.

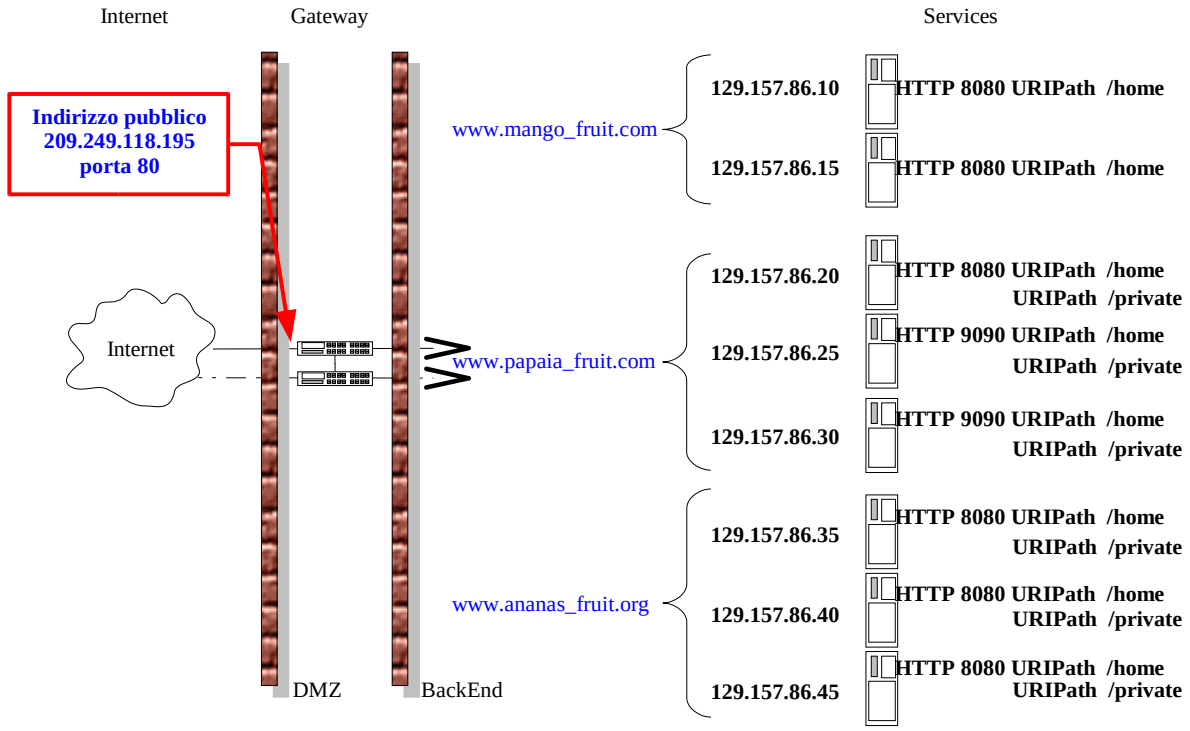
Examples of listeners and endPointsGroupingParams

```
<listeners>
  <!-- listenType ="STATIC" indirizzo statico (default)
                  ="DAL" Dynaminc Address Listen
                  ="NAT" Network Adapter Translation →

        <bind listenType="NAT"
              address="monster" port="80,443,8080"
              osiLayer="4"
              portForwarding="true"
              protocol="pure-forward"
              endPointsGrouping="pureForwardingGroup"
         enable="true"/>
</listeners>
...
...
<endPointsGroupingParams>
  <endPointsGrouping endPointsGroupingName="##pure-forward"
                     clientTimeOut="3600000"
                     clientTimeOutFactor="100"
                     clientTcpNoDelay="true"
                     clientTcpKeepAlive="false"
                     endPointTimeOut="3600010"
                     endPointTimeOutFactor="100"
                     endPointTcpNoDelay="true"
                     endPointTcpKeepAlive="false"
                     endPointNumRetryConnection="0"
                     endPointWaitPerRetryConnection="2"
                     endPointCreateConnectionTimeOut="200"
                     enable="true"/>
</endPointsGroupingParams>
...
...
<endpoints>
  <endPointsGrouping groupName="pureForwardingGroup" enable="true">
    <virtualDomain enable="true">
      <endp address="localhost" healthCheck="false" enable="true"/>
      <endp address="localhost" healthCheck="false" enable="true"/>
    </virtualDomain>
  </endPointsGrouping>
....
```

publicNetworkHealthCheck=: default value =”false”
This parameter is utilized only by the OPLON® Enterprise Edition version. If "true" the address is propagated to the federated nodes as an element to implement and determine the status of the node in the GRID.

enableVirtualDomain=: default value =”false”
If "true", the balancing engine will choose between endpoints only those that match the domain (host name) required by the client. This parameter applies to both the L7 HTTP/HTTS level and the OSI TCP level 4 If the protocol used is HTTP. A layer 4 HTTPS can be evaluated if the ADC component acts as an SSL session terminator.

As an example we can think of a service provider that provides services to multiple customers, each with its own domain.



eg. : www.mango_fruit.com, www.papaia_fruit.com, www.ananas_fruit.org.

Wanting to distribute different services to these domain names using the same TCP/IP port address for acceptance of requests you can configure OPLON® LoadBalancer through VirtualDomain, as described below

The virtualization of domains can be summarised as follows.

The endPointsGrouping is not set and therefore OPLON® LoadBalancer assumes the default. The endPointsGrouping and the virtualDomain can be used simultaneously.

LISTENER

```
<bind listenType="NAT"
  address="lblservice" port="5151"
  enableVirtualDomain="true"
  enable="true"/>
```

END-POINTS

```
<endPointsGrouping enable="true">
  <virtualDomain virtualDomainName="www.mango_fruit.com" enable="true">
        <endp address="129.157.86.10" port="8080" uriPath="home" enable="true"/>
        <endp address="129.157.86.15" port="8080" uriPath="home" enable="true"/>
  </virtualDomain>
  <virtualDomain virtualDomainName="www.papaia_fruit.com" enable="true">
        <endp address="129.157.86.20" port="8080" uriPath="home" enable="true"/>
        <endp address="129.157.86.20" port="8080" uriPath="private" enable="true"/>
        <endp address="129.157.86.25" port="8080" uriPath="home" enable="true"/>
        <endp address="129.157.86.25" port="8080" uriPath="private" enable="true"/>
        <endp address="129.157.86.30" port="8080" uriPath="home" enable="true"/>
        <endp address="129.157.86.30" port="8080" uriPath="private" enable="true"/>
```

```
        </virtualDomain>
        <virtualDomain virtualDomainName="www.ananas_fruit.org" enable="true">
            <endp address="129.157.86.35" port="8080" uriPath="home" enable="true"/>
            <endp address="129.157.86.35" port="8080" uriPath="private" enable="true"/>
            <endp address="129.157.86.40" port="8080" uriPath="home" enable="true"/>
            <endp address="129.157.86.40" port="8080" uriPath="private" enable="true"/>
            <endp address="129.157.86.45" port="8080" uriPath="home" enable="true"/>
            <endp address="129.157.86.45" port="8080" uriPath="private" enable="true"/>
        </virtualDomain>
    </endPointsGrouping>
```

enable=: default value=”true”
Enables or disables the listener. This parameter is used to keep the listener in the configuration file with the parameters set, but disable it during run-time.

backlog=: default value=”2048”
It is the number of connections that the listener can accept before rejecting the connection from the client.

reuseAddress=: default value=”true”
Is the corresponding socket parameter SO_REUSEADDR.

osiLayer=: default value=”7”
Indicates the interpretation of the data stream. The allowed values are: 2, 4, 7.
This parameter is associated with the "protocol" parameter that identifies the type of Protocol.
eg. : ftp, http, etc.. OPLON® LoadBalancer actually interprets at layer “7" HTTP 1.0/1.1 and can execute tunneling at layer “4”of TCP and UDP

protocol=: default value=”http”
Indicates the data stream protocol.
This parameter is bound to the parameter "osiLayer" and sets the connection parameters typical of the Protocol itself. Pre-defined protocol values are: http; ftpCMD; ftpDATA smtp; imap; udp, telnet, ssh, rdp-session-session, rdp-nosession affinity-affinity, ORAC, DBMS.

It is possible to define custom protocols, and associate the parameters through the section <endPointsGroupingParams>.
Other behaviors can be customized via the parameter "endPointGrouping" and its section as described below.

soLinger =: default value = "false"
If true, the soLingerTime is handled when the sockets are closed.

soLingerTime =: default value = "0"
If set to 0 and soLingher is "true", the socket closure propagates a TCP RST.

endPointsGrouping=: default value=”default”
In order to provide heterogeneous services in high availability with centralized management,

OPLON® LoadBalancer provides two types of aggregation that allow maximum flexibility and physical/logical resource partitioning: End-Points Grouping and Domains Virtualization. The End-Points Grouping and Domains Virtualization are the characteristics that rationalize and aggregate in different ways the association between Point (s) of of requests and service Resources.

With the End-Points Grouping we can assign symbolic names to the listeners (points of requests acceptance) and assign them to back-end resources (End-Points)

eg.:
LISTENER

```
<bind listenType="NAT"
  address="lblservice" port="5151"
  endPointsGrouping="ftpservers"
  ... for other parameters see layer 4 OSI/>
```

END-POINTS ASSOCIATI AL LISTENER ftpservers

```
<endPointsGrouping groupName="ftpservers" enable="true">
    <virtualDomain>
          <endp address="lblservicebackend" port="8181" uriPath="" enable="true"/>
          <endp address="lblservicebackend" port="8282" uriPath="" enable="true"/>
    </virtualDomain>
</endPointsGrouping>
```

Additional end-points within the endPointsGrouping will only be used by listeners with the same endPointsGrouping. By adding more endPointsGrouping with different names one can create islands of absolutely separate services and usable only through their listener/s.

The End-Points Grouping can be used both on layer 7 HTTP/S and on OSI layer 4 (port forwarding). At layer 4 OSI (port forwarding) this feature is used to aggregate services of the same type e.g.: ftp, smtp etc... On layer 7 HTTP/S the endPointsGrouping is used only as a group, associated TCP parameters only used on layer 4 TCP (port forwarding) or UDP are not taken into consideration.

transport=: default value=”tcp”
Indicates the transport protocol. Presently the possible transport protocols are: tcp; udp.

transportSessionAffinity=: default value=”false”
Session Affinity is possbile even at the transport layer, layer 4, and is achieved by setting this parameter to "true". The session will be distinguishable from the address of the client that is requesting the services.

TransportSessionAffinityType =: Default value = "IP" ("IP", "SESSION")
At Layer 4 TCP and with HTTP protocol, you can use IP-based session management features or a IdSessionsManagerName as a layer 7 HTTP.
If IP-based, the default, the session is manteined through the use of the client's IP.
If SESSION-based, the IdSessionsManagerName is evaluated to set the routing.

xForwardedFor=: default value=”false”
If set to true manages the entity X-Forwarded-For. The entity X-Forwarded-For is defined with a comma + space separated value and indicates from left to right, the IP address of the request until the last crossing.

eg.:
X-Forwarded-For: 192.168.43.100, 192.168.43.123

forwardHeaderCerts=: valore di default=”false”
If set to false it cleans the certificates in the header or in any case cleans up certificates not coming from the Socket..

h2Bridge=: default=”false”
If true, the listener will be able to establish a connection with HTTP2 protocol and maintain endpoints with HTTP1.1 protocol. All rewrite rules and endpoint application routing behavior remain unaffected. See also SSLApplicationProtocols.

h2MaxConcurrentStream=: default=”5”
It is the maximum number of simultaneous parallel connections (number of tunnels) that will be used to communicate with the endpoints related to an HTTP2 connection.

h2InitialWindowUpdateSize=: default=”3153920”
In some cases, the HTTP2 client does not perform the initial window_update leaving the default initial window size at 65535 (+1). It is used to set the initial upload value of the flow control to allow the upload. If the client increases the default value with window_update it is not used.

distinguishSingleConnection=: default value=”false”
If set to true for each TCP connection inserts the information LBLCOLOR in the statistical database in the L4_TCP_TCPSSL table within the COOKIES field.
This value is unique with respect to the connection and allows one to distinguish the connection as it evolves.

tcpInterceptorPrimerCapture=: default value=”true”
At the first client connection trigger (doPrimerFromClient), depending on the Protocol, one can exclude reading of the first packet coming from the client through the parameter tcpInterceptorPrimerCapture = "false". This feature must be disabled in all cases where one is performing the rewriting of protocols that do not call for a primer on the part of the client (e.g., telnet). The tcpInterceptorPrimerCapture parameter has no effect unless the code explicitly uses a rewriting TCP class.

tcpInterceptorClassPath=: default value=”true”
The loading path of the interceptor class defined in tcpInterceptorClass. This path is added to the classpath of the JVM that is running.

tcpInterceptorClass=: default value=”null”
Indicates the class to intercept TCP packets. This class is an extension of the class: loadbalancer.rewriter.LBLTCPRewriteInterceptorAbstr.

The distributions already contain a class template available in:

(LBL_HOME)/interceptors/rewriteclasses/LBLTCPRewriteInterceptorLogging.java

The classes that are contained in this directory can be compiled through the tools: compile.sh or compile.bat present in the directory itself. Ways to use the interceptor classes templates is described within the classes themsleves. This class implements 3 methods for intervening on both flow and routing.

```
public void doPrimerFromClient( tcpFragment)

public void doPacketFromClient(LBLTCPRewriteInterceptorFragment tcpFragment)

public  void  doPrimerFromEndpoint(LBLTCPRewriteInterceptorFragment  tcpFragment)

public void doPacketFromEndpoint(LBLTCPRewriteInterceptorFragment tcpFragment)
```

The first method is invoked after the first trigger of the client that makes the request, the second method is invoked for each packet that passes from the client toward the endpoint, the third method is invoked on the first packet that passes from the endpoint and the fourth method is invoked for each packet that passes from the endpoint toward the client.

NOTE: The two methods doPacketFromClient and doPacketFromEndpoint can also be used at the same time as long as the stream is full-duplex.

At the first client connection trigger (doPrimerFromClient), depending on the Protocol, one can exclude reading of the first packet coming from the client through the parameter tcpInterceptorPrimerCapture = "false". This feature must be disabled in all cases where one is performing the rewriting of protocols that do not call for a primer on the part of the client (e.g., telnet). The tcpInterceptorPrimerCapture parameter has no effect unless the code explicitly uses a rewriting TCP class.

Below is a list of functions provided by the class:
LBLTCPRewriteInterceptorFragment
The tcpFragment fragment passed in the call-back allows access different features of flow control and modification

```
/**
* buffer stream getter
* @return buffer stream or null if error
*/
public byte[] getStream()

/**
* set a new stream buffer
* @param newBufferStream
* @throws IOException
*/
public void setStream(byte[] newBufferStream) throws IOException
```

```
/**
* return client host address
* @return client host address or null  if not found
*/
public String getRequestClientAddress()

/**
* return incoming host address
* @return incoming host address or null  if not found
*/
public String getRequestIncomingAddress()

/**
* return incoming socket
* @return incoming socket or null if not found
*/
public Socket getRequestIncomingSocket()

/**
* return incoming SSLSocket or null if not found or not SSL Socket
* @return incoming SSLSocket or null if not found or not SSL Socket
*/
public SSLSocket getRequestIncomingSSLSocket()

/**
* Session SSL socket connected to the incoming
* @return SSL session connected to the incoming socket,
* or null if no SSL or non-existent socket
*/
public SSLSession getRequestIncomingSSLSession()
/**
* Peer certificates connected to the incoming socket
* @return Peer certificates connected to the incoming socket,
* or null if no SSL or non-existent socket
*/
public java.security.cert.Certificate[] getRequestIncomingSSLCertificates()

/**
* return incoming host name or address
* @return incoming host host name or address or null  if not found
*/
public String getRequestIncomingHostName()

/**
* client ssl connection
* @return true if client ssl connection
*/
public String isSSLClientConnection()

/**
* return endpoint socket
* @return endpoint socket or null if not found
*/
public Socket getResponseEndpointSocket()

/**
* return endpoint SSLSocket or null if not found or not SSL Socket
```

```
 * @return endpoint SSLSocket or null if not found or not SSL Socket
 */
 public SSLSocket getResponseEndpointSSLSocket()

 /**
 * SSL session connected to the incoming socket endpoint
 * @return SSL session endpoints connected to the socket,
 * or null if the incoming non-SSL sockets or nonexistent
 */
 public SSLSession getResponseEndpointSSLSession()

 /**
 * Peer certificates connected to the socket endpoint
 * @return Peer certificates connected to the socket endpoint,
 * or null if no SSL or non-existent socket
 */
 public java.security.cert.Certificate[] getResponseEndpointSSLCertificates()

 /**
 * return endpoint host address
 * @return endpoint host address or null  if not found
 */
 public String getResponseEndpointAddress()

 /**
 * endpoint ssl connection
 * @return true if endpoint ssl connection
 */
 public String isSSLEndpointConnection()

 /**
 * ssl reencryption
 * @return true if in ssl reencryption
 */
 public String isSSLReencryptionConnection()

 /**
 * if != null endPointsGrouping name on which displace the request
 * @return the endPointsGrouping
 */
 public String getEndPointsGrouping()

 /**
  * if != null endPointsGrouping name on which displace the request
  * @param endPointsGrouping the endPointsGrouping to isplace the request
 */
 public void setEndPointsGrouping(String endPointsGrouping)
```

udpInterceptorClassPath=:default=”interceptors/”
The loading path of the interceptor class defined in udpInterceptorClass. This path is added to the classpath of the JVM that is running.

udpInterceptorClass=: default value=”null”
Indicates the class to intercept the UDP data packet. This class is an extension of the class:loadbalancer.rewriter.LBLUDPRewriteInterceptorAbstr.

The distributions already contain a class template available in (LBL_HOME)/interceptors/rewriteclasses/LBLUDPRewriteInterceptorLogging.java

The classes that are contained in this directory can be compiled through the tools: compile.sh or compile.bat present in the directory itself. Ways to use the interceptor classes templates is described within the classes themsleves. This class implements 2 methods for intervening on both flow and routing.

The class must implemented with the following methods:

```
/**
 * Abstract method called after receive UDP Packet from client before send to endpoint
 * @param udpFragment string fragment before replace
 */
public abstract void doAfterReceivedUDPPacketFromClient(LBLUDPRewriteInterceptorFragment
udpFragment);

/**
 * Abstract method called after received UDP Packet from endpoint before resend to client
 * @param udpFragment string fragment before replace
 */
public abstract void doAfterReceivedUDPPacketFromEndpoint(LBLUDPRewriteInterceptorFragment
udpFragment);
```

You can exclude reading of the UDP packet back from the service by changing the timeout waiting to -1. The default value of awaiting the return package service is 5000 milliseconds:

```
<endPointsGroupingParams>
        ...
        ...
        <endPointsGrouping endPointsGroupingName="##udp"
                           endPointTimeOut="5000"/>    <!-- if set -1 not read packet from service ->
        ...
        ...
</endPointsGroupingParams>
```

The udpFragment contains the methods for accessing the UDP packet:

```
        /**
         * input buffer. This is not a copy of buffer.
         * Remind to setPacketLength after used the array.
         * @return input buffer
         */
        public byte[] getPacketByteArray()

        /**
         * Write pointer of the array
         * @return Write pointer of the array
         */
        public int getPacketLength()

        /**
         * Set a write pointer of the array
         * @param wp write pointer
         * @throws IOException
         */
```

```
public void setPacketLength(int wp)

/**
 * endpoint address before rewriting
 * @return the endpointAddress
 */
public InetAddress getEndPointAddress()

/**
 * endpoint address before rewriting
 * @param endPointAddress the endpointAddress to set
 */
public void setEndPointAddress(InetAddress endPointAddress)

/**
 * Return a local incoming port
 * @return local incoming port
 */
public int getLocalIncomingPort()

/**
 * Return a local incoming Inet Address
 * @return local incoming Inet Address
 */
public InetAddress getLocalIncomingInetAddress()

/**
 * Return the host inet address of the client that sent the packet
 * @returnhost inet address of the client that sent the packet
 */
public InetAddress getClientHostAddress()
```

layer2QueueIn=: valore di default=””
Frontend nfqueue queue number used in layer2 balancing.

layer2QueueOut=: valore di default=””
Backenfd nfqueue queue number used in layer2 balancing.

SSL=: default value=”false”
Indicates whether the listener will work as an SSL encrypted transmission terminator. If set to true, the following parameters are required.

SSLSNI=: default value=”false”
If the value is set to true the listener accepts TLS SNI connections. In this case, you can declare a list of keystore in paragraph <keystoresSNI> that may contain several digital certificates.

SSLSNIDefaultCertificateEnable =: default = "false"
If true, in case the client is not SNI enabled, the domain (host name) present in the SSLSNIDefaultCertificateDomainName parameter will be used.

SSLSNIDefaultCertificateDomainName =: default = ""
If SSLSNIDefaultCertificateEnable set to true, the parameter must contain the host name

that will be used for non-SNI compliant clients. ex .: www.myhostname.com

SSLContextVersion=: default value=”SSLv3”
Indicates the version of the SSL protocol. Normally set to "SSLv3" for JVM keystore or "TLS" for OpenSSL.

sslSessionCacheSize=: default value=”0” UM=entry
The number of entries in the SSL cache session table.

sslSessionCacheTimeout=: default value=”86400” UM=secondi
The SSL session timeout in SSL cache if not used.

certificateURL=: default value=””
If set, indicates the http address to collect the certificate.

certificateURIPath=: default value=””
If set, indicates the path to collect the certificate.

keyStore=: default value=”JKS”
Indicates the SSL keystore type from which to collect the certificate. Normally if the JVM keystore is used, this must be set to "JKS". If an OpenSSL keystore is used, this must be set to “PKCS12”.

keyStorePassword=: default value=”defaultpwd”
Password for accessing the keystore.

keyManagerFactory=: default value=”SunX509”
Indicates the interpretation paradigm of the certificate. Normally set to "SunX509".

alias=: default value=”lblcert”
The identifier of the certificate within the keystore.

aliasPassword=: default value=””
The password for accessing the certificate contained in the keystore.

needClientAuthentication=: default value=”false” UM=lista di valori
If set to "true" and is an SSL transmission, a client identification certificate is requested.
If set to "false", a certificate from the client is not requested.
If the L7 HTTP/S listener is set to "want" in the presence of the certificate is required of the client otherwise continue.
"Want" is normally used with a client authentication concerning a specific URIPath.
For this need, verify parameters "needClientCert" and "needClientCertMessage" in sections: <endPointsGrouping>; <virtualDomain>; <endp>.

trustAllCertificates=: default value=”false”
If this value is set to true, there is NO certificate verification through CA or truststore. Useful in testing or if the forwarding of all certificates through reencryption is desired.

checkClientCertificateValidity=: default value=”false”
If this value is set to true a check on dates of validity of the certificate is executed.

forwardClientCertificateChainDepth=: default value=”1”
The depth of forwarding the certificate chain. If associated to the parameter "forwardClientPemCertificateToEndpoint" set this value to the 2 KB approximately required for each certificate that is passed in the HTTP HEADER to the service.

forwardClientCertificateToEndpoint=: default value=”false”
If this value is set to true, the client certificate information is transferred to the service. The information transferred to the service in the form of entity is:

```
x-fwdcertserialnumber_0: 14587188816555638983
x-fwdcertdatenotbefore_0: 2010-01-15 10:59:18.0 UTC
x-fwdcertdatenotafter_0: 2011-01-15 10:59:18.0 UTC
x-fwdcertsubject_0: EMAILADDRESS=info@tcoproject.com, CN=LBL the best!
LoadBalancer Certificato di esempio, OU=TCOProject(r) CA, O=TCOGROUP SRL,
L=Abano Terme, ST=Italy, C=IT
x-fwdcertissuer_0: EMAILADDRESS=info@tcoproject.com, CN=LBL the best!
LoadBalancer Certificato di esempio, OU=TCOProject(r) CA, O=TCOGROUP SRL,
L=Abano Terme, ST=Italy, C=IT
```

The final part of the entity that is identified by an _ (underscore) indicates the depth of the certificate chain. The certificate chain depth can be set through the forwardClientCertificateChainDepth parameter.

forwardClientPemCertificateToEndpoint=: default value=”false”
If set to true the certificate chain from client is pem-encoded and then re-encoded to UTF-8 to be transferred in the HTTP header to the service. The entity in the HTTP HEADER takes a value similar to the following:

```
x-fwdcertencodedpem_0: -----BEGIN+CERTIFICATE-----%0D%0AMIIFJzCCBA
%2BgAwIBAgIJAMpwGvFCWYDHMA0GCSqGSIb3DQEBBQUAMIG9MQswCQYDVQQGEwJJVDEOMAwGA1U
ECBMFSXRhbHkx%0AFDASBgNVBAcTC0FiYW5vIFRlcm1lM
.... ///// certificato ////
VPNuyd%2FAqN%2BGZz%2BREwdyDPrHQMTt%2BtIVT1bWmmcPfACj0LvuNBt6gyrelICZzX1f7rG
%0AWEy3kZzTgOxAjjChvUaobvy9hGbmb4r4FCT8%0D%0A-----END+CERTIFICATE-----%0D
%0A
```

The size of each individual certificate is approximately 2 KB. The number of certificates in the certificate chain sent to endpoints is relative to the parameter “forwardClientCertificateChainDepth”.

trustCertificateURL=: default value=””
If set, indicates the http address to collect the trust certificate.

trustCertificateURIPath=: default value=””
If set, indicates the path to collect the trust certificate.

trustKeyStore=: default value=”JKS”
Indicates the type of SSL trust keystore. Normally if the JVM keystore is used, this must be

set to "JKS". If an OpenSSL keystore is used, this must be set to “PKCS12”.

trustKeyStorePassword=: default value=”defaultpwd”
Password for accessing the trust keystore.

trustKeyManagerFactory=: default value=”SunX509”
Indicates the interpretation paradigm of the certificate. Normally set to "SunX509".

netInterface=: default value=””
If "listenType" is set to "DAL" (Dynamic Address Listen), indicates the name of the network interface on which to listen.
Eg.: “eth0”

subInterface=: default value=”0”
If "listenType" set to "DAL" (Dynamic Address Listen), indicates the number of network interface aliases on which to listen.

Eg.: “eth0:1” the value will be “1”

monitorTimer=: default value=”10000” UM=Millisecondi
If "listenType" set to "DAL" (Dynamic Address Listen), indicates in milliseconds the amount of time to detect changes in interface settings during run-time.

doubleIncomingQueues=: default value=”false”
This parameter enables the listener to use a private queue of incoming connections. This feature was designed to provide "network" freezing” functionality which enables the “accumulation” of incoming connections in a queue and not directly on the forwarding threads thus avoiding "Slingshot effect " upon operations restoration and excessive load from backend services.

doubleIncomingQueuesHighWater=: default value=”5000”
This parameter sets the maximum number of connection requests that can be inserted in the queue before the system begins to report an overfilling. When twice this value is exceeded, incoming connections will be closed and no longer queued.

cipherSuites =: default value = ""
The parameter sets the ciphersuites to be used for communication with the backend.

SSLApplicationProtocols =: default value = ""
In this parameter you can list the protocols enabled on ALPN communications. The allowed values "h2 http / 1.1 undef".

```
<esempio: <params
...
endPointSSLApplicationProtocols="h2 http/1.1 undef"
endPointSSLUseCipherSuitesOrder="true"
.../>

<endPointsGrouping enable="true" SSLApplicationProtocols="h2 http/1.1 undef"
        SSLUseCipherSuitesOrder="true">
```

```
<virtualDomain enable="true" SSLApplicationProtocols="h2 http/1.1 undef"
        SSLUseCipherSuitesOrder="true">
        <endp address="192.168.56.131" port="8080" uriPath="/"
                SSLApplicationProtocols="h2 http/1.1 undef"
          SSLUseCipherSuitesOrder="true" enable="true"/>
```

SSLUseCipherSuitesOrder =: default value = "true"
If true, it uses the chip-suites in the order indicated for the SSL / TLS listeners

SSLProtocols =: default value = ""
The parameter sets the SSL protocols to be used for communication.

<keystoresSNI>

```
<serviceconf>
    <iproxy>
        <listeners>
            <bind>
                <keystoresSNI>
```

When the SSLSNI parameter is set to "true", the keystoresSNI section contains a list of SSL keystore with digital certificates associated with the managed host names in paragraphs virtualDomain. This section can be used an unlimited number of listeners to the same keystore (port address). Each keystore may contain multiple certificates with the same alias password associated with the host (domain) names described in the CN (common name) of individual certificates. It is possible to obtain various certificates with different password by simply using more keystores.

<keystore>

```
<serviceconf>
    <iproxy>
        <listeners>
            <bind>
                <keystoresSNI>
                    <keystore
```

Single keystore settings. A keystore may contain an unlimited number of digital certificates with different CN, which must have the same password. To use different passwords just add more keystore.

enable=: default value=”true”
Enables disables the use of the keystore.

description=: default value=””

certificateURIPath=: default value=””
Indicates the location of the keystore that contains the certificates. If the path is relative it

starts from LBL_HOME

keyStore=: default value=”JKS”
Indicates the SSL keystore type from which to withdraw the certificate.
Normally if you use the keystore of the JVM must be set to "JKS".

keyStorePassword=: default value=”defaultpwd”
Password to access the keystore.

aliasPassword=: default value=””
It is the certificate password in the keystore. All certificates are contained in the same keystore must have the same password. Certificates with different passwords can be managed using multiple keystore.

keyManagerFactory=: default value=”SunX509”
Indicates the interpretation of the certificate. Normally set to "SunX509"

EX:

```
<bind listenType="NAT" address="#LBL_ADDRESS_IPV4_PUBLIC#" port="443"
        SSLSNI="true"
        SSL="true"
        certificateURL=""
        certificateURIPath="security/certificate/serverkeys"
        keyStore="JKS"
        keyStorePassword="defaultpwd"
        keyManagerFactory="SunX509"
        SSLContextVersion="TLS"
        enable="true">
       <keystoresSNI>
            <keystore description="Keystore 1"
                    certificateURIPath="security/certificate/serverKeyStoreMulti.jks"
                     keyStore="JKS"
                     keyStorePassword="defaultpwd"
                     aliasPassword="adminadmin"
                     keyManagerFactory="SunX509"/>
            <keystore certificateURIPath="security/certificate/serverKeyStoreMultiOrg.jks"
                    keyStore="JKS"
                    keyStorePassword="defaultpwd"
                    aliasPassword="defaultpwd"
                    keyManagerFactory="SunX509"/>
       </keystoresSNI>
```

Note: IIf you want to use this configuration for the test, you can set in your hosts file the following associations addresses/names. These keystore are distributed for testing and are already present. File /etc/hosts or equivalent in ms windows os:
---fragment start
192.168.99.999 pippo.luigi.it www.luigi.it wwww.jason.com www.valerio.it
192.168. 99.999 www.pwd.it pippo.valerio.it
---fragment end
NOTE: pippo.valerio.it is not present in the list and produces an error if invoked by browser.
NOTE2: Jason's name association has 4 w (wwww.jason.com) :-)

When loading the keystores you will notice errors in the logs. This is due to the fact that LBL reports if there are any same CN on different keystores. When loading printed the list of CN loaded into memory and it will

be managed.

LOG EX:

```
|ERROR|1.8.0_05|UserService.Listener:192.168.43.110:443
KEYSTORE: Description=First keystore:  keystoreType=JKS keyManagerFactory=SunX509 keystorePath=security/certificate/serverkeys
KEYSTORE: Description=Keystore 1 keystoreType=JKS keyManagerFactory=SunX509 keystorePath=security/certificate/serverKeyStoreMulti.jks
KEYSTORE: Description=Keystore description keystoreType=JKS keyManagerFactory=SunX509
keystorePath=security/certificate/serverKeyStoreMultiOrg.jks|LBLR9SDN|1408699794205|20140822-02:29:54|Certificate discarded for CN duplication:
alias=defaultpwd uniqueAlias=-265999823_defaultpwd CN=lbl(r)loadbalancer certificato di esempio||
|ERROR|1.8.0_05|UserService.Listener:192.168.43.110:443
KEYSTORE: Description=First keystore:  keystoreType=JKS keyManagerFactory=SunX509 keystorePath=security/certificate/serverkeys
KEYSTORE: Description=Keystore 1 keystoreType=JKS keyManagerFactory=SunX509 keystorePath=security/certificate/serverKeyStoreMulti.jks
KEYSTORE: Description=Keystore description keystoreType=JKS keyManagerFactory=SunX509
keystorePath=security/certificate/serverKeyStoreMultiOrg.jks|LBLR9SDN|1408699794207|20140822-02:29:54|Certificate discarded for CN duplication:
alias=mykey uniqueAlias=1668377580_mykey CN=localhost||
|WARNING|1.8.0_05|UserService.Listener:192.168.43.110:443
KEYSTORE: Description=First keystore:  keystoreType=JKS keyManagerFactory=SunX509 keystorePath=security/certificate/serverkeys
KEYSTORE: Description=Keystore 1 keystoreType=JKS keyManagerFactory=SunX509 keystorePath=security/certificate/serverKeyStoreMulti.jks
KEYSTORE: Description=Keystore description keystoreType=JKS keyManagerFactory=SunX509
keystorePath=security/certificate/serverKeyStoreMultiOrg.jks|LBLR9SDN|1408699794207|20140822-02:29:54|
CERTIFICATES LIST LOADED:
alias=lblcert uniqueAlias=-265999823_lblcert CN=lbl(r)loadbalancer certificato di esempio
alias=calbl uniqueAlias=-265999823_calbl CN=tcoproject(r) certificato di esempio
alias=mykey uniqueAlias=1851544254_mykey CN=localhost
alias=mywildecard uniqueAlias=1851544254_mywildecard CN=*.luigi.it
alias=www.luigi.it uniqueAlias=1851544254_www.luigi.it CN=www.luigi.it
alias=www.jason.com uniqueAlias=1851544254_www.jason.com CN=wwww.jason.com
alias=www.pwd.it uniqueAlias=1668377580_www.pwd.it CN=www.pwd.it
||
```

Examples <bind>

Some examples of listeners <bind>

- hostname

```
<bind address="lblservice" port="5050" enable="true"/>
```

- network interface

```
<bind netInterface="eth0" port="5454" enable="true"/>
```

- hostname, SSL terminator keystore JVM

```
<bind address="lblservice" port="1443"
        SSL="true"
        certificateURL=""
        certificateURIPath="security/certificate/serverkeys"
        keyStore="JKS"
        keyStorePassword="defaultpwd"
        alias="lblcert"
        aliasPassword="defaultpwd"
        keyManagerFactory="SunX509"
        SSLContextVersion="SSLv3"
        enable="true"/>
```

- hostname, SSL terminator keystore OpenSSL

```
<bind address="lblservice" port="1443"
        SSL="true"
        certificateURL=""
        certificateURIPath="security/certificate/mycert.p12"
        keyStore="PKCS12"
```

```
keyStorePassword="defaultpwd"
alias="mycert"
aliasPassword="defaultpwd"
keyManagerFactory="SunX509"
SSLContextVersion="TLS"
enable="true"/>
```

- hostname, SSL terminator keystore OpenSSL + certificate forward and trust store

```
<bind listenType="NAT"
    address="lblservice" port="443"
    enableVirtualDomain="false"
    backlog="2048"

    SSL="true"

    alias="lblcertsigned"
    aliasPassword="defaultpwd"

    SSLContextVersion="SSLv3"
    certificateURL=""
    certificateURIPath="security/certificate/serverkeys"
    keyStore="JKS"
    keyStorePassword="defaultpwd"
    keyManagerFactory="SunX509"

    needClientAuthentication="true"
    trustAllCertificates="false"
    checkClientCertificateValidity="true"

    forwardClientCertificateChainDepth="1"
    forwardClientCertificateToEndpoint="true"
    forwardClientPemCertificateToEndpoint="true"

    trustCertificateURL=""
    trustCertificateURIPath="security/certificate/mycert.p12"
    trustKeyStore="PKCS12"
    trustKeyStorePassword="defaultpwd"
    trustKeyManagerFactory="SunX509"
    enable="true"/>
```

If a Web browser is used to try this listener, it is necessary to import the certificate into the browser. The certificate is included in the distribution in the directory:
(LBL_HOME)/security/certificate/mycert.p12

- hostname, NAT (Network Adapter Translation)

```
<bind listenType="NAT"
    address="lblservice" port="5050"
```

```
        enable="true"/>
```

- hostname, NAT (Network Adapter Translation), Virtual Domain

```
<bind listenType="NAT"
        address="lblservice" port="5050"
        enableVirtualDomain="true"
        enable="true"/>
```

- hostname, NAT (Network Adapter Translation), layer 4, EndPointsGrouoping

```
<bind listenType="NAT"
        address="lblservice" port="5151"
        osiLayer="4"
        protocol="ftp"
        endPointsGrouping="ftpserver"
        transport="tcp"
        transportSessionAffinity="false"
        enable="true"/>
```

- hostname, NAT (Network Adapter Translation) Layer 4, SSL, JVM keystore

```
<bind listenType="NAT"
        address="lblservice" port="1443"
        backlog="2048"
        SSL="true"
        certificateURL=""
        certificateURIPath="security/certificate/serverkeys"
        keyStore="JKS"
        keyStorePassword="defaultpwd"
        alias="lblcertsigned"
        aliasPassword="defaultpwd"
        keyManagerFactory="SunX509"
        SSLContextVersion="SSLv3"
        osiLayer="4"
        protocol="ftp"
        endPointsGrouping="ftpserver"
        transport="tcp"
        transportSessionAffinity="true"
        enable="true"/>
```

- interface, DAL (Dynamic Address Listen)

```
<bind listenType="DAL" monitorTimer="10000"
        netInterface="eth0" subInterface="0"
        port="5050" enable="true"/>
```

- example of Pure Forwarding

```
<bind listenType="STATIC"
 address="0.0.0.0" port="22,70-1000,8000-9000, 30, 60"
 portForwarding="true"
```

```
backlog="20"
osiLayer="4"
protocol="pure-forward"
endPointsGrouping="pure-forward-group"
transport="tcp"
enable="true"/>
```

- example of UDP listener

```
<bind enable="true"
        listenType="NAT"
        description="First FWD"
        address="lblservice"
        port="8888-8892,9999,1000-1001"
        osiLayer="4"
        protocol="udp"
        endPointsGrouping="udp-forward-group"
        transportSessionAffinity="true"
        transport="udp"

udpInterceptorClass="my_udprewriters.LBLUDPRewriteInterceptorLogging"/>
```

<endPointsGroupingParams>

```
<serviceconf>
        <iproxy>
                <endPointsGroupingParams>
```

Section for the management of TCP/UDP parameters of a particular group of services, or Protocol.

<endPointsGrouping>

```
<serviceconf>
        <iproxy>
                <endPointsGroupingParams>
                        <endPointsGrouping
```

endPointsGroupingName=: default value=””
The group name with which to associate the transmission parameters. If the group name begins with two pound signs (##) the reference is related to the Protocol and not with the name of the group. This feature enables creating parameters that can be utilized on multiple groups of the same protocol. It is possible to mention the same name multiple times and, in order of insertion, differences in the parameters will be applied. This is useful for changing e.g. the Protocol behavior included by default from OPLON® LoadBalancer at startup. It is sufficient to set the protocol name and indicate only the parameters to be modified. The other parameters remain unchanged.

templateName=: default value=””
The name of the group of parameters taken from a template on which to set the parameters for differences on the new group/Protocol. This group/Protocol is where are all the values are taken that are not present on the new group of endpoints. In other words if one wants to take as reference the parameters of the Protocol # #http and only change the "clientTimeOut", simply indicate in the endPointGrouping:

```
<endPointsGrouping templateName=”##http”
                   endPointsGroupingName=”myHttp”
                   clientTimeOut=”1000”/>
```

The parameters not mentioned will have the default value of "# #http".

clientTimeOut=: default value=”1500” UM=Thousandths of a second.
The TCP timeOut of the connection to the client

clientTimeOutFactor=: default value=”300”
The multiplicative factor of client timeOut

clientTimeOutContConnection=: default value=”1500” UM=Milliseconds
This parameter identifies the timeout between reading of valid HTTP headers during an HTTP 1.1 connection. This mode allows one to take advantage of the optimization of connections/disconnections of HTTP 1.1 and adapt to sometimes not entirely regular use by clients. A value of -1 disables the functionality.

clientTcpNoDelay=: default value=”true”
Enables/disables the Nagle algorithm to check for data buffering.

clientTcpKeepAlive=: default value=”true”
Enable/disable SO_KEEPALIVE in connections wih clients

endPointTimeOut=: default value=”1500” UM= Thousandths of a second.
The TCP timeOut of the connection toward the endPoints

endPointTimeOutFactor=: default value=”300”
The multiplicative factor of endPoints timeOut

endPointTimeOutContConnection=: default value=”-1” UM=Milliseconds
This parameter identifies the timeout between reading of valid HTTP headers during an HTTP 1.1 connection. This mode allows one to take advantage of the optimization of connections/disconnections of HTTP 1.1 and adapt to sometimes not entirely regular use by clients. A value of -1 disables the functionality.

endPointNumRetryConnection=: default value=”10”
Number of connection attempts before declaring an end point OutOfOrder.

endPointWaitPerRetryConnection=: default value=”350” UM=Milliseconds
Wait time for each connection attempt (endPointNumRetryConnection) before retrying the connection

endPointCreateConnectionTimeOut=: default value=”5000” UM=Millisec.
Time-out of the connection attempt to the end-point.

endPointTcpNoDelay=: default value=”true”
Enables/disables the Nagle algorithm to check for data buffering.

endPointTcpKeepAlive=: default value=”true”
Enable/disable SO_KEEPALIVE in connections with the endPoints.

sessionTimeOut=: default value=”1800000” UM=Milliseconds
If the load balancer is configured to handle the application sessions (stiky-sessions or loadBalancer managed sessions) this value indicates the session time-out in the sessions table. If the session in the table is not used for a period of time greater than this value, the routing references for the session are deleted.

sessionRefreshRateFactor=: default value=”5” UM=Minutes
If the load balancer is configured to handle the application sessions (stiky-sessions or loadBalancer managed sessions) this value indicates the value to calculate the fraction of the time of sessionTimeOut of the touch-session algorithm on the nodes.

The calculation to be carried out is as follows:
1800000/5 = 360000 (Refresh Rate in Milliseconds)

360000/1000 = 360 (Refresh Rate in seconds)
360/60 = 6 (Refresh Rate in minutes)

Therefore if the sessionTimeOut = 1800000 the refresh algorithm of the sessions in cache updates the modified sessions within 6 minutes. The larger the sessionRefreshRateFactor the smaller the window of verification and update of the session information on the nodes. The smaller the resulting value the higher the traffic on the private network.

sessionTouching=: default value=”true”
Setting the touch policy to false "sessionTouching = false" can be useful in cases where one wants to create a session linked to the TCP address in a defined time period and should not be renewed if a data exchange takes place. For example, if one wants to use the RDP Protocol and does not associate the server to the client but wants to maintain consistency during the negotiation phase.

sendHTTPResponseOnTimeout=: valore di default=”false” UM=boolean
Il true return a response 408 and 504 on service time-out.

enable=: default value=”true”
Enables/Disables the section <endPointsGrouping>.

Examples

Following are examples of <endPointsGroupingParams> and relative <endPointsGrouping>.

```
<endPointsGroupingParams>
  <endPointsGrouping endPointsGroupingName="myGroup"
                     clientTimeOut="3600000"
                     clientTimeOutFactor="100"
                     clientTcpNoDelay="true"
                     clientTcpKeepAlive="false"
                     endPointTimeOut="3600000"
                     endPointTimeOutFactor="300"
                     endPointTcpNoDelay="true"
                     endPointTcpKeepAlive="false"
                     sessionTimeOut="1800000"
                     sessionRefreshRateFactor="5"
                     sessionTouching="true"
                     enable="true"/>
  <endPointsGrouping endPointsGroupingName="myGroup001"
                     clientTimeOut="900"
                     clientTimeOutFactor="100"
                     clientTcpNoDelay="true"
                     clientTcpKeepAlive="false"
                     endPointTimeOut="400"
      ...
      ...
      ...
</endPointsGroupingParams>
```

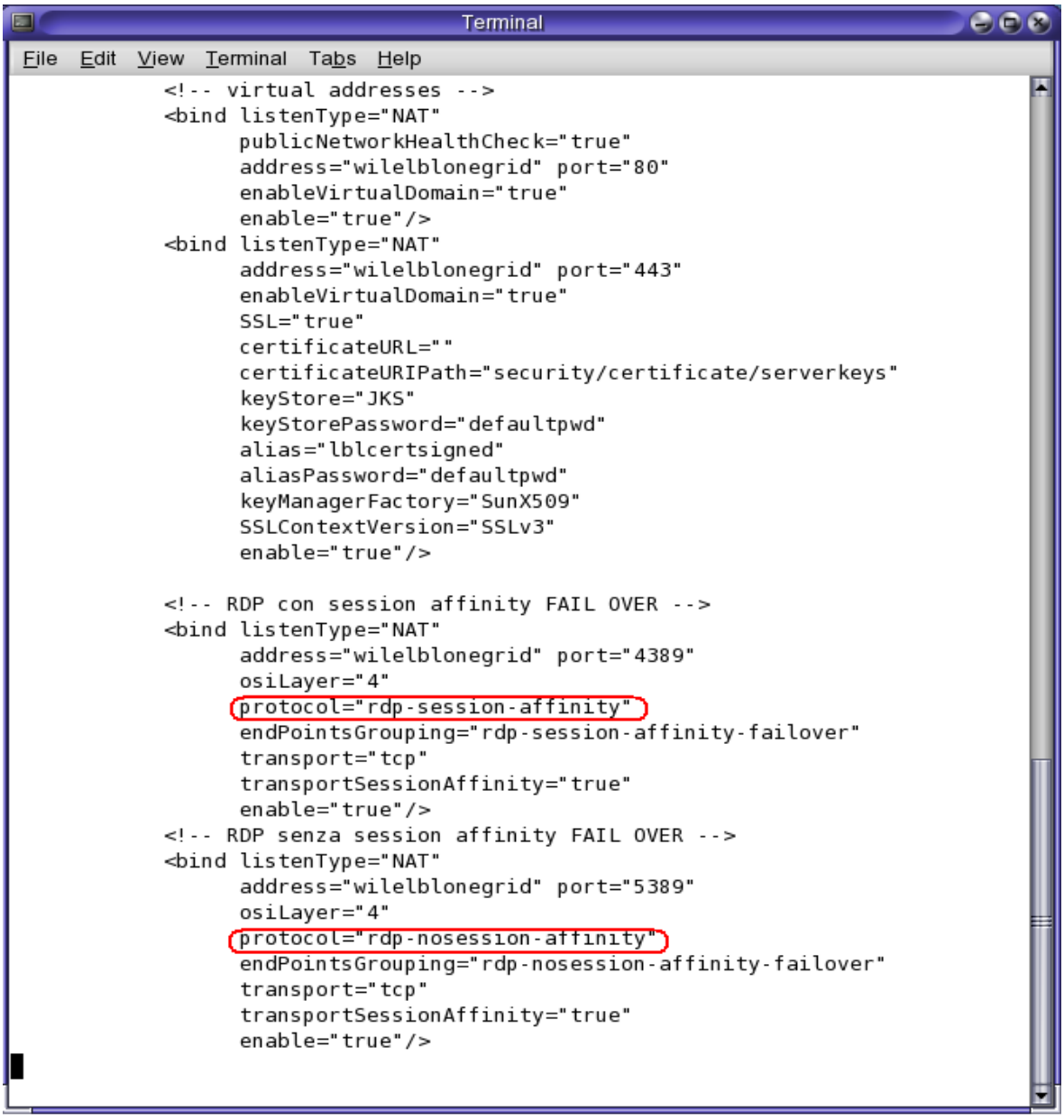
OPLON®Application Delivery Controller ha già preimpostati i parametri per i più diffusi protocolli:

- HTTP/S
- FTP (command and data)
- RDP without session management
- RDP with session management
- Telnet
- SSH
- ORACLE E ORACLE RAC
- Database generic
- UDP generic

To assign one of these protocols to a listener indicate in the <bind> section in the 'protocol' field one of the values listed in the table below:

Protocol	Value attribute <bind protocol="XXX")
HTTP/S	http
FTP command	ftpCMD
FTP data	ftpDATA
RDP with session management	rdp-session-affinity
RDP without session management	rdp-nosession-affinity
Telnet	telnet
SSH	ssh
ORAC Listener DB Oracle and Oracle RAC	ORAC
DBMS Connections to Database	DBMS
UDP generic	udp

Example of using the Protocol parameters section <bind>:



Http

```
<endPointsGroupingParams>
   <endPointsGrouping endPointsGroupingName="##http"
                      clientTimeOut="1500"
                      clientTimeOutFactor="300"
                      clientTcpNoDelay="true"
                      clientTcpKeepAlive="true"
                      endPointTimeOut="1500"
                      endPointTimeOutFactor="300"
                      endPointTcpNoDelay="true"
```

```
                    endPointTcpKeepAlive="true"
                    sessionTimeOut="1800000"
                    sessionRefreshRateFactor="5"
                    sessionTouching="true"
                    endPointNumRetryConnection="10"
                    endPointWaitPerRetryConnection="350"
            endPointCreateConnectionTimeOut="5000"
                    enable="true"/>
    ...
</endPointsGroupingParams>
```

FTPcmd

```
<endPointsGroupingParams>
  <endPointsGrouping endPointsGroupingName="##ftpCMD"
                    clientTimeOut="600000"
                    clientTimeOutFactor="1"
                    clientTcpNoDelay="true"
                    clientTcpKeepAlive="true"
                    endPointTimeOut="660000"
                    endPointTimeOutFactor="1"
                    endPointTcpNoDelay="true"
                    endPointTcpKeepAlive="true"
                    sessionTimeOut="1800000"
                    sessionRefreshRateFactor="5"
                    sessionTouching="true"
                    endPointNumRetryConnection="10"
                    endPointWaitPerRetryConnection="350"
        endPointCreateConnectionTimeOut="5000"
                    enable="true"/>
    ...
</endPointsGroupingParams>
```

FTPData

```
<endPointsGroupingParams>
  <endPointsGrouping endPointsGroupingName="##ftpDATA"
                    clientTimeOut="600000"
                    clientTimeOutFactor="1"
                    clientTcpNoDelay="true"
                    clientTcpKeepAlive="true"
                    endPointTimeOut="660000"
                    endPointTimeOutFactor="1"
                    endPointTcpNoDelay="true"
                    endPointTcpKeepAlive="true"
                    sessionTimeOut="1800000"
                    sessionRefreshRateFactor="5"
                    sessionTouching="true"
                    endPointNumRetryConnection="10"
```

```
                    endPointWaitPerRetryConnection="350"
        endPointCreateConnectionTimeOut="5000"
                    enable="true"/>
      ...
      ...
      ...
</endPointsGroupingParams>
```

rdp-session-affinity

```
<endPointsGroupingParams>
  <endPointsGrouping endPointsGroupingName="##rdp-session-affinity"
                    clientTimeOut="10800000"
                    clientTimeOutFactor="1"
                    clientTcpNoDelay="true"
                    clientTcpKeepAlive="true"
                    endPointTimeOut="10800000"
                    endPointTimeOutFactor="1"
                    endPointTcpNoDelay="true"
                    endPointTcpKeepAlive="true"
                    sessionTimeOut="1800000"
                    sessionRefreshRateFactor="5"
                    sessionTouching="true"
                    endPointNumRetryConnection="10"
                    endPointWaitPerRetryConnection="350"
        endPointCreateConnectionTimeOut="5000"
                    enable="true"/>
      ...
      ...
      ...
</endPointsGroupingParams>
```

rdp-nosession-affinity

```
<endPointsGroupingParams>
  <endPointsGrouping endPointsGroupingName="##rdp-nosession-affinity"
                    clientTimeOut="10800000"
                    clientTimeOutFactor="1"
                    clientTcpNoDelay="true"
                    clientTcpKeepAlive="true"
                    endPointTimeOut="10800000"
                    endPointTimeOutFactor="3"
                    endPointTcpNoDelay="true"
                    endPointTcpKeepAlive="true"
                    sessionTimeOut="10000"
                    sessionRefreshRateFactor="5"
                    sessionTouching="false"
                    endPointNumRetryConnection="10"
                    endPointWaitPerRetryConnection="350"
        endPointCreateConnectionTimeOut="5000"
```

```
                    enable="true"/>
        ...
        ...
        ...
</endPointsGroupingParams>
```

telnet

```
<endPointsGroupingParams>
  <endPointsGrouping endPointsGroupingName="##telnet"
                    clientTimeOut="10800000"
                    clientTimeOutFactor="1"
                    clientTcpNoDelay="true"
                    clientTcpKeepAlive="true"
                    endPointTimeOut="10800000"
                    endPointTimeOutFactor="1"
                    endPointTcpNoDelay="true"
                    endPointTcpKeepAlive="true"
                    sessionTimeOut="3600000"
                    sessionRefreshRateFactor="5"
                    sessionTouching="true"
                    endPointNumRetryConnection="10"
                   endPointWaitPerRetryConnection="350"
        endPointCreateConnectionTimeOut="5000"
                    enable="true"/>
        ...
        ...
        ...
</endPointsGroupingParams>
```

ssh

```
<endPointsGroupingParams>
  <endPointsGrouping endPointsGroupingName="##ssh"
                    clientTimeOut="10800000"
                    clientTimeOutFactor="1"
                    clientTcpNoDelay="true"
                    clientTcpKeepAlive="true"
                    endPointTimeOut="10800000"
                    endPointTimeOutFactor="1"
                    endPointTcpNoDelay="true"
                    endPointTcpKeepAlive="true"
                    sessionTimeOut="3600000"
                    sessionRefreshRateFactor="5"
                    sessionTouching="true"
                    endPointNumRetryConnection="10"
                    endPointWaitPerRetryConnection="350"
        endPointCreateConnectionTimeOut="5000"
                    enable="true"/>
```

```
        ...
        ...
        ...
</endPointsGroupingParams>
```

Listener Oracle/Oracle RAC

```
<endPointsGroupingParams>
  <endPointsGrouping endPointsGroupingName="##ORAC"
                     clientTimeOut="432000000"
                     clientTimeOutFactor="1"
                     clientTcpNoDelay="true"
                     clientTcpKeepAlive="true"
                     endPointTimeOut="432000000"
                     endPointTimeOutFactor="1"
                     endPointTcpNoDelay="true"
                     endPointTcpKeepAlive="true"
                     sessionTimeOut="3600000"
                     sessionRefreshRateFactor="5"
                     sessionTouching="false"
                     endPointNumRetryConnection="10"
                     endPointWaitPerRetryConnection="350"
                endPointCreateConnectionTimeOut="5000"
                     enable="true"/>
        ...
        ...
        ...
</endPointsGroupingParams>
```

Generic Database

```
<endPointsGroupingParams>
  <endPointsGrouping endPointsGroupingName="##DBMS"
                     clientTimeOut="432000000"
                     clientTimeOutFactor="1"
                     clientTcpNoDelay="true"
                     clientTcpKeepAlive="true"
                     endPointTimeOut="432000000"
                     endPointTimeOutFactor="1"
                     endPointTcpNoDelay="true"
                     endPointTcpKeepAlive="true"
                     sessionTimeOut="3600000"
                     sessionRefreshRateFactor="5"
                     sessionTouching="false"
                     endPointNumRetryConnection="10"
                     endPointWaitPerRetryConnection="350"
                endPointCreateConnectionTimeOut="5000"
                     enable="true"/>
        ...
        ...
```

```
        ...
</endPointsGroupingParams>
```

UDP generic with session affinity

```
<endPointsGroupingParams>
  <endPointsGrouping endPointsGroupingName="##udp"                    clientTimeOut="10800000"
                     clientTimeOutFactor="1"
                     clientTcpNoDelay="true"
                     clientTcpKeepAlive="false"
                     endPointTimeOut="10800000"
                     endPointTimeOutFactor="1"
                     endPointTcpNoDelay="true"
                     endPointTcpKeepAlive="true"
                     sessionTimeOut="1800000"
                     sessionRefreshRateFactor="5"
                     sessionTouching="true"
                     enable="true"/>...
        ...
        ...
</endPointsGroupingParams>
```

Listeners pure forwarding

```
<endPointsGroupingParams>
  <endPointsGrouping endPointsGroupingName="##pure-forward"
        clientTimeOut="3600000"
                     clientTimeOutFactor="100"
                     clientTcpNoDelay="true"
                     clientTcpKeepAlive="false"
                     endPointTimeOut="3600010"
                     endPointTimeOutFactor="100"
                     endPointTcpNoDelay="true"
                     endPointTcpKeepAlive="false"
                     endPointNumRetryConnection="0"
                     endPointWaitPerRetryConnection="1"
        endPointCreateConnectionTimeOut="100"
                     enable="true"/>          ...
        ...
        ...
</endPointsGroupingParams>
```

<params>

```
<serviceconf>
        <iproxy>
                <params
```

This section contains the General parameters of communications management.

NOTE Some of the parameters of this section are not taken into account if there are <endPointsGroupingParams> sections present. In such cases the values in the respective <endPointsGrouping> are valid instead of the section <params>.

---

cipherSuites=: Default value=""
deprecated: use cipherSuitesListeners instead
parameter sets the ciphersuites used for the communication with the backend.

SSLProtocols=: Default value=""
deprecated: use cipherSuitesListeners instead
parameter sets the SSL to use for communication with the backend.

cipherSuitesListeners=: default=””
General SSL encryption suite used by listeners

SSLProtocolsListeners=: default=””
General SSL protocols used by listeners

cipherSuitesEndpoints=: default=””
Cipher general SSL suites used by endpoints

SSLProtocolsEndpoints=: default=””
General SSL protocols used by listeners

SSLAlertExpirationDays=: default=”30”
Number of days in advance of the expiry of the certificates to notify the imminent deadline every 24 hours

analyzeHeaderAttackPrevention=: default=”false”
Internal use only

userAgentShortCircuit=: default=””
If present, overwrites the default ADC user agent in HTTP responses that do not end on an endpoint.

rewriteHeaderRules=: default=””
Rules for rewrite HTTP headers applied on all resources, you can exclude these rules selectively on the resources involved with the modifier NAME_RULE: NOP

rewriteBodyRules=: default=””

Rules of rewrite body HTTP applied on all resources, you can exclude these rules selectively on the resources involved with the modifier NAME_RULE: NOP

endPointsHealthCheckSetOutOfOrder=: default=”false”
If true forces the health chech of the endpoints

clientH2Bridge=: default=”false”
If true enables the HTTP2 bridge with HTTP2 client connections and HTTP server side

clientSSLUseCipherSuitesOrder=: default=”true”
Se true utilizza i chipersuite nell'ordine indicato per i listener SSL/TLS

clientSSLApplicationProtocols=: default=”” values:”h2 http/1.1 undef”
If true, use the chipersuite in the order specified for the SSL / TLS listeners

```
eg: <params
                    ...
                    endPointSSLApplicationProtocols="h2 http/1.1 undef"
                    endPointSSLUseCipherSuitesOrder="true"
                    .../>

<endPointsGrouping enable="true" SSLApplicationProtocols="h2 http/1.1 undef"
SSLUseCipherSuitesOrder="true">
        <virtualDomain enable="true" SSLApplicationProtocols="h2 http/1.1 undef"
SSLUseCipherSuitesOrder="true">
        <endp address="192.168.56.131" port="8080" uriPath="/" SSLApplicationProtocols="h2 http/1.1
undef"
          SSLUseCipherSuitesOrder="true" enable="true"/>
```

endPointSSLUseCipherSuitesOrder=: default=”true”
If true, use the chipersuite in the order indicated for the SSL / TLS endpoints

endPointSSLApplicationProtocols=: default=”” values:”h2 http/1.1 undef”
This parameter can list the protocols enabled on ALPN communications

```
eg: <params
                    ...
                    endPointSSLApplicationProtocols="h2 http/1.1 undef"
                    endPointSSLUseCipherSuitesOrder="true"
                    .../>

<endPointsGrouping enable="true" SSLApplicationProtocols="h2 http/1.1 undef"
SSLUseCipherSuitesOrder="true">
        <virtualDomain enable="true" SSLApplicationProtocols="h2 http/1.1 undef"
SSLUseCipherSuitesOrder="true">
        <endp address="192.168.56.131" port="8080" uriPath="/" SSLApplicationProtocols="h2 http/1.1
undef"
          SSLUseCipherSuitesOrder="true" enable="true"/>
```

monitorTimer=: default value=”10000” UM=Milliseconds
The value in milliseconds of monitoring of the load balancing system.

uniqueContextID=: default value=”NoVRRPHost”
Value used in installations of OPLON® LoadBalancer Platform Edition to create unique traffic values recorded in the database. The parameter must be set manually in case multiple

instances are in the same operating system. The resulting value in the database, to distinguish the origins, is as follows:
uniqueContextID+local host address+ hash(uniqueContextID+local host address)

eg. default: NoVRRPHost-192.168.41.169-1983464450

eg. with default value variation:

```
<params
        uniqueContextID="001"
        initialDim="68000"
          redimFactor="1"
          flushFactor="65535"
          maxDim="68000"
        concurrentSessions="500"
        maxConcurrentSessions="500">
</params>
```

result: 001-192.168.41.169-1983464450

concurrentSessions=: default value=”50”
The number of initial simultaneous serviceable connections.

maxConcurrentSessions =: default = "200"
The maximum number of simultaneous serviceable connections.
The system implements the Survivor Tunnel feature to preserve resources in case an endpoints-group exceeds usage. The system reserves 10% of the tunnels, with up to 500 tunnels, under these conditions:

A) The maximum number of tunnels is greater than 99;
B) The number of endpoints-gouping is greater than 1.

NOTE: Endpoints-grouping refers to the set of: group, domain, uripath.

In an environment where there are multiple endpoints-groups, if a service exceeds 90% of the tunnels resources, requests for that specific endpoint-group will no longer be accepted in order to have free tunnels for other services.

listenersPriority =: default = "5"
Priority of listeners (producers in queue Manager connection requests).

protocolsResolverPriority=: default value=”5”
Priority of Protocol resolvers (consumers in queue Manager connection requests).

expandConcurrentSessionsFactor=: default value=”10.0” UM=%
Growth factor with which new sessions are instantiated until "maxConcurrentSessions" is reached.

survivorTunnelsPercent=: default value =”30.0” UM=%
It is the threshold percentage of tunnels occupied by the same “group / domain / uripath” service. When this threshold is exceeded, all IP addresses within the tunnels belonging to the same "group / domain / uripath" are quarantined. Associated with the "dosAttackPreventionWashingMachine" parameter, it allows you to free up resources for other services even with violent DoS / DDoS attacks. **Values less than or equal to 0 disable the feature**.

The WasingMachine is enabled to manage the address only if
isDosAttackPreventionOnlyClose is true

Both the WashingMachine and the SurvirvistTunnelsPercent are activated in the presence of the DoS/DDoS license and if **isdosAttackPreventionDetection** is set to false.

Conditions for activating the washing machine:
dosAttackPreventionWashingMachine = dosAttackPrevention=true &&
!dosAttackPreventionDetection=false

Activation conditions survivalTunnelsPercent:
survivalTunnelsPercent = dosAttackPrevention=true &&
!dosAttackPreventionDetection &&
dosAttackPreventionWashingMachine=true

highWaterWarningLevel=: default value=”10.0” UM=%
The threshold percentage of connections waiting to be taken over by a forwarder. If this threshold is exceeded, the system warns with a specific message (yellow-alert) that the limit has been exceeded. If the limit is exceeded new forwarding sessions are instantiated until the maxConcurrentSessions " is reached.

highWaterDangerLevel=: default value=”40.0” UM=%
The threshold percentage of connections waiting to be taken over by a forwarding session. If this threshold is exceeded, the system warns with a specific message (red-alert) that the limit has been exceeded. If the limit is exceeded new forwarding sessions are instantiated until the maxConcurrentSessions " is reached. With the OPLON® DoS Attack Prevention function, if this limit is reached the system performs DDoS Congestion Resolver routines ©.

dosAttackPrevention=: default value=”false” boolean
If set to true, the rules of DoS (Denial of Service) attack prevention are applied.
dosAttackPrevention intervenes in different ways depending on the type of attack:

- Multiple requests derviving from the same IP
- Multiple requests deriving from different IP addresses

In the first case, once the defined number of simultaneous requests is reached, which can be set with the parameter "dosMaxTunnelForClientAddress", the connections are deleted including any pending internal queues requests.
In the second case, once the the highWaterDangerLevel threshold is exceeded all requests in progress and pending/in queues are deleted.

In both cases before the cancellation of pending requests, if HTTP/S connections, a courtesy page to the client that made the request will be sent.

NOTE. Enabling this feature is subject to the presence of a DoS Attack Prevention specific license.

dosAttackPreventionDetection=: default=”false” boolean
If true, when a DoS / DDoS attack is detected, it is reported but not blocked. It is very useful when activating the functionality to determine the addresses to be whitelisted.

dosAttackPreventionOnlyClose=: default value=”true”
When a DoS attack is detected, the default action is to close the channels that are identified as a threat. For some protocols, it is possile to provide a temporary failure courtesy page. This notification may also be used to improve the attack and so it is disabled by default.

dosAttackPreventionWashingMachine =: default value = ”false” boolean
If set to true all quarantined addresses are forwarded to a separate queue from non-quarantined addresses and a TCP reset (RST) is performed.

The WasingMachine is enabled to manage the address only if
isDosAttackPreventionOnlyClose is true

Both the WashingMachine and the SurvirvistTunnelsPercent are activated in the presence of the DoS/DDoS license and if **isdosAttackPreventionDetection** is set to false.

Conditions for activating the washing machine:
dosAttackPreventionWashingMachine = dosAttackPrevention=true &&
!dosAttackPreventionDetection=false

Activation conditions survivalTunnelsPercent:
survivalTunnelsPercent = dosAttackPrevention=true &&
!dosAttackPreventionDetection &&
dosAttackPreventionWashingMachine=true

dosDDoSPutEndPointsOutOfOrder=: default=”false” boolean
During a red alert event, if true DDoS attack prevention places the endpoints at Outoforder.

dosMaxTunnelForClientAddress=: default value=”100” UM=requests in the tunnel
Indicates the number of simultaneous requests served to a single IP address. If this threshold is exceeded all connections established from that same IP will be instantly deleted. If there are HTTP/S connections, all connections waiting in the queue will be repied to with the courtesy message (dosCMessage) or redirected based on the parameter (dosRedirect).

dosMaxTunnelForClientSubnet_255_255_255_0=: default value=”255” UM=rrequests in the tunnel
indicates the number of simultaneous requests served to at the same subnet (C). If this threshold is exceeded all connections established from that same subnet will be instantly

deleted. If there are HTTP/S connections, all connections waiting in the queue will be repied to with the courtesy message (dosCMessage) or redirected based on the parameter (dosRedirect).

dosMaxTunnelForClientSubnet_255_255_0_0=: default value=”500” UM=requests in the tunnel
indicates the number of simultaneous requests served to at the same subnet (B). If this threshold is exceeded all connections established from that same subnet will be instantly deleted. If there are HTTP/S connections, all connections waiting in the queue will be repied to with the courtesy message (dosCMessage) or redirected based on the parameter (dosRedirect).

dosMaxTunnelForClientSubnet_255_0_0_0=: default value=”2000” UM=requests in the tunnel
indicates the number of simultaneous requests served to at the same subnet (A). If this threshold is exceeded all connections established from that same subnet will be instantly deleted. If there are HTTP/S connections, all connections waiting in the queue will be repied to with the courtesy message (dosCMessage) or redirected based on the parameter (dosRedirect).

dosAddressQuarantineTime=: default value=”1800000” UM=milliseconds
Upon identification of an attack coming from a single ip address it is automatically placed in quarantine preventing access to services. The quarantine period is determined by this value (30 '). Once this time has passed, the client is again given the opportunity to access services. If this value is set to 0 or less han 0, the quarantine functionality is disabled.

dosCMessage=: default value=”messageServicesOverload.html”
Indicates the page of courtesy to expose in the case of activation of the DoS Attack Prevention. If you want to edit the page of courtesy is sufficient to modify the file: (LBL_HOME)/resources/html/messageServicesOverload.html or place for single process as for example :(LBL_HOME)/procsProfiles/A10_LBLGoStandardHA/resources/html. Once you have located the file you can also modify it during the runtime. This parameter is taken into consideration if dosAttackPreventionOnlyClose is set to false.

dosCaptchaCMessage=: default value=”messageServicesCaptchaOverload.html”
NOTE: THIS VALUE IS OBSOLETE, NOT USED IN THIS RELEASE
Indicates the page of courtesy to expose in the case of activation of the DoS Attack Prevention with confirmation of the Captcha Input to check the use by a person and not from an automaton. If you want to edit the page of courtesy is sufficient to modify the file: (LBL_HOME)/resources/html/messageServicesCaptchaOverload.html or place for single process as for example
:(LBL_HOME)/procsProfiles/A10_LBLGoStandardHA/resources/html. Once you have located the file you can also modify it during the runtime. This parameter is taken into consideration if dosAttackPreventionOnlyClose is set to false.

dosCaptchaGoCMessage=: default=”messageServicesCaptchaGoOverload.html”
Note: NOT USED IN THIS RELEASE
indicates the page of courtesy to expose after setting the captcha correctly in case of

activation of the DoS Attack Prevention. If you want to edit the page of courtesy is sufficient to modify the file: (LBL_HOME)/resources/html/ messageServicesCaptchaGoOverload.html or place for single process as for example:(LBL_HOME)/procsProfiles/A10_LBLGoStandardHA/resources/html. Once you have located the file you can also modify it during the runtime. This parameter is taken into consideration if dosAttackPreventionOnlyClose is set to false.

dosRedirect=: default value=””
If a value is entered, indicates the URI to redirect the request in the case that DoS Attack Prevention is active. eg.: http://www.caughtinfo.com/.
If this value is set, it has priority over the dosCMessage parameter. This parameter is taken into account if dosAttackPreventionOnlyClose is set to false.

rewriteHeaderRules =: default value=""
List of names of the rewriting rules for the HTTP HEADERS (layer 7 HTTP/S) to apply to all the endpoints. It is possible to indicate multiple rules separated by one or more spaces.

Ex.:

```
rewriteHeaderRules="redirSSLloginWhenNoSSL proxyTo"
```

The rules will be applied, if conditions permit, in sequence.

;**LAST**
For each rule name one can also indicate the parameter 'LAST' that in case the rule is performed determines stopping the application of the remaining rules of the sequence.
To set the value of the parameter the " ;" after the name and the name of the parameter.

eg.

```
rewriteHeaderRules="redirSSLloginWhenNoSSL;LAST proxyTo"
```

There must not be any spaces between the name and the parameters. In this case, if the rule redirSSLloginWhenNoSSL is applied the rule proxyTo will never run.

;ALWAYS
The parameter ALWAYS indicates that the rule is always performed to indicate the value of the parameter is sufficient to put after the name the " ;" and the name of the parameter.

eg.

```
rewriteHeaderRules="redirSSLloginWhenNoSSL;LAST proxyTo;ALWAYS"
```

In this case the rule proxyto is performed regardless of the execution of the rule redirssloginwhennossl.

;NOP
The NOP parameter indicates that the rule should not be performed.

eg.

```
rewriteHeaderRules="redirSSLloginWhenNoSSL;NOP proxyTo;ALWAYS"
```

In this case the rule redirssllloginwhennossl is not performed. The parameter NOP is useful to exclude the execution of general rules.

;FINAL
;ALWAYS-FINAL
;LAST_FINAL
;NOP-FINAL
Operations with FINAL name or extension have the same functionality as existing ones with the difference that they are performed at the end. In practice, even if described on a global level, the ADC, GROUP DOMAIN or ENDPOINT are executed, in the order indicated, at the end of all the other rules described on the underlying levels.

Rule setting sequence:
ADC (global)
GROUP
DOMAIN
ENDPOINT

An example of chaining rules:
AAAA**;ALWAYS-FINAL** BBBB CCCC DDDD**;FINAL** EEEE

The order of execution will be:
BBBB CCCC EEEE AAAA;ALWAYS DDDD

rewriteBodyRules =: default value="
List of names of the rewriting rules for the HTTP BODY (layer 7 HTTP/S) to apply to all the endpoints. It is possible to indicate multiple rules separated by one or more spaces.

Ex.
```
rewriteBodyRules="addTrademarkParam absoluteToRelative echoRewriteBody"
```

The rules will be applied, if conditions permit, in sequence.

;**LAST**
For each rule name one can also indicate the parameter 'LAST' that in case the rule is performed determines stopping the application of the remaining rules of the sequence.
To set the value of the parameter place a " ;" (semi-colon) after the name and the name of the parameter.

Ex.
```
rewriteBodyRules="addTrademarkParam;LAST absoluteToRelative echoRewriteBody"
```

There must not be any spaces between the name and the parameters.. In this case, if the rule addTrademarkParam is applied the rules absoluteToRelative echoRewriteBody will never be performed.

;ALWAYS
parameter ALWAYS indicates that the rule is always performed.
```
rewriteBodyRules="addTrademarkParam;LAST absoluteToRelative echoRewriteBody;ALWAYS"
```

In this case the rule echorewritebody is performed regardless of the execution of the rule addtrademarkparam.

;NOP

The NOP parameter indicates that the rule should not be performed.

```
rewriteBodyRules="addTrademarkParam;LAST absoluteToRelative;NOP echoRewriteBody;ALWAYS"
```

In this case the rule absolutetorelative is not performed. The parameter NOP is useful to exclude the execution of general rules.

;FINAL
;ALWAYS-FINAL
;LAST_FINAL
;NOP-FINAL

Operations with FINAL name or extension have the same functionality as existing ones with the difference that they are performed at the end. In practice, even if described on a global level, the ADC, GROUP DOMAIN or ENDPOINT are executed, in the order indicated, at the end of all the other rules described on the underlying levels.

Rule setting sequence:
ADC (global)
GROUP
DOMAIN
ENDPOINT

An example of chaining rules:
AAAA**;ALWAYS-FINAL** BBBB CCCC DDDD**;FINAL** EEEE

The order of execution will be:
BBBB CCCC EEEE AAAA;ALWAYS DDDD

clientDefaultTimeOut=: default value=”1500” UM=Thousandths of a second
TCP timeOut for the connection toward the client.

clientDefaultTimeOutFactor=: default value=”300”
The multiplicative factor of client timeOut.

clientDefaultTcpNoDelay=: default value=”true”
Enables/disables the Nagle algorithm to check for data buffering.

clientReceiveBufferSize=: default value=”-1”
The size of the socket buffer receiving from the client.

clientSendBufferSize=: default value=”-1”
The size of the socket buffer sending to the client.

endPointDefaultTimeOut=: default value=”1500” UM= Thousandths of a second
The TCP timeOut of the connection toward the endPoints.

endPointDefaultTimeOutFactor=: default value=”300”
The multiplicative factor of the endPoints timeOut

endPointDefaultTcpNoDelay=: default value=”true”
Enables/disables the Nagle algorithm to check for data buffering

endPointReceiveBufferSize=: default value=”-1”
The size of the socket buffer receiving from the endPoint

endPointSendBufferSize=: default value=”-1”
The size of the socket buffer sending to the endPoint.

healthCheckContextPath=: default value=”/LBLHealthCheck”
The path of the healthcheck of the operations of the balancing system. This value normally is never modified unless already used in other applications. If this value is changed, it must also be changed in "systemsmonitor_m.xml", "iproxy.xml" and "healthcheck.xml".

defaultCType=: default value=”text/html”
Value entered in the header from the endPoint if mime type (content type) correction is enabled.

clientCorrectionCType=: default value=”false”
Enables correction of the mime type (content type) in the header from the endPoint during delivery to the client.

clientCType=: default value=”Content-Type: +defaultCType”
The value to assign to the header from the endPoint during delivery to the client in case of correct mime type (Content type)

endPointCorrectionCType=: default value=”false”
Enables correction of the mime type (content type) in the client header during delivery to the endpoint.

maxPureStreamLength=: default value=”99999999” UM=bytes
The maximum length of a stream of the HTTP body in the absence of an explicit statement of the length. This behavior is typical of early HTTP 1.0 implementations and is often associated with the non-evaluation of the HTTP "connection: keep-alive" entity. For this purpose the evaluation of this parameter in OPLON® LoadBalancer is set to "false" by default (see keepAliveEvaluationHTTP10 parameter).

keepAliveEvaluationHTTP10=: default value=”false”
This parameter enables and disables the evaluation of the entity HTTP "connection: keep-alive". In the early HTTP implementations 1.0 this parameter was not evaluated and even today the majority of HTTP 1.0 application servers or proxy server do not evaluate this

parameter. For this purpose also OPLON® LoadBalancer adapts to this behavior by setting default to false.

endPointCType=: default value=”Content-Type:+defaultCType”
The value to assign to the client header during delivery to the endpoint in the case of correction of mime type (Content type).

monitorTimerStatSessions=: default value=”120000” UM=Milliseconds
The value in milliseconds and determines the temporal granularity of information logged in the database.

monitorTimerActivePoolQueueConsumer=: default value=”100” UM=mill
Milliseconds to wait for each cycle of the monitoring of the iProxy active pool queue consumer system.

monitorTimerActiveRoutingSessions=: default value=”8000” UM=mill
Milliseconds to wait for each cycle of the monitoring of the iProxy active routing sessions.

fileNameForceIncomingConnectionToWait=:
default value=”forceIncomingConnectionToWait” UM=mill
File name for wait notification of incoming connections to freeze.

monitorIncludeAddressStatSessions=: default value=”false”
Value instructing the statistical engine to include the client's address information in the session information. When the value is set to true, under some circumstances with large numbers of users, can lead to a large number of records in the SESSION_ACTIVITY table.

notificationDir=: default value=”lib/notificationDir”
Indicates the directory where to notify OPLON® LoadBalancer of an outOfOrder of a service or a group of backend services. It is possible to control an out-of-order of an end-point or a grouping of end-points from outside. In this regard in iproxy.xml it is possible to assign the associative names to groups/domains up to the single end-point. If files are created in the new directory "notificationDir" with the prefix "outOfOrder" followed by a symbolic name ("outOfOrder.symbolicName") the end-points, domain, or group of end-points will be out-of-order until the deletion of the file. (LBL_HOME)/lib/plugin contains 2 sample sources of health check and externally controlled disabling/enabling of backend services.

udpWaitTimeSessionInitialization=: default value=”3000”
The maximum wait time for any datagrams without association. The value is set by default to 3000 milliseconds (3 seconds).

monitorTimerStatUdpSessions=: default value=”10000” UM=milliseconds
Milliseconds to wait for each cycle of monitoring of the sessions list for UDP synchronization.

cMessage=: default value=”messageNoEndPoint.html”
Name of the html file containing the courtesy message in the context to which it is associated. At load time of parameters during the start of the balancing process, verifies in

sequence the file existence in the directories :

- (LBL_HOME)/procsProfiles/(MODULE_NAME)/resources/html/messageName.html
- (LBL_HOME)/resources/html/messageName.html

If the file does not exist, it is reported at boot time and assigns the default value:

- (LBL_HOME)/resources/html/messageNoEndPoint.html

initialDim=: default value=”68000” UM=Byte
Initial traffic forwarding buffer size.

redimFactor=: default value=”1”
The adaptive factor for taking the forwarding buffer to the maximum size indicated by the parameter "maxDim”.

flushFactor =: default value = "65535"
Buffered traffic flushing factor.

maxDim=: default value=”68000”
The maximum size of the forwarding buffer.

The four values initialDim, redimFactor, flushFactor, and maxDim are required to calculate the dimension of the IO buffers and to optimize the TCP Window size.

The size of the IO buffers is important because it determines the the use of the band and therefore throughput.

The default values are set for minimum expenditure of resources.

OPLON® LoadBalancer initializes 4 buffers per thread: 2 for streaming from the client to the services (endpoint); 2 for return values. Buffers use COW (Copy On Write) technology for modifications to the passing values.

InitialDim determines the initial size of the buffers, maxDim determines the maximum size of the buffer. Normally the stream is buffered and, upon reaching the threshold of flushing, flushFactor, the stream is forwarded. This mechanism is not valid during the reading of the HTTP HEADER. The HTTP HEADER for routing requirements must be read and fully buffered to be analyzed and interpreted.

This means that HTTP HEADERS cannot be larger, in bytes, than the parameter maxDim. In case the client or the endpoint is using large HEADERS, one can change the maxDim bearing in mind that, reading the HEADER doesn't happen in blocks, and thus an application that makes heavy use of this – in terms of volume - inevitably would take more time than other similar applications with HEADERS that are smaller.

In the event that a service or client exceed this value at runtime OPLON® LoadBalancer would react by interrupting the connection and generating an error message with "Buffer grows too much!" with an indication of where the limits were surpassed.

In cases where it is not possible to apply the flushing, for example in reading HTTP

HEADER, when initialDim reaches its limit it is expanded through the redimFactor parameter. This parameter expands the IO buffer in blocks up to maxDim. When maxDim is also exceeded, the connection is cosed and the “Buffer Grows too much!” error message is generated.

From the above descriptors it follows that the flushing factor, flushFactor, must be less than initialDim to be effective.

Based on tests carried out on different operating systems the values that guarantee excellent throughput/employment of resources during production are:

initialDim="68000"
redimFactor="1"
flushFactor="65535"
maxDim="68000"

In this case, considerable improvement in scalability and reliability is achieved over time by running OPLON® LoadBalancer in finite capacities predetermined when the service starts - this diminishes the chances of out of memory errors due to the enlargement of the dynamic buffers at run time.

On equal allocation of buffers by raising the flushFactor, improvement in throghput and a reduction in CPU utilization is achieved. Lowering the flushFactor, even up to 1 byte at a time, bandwidth usage is decreased and the CPU usage is increased as it is committed to process information blocks with smaller dimensions.

Taking as a reference a maxDim set to 68000, the calculation of the maximum memory occupied by IO buffers is as follows:

- maxConcurrentSessions * maxDim * 4 = MAX MEMORY FOR IO BUFFERS
eg:
- 5000 threads * 68000 * 4 = 1.360,00 MB (1,3GB)

With a 2GB JVM, about 500 MB remain available to maintain the routing tables. About 800,000 sticky sessions.

udpMaxDim=: default value=”65535” UM=Byte
Maximum size that can take the datagram. This parameter is important for memory sizing. Buffers are allocated only if a listener of type "udp" is used.

If there is at least one UDP listener the following will be prealloacted:
- maxConcurrentSessions*2*udpMaxDim bytes.

For 2000 Threads the calculation results are achieved as follows:
- 2000*2*65535=262.140.000 (250 MB)

File outOfOrder

Through the outOfOrder file a group of endpoints/services can be declared out of order.

```
<endpoints>
 <endPointsGrouping associateName="pippo pluto " enable="true">
   <virtualDomain associateName="paperino minni " portRewriting="true" enable="true"/>
     <endp address="wiletrbackend" port="8080" uriPath="/Flowers/album" enable="true"/>
     <endp address="roadtrbackend" port="8080" uriPath="/Flowers/album" enable="true"/>
     <endp address="wiletrbackend" port="8787" uriPath="/Flowers/album" enable="true"/>
     <endp address="roadtrbackend" port="8787" uriPath="/Flowers/album" enable="true"/>
     <endp address="wiletrbackend" port="8787" uriPath="/TCOProject"
                 associateName=" qui   quo    qua   " enable="true"/>
     <endp address="roadtrbackend" port="8787" uriPath="/TCOProject"
                 associateName=" qui   quo    qua   " enable="true"/>
     <endp address="wiletrbackend" port="8787" uriPath="/TCOProjectSrv" enable="true"/>
     <endp address="roadtrbackend" port="8787" uriPath="/TCOProjectSrv" enable="true"/>
     <endp address="wiletrbackend" port="8181" uriPath="/training" enable="true"/>
     <endp address="roadtrbackend" port="8181" uriPath="/training" enable="true"/>
     <endp address="wiletrbackend" port="8282" uriPath="/training" enable="true"/>
```

As can be seen in section <endPointsGrouping> 2 symbolic names were assigned "pippo" and “pluto”, in the section <virtualDomain> two other symbolic names were assigned "paperino" and "minni" and only the <endp> with uriPath = "/TCOProject" have been associated with the names "qui", "quo" and "qua".

If during execution a file is created in the new directory as follows:

(LBL_HOME)/lib/notificationDir/outOfOrder.quo

only the two end points with uriPath = "/TCOProject" will be placed in out of order.

If during execution a file is created in the new directory as follows:

(LBL_HOME)/lib/notificationDir/outOfOrder.minni

all end points with that domain will be placed in out of order.

If during execution a file is created in the new directory as follows:

(LBL_HOME)/lib/notificationDir/outOfOrder.pippo

all end points within that group will be placed in out of order.

This method allows the systems personnel to use any tool to perform services checks and only creating/deleting a file to enable or disable entire groups of backend services.

<idSessionsManagement>

```
<serviceconf>
        <iproxy>
          <idSessionsManagement>
```

This section identifies and sets parameters for the application session management.

```
<idSessionsManagement>
 <idSessions>
    <id/>
 </idsession>
</idSessionsManagement>
```

<idSessions>

```
<serviceconf>
        <iproxy>
          <idSessionsManagement>
                <idSessions
```

It is possible to create different contexts for determination of the session and then associate them with the name, to a group of services, domain, up to a group of URIPath depending on the services to be managed.
It consists of a number of <id> that identify the entities determining the session.

enable=: default value=”true”
Enables or disables this group of parameters.

name=: default value=”default”
The name of the group-specific session identifier. This parameter enables identification of the different contexts of determination of the session of several service groups. The automatic value, if non-existent, is "default" that gets associated to all services unless stated explicitly.

lblSessionGeneration=: default value=”true”
Enables or disables session creation and management by OPLON® LoadBalancer (Loadbalancer managed session).

caseSensitive=: default value=”true”
If case sensitive session tags, session identifier names are considred different when using uppercase and lowercase letters. If set to "false" there will be no differentiation between lowercase and uppercase.

For example, JSESSIONID, with caseSensitive=”false”, is the same as jSessionId.

lblTagSession=: default value=”LBLSESSIONID”
Name of the session cookie if lblSessionGeneration = true

httpOnly=: default value=”false”

Sets the attribute httpOnly in the session cookie if lblSessionGeneration = true

secure=: default value=”false”
Sets the attribute secure in the session cookie if lblSessionGeneration = true

maxDelaySessionPropagation=: default value=”3000” UM=Milliseconds
Parameter used in the Enterprise version of OPLON® LoadBalancer. Indicates the maximum wait for a node of a peered cluster before declaring a session not found.

<id>

```
<serviceconf>
        <iproxy>
                <idSessionsManagement>
                        <idSessions>
                                <id
```

In the application session management identifies the entities determinant of the session.

context=: default value=””
Contains the significant value (entity) in the HTTP header that identifies the session. To create a rule for the first string HTTP a value must be set for "FIRST-LINE":
eg.
<id context="FIRST-LINE" tag="JSESSIONID" path="firts_line" enable="true"/>

tag=: default value=””
The name of the value to search for e.g.: JSESSIONID. The value is case-sensitive i.e. "JSESSIONID" is different from "jsessionid" which is different than "JsessionId".

caseSensitive=: default value=”<idSessions caseSensitive”
If case sensitive session tags, session identifier names are considred different when using uppercase and lowercase letters. If set to "false" there will be no differentiation between lowercase and uppercase.

For example, if caseSensitive = "false"; JSESSIONID = jSessionId. The value is inherited from the section <idSessions>but can be set for difference.

path=: default value=””
can take the following values: “first_line”, “header_fields”, “url”.

- header_fields
  searches the value with the expression name =value
- first_line
  searches the value in a uri /aaa/bbb?name=value
- url
  searches the value in an http url: http://www.aaa. .../ccc?name=value

enable=: default value=”true”
Enables or disables a session identification element.

Example <idSessions>

```
<idSessionsManagement>
  <idSessions>
        <id context="FIRST-LINE"          tag="JSESSIONID" path="first_line" enable="true"/>
        <id context="cookie"          tag="JSESSIONID" path="header_fields" enable="true"/>
        <id context="set-cookie"          tag="JSESSIONID" path="header_fields" enable="true"/>
        <id context="referer"          tag="JSESSIONID" path="url" enable="true"/>
        <id context="FIRST-LINE"          tag="jsessionid"      path="first_line" enable="true"/>
        <id context="cookie"          tag="jsessionid"      path="header_fields" enable="true"/>
        <id context="set-cookie"          tag="jsessionid"      path="header_fields" enable="true"/>
        <id context="referer"          tag="jsessionid"      path="url" enable="true"/>
  </idSessions>
</idSessionsManagement>
```

If the functionality of creating and managing sessions from OPLON®Application Delivery Controller is enabled, other <id>in section <idSessions> will not be evaluated and session management will be completely entrusted to OPLON® LoadBalancer.

<idSessions lblSessionGeneration="true" lblTagSession="LBLSESSIONID">
</idSessions>

To facilitate configuration of settings, two profiles of session management were created. The first does not handle any session; the second one manages the session generated by OPLON® with default parameters. Of it is possible to create new profiles with different names and with additional new parameters. The two automatically generated profiles are:

- "nosessions" which is equal to:
  <idSessions name="nosessions">
  </idSessions>
- "autosessions" which is equal to:
  <idSessions name="autosessions" lblSessionGeneration="true">
  </idSessions>

In the OPLON® Standard HA and OPLON® Enterprise HA distributions, the “nosessions” profile was associated with the endpoint /HealthCheck that only needs to determine the status of activity without any control or session generation.

<dosAddressesQuarantineList>

```
<serviceconf>
        <iproxy>
                <dosAddressesQuarantineList>
```

This section identifies the rules to exclude addresses from DoS Attack Prevention features and provides specifics of the DoS Address In Quarantine function.

```
<dosAddressesQuarantineList>
    <address>regular expression</address>
   ...
   ...
 </cacheControl>
```

<address>

```
<serviceconf>
        <iproxy>
                <dosAddressesQuarantineList>
                        <address>
```

The section indicates, through regular expressions, which addresses are to be excluded from quarantine management. If this causes excessive usage of resources, warning messages are generated.

```
<dosAddressesQuarantineList>
        <address>^192\.168\.43</address>
        <address>^127\.0\.0</address>
</dosAddressesQuarantineList>
```

The section described above can also be used in the following manner since the value is a regular expression.

```
<dosAddressesQuarantineList>
        <address>^(192\.168\.43|127\.0\.0)</address>
</dosAddressesQuarantineList>
```

<cacheControl>

```
<serviceconf>
        <iproxy>
                <cacheControl>
```

This section identifies the insertion rules for the entity "cache-control" based on the "content-type" reply from the backend server. For the groupings of “content-type”, this section is used to describe the entities to be inserted into the response HEADER from the server.

The cache control, once the "content-type" in the HEADER of the response from the application server is identified, deletes any "pre-existing entities" based on the list of entities to be inserted, and then inserts all the "entities" describe in that section. This feature allows proxies and clients to leverage, in the case they adhere to , the "rfc2616" stndard to control cache systems in response.
Reference to rfc2616: Cache-control Mechanisms, Cache-Control and in particular cache-response-directive.

```
<cacheControl>
  <cacheControlId>
     <contentType>
     </contentType>
     <entity>
     </entity>
  </cacheControlId>
</cacheControl>
```

<cacheControlId>

```
<serviceconf>
        <iproxy>
                <cacheControl>
                        <cacheControlId>
```

name=: default value=””
Identifying name for the group of the content-type for which the entities must be inserted into the response Header from the backend service to the client that made the request.

enable=: default value=”true”
Enables or disables this content-type group.

<contentType>

```
<serviceconf>
        <iproxy>
                <cacheControl>
                        <cacheControlId>
                                <contentType>
```

value=: default value=””
The value for which a match determines a Header entry with the entity indicated in the

section <entity>. The contentType(s) can also be repeated.

<entity>

```
<serviceconf>
        <iproxy>
                <cacheControl>
                        <cacheControlId>
                                <entity>
```

name=: default value=””
The name of the entity that should be inserted. Usually is "cache-control".

value=: default value=””
The value to be attributed to the entity.
eg.:
“max-age=300, must-revalidate”

Example <cacheControl>

Here is an example of setting paramters in the cacheControl section:

```
...
    <id context="set-cookie" tag="jsessionid" path="header_fields" enable="true"/>
     <id context="referer" tag="jsessionid" path="url" enable="true"/>
  </idsessions>

  <cacheControl>
     <cacheControlId name="firstCC" enable="true">
        <contentType value="image/png"/>
        <contentType value="image/gif"/>
        <entity name="Cache-Control" value="max-age=300, must-revalidate"/>
     </cacheControlId>
     <cacheControlId name="secondCC" enable="true">
        <contentType value="image/jpeg"/>
        <entity name="Cache-Control" value="max-age=3600, must-revalidate"/>
     </cacheControlId>
  </cacheControl>

  <endpoints>
      <!-############################  HTTP L7 ######################-->
     <endPointsGrouping enable="true">
        <virtualDomain enable="true"
                    virtualDomainName="monster"
                    loadBalancingType="Adaptative"
                    portRewriting="true"
                    cacheControl=" firstCC     secondCC  ">
           <endp address="wiletrbackend" port="8080" uriPath="/Flowers/album" enable="true"/>
           <endp address="roadtrbackend" port="8080" uriPath="/Flowers/album" enable="true"/>
           <endp address="wiletrbackend" port="8787" uriPath="/Flowers/album" enable="true"/>
     ...
```

Following is the result in the response HEADER before and after modification:

RESPONSE HEADER PRIOR TO INSERTION OF THE ENTITY:

HTTP/1.1 200 OK
server: Apache-Coyote/1.1
etag: W/"8802-1208088532000"
last-modified: Sun, 13 Apr 2008 12:08:52 GMT
content-type: image/jpeg
content-length: 8802
date: Sat, 21 Mar 2009 17:09:21 GMT

RESPONSE HEADER AFTER INSERTION OF THE ENTITY:

HTTP/1.1 200 OK
server: Apache-Coyote/1.1
etag: W/"8802-1208088532000"
last-modified: Sun, 13 Apr 2008 12:08:52 GMT
content-type: image/jpeg
content-length: 8802
date: Sat, 21 Mar 2009 17:09:21 GMT
cache-control: max-age=3600, must-revalidate <====== entity added

NOTE Newer browsers perform this check automatically by checking with the entities "etag" and "last-modify" the last change made in the object on the server.

If the object has not been modified, the response is returned with a value of 304 (Not modified) and the browser then draws the object from its own cache without reading it again from the network.

<sslCertificatesManagement>

```
<serviceconf>
    <iproxy>
        <sslCertificatesManagement>
```

This section contains the list of keystores used by the ADC to connect to applications the need SSL/TLS Client Authentication

```
<sslCertificatesManagement>
    <SSLCerts
        name=”... enable=”...>
    </SSLCerts>
    <SSLCerts>
        ...
        ...
</sslCertificatesManagement>
```

<SSLCerts>

```
<serviceconf>
    <iproxy>
        <sslCertificatesManagement>
            <SSLCerts
```

name=: default value=””
The identifying name of certificate management characteristics in the backend.

enable=: default value=”true”
Enables or disables this section.

SSLContextVersion=: default value=”SSLv3”
Indicates the version of the SSL protocol. Normally set to "SSLv3" for JVM keystore or "TLS" for OpenSSL.

certificateURL=: default value=””
If set indicates the http address from which to get the client certificate.

certificateURIPath=: default value=””
If set indicates the path from where to get the client certificate.

keyStore=: default value=”JKS”
Indicates the SSL keystore type from which to get the client certificate. Normally if the JVM keystore is used, this must be set to "JKS" and if an OpenSSL keystore is used, it must be set to “PKCS12”.

keyStorePassword=: default value=”defaultpwd”
Password to access the keystore.

keyManagerFactory=: default value=”SunX509”
Indicates the interpretation paradigm of the certificate. Normally set to "SunX509".

enableClientCertificate=: default value=”false”
If set to true and is an SSL transmission, use of the client certificate is enabled.

trustAllCertificates=: default value=”false”
If this value is set to true, there is no verification through CA or certificate truststore. Useful in testing or to use the service certificate for the sole purpose of executing an encrypted transmission.

trustCertificateURL=: default value=””
If set indicates the http address to get the trusted certificate.

trustCertificateURIPath=: default value=””
If set indicates the path to get the trusted certificate.

trustKeyStore=: default value=”JKS”
Indicates the SSL trusted keystore type. Normally if the JVM keystore is used, this must be set to "JKS" and if an OpenSSL keystore is used, it must be set to “PKCS12”.

trustKeyStorePassword=: default value=”defaultpwd”
Password to access the trusted keystore.

trustKeyManagerFactory=: default value=”SunX509”
Indicates the interpretation paradigm of the certificate. Normally set to "SunX509".

Example <sslCertificatesManagement>

```
<sslCertificatesManagement>
        <SSLCerts name="mysslcerts" enable="true"
                SSLContextVersion="SSL"
                enableClientCertificate="true"
                certificateURL=""
                certificateURIPath="security/certificate/serverkeys"
                keyStore="JKS"
                keyStorePassword="defaultpwd"
                keyManagerFactory="SunX509"
                trustAllCertificates="false"
                trustCertificateURL=""
                trustCertificateURIPath="security/certificate/serverkeys"
                trustKeyStore="JKS"
                trustKeyStorePassword="defaultpwd"
                trustKeyManagerFactory="SunX509">
        </SSLCerts>
        <SSLCerts name="my1sslcerts" enable="true"
                SSLContextVersion="SSL"
                enableClientCertificate="true"
                certificateURL=""
                certificateURIPath="security/certificate/serverkeys"
                keyStore="JKS"
                keyStorePassword="defaultpwd"
                keyManagerFactory="SunX509"
                trustAllCertificates="false"
                trustCertificateURL=""
```

```
                trustCertificateURIPath="security/certificate/serverkeys"
                trustKeyStore="JKS"
                trustKeyStorePassword="defaultpwd"
                trustKeyManagerFactory="SunX509">
        </SSLCerts>
        <SSLCerts name="my2sslcerts" enable="true"
                SSLContextVersion="SSL"
                enableClientCertificate="true"
                certificateURL=""
                certificateURIPath="security/certificate/serverkeys"
                keyStore="JKS"
                keyStorePassword="defaultpwd"
                keyManagerFactory="SunX509"
                trustAllCertificates="false"
                trustCertificateURL=""
                trustCertificateURIPath="security/certificate/mycert.p12"
                trustKeyStore="PKCS12"
                trustKeyStorePassword="defaultpwd"
                trustKeyManagerFactory="SunX509">
        </SSLCerts>
</sslCertificatesManagement>
```

For use on endpoints simply include the SSLCert parameter in sections <endPointsGrouping> or <virtualDomain> or directly on the endpoint.

```
<endPointsGrouping enable="true" SSLCerts="mysslcerts">
        <virtualDomain rewriteHeaderRules="websphereEntities006_001" SSLCerts="my1sslcerts" enable="true">
          <endp address="wiletrbackend" port="1443" SSL="true" SSLCerts="my1sslcerts" uriPath="" enable="true"/>
          <endp address="roadtrbackend" port="1443" SSL="true" SSLCerts="my2sslcerts" uriPath="" enable="true"/>
          <endp address="wiletrbackend" port="2443" SSL="true" SSLCerts="my1sslcerts" uriPath="" enable="true"/>
          <endp address="roadtrbackend" port="2443" SSL="true" SSLCerts="my2sslcerts" uriPath="" enable="true"/>
        </virtualDomai
```

<rewriteManagement>

```
<serviceconf>
    <iproxy>
        <rewriteManagement>
```

This section describes the rules for rewriting both the HTTP HEADER and HTTP BODY contents during the forwarding of information. OPLON® LoadBalancer natively implements an advanced system of rewriting, combining both the expressive simplicity of XML with the ability to use (even in combination with XML), the JAVA programming language through extension of classes specifically designed for the purpose.
This tool provides engineers a powerful tool with endless possibilities not just of rewriting but also integration with other platforms such as SSO or traffic billing systems.

Within this section two types of sub-sections may co-exist:

- <rewriteHeaderRule> describes the rules for rewriting the HTTP HEADER components
- <rewriteBodyRule> describes the rules for rewriting the HTTP BODY components

The sub-sections/tags <rewriteHeaderRule> and <rewriteBodyRule> may be repeated several times inside the section <rewriteManagement> and describe behavioral templates to be applied subsequently at the Domain level <virtualDomain> up to the individual endpoint <endp>.

The general structure of the <rewriteManagement> section is as follows:

```
<rewriteManagement>
  <rewriteHeaderRule>
      <requestURLMatches></requestURLMatches>
      <mimeType/>
      <variables>
          <var>
            <regexTag></regexTag>
            <replaceTo></replaceTo>
          </var>
      </variables>
      <conditions>
          <cond>
            <regexTag></regexTag>
           <numOperatorTag></numOperatorTag>
          </cond>
      </conditions>
      <entities>
          <entity>
            <regexTag></regexTag>
            <replaceTo></replaceTo>
          </entity>
      </entities>
      <redirectTo/>
      <displaceEndPointsGrouping/>
```

```
        <connectionToCut/>
    </rewriteHeaderRule>

    <rewriteBodyRule>
            <requestURLMatches></requestURLMatches>
        <mimeType/>
        <requestURLMatches></requestURLMatches>
        <mimeType/>
         <variables>
            <var>
               <regexTag></regexTag>
               <replaceTo></replaceTo>
            </var>
         </variables>
         <conditions>
            <cond>
               <regexTag></regexTag>
            </cond>
         </conditions>
        <regexTag></regexTag>
        <replaceTo></replaceTo>
     </rewriteBodyRule>
</rewriteManagement>
```

<rewriteHeaderRule>

```
<serviceconf>
    <iproxy>
        <rewriteManagement>
            <rewriteHeaderRule
```

The sub-section <rewriteHeaderRule> describes a rewriting rule that applies to the HTTP HEADER. This sub-section can be repeated as many times as the rules to be applied in different contexts of forwarding require. The parameters for this section describe the characteristics of the rule to be expressed and the base conditions.

name=: default value=”null”
The name of the rule. This value, associated with one or more domains <virtualDomain> up to the individual endpoint <endp>, allows application of the rule in multiple contexts. If the value has not been set the rule is not loaded.

enable=: default value=”true”
Enables or disables this rule.

flow=: default value=”BOTH” available values: REQUEST|RESPONSE|BOTH
Identification of the flow/stream in which to apply the rule.
If the REQUEST flow is selected, the rule is applied in this manner (incoming request):
client----> LBL---> endp.

If the RESPONSE flow is selected, the rule is applied in this manner (outgoing response):

client-<---LBL <---endp.

If BOTH is selected, the rule is apllied for both incoming requests from clients and outgoing responses from the services to the client.

httpMethod=: default value=”” available values: Methods HTTP: POST, GET...
In REQUEST streams, identifies the HTTP method used
e.g. POST GET PATCH
If not set or empty value is performed on all methods.
In cases of flow = "BOTH", this value will be verified only in the REQUEST streams and thus can coexist with the responseCode parameter.

responseCode=: default value=”-1”
In RESPONSE streams, identifies the HTTP return code and for which the rule will be applied. In cases of flow = "BOTH", this value will be verified only in the RESPONSE streams and thus can coexist with the httpMethod parameter.

caseSensitive=: default value=”false”
This value describes the matching method for the rewriting rules (regular expressions). This value is propagated to its sub-sections where it can nevertheless be changed for required exceptions.

httpInterceptorClassPath=:default=”interceptors/”
The loading path of the interceptor class defined in httpInterceptorClass. This path is added to the running JVM classpath.

httpInterceptorClass=: default value=””
Indicates the class to intercept HTTP data packets.

This parameter specifies a class that implements the class:
- loadbalancer.rewriter.LBLHTTPInterceptorHeaderAbstr

This is made available from the OPLON® LoadBalancer platform to take action on the HEADER both before and after the application of the rules described in this section. The distributions already contain a class template available in

(LBL_HOME)/interceptors/rewriteclasses/LBLHTTP.... .java

The classes that are contained in this directory can be compiled through the tools: compile.sh or compile.bat present in the same directory.
Instructions for use of the interceptor classes templates is described within the classes themselves.

In any interceptor class you can use two methods called in the initialization of the class and the termination of the class. These methods are used to allocate resources or initialize values. The termination method of the class is used to deallocate resources during reinit or gracefull termination of load balancing process.

```
@Override
```

```
public void interceptorInit() {
        /* initialization code */
}

@Override
public void interceptorEnd() {
        /* termination code */
}
```

LBLHTTPInterceptorHeaderAbstr

The "loadbalancer.rewriter.LBLHTTPInterceptorHeaderAbstr" class provides 4 methods that identify the four key situations of the data flow:

```
package loadbalancer.rewriter;

import loadbalancer.rewriter.LBLHTTPInterceptorHeaderAbstr;
import loadbalancer.rewriter.LBLHTTPInterceptorHeaderStreamFragment;

/**
 * Test class HTTP HEADER Interceptor for dynamic change during stream...
 * @version 1.0 Created on 5-jun-2010
 */
public class LBLHTTPRewriteInterceptorHeaderLogging extends LBLHTTPInterceptorHeaderAbstr {

        /** copyright */
        public static final String COPYRIGHT="LBL and TCOProject are trademarks all rights reserved";

        @Override
        public void doRequestHeaderBeforeReplace(LBLHTTPInterceptorHeaderStreamFragment streamFragment) {
                logWarning("REQUEST HEADER BEFORE REPLACE\n"+
                streamFragment.getRequestRowImageStreamFragment());
                for (String varName: streamFragment.getVariables())
                logWarning("RQBR HEADER VarName:"+varName+" value:"+streamFragment.getVariable(varName));
        }

        @Override
        public void doRequestHeaderAfterReplace(LBLHTTPInterceptorHeaderStreamFragment streamFragment) {
                logWarning("REQUEST HEADER AFTER REPLACE\n"+
                streamFragment.getRequestRowImageStreamFragment());
        }

        @Override
        public void doResponseHeaderBeforeReplace(LBLHTTPInterceptorHeaderStreamFragment streamFragment) {
                logWarning("RESPONSE HEADER BEFORE REPLACE\n"
                +streamFragment.getResponseRowImageStreamFragment());
                for (String varName: streamFragment.getVariables())
                        logWarning("REBR HEADER VarName:"+varName
                        +" value:"+streamFragment.getVariable(varName));
        }

        @Override
        public void doResponseHeaderAfterReplace(LBLHTTPInterceptorHeaderStreamFragment streamFragment) {
                logWarning("RESPONSE HEADER AFTER REPLACE\n" +
                streamFragment.getResponseRowImageStreamFragment());
        }
}
```

Each method provides the LBLHTTPInterceptorHeaderStreamFragment object with which enables actions on the data fragment during the stream (in this case the entire HTTP HEADER).

LBLHTTPInterceptorHeaderStreamFragment already contains many methods for acting on the HTTP protocol both for query of values as well as controlled modifications. A full dissertation with examples can be found in the manual "OPLON® LoadBalancer Content Rewriting".

LBLHTTPInterceptorHeaderAbstr is the class from which the interceptor classes are derived. It contains some utility methods for the most common operations such as processing digital certificates in different formats and/or verification of significant values such as the fingerprint or the serial number.

Following is a detailed list of methods of the classes:
LBLHTTPInterceptorHeaderStreamFragment
LBLHTTPInterceptorHeaderAbstr

LBLHTTPInterceptorHeaderStreamFragment methods

```
/**
 * return client host address
 * @return client host address or null  if not found
 */
public String getRequestClientAddress()

/**
 * return endpoint host address
 * @return endpoint host address or null  if not found
 */
public String getResponseEndpointAddress()

/**
 * Vector with cookies names
 * @return Vector with cookies names
 */
public Vector<String> getHTTPRequestCookiesNames()

/**
 * Get a cookie value
 * @param cookieName cookie name to find
 * @return cookie value or null if not found
 */
public String getHTTPRequestCookie(String cookieName)

/**
 * get request URL with params and query string
 * @return URL with params and query string or null if not found
 */
public String getHTTPRequestURL()

/**
 * get request URIPath without params and query string
 * @return URIPath without params and query string or null if not found
 */
public String getHTTPRequestURIPath()

/**
 * get request URL last element
 * @return URL last element
 */
public String getHTTPRequestURLLastElement()
```

```
/**
 * get host name of request
 * @return host name of request, null if not found
 */
public String getHTTPRequestHostName()

/**
 * get port number of request
 * @return port number of request, -1 if not found
 */
public int getHTTPRequestHostPort()

/**
 * get http request method
 * @return request method, null if not found
 */
public String getHTTPRequestMethod()

/**
 * get http version
 * @return 10 for HTTP 1.0, 11 for HTTP 1.1. -1 if not found
 */
public int getHTTPVersion()

/**
 * get parameter o query string in request
 * @param parameterName parameter name to find
 * @return parameter value or null if not found
 */
public String getHTTPRequestParam(String parameterName)

/**
 * get parameter o query string in a url
 * @param url url to find parameter
 * @param parameterName parameter name to find
 * @return parameter value or null if not found
 */
public String getHTTPUrlParam(String url, String parameterName)

/**
 * get response code
 * @return response code or -1 if not found
 */
public int getHTTPResponseCode()


// ******************************************************
// ENTITIES
// ******************************************************

/**
 * get entity value
 * @param entityName entity name es.: Content-Type
 * @return entity value or null if not match
 */
public String getHTTPEntity(String entityName)

/**
 * get entities
 * @return a vector of strings array where element[0] is enttity name (name in lowercase) and element[1] is entity value
 */
public Vector<String[]> getHTTPEntities() {

/**
```

```
 * Remove an enity
 * @param entityName
 */
public void removeEntity(String entityName)

/**
 * Append an entity
 * Warning: this method doesn't remove any entity... use changeEntity for change it.
 * @param entity
 */
public void appendEntity(String entityName, String entityValue)

/**
 * Change an entity
 * @param entity
 */
public void changeEntity(String entityName, String entityValue)

/**
 * Replace first line
 * @param firstLineHeader new first header line
 */
public void changeFirstHeaderLine(String firstLineHeader)


// ************************************************************************
// VARIABLES
// ************************************************************************

/**
 * Add variable
 * @param name variable name
 * @param value variable value
 */
public void addVariable(String name, String value)

/**
 * Get a variable value
 * @param name variable name
 * @return variable value or null
 */
public String getVariable(String name)

/**
 * Replace all variable with their values
 * @param stringToReplace string to replace
 * @return string replaced
 */
public String replaceStringWithVariable(String stringToReplace)

/**
 * Get a list of variables names
 * @return list of variables names
 */
public Vector<String> getVariables()

// ========================================
// GETTER
// ========================================

/**
 * @return the contentType
 */
public TCONameValue getContentType()
```

```
/**
 * @return the endPointGroup
 */
public LBLEndPointGroup getEndPointGroup()

// ************************************************************************
// REDIRECT
// ************************************************************************

/**
 * @return the redirectTo
 */
public String getRedirectTo()

/**
 * @param redirectTo the redirectTo to set
 */
public void setRedirectTo(String redirectTo)

/**
 * @return the redirectResponseCode
 */
public int getRedirectResponseCode()

/**
 * @param redirectResponseCode the redirectResponseCode to set
 */
public void setRedirectResponseCode(int redirectResponseCode)

// ************************************************************************
// END POINTS GROUPING
// ************************************************************************
/**
 * se != null nome dell'endPointsGrouping su cui spiazzare la richiesta
 * @return the endPointsGrouping
 */
public String getEndPointsGrouping()

/**
 * se != null nome dell'endPointsGrouping su cui spiazzare la richiesta
 * @param endPointsGrouping the endPointsGrouping to set
 */
public void setEndPointsGrouping(String endPointsGrouping)

// ************************************************************************
// STREAM VIEW
// ************************************************************************

/**
 * Image of consolidated request header
 * @return Image of consolidated request header
 */
public String getRequestImageStreamFragment()

/**
 * Image of consolidated response header
 * @return Image of consolidated response header
 */
public String getResponseImageStreamFragment()

/**
 * Image of consolidated response header
 * @return Image of consolidated response header
```

```
 */
private String getImageStreamFragment(boolean request)

/**
 * Return an image of request header in row format
 * @return image of request header in row format
 */
public String getRequestRowImageStreamFragment()

/**
 * Return an image of resposne header in row format
 * @return image of resposne header in row format
 */
public String getResponseRowImageStreamFragment()
```

LBLHTTPInterceptorHeaderAbstr methods

```
/**
 * Log a message as error
 * @param logMessage message to log
 */
public void logError(String logMessage)

/**
 * Log a message as warning
 * @param logMessage message to log
 */
public void logWarning(String logMessage)

/**
 * Log a message as debug
 * @param logMessage message to log
 */
public void logDebug(String logMessage)

// ============================================================================
// UTILITY
// ============================================================================
/**
 * base-64 encode a string
 * @param s The ascii string to encode
 * @return The base64 encoded result or null if error occurs
 */
public static byte[] encodeBase64(String s)

/**
 * base-64 encode a byte array
 * @param src The byte array to encode
 * @return The base64 encoded result or null if error occurs
 */
public static byte[] encodeBase64(byte[] src)

/**
 * base-64 encode a byte array
 * @param src The byte array to encode
 * @param start The starting index
 * @param length The number of bytes
 * @return The base64 encoded result or null if error occurs
 */
public static byte[] encodeBase64(byte[] src, int start, int length)

/**
 * A Base64 decoder.  This implementation is slow, and
```

```
 * doesn't handle wrapped lines.
 * The output is undefined if there are errors in the input.
 * @param s a Base64 encoded string
 * @return The byte array eith the decoded result or null if error occurs
 */
public static byte[] decodeBase64(String s)

/* ========================================================================== *
 * USEFULL METHODS
 * ========================================================================== */

/**
 * Certificate dates validation
 * @param mSession SSL session instance
 * @return true certificate dates in range
 */
public static boolean checkCertValidity(SSLSession mSession)

/**
 * Serial number string rappresentation
 * @param certImpl certificate
 * @return Serial number string rappresentation or null
 */
public static String getCertSerialNumberStringFormat(java.security.cert.X509Certificate certImpl)

/**
 * Serial number string hex format rappresentation example: ca701af1425980c7
 * @param certImpl certificate
 * @return Serial number string hex format rappresentation example: ca701af1425980c7
 */
public static String getCertSerialNumberHexStringFormat(java.security.cert.X509Certificate certImpl)

/**
 * Certificate fingerprint with specified alcorithm example: b751c47e45b51ba79a1c167a335a67a427f4d702
 * @param certImpl certificate
 * @param algorithm algorithm example: SHA, MD5
 * @return Certificate fingerprint with specified alcorithm example: b751c47e45b51ba79a1c167a335a67a427f4d702 or null
 */
public static String getCertFingerprintStringFormat(java.security.cert.X509Certificate certImpl, String algorithm)

/**
 * Certificate exipiration date string rappresentation
 * @param certImpl certificate
 * @return Certificate exipiration date string rappresentation example: "2010-08-09 13:47:10.532 UTC" or null
 */
public static String getCertNotAfterStringFormat(java.security.cert.X509Certificate  certImpl)

/**
 * Certificate activation date string rappresentation
 * @param certImpl certificate
 * @return Certificate activation date string rappresentation example: "2010-08-09 13:47:10.532 UTC" o null se non applicabile
 */
public static String getCertNotBeforeStringFormat(java.security.cert.X509Certificate certImpl)

/**
 * Certificate subject
 * @param certInfo certificate
 * @return Certificate subject or null
 */
public static String getCertSubject(java.security.cert.X509Certificate certInfo)

/**
 * Certification Authority subject (CA Subject)
 * @param cert certificate
```

```
 * @return Certification Authority subject (CA Subject) or null
 */
public static String getCertIssuer(java.security.cert.X509Certificate cert)
```

<requestURLMatches></requestURLMatches>

```
<serviceconf>
        <iproxy>
                <rewriteManagement>
                        <rewriteHeaderRule>
                                <requestURLMatches></requestURLMatches>
```

The value inside the section <requestURLMatches> identifies a regular expression. This is used to verify the conformity with the URL request, including the URI Params and Query String. If this condition is satisfied the rule is applied. If this section is not given a value or is non-existent, all URL requests will be considered valid. The value of the URL request is formed without UTF-8 encoding (spaces are represented by spaces and not by their URL encoded representation).

Example:
<requestURLMatches>^/viewProcessProperties.html\?
process=A05_LBLGoDNSManager$</requestURLMatches>

<mimeType>

```
<serviceconf>
        <iproxy>
                <rewriteManagement>
                        <rewriteHeaderRule>
                                <mimeType
```

This section, which may be repeated, is another condition for which the rule may be applied to the stream.

enable=: default value=”true”
Enables or disables this section.

value=: default value=””
Value of the mimeType with which the rule may be applied.

Examples:
"text/html"
"text/css".

<variables>

```
<serviceconf>
        <iproxy>
                <rewriteManagement>
                        <rewriteHeaderRule>
                                <variables>
```

This section enables the loading, through several sources, of variables for use within the rule to compose or edit the values and make them available during the rewriting.

Creation of a variable takes place using the sub-section <var> which can be repeated many times.

eg.:
```
<var varName="MY_VAR_FROM_ENTITY"       name="Connection"
                                        from="ENTITY">
</var>
```

In this case the MY_VAR_FROM_ENTITY variable is created taking the value from the "Connection" entity in the HEADER during the transfer.
The value contained in MY_VAR_FROM_ENTITY could be assessed during the data flow with "Keep-alive" or "Close" or as called for by the HTTP protocol.

In the following example we have also introduced a modifier with regular expression.

```
<var varName="MY_VAR_FROM_ENTITY"       name="Connection"
                                        from="ENTITY">
            <regexTag>Keep-alive</regexTag>
            <replaceTo>Close</replaceTo>
</var>
```

In this case the MY_VAR_FROM_ENTITY variable is created taking the value from the "Connection" entity in the HEADER during the transfer and the value "Keep-alive" is transformed to "Close".

The end result will be a MY_VAR_FROM_ENTITY variable with contents "Close" if the previous value "Connection" contained "Keep-alive" otherwise it will contain the original entity value.

It is also possible to use other variables previously created in the same rule to form more variables.

eg.:
```
<var varName="MY_PARAM"       name="paramName"
                              from="URI_PARAM">
      <regexTag>valueInParamQueryString%MY_VAR_FROM_ENTITY%</regexTag>
            <replaceTo>newValueInVar</replaceTo>
</var>
```

In this case the value of the <regexTag>, prior to being used, will be converted to the value of the variable at the instant of transmission.

<var>

```
<serviceconf>
    <iproxy>
        <rewriteManagement>
            <rewriteHeaderRule>
                <variables>
                    <var
```

The parameters of the variable definitions contained in <variables> are:

enable=: default value=”true”
Enables or disables this section

varName=: default value=””
Name of the variable to be used in the rule.

name=: default value=””
Resource name/value from which to derive the value to be loaded into the variable (If value is constant and supports the modifier % VARIABLE%)

caseSensitive=: default value=”caseSensitive del paragrafo rewrite rule”
This value describes the matching method for the rewriting rules (regular expressions).

from=: default value=”” values: INNERVAR|ENTITY|URI_PARAM|CONSTANT| COOKIE|VARIABLE
Source from which to load the value into the variable.

INNERVAR

- **REQUEST_HTTP_URL**
  Request URL with params and query string
- **REQUEST_HTTP_URL_DECODED**
  Request URL with params and query string decoded
- **REQUEST_HTTP_URL_LAST_ELEMENT**
  Only last element of the URL without params and query string
- **REQUEST_HTTP_URL_LAST_ELEMENT_DECODED**
  last element of the URL without params and query string in decoded format
- **REQUEST_HTTP_URI_PATH**
  Only URI Path whithout parameters and query string
- **REQUEST_HTTP_URI_PATH_DECODED**
  URI Path whithout parameters and query string in decoded format
- **REQUEST_HTTP_HOST_NAME**
  Hostname in entity "Host"

ATTENTION: Use of REQUEST_HTTP_HOST_NAME can result in name resolution through DNS. If the name is not associated with any address its timeout can cause severe slowdown.

- **REQUEST_HTTP_HOST_PORT**
  Port number in entity "Host"
- **REQUEST_HTTP_COOKIES_LIST**
  List of cookies names separated by ";"
- **REQUEST_CLIENT_ADDRESS**
  TCP client address
- **RESPONSE_ENDPOINT_ADDRESS**

TCP endpoint address

- **REQUEST_INCOMING_COUNTRY**
  Country state (ISO 3166 2-letter code) or ".." "ZZ"
- **ENTITY_REQUEST**
  Return the value of request entity name or empty if no entity found
- **ENTITY_RESPONSE**
  Return the value of response entity name or empty if no entity found
- **SSL_CONNECTION_CLIENT**
  True if the client connects with SSL through LBL
- **SSL_CONNECTION_ENDPOINT**
  True if endpoint connects with SSL through LBL
- **SSL_CONNECTION_REENCRYPTION**
  True if SSL reencryption is performed (LBL terminates SSL and connects to the endpoint in SSL)
- **REQUEST_INCOMING_ADDRESS**
  Local address on which the service request was accepted
- **REQUEST_INCOMING_HOST_NAME**
  Host name or local address on which the service request was accepted

---

ATTENTION: Use of REQUEST_HTTP_HOST_NAME can result in name resolution through DNS. If the name is not associated with any address its timeout can cause severe slowdown.

- **REQUEST_HTTP_SCHEME**
  http or https depending on the client's connection to LBL
- **HIGH_WATER_YELLOW_WARNING_REACHED**
  If true the Yellow Warning threshold has been exceeded. Indicates a significant load but still not critical
- **HIGH_WATER**
  Number of connection requests in the queue in long format.
- **HIGH_WATER_LEVEL**
  Float value, % Of connection requests in the queue compared to the number of tunnels in current settings.
- **TUNNEL_SESSIONS_ACTIVE**
  Simultaneous active tunnels, numerical format
- **TUNNEL_SESSIONS_COMMITTED**
  Simultaneous tunnels committed, (subset of TUNNEL_SESSIONS_ACTIVE)
- **ACTUAL_TUNNEL_SESSIONS_SIZE**
  The actual size of tunnels: (usually equal to"MAX_TUNNEL_SESSIONS_SIZE")
- **MAX_TUNNEL_SESSIONS _SIZE**
  Maximum number of tunnels set.

NOTE: These 'pre-loaded" values to be used as modifiers (% xxx%) must be loaded into a local variable to the rule.

ENTITY

ENTITY = loading of the variable described in varName with the value of the HTTP HEADER Entity whose name is indicated in name.

URI_PARAM
URI_PARAM = loading of the variable described in varName with the parameter value or query string of the HTTP HEADER whose name is indicated in name.

URI_PARAM_DECODED
URI_PARAM = loading of the variable described in varName with the parameter value or query string of the HTTP HEADER whose name is indicated in name.

CONSTANT
CONSTANT = loading of the variable described in varName with the parameter value indicated in name. Only in this case may the value in name consist of another variable previously loaded.

COOKIE

COOKIE = loading of the variable described in varName with the value of the HTTP HEADER Cookie whose name is indicated in name.

VARIABLE
VARIABLE = loading of the variable described in varName with the value of another variable whose name is indicated in name.

<regexTag></regexTag>
<replaceTo></replaceTo>

```
<serviceconf>
    <iproxy>
        <rewriteManagement>
            <rewriteHeaderRule>
                <variables>
                    <var>
                        <regexTag></regexTag>
                        <replaceTo></replaceTo>
                    </var>
```

The <regexTag> and <replaceTo> sections are not mandatory but unique if present in the <var> section, are assessed with regular expressions to apply changes to the original values obtained from various sources.
These two values can also contain further modifying variables expressed in previous <var> sections inherent of the same rewriting rule.

In particular:
<regexTag>
Regular expression applied in the extracted value (Supports % VARIABLE% and caseSensitive)
</regexTag>
<replaceTo>
Value with which to replace (Supports % VARIABLE% and caseSensitive)

```
        </replaceTo>
```

Example <variables>

```
<variables>
  <!-- load variables from innervar -->
  <var varName="MY_REQUEST_HTTP_URL"                name="REQUEST_HTTP_URL"                from="INNERVAR"/>
  <var varName="MY_REQUEST_LAST_URL_ELEMENT"        name="REQUEST_HTTP_URL_LAST_ELEMENT"   from="INNERVAR"/>
  <var varName="MY_REQUEST_HTTP_URI_PATH"           name="REQUEST_HTTP_URI_PATH"           from="INNERVAR"/>
  <var varName="MY_REQUEST_HTTP_HOST_NAME"          name="REQUEST_HTTP_HOST_NAME"          from="INNERVAR"/>
  <var varName="MY_PORT"                            name="REQUEST_HTTP_HOST_PORT"          from="INNERVAR"/>
  <var varName="MY_RESPONSE_ENDPOINT_ADDRESS"       name="RESPONSE_ENDPOINT_ADDRESS"       from="INNERVAR"/>
  <var varName="MY_REQUEST_HTTP_HOST_PORT"          name="REQUEST_HTTP_HOST_PORT"          from="INNERVAR">
        <regexTag>%MY_PORT%</regexTag>
        <replaceTo>10100</replaceTo>
  </var>
  <var varName="MY_REQUEST_CLIENT_ADDRESS"          name="REQUEST_CLIENT_ADDRESS"          from="INNERVAR"/>
  <var varName="MY_RESPONSE_ENDPOINT_ADDRESS"       name="RESPONSE_ENDPOINT_ADDRESS"       from="INNERVAR"/>
  <var varName="MY_REQUEST_HTTP_COOKIES_LIST"       name="REQUEST_HTTP_COOKIES_LIST"       from="INNERVAR"/>

  <!-- load variables from contant values -->
  <var varName="MY_IS"                              name="is"                              from="CONSTANT"/>
  <var varName="MY_VAR_FROM_THIS_VALUE"             name="this is a my value %MY_REQUEST_HTTP_URL%"  from="CONSTANT">
        <regexTag>this %MY_IS% a my value</regexTag>
        <replaceTo>this %MY_IS% a my value %MY_REQUEST_HTTP_HOST_PORT% HTTP_URL</replaceTo>
  </var>

  <!-- load variables from header entity -->
  <var varName="MY_VAR_FROM_ENTITY"                 name="Connection"                      from="ENTITY">
        <regexTag>Keep-alive</regexTag>
        <replaceTo>Close</replaceTo>
  </var>

  <!-- load variables from uri params or query string -->
  <var varName="MY_PROCESS"                         name="process"                         from="URI_PARAM"/>
  <var varName="MY_PARAM"                           name="paramName"                       from="URI_PARAM">
        <regexTag>valueInParamQueryString</regexTag>
        <replaceTo>newValueInVar</replaceTo>
  </var>

  <!-- extract cookie value -->
  <var varName="MY_VAR_COOKIE_LBLSESSIONID"         name="LBLSESSIONID"                    from="COOKIE"/>
  <var varName="MY_VAR_COOKIE_TCOPROJECTAUTH"       name="TCOPROJECTAUTH"                  from="COOKIE"/>
  <var varName="MY_VAR_COOKIE_TCOPROJECTSESSIONID"  name="TCOPROJECTSESSIONID"             from="COOKIE"/>
  <var varName="MY_VAR_COOKIE_JSESSIONID"           name="JSESSIONID"                      from="COOKIE"/>
  <var varName="MY_VAR_COOKIE_jsessionid"    name="jsessionid"                   from="COOKIE"/>
  <var varName="MY_VAR_COOKIE_NOTFOUND"             name="NOTFOUND"                        from="COOKIE"/>

  <!-- only cookies list names -->
  <var varName="MY_REQUEST_HTTP_COOKIES_LIST000" name="REQUEST_HTTP_COOKIES_LIST"          from="INNERVAR">
        <regexTag>(.*)(LBLSESSIONID)(.*)</regexTag>
        <replaceTo>$2</replaceTo>
  </var>
  <var varName="MY_REQUEST_HTTP_COOKIES_LIST001" name="REQUEST_HTTP_COOKIES_LIST"          from="INNERVAR">
        <regexTag>(.*)(TCOPROJECTAUTH)(.*)</regexTag>
        <replaceTo>$2</replaceTo>
  </var>
  <var varName="MY_REQUEST_HTTP_COOKIES_LIST002" name="REQUEST_HTTP_COOKIES_LIST"          from="INNERVAR">
        <regexTag>(.*)(TCOPROJECTSESSIONID)(.*)</regexTag>
        <replaceTo>$2</replaceTo>
  </var>

  <!-- extract cookies from entities -->
  <var varName="MY_VAR_COOKIES"                     name="Cookie"                          from="ENTITY">
        <regexTag>(.+)</regexTag>
        <replaceTo>$1;</replaceTo>
  </var>
  <!-- extract cookies from entity through variable -->
  <var varName="MY_VAR_COOKIE_LBLSESSIONID_FV00" name="MY_VAR_COOKIES"                     from="VARIABLE">
        <regexTag>(.*)(LBLSESSIONID=)(.+?)((;+?)(.*))</regexTag>
        <replaceTo>$3</replaceTo>
  </var>
```

```
<var varName="MY_VAR_COOKIE_LBLSESSIONID_FV01" name="MY_VAR_COOKIES"          from="VARIABLE">
        <regexTag>(.*)(TCOPROJECTAUTH=)(.+?)((;+?)(.*))</regexTag>
        <replaceTo>$3</replaceTo>
</var>
<var varName="MY_VAR_COOKIE_LBLSESSIONID_FV02" name="MY_VAR_COOKIES"          from="VARIABLE">
        <regexTag>(.*)(TCOPROJECTSESSIONID=)(.+?)((;+?)(.*))</regexTag>
        <replaceTo>$3</replaceTo>
</var>

</variables>
```

<conditions>

```
<serviceconf>
        <iproxy>
                <rewriteManagement>
                        <rewriteHeaderRule>
                                <conditions>
```

In this optional section other conditions for applying the rule can be added along those specified in section <rewriteHeaderRule>.
This section is all-inclusive and conditions can be realized a on all HEADER values from streams.

This section can also be used in "AND" or "OR" modes as it relates to the conditions described in the sub-sections.

enable=: default value=”true”
Enables or disables this section.

operator=: default value=”AND” values: AND|OR
This value describes the matching methods for the conditions of rewriting. If used in "AND" mode all the conditions described in the sub-sections must be met in order for the rule to be applied otherwise if in "OR" mode the rule is applied even if one condition is met.

<cond>

```
<serviceconf>
        <iproxy>
                <rewriteManagement>
                        <rewriteHeaderRule>
                                <conditions>
                                        <cond
```

enable=: default value=”true”
Enables or disables this section.

name=: default value=””
Name of the condition.

caseSensitive=: default value=”caseSensitive value of the <rewriteHeaderRule> section”
This value describes the matching method for the rewriting rules (regular expressions).

eval=: default value=”” values: NOT|""
The value expressed in eval, if loaded as "NOT" , is the negation of the result of the regular

expression.

from=: default value="" values: INNERVAR|ENTITY|URI_PARAM|CONSTANT|COOKIE|VARIABLE
Source from which to load the value of the condition.

INNERVAR

- **REQUEST_HTTP_URL**
  Request URL with params and query string
- **REQUEST_HTTP_URL_DECODED**
  Request URL with params and query string decoded
- **REQUEST_HTTP_URL_LAST_ELEMENT**
  Only the last element of the URL without params and query string
- **REQUEST_HTTP_URL_LAST_ELEMENT**
  only last element of the URL without params and query string
- **REQUEST_HTTP_URI_PATH**
  Only URI Path whithout parameters and query string
- **REQUEST_HTTP_URI_PATH_DECODED**
  URI Path whithout parameters and query string in decoded format
- **REQUEST_HTTP_HOST_NAME**
  Hostname in entity "Host"

ATTENTION: Use of REQUEST_HTTP_HOST_NAME can result in name resolution through DNS. If the name is not associated with any address its timeout can cause severe slowdown.

- **REQUEST_HTTP_HOST_PORT**
  Port number in entity "Host"
- **REQUEST_HTTP_COOKIES_LIST**
  List of cookies names separated by ";"
- **REQUEST_CLIENT_ADDRESS**
  TCP client address
- **RESPONSE_ENDPOINT_ADDRESS**
  TCP endpoint address
- **REQUEST_INCOMING_COUNTRY**
  Country state (ISO 3166 2-letter code) or ".." "ZZ"
- **ENTITY_REQUEST**
  Return the value of request entity name or empty if no entity found
- **ENTITY_RESPONSE**
  Return the value of response entity name or empty if no entity found
- **SSL_CONNECTION_CLIENT**
  True if the client connects with SSL through LBL
- **SSL_CONNECTION_ENDPOINT**
  True if endpoint connects with SSL through LBL
- **SSL_CONNECTION_REENCRYPTION**
  True if SSL reencryption is performed (LBL terminates SSL and connects to the endpoint in SSL)
- **REQUEST_INCOMING_ADDRESS**

Local address on which the service request was accepted

- **REQUEST_INCOMING_HOST_NAME**
  Host name or local address on which the service request was accepted

---

ATTENTION: Use of REQUEST_HTTP_HOST_NAME can result in name resolution through DNS. If the name is not associated with any address its timeout can cause severe slowdown.

- **REQUEST_HTTP_SCHEME**
  http or https depending on the client's connection to LBL
- **HIGH_WATER_YELLOW_WARNING_REACHED**
  If true the Yellow Warning threshold has been exceeded. Indicates a significant load but still not critical
- **HIGH_WATER**
  Number of connection requests in the queue, numerical format
- **HIGH_WATER_LEVEL**
  Float value, % Of connection requests in the queue compared to the number of tunnels in current settings.
- **TUNNEL_SESSIONS_ACTIVE**
  Simultaneous active tunnels, numerical format
- **TUNNEL_SESSIONS_COMMITTED**
  Simultaneous tunnels committed, (subset of TUNNEL_SESSIONS_ACTIVE)
- **ACTUAL_TUNNEL_SESSIONS_SIZE**
  The actual size of tunnels: (usually equal to"MAX_TUNNEL_SESSIONS_SIZE")
- **MAX_TUNNEL_SESSIONS _SIZE**
  Maximum number of tunnels set.

NOTE: These ‘pre-loaded” values to be used as modifiers (% xxx%) must be loaded into a local variable to the rule.

ENTITY
ENTITY = loading of the variable described in varName with the value of the HTTP HEADER Entity whose name is indicated in name.

URI_PARAM
URI_PARAM = loading of the variable described in varName with the parameter value or query string of the HTTP HEADER whose name is indicated in name.

URI_PARAM_DECODED
URI_PARAM = loading of the variable described in varName with the parameter value or query string of the HTTP HEADER whose name is indicated in name.

CONSTANT
CONSTANT = loading of the variable described in varName with the parameter value indicated in name. Only in this case may the value in name consist of another variable previously loaded.

COOKIE

COOKIE = loading of the variable described in varName with the value of the HTTP HEADER Cookie whose name is indicated in name.

VARIABLE
VARIABLE = loading of the variable described in varName with the value of another variable whose name is indicated in name.

<regexTag></regexTag>

```
<serviceconf>
    <iproxy>
        <rewriteManagement>
            <rewriteHeaderRule>
                <conditions>
                    <cond
                        <regexTag></regexTag>
```

The <regexTag> section is not mandatory but unique if present in the <cond> section and is assessed with regular expressions to verify the condition.

In particular:
<regexTag>
Regular expression applied in the extracted value (Supports % VARIABLE% and caseSensitive)
</regexTag>

<numOperatorTag></numOperatorTag>

```
<serviceconf>
    <iproxy>
        <rewriteManagement>
            <rewriteHeaderRule>
                <conditions>
                    <cond
                        <numOperatorTag></numOperatorTag>
```

The <numOperatorTag> sub-section is not mandatory but unique at the <cond> section level and is assessed with numeric expressions for comparison.

This section was introduced to facilitate numerical comparisons and is used in place of the section <regexTag>.

The new section can have the following operators:

- **eq**=equal
- **neq**=not equal
- **gt**=greater than
- **geq**=greater than or equal to
- **lt**=less than
- **leq**=less than or equal to

Example <numOperatorTag>:

Following is an example where two conditions are verified. The first condition tests INNERVAR "HIGH_WATER" with a greater than or equal to 10 operator, the second condition verifies that the requested URIPath is different from /training.

```
<rewriteHeaderRule flow="REQUEST"  name="VIPiRedCarpetXMLRuleDynamic"
caseSensitive="false">
        <conditions>
                <cond from="INNERVAR" name="HIGH_WATER">
                        <numOperatorTag>geq 10</numOperatorTag>
                </cond>
                <cond from="INNERVAR" name="REQUEST_HTTP_URI_PATH" eval="NOT">
                        <regexTag>^/trainingw</regexTag>
                </cond>
        </conditions>
        <connectionToCut connectionToCut="true"/>
</rewriteHeaderRule>
```

The action <connectionToCut> from this example may be replaced by the actions:

<displaceEndPointsGrouping> or <entities> or <redirectTo>.

Example <conditions>

```
<conditions operator="AND">
   <cond enable="false" from="COOKIE" name="LBLSESSIONID" eval="NOT">
        <regexTag></regexTag>
 </cond>
   <cond from="VARIABLE" name="MY_IS" caseSensitive="true">
        <regexTag>is</regexTag>
 </cond>
  <cond from="INNERVAR" name="REQUEST_HTTP_HOST_NAME">
        <regexTag>localhost</regexTag>
 </cond>
  <cond from="INNERVAR" name="REQUEST_CLIENT_ADDRESS">
        <regexTag>127.0.0.1</regexTag>
 </cond>
  <cond from="INNERVAR" name="REQUEST_HTTP_URL" caseSensitive="true">
        <regexTag>^/viewProcessProperties.html\?process=A05_LBLGoDNSManager$</regexTag>
 </cond>
  <cond from="INNERVAR" name="REQUEST_HTTP_URI_PATH">
        <regexTag>^/viewProcessProperties.html$</regexTag>
  </cond>
  <cond from="ENTITY" name="Connection">
        <regexTag>Keep-alive</regexTag>
 </cond>
  <cond from="URI_PARAM" name="process">
        <regexTag>A05_LBLGoDNSManager</regexTag>
 </cond>
</conditions>
```

<entities>
<entity>

```
        <serviceconf>
```

```
<iproxy>
    <rewriteManagement>
        <rewriteHeaderRule>
            <entities>
                <entity
```

In this optional section the HEADER can be modified in any aspect if the conditions have ensured consistency.

In particular, this section can add and modify the initial request/response, as well delete every single entity (entity is a HEADER line divided into "name: value").

enable=: default value=”true”
Enables or disables this section.

entityName=: default value=””
Name of the entity to use (e.g.: "Connection", "Content-Type", "Accept-Encoding").

In the HEADER below the first line identifies the request, while the lines that follow identify the Entities.

GET /favicon.ico HTTP/1.1
Host: www.oplon.net
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 6.0; it; rv:1.9.2.3) Gecko/20100401 Firefox/3.6.3 (.NET CLR 3.5.30729)
Accept: image/png,image/*;q=0.8,*/*;q=0.5
Accept-Language: it-it,it;q=0.8,en-us;q=0.5,en;q=0.3
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 115
Connection: keep-alive

To make changes to the first line (for both REQUEST and RESPONSE) simply set:
entityName = "FIRST-LINE"

action=: default value=”” values: remove|change|add
The action to be taken on the entity.
It is important to note that:

- "add" does not remove an existing entity
- "change" edits the value of the entity through regular expressions starting with existing value or with a completely new value introduced by the "value" parameter.

value=: default value=””
If existent replaces the value of the entity (Supports the %VARIABLE% modifier).

caseSensitive=:default value=”caseSensitive value of the <rewriteHeaderRule>
This value describes the matching method for the rewriting rules (regular expressions).

Example <entities>

```
<entities>
  <entity entityName="My-Entity0000" action="add" caseSensitive="false"
          value="%MY_VAR_FROM_ENTITY% localhost %MY_REQUEST_HTTP_HOST_PORT% requestHostName
%MY_REQUEST_HTTP_HOST_NAME%">
          <regexTag>localhost</regexTag>
          <replaceTo>127.0.0.1</replaceTo>
</entity>

<entity enable="false" entityName="REQUEST-LINE" action="change">
          <regexTag>HTTP/1.1</regexTag>
          <replaceTo>HTTP/1.0</replaceTo>
</entity>

<entity enable="false" entityName="Connection" action="change">
          <regexTag>Keep-alive</regexTag>
          <replaceTo>Close</replaceTo>
</entity>

<entity entityName="Host" action="change">
          <regexTag>localhost</regexTag>
          <replaceTo>127.0.0.1</replaceTo>
</entity>

<entity entityName="My-Entity0001" action="add"
          value="%MY_VAR_FROM_ENTITY% localhost %MY_REQUEST_HTTP_HOST_PORT% requestHostName
%MY_REQUEST_HTTP_HOST_NAME%001">
          <regexTag>localhost</regexTag>
          <replaceTo>127.0.0.1</replaceTo>
</entity>

<entity entityName="My-Entity0002" action="add"
          value="%MY_VAR_FROM_ENTITY% localhost %MY_REQUEST_HTTP_HOST_PORT% requestHostName
%MY_REQUEST_HTTP_HOST_NAME%002">
          <regexTag>localhost</regexTag>
          <replaceTo>127.0.0.1</replaceTo>
</entity>

<entity entityName="My-Entity0003" action="add"
         value="%MY_VAR_FROM_ENTITY% localhost %MY_REQUEST_HTTP_HOST_PORT% requestHostName
%MY_REQUEST_HTTP_HOST_NAME%003">
          <regexTag>localhost</regexTag>
          <replaceTo>127.0.0.1</replaceTo>
</entity>

<entity entityName="My-Entity0000" action="remove"/>
</entities>
```

<redirectTo>

```
        <serviceconf>
                <iproxy>
                        <rewriteManagement>
                                <rewriteHeaderRule>
                                        <redirectTo
```

In case redirect of the request is desired, it is possible to insert this section, not mandatory and unique with the URL value of the redirection target. This section allows use of variables to construct the destination address thus using the same request to build a new URL.

enable=: default value=”true”
Enables or disables this section.

responseCode=: default value=”302”
The response code that performs the redirection. If <= 0 the default value of 302 is used. The possible values for use are normally 301 and 302.

redirectURL=: default value=””
URL value of redirect target. This value can contain variables that will be substituted (Supports the % VARIABLE% modifier).

Example:

```
<redirectTo responseCode="302"
redirectURL="http://www.oplon.net/?http://%MY_REQUEST_HTTP_HOST_NAME% "/>
```

<displaceEndPointsGrouping>

```
<serviceconf>
    <iproxy>
        <rewriteManagement>
            <rewriteHeaderRule>
                <displaceEndPointsGrouping
```

With this directive a request from a listener related to another EndPointsGrouping can be displaced to a different EndPointsGrouping.

This functionality is achieved by using the rewriting rules of the Header with a new section that indicates the displacement.

The simplest form follows:

```
<rewriteHeaderRule enable="true" flow="REQUEST" name="setEndpGrouping">
    <displaceEndPointsGrouping enable="true" endPointsGrouping="services001"/>
</rewriteHeaderRule>
```

The <displaceEndPointsGrouping> section indicates where the request should refer to as endPointsGrouping.

It is possible to use all the forms already provided to compose the displacement such as variables and conditions:

```
<rewriteHeaderRule enable="true" flow="REQUEST" name="setEndpGrouping">
    <variables>
        <var varName="DISPLACE_TO_GROUP" name="services001" from="CONSTANT"/>
    </variables>
    <displaceEndPointsGrouping enable="true" endPointsGrouping="%DISPLACE_TO_GROUP%"/>
</rewriteHeaderRule>
```

It is possible to control the displacement from Java HEADER interceptor classes:

```
<rewriteHeaderRule enable="false" flow="REQUEST" name="setEndpGrouping"
                  httpInterceptorClass="my_httprewriters.LBLDisplaceEndPointGroupingTemplate">
</rewriteHeaderRule>
```

```
import loadbalancer.rewriter.LBLHTTPInterceptorHeaderAbstr;
import loadbalancer.rewriter.LBLHTTPInterceptorHeaderStreamFragment;

/**
 * Template interceptor class for displace EndPointsGrouping
 * @author TCOProject(r)
 * @version 1.0 Created on 12-feb-2011, 17.04.05
 */
public class LBLDisplaceEndPointGroupingTemplate extends LBLHTTPInterceptorHeaderAbstr
{
        /** copyright */
        public static final String COPYRIGHT="LBL and TCOProject are trademarks of F.Pieretti";

      @Override
      public  void doRequestHeaderBeforeReplace(LBLHTTPInterceptorHeaderStreamFragment streamFragment) {
            streamFragment.setEndPointsGrouping("services001");
      }

      @Override
      public  void doRequestHeaderAfterReplace(LBLHTTPInterceptorHeaderStreamFragment streamFragment) {
      }

      @Override
      public  void doResponseHeaderBeforeReplace(LBLHTTPInterceptorHeaderStreamFragment streamFragment) {
      }

      @Override
      public  void doResponseHeaderAfterReplace(LBLHTTPInterceptorHeaderStreamFragment streamFragment) {
      }
}
```

Example of Header rewriting:

This example modifies redirect requests from the server that are in https protocol and sends them to the same address using the http protocol.

```
<rewriteHeaderRule enable="true" flow="RESPONSE" name="test003" caseSensitive="false"
                        httpInterceptorClass="loadbalancer.rewriter.LBLHTTPRewriteInterceptorHeaderLogging"
                        responseCode="302">
          <conditions>
                    <cond from="ENTITY" name="Location">
                              <regexTag>^https://www.tcoproject.dev:8085</regexTag>
                    </cond>
                    <cond from="INNERVAR" name="REQUEST_HTTP_HOST_NAME">
                              <regexTag>www.tcoproject.dev</regexTag>
                    </cond>
          </conditions>
          <entities>
                    <entity entityName="Location" action="change">
                              <regexTag>^https</regexTag>
                              <replaceTo>http</replaceTo>
                    </entity>
          </entities>
</rewriteHeaderRule>
<rewriteHeaderRule enable="true" flow="RESPONSE" name="test004" caseSensitive="false"
                        httpInterceptorClass="loadbalancer.rewriter.LBLHTTPRewriteInterceptorHeaderLogging"
                        responseCode="301">
```

```
        <conditions>
                <cond from="ENTITY" name="Location">
                        <regexTag>^https://www.tcoproject.dev:8085</regexTag>
                </cond>
                <cond from="INNERVAR" name="REQUEST_HTTP_HOST_NAME">
                        <regexTag>www.tcoproject.dev</regexTag>
                </cond>
                </conditions>
                <entities>
                        <entity entityName="Location" action="change">
                                <regexTag>^https</regexTag>
                                <replaceTo>http</replaceTo>
                        </entity>
                </entities>
</rewriteHeaderRule>
```

<connectionToCut>

```
        <serviceconf>
                <iproxy>
                        <rewriteManagement>
                                <rewriteHeaderRule>
                                        <connectionToCut
```

This section is dedicated to the Quality Of Application Service. It is possible through this section to arbitrarily cut the connection and to perform, depending on the case, a closure of tunnels, a redirect or display the courtesy page as described in the DoS Attack prevention functions.

Following are the templates to make us of this section using both XML rules as well as interceptor classes.

The feature is available with the DoS Attack prevention module enabled accompanied by the appropriate license.

The use of this feature is described by some examples where, upon verification of programmable conditions, the cutting of connections is activated.
For details see section <numOperatorTag>.

Following is an example where two conditions are verified. The first condition tests the INNERVAR "HIGH_WATER" with a greater than or equal to 10 operator, the second condition verifies that the requested URIPath is different than /training.

```
<rewriteHeaderRule flow="REQUEST"  name="VIPiRedCarpetXMLRuleDynamic" caseSensitive="false">
        <conditions>
                <cond from="INNERVAR" name="HIGH_WATER">
                        <numOperatorTag>geq 10</numOperatorTag>
                </cond>
                <cond from="INNERVAR" name="REQUEST_HTTP_URI_PATH" eval="NOT">
                        <regexTag>^/trainingw</regexTag>
                </cond>
        </conditions>
        <connectionToCut connectionToCut="true"/>
</rewriteHeaderRule>
```

This function can also be executed by interceptor class as shown below. The end result is the

same as that expressed in XML form.

XML rule:

```
<rewriteHeaderRule flow="REQUEST"  name="VIPiRedCarpetInterceptorDynamic" caseSensitive="false"
   httpInterceptorClass="rewriteclasses.LBLHTTPRewriteInterceptorHeaderVIPiRedCarpetDynamic">
   <variables>
          <var varName="LOG_INFO" name="true" from="CONSTANT"/>
          <var varName="MY_THRESHOLD" name="10" from="CONSTANT"/>
  </variables>
</rewriteHeaderRule>
```

Interceptor class:

```
public class LBLHTTPRewriteInterceptorHeaderVIPiRedCarpetDynamic extends LBLHTTPInterceptorHeaderAbstr {
   /** copyright */
   public static final String COPYRIGHT="LBL and TCOProject are trademarks all rights reserved";
  @Override
  public void doRequestHeaderBeforeReplace(LBLHTTPInterceptorHeaderStreamFragment streamFragment) {
          String sLogInfo = streamFragment.getVariable("LOG_INFO");
          boolean logInfo = Boolean.valueOf(sLogInfo!=null ? sLogInfo.toLowerCase() : "false");
          String sMyThreshold = streamFragment.getVariable("MY_THRESHOLD");
          long myThreshold = Long.valueOf(sMyThreshold==null ? "-1": sMyThreshold);
          if (streamFragment.getHighWater() > myThreshold) {
                  if (streamFragment.getHTTPRequestURIPath().startsWith("/trainingw"))
                          streamFragment.setConnectionToCut(false);
                  else
                          streamFragment.setConnectionToCut(true);
          }
   }

   @Override
    public void doRequestHeaderAfterReplace(LBLHTTPInterceptorHeaderStreamFragment streamFragment) {
    }

    @Override
    public void doResponseHeaderBeforeReplace(LBLHTTPInterceptorHeaderStreamFragment streamFragment) {
    }

   @Override
    public void doResponseHeaderAfterReplace(LBLHTTPInterceptorHeaderStreamFragment streamFragment) {
    }
```

<rewriteBodyRule>

```
        <serviceconf>
                <iproxy>
                        <rewriteManagement>
                                <rewriteBodyRule
```

The <rewriteBodyRule> section describes a rewriting rule applicable to the HTTP BODY. This section can be repeated as many times as the rules to be applied even for different contexts of forwarding. The parameters for this section describe the characteristics of the rule to be expressed and base conditions.

name=: default value=”null”
The name of the rule.
The name of the rule. This value, associated with one or more domains <virtualDomain> up to the individual endpoint <endp>, allows application of the rule in multiple contexts. If the value has not been set the rule is not loaded.

enable=: default value=”true”
Enables or disables this rule.

flow=: default value=”BOTH” available values: REQUEST|RESPONSE|BOTH
Identification of the flow/stream in which to apply the rule.
If the REQUEST flow is selected, the rule is applied in this manner (incoming request):
client----> LBL---> endp.

If the RESPONSE flow is selected, the rule is applied in this manner (outgoing response):
client-<---LBL <---endp.

If BOTH is selected, the rule is apllied for both incoming requests from clients and outgoing responses from the services to the client.

httpMethod=: default value=”” available values: Methods HTTP: POST, GET...
In REQUEST streams, identifies the HTTP method used
e.g. POST GET PATCH
If not set or empty value is performed on all methods.
In cases of flow = "BOTH", this value will be verified only in the REQUEST streams and thus can coexist with the responseCode parameter.

responseCode=: default value=”-1”
In RESPONSE streams, identifies the HTTP return code and for which the rule will be applied. In cases of flow = "BOTH", this value will be verified only in the RESPONSE streams and thus can coexist with the httpMethod parameter.

caseSensitive=: default value=”false”
This value describes the matching method for the rewriting rules (regular expressions). This value is propagated to its sub-sections where it can nevertheless be changed for required exceptions.

inspectionOnly =: default value = "false"
If true, it is declared that the rule does not alter the body content and therefore the body type will not be changed to "Transfer-Encoding: chunked" to allow its modification on-the-fly. This determines the use of inspection body classes even on application servers that do not implement chunked POST.

charset=: default value=”JVM default charset”
The attributes hierarchy of the charset during the rewriting is as follows:

- Checks for an indication of the charset to use in the "Content-Type: text/html; charset = iso-8859-1 " entity
- If the charset is not found in the Content-Type entity, the definition of the charset is taken from section <rewriteBodyRule ... charset="iso-8859-1">
- If the <rewriteBodyRule ...="" charset="iso-8859-1"> parameter was not specified, the charset of the user that launche the JVM will be used.

The charset for content rewriting is acquired from the HTTP HEADER from the "Content-

Type: text/html; charset = iso-8859-1" entity. If the “Content-Type” entity does not contain the designation of the charset it is possible to force the charset via the "charset" parameter in the rewriting rule in section <rewriteBodyRule> as follows.

```
<rewriteBodyRule flow="RESPONSE" name="ChangeValues" charset="iso-8859-1">
   <requestURLMatches>/TestCharset.html(.*)</requestURLMatches>
   <mimeType value="text/html" fragmentClose="&gt;" fragmentOpen="&lt;" enable="true"/>
   <regexTag>di</regexTag>
   <replaceTo>il</replaceTo>
</rewriteBodyRule>
```

If the charset value is not specified in the <rewriteBodyRule> section, the default value will be the charset of the JVM that was started.

When the JVM is started it acquires the platform’s default charset per the user that executed the start. To change the JVM's default charset simply set the following parameter in the launch profile:

-Dfile.encoding=UTF-8 A10_LBLGo.xml

```
 <exec>java -Xrs -server -Xss128k -Xms512m -Xmx1024m -DLBL_INTERACTIVE_CMD=true -
DLBL_RUNLEVEL=1 -Dfile.encoding=UTF-8 loadbalancer.starter.LBLServerStarterApp</exec>
 <logDirFiles>lib\logs</logDirFiles>
```

httpInterceptorClass=: default value=””

This parameter specifies a class that implements the class:

- loadbalancer.rewriter.LBLHTTPInterceptorBodyAbstr

This is made available from the OPLON® LoadBalancer platform to take action on the BODY both before and after the application of the rules described in this section.

LBLHTTPInterceptorBodyAbstr

The "loadbalancer.rewriter.LBLHTTPInterceptorBodyAbstr" class provides 4 methods that identify four key situations of the data stream.

```
package loadbalancer.rewriter;

import loadbalancer.rewriter.LBLHTTPInterceptorBodyAbstr;
import loadbalancer.rewriter.LBLHTTPInterceptorBodyStreamFragment;

/**
 * Test class HTTP BODY Interceptor for dynamic change during stream...
 * @version 1.0 Created on 5-jun-2010
 */
public class LBLHTTPRewriteInterceptorBodyLogginng extends LBLHTTPInterceptorBodyAbstr {
        /** copyright */
        public static final String COPYRIGHT="LBL and TCOProject are trademarks all rights reserved";

        /**
         * Metodo richiamato alla richiesta del client prima di effettuare le modifiche ad opera delle espressioni regolari
         * @param streamFragment frammento consistente del body (HTML/CSS/JS etc)
         */
        @Override
        public void doRequestBodyBeforeReplace(LBLHTTPInterceptorBodyStreamFragment streamFragment) {
                logWarning("REQUEST BODY BEFORE REPLACE\n"+streamFragment.getStreamFragment());
```

```
                for (String varName: streamFragment.getVariables())
                        logWarning("RQBR BODY VarName:"+varName+" value:"+streamFragment.getVariable(varName));
        }

        /**
         * Metodo richiamato alla richiesta del client dopo aver effettuato le modifiche ad opera delle espressioni regolari
         * @param streamFragment frammento consistente del body (HTML/CSS/JS etc)
         */
        @Override
        public void doRequestBodyAfterReplace(LBLHTTPInterceptorBodyStreamFragment streamFragment) {
                logWarning("REQUEST BODY AFTER REPLACE\n"+streamFragment.getStreamFragment());
        }

        /**
         * Metodo richiamato al response del servizio prima di effettuare le modifiche ad opera delle espressioni regolari
         * @param streamFragment frammento consistente del body (HTML/CSS/JS etc)
         */
        @Override
        public void doResponseBodyBeforeReplace(LBLHTTPInterceptorBodyStreamFragment streamFragment) {
                logWarning("RESPONSE BODY BEFORE REPLACE\n"+streamFragment.getStreamFragment());
                for (String varName: streamFragment.getVariables())
                        logWarning("REBR BODY VarName:"+varName+" value:"+streamFragment.getVariable(varName));
        }

        /**
         * Metodo richiamato al response del servizio dopo aver effettuato le modifiche ad opera delle espressioni regolari
         * @param streamFragment frammento consistente del body (HTML/CSS/JS etc)
         */
        @Override
        public void doResponseBodyAfterReplace(LBLHTTPInterceptorBodyStreamFragment streamFragment) {
                logWarning("RESPONSE BODY AFTER REPLACE\n"+streamFragment.getStreamFragment());
        }
}
```

Each method provides the LBLHTTPInterceptorBodyStreamFragment object with which actions can be taken on the data fragment during the flow.

LBLHTTPInterceptorBodyStreamFragment already contains many methods for acting on the HTTP BODY fragment both for query of values as well as controlled modifications. A full dissertation with examples can be found in the manual "OPLON® LoadBalancer Content Rewriting".

LBLHTTPInterceptorBodyAbstr is the class from which the interceptor classes are derived. It contains some utility methods for the most common operations such as processing digital certificates in different formats and/or verification of significant values such as the fingerprint or the serial number.

The following is a detailed list of methods of classes:
LBLHTTPInterceptorHeaderStreamFragment
LBLHTTPInterceptorBodyAbstr[1].

LBLHTTPInterceptorBodyStreamFragment methods

```
/**
 * return client host address
 * @return client host address or null  if not found
 */
public String getRequestClientAddress()

/**
 * return endpoint host address
```

1 For the mehods refer to LBLHTTPInterceptorHeaderAbstr

```
 * @return endpoint host address or null  if not found
 */
public String getResponseEndpointAddress()

/**
 * Vector with cookies names
 * @return Vector with cookies names
 */
public Vector<String> getHTTPRequestCookiesNames()

/**
 * Get a cookie value
 * @param cookieName cookie name to find
 * @return cookie value or null if not found
 */
public String getHTTPRequestCookie(String cookieName)

/**
 * get request URL with params and query string
 * @return URL with params and query string or null if not found
 */
public String getHTTPRequestURL()

/**
 * get request URIPath without params and query string
 * @return URIPath without params and query string or null if not found
 */
public String getHTTPRequestURIPath()

/**
 * get request URL last element
 * @return URL last element
 */
public String getHTTPRequestURLLastElement()

/**
 * get host name of request
 * @return host name of request, null if not found
 */
public String getHTTPRequestHostName()

/**
 * get port number of request
 * @return port number of request, -1 if not found
 */
public int getHTTPRequestHostPort()

/**
 * get http request method
 * @return request method, null if not found
 */
public String getHTTPRequestMethod()

/**
 * get http version
 * @return 10 for HTTP 1.0, 11 for HTTP 1.1. -1 if not found
 */
public int getHTTPVersion()

/**
 * get parameter o query string in request
 * @param parameterName parameter name to find
 * @return parameter value or null if not found
 */
```

```
public String getHTTPRequestParam(String parameterName)

/**
 * get parameter o query string in a url
 * @param url url to find parameter
 * @param parameterName parameter name to find
 * @return parameter value or null if not found
 */
public String getHTTPUrlParam(String url, String parameterName)

/**
 * get response code
 * @return response code or -1 if not found
 */
public int getHTTPResponseCode()


// ********************************************************
// ENTITIES
// ********************************************************

/**
 * get entity value
 * @param entityName entity name es.: Content-Type
 * @return entity value or null if not match
 */
public String getHTTPEntity(String entityName)

/**
 * get entities
 * @return a vector of strings array where element[0] is enttity name (name in lowercase) and element[1] is entity value
 */
public Vector<String[]> getHTTPEntities() {

// ************************************************************************
// VARIABLES
// ************************************************************************

/**
 * Add variable
 * @param name variable name
 * @param value variable value
 */
public void addVariable(String name, String value)

/**
 * Get a variable value
 * @param name variable name
 * @return variable value or null
 */
public String getVariable(String name)

/**
 * Replace all variable with their values
 * @param stringToReplace string to replace
 * @return string replaced
 */
public String replaceStringWithVariable(String stringToReplace)

/**
 * Get a list of variables names
 * @return list of variables names
 */
public Vector<String> getVariables()
```

```
// ========================================
// GETTER
// ========================================

/**
 * @return the contentType
 */
public TCONameValue getContentType()

/**
 * @return the endPointGroup
 */
public LBLEndPointGroup getEndPointGroup()
/**
 * get all parameters names from body
 * @return all parameters names from body
 */
public Vector<String> getHTTPAllParametersNamesFromBody()

/**
 * get parameter from body
 * @param parameterName parameter name to find
 * @return parameter value or null if not found
 */
public String getHTTPParameterFromBody(String parameterName)

/**
 * add new parameter in the body
 * @param parameterName parameter name
 * @param parameterValue parameter value
 */
public void addParameterInTheBody(String parameterName, String parameterValue)

// ************************************************************************
// STREAM
// ************************************************************************

/**
 * @return the streamFragment
 */
public byte[] getStreamFragment()

/**
 * @param streamFragment the streamFragment to set
 */
public void setStreamFragment(byte[] streamFragment)
```

ATTENTION

Some application servers do not support the POST method with "transfer-encoding: chunked". For example Apache Tomcat has resolved this issue as of version 6.0.26 but previous versions (eg.:6.0.14) do not work.

Verify that the application server is compliant with the recommended: HTTP1.1 transfer-encoding: chunked

<requestURLMatches></requestURLMatches>

```
<serviceconf>
    <iproxy>
        <rewriteManagement>
            <rewriteBodyRule>
                <requestURLMatches></requestURLMatches>
```

The value inside the <requestURLMatches> section identifies a regular expression. This is used to verify the conformity with the URL request, including the URI Params and Query String. If this condition is satisfied the rule is applied. If this section is not given a value or is non-existent, all URL requests will be considered valid. The value of the URL request is formed without UTF-8 encoding (spaces are represented by spaces and not by their URL encoded representation).

```
<requestURLMatches>^/viewProcessProperties.html\?process=A05_LBLGoDNSManager$</requestURLMatches>
```

<mimeType>

```
<serviceconf>
    <iproxy>
        <rewriteManagement>
            <rewriteBodyRule>
                <mimeType
```

This section, which may be repeated, is another condition for which the rule may be applied to the stream.

In the case of the BODY, unlike the treatment of the HEADER, the mimeType assumes a great importance because in addition to defining the mime type on which to apply the rule it also defines the fragmentation rules for the body.

In fact the BODY may be substantial in size and it is therefore necessary to fragment it into many "buffers" (called chunks) to transfer it through the balancer. The changes executed by the rewriter must be made consistently even when a value to change is between two "chunks".

OPLON® LoadBalancer is able to recompose a fragment in with its appropriate parts given the rules for that particular mime type.

For text/html, for example, the characters “<” e “>” distinguish the consistency of a block. For text/css the braces “{“ e “}” distinguish a consistent block.

enable=: default value=”true”
Enables or disables this section.

value=: default value=””
MimeType value with which it is possible to apply the rule, e.g. "text/html" or "text/css" etc.

fragmentOpen=: default value=””
This parameter, together with the fragmentClose parameter, is very important to determine a

consistent fragment of data. The rewriting of the body takes place using consistent fragments and not the entire body but rather a forwarding "buffer" of considerable length. Being an xml notation some characters must be substituted with their tags, e.g.: for “>” “<” the equivalents are “>” and “<” for the “&” the substitute is “&”

fragmentClose=: default value=””
This parameter, together with the fragmentOpen parameter, is very important to determine a consistent fragment of data. The rewriting of the body takes place using consistent fragments and not the entire body but rather a forwarding "buffer" of considerable length. Being an xml notation some characters must be substituted with their tags, e.g.: for “>” “<” the equivalents are “>” and “<” for the “&” the substitute is “&”

eg.:

```
<mimeType enable="true" value="text/html" fragmentOpen="&lt;" fragmentClose="&gt;"/>
<mimeType enable="true" value="text/css" fragmentOpen="{" fragmentClose="}"/>
<mimeType enable="true" value="text/plain" fragmentOpen=" " fragmentClose=" "/>
<mimeType enable="true" value="application/json" fragmentOpen="{" fragmentClose="}"/>
```

If fragmentOpen or fragmentClose are not set, no fragmentation will be performed and the frame will be available without logic. This is very useful in cases where for example you want to perform bulk operations such as compression..

<variables>

```
<serviceconf>
    <iproxy>
        <rewriteManagement>
            <rewriteBodyRule>
                <variables>
```

See <variables> of the <rewriteHeaderRule> section.

<conditions>

```
<serviceconf>
    <iproxy>
        <rewriteManagement>
            <rewriteBodyRule>
                <conditions>
```

See <conditions> of the <rewriteHeaderRule> section.

<regexTag></regexTag>

See <replaceTo>.

<replaceTo></replaceTo>

```
<serviceconf>
    <iproxy>
        <rewriteManagement>
            <rewriteBodyRule>
                <regexTag></regexTag>
                <replaceTo></replaceTo>
```

The <regexTag> and <replaceTo> sections are not mandatory but unique if present in the

section level and are assessed with regular expressions to apply changes to the content of the BODY.
These two values can also contain further modifying variables expressed in previous <var> sections inherent of the same rewriting rule.

In particular:
<regexTag>
regular expression applied to the value retrieved (Supports %VARIABLE% and caseSensitive)
</regexTag>

<replaceTo>
Value with which to replace (Supports %VARIABLE% and caseSensitive)
</replaceTo>

Example of rewriting body

Rewriting of the body to change a URL reference from absolute to relative:

```
<rewriteBodyRule enable="true" flow="BOTH" name="tcoprojectDevToRelative" caseSensitive="false"
                                    httpInterceptorClass="testrewrite.HTTPRewriteInterceptorBodyTest">
        <mimeType enable="true" value="text/html" fragmentClose="&gt;" fragmentOpen="&lt;"/>
        <regexTag>(href|src)=\"(http|https)://www.tcoproject.dev/</regexTag>
        <replaceTo>$1=\"</replaceTo>
</rewriteBodyRule>
```

Change a value on a css:

```
<rewriteBodyRule enable="true" flow="BOTH" name="changeColor" caseSensitive="false"
                    httpInterceptorClass="testrewrite.HTTPRewriteInterceptorBodyTest"
                    httpMethod="GET"
                    responseCode="200">
        <requestURLMatches>styles.css</requestURLMatches>
        <mimeType enable="true" value="text/css" fragmentClose="}" fragmentOpen="{"/>
        <regexTag>255</regexTag>
        <replaceTo>100</replaceTo>
</rewriteBodyRule>
```

Change of descriptions within the body:

```
<rewriteBodyRule enable="true" flow="BOTH" name="LBL" caseSensitive="false">
        <mimeType enable="true" value="text/html" fragmentClose="&gt;" fragmentOpen="&lt;"/>
        <variables>
                <var varName="MY_VAR_FROM_THIS_VALUE" name="Renew page! " from="CONSTANT">
                </var>
        </variables>
        <regexTag>Reload frame</regexTag>
        <replaceTo>%MY_VAR_FROM_THIS_VALUE%</replaceTo>
</rewriteBodyRule>
```

<endpoints>

```
        <serviceconf>
                <iproxy>
```

```
                    <endpoints>
```

This section defines the list of services on the backend. Its syntax is:

<endpoints>
<endPointsGrouping>
<virtualDomain>
< /endp>
….
< /endp>
< /virtualDomain>
< /endPointsGrouping>
< /endpoints>

<endPointsGrouping>

```
<serviceconf>
<iproxy>
<endpoints>
<endPointsGrouping
```

groupName =: default value= "default"
The name of the groups to which the services belong.
This value, associated with one or more listeners, identifies a group of services for which to balance the traffic. It is possible to insert more groups separated by at least one space as in the example below.

```
<endPointsGrouping groupName=" services  test 001    test002  test003" enable= "true">
```

In this way, for example, it is possible to associate multiple listeners with different parameters but associated with the same services.

enable =: default value= "true"
Enables or disables this group of services.

protocol =: default value = ""
If not empty, the protocol set on the listener changes for entire the group.

redirectToHttps =: default value = "false"
If true, in the presence of layer 7 services, HTTP / S checks whether the request is in HTTP and redirects all the resources into HTTPS.

sniHostName=: valore di default=””
Host name used for the TLS SNI connections to the backend services.
It is used to balance in SSL backend services that implement the Server Name Indication Protocol. If you set the parameter is inherited in VirtualDomain and endp are contained in section endPointsGrouping. WARNING: If you use this value, the service address in endp tags must be entered with the numeric value (ipv4 or ipv6 notation) and not with the host name.

sniForwarding =: default value = "false"
If true, it allows SNI to set endpoints and to automatically forward the request hortname to the backend with the same value as the origin.

SSLApplicationProtocols=: default="" values:"h2 http/1.1 undef"
If true, use the chipersuite in the order specified for the SSL / TLS listeners

eg:
```
<params
        ...
        endPointSSLApplicationProtocols="h2 http/1.1 undef"
        endPointSSLUseCipherSuitesOrder="true"
        .../>

        <endPointsGrouping enable="true" SSLApplicationProtocols="h2 http/1.1 undef"
                SSLUseCipherSuitesOrder="true">
                <virtualDomain enable="true" SSLApplicationProtocols="h2 http/1.1 undef"
                        SSLUseCipherSuitesOrder="true">
                                <endp address="192.168.56.131" port="8080" uriPath="/"
                                        SSLApplicationProtocols="h2 http/1.1 undef"
                  SSLUseCipherSuitesOrder="true" enable="true"/>
```

SSLUseCipherSuitesOrder=: default="true"
If true, it uses the chip-suites in the order indicated for the SSL / TLS listeners

SSLProtocols =: default value = ""
The parameter sets the SSL protocols to be used for communication.

associateName =: default value=""
With this parameter symbolic names can be associated to the group of end-points in the form : "symbolic_name_a symbolic_name_b symbolic_name_c".

Each symbolic name must be separated by a space from another symbolic name. The symbolic name cannot contain spaces.
For a dissertation of associative names refer to: <params notificationDir="" />

loadBalancingType =: default value= "Adaptive"
Sets the policy of balancing.
Possible values:

- RoundRobin
- Adaptative
- Failover

idSessionsManagerName =: default value= "default"
Sets the type of recognition for the layer 7 HTTP/S session.
This value should be associated with a <idSessions> section. The value will be reported to all domains and groups of URIPath within this grouping if not indicated otherwise.

dosMaxConcurrentConnectionsReaction =: default value= "false"
Activates the DDoS congestion resolver capping service, indicated by the parameter maxConcurrentConnections in section <endp>.

NOTE The activation of this feature is subject to the presence of the specific license for DoS Attack Prevention.

realmLogin =: default value= "": default value=""
Sets the basic login authentication of the service.
This login and the password will be delivered to services 'endp', unless otherwise specified in the domain or the 'endp', such as HTTP entity through the form e.g.

- authorization: Basic dXNyMjp1c3Iy

realmPassword =: default value=""
Sets the password for basic authentication of the service.
This password and the login will be delivered to services 'endp', unless otherwise specified in the domain or the 'endp', such as HTTP entity through the form e.g.

- authorization: Basic dXNyMjp1c3Iy

rewriteHeaderRules =: default value=""
List of names of the rewriting rules for the HTTP HEADERS (layer 7 HTTP/S) to apply to the group of endpoints. It is possible to indicate multiple rules separated by one or more spaces.

Ex.

```
rewriteHeaderRules="redirSSLloginWhenNoSSL proxyTo"
```

The rules will be applied, if conditions permit, in sequence.

;**LAST**
For each rule name one can also indicate the parameter 'LAST' that in case the rule is performed determines stopping the application of the remaining rules of the sequence.
To set the value of the parameter the " ;" after the name and the name of the parameter.

e.g.:

```
rewriteHeaderRules="redirSSLloginWhenNoSSL;LAST proxyTo"
```

There must not be any spaces between the name and the parameters. In this case, if the rule redirSSLloginWhenNoSSL is applied the rule proxyTo will never run.

;ALWAYS
The parameter ALWAYS indicates that the rule is always performed.
To set the value of the parameter the " ;" after the name and the name of the parameter.

e.g.:

```
rewriteheaderrules= "redirssllogінwhennossl;LAST proxyto;ALWAYS"
```

In this case the rule proxyto is performed regardless of the execution of the rule redirssllogінwhennossl.

;NOP

The NOP parameter indicates that the rule must not be carried out

e.g.
rewriteheaderrules= "redirssllloginwhennossl;NOP proxyto;ALWAYS"

In this case the rule redirssllloginwhennossl is not performed. The parameter NOP is useful to exclude the execution of general rules.

rewriteBodyRules =: default value=""
List of names of the rewriting rules for the HTTP BODY (layer 7 HTTP/S) to apply to the group of endpoints. It is possible to indicate multiple rules separated by one or more spaces.

Ex.
```
rewriteBodyRules="addTrademarkParam absoluteToRelative echoRewriteBody"
```

The rules will be applied, if conditions permit, in sequence.

;**LAST**
For each rule name one can also indicate the parameter 'LAST' that in case the rule is performed determines stopping the application of the remaining rules of the sequence.
To set the value of the parameter place a " ;" (semi-colon) after the name and the name of the parameter.

Ex.
```
rewriteBodyRules="addTrademarkParam;LAST absoluteToRelative echoRewriteBody"
```

There must not be any spaces between the name and the parameters.. In this case, if the rule addTrademarkParam is applied the rules absoluteToRelative echoRewriteBody will never be performed.

;ALWAYS
The parameter ALWAYS indicates that the rule is always performed.rewritebodyrules= "addtrademarkparam;LAST absolutetorelative echorewritebody;ALWAYS"

In this case the rule echorewritebody is performed regardless of the execution of the rule addtrademarkparam.

;NOP
The NOP parameter indicates that the rule must not be carried out rewritebodyrules= "addtrademarkparam;LAST absolutetorelative;NOP echorewritebody;ALWAYS"

In this case the rule absolutetorelative is not performed. The parameter NOP is useful to exclude the execution of general rules.

cMessage =: default value= "messageNoEndPoint.html"
Name of the html file containing the appropriately contextual courtesy message.
At the time of loading parameters, at the start of the balancing process, existence of the file is verified in the directory:

- (LBL_HOME)/procsProfiles/A10_LBLGo/resources/html/messageName.html

If the file does not exist, it is reported at the time of startup and assigned the default value:

- (LBL_HOME)/procsProfiles/A10_LBLGo/resources/html/messageNoEndPoint.html

redirectNoHostsFound=: default value="" UM=URL
For Layer 7 HTTP/S (in case of hosts not available for the grouping "endPointsGrouping-virtualDomain-uriPath ") it is possible to execute a redirect to another URL as an alternative to the courtesy message. If this parameter is not set or the value “”, the courtesy message will be used.

needClientCert =: default value= "false" UM=boolean
For Layer 7 HTTP/S this value imposes on the whole grouping of domains and endpoints use of the client certificate to access associated services. Domains and endpoints can change this value in case of exceptions.

needClientCertMessage =: default value= "messageNeedClientCert.html"
Name of the html file containing the appropriately contextual courtesy message.
At the time of loading parameters, at the start of the balancing process, existence of the file is verified in the directory:

- (LBL_HOME)/procsProfiles/A10_LBLGo/resources/html/messageName.html

If the file does not exist, it is reported at the time of startup and assigned the default value:

- (LBL_HOME)/procsProfiles/A10_LBLGo/resources/html/ messageNeedClientCert.html

SSLCert =: default value=""
The reference to the digital certificate to be used as "client certificate" from OPLON®Application Delivery Controller toward endpoints of this group.

<virtualDomain>

```
<serviceconf>
    <iproxy>
        <endpoints>
            <endPointsGrouping>
                <virtualDomain
```

enable =: default value= "true"
Enables or disables this group of services

virtualDomainName =: default value= "default"
The name of the virtual domains to which the services belong.
This value identifies a group of homogeneous services on which to balance the traffic. It is possible to insert multiple domain names associating them to the same endpoint services as in the example below:

```
<virtualDomain virtualDomainName="  localhost   www.tcoproject.dev pandorum  127.0.0.1" enable="true"
        rewriteHeaderRules...
```

It is also possible to indicate the name of the domain with a regular expression.

```
<endPointsGrouping enable="true">
                <virtualDomain virtualDomainName="www.pluto.com (.*)\.pippo\.com"
```

protocol =: default value = ""
If not empty, the protocol set on the listener changes for entire the group.

redirectToHttps =: default value = "false"
If true, in the presence of layer 7 services, HTTP / S checks whether the request is in HTTP and redirects all the resources into HTTPS.

sniHostName=: valore di default=”value in endPointsGrouping:=sniHostName”
Host name used for the TLS SNI connections to the backend services.
It is used to balance in SSL backend services that implement the Server Name Indication Protocol. If you set the parameter is inherited in endp are contained in section virtualDomain. WARNING: If you use this value, the service address in endp tags must be entered with the numeric value (ipv4 or ipv6 notation) and not with the host name.

sniForwarding =: default value = "false"
If true, it allows SNI to set endpoints and to automatically forward the request hortname to the backend with the same value as the origin.

SSLApplicationProtocols=: default=”” values:”h2 http/1.1 undef”
If true, use the chipersuite in the order specified for the SSL / TLS listeners

eg: <params

```
...
endPointSSLApplicationProtocols="h2 http/1.1 undef"
endPointSSLUseCipherSuitesOrder="true"
.../>

<endPointsGrouping enable="true" SSLApplicationProtocols="h2 http/1.1 undef"
        SSLUseCipherSuitesOrder="true">
        <virtualDomain enable="true" SSLApplicationProtocols="h2 http/1.1 undef"
                SSLUseCipherSuitesOrder="true">
                        <endp address="192.168.56.131" port="8080" uriPath="/"
                                SSLApplicationProtocols="h2 http/1.1 undef"
                SSLUseCipherSuitesOrder="true" enable="true"/>
```

SSLUseCipherSuitesOrder=: default=”true”
If true, it uses the chip-suites in the order indicated for the SSL / TLS listeners

SSLProtocols =: default value = ""
The parameter sets the SSL protocols to be used for communication.

portRewriting =: default value= "true"
If set to true rewrites the port number in the header from the endpoint with the value of the port requested by the client.
This behavior is necessary to balance backend services that respond by modifying the original header of the request with its own port number. With IIS this parameter must be set to true.

sslRewriting =: default value= "true"

If set to true verifies that the listener is in SSL and if so rewrites the protocol from http to https in the case of redirect from an http protocol service.
This feature is useful in case the listener acts as an SSL terminator and the backend services are not encrypted (transparent).

SSLCert =: default value= "endPointGrouping:=SSLCert"
The reference to the digital certificate to be used as "client certificate" from OPLON®Application Delivery Controller toward endpoints of this group.

associateName =: default value=""
This parameter enables association of symbolic names to the group of end-points in the form:
"symbolic_name_a symbolic_name_b symbolic_name_c ".
Each symbolic name must be separated by a space. The symbolic name cannot contain spaces.
For a dissertation of associative names refer to <params notificationDir="" />

loadBalancingType =: default value= "Adaptative"
Sets the balancing policy.
Possible values:

- RoundRobin
- Adaptative
- FailOver

idSessionsManagerName =: default value= "default"
Sets the type of recognition of the layer 7 HTTP/S session.
This value should be associated with a section <idSessions>. The value will be reported on all the URIPath groups within this grouping unless otherwise indicated.

cacheControl =: default value=""
This parameter will have a list of filters previously created in section <cacheControl> reporting the id, or ids, of reference for this domain.

dosMaxConcurrentConnectionsReaction =: default value= "false"
Activates the DDoS congestion resolver capping service, indicated by the parameter maxConcurrentConnections in section <endp>.

NOTE The activation of this feature is subject to the presence of the specific license for DoS Attack Prevention.

realmLogin =: default value= "": default value="realmLogin from endPointsGrouping"
Sets the basic login authentication of the service.
This login and the password will be delivered to services 'endp', unless otherwise specified in the domain or the 'endp', such as HTTP entity through the form e.g.

- authorization: Basic dXNyMjp1c3Iy

realmPassword =: default value="realmPassword from endPointsGrouping"
Sets the password for basic authentication of the service.

This password and the login will be delivered to services 'endp', unless otherwise specified in the domain or the 'endp', such as HTTP entity through the form e.g.

- authorization: Basic dXNyMjp1c3Iy

rewriteHeaderRules =: default value="endPointsGrouping +"
List of names of the rewriting rules for the HTTP HEADERS (layer 7 HTTP/S) to apply to the group of endpoints. It is possible to indicate multiple rules separated by one or more spaces.

Ex.

```
rewriteHeaderRules="redirSSLloginWhenNoSSL proxyTo"
```

The rules will be applied, if conditions permit, in sequence.

;**LAST**
For each rule name one can also indicate the parameter 'LAST' that in case the rule is performed determines stopping the application of the remaining rules of the sequence.
To set the value of the parameter the " ;" after the name and the name of the parameter.

eg.

```
rewriteHeaderRules="redirSSLloginWhenNoSSL;LAST proxyTo"
```

There must not be any spaces between the name and the parameters. In this case, if the rule redirSSLloginWhenNoSSL is applied the rule proxyTo will never run.

;ALWAYS
parameter ALWAYS indicates that the rule is always performed to indicate the value of the parameter is sufficient to put after the name the " ;" and the name of the parameter.

eg.

```
rewriteHeaderRules="redirSSLloginWhenNoSSL;LAST proxyTo;ALWAYS"
```

In this case the rule proxyto is performed regardless of the execution of the rule redirssloginwhennossl.

;NOP
The NOP parameter indicates that the rule should not be performed.

eg.

```
rewriteHeaderRules="redirSSLloginWhenNoSSL;NOP proxyTo;ALWAYS"
```

In this case the rule redirssloginwhennossl is not performed. The parameter NOP is useful to exclude the execution of general rules.

rewriteBodyRules =: default value="endPointsGrouping +"
List of names of the rewriting rules for the HTTP BODY (layer 7 HTTP/S) to apply to the group of endpoints. It is possible to indicate multiple rules separated by one or more spaces.

Ex.

```
rewriteBodyRules="addTrademarkParam absoluteToRelative echoRewriteBody"
```

The rules will be applied, if conditions permit, in sequence.

;**LAST**
For each rule name one can also indicate the parameter 'LAST' that in case the rule is performed determines stopping the application of the remaining rules of the sequence.
To set the value of the parameter place a " ;" (semi-colon) after the name and the name of the parameter.

Ex.

```
rewriteBodyRules="addTrademarkParam;LAST absoluteToRelative echoRewriteBody"
```

There must not be any spaces between the name and the parameters.. In this case, if the rule addTrademarkParam is applied the rules absoluteToRelative echoRewriteBody will never be performed.

;ALWAYS
The parameter ALWAYS indicates that the rule is always performed.
rewritebodyrules= "addtrademarkparam;LAST absolutetorelative echorewritebody;ALWAYS"

In this case the rule echorewritebody is performed regardless of the execution of the rule addtrademarkparam.

;NOP
The NOP parameter indicates that the rule must not be carried out
rewritebodyrules= "addtrademarkparam;LAST absolutetorelative;NOP echorewritebody;ALWAYS"

In this case the rule absolutetorelative is not performed. The parameter NOP is useful to exclude the execution of general rules.

cMessage =: default value= "endPointsGrouping: cMessage"
Name of the html file containing the appropriately contextual courtesy message.
At the time of loading parameters, at the start of the balancing process, existence of the file is verified in the directory:

- (LBL_HOME)/procsProfiles/A10_LBLGo/resources/html/messageName.html

If the file does not exist, it is reported at the time of startup and assigned the default value:

- (LBL_HOME)/procsProfiles/A10_LBLGo/resources/html/messageNoEndPoint.html

redirectNoHostsFound=: default value="endPointsGrouping:redirectNoHostsFound"
For Layer 7 HTTP/S (in case of hosts not available for the grouping "endPointsGrouping-virtualDomain-uriPath ") it is possible to execute a redirect to another URL as an alternative to the courtesy message. If this parameter is not set or the value “”, the courtesy message will be used.

needClientCert =: default value= "endPointsGrouping:needClientCert" UM=boolean
For Layer 7 HTTP/S this value imposes on the whole grouping of domains and endpoints use of the client certificate to access associated services. Domains and endpoints can change this value in case of exceptions.

needClientCertMessage =: default value= "endPointsGrouping:needClientCertMessage"
Name of the html file containing the appropriately contextual courtesy message.
At the time of loading parameters, at the start of the balancing process, existence of the file is verified in the directory:

- (LBL_HOME)/procsProfiles/A10_LBLGo/resources/html/messageName.html

If the file does not exist, it is reported at the time of startup and assigned the default value:

- (LBL_HOME)/procsProfiles/A10_LBLGo/resources/html/ messageNeedClientCert.html

cipherSuites=: valore di default=””
Set the ciphersuites

SSLProtocols=: valore di default=””
Set the SSL protcols

proxyHost =: default value = ""
Host name / domain of the service exposed internally
If enhanced, the host of the service is masked with the name indicated in the virtualDomainName

proxyPort =: default value = ""
Service door exposed internally
If enhanced, the service port is masked with the port present in the endpoint.

removePortFromHost =: default value = "false"
If true, remove the port from the Host header element:

proxyUrl =: default value = ""
URI / context of the service
Allows you to mask the service URI with the value indicated in the uriPath parameter

proxyLocation =: default value = "true"
Rewrites the content of the Location header with consistently with the values indicated in the proxyHost and proxyUrl parameters

proxyDestination =: default value = ""
Rewrites the contents of the Destination header with consistently with the values indicated in the proxyHost and proxyUrl parameters

proxyCookie =: default value = "true"
Rewrites the path and domain attribute of cookies consistently with the values indicated in the proxyHost and proxyUrl parameters

proxyRedirect =: default value = "false"
If set to true, it checks if the request matches the value set in proxyURL and redirects to uriPATH.
eg .:
uriPath=/images
proxyURL=/service/images
proxyRedirect="true"
proxyRedirectResponseCode="308"

if the request is:
/service/images/image180x71.png

In this case a redirect will be redirects to:
/images/image180x71.png

proxyRedirectResponseCode =: default value = "308"
If proxyRedirect is set to true it is the response code that will be used in the redirect

<endp>

```
<serviceconf>
        <iproxy>
                <endpoints>
                        <endPointsGrouping>
                                <virtualDomain>
                                        <endp
```

address =: default value= mandatory
The name or address of the backend server that hosts the service

dhcp =: default value = "false"
If true, it verifies the host name and converts the name into an address to have endpoints with a dynamic address, such as happens with AWS EC2 scalable services.

protocol =: default value = ""
If not empty, the protocol set on the listener changes for entire the group.

redirectToHttps =: default value = "false"
If true, in the presence of layer 7 services, HTTP / S checks whether the request is in HTTP and redirects all the resources into HTTPS.

sniHostName=: default value=”value in endPointsGrouping:=sniHostName and virtualDomain:=sniHostName”
Host name used for the TLS SNI connections to the backend services.
It is used to balance in SSL backend services that implement the Server Name Indication Protocol. If you set the parameter is inherited in endp. WARNING: If you use this value, the service address in endp tags must be entered with the numeric value (ipv4 or ipv6 notation) and not with the host name.

N.B.: In any case verify if the host name proposed from client HEADER is the same of backend. Otherwise you must rewrite the header with the backend host name for matching SNI Host in the protocol with the Host in the HTTP HEADER.

Ex

```
<endpoints>
  <endPointsGrouping enable="true">
      <virtualDomain rewriteHeaderRules="changeHost"  enable="true">
          <endp address="85.25.46.13" sniHostName="sni.velox.ch" port="443" SSL="true" uriPath=""/>
        <endp address="85.25.46.13" sniHostName="sni.velox.ch" port="443" SSL="true" uriPath=""/>
     </virtualDomain>
  </endPointsGrouping>
</endpoints>

<!-- Example of rewrite rule for change host name in HTTP HEADER: -->
<rewriteHeaderRule enable="true" name="changeHost" flow="REQUEST">
   <entities>
         <entity entityName="Host" value="sni.velox.ch" action="change"/>
   </entities>
</rewriteHeaderRule>

   Note: sni.velox.ch is an external SNI SSL test site.
```

sniForwarding =: default value = "false"
If true, it allows SNI to set endpoints and to automatically forward the request hortname to the backend with the same value as the origin.

SSLApplicationProtocols=: default="" values:"h2 http/1.1 undef"
If true, use the chipersuite in the order specified for the SSL / TLS listeners

```
eg: <params
                ...
                endPointSSLApplicationProtocols="h2 http/1.1 undef"
                endPointSSLUseCipherSuitesOrder="true"
                .../>

                <endPointsGrouping enable="true" SSLApplicationProtocols="h2 http/1.1 undef"
                        SSLUseCipherSuitesOrder="true">
                        <virtualDomain enable="true" SSLApplicationProtocols="h2 http/1.1 undef"
                                SSLUseCipherSuitesOrder="true">
                                        <endp address="192.168.56.131" port="8080" uriPath="/"
                                                SSLApplicationProtocols="h2 http/1.1 undef"
                          SSLUseCipherSuitesOrder="true" enable="true"/>
```

SSLUseCipherSuitesOrder=: default="true"
If true, it uses the chip-suites in the order indicated for the SSL / TLS listeners

redirectTo =: default value=""
This parameter is used to instruct a redirect in the face of a service request.
In fact, through this parameter, OPLON®Application Delivery Controller will respond to a service request with a redirection message to the URL indicated in the value. Redirects can also be done in a cyclical manner by introducing more redirect references for the same group of requests of similar end-point routines. The URL can also take on parameters or a query

string as long as one remembers to add the " &" in xml notation " & ".

port =: default value= " 0"
The port on which the backend service responds

sslPort =: default value= " <endp port>"
The port on which to attest the ssl connection if the listener is in SSL. The 'port' parameter is loaded by default.

SSLlistenerEval =: default value= "true"
This parameter, set to true by default, indicates the use of the listener as the SSL discriminant. If the listener is SSL then itwill t ake the SSL port otherwise it will take the plain port. This parameter is also evaluated to establish an SSL connection on the backend or not.
Used in conjunction with SSL the result will be:

CONDITION (usual):
SSL Listener && SSLlistenerEval==true && SSL=false
RESULT:
Front end connection SSL (encrypted)
Back end connection plain (transparent/clear) on port declared as SSL
NOTE:
In the cases of redirection SSL rewriting is applied automatically.
It can be disabled through the parameter sslRewriting= "false"

CONDITION (usual):
SSL Listener && SSLlistenerEval==true && SSL=true
RESULT:
Front end connection SSL (encrypted)
Back end connection SSL on SSL port

CONDITION (not usual):
Plain (transparent/clear) Listener && SSLlistenerEval==false && SSL=true
RESULT:
Front end connection plain (transparent/clear)
Back end connection SSL on SSL port
NOTE:
Apply rewrite on redirections from https to http

CONDITION (not usual in this form):
SSL Listener && SSLlistenerEval==false && SSL=true
RESULT:
Front end connection SSL (encrypted)
Back end connection SSL on SSL port

SSL =: default value= "false"
If true indicates that if the 'sslport' port is addressed, encrypted SSL traffic will be implemented.

SSLCert =: default value= "virtualDomain:=SSLCert"
The reference to the digital certificate to be used as "client certificate" from OPLON®Application Delivery Controller toward endpoints of this domain.

uriPath =: default value=""
On layer 7 (HTTP/S 1.0 /1.1) is the path that identifies the root path context of the service.
On layer 4 (TCP) this value must be set to "" or not be present.

proxyHost =: default value = ""
Host name / domain of the service exposed internally
If enhanced, the host of the service is masked with the name indicated in the virtualDomainName

proxyPort =: default value = ""
Service door exposed internally
If enhanced, the service port is masked with the port present in the endpoint.

removePortFromHost =: default value = "false"
If true, remove the port from the Host header element:

proxyUrl =: default value = ""
URI / context of the service
Allows you to mask the service URI with the value indicated in the uriPath parameter

proxyLocation =: default value = "true"
Rewrites the content of the Location header with consistently with the values indicated in the proxyHost and proxyUrl parameters

proxyDestination =: default value = ""
Rewrites the contents of the Destination header with consistently with the values indicated in the proxyHost and proxyUrl parameters

proxyCookie =: default value = "true"
Rewrites the path and domain attribute of cookies consistently with the values indicated in the proxyHost and proxyUrl parameters

portRewriting =: default value= "<virtualDomain> portRewriting"
If set to true rewrites the port number in the header from the endpoint with the value of the port requested by the client.
This behavior is necessary to balance backend services that respond by modifying the original header of the request with its own port number. With IIS this parameter must be set to true.

sslRewriting =: default value= " <virtualDomain> sslRewriting"
If set to true verifies that the listener is in SSL and if so rewrites the protocol from http to https in the case of redirect from an http protocol service.
This feature is useful in case the listener acts as an SSL terminator and the backend services

are not encrypted (transparent).

uriPathMatcher =: default value=""
On layer 7 (HTTP/S 1.0 /1.1), as an alternative to the parameter "uripath", the "uripathmatcher" parameter can be used with the regular expression applied to the URL of the request.

A possible application for example, is to route requests for static content to dedicated servers while requests for dynamic content will be routed to the application servers.

It is possible to manage different session policies for individual 'endp' - not managing for example, the session for the static content and managing instead the session for the dynamic content.

The following are some examples:

Ex.

```
<endp address="wiletrbackend" port="8080" uriPath="/Flowers/album" enable="true"/>
<endp address="roadtrbackend" port="8080" uriPath="/Flowers/album" enable="true"/>
<endp address="wiletrbackend" port="8787"
                    uriPathMatcher="/Flowers/album/thumbs/(.*).(jpg|gif|ico)"
                    idSessionsManagerName="nosessions" enable="true"/>
<endp address="roadtrbackend" port="8787"
                    uriPathMatcher="/Flowers/album/thumbs/(.*).(jpg|gif|ico)"
                    idSessionsManagerName="nosessions" enable="true"/>
```

Ex.

```
<endp address="wiletrbackend" port="8080"
                    uriPathMatcher="/Flowers/album/(.*)/(.*).jpg\?aaaaaaaa=1"
                    uriPathContextForSession="/Flowers/album/thumbs"/>
<endp address="roadtrbackend" port="8080"
                    uriPathMatcher="/Flowers/album/(.*)/(.*).jpg\?aaaaaaaa=1"
                    uriPathContextForSession="/Flowers/album/thumbs"/>
```

In these two fragments the parameter "uripathcontextforsession" can also be seen. This parameter is used in cases where LBL should manage the session to give the path of context that is not possible otherwise. In the case session management is not required or the session is managed by the service this value may be blank.

uriPathContextForSession =: default value=""
Parameter to manage the session and give the path of context in the presence of 'uriPathMatcher'.
In the case session management is not required or the session is managed by the service this value may be blank.

sequence =: default value=" " (space)
OPLON®Application Delivery Controller automatically determines the topologies of routing through an algorithm of path definitions. Normally the automatic determination of the paths is the best solution but manual intervention is possible such that arbitrary sequences can be applied through this parameter.
It is possible to analyze the result of the path hierarchies and the sequences through the log

at the start or the reinit of the balancing and routing services.

The following is an abbreviated example of the output of the generation of the hierarchies of routing:

```
|WARNING|1. ... -12:35:22|Hierarchy Sequence endPoints URIPath groups: -default_default_ _/training_-||
|WARNING|1. ... -12:35:22|Hierarchy Sequence endPoints URIPath groups: -default_default_ _/prvServletSimulateElaborationWait_-||
|WARNING|1. ... -12:35:22|Hierarchy Sequence endPoints URIPath groups: -default_default_ _/prvServletEcho_-||
|WARNING|1. ... -12:35:22|Hierarchy Sequence endPoints URIPath groups: -default_default_ _/papaia_/papaia-||
|WARNING|1. ... -12:35:22|Hierarchy Sequence endPoints URIPath groups: -default_default_ _/doRedirect_-||
|WARNING|1. ... -12:35:22|Hierarchy Sequence endPoints URIPath groups: -default_default_ _/TCOProject_-||
|WARNING|1. ... -12:35:22|Hierarchy Sequence endPoints URIPath groups: -default_default_ _/LBLUploadTest_-||
|WARNING|1. ... -12:35:22|Hierarchy Sequence endPoints URIPath groups: -default_default_ _/Flowers/album_-||
|WARNING|1. ... -12:35:22|Hierarchy Sequence endPoints URIPath groups: -default_default_ _/_-||
|WARNING|1. ... -12:35:22|Hierarchy Sequence endPoints URIPath groups: -default_default_ __-||
|WARNING|1. ... -12:35:22|Hierarchy Sequence endPoints URIPath groups: -HTTP_default_ _/_^/(areas$|areas/|areas_AHD$|areas_AHD/|areas_BHD$|
areas_BHD/)(.*)-||
|WARNING|1. ... -12:35:22|Hierarchy Sequence endPoints URIPath groups: -HTTP_default_ _/_^/(areassan$|areassan/|areassan_AHD$|areassan_AHD/|
areassan_BHD$|areassan_BHD/)(.*)-||
|WARNING|1. ... -12:35:22|Hierarchy Sequence endPoints URIPath groups: -default_default_B_/Flowers/album/thumbs_/Flowers/album/thumbs/(.*).
(jpg|gif|ico)-||
|WARNING|1. ... -12:35:22|Hierarchy Sequence endPoints URIPath groups: -default_default_A_/CEC2003_-||
|WARNING|1. ... -12:35:22|Hierarchy Sequence endPoints URIPath groups: -default_default_ _/trainingw_-||
|WARNING|1. ... -12:35:22|Hierarchy Sequence endPoints URIPath groups: -default_default_ _/training_-||
|WARNING|1. ... -12:35:22|Hierarchy Sequence endPoints URIPath groups: -default_default_ _/prvServletSimulateElaborationWait_-||
|WARNING|1. ... -12:35:22|Hierarchy Sequence endPoints URIPath groups: -default_default_ _/prvServletEcho_-||
|WARNING|1. ... -12:35:22|Hierarchy Sequence endPoints URIPath groups: -default_default_ _/papaia_/papaia-||
|WARNING|1. ... -12:35:22|Hierarchy Sequence endPoints URIPath groups: -default_default_ _/doRedirect_-||
|WARNING|1. ... -12:35:22|Hierarchy Sequence endPoints URIPath groups: -default_default_ _/TCOProject_-||
|WARNING|1. ... -12:35:22|Hierarchy Sequence endPoints URIPath groups: -default_default_ _/LBLUploadTest_-||
|WARNING|1. ... -12:35:22|Hierarchy Sequence endPoints URIPath groups: -default_default_ _/Flowers/album_-||
|WARNING|1. ... -12:35:22|Hierarchy Sequence endPoints URIPath groups: -default_default_ _/_-||
|WARNING|1. ... -12:35:22|Hierarchy Sequence endPoints URIPath groups: -default_default_ __-||
|WARNING|1. ... -12:35:22|Hierarchy Sequence endPoints URIPath groups: -HTTP_default_ _/_^/(areas$|areas/|areas_AHD$|areas_AHD/|areas_BHD$|
areas_BHD/)(.*)-||
|WARNING|1. ... -12:35:22|Hierarchy Sequence endPoints URIPath groups: -HTTP_default_ _/_^/(areassan$|areassan/|areassan_AHD$|areassan_AHD/|
areassan_BHD$|areassan_BHD/)(.*)-||
|WARNING|1. ... -12:35:22|Sequence endPoints URIPath groups: -default_default_B_/Flowers/album/thumbs_/Flowers/album/thumbs/(.*).(jpg|gif|ico)-||
|WARNING|1. ... -12:35:22|Sequence endPoints URIPath groups: -default_default_A_/CEC2003_-||
|WARNING|1. ... -12:35:22|Sequence endPoints URIPath groups: -default_default_ _/trainingw_-||
|WARNING|1. ... -12:35:22|Sequence endPoints URIPath groups: -default_default_ _/training_-||
|WARNING|1. ... -12:35:22|Sequence endPoints URIPath groups: -default_default_ _/prvServletSimulateElaborationWait_-||
|WARNING|1. ... -12:35:22|Sequence endPoints URIPath groups: -default_default_ _/prvServletEcho_-||
|WARNING|1. ... -12:35:22|Sequence endPoints URIPath groups: -default_default_ _/papaia_/papaia-||
|WARNING|1. ... -12:35:22|Sequence endPoints URIPath groups: -default_default_ _/doRedirect_-||
|WARNING|1. ... -12:35:22|Sequence endPoints URIPath groups: -default_default_ _/TCOProject_-||
|WARNING|1. ... -12:35:22|Sequence endPoints URIPath groups: -default_default_ _/LBLUploadTest_-||
|WARNING|1. ... -12:35:22|Sequence endPoints URIPath groups: -default_default_ _/Flowers/album_-||
|WARNING|1. ... -12:35:22|Sequence endPoints URIPath groups: -default_default_ _/_-||
|WARNING|1. ... -12:35:22|Sequence endPoints URIPath groups: -default_default_ __-||
|WARNING|1. ... -12:35:22|Sequence endPoints URIPath groups: -HTTP_default_ _/_^/(areas$|areas/|areas_AHD$|areas_AHD/|areas_BHD$|
areas_BHD/)(.*)-||
|WARNING|1. ... -12:35:22|Sequence endPoints URIPath groups: -HTTP_default_ _/_^/(areassan$|areassan/|areassan_AHD$|areassan_AHD/|
areassan_BHD$|areassan_BHD/)(.*)-||
```

near =: default value= "false"
In the event of geographically distributed peer sites, it is possible to indicate the closest endpoint relative to the same service.
The requests will be routed to the sites with near=true and only if all the endpoints declared 'near' are not reachable will the 'far' (not near) endpoints be selected.

maxConcurrentConnections =: default value= " 0"
If > 0 checks to see if the threshold of concurrent connections for this endpoint is exceeded. Additionally, the system counts all the limits for that virtual domain and notifies even if the sum of the connections for that virtual domain is exceeded.
If licensed for DoS Attack Prevention, upon exceeding the threshold of connections for this context (endpointsgrouping, domain, uripath) connections are cut so as to avoid overload on the back end.

uriPathContext=: default value= "uriPath"
On layer 7 (HTTP/S 1.0 /1.1) is the path that identifies the service root path context if the rewriter is used to change the context. In this particular case this value is used to populate the value of the path of the cookie generated by LBL with the value of the new context. Thus this value will only be used if a rule for a change of context (e.g. : /training /papaya) has been set up and LBL generates the session. This value in all other cases will never be used.

healthCheck =: default value= "true"
Enables/disables the functionality of the endpoint healtcheck. This feature is useful for some protocols that use variable ports.

downtimeOutOfOrder=: valore di default=”0”
The downtimeOutOfOrder parameter is used to temporarily put the endpoint out of service in case of udp timeout.
IMPORTANT: Use the new parameter with a healthCheck = "false". If we set healthCheck to "true", the endpoint will no longer come back online after a timeout.
Example:
<endp SSL = "false" address = "192.168.56.208" healthCheck = "false" downtimeOutOfOrder = "40000" port = "4501" associateName = "VPN000" enable = "true" />

enable =: default value= "true"
Enables or disables this service

idSessionsManagerName =: default value= "default"
Sets the type of recognition of the session to layer 7 HTTP/S.
This value must be associated with a section <idsessions>. The value will be reported to the group of endpoints with same URIPath. If the same URIPath are wrongly assigned to multiple groups of session recognition, an error is generated and the error is corrected with the name of the first assignment. The error will be available in the log.

associateName =: default value=""
This parameter enables association of symbolic names to the group of end-points in the form:
"symbolic_name_a symbolic_name_b symbolic_name_c ".
Each symbolic name must be separated by a space. The symbolic name cannot contain spaces.
For a dissertation of associative names refer to <params notificationDir=”” />

loadBalancingType =: default value= "Adaptative"
Sets the balancing policy.
Possible values:
- RoundRobin
- Adaptative
- FailOver

cMessage =: default value= "virtualDomain: cMessage"

Name of the html file containing the appropriately contextual courtesy message.
At the time of loading parameters, at the start of the balancing process, existence of the file is verified in the directory:

- (LBL_HOME)/procsProfiles/A10_LBLGo/resources/html/messageName.html

If the file does not exist, it is reported at the time of startup and assigned the default value:

- (LBL_HOME)/procsProfiles/A10_LBLGo/resources/html/messageNoEndPoint.html

This parameter must be uniform for uriPath. If not, the value of the first endpoint with the same uriPath will be used.

redirectNoHostsFound=: default value="virtualDomain:redirectNoHostsFound"
For Layer 7 HTTP/S (in case of hosts not available for the grouping "endPointsGrouping-virtualDomain-uriPath ") it is possible to execute a redirect to another URL as an alternative to the courtesy message. If this parameter is not set or the value “”, the courtesy message will be used.

needClientCert =: default value= "virtualDomain:needClientCert" UM=boolean
For Layer 7 HTTP/S this value imposes on the whole grouping of domains and endpoints use of the client certificate to access associated services. Domains and endpoints can change this value in case of exceptions.

needClientCertMessage =: default value= "virtualDomain:needClientCertMessage"
Name of the html file containing the appropriately contextual courtesy message.
At the time of loading parameters, at the start of the balancing process, existence of the file is verified in the directory:

- (LBL_HOME)/procsProfiles/A10_LBLGo/resources/html/messageName.html

If the file does not exist, it is reported at the time of startup and assigned the default value:

- (LBL_HOME)/procsProfiles/A10_LBLGo/resources/html/ messageNeedClientCert.html

realmLogin =: default value= "": default value="realmlogin from virtualDomain"
Sets the basic login authentication of the service.
This login and the password will be delivered to services 'endp', unless otherwise specified in the domain or the 'endp', such as HTTP entity through the form e.g.

- authorization: Basic dXNyMjp1c3Iy

realmPassword =: default value="realmPassword from virtualDomain"
Sets the password for basic authentication of the service.
This password and the login will be delivered to services 'endp', unless otherwise specified in the domain or the 'endp', such as HTTP entity through the form e.g.

- authorization: Basic dXNyMjp1c3Iy

rdPool=: default value="null"
Remote Desktop Pool is the subnet address used as a pool identifier

rdTarget=: default value="null"

Remote Desktop Target Server is the real address of that server referring to the endpoint

Ex. rdPool & rdTarget :
endp 0
rdPool 192.168.10.0
rdTarget 192.168.10.100

endp 1
rdPool 192.168.10.0
rdTarget 192.168.10.110

The system has been designed to allow simultaneous access to both pooled resources and resources dedicated to a specific user or groups of users. In practice, the two functions coexist at the same time giving maximum flexibility.

rewriteHeaderRules =: default value="endPointsGrouping + virtualDomain +"
List of names of the rewriting rules for the HTTP HEADERS (layer 7 HTTP/S) to apply to the endpoints. It is possible to indicate multiple rules separated by one or more spaces.

Ex.:

```
rewriteHeaderRules="redirSSLloginWhenNoSSL proxyTo"
```

The rules will be applied, if conditions permit, in sequence.

;**LAST**
For each rule name one can also indicate the parameter 'LAST' that in case the rule is performed determines stopping the application of the remaining rules of the sequence.
To set the value of the parameter the " ;" after the name and the name of the parameter.

eg.

```
rewriteHeaderRules="redirSSLloginWhenNoSSL;LAST proxyTo"
```

There must not be any spaces between the name and the parameters. In this case, if the rule redirSSLloginWhenNoSSL is applied the rule proxyTo will never run.

;ALWAYS
parameter ALWAYS indicates that the rule is always performed to indicate the value of the parameter is sufficient to put after the name the " ;" and the name of the parameter.

eg.

```
rewriteHeaderRules="redirSSLloginWhenNoSSL;LAST proxyTo;ALWAYS"
```

In this case the rule proxyto is performed regardless of the execution of the rule redirssloginwhennossl.

;NOP
The NOP parameter indicates that the rule should not be performed.

eg.

```
rewriteHeaderRules="redirSSLloginWhenNoSSL;NOP proxyTo;ALWAYS"
```

In this case the rule redirssllogінwhennossl is not performed. The parameter NOP is useful to exclude the execution of general rules.

rewriteBodyRules =: default value="endPointsGrouping + virtualDomain +"
List of names of the rewriting rules for the HTTP BODY (layer 7 HTTP/S) to apply to the endpoints. It is possible to indicate multiple rules separated by one or more spaces.

Ex.

```
rewriteBodyRules="addTrademarkParam absoluteToRelative echoRewriteBody"
```

The rules will be applied, if conditions permit, in sequence.

;LAST
For each rule name one can also indicate the parameter 'LAST' that in case the rule is performed determines stopping the application of the remaining rules of the sequence.
To set the value of the parameter place a " ;" (semi-colon) after the name and the name of the parameter.

Ex.

```
rewriteBodyRules="addTrademarkParam;LAST absoluteToRelative echoRewriteBody"
```

There must not be any spaces between the name and the parameters.. In this case, if the rule addTrademarkParam is applied the rules absoluteToRelative echoRewriteBody will never be performed.

;ALWAYS
The parameter ALWAYS indicates that the rule is always performed.rewritebodyrules= "addtrademarkparam;LAST absolutetorelative echorewritebody;ALWAYS"

In this case the rule echorewritebody is performed regardless of the execution of the rule addtrademarkparam.

;NOP
The NOP parameter indicates that the rule must not be carried out
rewritebodyrules= "addtrademarkparam;LAST absolutetorelative;NOP echorewritebody;ALWAYS"

In this case the rule absolutetorelative is not performed. The parameter NOP is useful to exclude the execution of general rules.

healthcheck <endpoints>

In the Standard and Enterprise versions of OPLON®Application Delivery Controller it is mandatory that the endpoint below be included so that OPLON®Application Delivery Controller can perform the health check of the nodes:

```
<!-- this group and virtual domain must be healthcheck for proxy health check across all public network -->
        <endPointsGrouping groupName="healthcheck" enable="true">
```

```
            <virtualDomain virtualDomainName="healthcheck" enable="true">
                <!-- this address must be localhost for proxy health check across public network -->
                <endp address="localhost" port="5991" uriPath="/HealthCheck" enable="true"/>
            </virtualDomain>
        </endPointsGrouping>
```

Example <endpoints>

```
        <endPointsGrouping enable="true">
          <virtualDomain virtualDomainName="www.mango_fruit.com" portRewriting="true"
                         enable="true">
            <endp address="129.157.86.10" port="8080" uriPath="home" enable="true"/>
            <endp address="129.157.86.15" port="8080" uriPath="home" enable="true"/>
          </virtualDomain>
          <virtualDomain virtualDomainName="www.papaia_fruit.com" enable="true">
            <endp address="129.157.86.20" port="8080" uriPath="home" enable="true"/>
            <endp address="129.157.86.20" port="8080" uriPath="private" enable="true"/>
            <endp address="129.157.86.25" port="8080" uriPath="home" enable="true"/>
            <endp address="129.157.86.25" port="8080" uriPath="private" enable="true"/>
            <endp address="129.157.86.30" port="8080" uriPath="home" enable="true"/>
            <endp address="129.157.86.30" port="8080" uriPath="private" enable="true"/>
          </virtualDomain>
          <virtualDomain virtualDomainName="www.ananas_fruit.org" enable="true">
            <endp address="129.157.86.35" port="8080" uriPath="home" enable="true"/>
            <endp address="129.157.86.35" port="8080" uriPath="private" enable="true"/>
            <endp address="129.157.86.40" port="8080" uriPath="home" enable="true"/>
            <endp address="129.157.86.40" port="8080" uriPath="private" enable="true"/>
            <endp address="129.157.86.45" port="8080" uriPath="home" enable="true"/>
            <endp address="129.157.86.45" port="8080" uriPath="private" enable="true"/>
          </virtualDomain>
</endPointsGrouping>
```

Examples redirect

Configuration example with redirect on different URL:

```
    <endpoints>
      <endPointsGrouping enable="true">
        <virtualDomain portRewriting="true" enable="true">
          <endp uriPath="/doRedirect" redirectTo="https://monster/prvServletEcho?A=0" enable="true"/>
          <endp uriPath="/doRedirect" redirectTo="https://monster/prvServletEcho?A=1" enable="true"/>
          <endp uriPath="/doRedirect" redirectTo="https://monster/prvServletEcho?A=2" enable="true"/>
            …
```

NOTE-1: The HTTP “redirect” command by its very nature cannot store any cookies and for this reason, in this particular condition, it is not possible to assign a session identifier automatically. On the redirection end-points the round-robin will be performed each time the group/URI specified is invoked.

NOTE-2: LBL(tm)LoadBalancer does not perform any autonomous "health check" of the point of redirection because it may not be reachable, for example an external address protected by the firewall.
It is possible to use the directory "(LBL_HOME)\lib\notificationDir" and the tag "associateName" to notify of the unavailability of the service. For this purpose an outside

plugin suitable to the situtation may be used.

NOTE-3: If the parameters "address" and "port" are added to "redirectTo" parameter the health check will be executed on the port address specified as for the other end-points.

Example of a configuration with redirect to different URLs and associative name for unreachable outside interaction (healthcheck not possible):

```
<endpoints>
  <endPointsGrouping  enable="true">
    <virtualDomain portRewriting="true" enable="true">
      <endp uriPath="/doRedirect" redirectTo="https://monster/prvServletEcho?A=0"
            enable="true" associateName="urlRedirectService"/>
      <endp uriPath="/doRedirect" redirectTo="https://monster/prvServletEcho?A=1"
            enable="true" associateName="urlRedirectService"/>
      <endp uriPath="/doRedirect" redirectTo="https://monster/prvServletEcho?A=2"
            enable="true" associateName="urlRedirectService"/>
        …
```

Configuration example with redirect on different URLs , associative name and address and port for automatic health-check of the resource:

```
<endpoints>
   <endPointsGrouping enable="true">
       <virtualDomain portRewriting="true" enable="true">
         <endp uriPath="/doRedirect" redirectTo="https://monster/prvServletEcho?A=0"
               address="monster" port="443" enable="true" associateName="urlRedirectService"/>
         <endp uriPath="/doRedirect" redirectTo="https://monster/prvServletEcho?A=1"
               address="monster" port="443" enable="true" associateName="urlRedirectService"/>
         <endp uriPath="/doRedirect" redirectTo="https://monster/prvServletEcho?A=2"
               address="monster" port="443" enable="true" associateName="urlRedirectService"/>
          ...
```

Example of a configuration with request of the SSL certificate on a defined (URIPath) context:

```
 <endpoints>
   <endPointsGrouping enable="true">
       <virtualDomain portRewriting="true" enable="true">
         <endp uriPath="/mySecureContext" needClientCert="true" needClientCertMessage=""
               address="monster" port="8080" enable="true"/>
         <endp uriPath="/mySecureContext"  needClientCert="true" needClientCertMessage=""
               address="monster" port="8080" enable="true"/>
         <endp uriPath="/mySecureContext"  needClientCert="true" needClientCertMessage=""
               address="monster" port="8080" enable="true"/>
         <endp uriPath="/"
               address="monster" port="8080" enable="true"/>
         <endp uriPath="/"
               address="monster" port="8080" enable="true"/>
         <endp uriPath="/"
               address="monster" port="8080" enable="true"/>
          …
```

In this case the client certificate is only needed for the context "/mySecureContext".

<sysobserver>

```
<serviceconf>
    <iproxy>
        <sysobserver>
```

OPLON®Application Delivery Controller is a system based on services that communicate with each other through message queues. This section allows association of a service to a name.
In fact OPLON®Application Delivery Controller can be virtualized and contain multiple independent instances of balancers in the same JVM .

<service>

```
<serviceconf>
    <iproxy>
        <sysobserver>
            <service
```

name =: default value=""
The name of the logical service.

id =: default value=""
The name of the virtualized instance service.

2

OPLON®Application Delivery Controller healthcheck.xml

(LBL_HOME)/procsProfiles/A10_LBLGo/conf/healthcheck.xml

This parameter allows verifying the operational status of the instance. Normally this does not require changes unless the default port "5991" on localhost should already be occupied by some other service.

The structure of the healthcheck.xml file is as follows:

```
<serviceconf>
  <copyright>
  </copyright>
  <healthcheck>
    <params>
    </params>
    <sysobserver>
      <service>
      </service>
    </sysobserver>
  </healthcheck>
</serviceconf>
```

<serviceconf>

<healthcheck>

<params>

```
<serviceconf>
    <healthcheck>
        <params
```

The section contains the healthcheck service configuration parameters.

address=: default value=”localhost”
The value must be equal to localhost.
The loadbalancer nodes access this service by passing through the balancing layer and simultaneously verifying the network status and the balancing service itself.

port=: default value=”5991”
The port on which the service responds.

NOTE: This is the only parameter to change only if there is already an active service on the same port.

reuseAddress=: default value=”true”
The corresponding SO_REUSEADDR socket parameter.

timeOut=: default value=”1500” UM=Millesimi di secondo
The healthCheck service is an HTTP 1.0/1.1 service and this value indicates the connection timeout.

timeOutFactor=: default value=”300”
The multiplicative time out factor.

tcpNoDelay=: default value=”true”
Enables/disables the Nagle algorithm for checking data buffering.

concurrentWorkers=: default value=”20”
Initial number of workers for the resolution of connection requests.

maxConcurrentWorkers=: default value=”100”
Maximum number of workers for resolution of connection requests.

healthCheckContextPath=: default value=”/LBLHealthCheck”
The path to the operational healthcheck of the balancing system. This value normally will never change unless already in use in other applications. If this value is changed, it must also be changed in "systemsmonitor_m.xml", "iproxy.xml" and "healthcheck.xml".

webAppsDir=: default value=”lib/webroot_healtcheck/webapps”
Web application home Directory.

webAppsConfDir=: default value=”lib/webroot_healtcheck/webappsconf”
Web application configuration Directory.

webSecurityDir=: default value=”lib/webroot_healtcheck/websecurity”
Web application security configuration Directory.

<sysobserver>

```
<serviceconf>
        <healthcheck>
                <sysobserver>
```

OPLON® LoadBalancer is a system based on services that communicate with each other through message queues. This section allows association of a service with a name. In fact OPLON® LoadBalancer can be virtualized so that multiple independent balancer instances can be run on the same JVM.

<service>

```
<serviceconf>
        <healthcheck>
                <sysobserver>
                        <service
```

name=: default value=””
he logical service name.

id=: default value=””
The service name of the virtualized instance.

3 OPLON®Application Delivery Controller lookup.xml

(LBL_HOME)/procsProfiles/A10_LBLGo/conf/lookup.xml

The lookup service enables OPLON® LoadBalancer to automatically identify a node. This functionality is provided through multicast services propagated through the private network. The private network must be used exclusively by OPLON® LoadBalancer nodes.

```
<serviceconf>
  <copyright>
  </copyright>
  <lookup>
    <params>
    </params>
    <peersInstances>
      <peer>
      </peer>
    </peersInstances>
    <redundantLookupInterfaces>
      <interface>
          <peersInstances>
              <peer>
              </peer>
          </peersInstances>
      </interface>
    </redundantLookupInterfaces>
    <sysobserver>
       <service>
       </service>
    </sysobserver>
  </lookup>
</serviceconf>
```

<lookup>

<params>

```
<serviceconf>
        <lookup>
                <params
```

protocol=: default value=”udp”
Possible values: multicast, udp
Defines the communication protocol of the cluster nodes.
In the case of geographic clusters, or networks that do not support the multicast protocol, the cluster nodes can be set to use the udp protocol.
In this case, all cluster nodes must be declared in the <peersInstances> paragraph

Example:

```
<params
        protocol="udp"
        group="LBLStandardGroup"
        subGroup="LBLSubGroup"
        groupWeight="100"
        interfaceAddress="192.168.44.208"
        address="192.168.44.208"
        port="6789"
        timeToLive="1"
        timeOut="-1"
        datagramLength="1024">
</params>

<peersInstances>
        <peer enable="true"
                description="udp peer 1"
                address="192.168.44.110" port="6789" timeOut="-1"/>
        <peer enable="true"
                description="udp peer 2"
                address="192.168.44.111" port="6789" timeOut="-1"/>
        <peer enable="true"
                description="udp peer 3"
                address="192.168.44.112" port="6789" timeOut="-1"/>
</peersInstances>
```

encryptionPhrase =: default value = ""
When set, communications between cluster nodes are encrypted and the parameter value defines the encryption key, which must be set with the same value on all cluster nodes.

group=: default value=”LBLDistributionGroup”
The name of the group of nodes.
Multiple groups of nodes can coexist within the same private network. This name is used to to distinguish messages that originate from the same group of balancers. Messages from other groups will be discarded. Depending on the OPLON® distribution used, the default name can be LBLEnterpriseGroup or LBLStandardGroup.

subGroup=: default value=”LBLSubGroup”
The name of the subgroup of nodes.
This parameter is usually used in the Standard distribution of OPLON® LoadBalancer for managing Disaster Recovery. By setting the Sub-group (subGroup) of the node(s) installed on the DR site with a different name from the subgroups of the nodes in the main site, routing information from sessions is not propagated to and from the nodes with different subgroups. In an infrastructure with DR management, node(s) placed in the DR site must route service requests to their own backend services while maintaining common Virtual IP (VIP) services neded for cluster management.

groupWeight=: default value=”100”
The weight of this balancer.
In the Standard version of OPLON® LoadBalancer this value determines assignment of the master node among the nodes. If the nodes have the same weight, OPLON® LoadBalancer calculates random numbers and whichever node gets the highest score takes over as master. As of version 6.0, it is possible to use Standard Edition nodes for DR purposes and Enterprise Edition for the take over of the main site's addresses.

interfaceAddress=: default value=””
The host name/address of the private network interface

address==: default value=”228.5.6.7”
The transmit/receive multicast address.

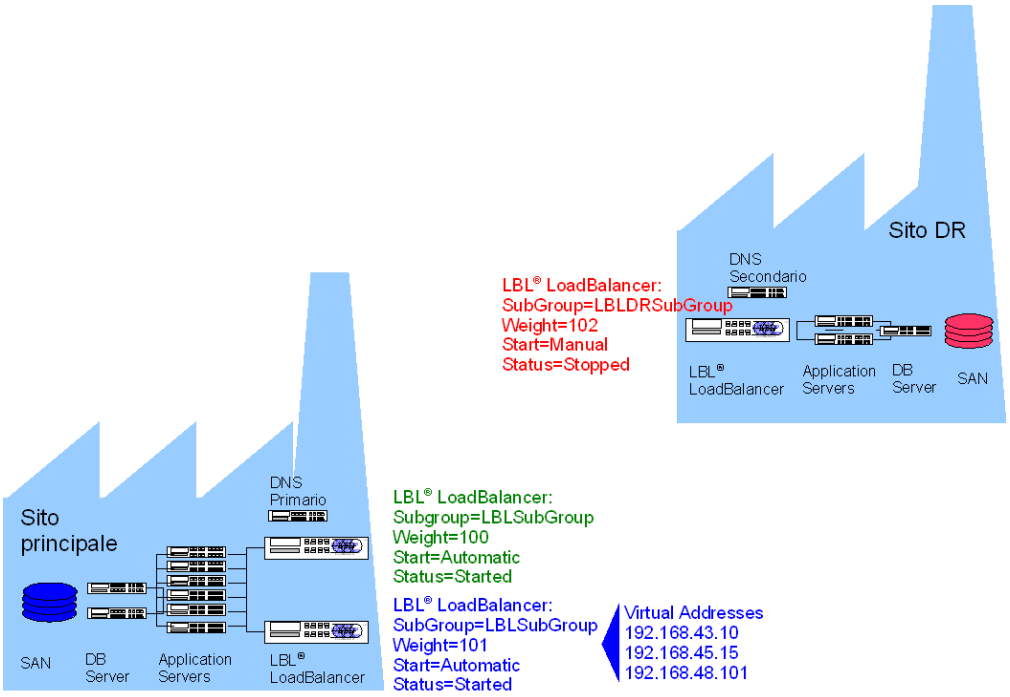
port=: default value=”6789”
The transmit/receive multicast port.

timeToLive=: default value=”1”
The multicast packet life duration.

timeOut=: default value=”-1” UM=Milliseconds
Amount of time in milliseconds to wait for a packet in reception before timing out. Default value is -1 (no time out)

datagramLength=: default value=”1024”
Length of datagram.

Example using the subgroup in a Business Continuity scenario:



<peersInstances>

```
<serviceconf>
        <lookup>
            <peersInstances>
```

In this paragraph all the nodes of the balancing cluster are declared if the udp protocol is used for the communication between the nodes (lookup).

<peer>

```
<serviceconf>
        <lookup>
            <peersInstances>
                <peer
```

The peer tag describes a cluster node. The tag is used in case the lookup protocol is udp.

enable=: default value=”true”
Enables or disables this section.

description =: default value = ""
It is a description of the node to which the paragraph refers

address ==: default value = ""
It is the address of the node to send messages to

port=: default value=”6789”
The transmit/receive multicast port.

timeToLive=: default value=”1”
The multicast packet life duration.

timeOut=: default value=”-1” UM=Milliseconds
Amount of time in milliseconds to wait for a packet in reception before timing out. Default value is -1 (no time out)

<redundantLookupInterfaces>

```
<serviceconf>
        <lookup>
                <redundantLookupInterfaces>
```

This optional section contains redundant private interfaces where parallel lookup services as described in the params section are assigned. More interfaces that work together to perform a proper multicast routing in the event of failure of a private network card/adapter.

<interface>

```
<serviceconf>
        <lookup>
                <redundantLookupInterfaces>
                        <interface
```

enable=: default value=”true”
Enables or disables this section.

description=: default value=””
Description of the host address/name of the private network interface.

interfaceAddress=: default value=””
Host address/name of the private network interface.

address==: default value=”228.5.6.7”
The transmit/receive multicast address.

port=: default value=”6789”
The transmit/receive multicast port.

timeToLive=: default value=”1”
The multicast packet life duration.

timeOut=: default value=”-1” UM=Milliseconds
Amount of time in milliseconds to wait for a packet in reception before timing out. Default value is -1 (no time out)

<peersInstances>

```
<serviceconf>
        <lookup>
                <peersInstances>
```

In this paragraph all the nodes of the balancing cluster are declared in case the udp protocol is used for the communication between the nodes (lookup).

<peer>

```
<serviceconf>
        <lookup>
                <peersInstances>
                        <peer
```

The peer tag describes a cluster node. The tag is used in case the lookup protocol is udp.

enable=: default value=”true”
Enables or disables this section.

description =: default value = ""
It is a description of the node to which the paragraph refers

address ==: default value = ""
It is the address of the node to send messages to

port=: default value =””
The receive udp port.

timeToLive=: default value=”1”
The multicast packet life duration.

timeOut=: default value=”-1” UM=Milliseconds
Amount of time in milliseconds to wait for a packet in reception before timing out. Default value is -1 (no time out)

Eg: redundantLookupInterfaces multicast protocol

```
 <redundantLookupInterfaces>
        <interface description="monitor redundant interface"
                interfaceAddress="roadlbloneredundant"
                address="228.5.6.7" port="6789" timeToLive="1" timeOut="-1"/>
        <interface description="monitor redundant interface001"
                interfaceAddress="roadlbloneredundant001"
                address="228.5.6.7" port="6789" timeToLive="1" timeOut="-1"/>
 </redundantLookupInterfaces>
```

Eg: redundantLookupInterfaces udp protocol

```
<redundantLookupInterfaces>
		<interface enable="true"
			description="monitor redundant interface" interfaceAddress="monstermonitor"
			address="192.168.45.208" port="6789" timeToLive="1" timeOut="-1">
			<peersInstances>
				<peer enable="true"
					description="monitor redundant interface"
					address="192.168.45.110" port="6789" timeToLive="1"
					timeOut="-1"/>
				<peer enable="true"
					description="monitor redundant interface"
					address="192.168.45.111" port="6789" timeToLive="1"
					timeOut="-1"/>
				<peer enable="true"
					description="monitor redundant interface"
					address="192.168.45.112" port="6789" timeToLive="1"
					timeOut="-1"/>
			</peersInstances>
		</interface>
		<interface enable="true"
			description="monitor redundant interface" interfaceAddress="monstermonitor"
			address="192.168.46.208" port="6789" timeToLive="1" timeOut="-1">
			<peersInstances>
				<peer enable="true"
					description="monitor redundant interface"
					address="192.168.46.110" port="6789" timeToLive="1"
					timeOut="-1"/>
				<peer enable="true"
					description="monitor redundant interface"
					address="192.168.46.111" port="6789" timeToLive="1"
					timeOut="-1"/>
				<peer enable="true"
					description="monitor redundant interface"
					address="192.168.46.112" port="6789" timeToLive="1"
					timeOut="-1"/>
			</peersInstances>
		</interface>
		</redundantLookupInterfaces>
```

4 OPLON®Application Delivery Controller vrrpserver.xml

(LBL_HOME)/procsProfiles/A10_LBLGo//conf/vrrpserver.xml

The Virtual Routing Redundancy Protocol system allows the Gateway to automatically update the information necessary for the maintenance of sessions and determine the operating status during run-time.

```
<serviceconf>
  <copyright>
  </copyright>
  <vrrpserver>
    <params>
    </params>
    <sysobserver>
      <service>
      </service>
    </sysobserver>
  </vrrpserver>
</serviceconf>
```

<serviceconf>

<vrrpserver>

<params>

```
<serviceconf>
    <vrrpserver>
        <params
```

address=: default value=”localhost”
Il valore deve essere corrispondente ad un indirizzo o nome host afferente alla rete privata.

The value must be equal to an address or host name pertinent to the private network.

port=: default value=”5991”
The port on which the service responds.

reuseAddress=: default value=”true”
The corresponding SO_REUSEADDR socket parameter.

timeOut=: default value=”1500” UM=Thousandths of a second
The vrrpserver is an HTTP 1.0/1.1 service and this value indicates the connection timeout.

timeOutFactor=: default value=”300”
The multiplicative time out factor.

tcpNoDelay=: default value=”true”
Enables/disables the Nagle algorithm for checking data buffering.

concurrentWorkers=: default value=”20”
Initial number of workers for the resolution of connection requests.

maxConcurrentWorkers=: default value=”100”
Maximum number of workers for resolution of connection requests.

webAppsDir=: default value=”lib/webroot_vrrp/webapps”
Web application home Directory.

webAppsConfDir=: default value=”lib/webroot_vrrp/webappsconf”
Web application configuration Directory.

webSecurityDir=: default value=”lib/webroot_vrrp/websecurity”
Web application security configuration Directory.

7 OPLON®Application Delivery Controller systemsmonitor_m

(LBL_HOME)/procsProfiles/A10_LBLGo/conf/systemsmonitor_m.xml

This parameters file describes the parameters for managing the virtual addresses.
As of version 7.0 multiple addresses and multiple network cards can be specified to achieve redundancy for local hardware of the virtual address itself.

```
<serviceconf>
  <copyright>
  </copyright>
  <systemsmonitor_m>
    <params>
    </params>
    <virtualAddressesMgr>
      <virtualAddresses>
          <virtualInterface>
          </virtualInterface>
          <virtualInterface>
          </virtualInterface>
          ...
          <publicNetworkHealthCheckPolicy>
          </publicNetworkHealthCheckPolicy>
           <backendNetworkHealthCheckPolicy>
           </backendNetworkHealthCheckPolicy>
      </virtualAddresses>
      ...
    </virtualAddressesMgr>
    <sysobserver>
       <service>
       </service>
    </sysobserver>
  </systemsmonitor_m>
```

```
</serviceconf>
```

<serviceconf>

<systemsmonitor_m>

<params>

```
<serviceconf>
    <systemsmonitor_m>
        <params
```

systemsMonitorGroup =: default value= "systemsMonitorGroup"
Logical name of the addresses virtualization system.

notificationDir =: default value= "lib/notificationdir"
Directory for notification of external events.
By creating a file name with the name set in systemsMonitorGroup with the extension "outOfOrder" it is possible to declare the entire balancing service out of order.

Ex.:

```
# touch  outOfOrder.systemsMonitorGroup
```

monitorTimer =: default value= "6000" UM=Milliseconds
Timing of healthcheck in Milliseconds

systemLostTime =: value of default="monitortimer * 2" UM=Milliseconds
Time necessary to declare a system OutOfOrder

lostTransactionTimeOut =: default value= " 15000" UM=Milliseconds
Indicates the time required to declare lost transactions from the HeartBeat not yet received and then arrange a total realignment (BULK-LOAD SESSIONS).

dropTimeFederatedSystemOutOfOrder =: default= " 86400000" UM=Milliseconds
Indicates the time required to delete an OutOfOrder system from the table of federated systems. The default value is equal to 1 day in milliseconds.

sysCommandTimeOut =: default value= " 10000" UM=Milliseconds
Indicates the time required to declare time-out on a system command. If the command exceeds this limit an abort command is executed, and subsequently the control is released to the application.

sysCommandCheckRate =: default value= " 300" UM=Milliseconds
The frequency by which to check on the state of the system command.

sysCommandRemoteURL =: default= "https://localhost:5992/sysCommand"
The URL of the service to run system commands

createConnectionTimeOut =: default= " 5000" UM=Milliseconds

It is the timeout set for ICMP (ping) messages and TCP connect to verify the status of the network.
This value is also used as the default value if not set otherwise on individual points of healthcheck (sections: <publicNetworkHealthCheckPolicy> and <backendNetworkHealthCheckPolicy>)

stayMasterAfterFailOverEvent=:default=”true”
This parameter allows you to change the conditions of increased priority during the virtual addresses fail-over steps. Set to "false" the priority increase if the virtual address exists and has no balancing activities (AND). Set to "true” the priority increase if one of the two tests lack, virtual address absent or no balance activity in the address (OR). This parameter is designed for CLOUD installations where the virtual addresses can not be resident on the same instances of LBL. This parameter set to "true" is normally used with the virtual address indication "external" to be checked in the stages of fail-over. The external virtual address is characterized by having device="" and deviceName = "".

```
Example:
  <params
        systemsMonitorGroup="systemsMonitorGroup"
                sysCommandRemoteURL="https://localhost:5992/SysCommand"
                stayMasterAfterFailOverEvent="true">
 </params>

 <virtualAddressesMgr>
    <virtualAddress enable="true"
          description="virtual address INTERNAL"
          address="192.168.44.10"
          netmask="255.255.255.0"
          healthCheckPort="80"
          healthCheckUriPath="/LBLHealthCheck">
        <virtualInterface device="eth2"
                  deviceName="eth2"/>
        <publicNetworkHealthCheckPolicy>
                <publicNetwork address="192.168.43.131" description="Sys A1 public"/>
        </publicNetworkHealthCheckPolicy>
        <backendNetworkHealthCheckPolicy>
                <backendNetwork address="192.168.43.131" description="Sys A1 backend"/>
        </backendNetworkHealthCheckPolicy>
    </virtualAddress>
    <virtualAddress enable="true"
        description="virtual address CLOUD DUMMY"
        address="192.168.43.114"
        netmask="255.255.255.0"
        healthCheckPort="80"
        healthCheckUriPath="/LBLHealthCheck">
       <virtualInterface device=""
                    deviceName=""/>
       <publicNetworkHealthCheckPolicy>
       </publicNetworkHealthCheckPolicy>
       <backendNetworkHealthCheckPolicy>
       </backendNetworkHealthCheckPolicy>
     </virtualAddress>
 </virtualAddressesMgr>
```

<virtualAddressesMgr>

The section contains the definition of the virtual address

<virtualAddress>

```
<serviceconf>
    <systemsmonitor_m>
        <virtualAddressesMgr>
            <virtualAddress
```

enable =: default value= "true"
Enable / disable virtual address

description =: default value=""
Describes the virtual address

staticAddress =: default value=""
The static address which is used as quorum to determine the reachability of the network. When used with hardware redundancy it is necessary to have an address associated with the device used to check the reachability of the network. This parameter is used as the setting of the network card, to verify the existence of the virtual address and then draw the conclusions.
If not set the local network cards fail over capability cannot be utilized.
For IPv6 the representation must be made between square brackets [fdd4:3c3f:aaaa::99].

staticNetmask =: default value= "255.255.255 .0"
The netmask in digits (e.g. : 255.255.255.0) used during the setting of address described on <staticAddress>.
If the address is in IPv6 the value is determined by the precision desired. Ex: for 64 setting of the address will be fdd4:3c3f:aaaa::99/64

address =: default value=""
The virtual address in digits (e.g. : 192.168.43.10).
For IPv6 the representation must be made between square brackets [fdd4:3c3f:aaaa::99].

netmask =: default value= "255.255.255 .0"
The netmask of the virtual address in digits (e.g. : 255.255.255.0).
If the address is in IPv6 the value is determined by the precision desired. Ex: for 64 setting of the address will be fdd4:3c3f:aaaa::99/64

healthCheckPort =: default value="-1"
The port on which to run the healthcheck test. If "" the health check is not performed. This value is very important because it determines the status of activity not only of the IP address, but also of the balancing and routing system.

healthCheckSSL=:default value=”false”
If true run a health check HTTP by establishing an encrypted connection.

healthCheckUriPath =: default value= " /LBLHealthCheck"
The path of the balancing system activities healthcheck. This value is usually never changed unless already in use in other applications. If this value is changed here, it must also be changed in “systemsmonitor_m.xml”, “iproxy.xml” e in “healthcheck.xml”.

<virtualInterface>

```
<serviceconf>
    <systemsmonitor_m>
        <virtualAddressesMgr>
            <virtualAddress>
                <virtualInterface
```

As of version 7.0 this section may be duplicated to manage the redundancy of the network interfaces hardware. In this case, the defined parameters starting with "static" (ex.: staticDevice) identify the parameters to be used for the local address of quorum described in the previous section.
On MS Windows the parameters will be the same while for Unix/Linux they will be different because they will indicate 2 distinct aliases for a static address in determining the quorum and another alias for the virtual address. (See examples)

staticDevice =: default value=""
Parameter used for the hardware redundancy of the network cards, and the identifier of the alias associated with a physical device for Unix/Linux (e.g. : e1000g4 / eth4) and the physical device for MS Windows

```
Ex.: PCI\\
VEN_10EC&amp;DEV_8139&amp;SUBSYS_813910EC&amp;REV_10\\4&amp;2E98101C&amp;0&amp;10
F0
```

This parameter must not be set if there is no redundancy in hardware of the network adapters.

staticDeviceName =: default value=""
The name of the device.
On MS Windows must be the value of the name visible in network resources. Refer to the OPLON®Application Delivery Controller Standard&Enterprise Installation manual for more details on installation.
This parameter must not be set if there is no redundancy in hardware of the network adapters.

device =: default value=""
The identifier of the alias associated with a physical device for Unix/Linux and the physical device for MS Windows.
To obtain this name "devcon.exe " for MS Windows and "ifconfig" for Unix/Linux systems will be used.
Refer to the OPLON®Application Delivery Controller Standard&Enterprise Installation manual for more details about installing and setting up. If set to "" with the deviceName means an external address (eg .: CLOUD).

deviceName =: default value=""
The name of the device.

On MS Windows must be the value of the name visible in network resources. Refer to the OPLON®Application Delivery Controller Standard&Enterprise Installation manual for more details on installation. If set to "" with the device means an external address (eg .: CLOUD).

The following are some possible configurations:
MS Windows:

```
<virtualInterface device="PCI\\
VEN_8086&amp;DEV_1019&amp;SUBSYS_80F71043&amp;REV_00\\4&amp;3B3CB9B1&amp;0&amp;08
18"
  deviceName="LBLPublic">
```

Linux:

```
<virtualInterface device="eth1:0"
                  deviceName="eth1:0">
```

Solaris/OpenSolaris (7,8,9,10)

```
<virtualInterface device="iprb2:1"
                   deviceName="iprb2:1">
```

ATTENTION MULTIPLE VIRTUAL NETWORK CARDS
If there are more network cards available in the public network for redundancy or for management of addresses, lengthen the OPLON®Application Delivery Controller shutdown time in order to allow the balancer to deallocate the addresses.
To effect this change simply insert the parameter
-DLBL_DELAY_SWITCHOFF in the startup phase on A10_LBLGo.xml
Ex.: -DLBL_DELAY_SWITCHOFF=60

The default value is 40".

Example of configuration files for MS Windows with network hardware redundancy

```
<virtualAddressesMgr>
    <virtualAddress enable="true"
                    description="Rete internet A"
                    staticAddress="192.168.43.100"
                    staticNetmask="255.255.255.0"
                    address="192.168.43.10"
                    netmask="255.255.255.0"
                    healthCheckPort="5656"
                    healthCheckUriPath="/HealthCheck">
        <virtualInterface staticDevice="PCI\\VEN_10EC&amp;DEV_8139&amp;SUBSYS_81391OEC&amp;REV_10\\4&amp;2E98101C&amp;0&amp;10F0"
                          staticDeviceName="LBLPublic"
                          device="PCI\\VEN_10EC&amp;DEV_8139&amp;SUBSYS_81391OEC&amp;REV_10\\4&amp;2E98101C&amp;0&amp;10F0"
                          deviceName="LBLPublic">
        </virtualInterface>
        <virtualInterface staticDevice="PCI\\VEN_10EC&amp;DEV_8139&amp;SUBSYS_81391OEC&amp;REV_10\\4&amp;2E98101C&amp;0&amp;18F0"
                          staticDeviceName="LBLBackendMonitor"
                          device="PCI\\VEN_10EC&amp;DEV_8139&amp;SUBSYS_81391OEC&amp;REV_10\\4&amp;2E98101C&amp;0&amp;18F0"
                          deviceName="LBLBackendMonitor">
        </virtualInterface>
        <publicNetworkHealthCheckPolicy>
            <publicNetwork address="192.168.43.103" port="22" description="Sys A2 public"  createConnectionTimeOut="4000"/>
            <publicNetwork address="192.168.43.104" description="Sys A3 public"/>
        </publicNetworkHealthCheckPolicy>
        <backendNetworkHealthCheckPolicy>
            <backendNetwork address="192.168.45.103" description="Sys A2 backend" createConnectionTimeOut="6000"/>
            <backendNetwork address="192.168.45.104" port="22" description="Sys A4 backend"/>
        </backendNetworkHealthCheckPolicy>
    </virtualAddress>
</serviceconf>
```

Example of configuration files for Solaris without network hardware redundancy

```
<virtualAddress enable="true"
                description="Internet A"
                address="192.168.43.136"
                netmask="255.255.255.0">
    <virtualInterface device="e1000g0:1"
                      deviceName="e1000g0:1"/>
    <publicNetworkHealthCheckPolicy>
        <publicNetwork address="192.168.43.114" port="22" description="Sys A2 public"/>
        <publicNetwork address="192.168.43.104" port="22" description="Sys A2 public"/>
        <publicNetwork address="192.168.43.101" description="Sys A3 public"/>
    </publicNetworkHealthCheckPolicy>
    <backendNetworkHealthCheckPolicy>
        <backendNetwork address="192.168.45.114" port="22" description="Sys A2 backend"/>
        <backendNetwork address="192.168.45.104" port="22" description="Sys A2 backend"/>
        <backendNetwork address="192.168.45.101" description="Sys A3 backend"/>
    </backendNetworkHealthCheckPolicy>
</virtualAddress>
```

Example of configuration files for Solaris with network hardware redundancy

```
<virtualAddress enable="true"
                description="redundant interfaces address A"
                staticAddress="192.168.43.113"
                address="192.168.43.136"
                netmask="255.255.255.0">
    <virtualInterface staticDevice="e1000g4:1"
                      staticDeviceName="e1000g4:1"
                      device="e1000g4:2"
                      deviceName="e1000g4:2"/>
    <virtualInterface staticDevice="e1000g5:1"
                      staticDeviceName="e1000g5:1"
                      device="e1000g5:2"
                      deviceName="e1000g5:2"/>
    <publicNetworkHealthCheckPolicy>
       <publicNetwork address="192.168.43.114" port="22" description="Sys A2 public"/>
       <publicNetwork address="192.168.43.104" port="22" description="Sys A2 public"/>
       <publicNetwork address="192.168.43.101" description="Sys A3 public"/>
    </publicNetworkHealthCheckPolicy>
    <backendNetworkHealthCheckPolicy>
       <backendNetwork address="192.168.45.114" port="22" description="Sys A2 backend"/>
       <backendNetwork address="192.168.45.104" port="22" description="Sys A2 backend"/>
       <backendNetwork address="192.168.45.101" description="Sys A3 backend"/>
    </backendNetworkHealthCheckPolicy>
</virtualAddress>
```

<publicNetworkHealthCheckPolicy>

This section shows which network addresses to use to check the status of the public network. Enter at least 3 addresses, the recommended number is 5 addresses. Warning: If all health check addresses are not reachable, the virtual IP is removed and the system is set in out of order to allow the twin system to be able to detect the address.

<publicNetwork>

```
<serviceconf>
        <systemsmonitor_m>
                <virtualAddressesMgr>
                        <virtualAddress>
                                <publicNetworkHealthCheckPolicy>
                                        <publicNetwork
```

This section specifies the network address to use to check the status of the public network.

address =: default value=""
Address on which to perform the verification tests of the public network

port =: default value=""
If the value > 0 (greater than zero) the health check will be run through TCP/IP with a connection and subsequent disconnect. If no value is provided the health check will run through ICMP (ping).

description =: default value=""
Description of the verification address.

createConnectionTimeOut =: default= " 5000" UM=Milliseconds
It is the timeout set for ICMP (ping) messages and TCP connect to verify the status of the network.
If not set the default value set in section <systemsmonitor_m> <params> is used.

The following is a possible configuration:

```
<publicNetworkHealthCheckPolicy>
        <publicNetwork address="192.168.43.2"   description="System A con LBL(tm)"/>
        <publicNetwork address="192.168.43.55" description="System B"
                        createConnectionTimeOut="7000"/>
        <publicNetwork address="192.168.43.57" description="System C"/>
</publicNetworkHealthCheckPolicy>
```

ATTENTION: If Healthcheck rules are not set, the response will always be positive. Be sure to have set at least 5 health check rules per physical adapter or logical application.

<backendNetworkHealthCheckPolicy>

This section shows which network addresses to use to check the status of the backend network.
Enter at least 3 addresses, the recommended number is 5 addresses. Warning: If all health check addresses are not reachable, the virtual IP is removed and the system is set in out of order to allow the twin system to be able to detect the address.

<backendNetwork>

```
<serviceconf>
        <systemsmonitor_m>
                <virtualAddressesMgr>
                        <virtualAddress>
```

<backendNetworkHealthCheckPolicy>
<backendNetwork

This section specifies the network address to use to check the status of the backend network.

address =: default value=""
Address on which to perform the verification tests of the backend network

port =: default value=""
If the value > 0 (greater than zero) the health check will be run through TCP/IP with a connection and subsequent disconnect. If no value is provided the health check will run through ICMP (ping).

description =: default value=""
Description of the verification address.

healthCheckTimeOut =: default= " 4000" UM=Milliseconds
It is the timeout set for ICMP (ping) messages and TCP connect to verify the status of the network.
If not set the default value set in section <systemsmonitor_m> <params> is used.

The following is a possible configuration:

```
<backendNetworkHealthCheckPolicy>
        <backendNetwork address="192.168.45.2" description="System A con LBL(tm)"/>
        <backendNetwork address="192.168.45.55" description="System B"
                                healthCheckTimeOut="7000"/>
        <backendNetwork address="192.168.45.56" description="System C"/>
</backendNetworkHealthCheckPolicy>
```

ATTENTION: If Healthcheck rules are not set, the response will always be positive. Be sure to have set at least 5 health check rules per physical adapter or logical application.

5 OPLON® systemsmonitor_m.xml

(LBL_HOME)/procsProfiles/XXX_procName/conf/syslog.xml

This file is not offered by default in the directory
(LBL_HOME)/procsProfiles/XXX_procName/conf
in fact it is normally never modified and hence is distributed within the library/program LBLLoadBalancer.jar

To change it, simply copy it from the inside of the library into the directory
(LBL_HOME)/procsProfiles/XXX_procName/conf
LBL ® S.A.A.I. will acquire the newly placed file and will no longer take into account the original inside the library.

```
<serviceconf>
   <copyright>
   </copyright>
        <syslog>
                <params>
                </params>
        </syslog>
</serviceconf>
```

<syslog>

<params>

```
<serviceconf>
        <syslog>
                <params
```

logDir=:default value=”logs”
The directory log files are created and used. This value can take on values relative to the process that is running as well as absolute values at the filesystem level.

logFileSuffix=:default value=”LBLLog.txt”
The completion of the log file name. eg.: YYYYMMDD_hostname_LBLLog.txt

logRotationHistoryDays=:default value=”15” UM=Days
Specifies the retention time for the log file. When the number of days indicated is exceeded, the file is deleted.

logMessageDeduplication=:default value=”20” UM=number of same messages
As of version 7.0 a de-duplication of log messages feature was introduced. Consecutive messages in arrival being the same, the log file is not executed immediately. The log is only executed if temporally a different message arrives or if the number of same messages indicated by this parameter is exceeded. To this end, a new field was added to the end of the message indicating the number of repetitive messages prior to that output. eg.:

```
CONNECTION ERROR PERSISTS TO END POINT: wiletrbackend:8181/trai
--- EndPoint roadtrbackend:8282/training DOWN!!!!||
--- EndPoint roadtrbackend:8282/training DOWN!!!!|20|

-- END POINT : wiletrbackend:8181/training READY AGAIN!||


-- END POINT : roadtrbackend:8282/training READY AGAIN!||

--- EndPoint wiletrbackend:8181/training DOWN!!!!||
--- EndPoint wiletrbackend:8181/training DOWN!!!!|16|
--- EndPoint roadtrbackend:8282/training DOWN!!!!||
--- EndPoint wiletrbackend:8181/training DOWN!!!!||
--- EndPoint wiletrbackend:8181/training DOWN!!!!|6|
--- EndPoint roadtrbackend:8282/training DOWN!!!!||
--- EndPoint roadtrbackend:8282/training DOWN!!!!|3|
--- EndPoint roadtrbackend:8181/training DOWN!!!!||
--- EndPoint roadtrbackend:8181/training DOWN!!!!|3|
--- EndPoint wiletrbackend:8181/training DOWN!!!!||
--- EndPoint wiletrbackend:8181/training DOWN!!!!|9|
--- EndPoint roadtrbackend:8181/training DOWN!!!!||
--- EndPoint roadtrbackend:8181/training DOWN!!!!|2|
--- EndPoint wiletrbackend:8181/training DOWN!!!!||
--- EndPoint wiletrbackend:8181/training DOWN!!!!|13|

CONNECTION ERROR PERSISTS TO END POINT: wiletrbackend:8181/trai
```

6 OPLON® statisticbrokercache.xml

This file describes the service responsible for collecting statistics of 1° and 2° levels. The 1° level statistics are collected in memory through an asynchronous service.

By way of tempering (defined in a parameter), the cache level 1 is brought into permanent storage defined as cache level 2. Using another service the cache level 2 is sent to a completely separate process that collects the values, in a transactional manner, again into an area of permanent storage ready for archiving in a relational DataBase. The file therefore describes the parameters for caching service of 1° and 2° level statistics.

```
<serviceconf>
   <copyright>
   </copyright>
        <statisticbrokercache>
                <params>
                </params>
        </statisticbrokercache>
</serviceconf>
```

<statisticbrokercache>

<params>

```
<serviceconf>
        <statisticbrokercache>
                <params
```

frequency=:valore di default=”10000” UM=Milliseconds
The frequency for updating and transferring of statistics from one cache to another.

maxFilesCacheHistory=:valore di default=”8640” UM=Number of files
The maximum number of cached files beyond which the oldest is deleted. Normally, the cache is populated with 1 file and only in the event that the level 3 cache service is not available the directory will hold the files until this number, which is equal to 1 full working day divided by the frequency parameter, is reached.

In other words if the frequency of emptying the level 1 in memory cache is “10” and there were 24 hours of network traffic the resulting number of files will be 8640. Hence there would be 8640 files sent to the level 3 cache in a 24-hour day in which there had been no resolution to the interruption.

dateFormat=:default=”dd/MM/yyyy HH:mm:ss:SSSS”
The date format of the statistical data coming from Statistic Web Cache Broker

delimiter=:default=”|” (pipe)
The character that delimits the fields during the exchange of information.

statisticCacheHistoryDir=:valore di default=”lib/statisticCacheHistory”
The level 2 cache directory.

remoteServerURL=:default=”http://localhost:5993/updateStatistic”
Upload URL for moving the statistics to level 3 cache. The deletion of the level 2 cache takes place only upon final writing to the level 3 cache which is confirmed with an appropriate message.

SSL= default=”false”
Indicates whether the statistics upload service is using SSL.

description= default=”LBL(r)LoadBalancer statistic web cache manager”
Indicates the description of the HTTP client for delivery of the statistics to the web service.

7 OPLON® statisticbrokerwebcache.xml

This file describes the service responsible for collecting statistics of 3° and 4° level.

The 1° level statistics are collected in memory through an asynchronous service.
By way of tempering (defined in a parameter), the cache level 1 is brought into permanent storage defined as cache level 2. Using another service the cache level 2 is sent to a completely separate process that collects the values, in a transactional manner, again into an area of permanent storage, level 3 cache, ready for archiving in a relational DataBase at level 4.

The file describes the parameters and caching of 3° and 4° level statistics service.

```
<serviceconf>
   <copyright>
   </copyright>
        <statisticbrokerwebcache>
                <params>
                </params>
        </statisticbrokerwebcache>
</serviceconf>
```

<statisticbrokerwebcache>

<params>

```
<serviceconf>
        <statisticbrokerwebcache>
                <params
```

address=: default=”localhost”
The value must be equal to localhost. In fact the other loadbalancers access this service through the balancing layer verifying simultaneously the network status and balancing service itself.

port=: default=”5993”
The port on which the service responds. This paremeter nly needs to be modified if there is

already an active service on the same port.

timeOut=: default=”1500” UM=Thousandths of a second
The healthCheck service is an HTTP 1.0/1.1 and this value indicates the connection timeout.

timeOutFactor=: default=”300”
The multiplicative factor of time out.

tcpNoDelay=: default=”true”
Enables/disables the Nagle algorithm to check for data buffering.

concurrentWorkers=: default=”20”
The initial number of workers for resolution of the connection requests.

maxConcurrentWorkers=: default=”100”
The maximum number of workers for resolution of connection requests.

contextPath=: default=”/updateStatistic”
The path for activities healthcheck. This value will normally never change unless already use in other applications. If this value is changed, it must also be changed in "systemsmonitor_m.xml".

webAppsDir=: default=”lib/webroot_statisticbrokerweb/webapps”
Web applications home directory.

webAppsConfDir=: default=”lib/webroot_statisticbrokerweb/webappsconf”
Web applications configuration directory.

webSecurityDir=: default=”lib/webroot_healtcheck/websecurity”
Web applications security configuration directory.

frequency=:default=”10000” UM=Milliseconds
Frequency for updating of statistics.

statisticBrokerCacheFrequency=:default=”10000” UM=Milliseconds
The frequency for emptying the cache for temporal alignment with the process of creating the level 1° and 2° caches.

timeLimitObsoleteStatisticSnap=:default=”60000” UM=Milliseconds
The time limit for which the log within the database is no longer considered current for instant view.

timeLimitObsoleteStatisticSnapSessions=:default=”420000” UM=Milliseconds
The time limit for which the log within the database is no longer considered current for instant statistical sessions snapshot view.

maxFilesCacheHistory=:default=”8640” UM=Number of files

The maximum number of cached files beyond which the oldest is deleted. Normally, the cache is populated with 1 file and only in the event that the level 3 cache service is not available the directory will hold the files until this number, which is equal to 1 full working day divided by the frequency parameter, is reached.

In other words if the frequency of emptying the level 1 in memory cache is “10” and there were 24 hours of network traffic the resulting number of files will be 8640. Hence there would be 8640 files sent to the level 3 cache in a 24-hour day in which there had been no resolution to the interruption.

statisticCacheHistoryDir=:default=”lib/statisticWebCacheHistory”
The level 3° cache directory.

dateFormat=:default=”dd/MM/yyyy HH:mm:ss:SSSS”
The date format of the statistical data coming from Statistic Broker Cache (balancing process)

delimiter=:default=”|” (pipe)
The character that delimits the fields during the exchange of information.

maxEmbeddedDBSize=:default=”5368709120” UM=byte
In the configuration with DB Embedded (embedded JavaDB) this parameter specifies the maximum size of the database on mass storage unit. Upon exceeding this threshold the log flags the problem and the DB is recreated. This measure is to prevent the depletion of system resources on the node that contains the load balancing process. This feature was created bearing in mind that given this is an embedded DB, thus the inability to query external statistics, the data are merely statistical and therefore not critical. This parameter is not taken into account in all other cases of logging and stroage of statistics on the networked DB.

DBInitialConnectionPool=:default=”5”
Initial number of database connections.

DBMaxConnectionPool=:default=”5”
Maximum number of database connections. Upon exceeding this limit a new connection will still be opened but it will cease at the end of the request and will not be reused by the connection pool.

DBDriver=:default=”org.apache.derby.jdbc.EmbeddedDriver”
Driver for management of the relational database.

DBProtocol=:default=”jdbc:derby:”
The Protocol for the database connection string.

DBName=:default=”lib/LBLDBStatistics”
The name of the database. Together with the DBProtocol determines the connection string to the DB. Another example of DBName might be “//trantor:1527/S:\javaDB\ LBLDBStatistics” that identifies a networked Database service on the sytem “trantor” on port 1527.

DBLogin=:default=”null”
The login to connect to the DB. A value of null indicates that no authentication is required. OPLON® LoadBalancer by default does not give access to the embedded DB from the outside to the listeners of the DB that can run SQL scripts. By establishing the DB as networked and changing the parameters, a login can be specified using this parameter.

DBPassword=:default=”null”
The password to connect to the DB. A value of null indicates that no authentication is required. OPLON® LoadBalancer by default does not give access to the embedded DB from the outside to the listeners of the DB that can run SQL scripts. By establishing the DB as networked and changing the parameters, a login can be specified using this parameter.

DBSchemaName=:default for JavaDB=”APP.” default other DB=””
The schema name that generated the tables. For JavaDB (derby, aka cloudscape) the default is "APP" and is prefixed to all queries. For the other DBs the default is "" (blank) and thus the table name resulting from the query returns only the name.

DBDateFormat=:default=”yyyy-MM-dd”
The date format that expresses the values of the day, month, and year used in the database.

DBTimeFormat=:default=”HH:mm:ss”
The date format that expresses the values of hours, minutes, and seconds used in the database

DBSetDateFormat=:default=”” UM=Comand SQL
In some circumstances, it is necessary to run an SQL command to set the date/time format. This parameter when set runs in the context of the DataBase session to determine the manner of access to the data. This parameter is used only in the case of Oracle DB and its value should be:

```
ALTER SESSION set NLS_DATE_FORMAT = &apos;YYYY-MM-DD HH24:MI:SS&apos;
```

DBMaxHistoryDays=:default=”2” UM=Days
For values greater than 0 this is the maximum number of days logs will be stored in the Database. The date change will clear the traffic logs previously created based on the number of days set relative to the new date. If the value is less than 0 (zero) the statistics for all tables will never be cleared regardless of the values set in the parameters that are specific to the tables.

DBMaxHistoryDaysSessionActivity=:default=”DBMaxHistoryDays” UM=Days
For values greater than 0 this is the maximum number of days stored in the SESSION_ACTIVITY Database table. The date change will clear the traffic logs previously created based on the number of days set relative to the new date. If the value is less than 0 (zero) the statistics for all tables will never be cleared regardless of the values set in the parameters that are specific to the tables.

DBMaxHistoryDaysPoolQueuesActivity=:default=”DBMaxHistoryDays” UM=Days

For values greater than 0 this is the maximum number of days stored in the POOL_QUEUES_ACTIVITY Database table. The date change will clear the traffic logs previously created based on the number of days set relative to the new date. If the value is less than 0 (zero) the statistics for all tables will never be cleared regardless of the values set in the parameters that are specific to the tables.

DBMaxHistoryDaysIncomingQueueHighWater=:default=”DBMaxHistoryDays” UM=Days
For values greater than 0 this is the maximum number of days stored in the INCOMING_QUEUE_HIGHWATER_LEVEL. Database table. The date change will clear the traffic logs previously created based on the number of days set relative to the new date. If the value is less than 0 (zero) the statistics for all tables will never be cleared regardless of the values set in the parameters that are specific to the tables.

DBMaxHistoryDaysL4TcpTcpSSL=:default=”DBMaxHistoryDays” UM=Days
For values greater than 0 this is the maximum number of days stored in the L4_TCP_TCPSSL. The date change will clear the traffic logs previously created based on the number of days set relative to the new date. If the value is less than 0 (zero) the statistics for all tables will never be cleared regardless of the values set in the parameters that are specific to the tables.

DBMaxHistoryDaysL7HttpHttps=:default=”DBMaxHistoryDays” UM=Days
For values greater than 0 this is the maximum number of days stored in the L7_HTTP_HTTPS Database table. The date change will clear the traffic logs previously created based on the number of days set relative to the new date. If the value is less than 0 (zero) the statistics for all tables will never be cleared regardless of the values set in the parameters that are specific to the tables.

DBMaxHistoryDaysL4Datagram=:default=”DBMaxHistoryDays” UM=Days
For values greater than 0 this is the maximum number of days stored in the L4_DATAGRAM Database table. The date change will clear the traffic logs previously created based on the number of days set relative to the new date. If the value is less than 0 (zero) the statistics for all tables will never be cleared regardless of the values set in the parameters that are specific to the tables.

DBLoadHistorySessionActivity=:default=”true” UM=boolean
If set to false, the database will not be populated with the corresponding values.

DBLoadHistoryPoolQueuesActivity=:default=”true” UM=boolean
If set to false, the database will not be populated with the corresponding values.

DBLoadHistoryIncomingQueueHighWater=:default=”true” UM=boolean
If set to false, the database will not be populated with the corresponding values.

DBLoadHistoryL4TcpTcpSSL=:default=”true” UM=boolean
If set to false, the database will not be populated with the corresponding values.

DBLoadHistoryL7HttpHttps=:default=”true” UM=boolean

If set to false, the database will not be populated with the corresponding values.

DBLoadHistoryL4Datagram=:default=”true” UM=boolean
If set to false, the database will not be populated with the corresponding values.

DBLoadHistorySyslogEvent=:default=”true” UM=boolean
If set to false, the database will not be populated with the corresponding values.

DBQueryLimit=:default=”5000” UM=Records
It is the maximum number of records that can be extracted in queries to avoid overflow.

DBVarcharLimit=:default=”4000” UM=Byte
The maximum number of bytes that will be inserted into the VARCHAR column types. See LBL_DBNetworkedConfiguration.pdf manual for more details on this value for different database platforms.

DBMaxScriptBatch=:default=”200” UM=Byte
The maximum number of SQL batch scripts used to load the database. Some databases support a limited number of batch script. Currently JavaDB (aks derby, cloudscape) supports up to 65534 batch scripts. Refer to the manual of the of the database provider to change this value.

cookieValueToExtract=:default=”null” UM=Byte
In the L7_HTTP_HTTPS a new column DUMMY varchar(300) can be created to get a cookie value from a cookie in the COOKIES column. It is simple to extract the cookie value. In the parameter file statisticbrokerwebcache.xml create a parameter cookieValueToExtract with the name value of the cookie to extract.

Example with LBLSESSIONID:

```
<params>
        ...
        ...
        DBMaxHistoryDays="0"
        DBMaxHistoryDaysSessionActivity="2"
        DBMaxHistoryDaysPoolQueuesActivity="2"
        DBMaxHistoryDaysIncomingQueueHighWater="2"
        DBMaxHistoryDaysL4TcpTcpSSL="2"
        DBMaxHistoryDaysL7HttpHttps="2"
        DBMaxHistoryDaysL4Datagram="2"
        DBMaxHistoryDaysSyslogEvent=”2”
        DBQueryLimit="5000"
        cookieValueToExtract="LBLSESSIONID">
</params>
```

before setting the parameter, run the following script with the new DUMMY column in L7_HTTP_HTTPS:

Oracle:

```
ALTER TABLE L7_HTTP_HTTPS ADD DUMMY varchar2(300);
```

Other DB:

```
ALTER TABLE L7_HTTP_HTTPS ADD DUMMY varchar(300);
```

8 OPLON®Traffic Monetizer statisticbrokerwebcachedwh.xml

This file describes the service responsible for collecting statistics of 3° and 4° level in advanced mode.

The 1° level statistics are collected in memory through an asynchronous service.
By way of tempering (defined in a parameter), the cache level 1 is brought into permanent storage defined as cache level 2. Using another service the cache level 2 is sent to a completely separate process that collects the values, in a transactional manner, again into an area of permanent storage, level 3 cache, ready for archiving in a relational DataBase at level 4.

The file describes the parameters and caching of 3° and 4° level statistics service.

```
<serviceconf>
   <copyright>
   </copyright>
        <statisticbrokerwebcachedwh>
                <params>
                </params>
        </statisticbrokerwebcachedwh>
</serviceconf>
```

<statisticbrokerwebcachedwh>

<params>

```
<serviceconf>
        <statisticbrokerwebcache>
                <params
```

address=: default=”localhost”
The value must be equal to localhost. In fact the other loadbalancers access this service through the balancing layer verifying simultaneously the network status and balancing

service itself.

port=: default=”5993”
The port on which the service responds. This paremeter nly needs to be modified if there is already an active service on the same port.

timeOut=: default=”1500” UM=Thousandths of a second
The healthCheck service is an HTTP 1.0/1.1 and this value indicates the connection timeout.

timeOutFactor=: default=”300”
The multiplicative factor of time out.

tcpNoDelay=: default=”true”
Enables/disables the Nagle algorithm to check for data buffering.

concurrentWorkers=: default=”20”
The initial number of workers for resolution of the connection requests.

maxConcurrentWorkers=: default=”100”
The maximum number of workers for resolution of connection requests.

contextPath=: default=”/updateStatistic”
The path for activities healthcheck. This value will normally never change unless already use in other applications. If this value is changed, it must also be changed in "systemsmonitor_m.xml".

webAppsDir=: default=”lib/webroot_statisticbrokerweb/webapps”
Web applications home directory.

webAppsConfDir=: default=”lib/webroot_statisticbrokerweb/webappsconf”
Web applications configuration directory.

webSecurityDir=: default=”lib/webroot_healtcheck/websecurity”
Web applications security configuration directory.

frequency=:default=”10000” UM=Milliseconds
Frequency for updating of statistics.

statisticBrokerCacheFrequency=:default=”10000” UM=Milliseconds
The frequency for emptying the cache for temporal alignment with the process of creating the level 1° and 2° caches.

timeLimitObsoleteStatisticSnap=:default=”60000” UM=Milliseconds
The time limit for which the log within the database is no longer considered current for instant view.

timeLimitObsoleteStatisticSnapSessions=:default=”420000” UM=Milliseconds
The time limit for which the log within the database is no longer considered current for

instant statistical sessions snapshot view.

maxFilesCacheHistory=:default=”8640” UM=Number of files

The maximum number of cached files beyond which the oldest is deleted. Normally, the cache is populated with 1 file and only in the event that the level 3 cache service is not available the directory will hold the files until this number, which is equal to 1 full working day divided by the frequency parameter, is reached.

In other words if the frequency of emptying the level 1 in memory cache is “10” and there were 24 hours of network traffic the resulting number of files will be 8640. Hence there would be 8640 files sent to the level 3 cache in a 24-hour day in which there had been no resolution to the interruption.

statisticCacheHistoryDir=:default=”lib/statisticWebCacheHistory”
The level 3° cache directory.

dateFormat=:default=”dd/MM/yyyy HH:mm:ss:SSSS”
The date format of the statistical data coming from Statistic Broker Cache (balancing process)

delimiter=:default=”|” (pipe)
The character that delimits the fields during the exchange of information.

maxEmbeddedDBSize=:default=”5368709120” UM=byte
In the configuration with DB Embedded (embedded JavaDB) this parameter specifies the maximum size of the database on mass storage unit. Upon exceeding this threshold the log flags the problem and the DB is recreated. This measure is to prevent the depletion of system resources on the node that contains the load balancing process. This feature was created bearing in mind that given this is an embedded DB, thus the inability to query external statistics, the data are merely statistical and therefore not critical. This parameter is not taken into account in all other cases of logging and stroage of statistics on the networked DB.

DBLoaderInterceptorClassPath =: default= "interceptors/"
The loading path of the interceptor class defined in DBLoaderInterceptorClass. This path is added to the classpath of the running JVM.

DBLoaderInterceptorClass=:default="loadbalancer.services.statisticbroker.webcachedwh.LBLDBStatisticBrokerBatchStatementsInterceptor "
The class loading of the database. The distributions already contain a class template available in

- (LBL_HOME)/interceptors/dwhInterceptors/ LBLDBStatisticBrokerBatchStatementsInterceptorTemplate.java

The classes contained in this directory can be compiled through the tools: compile.bat or compile.sh present in the directory.

DBInitialConnectionPool=:default=”1”
Initial number of database connections.

DBMaxConnectionPool=:default=”1”
Maximum number of database connections. Upon exceeding this limit a new connection will still be opened but it will cease at the end of the request and will not be reused by the connection pool.

DBDriver=:default=”oracle.jdbc.driver.OracleDriver”
Driver for management of the relational database.

DBProtocol=:default=”jdbc:oracle:”
The Protocol for the database connection string.

DBName=:default=””
The name of the database. Together with the DBProtocol determines the connection string to the DB.

DBLogin=:default=”null”
The login to connect to the DB.

DBPassword=:default=”null”
The password to connect to the DB.

DBDateFormat=:default=”yyyy-MM-dd”
The date format that expresses the values of the day, month, and year used in the database.

DBTimeFormat=:default=”HH:mm:ss”
The date format that expresses the values of hours, minutes, and seconds used in the database.

DBSetDateFormat =): default="" UM=SQL Command
SQL command to set the format date/time. This parameter when set is performed in the context of a session of the database. In the case of Oracle DB its value must be:

```
ALTER SESSION set NLS_DATE_FORMAT = &apos;YYYY-MM-DD HH24:MI:SS&apos;
```

DBMaxHistoryDays=:default=”-1” UM=Days
For values greater than 0 this is the maximum number of days logs will be stored in the Database. The date change will clear the traffic logs previously created based on the number of days set relative to the new date. If the value is less than 0 (zero) the statistics for all tables will never be cleared regardless of the values set in the parameters that are specific to the tables.

DBMaxHistoryDaysSessionActivity=:default=”DBMaxHistoryDays” UM=Days
For values greater than 0 this is the maximum number of days stored in the SESSION_ACTIVITY Database table. The date change will clear the traffic logs previously created based on the number of days set relative to the new date. If the value is less than 0 (zero) the statistics for all tables will never be cleared.

DBMaxHistoryDaysPoolQueuesActivity=:default=”DBMaxHistoryDays” UM=Days
For values greater than 0 this is the maximum number of days stored in the

POOL_QUEUES_ACTIVITY Database table. The date change will clear the traffic logs previously created based on the number of days set relative to the new date. If the value is less than 0 (zero) the statistics for all tables will never be cleared.

DBMaxHistoryDaysIncomingQueueHighWater=:default=”DBMaxHistoryDays” UM=Days
For values greater than 0 this is the maximum number of days stored in the INCOMING_QUEUE_HIGHWATER_LEVEL. Database table. The date change will clear the traffic logs previously created based on the number of days set relative to the new date. If the value is less than 0 (zero) the statistics for all tables will never be cleared.

DBMaxHistoryDaysL4TcpTcpSSL=:default=”DBMaxHistoryDays” UM=Days
For values greater than 0 this is the maximum number of days stored in the L4_TCP_TCPSSL. The date change will clear the traffic logs previously created based on the number of days set relative to the new date. If the value is less than 0 (zero) the statistics for all tables will never be cleared.

DBMaxHistoryDaysL7HttpHttps=:default=”DBMaxHistoryDays” UM=Days
For values greater than 0 this is the maximum number of days stored in the L7_HTTP_HTTPS Database table. The date change will clear the traffic logs previously created based on the number of days set relative to the new date. If the value is less than 0 (zero) the statistics for all tables will never be cleared.

DBMaxHistoryDaysL4Datagram=:default=”DBMaxHistoryDays” UM=Days
For values greater than 0 this is the maximum number of days stored in the L4_DATAGRAM Database table. The date change will clear the traffic logs previously created based on the number of days set relative to the new date. If the value is less than 0 (zero) the statistics for all tables will never be cleared.

DBLoadHistorySessionActivity=:default=”true” UM=boolean
If set to false, the database will not be populated with the corresponding values.

DBLoadHistoryPoolQueuesActivity=:default=”true” UM=boolean
If set to false, the database will not be populated with the corresponding values.

DBLoadHistoryIncomingQueueHighWater=:default=”true” UM=boolean
If set to false, the database will not be populated with the corresponding values.

DBLoadHistoryL4TcpTcpSSL=:default=”true” UM=boolean
If set to false, the database will not be populated with the corresponding values.

DBLoadHistoryL7HttpHttps=:default=”true” UM=boolean
If set to false, the database will not be populated with the corresponding values.

DBLoadHistoryL4Datagram=:default=”true” UM=boolean
If set to false, the database will not be populated with the corresponding values.

DBLoadHistorySyslogEvent=:default=”true” UM=boolean

If set to false, the database will not be populated with the corresponding values.

DBQueryLimit=:default=”5000” UM=Records
It is the maximum number of records that can be extracted in queries to avoid overflow.

DBVarcharLimit=:default=”4000” UM=Byte
The maximum number of bytes that will be inserted into the VARCHAR column types. See LBL_DBNetworkedConfiguration.pdf manual for more details on this value for different database platforms.

DBMaxScriptBatch=:default=”200” UM=Byte
The maximum number of SQL batch scripts used to load the database.

9 OPLON®Traffic Monetizer statisticbrokerwebcachectrl.xml

(LBL_HOME)/procsProfiles/A05_LBLGoStatisticsWebCacheDWHCTR/conf/
statisticbrokerwebcachedwhctr.xml

This file describes the manager of the staging tables and their life cycle.

```
<serviceconf>
   <copyright>
   </copyright>
        <statisticbrokerwebcachedwhctr>
                <params>
                </params>
                <tables>
                    <id/>
                    …
                    …
                </tables>
        </statisticbrokerwebcachedwhctr>
</serviceconf>
```

<statisticbrokerwebcachedwhctr>

<params>

```
<serviceconf>
        <statisticbrokerwebcachedwhctr>
                <params
```

DBInitialConnectionPool=:default=”1”
Initial number of database connections.

DBMaxConnectionPool=:default=”1”
Maximum number of database connections. Upon exceeding this limit a new connection will still be opened but it will cease at the end of the request and will not be reused by the

connection pool.

DBDriver=:default=”oracle.jdbc.driver.OracleDriver”
Driver for management of the relational database.

DBProtocol=:default=”jdbc:oracle:”
The Protocol for the database connection string.

DBName=:default=””
The name of the database. Together with the DBProtocol determines the connection string to the DB.

DBLogin=:default=”null”
The login to connect to the DB.

DBPassword=:default=”null”
The password to connect to the DB.

<tables>

```
<serviceconf>
    <statisticbrokerwebcachedwhctr>
        <tables
```

The section lists the table of staging managed by OPLON®Traffic Monetizer .

statement=:default=”{call lbl_admin.consmanager(?,?,?,?)}”
The SQL invoked statement to execute the control of the aggregation engines and of the life cycle of the staging data.

<id>

```
<serviceconf>
    <statisticbrokerwebcachedwhctr>
        <tables
            <id
```

name=:default=”null”
The name of the table to manage

partitionStep=:default=”2”
Number of minutes for the collection of a partition.

flashForward=:default=”5”
Number of future partitions.

rememberBack=:default=”0”
Number of days of retention of analytical statistics prior to deletion.

10 OPLON®Traffic Monetizer statisticbrokerwebcacheagr.xml

(LBL_HOME)/procsProfiles/D10_LBLDWH_xx_xxx_AGRxx/conf/
statisticbrokerwebcachedwhagr.xml

The file sets a data aggregator.

```
<serviceconf>
   <copyright>
   </copyright>
        <statisticbrokerwebcachedwhagr>
                <params>
                </params>
        </statisticbrokerwebcachedwhagr>
</serviceconf>
```

<statisticbrokerwebcachedwhagr>

<params>

```
<serviceconf>
        <statisticbrokerwebcachedwhagr>
                <params
```

DBInitialConnectionPool=:default=”1”
Initial number of database connections.

DBMaxConnectionPool=:default=”1”
Maximum number of database connections. Upon exceeding this limit a new connection will still be opened but it will cease at the end of the request and will not be reused by the connection pool.

DBDriver=:default=”oracle.jdbc.driver.OracleDriver”
Driver for management of the relational database.

DBProtocol=:default=”jdbc:oracle:”
The Protocol for the database connection string.

DBName=:default=””
The name of the database. Together with the DBProtocol determines the connection string to the DB.

DBLogin=:default=”null”
The login to connect to the DB.

DBPassword=:default=”null”
The password to connect to the DB.

aggregatorID =: default= "null"
Aggregator Identifier.

table=:default=”null”
Table Name to aggregate (ex,: L7_HTTP_HTTPS)

partitionStatementStart=:default=”{call lbl_admin.partition_to_consolidate(?,?,?,?,?)}”
Start registration statement

aggregatorStatement=:default=”null”
Aggregator (e.g. : {call lbl_l7.cons_l7_fact_01_template(?)})

partitionStatementEnd=:default=”{call lbl_admin.set_partition_consolidated(?,?,?)}”
Sql end statement to call

11 OPLON® Tables logging

The persistence of the traffic data in thestatistical database is created through the following tables:

- SESSION_ACTIVITY
  usage analysis of the sessions.
- L7_HTTP_HTTPS
  persistence of information of HTTP and HTTPS traffic
- L4_TCP_TCPSSL
  persistence of the information of TCP traffic, and TCP as the SSL terminator.
- L4_DATAGRAM
  persistence of information of UDP and multicast traffic.
- POOL_QUEUES_ACTIVITY
  usage analysis of the pools of resolution for the service requests.
- INCOMING_QHIGHWATER_LEVEL
  usage analysis of the filling of the queue of inbound connections.
- SYSLOG_EVENT storage of log messages from OPLON®AAI. infrastructure.

Table SESSION_ACTIVITY

Table containing the snapshots of the status of routing sessions at 10” intervals.

RECORD_TYPE	int	6	Record Type
VRRP_HOST_NAME	varchar(4000)		Name of the host of the VRRP service. This value associated with the VRRP_PORT_NUMBER, define the balancing instance for the consolidation of traffic data. If Platform Edition see Reference Guide parameter uniqueContextID in iproxy.xml
VRRP_PORT_NUMBER	int		This value associated with the VRRP_HOST_NAME, define the balancing instance for the

			consolidation of traffic data.
END_POINTS_GROUPING	varchar(4000)		The name of the group of endpoints
DOMAIN_REQUEST	varchar(4000)		The domain associated with the session
COMMAND	varchar(4000)		EMPTY
URI_PATH_REQUEST	varchar(4000)		EMPTY
RESPONSE_CODE	int		0
END_POINT_HOST_NAME	varchar(4000)		Name of the host of the endpoint associated with the session
END_POINT_PORT_NUMBER	int		Port number of the host of endpoint associated with the session
END_POINT_URI_PATH	varchar(4000)		The URIPath of context in processing
USER_ID	varchar(4000)		Future Use
CLIENT_ADDRESS	varchar(4000)		The client address contains the address of the client that requested the service. Ex.: 192.168.43.150 If a transmission at layer 7 HTTP/S and system has been configured with management of the entity X-Forwarded-For (HTTP headers) through the parameter xforwardedfor= "true" in listener in iproxy.xml the value of the whole IP chain will be transferred. Obviously LBL can ensure only the last element of the chain since the other elements are useful only for statistical purposes or they have been populated by other infrastructure tools such as proxies. Ex.: 192.168.32.115,192 .168.41.10,192.168.43.150
THIS_DATE	date		Actual date and time set to 00:00:00
THIS_TIME	time		Actual time and date set to 01-01-1970
NUMBER_OF_ACTIVE_SESSIONS	int		The number of sessions relating to the grouping key (in red)

The keys used are:

- index K1_SESSION_ACTIVITY ON SESSION_ACTIVITY (THIS_DATE);

- index K2_SESSION_ACTIVITY ON SESSION_ACTIVITY (THIS_TIME);

Table L7_HTTP_HTTPS

This table contains the traffic of HTTP and HTTPS (SSL) protocols.

RECORD_TYPE	int	0=HTTP 1=HTTPS	Record Type
VRRP_HOST_NAME	varchar(4000)		Name of the host of the VRRP service. This value associated with the VRRP_PORT_NUMBER, define the balancing instance for the consolidation of traffic data. If Platform Edition see Reference Guide parameter uniqueContextID in iproxy.xml
VRRP_PORT_NUMBER	int		This value associated with the VRRP_HOST_NAME, define the balancing instance for the consolidation of traffic data.
END_POINTS_GROUPING	varchar(4000)		The name of the group of endpoints
DOMAIN_REQUEST	varchar(4000)		The domain requested by the client
COMMAND	varchar(4000)		HTTP cpmmand requested by the cient (GET, POST, etc)
URI_PATH_REQUEST	varchar(4000)		Path referring to the domain requested by the client. This value returns the result of the request after rewriting if it exists. The actual request that is made on the backend.
URI_PATH_REQUEST_ORG	varchar(4000)		Original Path referred to the domain requested by the client before any rewrite
CONTENT_TYPE_REQUEST	varchar(4000)		Content type in request
CONTENT_TYPE_RESPONSE	varchar(4000)		Content type in response
RESPONSE_CODE	int		Response CODE HTTP sent from the server to the client in response to request
END_POINT_HOST_NAME	varchar(4000)		Name of the host on which the request for service was executed
END_POINT_PORT_NUMBER	int		Port number of the host on which the request for service was executed
END_POINT_URI_PATH	varchar(4000)		The URIPath of context on which the request was processed
USER_ID	varchar(4000)		This column contains the information of the profiled and authenticated user. The information is a sum of characteristic

			elements of a basic authentication (Basic Authentication) and the authorization arising from digital certificate. This column contains respectively: If authorisation with digital certificate the Subject with an additional value stating the Serial Number of the certificate ex.: “CN=clientname, OU=clientlob, O=clientcompany, L=clientcountry, ST=clientdistrict, C=IT, SERIAL=1282479557” If basic authentication “BASIC=usr1” “Digest=usr2” “AWS= usr3” In the case both are credentials are present the resulting value will be : “BASIC=usr1, CN=clientname, OU=clientlob, O=clientcompany, L=clientcountry, ST=clientdistrict, C=IT, SERIAL=1282479557”
CLIENT_ADDRESS	varchar(4000)		The client address contains the address of the client that requested the service. Ex.: 192.168.43.150 If a transmission at layer 7 HTTP/S and system has been configured with management of the entity X- Forwarded-For (HTTP headers) through the parameter xforwardedfor= "true" in listener in iproxy.xml the value of the whole IP chain will be transferred. Obviously LBL can ensure only the last element of the chain since the other elements are useful only for statistical purposes or they have been populated by other infrastructure tools such as proxies. Ex.: 192.168.32.115,192 .168.41.10,192.168.43. 150
USER_AGENT	varchar(4000)		Value User-agent in HTTP HEADER
COOKIES	varchar(4000)		Cookie Value in HTTP HEADER
REFERER	varchar(4000)		Referer Value in HTTP HEADER
URI_PARAMETERS	varchar(4000)		Value with parameters and/or query string. The parameters defined in the URL by ; character and the query string defined by the character ? are normalized into a single

			value separated by &. Ex: /aaa/bbb;c1=aaa&c2=bbb?q1=ccc&q2=dddd VALUE IN URI_PARAMETERS c1 =aaa&c2 =bbb&q1 =ccc&q2 =dddd
PROTOCOL_VERSION	varchar(4000)		HTTP/1.0 HTTP/1.1
INCOMING_ADDRESS	varchar(4000)		Local Address on which the connection was accepted
INCOMING_PORT_NUMBER	int		Local Port on which the connection was accepted
THIS_DATE	Dates		Actual date and time set to 00:00:00
THIS_TIME	Time		Actual time and date set to 01-01-1970
COUNTER	bigint		Number of operations summed up in this record (if the default has not been changed is the number of operations carried out in its temporal window in the last 10 ")
RESPONSE_TIME	bigint		The response time of the service being requested (This value is relative to traffic of the last 10". For an average value divide by the COUNTER). Value expressed in nanoseconds.
LAP_TIME_A	bigint		Time between the end of the reading of the client HEADER and the beginning of the connection to the endpoint (This value is relative to traffic of the last 10". For an average value divide by the COUNTER). Value expressed in nanoseconds.
LAP_TIME_B	bigint		Connection time to the endpoint (This value is relative to traffic of the last 10". For an average value divide by the COUNTER). Value expressed in nanoseconds.
LAP_TIME_C	bigint		Time between the connection established at the endpoint and the beginning of the reading of the response HEADER. If the client sends the body it is the transmission time of the body from the client to the endpoint. (This value is relative to traffic of the last 10". For an average value divide by the COUNTER). Value expressed in nanoseconds.
LAP_TIME_D	bigint		Time to read the HEADER of the endpoint (This value is relative to traffic of the last 10". For an average value divide by the COUNTER). Value expressed in

			nanoseconds.
LAP_TIME_E	bigint		Time between the end of the reading of the HEADER and the end of the data in response (This value is relative to traffic of the last 10". For an average value divide by the COUNTER). Value expressed in nanoseconds.
HEADER_LENGTH_FROM_CLIENT	bigint		The length of the request HEADER from the clients to the endpoints (This value is relative to traffic of the last 10". For an average value divide by the COUNTER). Value expressed in nanoseconds.
BYTES_SENT_FROM_CLIENT	bigint		Total transferred including the HEADER from the client to the endpoints (This value is relative to traffic of the last 10". For an average value divide by the COUNTER). Value expressed in nanoseconds.
HEADER_LENGTH_FROM_END_POINT	bigint		The length of the response HEADER from endpoints to clients (This value is relative to traffic of the last 10". For an average value divide by the COUNTER). Value expressed in nanoseconds.
BYTES_SENT_FROM_END_POINT	bigint		Total transferred including the HEADER from endpoints to clients (This value is relative to traffic of the last 10". For an average value divide by the COUNTER). Value expressed in nanoseconds.

The keys used are:

- index K1_L7_HTTP_HTTPS ON L7_HTTP_HTTPS (THIS_DATE);
- index K2_L7_HTTP_HTTPS ON L7_HTTP_HTTPS (THIS_TIME);

Table L4_TCP_TCPSSL

As for the preceding table this table contains the traffic data relating to the activities of the TCP connector and TCP as the SSL terminator. The initial structure is very similar to the HTTP/S and differentiates itself then on traffic data

RECORD_TYPE	int	2=TCP 3=SSL	Record Type
VRRP_HOST_NAME	varchar(4000)		Name of the host of the VRRP service. This value associated with the VRRP_PORT_NUMBER, define the balancing instance for the consolidation of traffic data. If Platform Edition see Reference Guide parameter uniqueContextID in iproxy.xml
VRRP_PORT_NUMBER	int		This value associated with the VRRP_HOST_NAME, define the balancing instance for the consolidation of traffic data.
END_POINTS_GROUPING	varchar(4000)		The name of the group of endpoints
DOMAIN_REQUEST	varchar(4000)		If WebSocket it is the host requested at layer 7
COMMAND	varchar(4000)		Identifies the data flow: CLIENT_FLOW from the client to the endpoints ENDPOINT_FLOW from endpoints to the client WebSocket: WSCLIENT_FLOW from the client to the endpoints WSENDPOINT_FLOW from endpoints to the client
URI_PATH_REQUEST	varchar(4000)		If WebSocket it is the URIPath requested at layer 7
RESPONSE_CODE	int		0
END_POINT_HOST_NAME	varchar(4000)		Name of the host on which the service request was executed
END_POINT_PORT_NUMBER	int		Port number of the host on which the service request was executed
END_POINT_URI_PATH	varchar(4000)		If WebSocket it is the URIPath requested at layer 7
USER_ID	varchar(4000)		If SSL client authentication contains the Subject and the additional values of the certificate bearing the Serial Number eg.: "CN=clientname, OU=clientlob, O=clientcompany, L=clientcountry, ST=clientdistrict, C=IT, SERIAL=1282479557"
CLIENT_ADDRESS	varchar(4000)		The client address contains the address of the client that requested the service. Ex.: 192.168.43.150

COOKIES	varchar(4000)		Unique value associated with the connection. This value is populated with the value LBLCOLOR to identify a single connection at layer 4. To activate the population of this value it is necessary to set the parameter **distinguishsingleconnection** ="**true** " in the **<bind>** section of the parameters file iproxy.xml If WebSocket contains the cookise of the layer 7 HTTP/S connection
INCOMING_ADDRESS	varchar(4000)		Local Address in which the connection is accepted
INCOMING_PORT_NUMBER	int		Local Port on which the connection is accepted
THIS_DATE	dates		Actual date and time set to 00:00:00
THIS_TIME	time		Actual time and date set to 01-01-1970
COUNTER	bigint		Number of operations summed up in this record (if the default has not been changed is the number of operations carried out in its temporal window in the last 10 ")
BYTES_FORWARDED	bigint		The number of bytes exchanged according to flow (client>endpoint or endpoint>client) relative to the grouping key (in red)
START_ADV_TIME	bigint		The point at which the first buffer/character is detected expressed in nanoseconds
END_ADV_TIME	bigint		Te point at which the flush of the last buffering stream occurs expressed in nanoseconds
TOTAL_ADV_TIME	bigint		The total time taken to forward the information (END_ADV_TIME - START_ADV_TIME) expressed in nanoseconds

The keys used are:

- index K1_L4_TCP_TCPSSL ON L4_TCP_TCPSSL (THIS_DATE);
- index K2_L4_TCP_TCPSSL ON L4_TCP_TCPSSL (THIS_TIME);

Table L4_DATAGRAM

As for the preceding table this table contains the traffic data relating to the activities of the TCP connector and TCP as the SSL terminator. The initial structure is very similar to the HTTP/S and differentiates itself then on traffic data

RECORD_TYPE	int	10 =UDP 11 =MULTICAST	Record Type
VRRP_HOST_NAME	varchar(4000)		Name of the host of the VRRP service. This value associated with the VRRP_PORT_NUMBER, define the balancing instance for the consolidation of traffic data. If Platform Edition see Reference Guide parameter uniqueContextID in iproxy.xml
VRRP_PORT_NUMBER	int		This value associated with the VRRP_HOST_NAME, define the balancing instance for the consolidation of traffic data.
END_POINTS_GROUPING	varchar(4000)		The name of the group of endpoints
DOMAIN_REQUEST	varchar(4000)		EMPTY
COMMAND	varchar(4000)		Identifies the data flow: always empty because the flow is always from the client to the endpoints
URI_PATH_REQUEST	varchar(4000)		EMPTY
RESPONSE_CODE	int		0
END_POINT_HOST_NAME	varchar(4000)		Name of the host on which the report was drafted a request for service
END_POINT_PORT_NUMBER	int		Port number of the host on which the report was drafted a request for service
END_POINT_URI_PATH	varchar(4000)		EMPTY
USER_ID	varchar(4000)		Future Use
CLIENT_ADDRESS	varchar(4000)		The client address contains the address of the client that requested the service. Ex.: 192.168.43.150
INCOMING_ADDRESS	varchar(4000)		Local Address in which the connection is accepted
INCOMING_PORT_NUMBER	int		Local Port on which the connection is accepted
THIS_DATE	dates		Actual date and time set to 00:00:00

THIS_TIME	time		Actual time and date set to 01-01-1970
COUNTER	bigint		Number of operations summed up in this record (if the default has not been changed is the number of operations carried out in its temporal window in the last 10 ")
BYTES_FORWARDED	bigint		The number of bytes exchanged

The keys used are:

- index K1_L4_DATAGRAM ON L4_DATAGRAM (THIS_DATE);
- index K2_L4_DATAGRAM ON L4_DATAGRAM (THIS_TIME);

Table POOL_QUEUES_ACTIVITY

This table contains 10" interval snapshots of the status of activities of protocol resolvers. Its interpretation cannot be used as a summary of the activities of the day but as instant statistical data relative to the date and time listed on the records. These snapshots can be used to view, on a timebased axis, activity status and the peaks of use for a 10" interval.
This table contains two types of snapshots, one relating to the state of activity in the protocol resolvers and the other relating to the state of "committed" protocol resolvers.

RECORD_TYPE	int	4 =ACTIVITY 5 =COMMITTED	Record Type
VRRP_HOST_NAME	varchar(4000)		Name of the host of the VRRP service. This value associated with the VRRP_PORT_NUMBER, define the balancing instance for the consolidation of traffic data. If Platform Edition see Reference Guide parameter uniqueContextID in iproxy.xml
VRRP_PORT_NUMBER	int		This value associated with the VRRP_HOST_NAME, define the balancing instance for the consolidation of traffic data.
END_POINTS_GROUPING	varchar(4000)		The name of the group of endpoints
DOMAIN_REQUEST	varchar(4000)		The domain processing on the resolver
COMMAND	varchar(4000)		EMPTY
URI_PATH_REQUEST	varchar(4000)		EMPTY
RESPONSE_CODE	int		0

END_POINT_HOST_NAME	varchar(4000)		Name of the host on which the report was drafted a request for service
END_POINT_PORT_NUMBER	int		Port number of the host on which the report was drafted a request for service
END_POINT_URI_PATH	varchar(4000)		URIPath in context of processing
USER_ID	varchar(4000)		Future Use
CLIENT_ADDRESS	varchar(4000)		The client address contains the address of the client that requested the service. Ex.: 192.168.43.150 If a transmission at layer 7 HTTP/S and system has been configured with management of the entity X- Forwarded-For (HTTP headers) through the parameter xforwardedfor= "true" in listener in iproxy.xml the value of the whole IP chain will be transferred. Obviously LBL can ensure only the last element of the chain since the other elements are useful only for statistical purposes or they have been populated by other infrastructure tools such as proxies. Ex.: 192.168.32.115,192 .168.41.10,192 .168.43.150
THIS_DATE	Dates		Actual date and time set to 00:00:00
THIS_TIME	Time		Actual time and date set to 01-01-1970
NUMBER_OF_BUSY_CONSUMER	int		The number of active consumers at the instant THIS_DATE and THIS_TIME relating to grouping key (in red)

The keys used are:

- index K1_POOL_QUEUES_ACTIVITY ON POOL_QUEUES_ACTIVITY (THIS_DATE);
- index K2_POOL_QUEUES_ACTIVITY ON POOL_QUEUES_ACTIVITY (THIS_TIME);

Table INCOMING_QHIGHWATER_LEVEL

The table that records a 10” snapshot of the “fill” level of the incoming requests queue connections. This table is important because in relation to the number of protocol resolvers it is an indicator of a possible DoS attack or insufficient resources to process the load of requests.

RECORD_TYPE	int	7	Record Type
VRRP_HOST_NAME	varchar(4000)		Name of the host of the VRRP service. This value associated with the VRRP_PORT_NUMBER, define the balancing instance for the consolidation of traffic data. If Platform Edition see Reference Guide parameter uniqueContextID in iproxy.xml
VRRP_PORT_NUMBER	int		This value associated with the VRRP_HOST_NAME, define the balancing instance for the consolidation of traffic data.
THIS_DATE	dates		Actual date and time set to 00:00:00
THIS_TIME	time		Actual time and date set to 01-01-1970
HIGH_WATER	int		The number of connection requests in the queue before waiting to be processed
HIGH_WATER_LEVEL	float		The calculation factor derived from: (100 * HIGH_WATER)/ACT_SESSIONS
HIGH_WATER_WARNING_LEVEL	float		The threshold in % beyond which a warning message is sent
HIGH_WATER_DANGER_LEVEL	float		The threshold in % beyond which a danger message is sent
ACT_SESSIONS	int		The current availability of protocol resolvers
MAX_CONCURRENT_SESSIONS	int		The maximum number of protocol resolvers

The keys used are:

- index K1_INCOMING_QUEUE_HL ON INCOMING_QUEUE_HIGHWATER_LEVEL (THIS_DATE);
- index K2_INCOMING_QUEUE_HL ON INCOMING_QUEUE_HIGHWATER_LEVEL (THIS_TIME);

Table SYSLOG_EVENT

This table collects all the messages from all processes of OPLON® A. A. I. . in complex environments and therefore is a useful tool that, if centralized, it can be used as a system to detect any abnormalities in the operations.

RECORD_TYPE	int	12	Record Type
VRRP_HOST_NAME	varchar(4000)		It is a string separated by @ containing: host name@monitor management url@server process name@absolute log dir@date YYYYMMDD@hostname_logfileSuffix
VRRP_PORT_NUMBER	int		
COOKIES	varchar(4000)		It is a value identifying the message always different and unique for es.: ID= " 16231623"
INCOMING_ADDRESS	varchar(4000)		Host name
THIS_DATE	Dates		Actual date and time set to 00:00:00
THIS_TIME	Time		Actual time and date set to 01-01-1970
Repetitions	bigint		Number of repetitions of the same message
TIME_SEQUENCE	bigint		Event timeline
SEVERITY	varchar(4000)		ERROR | WARNING | DEBUG | FATAL
JAVA_REL	varchar(4000)		Java release
LBL_REL	varchar(4000)		LBL release
MESSAGE_GRP	varchar(4000)		Name of the processing unit that generated the message
HOST_ID	varchar(4000)		Name of the host that generated the message
MESSAGE	varchar(4000)		Message
MONITOR_MNG_URL	varchar(4000)		
MONITOR_PROCESS_NAME	varchar(4000)		

The keys used are:

- index K1_SYSLOG_EVENT ON SYSLOG_EVENT (THIS_DATE);
- index K2_SYSLOG_EVENT ON SYSLOG_EVENT (THIS_TIME);

NOTE: At the start of the Monitor, the column is set with MONITOR_MNG_URL ** UNDEFINED ** Upon completion of the start process this column will correctly report the value of management
eg.: https://192.168.46.109:54443/WebRegister

NOTE1: For the Monitor process the MONITOR_PROCESS_NAME is set with ** MONITOR **

8

OPLON®Application Delivery Controller TCOHTTPUtils.xml

(LBL_HOME)/lib/tco/resources/net/http/TCOHTTPUtils.xml

Through the optional TCOHTTPUtils.xml file it is possible to intervene on the keep-alive behavior depending on the content of the HTTP HEADER. In particular it is possible to command a close of the channel at the end of the response based on a value that is contained in the user-agent and response code.

```
<root>
 <copyright>
 </copyright>
 <TCOHTTPUtil>
  <uaKeepAliveEvaluation>
   <userAgent/>
   ...
  </uaKeepAliveEvaluation>
 </TCOHTTPUtil>
</root>
```

<TCOHTTPUtil>

<uaKeepAliveEvaluation>

<userAgent>

```
<root>
        <TCOHTTPUtil>
                <uaKeepAliveEvaluation>
                        <userAgent
```

value=:default value=””
The value to be searched for within the user-agent in the HTTP header.

ignoreCase=:default value=”false”
Defines case sensitivity of match within the string. If true, the comparison does not differentiate between lowercase and uppercase characters.

compareType=:default value=”0”
Indicates the type of search within the string:

- 0=contains (default)
- 1=equals
- 2=startsWith
- 3=endsWith
- 4=matches - regex compare

returnCode=:default value=”999”
The HTTP return code to which to apply the user-agent match. When the return code and user-agent match are equal, the channel will be closed at the end of the response to the client.

Example of a TCOHTTPUtils.xml file:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
   <copyright>
            TCOProject(tm) SoftwareLibrary

             This is a commercial software
 You shall not disclose such Confidential Information and shall use
  it only in accordance with the terms of the license agreement

                www.tcoproject.com
              mailto:info@tcoproject.com

    TCOProject is a trademark of F.Pieretti. All rights reserved.
   </copyright>
   <TCOHTTPUtil>
      <uaKeepAliveEvaluation>
        <!--
                0=contains (default)
                1=equals
                2=startsWith
                3=endsWith
                4=matches - regex compare
        -->
         <userAgent value="Mozilla/5.0" ignoreCase="false" compareType="2" returnCode="304"/>
         <userAgent value="Firefox/3."  ignoreCase="false" compareType="0" returnCode="200"/>
      </uaKeepAliveEvaluation>
   </TCOHTTPUtil>
</root>
```

9

OPLON®Application Delivery Controller Courtesy message

(LBL_HOME)/procsProfiles/A10_LBLGo/resources/html

The courtesy message is important because it informs users of an outage and if desired, communicate alternative actions.

The message is used by LBL LodBalancer when the requested service cannot be routed to any endpoints because they are unavailable due to maintenance or actual problems.

The message can be modified at runtime. Upon verification of the event, OPLON® LoadBalancer will always use the latest version or, in absence of the file, the default message.

In order to customize the message simply create the directory:

- (LBL_HOME)/procsProfiles/A10_LBLGo/resources/html

In this directory create the file "messageNoEndPoint.html"

The resulting path will then be:

- (LBL_HOME)/procsProfiles/A10_LBLGo/resources/html/messageNoEndPoint.html

The file "messageNoEndPoint.html" must be a simple HTML file with no external references (for example, images or videos).

The following parameters within the HTML files are replaced with the appropriate message:

- %title% = Title (loaded by "LBL(r)LoadBalancer")
- %msg1% = The HTTP response code that will be displayed in red font
- %msg2% = The URIPath that will be displayed in black font
  eg.: NO HOST AVAILABLE FOR URI PATH

It is recommended that the HTML form be modified to reflect a cutom message but to leave, even if de-emphasized, the original message to understand the problem if not a scheduled down-time of the service but it in fact is a real problem in the backend service.

A compressed archive (zip) of default messages is available and can be downloaded from TCO's customer support web site (restricted area).

10 OPLON® notificationDir

With OPLON® S.A.A.I. it is possible to intervene from outside to communicate any malfunctions or scheduled maintenance of backend services. Take for example a situation in which the services being balanced are working fine and responding but resources assigned to them - database for example - are unavailable due to failure or maintenance. In this case, although the web services are actually available and replying to the connection, they are effectively out of service because their back end dependency is out of service.

For the reasons stated, in the file iproxy.xml associative reference names to groups/domains up to the single end-point can be assigned. If a file is created in the directory "(LBL_HOME)/lib/notificationDir" with the name "outOfOrder.associateName" the end-point or domain or the end-point group associated with that name will be out-of-order.

The package LBLLoadBalancer_xxxxxxxxx_009_00x_00x.zip contains two examples of health check programs provided in source form for running custom resources checks. Both programs are supplied with the startup process file for OPLON® LoadBalancer Monitor which is already present in the directory (LBL_HOME)/lib/confMonitor and thus immediately available for use. See OPLON® LoadBalancer manual, section How To Use plugins for a complete dissertation of the use of these sample programs.

(LBL_HOME)/lib/plugin/lblplugin/	Descrizione
Z99_LBLPluginHTTPCheckWithCmdStart.xml	Executes healthcheck of an HTTP address and checks the connectivity and the return code. At start and at change of state ready/not- ready and vice-versa can launch a program for performing procedures relevant to the detected anomaly.
Z99_LBLPluginNetworkCheckWithCmdStart.xml	Executes the connectivity healthcheck of a TCP/IP port address. At start and at change of state ready/not- ready and vice-versa can launch a program for performing procedures relevant to the detected anomaly.

```
<endpoints>
  <endPointsGrouping associateName="pippo pluto " enable="true">
    <virtualDomain associateName="paperino minni " portRewriting="true" enable="true">
       <endp address="wiletrbackend" port="8080" uriPath="/Flowers/album" enable="true"/>
       <endp address="roadtrbackend" port="8080" uriPath="/Flowers/album" enable="true"/>
       <endp address="wiletrbackend" port="8787" uriPath="/Flowers/album" enable="true"/>
       <endp address="roadtrbackend" port="8787" uriPath="/Flowers/album" enable="true"/>
        <endp address="wiletrbackend" port="8787" uriPath="/TCOProject" associateName=" qui   quo
qua   " enable="true"/>
       <endp address="roadtrbackend" port="8787" uriPath="/TCOProject" associateName=" qui   quo
qua   " enable="true"/>
        <endp address="wiletrbackend" port="8787" uriPath="/TCOProjectSrv" enable="true"/>
       <endp address="roadtrbackend" port="8787" uriPath="/TCOProjectSrv" enable="true"/>
        <endp address="wiletrbackend" port="8181" uriPath="/training" enable="true"/>
       <endp address="roadtrbackend" port="8181" uriPath="/training" enable="true"/>
       <endp address="wiletrbackend" port="8282" uriPath="/training" enable="true"/>
    ...
```

One can see that in the section"endPointsGrouping" 2 symbolic names were assigned "pippo" and "pluto", in the section "virtualDomain" another 2 symbolc names were assigned "paperino" and "minni" and only in the"endp" with uriPath="/TCOProject" were associated with the names "qui", "quo" and "qua".

If during the execution a file is created in the new directory:

(LBL_HOME)/lib/notificationDir/outOfOrder.quo
only the two endpoints with uriPath="/TCOProject" will be placed in out of order

If during the execution a file is created in the new directory:

(LBL_HOME)/lib/notificationDir/outOfOrder.minni
all end points with that domain will be placed in out of order

If during the execution a file is created in the new directory:

(LBL_HOME)/lib/notificationDir/outOfOrder.pippo
all end points with that group will be placed in out of order

This method allows the system administrators to use any tool to check the services and to enable or disable entire groups of services by simply creating/deleting a file.

This feature adds the ability to logically place the entire balancing service out of order. This feature is activated in the same way as the previous one, that is by creating a file with the extension name of the balancer instance contained in the file systemsmonitor_m.xml:

File parameters:
(LBL_HOME)/lib/conf/systemsmonitor_m.xml

Fragment of file parameters:

```
<systemsmonitor_m>
```

```
<params
        systemsMonitorGroup="systemsMonitorGroup"   <=== Nome
        notificationDir="build/classes/notificationDir"
        monitorTimer="100....
```

To put this service in OutOfOrder simply create a file with name:

- outOfOrder.systemsMonitorGroup

As soon as the load balancer instance detects the file existence it places itself in OutOfOrder removing the virtual addresses as well. To restore the situation, simply delete the file and the load balancing instance will resume its role in the balancing pool.

DISABLING OF SERVICES

The file with the prefix outOfOrder establishes a service as no longer available and continually reports the unavailability through the log and, if enabled, mail and/or HTTP post.

However, there are cases that require endpoint disabling without the reporting of the event. These cases are very frequent in cases of services in fail-over or business-continuity, fast disaster recovery or disaster recovery where the need to close a channel in advance is very important.

To disable a group of endpoints without reporting the event simply create an empty file with the disable prefix.

- eg.: (LBL_HOME)/lib/notificationDir/disable.quo

- **ATTENTION**: As is the case for the outOfOrder one can disable an entire group of services based on the location of the associative name.

---

11 OPLON® statisticCacheHistory

With the introduction of statistics methods have been introduced to control the flow during runtime.

In order to activate the exclusions simply create a file in the cache directory with the following names for their respective disablement:

Creation and population of cache levels 1 and 2 statistics
Name of the file to stop the populationof the statistics

- (LBL_HOME)/procsProfiles/XXX_procName/statisticCacheHistory/stopLoadStat

ATTENTION: Stopping the population of the statistics at this level stops generating cache too. Data generated during the stoppage will not be recoverable

The delivery of the balancing process statistics to the application server StatisticsBorokerWebCache.
File name to stop the delivery of statistics to web cache:

- (LBL_HOME)/procsProfiles/XXX_procName/statisticCacheHistory/stopSendStat

This stoppage does not affect subsequent reactivation and delivery of statistics generated up to that point if, within the limits of cache level 2 statistics retention of 1 day as per default.

The trace on the log of the two flow controls for deactivationof the statistics service
Name of file to stop the log warning messages about the stoppage of the statistics

- (LBL_HOME)/procsProfiles/XXX_procName/statisticCacheHistory/stopNoLogError

12

OPLON®Application Delivery Controller

forceIncomingConnectionToWait

It is possible to stop emptying the incoming connections queue. This is event very important as it provides the ability to freeze the communications for a period of time – naturally a very limited time - but which will create synchronization points.

To create this event and therefore pause the emptying of the queue, simply set the file:

- (LBL_HOME)/lib/notificationDir/forceIncomingConnectionToWait

To resume the emptying of the incoming connections queue, simply delete the file.

ATTENTION: This file must be used prudently and for a short period of time. Executing this event for too long can cause Out of Memory issues.

12 OPLON®DNS&Proxy Manager dnsmanager.xml

This parameter file defines the rules of behavior and the positions of the template file of the product OPLON®dnsmanager. The file is formed by two sections; the first <params> describes the general variables while the second , <zone> that can be repeated UMltiple times, defines the zones of the DNS. Inside the second section <zone> are defined the namespaces and the verification conditions of the "vitality of services" <namespace>.

```
<serviceconf>
   <copyright>
   < /copyright>
         <dnsmanager>
                  <params>
                  < /params>
                  <zone>
                     <namespace>
                        <condition>
                        < /condition>
                    < /namespace>
                  < /zones>
         < /dnsmanager>
< /serviceconf>
```

<dnsmanager>

<params>

```
<serviceconf>
        <dnsmanager>
                <params
```

frequency =: default value= " 60000" UM=milliseconds
The frequency with which the dnsmanager performs verification of the vitality of associations names-services.

numRetryConnection =: default value= " 3"
The number of attempts to declare a service no longer active.

waitPerRetryConnection =: default value= " 300" UM=milliseconds
Time to wait between connection attempts and until numRetryConnection is reached.

createConnectionTimeOut=: default value= " 5000" UM=Milliseconds
Time-out of the connection attempt.

templateDir=: default value= "templatednsmanager" UM=directory
Directory where zone files templates are contained.

ATTENTION : This directory is interrogated during run-time and so it is a good idea not to modify the templates directly, but bring them in this directory only at the end of the writing in another directory.

templateSerialWithDate =: default value= "true"
This indicator is used to calculate the serial of the zone, such as pure progressive positive 32-bit or as the “talking” value expressed in "YYYYMMDDss" where "ss" is the progressive starting from 01.
If true the value is expressed with the indication of the date.

oneCommandForAllZones =: default value= "false"
If set to true performs a detailed comand for each zone by using the command of the zone itself. If false it executes a single command described in this chapter <params>.

reloadCommand =: default value=""
The reloadCommand is the name of the command invoked to reload the DNS due to a topology change of the associations names-addresses.
If BIND is used for DNS its value normally will be " /AbsoluteLocation/rndc reload " .If to be used with other DNS the important thing is that the return code is 0 (zero) for the successful operations and <> 0 (non-zero) for those that ended in failure. OPLON®DNS & Proxy Manager is supported directly if using the BIND DNS or Microsoft DNS.

reloadReverseNamespaceCommand =: default value=""
The reloadReverseNamespaceCommand is the name of the command invoked to reload the DNS due to a topology change of the associations names-addresses. This command in particular is used to reload/load the reverse resolution of addresses. With BIND this parameter will be practically never used since the rndc command is able to perform a reload of the zone namespace+reverse namespace. In the Microsoft environment DNS command batch files are generated during the runtime by OPLON®DNS & Proxy Manager and therefore they must be performed separately to set the resolution of the name- >address and address- >name.

sysCommandTimeOut=: default value= " 10000" UM=Milliseconds
Indicates the time required to declare time-out of a system command. If the command exceeds this limit an abort command is executed, and subsequently control is released to the

application.

sysCommandCheckRate =: default value= " 300" UM=Milliseconds
The frequency with which to check the state of the system command.

sysCommandRemoteURL= : default= "https://localhost:5992/SysCommand"
The URL of the service to run system commands

<zone>

```
<serviceconf>
    <dnsmanager>
        <zone>
```

This section identifies a given zone of the DNS. This section may be repeated in order to simultaneously manage multiple zones.

enable =: default= "false"
If true this section for the zone is active. If false the section is not taken into account.

templateSerialWithDate =: default value= " <params templateSerialWithDate>"
This indicator is used to calculate the serial of the zone, such as pure progressive positive 32-bit or as the “talking” value expressed in "YYYYMMDDss" where "ss" is the progressive starting from 01.
If true the value is expressed with the indication of the date.

reloadCommand =: default value= " <params reloadCommand>"
The reloadCommand is the name of the command invoked to reload the DNS due to a topology change of the associations names-addresses.
If BIND is used for DNS its value normally will be "/AbsoluteLocation/rndc reload" .If to be used with other DNS the important thing is that the return code is 0 (zero) for the successful operations and <> 0 (non-zero) for those that ended in failure. OPLON®DNS & Proxy Manager is supported directly if using the BIND DNS or Microsoft DNS.

reloadReverseNamespaceCommand =: default value= " <params reloadReverseNamespaceCommand>"
The reloadReverseNamespaceCommand is the name of the command invoked to reload the DNS in the face of a topology change of the associations names-addresses. This command in particular is used to reload/load the reverse resolution of addresses. If BIND is used for DNS, this parameter will be practically never used since the rndc command is able to perform a reload of the zone namespace+reverse namespace. In the Microsoft DNS environment command batch files are generated during the runtime by OPLON®DNS & Proxy Manager and therefore must be performed separately to set the resolution of the name->address and address- >name.

namespaceFile =: default=""
It is the physical location of the target zone at the disposal of the DNS. This file will be completely rebuilt by OPLON®DNS & Proxy Manager starting from the template file.

ATTENTION : This directory is interrogated during run-time and so it is a good idea not to modify the templates directly, but bring them in this directory only at the end of the writing in another directory.

namespaceTemplateFile =: default=""
The name of the template file of the zone. This file will be used to construct the zone file available to the DNS.

ATTENTION : This directory is interrogated during run-time and so it is a good idea not to modify the templates directly, but bring them in this directory only at the end of the writing in another directory.

namespaceReverseFile =: default=""
The physical location of the target zone at the disposal of the DNS for the reverse resolution (from address(es)/to the name). This file will be completely rebuilt by OPLON®DNS & Proxy Manager from template file.

ATTENTION : This directory is interrogated during run-time and so it is a good idea not to modify the templates directly, but bring them in this directory only at the end of the writing in another directory.

namespaceReverseTemplateFile = : default=""
The name of the template file of the zone for reverse resolution of (address(es)/to the name). This file will be used to construct the zone file available to the DNS.

ATTENTION : This directory is interrogated during run-time and so it is a good idea not to modify the templates directly, but bring them in this directory only at the end of the writing in another directory.

namespaceNegativeCachePrevention = : default= "the first namespace"
The namespace that will be used if all addresses/services are not available. This namespace is extremely important because it prevents the occurrence of negative cache on the client (negative-cache). If no value is provided, it will be set as default with the first namespace of the zone.
A typical value of this field could resemble "www IN A192.168.43 .155 ". This value may be different from the value of the zones and can be used to provide information. When using MS DNS the value corresponds to the fragment of batch file that is used to control the addition of the CNAME
Ex.:

```
dnscmd /recordadd tcoproject.dev www /CreatePTR 10 A 192.168.43.155
```

namespaceNegativeCachePreventionInactive = : default=""
If it exists the value to use in the composition of the file/batch in case of negative result of the HealthCheck. In the use of MS DNS is normally set with the cancellation of the CNAME.
Ex.:

```
dnscmd /recorddelete tcoproject.dev www A 192.168.43.155 /F
```

namespaceReverseNegativeCachePrevention = : default= "the first namespaceReverse"
The namespaceReverse that will be used if all addresses/services are not available. This namespace is extremely important because it prevents the occurrence of negative cache on the client (negative-cache). If no value is provided, it will be set as default with the first namespace of the zone. A typical value of this field could resemble "155 PTR www.tcoproject.dev. ". This value may be different from the value of the zones and can be used to provide information.

namespaceReverseNegativeCachePreventionInactive = : default=""
If it exists it is the value to use in the composition of the file/batch in case of negative result of the HealthCheck. In the use of MS DNS can be set with the cancellation of the reverse lookup
Ex.:

```
dnscmd /recorddelete 43.168.192 .in-addr.arpa 155.43.168.192 PTR /F
```

This command can also be omitted dependending on some evaluations. The first is that it is possible that in the face of several CNAMEs being associated with the same addresses and thus a superset of reverse lookup abstract was probably set up with respect to the CNAME (e.g. : app1-vip app2-vip etc. ..). The other consideration is that MS DNS from Win2008 onwards is being used and the CNAME was created with the option /CreatePTR the cancellation of the reverse lookup is automatic.

<namespace>

```
<serviceconf>
        <dnsmanager>
                <zone>
                        <namespace
```

The section namespace describes each association name-address and address-name.

enable = : default= "false"
If true this section namespace is active. If false the section is not taken into account.

address =: default value=""
The address on which the health check is performed.

ATTENTION: In the event of health check, do not set this value with a name by DNS definition but set it with an address (e.g. : 192.168.43.142). In the case of definition for other reasons, e.g. construction wpad javascript can also be set to the host name (e.g. : www.google.it .)

port =: default value= " 0"
The port on which the health check service responds. If <=0 a ICMP check is executed

login =: default value=""
User Name login for BASIC authentication.
It is possible to enter the login and password for the service on which to perform the health

check.

p**assword** =: default value=""
Login Password for basic authentication.

addressProxy =: default value=""
The address of the proxy for the connection of the health check.

ATTENTION: do not set this value with a name but set it with an address (e.g. : 192.168.43.142).

portProxy =: default value= " 0"
The proxy port of the health check.

loginProxy =: default value=""
Login for BASIC authentication to the proxy server.

passwordProxy =: default value=""
Password for Basic authentication to the proxy server.

SSL =: default value= "false"
If set to true performs the health check of the service through an SSL connection (HTTPS).

uriPath =: default value= " /HealthCheck"
The uriPath on which the health check service responds.

numRetryConnection =: default value= "default from <params>"
The number of attempts before declaring a service no longer active. If no value is provided takes the default from <params> section.

waitPerRetryConnection =: default value= "default from <params>"
Time to wait between connection attempts and until numRetryConnection is reached. If no value is provided takes the default from <params> section.

createConnectionTimeOut=: default value= "default from <params>"
Time to wait before declaring a timeout connection attempt. If no value is provided takes the default from <params> section.

namespace =: default value=""
The value of association name-address. An example of value could look like: "www IN A 192.168.43 .136 ". On MS DNS the following control value of association is normally used ex.:

- dnscmd /recordadd tcoproject.dev www /CreatePTR 10 A 192.168.43.136

Attention: /CreatePTR is valid only from Windows 2008 Server. The command for Windows 2003 Server is:
dnscmd /recordadd tcoproject.dev www A 192.168.43.136

namespaceInactive =: default value=""
The value of association name-address in the event of negative outcome of the HealthCheck. Is normally used on MS DNS and removes the association
Ex.:
dnscmd /recorddelete tcoproject.dev www A 192.168.43.136 /F

namespaceReverse =: default value=""
The value of association address-name. An example of a value could look like: "136 PTR www.tcoproject.dev. ". In the case of MS DNS the value corresponds to a command to insert new PTR record.

- dnscmd /recordadd 43.168.192 .in-addr.arpa www.tcoproject.dev PTR 192.168.43.136

namespaceReverseInactive =: default value=""
The value of association address-name. An example of value could look like: "136 PTR www.tcoproject.dev. ". Or for MS DNS the value could take the value corresponding to the command for the cancellation of the PTR. However the considerations made previously on the release and the context of use do apply such as situations where different CNAME are associated with the same addresses, or in the presence of Windows 2008 Server with the creation of the CNAME with the parameter /CreatePTR. This value will then not often be valued. In the case it is valued it could take the value of a cancellation command of the PTR ex.:

- dnscmd /recorddelete tcoproject.dev www A 192.168.43.136 /F

<condition>

```
<serviceconf>
    <dnsmanager>
        <zone>
            <namespace>
                <condition
```

The condition section, which can be repeated, describes the health check of the services if resulting as active do not allow the activation of the namespace to which they belong. These conditions can then be used to condition the availability of the association in architectures of Disaster Recovery where, for example, only if the main site is no longer operative then the association name-address of the secondary site can be set.

enable = : default= "false"
If true this section namespace is active. If false the section is not taken into account.

address =: default value=""
The address on which the health check is performed. CAUTION: do not set this value with a name in the event of health check by DNS definition but set it with an address (e.g. : 192.168.43.142). In the case of definition for other reaspn, e.g. construction wpad javascript can also be set with the host name (e.g. : www.google.com.it.)

port =: default value= " 0"
The port on which the health check service responds. If <=0 a ICMP check is executed

login =: default value=""
User Name login for BASIC authentication.
It is possible to enter the login and password for the service on which to perform the health check.

p**assword** =: default value=""
Login Password for basic authentication.

addressProxy =: default value=""
The address of the proxy for the connection of the health check.

portProxy =: default value= " 0"
The proxy port of the health check.

loginProxy =: default value=""
Login for BASIC authentication to the proxy server.

passwordProxy =: default value=""
Password for Basic authentication to the proxy server.

SSL =: default value= "false"
If set to true performs the health check of the service through an SSL connection (HTTPS).

uriPath =: default value= " /HealthCheck"
The uriPath on which the health check service responds.

numRetryConnection =: default value= "default from <params>"
The number of attempts before declaring a service no longer active. If no value is provided takes the default from <params> section.

waitPerRetryConnection =: default value= "default from <params>"
Time to wait between connection attempts and until numRetryConnection is reached. If no value is provided takes the default from <params> section.

createConnectionTimeOut=: default value= "default from <params>"
Time to wait before declaring a timeout connection attempt. If no value is provided takes the default from <params> section.

On the following page a complete example of a parameters file dnsmanager.xml.

Ex.: BIND

```
< ?XML version= "1.0 " encoding= "windows-1252 "Help >

<serviceconf>
 <copyright>
                OPLON(r)ADC

            This is a commercial software
You shall not disclose such Confidential Information and shall use
   It only in accordance with the terms of the license agreement
```

```
                Www.tcoproject.com
               Www.lblloadbalancer.com
              Mailto:info@oplon.net

  OPLON(r)ADC is built__LW_NL__on TCOProject(tm) SoftwareLibrary
LBL and TCOProject are trademarks of F. Pieretti. All rights reserved.
   < /copyright>
      <dnsmanager>
        <params
                frequency =" 60000 "
                numRetryConnection =" 3 "
                waitPerRetryConnection =" 300 "
                createConnectionTimeOut=" 30 "
                templateDir=" templatednsmanager"
                templateSerialWithDate =" true "
                reloadCommand =" C:/WORK1/BIN/NAMED/BIN/RNDC reload "
                sysCommandTimeOut=" 10000 "
                sysCommandCheckRate =" 300 "
                sysCommandRemoteURL=" http://localhost:5992/syscommand " >
        < /params>

        <zones enable = "true"
                namespaceFile = "C:/WORK1/BIN/NAMED/ETC/zoneS/LOCAL/TCOPROJECT dev.db . "
                namespaceTemplateFile =" tcoproject.dev.db.template "
                namespaceReverseFile ="
C:/WORK1/BIN/NAMED/ETC/zoneS/LOCAL/REV .43.168.192.in-addr.arpa"
                namespaceReverseTemplateFile =" rev.43.168.192.in -addr.arpa.template "
                namespaceNegativeCachePrevention =" www IN A 192.168.43.255 "
                namespaceReverseNegativeCachePrevention =" 255 PTR www.tcoproject.dev.' >
               <namespace enable =" true "
                        address =" 192.168.43.136 " port =" 80 " uriPath =" /HealthCheck " SSL =" false "
                        numRetryConnection =" 3 "
                        waitPerRetryConnection =" 300 "
                        createConnectionTimeOut=" 30 "
                        namespace =" www IN A 192.168.43.136 "
                        namespaceReverse =" 136 PTR www.tcoproject.dev.' >
                         <condition enable =" true "
                                address =" 192.168.43.142 " port =" 80 " uriPath =" /HealthCheck " SSL ="
false "
                                numRetryConnection =" 3 "
                                waitPerRetryConnection =" 300 "
                                createConnectionTimeOut=" 30 " / >
                        <condition enable =" true " address =" 192.168.43.144 " port =" 80 " uriPath ="
/HealthCheck " SSL =" false " />
               < /namespace>
               <namespace enable =" true "
                        address =" 192.168.43.138 " port =" 80 " uriPath =" /HealthCheck " SSL =" false "
                        numRetryConnection =" 3 "
                        waitPerRetryConnection =" 300 "
                        createConnectionTimeOut=" 30 "
                        namespace =" www IN A 192.168.43.138 "
                        namespaceReverse =" 138 PTR www.tcoproject.dev . " />
               <namespace enable =" true "
                        address =" 192.168.43.124 " port =" 80 " uriPath =" /HealthCheck " SSL =" false "
                        namespace =" www IN A 192.168.43.124 "
                        namespaceReverse =" 124 PTR www.tcoproject.dev . " / >
               <namespace enable =" true "
```

```
                    address =" 192.168.43.125 " port =" 80 " uriPath =" /HealthCheck " SSL =" false "
                    namespace =" www IN A 192.168.43.125 "
                    namespaceReverse =" 125 PTR www.tcoproject.dev . " / >
                <namespace enable =" true "
                    address =" 192.168.43.142 " port =" 80 " uriPath =" /HealthCheck " SSL =" false "
                    namespace =" www IN A 192.168.43.142 "
                    namespaceReverse =" 142 PTR www.tcoproject.dev . " / >
                <namespace enable =" true "
                    address =" 192.168.43.144 " port =" 80 " uriPath =" /HealthCheck " SSL =" false "
                    namespace =" www IN A 192.168.43.144 "
                    namespaceReverse =" 144 PTR www.tcoproject.dev . " / >
                < /zones>
        <Sysobserver>
        < /Sysobserver>
    < /dnsmanager>
< /serviceconf>
```

Ex.: MS DNS

```
< ?XML version= "1.0 " encoding= "windows-1252" ?>

<serviceconf>
        <copyright>
            OPLON(r)ADC

          This is a commercial software
You shall not disclose such Confidential Information and shall use
  It only in accordance with the terms of the license agreement

            Www.tcoproject.com
           Www.lblloadbalancer.com
          Mailto:info@oplon.net

  OPLON(r)ADC is built__LW_NL__on TCOProject(tm) SoftwareLibrary
LBL and TCOProject are trademarks of F. Pieretti. All rights reserved.
   < /copyright>
        <dnsmanager>
         <params
            frequency =" 60000 "
            templateDir=" templatednsmanager "
            templateSerialWithDate =" true "
            reloadCommand="C:\work1\bin\TCOProject\
LBLLoadBalancer_monitor_007_000_000RC001\lib\scriptDNSManager\reloadMSDns.bat"
            sysCommandRemoteURL=" https://localhost:5992/syscommand " >
        < /params>

        <zones enable =" true "                                    namespaceFile =" namespaceFile="C:\
work1\bin\TCOProject\LBLLoadBalancer_monitor_007_000_000RC001\lib\scriptDNSManager\
reloadMSDns.bat"
                namespaceTemplateFile="C:\work1\bin\TCOProject\
LBLLoadBalancer_monitor_007_000_000RC001\lib\templateDNSManager\www.dev.db.template"
            namespaceReverseFile="C:\work1\bin\TCOProject\
LBLLoadBalancer_monitor_007_000_000RC001\lib\scriptDNSManager\empty.bat"
            namespaceReverseTemplateFile="C:\work1\bin\TCOProject\
LBLLoadBalancer_monitor_007_000_000RC001\lib\templateDNSManager\empty.template"
namespaceNegativeCachePrevention="dnscmd /recordadd tcoproject.dev www /CreatePTR 10 A
10.10.252.125”
 namespaceNegativeCachePreventionInactive="dnscmd /recorddelete tcoproject.dev www A
10.10.252.125 /F">
```

```
                <namespace enable="true"
                  address="192.168.43.177" port="8080" uriPath="/" SSL="false"
                  namespace="dnscmd /recordadd tcoproject.dev www /CreatePTR 10 A 10.10.252.101"
                  namespaceInactive="dnscmd /recorddelete tcoproject.dev www A 10.10.252.101 /F"/>
                    <namespace enable="true"
                  address="192.168.44.177" port="8080" uriPath="/" SSL="false"
                  namespace="dnscmd /recordadd tcoproject.dev www /CreatePTR 10 A 10.10.252.110"
                  namespaceInactive="dnscmd /recorddelete tcoproject.dev www A 10.10.252.110 /F">
                <namespace enable="true"
                  address="192.168.45.177" port="8080" uriPath="/" SSL="false"
                  namespace="dnscmd /recordadd tcoproject.dev www /CreatePTR 10 A 10.10.252.111"
                  namespaceInactive="dnscmd /recorddelete tcoproject.dev www A 10.10.252.111 /F"/>
          </zone>

          <sysobserver>
          <service name="syslog" id="syslogdnsmanager"/>
          </sysobserver>
      </dnsmanager>
</serviceconf>
```

13 OPLON®IPNetworkCardRedundancy ipncr.xml

IP Network Card Redundancy is the service that enables redundancy of network interfaces. It is possible to perform the health check with both positive and negative responses and then based on the result move the address to another network interface. This service is available on all versions of OPLON® distributed to provide high availability IP capability.

It is possible to configure multiple groups of redundancy, which will be treated in a distinct manner. Each redundancy group is contained in the <floatingAddress> section with its own characteristics.

```
<serviceconf>
   <copyright>
   < /copyright>
         <ipncr>
                  <params>
                  < /params>
                  <floatingAddressesMgr>
                     <floatingAddress>
                        <floatingInterface>
                        < /floatingInterface>
                        ...
                        <healthCheckPolicy>
                          <healthCheck>
                          < /healthCheck>
                        < /healthCheckPolicy>
                        ...
                        <healthCheckConditionPolicy>
                          <healthCheck>
                          < /healthCheck>
                        < /healthCheckConditionPolicy>
                        ...
                     < /floatingAddress>
                  < /floatingAddressesMgr>
         < /ipncr>
< /serviceconf>
```

Example from a Solaris installation.

```
<floatingAddress enable="true"
          description="Private Network"
                address="192.168.47.113"
                netmask="255.255.255.0"
                resetOnShutdown="true">
          <floatingInterface device="e1000g1:1"
                deviceName="e1000g1:1"/>
          <floatingInterface device="e1000g3:1"
                 deviceName="e1000g3:1"/>
      <healthCheckPolicy>
            <healthCheck address="192.168.47.130" description="roadwinvistabench"/>
            <healthCheck address="192.168.47.131" description="wilewinvistabench"/>
      </healthCheckPolicy>
</floatingAddress>
```

With reference to the previous example we note that the floatingAddress 192.168.47.113 may be associated with the interfaces e1000g1:1 and e1000g3:1.
The process constantly checks that the interface that holds the floatingAddress is able to successfully perform the health check described in the appropriate section. In the case these should all result as failed, the system verifies the operation of the interface e1000g3:1 and in the event of a positive response assigns the floatingAddress to the last one.

It is possible to use multiple health check groups to differentiate the tests, for example, on different networks. In the example below a new Heath Check group was added. If a Health Check group should be completely unresponsive, a switch to the next interface will occur.

```
<floatingAddress enable="true"
                 description="Rete privata"
                 address="192.168.47.113"
                 netmask="255.255.255.0"
                 resetOnShutdown="true">
    <floatingInterface device="e1000g1:1"
                       deviceName="e1000g1:1"/>
    <floatingInterface device="e1000g3:1"
                       deviceName="e1000g3:1"/>
    <healthCheckPolicy>
        <healthCheck address="192.168.47.101" description="Sys A2 private"/>
        <healthCheck address="192.168.47.126" description="Sys A3 private"/>
    </healthCheckPolicy>
    <healthCheckPolicy>
        <healthCheck address="192.168.45.101" description="Sys A2 private"/>
        <healthCheck address="192.168.45.126" description="Sys A3 private"/>
    </healthCheckPolicy>
</floatingAddress>
```

In addition addresses can be set based on conditions such as the existence of a service. The condition can be positive, defned as services/addresses do exist, or negative, defined as services/addresses do not exist.

An example of setting conditions are set out below:

```
<floatingAddress enable="true"
                 description="Rete privata"
                 address="192.168.47.113"
                 netmask="255.255.255.0"
                 resetOnShutdown="true">
   <floatingInterface device="e1000g1:1"
                      deviceName="e1000g1:1"/>
   <floatingInterface device="e1000g3:1"
                      deviceName="e1000g3:1"/>
   <healthCheckPolicy>
       <healthCheck address="192.168.47.101" description="Sys A2 private"/>
       <healthCheck address="192.168.47.126" description="Sys A3 private"/>
   </healthCheckPolicy>
   <healthCheckPolicy>
       <healthCheck address="192.168.45.101" description="Sys A2 private"/>
       <healthCheck address="192.168.45.126" description="Sys A3 private"/>
   </healthCheckPolicy>
   <healthCheckConditionPolicy positiveCondition="true">
       <healthCheck address="www.tcoproject.dev" port="80" uriPath="/HealthCheck" SSL="false"
                    description="Sys A2 public"  createConnectionTimeOut="4000" numRetryConnection="4"/>
       <healthCheck address="192.168.45.103" port="22" description="Sys A2 public"
                    createConnectionTimeOut="4000" numRetryConnection="4"/>
       <healthCheck address="192.168.45.104" description="Sys A3 public"/>
   </healthCheckConditionPolicy>
```

In this case, upon verification of the health check condition, the section healthCheckConditionPolicy, with parameter positiveCondition= "true", indicates that if at least one of the services/addresses identified is active, the floatingAddress must be assigned to one of the two interfaces.

On the other hand, in the case of positiveCondition= "false" the floating address would be set only if the services/addresses associated with it did not meet the health check.

NOTE: It is only necessary that one service/address be active for the health check condition to be considered valid.

<ipncr>

```
<serviceconf>
    <ipncr>
```

<params>

```
<serviceconf>
    <ipncr>
        <params
```

frequency =: default value= " 10000" UM=Milliseconds
The frequency with which to verify status changes in processes or parameters.

sysCommandTimeOut =: default value= " 10000" UM=Milliseconds
Indicates the time required to declare a system command in time-out. If the command exceeds this limit an abort command is executed, and control is released to the application.

sysCommandCheckRate =: default value= "300" UM=Milliseconds
The frequency with which to check the state of the system command.

sysCommandRemoteURL =: default= "https://localhost:5992/syscommand"
The URL of the service to run system commands.

createConnectionTimeOut =: default value= "5000" UM=Milliseconds
Time to wait before declaring time-out of the connection attempt.

numRetryConnection =: default value= "3"
Number of connection attempts before declaring the resource OutOfOrder.

<floatingAddressesMgr>

```
<serviceconf>
        <ipncr>
                <floatingAddressesMgr>
```

This section contains the definitions of the floatingAddress

<floatingAddress>

```
<serviceconf>
        <ipncr>
                <floatingAddressesMgr>
                        <floatingAddress
```

enable =: default value= "false"
Enable / disable the loading and the assessment of the section.

description =: default value=""
Description of the floating address.

address =: default value=""
Address of the floatingAddress.

netmask =: default value=""
Netmask of the floating address.

resetOnShutdown =: default value= "true"
If true during the controlled shutdown of the service, the address is deleted from any network interface described in the <floatingInterface> section. If false no action is taken during shutdown of the service leaving network interfaces in the current state. It should be remembered that at startup of the service all floating addresses are eliminated from the network interfaces. The address(es) is reset only after an analysis of the health check conditions.

associateFItoHCP =: default value= "false"
In a situation with multiple network cards connected to each other through crosscable or switch and a floating address for each node, as in the diagram below, it is possible to insert the indicator "associateFItoHCP" (associate Floating Address to Health Check Policy) in the file ipncr.xml . This indicator associates the healthCheck group to the network card.

ATTENTION : With crosscable, backwards migration of the address must be controlled manually with a stop and start of the service in case of recovery/refresh of the physical interface.

By setting associateFItoHCP= "true" the health check will be associated with the interface in this way:

- First floating interface to first healthcheck group;
- Second floating interface to second healthcheck group:
- Third floating interface a third healthcheck group:

What is achieved is a migration of the address floating between the network cards.

Example of migration as a result of a failure:

```
Status A: All ok
=====================================================
NODE A                                                NODE B
+staticAddress--------------+  < --CrossCable-->   +staticAddress--------------+
192.168.43.101                                       192.168.43.100
+floatingAddress                                    +floatingAddress
192.168.48.108                                       192.168.48.107

+staticAddress--------------+   < --CrossCable-->  +staticAddress--------------+
192.168.44.101                                       192.168.44.100

+staticAddress--------------+   < --CrossCable-->  +staticAddress--------------+
192.168.45.101                                       192.168.45.100

Status B: NODE A first card failure
=====================================================
NODE A                                                NODE B
+staticAddress--------------+  < --CrossCable-->   +staticAddress--------------+
XXXXXXXXXXXXX                                        192.168.43.100

+staticAddress--------------+   < --CrossCable-->  +staticAddress--------------+
192.168.44.101                                       192.168.44.100
+floatingAddress                                     +floatingAddress
192.168.48.108                                       192.168.48.107
<====================Migration

+staticaddress--------------+   < --CrossCable-->  +staticAddress--------------+
192.168.45.101                                       192.168.45.100
```

Configuration of ipcnr.xml

```
<floatingAddress enable="true"
          description="Rete internet A"
          address="192.168.48.108"
          netmask="255.255.255.0"
          resetOnShutdown="true"
          associateFItoHCP="true">
         <floatingInterface enable="true"
                  device="PCI\\
VEN_10EC&amp;DEV_8139&amp;SUBSYS_813910EC&amp;REV_10\\4&amp;2E98101C&amp;0&amp;10F0"
                  deviceName="LBLPublic"/>
         <floatingInterface enable="true"
                  device="PCI\\
VEN_10EC&amp;DEV_8139&amp;SUBSYS_81391849&amp;REV_10\\4&amp;2E98101C&amp;0&amp;28F0"
```

```
                deviceName="LBLPrivate"/>
        <floatingInterface enable="true"
                device="PCI\\
VEN_10EC&amp;DEV_8139&amp;SUBSYS_813910EC&amp;REV_10\\4&amp;2E98101C&amp;0&amp;18F0"
                deviceName="LBLBackendMonitor"/>
        <healthCheckPolicy enable="true">
                <healthCheck enable="true" address="192.168.43.100" description="Sys A2 public"
createConnectionTimeOut="4000" numRetryConnection="3"/>
                <healthCheck address="192.168.48.107" description="Sys A3 public"/>
        </healthCheckPolicy>

        <healthCheckPolicy enable="true">
                <healthCheck enable="true" address="192.168.44.100" description="Sys A2 public"
createConnectionTimeOut="4000" numRetryConnection="4"/>
                <healthCheck address="192.168.48.107" description="Sys A3 public"/>
        </healthCheckPolicy>

        <healthCheckPolicy enable="true">
                <healthCheck enable="true" address="192.168.45.100" description="Sys A2 public"
createConnectionTimeOut="4000" numRetryConnection="4"/>
                <healthCheck address="192.168.48.107" description="Sys A3 public"/>
        </healthCheckPolicy>
 </floatingAddress>
```

<floatingInterface>

```
        <serviceconf>
                <ipncr>
                        <floatingAddressesMgr>
                                <floatingAddress>
                                        <floatingInterface
```

enable =: default value= "false"
Enable / disable the loading and the assessment of the section.

device =: default value=""
The identifier of the alias associated with a physical device for Unix/Linux and of the physical device for MS Windows. To obtain this name "devcon.exe " is used for MS Windows and "ifconfig" for Unix/Linux systems. Refer to the OPLON®ADC IP Network Card redundancy installation manual for more details about installing and setting up.

deviceName =: default value=""
The name of the device. On MS Windows must assume the value of the name visible in network resources. Refer to the OPLON®ADC IP Network Card redundancy installation manual for more details about installing and setting up.

The following are some possible configurations:

MS Windows:

```
<floatingInterface
device="PCI\\
VEN_8086&amp;DEV_1019&amp;SUBSYS_80F71043&amp;REV_00\\4&amp;3B3CB9B1&amp;0&amp;08
18"
deviceName="LBLPublic">
```

Linux:

```
<floatingInterface device="eth1:0"
                deviceName="eth1:0">
```

Solaris/OpenSolaris (7,8,9,10)

```
<floatingInterface device="iprb2:1"
                   deviceName="iprb2:1">
```

<healthCheckPolicy>

```
<serviceconf>
        <ipncr>
                <floatingAddressesMgr>
                        <floatingAddress>
                                <healthCheckPolicy
```

It is the section responsible for the health check of the addresses to determine the reachability of the selected network interface. If all the health checks configured in this section should give a negative result then the address will be migrated to another network interface. It is also possible to arrange several healthcheck groups by duplicating this section as described at the beginning of this chapter.

enable =: default value= "false"
Enable disable the loading and the assessment of section

description =: default value=""
Description of control

positiveCondition =: default value= "true"
Changes the direction of the control.
If true, and at least one control has a positive result then the entire section is positive.
If false and at least one control has a positive result then the entire section results as negative.

<healthCheck>

```
<serviceconf>
        <ipncr>
                <floatingAddressesMgr>
                        <floatingAddress>
                                <healthCheckPolicy>
                                        <healthCheck
```

enable =: default= "false"
Enable / disable the loading and the assessment of the section.

address =: default value=""
The address on which to perform the health check.

port =: default value= "0"
The port on which the health check service responds. If <=0 an ICMP check is run.

SSL =: default value= "false"
If true, it executes the health check service through an SSL connection (HTTPS).

uriPath =: default value=""

The URIPath on which the health check service responds. If the value is not present the health check will be run using a TCP connection.

numRetryConnection =: default value= "default from <params>"
Number of connection attempts before declaring the service inactive. If a value is not provided the system takes the default from the <params> section.

waitPerRetryConnection =: default value= " 300" UM=Milliseconds
Time to wait between connection attempts until the numRetryConnection is reached. If a value is not provided the system takes the default from the <params> section.

createConnectionTimeOut =: default value= "default from <params>"
Time to wait before declaring time-out of the connection attempt. If a value is not provided the system takes the default from the <params> section.

<healthCheckConditionPolicy>

```
<serviceconf>
        <ipncr>
                <floatingAddressesMgr>
                        <floatingAddress>
                                <healthCheckConditionPolicy
```

The section for the health check of the addresses to determine the operating condition of the network interface. If all the health checks provided in this section should give a negative result then the address will be set in one of the pre-establilshed network interfaces. It is also possible to arrange several healthcheck groups by duplicating this section as described at the beginning of this chapter. This control is conditional and does not determine the migration of the address from one interface to another but rather determines the possibility of setting the floating address on the basis of external conditions.

enable =: default value= "false"
Enable / disable the loading and the assessment of section.

Description =: default value=""
Description of control.

positiveCondition =: default value= "true"
Changes the direction of the control.
If true, and at least one control has a positive result then the entire section is positive.
If false and at least one control has a positive result then the entire section results as negative.

<healthCheck>

```
        <serviceconf>
                <ipncr>
                        <floatingAddressesMgr>
                                <floatingAddress>
                                        <healthCheckConditionPolicy>
                                                <healthCheck
```

enable =: default= "false"
Enable / disable the loading and the assessment of the section.

address =: default value=""
The address on which to perform the health check.

port =: default value= "0"
The port on which the health check service responds. If <=0 an ICMP check is run.

SSL =: default value= "false"
If true, it executes the health check service through an SSL connection (HTTPS).

uriPath =: default value=""
The URIPath on which the health check service responds. If the value is not present the health check will be run using a TCP connection.

numRetryConnection =: default value= "default from <params>"
Number of connection attempts before declaring the service inactive. If a value is not provided the system takes the default from the <params> section.

waitPerRetryConnection =: default value= " 300" UM=Milliseconds
Time to wait between connection attempts until the numRetryConnection is reached. If a value is not provided the system takes the default from the <params> section.

createConnectionTimeOut =: default value= "default from <params>"
Time to wait before declaring time-out of the connection attempt. If a value is not provided the system takes the default from the <params> section.

14 OPLON®AAI WorkFlow surfaceclusterwf

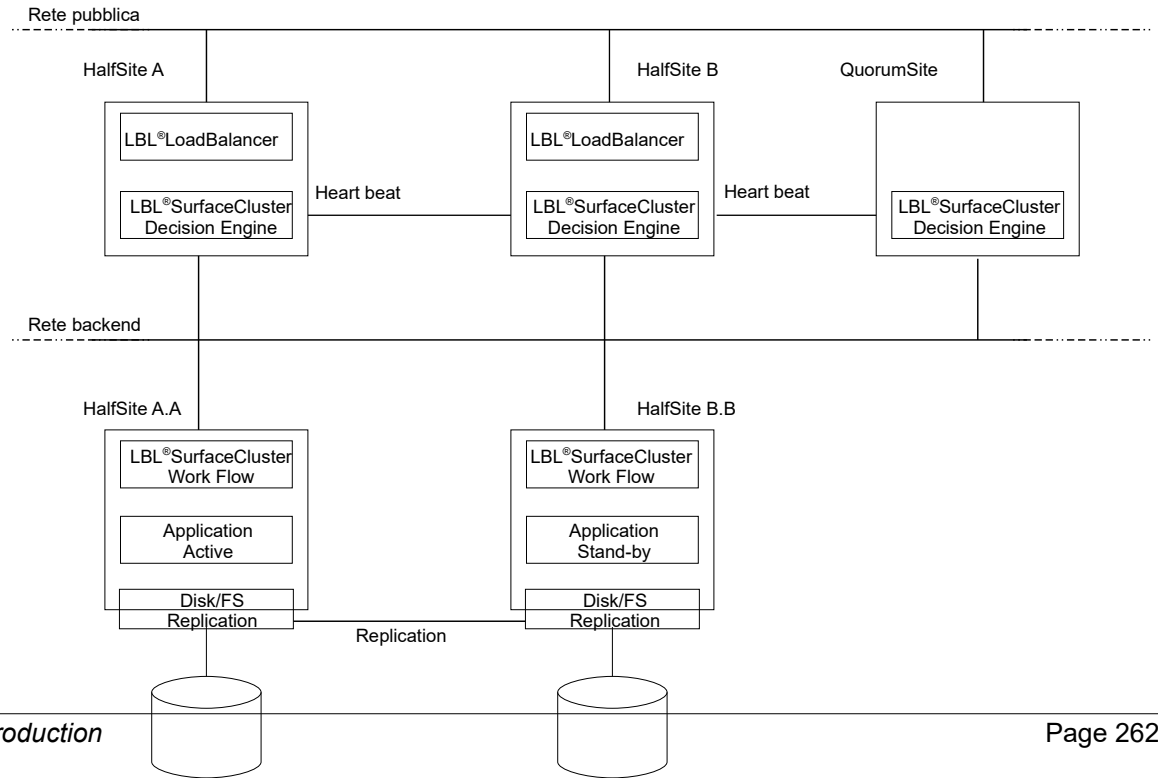
Introduction

OPLON®Commander introduces a new concept in high reliability of applications/services by taking the role of coordinator of the activities of a mission-critical data center.
OPLON®Commander consists of two main modules:

- OPLON®Commander Work Flow
- OPLON®Commander Decision Engine.

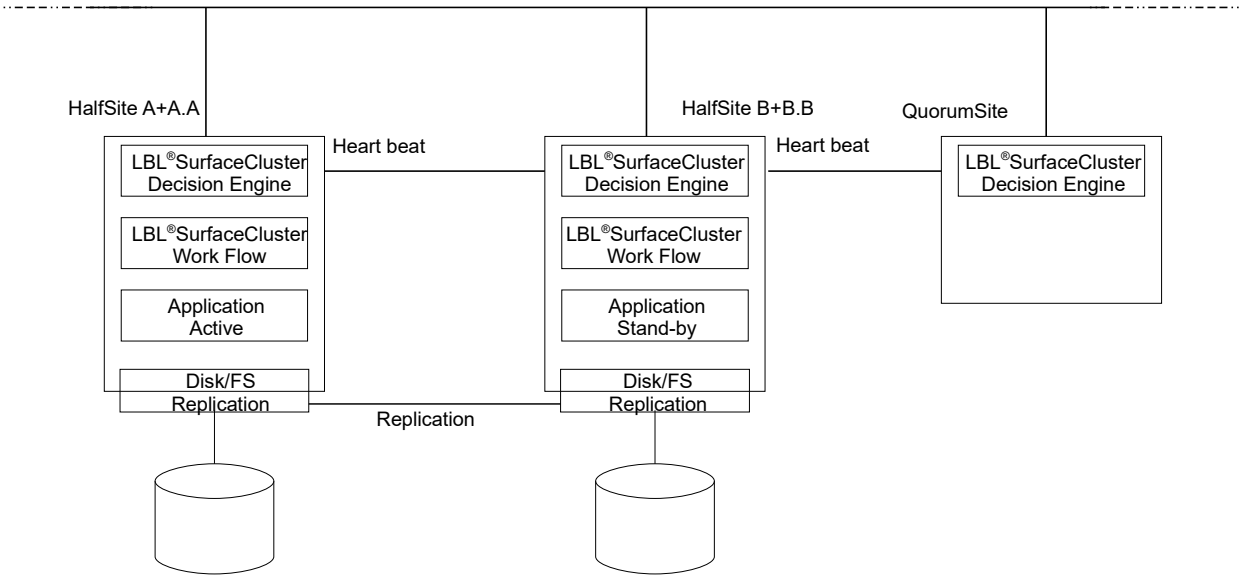
The two modules have been designed to work together, but in cases where automation of operations is not required, the OPLON®Commander Work Flow component can be used alone.

The general architecture can be summarized in the following scheme:

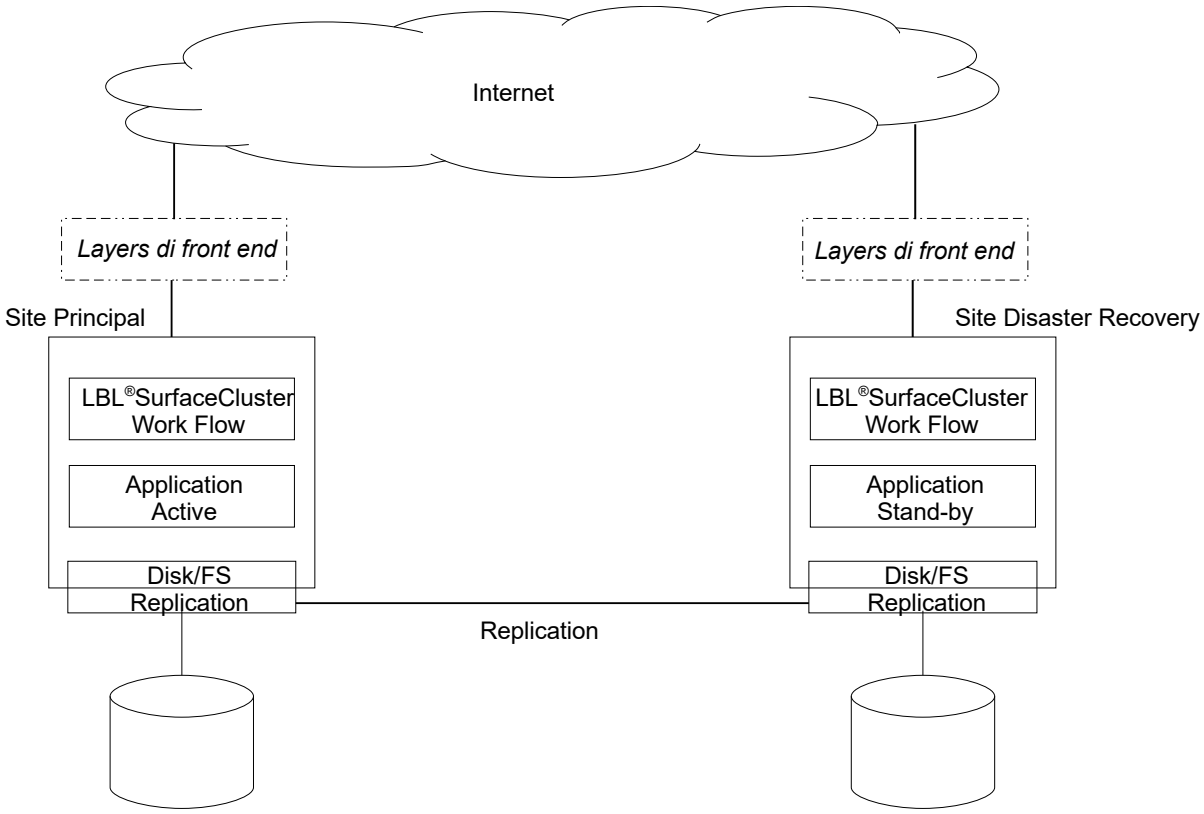


The flexibility of the tool can be seen in different architectures.

Here is another example of possible architecture:



The following is a Disaster Recovery scenario:



Individually, the work flow service can be thought of as a procedures automator. In fact, the service provides for the ability to describe the logic of the operations inside the XML file. These operations will then be performed in a timely manner, and the assessment of the return code allows verification of proper operation and programming of the next "steps" to perform.

The Work Flow service parameters file therefore describes the actions to perform and for each event the subsequent corresponding actions to take.

OPLON®Commander Work Flow introduces a new paradigm being a service of Remote Workflow Command (RWC), and introduces structured programming within this systems environment.

The structure of the configuration file is as follows:

```
<serviceconf>
   <copyright>
   </copyright>
        <surfaceclusterwf>
                    <params>
                    </params>

                    <workflow>
                       <step>
                          <returncode>
                          </returncode>
                       </step>
                        ...
                    </workflow>
                     ...
        </surfaceclusterwf>
</serviceconf>
```

<surfaceclusterwf>

```
<serviceconf>
        <surfaceclusterwf>
```

<params>

```
<serviceconf>
        <surfaceclusterwf>
                <params
```

The section contains the general configuration parameters of the workflow service.

address =: default value= "localhost"
Address on which the service responds. Usually the service is on the backend network.

port =: default value= " 54444"

The port on which the service responds.

backlog =: default value= "20"
The maximum number of incoming connections at the socket-level where the operating system allows this setting.
The TCOProject® libraries have their own connection requests management system in case the operating system does not allow taking advantage of this feature.

reuseAddress =: default value= "true"
The corresponding socket parameter SO_REUSEADDR.

concurrentWorkers =: default value= " 20"
The initial number of requests simultaneously handled from the WebConsole.

maxConcurrentWorkers =: default value= " 100"
The maximum number of requests simultaneously handled from the WebConsole.

webAppsDir=: default value= "lib/webroot_surfaceclusterwf/webapps"
Home directory of web applications.

webAppsConfDir =: value of default="lib/webroot/webappsconf"
Configuration directory of web applications.

webSecurityDir =: value of default=" lib/webroot/websecurity "
Configuration directory for security of web applications

certificateURL =: value of default=" /certificate/serverkeys "
If set indicates the http address from which to get the certificate.

keyStore=: default value= "JKS"
Indicates the type of SSL keystore from which to get the certificate. Normally if the JVM kestore is used it must be set to "JKS" if an OpenSSL keystore is used, it must be set "PKCS12 ".

keyStorePassword =: default value= "defaultpwd"
Password to access the keystore.

alias =: default value= "lblcert"
The identifier of the certificate in the keystore.

aliasPassword =: default value= "defaultpwd"
The password to gain access to the certificate contained in the keystore.

keyManagerFactory =: default value= "SunX509"
This indicates the method of interpretation of the certificate. Normally set to "SunX509".

SSLContextVersion =: default value= "SSLv3"
Indicates the version of the SSL. Normally set to "SSLv3" for the JVM keystore or "TLS"

for OpenSSL

healthCheckUriPath =: default value= " /HealthCheck"
It is the path for the activities healthcheck. This value is usually never changed unless it is already being used in other applications.

workFlowCommandUriPath =: default value= " /SCWFCommand"
The path on which the Remote Work Flow Command (RWC henceforth) management web service will respond .

dateFormat=:default=”dd/MM/yyyy HH:mm:ss:SSSS”
The format of the date of the statistical data from web service.

delimiter =: default= " |" (pipe)
The character that delimits the fields during the exchange of information.

maxRecords =: default value= " 500" UM=Record
Maximum number of records in response to a Web Service request.

sysCommandTimeOut =: default value= " 10000" UM=Milliseconds
Indicates the time required to declare time-out on a system command. If the command exceeds this limit an abort command is executed, and subsequently the control is released to the application.

sysCommandCheckRate =: default value= " 300" UM=Milliseconds
The frequency by which to check on the state of the system command.

sysCommandRemoteURL =: default= "https://localhost:5992/sysCommand"
The URL of the service to run system commands

exclusiveRunWorkFlow =: default value= "false" UM=boolean
This value indicates whether one Work Flow at a time should be performed and only at the end (@STOP_RUN) of the workflow can a new Work Flow be started.

surfaceClusterWFCommandDir =: default value= "surfaceClusterWFCommandDir"
The base directory where searches for scripts/programs will be performed unless otherwise specified, with an absolute path, on each "step" of work. If an absolute address is not indicated, everything will be referred to (LBL_HOME)/procsProfile/XXX_procName/.

<workflow>

```
<serviceconf>
        <surfaceclusterwf>
                <workflow
```

enable =: default= "true"
If true this section is active. If false the section is not taken into consideration during the initial loading.

name =: default value=""
The name of the work flow. The name must be unique to this XML file and identifies a set of "steps".

description =: default value= "workflow: (name)"
The description of the work flow. Enter a short but significant description as it will appear in the user interface views.

startName =: default value= "first <step> in sequence"
The first step to be performed. If not specified it takes the first <step> in sequence.

startType =: default value= "manual"
Indicates if the workflow must be started at the start of the service.
If "manual" the service must be started: manually by the operator, or by an OPLON®Commander Decision Engine.
If "automatic" the service will start automatically when the service started.

requiresHumanDecision =: default value= "false"
Indicates if the work flow once started will automatically evolve or will require human intervention to start each step one at a time. Can also be used for not proliferating the number of work flows with a single <step> and maintain a library of components.

<step>

```
<serviceconf>
        <surfaceclusterwf>
                <workflow>
                        <step
```

enable =: default= "true"
If true this section is active. If false the section is not taken into consideration during the initial loading.

name =: default=""
Name of the step. This parameter is the unique reference in this section <workflow>.

description =: default= "workflow: (<workflow name>) - (name)"
Description of the step. Enter a brief but comprehensive description of the action relating to this step.

commitWorkFlow =: default= "false" UM=boolean
Indicates if this step is to be considered conclusive.
Can be used to indicate that the step is in loop on itself because it wants to engage that particular workflow not giving the possibility to be used again until the next reboot of the system. This flag if set to true, and in the presence of a loop on itself with a positive result, stops logging in order to avoid clogging and making it easier to read.

waitBeforeExecute =: default= "0" UM=Milliseconds
Time to wait before executing the command.

sysCommandTimeOut =: default= "1000" UM=Milliseconds
Time to wait before declaring that the command unsuccessful.
This value is very important because a command may even last for hours or days, consider for example a restart of a database with many terabytes in line. It is a value that must always be well thought out. Of course, if the command terminates prior to this time, then the processing would proceed.

evaluateReturnCode =: default= "true" UM=Boolean
Indicates if the return code must be assessed .
If "false" any result of the command will be assessed as positive. If "true" the return codes of operations will be assessed. In the absence of <returncode> sections, the returncode 0 (zero) will be evaluated as a positive result otherwise any other value will be interpreted as negative.
Also in the presence of only one <returncode> section all the return codes will be interpreted as negative response if not otherwise indicated. Hence if return codes 0 (zero) were not specified these would be interpreted as negative.

command =: default=""
The command which will be performed in this step.
Can be any executable (batch including .sh/bash etc. or .bat in Microsoft Windows). If the command is not indicated with an absolute path it is added as specified in the parameter "surfaceClusterWFCommandDir" from section <params> .
It is possible to define in this parameter a command inside of RWC. To carry out a RWC enter:

- @RWC hostName=localhost workFlow=takeControl

This reserved keyword allows propagation of a initial RWC on more systems thereby carrying out articulated operations on more infrastructure layers.

The complete command with all of the parameters is as follows:
@RWC Remote Workflow command
hostname=mandatory
portNumber=optional, default 54444
uriPath=foptional, default /SCWFCommand
Workflow=mandatory
step=optional, default ""
command=optional, default ""
frhd=optional, default false

The command parameter can take the following values that can be displayed even through browser connection.
Ex.:

URL: [https://jessicabackend:54444/SCWFCommand]
/SCWFCommand?command=getWorkFlows
/SCWFCommand?command=getViewLog&workFlow=workFlowName
/SCWFCommand?command=getStepsList&workFlow=workFlowName
/SCWFCommand?command=getStep&workFlow=workFlowName&step=stepName
/SCWFCommand?command=runWorkFlow&workFlow=workFlowName[&frhd=(false|true)]
/SCWFCommand?command=runStep&workFlow=workFlowName&step=stepName[&frhd=(false|true)]
/SCWFCommand?command=stopWorkFlow&workFlow=workFlowName

maxRetry=: default= "0" UM= whole number
In the face of a negative return code or interpreted as negative, the step will be repeated for a number of times equal to this parameter. As will be seen later in section <returncode> it is possible to change this value for an exception, in the face of a particular return code or cancel it completely.

gotoWhenTrue =: default= "next step"
If the command is successful and if not otherwise indicated in section <returncode>, this value indicates the step the work flow will seek to perform as the next step.
If omitted the next step will be automatically taken.

gotoWhenFalse=: default= "next step"
If the command gave a negative result and unless otherwise stated in section <returncode> this value indicates the step the work flow will aim to perform as the next step.
If omitted the next step will be automatically taken.

<returncode>

```
<serviceconf>
        <surfaceclusterwf>
                <workflow>
                        <step>
                                <returncode
```

enable =: default= "true"
If true this section is active. If false the section is not taken into consideration during the initial loading.

value =: default= " 0" UM=whole number
Value of the return code that will be considered for this step. This value can not be duplicated within the same step.

description =: default= "returncode: (<step> name)-(value)"
Description of the return code. This description should be brief but comprehensive to provide a valid trace during the display of steps and for the correct timely human interpretation.

result =: default= "true" UM=Boolean
If true the return value will be interpreted as positive and will trigger the steps of positive return if not otherwise indicated in this section.

gotoStep =: default= "next step"
Indicates the step the work flow will target for this return code. If not indicated, and the result is considered positive it will point to the next step in sequence. In either case, the step will be repeated for maxRetry in section <step> unless otherwise indicated in this same section using the 'retry' parameter.

retry =: default= "true" UM=Boolean
If true the operation will be repeated for maxRetry as defined in <step>section - in the case the parameter result of this return code was set as a result negative "false". If false in any

case no further attempts will be made.

waitBeforeRetry =: default= " 0" UM=Milliseconds
The time to wait before retrying the operation if provided by parameter 'retry'

Example: start Tomcat

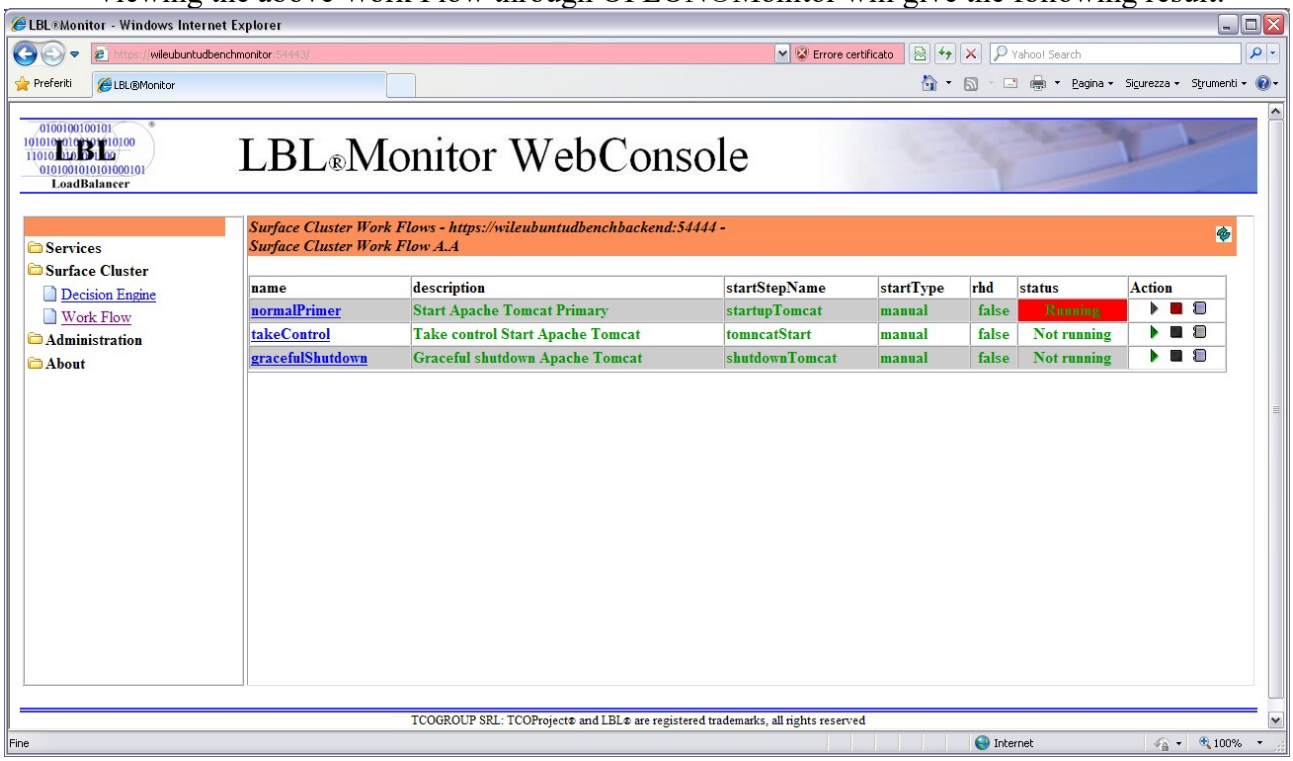
Below is an example of configuring a minimal Work Flow where the start of a Tomcat instance is performed.

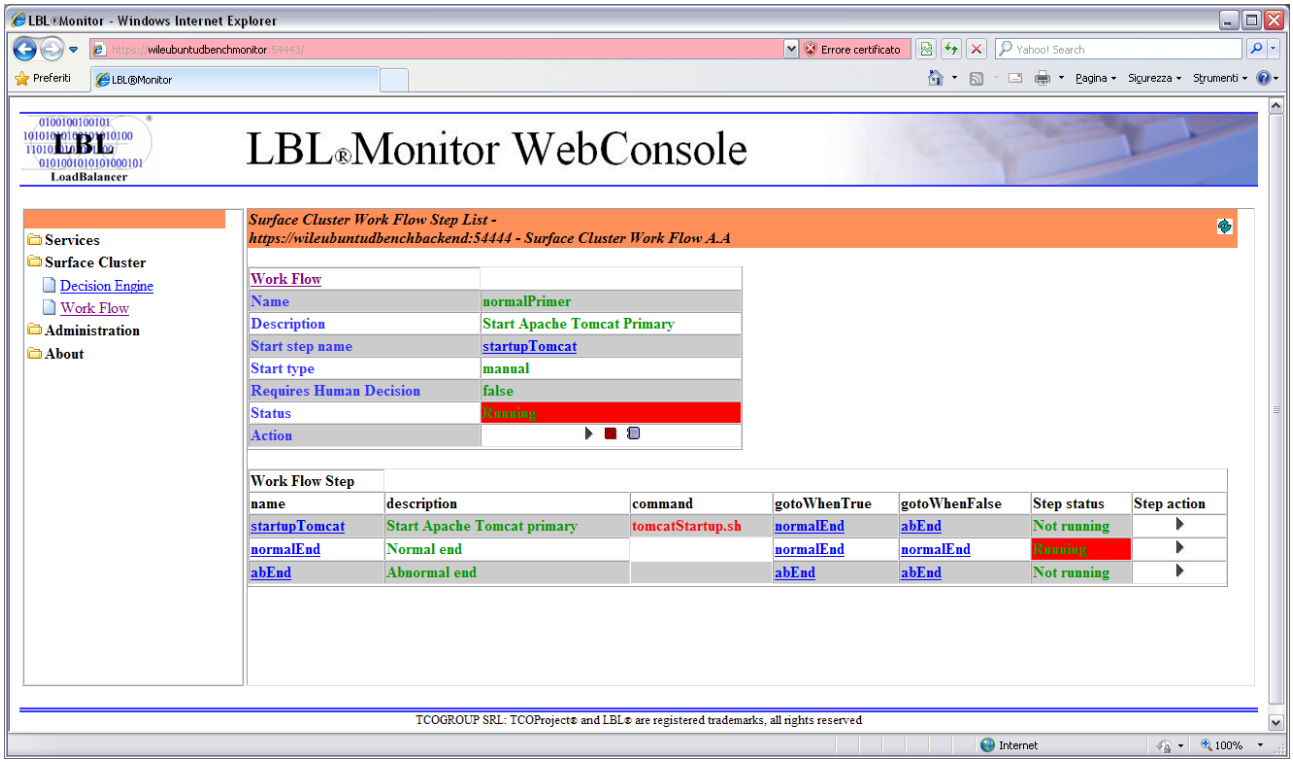
...

```
  LBL(tm) LoadBalancer is built on TCOProject(tm) SoftwareLibrary
LBL and TCOProject are trademarks of F.Pieretti. All rights reserved.
   </copyright>
         <surfaceclusterwf>
            <params
               frequency="10000"

               address="wileubuntudbenchbackend"
               port="54444"

              sysCommandRemoteURL="https://localhost:5992/SysCommand"
              exclusiveRunWorkFlow="false">
           </params>

   <!-- ********************************************************************** **
        ********** START WORK FLOW NORMAL STARTUP ******************
     ** ********************************************************************** -->
      <workflow name="normalPrimer"
            description="Start Apache Tomcat Primary"
            startName="startupTomcat">

         <step enable="true"
            name="startupTomcat"
            description="Start Apache Tomcat primary"
            waitBeforeExecute="1000"
            sysCommandTimeOut="600000"
            evaluateReturnCode="true"
            command="tomcatStartup.sh"
            maxRetry="3"
            gotoWhenFalse="abEnd">
         </step>

         <!-- ********************************************************** **
            WORK FLOW  NORMAL STARTUP END FUNCTIONS
           ** ********************************************************** -->
         <step enable="true"
            commitWorkFlow="true"
            name="normalEnd"
            description="Normal end"
            waitBeforeExecute="10000"
            evaluateReturnCode="false"
            gotoWhenTrue="normalEnd"
            gotoWhenFalse="normalEnd">
         </step>

         <step enable="true"
            name="abEnd"
```

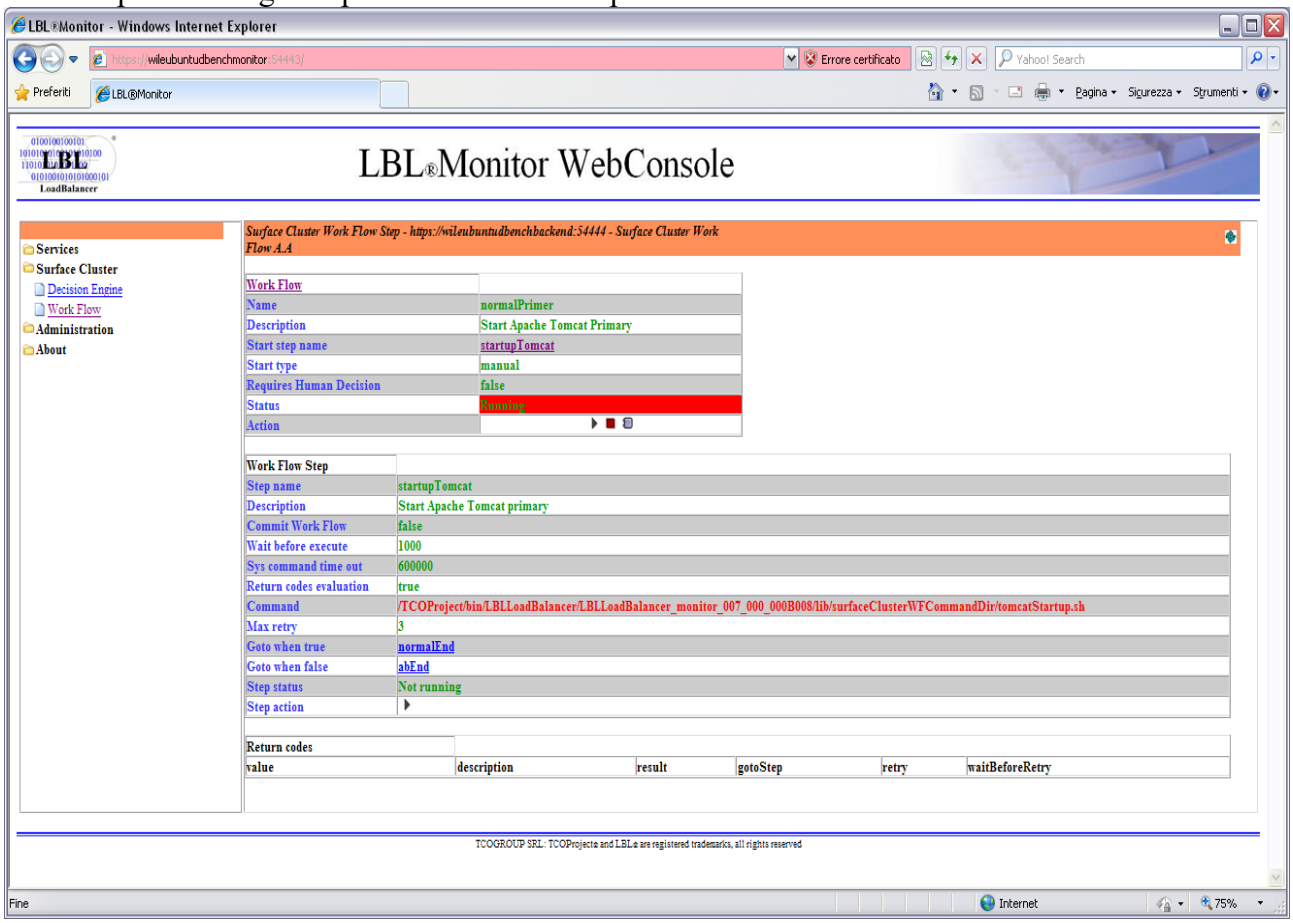
```
            description="Abnormal end"
            evaluateReturnCode="true"
            waitBeforeExecute="10000"
            gotoWhenTrue="abEnd"
            gotoWhenFalse="abEnd">
          <returncode value="0"
                  description="KO FOR ABNORMAL END"
                  result="false"/>
      </step>
    </workflow>

<!-- ********************************************************************** **
      ********** START WORK FLOW GRACEFUL SHUTDOWN ******
 ** ********************************************************************** -->
    <workflow name="gracefulShutdown"
            description="Graceful shutdown Apache Tomcat"
        ...
              ...
```

Viewing the above Work Flow through OPLON®Monitor will give the following result:





Up to the single step view with detailed parameters:



15 OPLON®WorkFlow Remote Batch

This service enables secure remote execution of batch or executables.

```
<serviceconf>
   <copyright>
   </copyright>
        <remotebatch>
                <params>
                </params>
        </remotebatch>
</serviceconf>
```

<params>

```
<serviceconf>
        <remotebatch>
                <params>
```

address =: default value= "localhost"
The value, set by default to localhost, should be set with the monitoring network. In the absence of a monitoring network it is recommended to set this to the backend network.

port =: default value= " 5994"
The port on which the service responds.

backlog =: default value= "20"
The maximum number of incoming connections at the socket-level where the operating system allows this setting.
OPLON®ADC its own connection requests management system in case the operating system does not allow taking advantage of this feature.

concurrentWorkers=: default value= " 20"
The initial number of requests handled simultaneously from the WebConsole.

maxConcurrentWorkers =: default value= " 100"
The maximum number of requests handled simultaneously from the WebConsole.

healthCheckContextPath =: default value= " /HealthCheck"
The path of the activities healthcheck. This value is usually never changed unless already used in other applications.

webAppsDir =: default value= "lib/webroot_remotebatch/webapps"
Home directory of web applications.

webAppsConfDir =: default value =”lib/webroot_remotebatch/webapps”
Configuration directory of web applications.

webSecurityDir =: default value =”lib/webroot/websecurity”
Configuration directory for security of web applications.

Certificateurl =: default value= "certified/serverkeys"
If set indicates the http address from which to get the certificate.

keyStore=: default value= "JKS"
Indicates the type of SSL keystore from which to get the certificate. Normally if the JVM kestore is used it must be set to "JKS" if an OpenSSL keystore is used, it must be set "PKCS12 ".

keyStorePassword =: default value= "defaultpwd"
Password to access the keystore.

alias =: default value= "lblcert"
The identifier of the certificate in the keystore.

aliasPassword =: default value= "defaultpwd"
The password to gain access to the certificate contained in the keystore.

keyManagerFactory =: default value= "SunX509"
This indicates the method of interpretation of the certificate. Normally set to "SunX509".

SSLContextVersion =: default value= "SSLv3"
Indicates the version of the SSL. Normally set to "SSLv3" for the JVM keystore or "TLS" for OpenSSL.

Profile File for launch executables and batch

OPLON®Cluster Remote Batch interprets the profile file contained in the directory

- (LBL_HOME)/lib/webroot_remotebatch/webapps/RemoteBatch

The launch profile file must have the following format:

```
<startBatch>
        <params remoteBatchTimeOut="10000"
                remoteBatchCheckRate="300"
```

```
                    allowURIParams="false" debug="false"/>
        <legacyCommand>T:\work0\tmp\vai.bat 11 </legacyCommand>
</startBatch>
```

Since it is interpreted at the time of the recall, exactly as an HTML page, the values can be changed dynamically on the fly.

The name of the file must be the same as the main section.
In the case described above, the file must be named 'startbatch' and have an extension such as: xml; txt, etc. (mime type text/html).

The parameters are few and fairly intuitive.
The following is a detailed description.

Profile File Parameters

remoteBatchTimeOut =: default value " 10000"
The time to wait before releasing the batch process/executable.

remoteBatchCheckRate =: default value " 300"
Time to wait between attempts after having waited for remoteBatchTimeOut. After 3 attempts the launch of the batch is declared failed.

allowURIParams =: default value "false"
If true allows additional parameters from the URI.
The default is false for security reasons. If not set or set to false any parameters set in URIPath will be ignored.

debug =: default value "false"
If true executes a warning log with the values of the start process and the result obtained.

<legacyCommand> =: default value "null"
The batch to launch. For security reasons, this value may not be blank.

The launch of the batch will occur through a request for URL that reports the name of the file descriptor to interpret for the launch of the batch/executable.
It is possible to specify additional parameters during the launch of the program in the URL itself as in the example below:

- http://localhost:5994/RemoteBatch/startBatch.txt?parameters= 11 2222 3333 4444

In this case the launch of the batch will be composed as follows in the parameters part:
T:\work0\tmp\vai.bat 11 2222 3333 4444

ATTENTION : The contents of the batch file and the various compositions of the parameters are responsibility of the batch itself. The LBL service does not verify the correctness of the batch or various compositions of the parameters that are completely at discretion of the implementer.

16

OPLON®DecisionEngine surfaceclusterde.xml

The module OPLON®Commander Decision Engine is the engine of the Surface Cluster solution. Its purpose is to constantly check the status of applications and in the face of criticality trigger processes (Work Flow) to resolve the problem.

The main characteristic of the OPLON®Commander solution is the decoupling of automatic decision and resolution of the problem. This characteristic makes it ultra-flexible and allows the persons responsible to keep the situation under control and eventually intervene security even manually by acting directly from the Work Flow module.

OPLON®Commander Decision Engine through Health Checks crossing over the public network, back end network of backend, heartbeat network and application vitality, is able to obtain a comprehensive picture of the situation that allows it to take decisions.

The parameters described herein provide fundamental elements in order to discriminate a critical situation from a false problem. The organization and set up of the parameters have been deliberately designed to be both, simple but effective needing only to indicate the essential data for the specific purpose.

The outline below distinguishes 4 areas of operations enclosed in section <decisionEngine>

- < decisionEnginesPeers>
- < healthCheckServicesPolicy>
- < healthCheckPublicPolicy>
- < healthCheckBackendPolicy>

These four sections represent the high level view that allows the Decision Engine to draw conclusions and implement interventions as needed to restore applications operations.

```
<serviceconf>
   <copyright>
   </copyright>
         <surfaceclusterde>
                  <params>
                  </params>
                  <decisionEngineMgr>
                     <decisionEngine>
```

```
                        <decisionEnginesPeers>
                          <peer/>
                          <peer/>
                          ...
                        </decisionEnginesPeers>

                        <healthCheckServicesPolicy>
                            <failOverService>
                                <healthCheck/>
                                <healthCheck/>
                                ...
                            </failOverService>
                            ...
                        </healthCheckServicesPolicy>

                        <healthCheckPublicPolicy>
                                <healthCheck/>
                                <healthCheck/>
                                ...
                        </healthCheckPublicPolicy>

                        <healthCheckBackendPolicy>
                                <healthCheck/>
                                <healthCheck/>
                                ...
                        </healthCheckBackendPolicy>
                    </decisionEngine>
                    ...
                </decisionEngineMgr>
        </surfaceclusterde>
</serviceconf>
```

The remainder of the document details each section and each parameter within the section. Many values are proposed by default making it simple to compose while at the same time allowing flexibility to customize with specific implementational details.

<decisionEnginesPeers> Encloses the parameters that are needed to give the single OPLON®Commander Decision Engine instance knowledge of the existence of the other decision engine instances which contribute to the decision making process. Remember that the minimum to ensure a reliable decision is 3 (three) instances of OPLON®Commander Decision Engine.

<healthCheckServicesPolicy> Contains the profiles of the services to observe and to activate in the event of necessity. The sequence of application profiles also determines the priority of the service with respect to the corresponding services in stand-by.

<healthCheckPublicPolicy> Contains the health check policies for declaring the operation

of the public network. A good sample for the health check may not be less than 3 (three) health checks.

<healthCheckBackendPolicy> Contains the health check policies for declaring the operation of the back end network. A good sample for the health check may not be less than 3 (three) health checks. It is useful to also enter into the sample isntances of OPLON®Commander Work Flow so as to also constantly check the reachability and operation of the service that allows taking corrective actions.

An example of a configuration file for an OPLON®Commander Decision Engine instance can be synthesized into a single page where the items to complete are very few given most are pre-populated or default.

```
LBL and TCOProject are trademarks of F.Pieretti. All rights reserved.
    </copyright>
    <surfaceclusterde>
        <params
            frequency="10000"
            address="solu6bench001monitor"
            port="5445"
            addressHeartBeat="solu6bench001private"
            portHeartBeat="5445"
            sysCommandRemoteURL="https://localhost:5992/SysCommand">
        </params>

        <decisionEngineMgr>
            <decisionEngine enable="true"
                            groupName="SCDEGroup"
                            description="SCDE halfSite A"
                            frequency="10000"
                            firstThinkingTime="45000">
                <!-- ********************************************************
                     PEERS DECISION ENGINES NODES (Min 3 nodes)
                   ** ******************************************************** -->
                <decisionEnginesPeers>
                    <peer enable="true"
                          description="HalfSite B"
                          URL="https://solu6bench002private:5445/"/>
                    <peer enable="true"
                          description="QuorumSite"
                          URL="https://wilelbloneprivate:5445/"/>
                </decisionEnginesPeers>
                <!-- ********************************************************
                     APPLICATIONS SERVICES AND ASSOCIATED SURFACE WORK FLOW SERVER
                   ** ******************************************************** -->
                <healthCheckServicesPolicy description="Services switch policy"
                                           waitTimeAfterNormalPrimer="180000"
                                           waitTimeAfterFlagBroken="60000"
                                           waitTimeBeforeTakeControl="180000"
                                           waitTimeAfterTakeControl="900000"
                                           applicationLostTime="30000">
                    <failOverService enable="true"
                                     description="HalfSite A.A"
                                     surfaceClusterWorkFlowURL="https://wileubuntudbenchbackend:54444/">
                        <healthCheck enable="true"
                                     description="primary application HalfSite A.A"
                                     address="wileubuntudbenchbackend"
                                     port="8080" uriPath="/" SSL="false"/>
                    </failOverService>
                    <failOverService enable="true"
                                     description="HalfSite B.B"
4                                     surfaceClusterWorkFlowURL="https://roadubuntudbenchbackend:54444/">
                        <healthCheck enable="true"
                                     description="Secondary application HalfSite B.B"
                                     address="roadubuntudbenchbackend"
                                     port="8080" uriPath="/" SSL="false"/>
                    </failOverService>
                </healthCheckServicesPolicy>
                <!--
                     PUBLIC HEALTH CHECK
                   ** ******************************************************** -->
                <healthCheckPublicPolicy>
                    <healthCheck address="192.168.43.104" port="22" description="wilecoyote"/>
                    <healthCheck address="192.168.43.103" port="22" description="roadrunner"/>
                    <healthCheck address="192.168.43.101" description="gundam"/>
                </healthCheckPublicPolicy>
                <!--
                     BACKEND HEALTH CHECK
                   ** ******************************************************** -->
                <healthCheckBackendPolicy>
                   <healthCheck address="192.168.45.104" port="22" description="wilecoyote"/>
                   <healthCheck address="192.168.45.103" port="22" description="roadrunner"/>
                   <healthCheck address="192.168.45.101" description="gundam"/>
                   <healthCheck enable="true" address="wileubuntudbenchbackend" port="54444"
                                uriPath="/HealthCheck" SSL="true"
                                description="Surface Cluster Work Flow A.A"/>
                   <healthCheck enable="true" address="roadubuntudbenchbackend" port="54444"
                                uriPath="/HealthCheck" SSL="true"
                                description="Surface Cluster Work Flow B.B"/>
                </healthCheckBackendPolicy>
            </decisionEngine>
        </decisionEngineMgr>
        <sysobserver>
            <service name="syslog" id="syslogsurfaceclusterde"/>
        </sysobserver>
    </surfaceclusterde>
</serviceconf>
```

<surfaceclusterde>

```
<serviceconf>
    <surfaceclusterde>
        <params
```

Paragrafo contenitore di tutte le configurazioni

<params>

```
<serviceconf>
    <surfaceclusterde>
        <params
```

General Parameters of the service

frequency =: default value= " 10000" UM=Milliseconds
The frequency of verification of status changes.

address =: default value= "localhost"
Set by default to localhost. The service normally will be assigned on the monitoring network.

port =: default value= " 54445"
The port on which the service responds.

addressHeartBeat =: default value= "address"
The address of the heartbeat. The Decision Engine can be reached through two listeners. This listener will respond to the address assigned on the HeartBeat.

portHeartBeat =: default value= "port+1 (ex. 54446)"
The port on which the service responds attested on the heartbeat network.

backlog =: default value= "20"
The maximum number of incoming connections at the socket-level where the operating system allows this setting.
The TCOProject® libraries have their own connection requests management system in case the operating system does not allow taking advantage of this feature.

reuseAddress =: default value= "true"
The corresponding socket parameter SO_REUSEADDR.

concurrentWorkers =: default value= " 20"
The initial number of requests simultaneously handled from the WebConsole.

maxConcurrentWorkers =: default value= " 100"
The maximum number of requests simultaneously handled from the WebConsole.

webAppsDir=: default value= "lib/webroot_surfaceclusterde/webapps"
Home directory of web applications.

webAppsConfDir =: value of default="lib/webroot/webappsconf"

Configuration directory of web applications.

webSecurityDir =: value of default=" lib/webroot/websecurity "
Configuration directory for security of web applications

certificateURL =: value of default=" /certificate/serverkeys "
If set indicates the http address from which to get the certificate.

keyStore=: default value= "JKS"
Indicates the type of SSL keystore from which to get the certificate. Normally if the JVM kestore is used it must be set to "JKS" if an OpenSSL keystore is used, it must be set "PKCS12 ".

keyStorePassword =: default value= "defaultpwd"
Password to access the keystore.

alias =: default value= "lblcert"
The identifier of the certificate in the keystore.

aliasPassword =: default value= "defaultpwd"
The password to gain access to the certificate contained in the keystore.

keyManagerFactory =: default value= "SunX509"
This indicates the method of interpretation of the certificate. Normally set to "SunX509".

SSLContextVersion =: default value= "SSLv3"
Indicates the version of the SSL. Normally set to "SSLv3" for the JVM keystore or "TLS" for OpenSSL

healthCheckUriPath =: default value= " /HealthCheck"
It is the path for the activities healthcheck. This value is usually never changed unless it is already being used in other applications.

decisionEngineCommandUriPath =: default value= "/SCDECommand"
The path in which the the management web service will respond.

sysCommandTimeOut =: default value= " 10000" UM=Milliseconds
Indicates the time required to declare time-out on a system command. If the command exceeds this limit an abort command is executed, and subsequently the control is released to the application.

sysCommandCheckRate =: default value= " 300" UM=Milliseconds
The frequency by which to check on the state of the system command.

sysCommandRemoteURL =: default= "https://localhost:5992/sysCommand"
The URL of the service to run system commands

dateFormat=:default=”dd/MM/yyyy HH:mm:ss:SSSS”

The format of the date of the statistical data from web service.

delimiter =: default= " |" (pipe)
The character that delimits the fields during the exchange of information.

maxRecords =: default value= " 500" UM=Record
Maximum number of records in response to a Web Service request.

maxSizeInBuffer=: default value= " 10485760" UM=Bytes
Maximum Number of bytes in response to a Web Service request.

statusDir =: default value= "lib/surfaceClusterDEStatus"
The directory for the persistence of the state of services (do not to alter manually, is managed automatically).

notificationDir =: default value= "lib/notificationDir"
The directory for the persisitence of the states which are verified by OPLON®ADC to force an Out Of Order for a service that is being balanced.
OPLON®Commander Decision Engine also notifies in this directory in the format interpreted by OPLON®ADC as an Out of Order service.

numRetryConnection =: default value= " 3"
The default number of attempts before declaring a service as no longer active.

createConnectionTimeOut =: default value= " 5000" UM=Milliseconds
Time-out of the connection attempt.

<decisionEngineMgr>

```
<serviceconf>
    <surfaceclusterde>
        <decisionEngineMgr
```

This section contains a list of all the Decision Engine engines. It is possible to associate multiple independent decision engines to the same instance.

<decisionEngine>

```
<serviceconf>
    <surfaceclusterde>
        <decisionEngineMgr>
            <decisionEngine
```

This section describes the parameters related to a single decision engine. One instance of OPLON®Commander Decision Engine can have multiple independent decision engines.

enable =: default= "true" UM=boolean
Enables or disables the interpretation of this section in the instance.

groupname =: default value= "SCDEGroup"
The name of the Decision Engine group. It is very important because the decision engines

contained in an instance can be distinguished from this name.

description =: default value= "description: groupName"
The description of this decision engine. Must be concise, yet comprehensive.

frequency =: default value= " <params frequency>" UM=Milliseconds
The frequency of verification of status changes. If not specified the frequency of the <params> section is used.

firstThinkingTime =: default value= " 45000" UM=Milliseconds
The waiting time of initialization and first tests. Once this time has passed, the decision-making engine will producet a message. The decision-making engine does not proceed, however, even after this initial time has passed, and waits for the total initialization of the states.

applicationLostTimeBeforeRestart =: default= "1/2 of applicationLostTime"
UM=Milliseconds
Time to wait before declaring the application lost and try a restart, if provided.
Even if the switch quorum is reached this time is still waited for before the restart of the resource is definitely in failure. If after this period the resource is back UP no further action is taken. The event is recorded in the log files and eventually reported via e-mail or HTTP post.
Default is 1/2 of applicationLostTime (15 seconds). If set to > (greater than) applicationLostTime nothing is executed.

operator =: default= "OR"
Boolean Operator applied to the services in which a healthcheck is being executed . If "OR" even if only one service is an a state of failure the switch procedures are activitated, if "AND" all services must be in a state of failure to activate the restart procedures.

applicationLostTime =: default value= " 30000" UM=Milliseconds
The waiting time from when the application being verified was declared in a state of failure (down). This is a very important value because once this period of time has passed and the application should still be unreachable, and the switch quorum verified, the process of recovery is triggered. The 30" are the minimum to avoid false positives.

numRetryConnection =: default value= " <params numRetryConnection>"
The default number of attempts before declaring a service as no longer active.

createConnectionTimeOut =: default value= " <params createConnectionTimeOut>"
UM=Milliseconds
Time-out of the connection attempt.

<decisionEnginesPeers>

```
<serviceconf>
    <surfaceclusterde>
        <decisionEngineMgr>
            <decisionEngine>
                <decisionEnginesPeers
```

This section contains information relating to the other OPLON®Commander Decision Engine peer instances. The minimum number of decision engines is 3 in order to be able to declare a quorum for switches.

<peer>

```
<serviceconf>
    <surfaceclusterde>
        <decisionEngineMgr>
            <decisionEngine>
                <decisionEnginesPeers>
                    <peer
```

This section contains the information of a peered (joint) Decision Engine service.

enable =: default= "true" UM=boolean
Enables or disables the interpretation of this section in the instance.

URL =: default="" UM=URL W3c
The connection URL to the peered service through the heartbeat network.

description =: default= "description peer: URL"
Brief but comprehensive description of the service.

healthCheckUriPath =: default= " <params healthCheckUriPath>"
URIpath that associated with the URL determines the Health Check service of the peered instance.

decisionEngineCommandUriPath=:default= "<paramsdecisionEngineCommandUriPath>"
URIpath that associated with the URL determines the Health Check service of the peered instance.

numRetryConnection =: default value= " <decisionEngine numRetryConnection>"
The default number of attempts before declaring a service as no longer active.

createConnectionTimeOut =: default value= "< decisionEngine createConnectionTimeOut>" UM=Milliseconds
Time-out of the connection attempt.

addressProxy =: default value=""
The address of the proxy on which the health check is carried out.

WARNING: Do not set this value with a name but set it with an address (e.g. : 192.168.43.142).

portProxy =: default value= " 0"
The proxy port of the health check.

<healthCheckServicesPolicy>

```
<serviceconf>
    <surfaceclusterde>
        <decisionEngineMgr>
            <decisionEngine>
                <healthCheckServicesPolicy
```

This section contains the general information of an application service to be able to verify its functionality and then undertake recovery actions. The succession of activation in the face of an failure event occurs on the basis of the insertion sequence in the xml file.

description =: default=""
General service description placed in high reliability/availability.

operator =: default= " <decisionEngine sameParam>"
Boolean Operator applied to the services in which a healthcheck is being executed . If "OR" even if only one service is an a state of failure the switch procedures are activitated, if "AND" all services must be in a state of failure to activate the restart procedures.

waitTimeAfterNormalPrimer =: default= " 180000" UM=Milliseconds
The waiting time after starting the first service available (normal startup). Once this time has passed the decision engine will begin to verify if the service has reached active status and if not, recovery procedures will be initiated. If the application starts up before this value is reached the Decision Engine starts verification immediately.

waitTimeAfterFlagBroken =: default= " 60000" UM=Milliseconds
The time to wait after the setting of persistent flags for the propagation to the other peers.

waitTimeBeforeTakeControl =: default= " 180000" UM=Milliseconds
The waiting time before taking control. Time allowed for a possible graceful shutdown of the previously active service if the resource is still reachable.

waitTimeAfterTakeControl =: default= ”900000” (15') UM=Milliseconds
The waiting time after taking control. Time allowed to wait for the conclusion of the take control process before returning to verifying the status of activities and then decide whether the operation was successful or retry with another resource if available. If the application starts up before this value is reached the Decision Engine starts verification immediately. This time is variable depending on the type of recovery and the amount of data if a database i s present.

applicationLostTime =: default value= " decisionEngine applicationLostTime" UM=Milliseconds
The waiting time from when the application being verified was declared in a state of failure (down). This is a very important value because once this period of time has passed and the application should still be unreachable, and the switch quorum verified, the process of recovery is triggered. The 30" are the minimum to avoid false positives.

applicationLostTimeBeforeRestart =: default= " <decisionEngine applicationLostTimeBeforeRestart>"

UM=Milliseconds
Time to wait before declaring the application lost and try a restart, if provided.
Even if the switch quorum is reached this time is still waited for before the restart of the resource is definitely in failure. If after this period the resource is back UP no further action is taken. The event is recorded in the log files and eventually reported via e-mail or HTTP post.

surfaceClusterWorkFlowHealthCheckUriPath =: default= "/HealthCheck"
The uri path of health check for the OPLON®Commander Work Flow service relating to this application.

surfaceClusterWorkFlowCommandUriPath =: default= "/SCWFCommand"
The uri path for the command of the OPLON®Commander Work Flow service relating to this application.

normalPrimerWorkFlow =: default= "normalPrimer" UM=Work Flow name
The name of the Work Flow that will be triggered if it is determined that intial startup of the service has occurred.

gracefulShutdownWorkFlow =: default= "gracefulShutdown”UM=Work Flow name
The name of the Work Flow that will be triggered if it is determined that the service is down and is executed immediately before performing the recovery action.

restartWorkFlow =: default= "null" UM=Work Flow name
The name of the Work Flow that will be triggered to attempt a restart before triggering the take-over procedures in another half-site.
If a value is not provided, this phase will not be executed.

takeControlWorkFlow =: default= "takeControl”UM=Work Flow name
The name of the Work Flow that will be triggered after the start of the gracefulShutdownWorkFlow to initiate the recovery action.

<failOverService>

```
<serviceconf>
        <surfaceclusterde>
                <decisionEngineMgr>
                        <decisionEngine>
                        <healthCheckServicesPolicy>
                                <failOverService
```

This section contains the information of an application service located on a specific system. The section therefore contains, by exception with respect to the <healthCheckServicesPolicy> section, all the parameters for performing a health check and application recovery.

enable =: default= "true" UM=boolean
Enables or disables the interpretation of this section in the instance.

description =: default=""

Precise description of the service placed in high reliability/availability. It is recommended, to the extent possible, to use the OPLON®Surface Clsuter nomenclature ex.: HalfSite A.A rather than HalfSite B.B with the short but comprehensive description of the service.

operator =: default= "<healthCheckServicesPolicy sameParam"
Boolean Operator applied to the services in which a healthcheck is being executed . If "OR" even if only one service is an a state of failure the switch procedures are activitated, if "AND" all services must be in a state of failure to activate the restart procedures.

surfaceClusterWorkFlowURL=: default= "https://"
The URL of the OPLON®Commander Work Flow instance relating to this application service.

waitTimeAfterNormalPrimer =: default= " <healthCheckServicesPolicy sameParam" UM=Milliseconds
The waiting time after starting the first service available (normal startup). Once this time has passed the decision engine will begin to verify if the service has reached active status and if not, recovery procedures will be initiated. If the application starts up before this value is reached the Decision Engine starts verification immediately.

waitTimeAfterFlagBroken =: default= " <healthCheckServicesPolicy sameParam" UM=Milliseconds
The time to wait after the setting of persistent flags for the propagation to the other peers.

waitTimeBeforeTakeControl =: default= " <healthCheckServicesPolicy sameParam" UM=Milliseconds
The waiting time before taking control. Time allowed for a possible graceful shutdown of the previously active service if the resource is still reachable.

waitTimeAfterTakeControl =: default= " <healthCheckServicesPolicy sameParam" UM=Milliseconds
The waiting time after taking control. Time allowed to wait for the conclusion of the take control process before returning to verifying the status of activities and then decide whether the operation was successful or retry with another resource if available. If the application starts up before this value is reached the Decision Engine starts verification immediately. This time is variable depending on the type of recovery and the amount of data if a database is present.

applicationLostTime =: default="<healthCheckServicesPolicy sameParam"UM=Milliseconds
The waiting time from when the application being verified was declared in a state of failure (down). This is a very important value because once this period of time has passed and the application should still be unreachable, and the switch quorum verified, the process of recovery is triggered. The 30" are the minimum to avoid false positives.

applicationLostTimeBeforeRestart =: default = " <healthCheckServicesPolicy sameParam" UM=Milliseconds
Time to wait before declaring the application lost and try a restart, if provided.

surfaceClusterWorkFlowCommandUriPath =: default= " <healthCheckServicesPolicy sameParam"
The uri path for the command of the OPLON®Commander Work Flow service relating to this application.

surfaceClusterWorkFlowHealthCheckUriPath =: default= " <healthCheckServicesPolicy sameParam"
The uri path of health check for the OPLON®Commander Work Flow service relating to this application.

normalPrimerWorkFlow =: default= " <healthCheckServicesPolicy sameParam"
The name of the Work Flow that will be triggered if it is determined that intial startup of the service has occurred.

gracefulShutdownWorkFlow =: default= "<healthCheckServicesPolicy sameParam”
The name of the Work Flow that will be triggered if it is determined that the service is down and is executed immediately before performing the recovery action.

restartWorkFlow =: default= "<healthCheckServicesPolicy sameParam" UM=Work Flow name
The name of the Work Flow that will be triggered to attempt a restart before triggering the take-over procedures in another half-site.
If a value is not provided, this phase will not be executed.

takeControlWorkFlow =: default= "<healthCheckServicesPolicy sameParam”
The name of the Work Flow that will be triggered after the start of the gracefulShutdownWorkFlow to initiate the recovery action.

<healthCheck>

```
<serviceconf>
    <surfaceclusterde>
        <decisionEngineMgr>
            <decisionEngine>
                <healthCheckServicesPolicy>
                    <failOverService>
                        <healthCheck
```

The healthCheck sections relating to application service placed in fail over, serve to determine the activity of the application service.
Applications on which Health Checks can be performed can be more than one, and even if only one of them fails during the test, the system triggers the verification of 'lostApplication' time out that once finished, if the switch quorum is positive, the recovery procedure is triggered.

enable =: default= "true"
If false the section is not taken into account.

description =: default=""
Summary description of the HealthCheck.

address =: default value=""
The address on which the health check is performed.

port =: default value= " 0"
The port on which the health check service responds. If <=0 will run a ICMP check.

SSL =: default value= "false"
If set to true performs the health check of the service through an SSL connection (HTTPS).

uriPath =: default value=""
The URIPath on which the health check service responds. If not present it will run a health check with a TCP connection.

numRetryConnection =: default value= " default from <params>"
The default number of attempts before declaring a service as no longer active. If unrated takes the default section <params>.

waitPerRetryConnection =: default value= " 300" UM=Milliseconds
Time to wait between connection attempts until numRetryConnection attempts is reached. If unrated takes the default section <params>.

createConnectionTimeOut =: default value= "default from <params>"
Time-out of the connection attempt. If unrated takes the default section <params>.

<healthCheckPublicPolicy>

```
<serviceconf>
    <surfaceclusterde>
        <decisionEngineMgr>
            <decisionEngine>
                <healthCheckPublicPolicy
```

Section designated to check the reachability of the public network through a minimum of 3 HealthChecks on addresses and distinct services.

<healthCheck>

```
<serviceconf>
    <surfaceclusterde>
        <decisionEngineMgr>
            <decisionEngine>
                <healthCheckPublicPolicy>
                    <healthCheck
```

Address or service to check based of the parameters entered.

enable =: default= "false"
If false the section is not taken into account.

description =: default=""
Summary description of the HealthCheck.

address =: default value=""
The address on which the health check is performed.

port =: default value= " 0"
The port on which the health check service responds. If <=0 will run a ICMP check.

SSL =: default value= "false"
If set to true performs the health check of the service through an SSL connection (HTTPS).

uriPath =: default value=""
The URIPath on which the health check service responds. If not present it will run a health check with a TCP connection.

numRetryConnection =: default value= " default from <params>"
The default number of attempts before declaring a service as no longer active. If unrated takes the default section <params>.

waitPerRetryConnection =: default value= " 300" UM=Milliseconds
Time to wait between connection attempts until numRetryConnection attempts is reached. If unrated takes the default section <params>.

createConnectionTimeOut =: default value= "default from <params>"
Time-out of the connection attempt. If unrated takes the default section <params>.

<healthCheckBackendPolicy>

```
<serviceconf>
    <surfaceclusterde>
        <decisionEngineMgr>
            <decisionEngine>
                <healthCheckBackendPolicy
```

Section designated to check the reachability of the back end network through a minimum of 3 HealthChecks on addresses and distinct services.

<healthCheck>

```
<serviceconf>
    <surfaceclusterde>
        <decisionEngineMgr>
            <decisionEngine>
                <healthCheckBackendPolicy>
                    <healthCheck
```

Address or service to check based on the parameters entered.

enable =: default= "false"
If false the section is not taken into account.

description =: default=""
Summary description of the HealthCheck.

address =: default value=""
The address on which the health check is performed.

port =: default value= " 0"
The port on which the health check service responds. If <=0 will run a ICMP check.

SSL =: default value= "false"
If set to true performs the health check of the service through an SSL connection (HTTPS).

uriPath =: default value=""
The URIPath on which the health check service responds. If not present it will run a health check with a TCP connection.

numRetryConnection =: default value= " default from <params>"
The default number of attempts before declaring a service as no longer active. If unrated takes the default section <params>.

waitPerRetryConnection =: default value= " 300" UM=Milliseconds
Time to wait between connection attempts until numRetryConnection attempts is reached. If unrated takes the default section <params>.

createConnectionTimeOut =: default value= "default from <params>"
Time-out of the connection attempt. If unrated takes the default section <params>.

17 OPLON®WorkFlow Split Brain Assasin

This service manages, in distributed cluster and stretch cluster (peered) environments , eventual possibilities of 'Split Brain'. The service bases its logic on the reachability of at least one of the two OPLON®Surface Cluster Decision Engine Peer nodes.

The structure of the configuration file reflects the simplicity that the service must maintain:

```
<serviceconf>
   <copyright>
   </copyright>
        <splitbrainassassin>
                <params>
                </params>
                <decisionEnginesPeers>
                    <peer/>
                    <peer/>
                    <notification/>
                    <notification/>
                    ...
                </decisionEnginesPeers>
        </splitbrainassassin>
</serviceconf>
```

<*splitbrainassassin*>

```
<serviceconf>
        <splitbrainassassin>
```

<*params*>

```
<serviceconf>
        <splitbrainassassin>
                <params
```

The <params> section describes the general operation of the service:

frequency =: default value= " 10000" UM=Milliseconds
The frequency of verification of status changes.

splitBrainTimeDetection =: default value= " 30000" UM=Milliseconds
The waiting time after an event of failure before setting up the outOfOrder notification files.

createConnectionTimeOut =: default value= " 5000" UM=Milliseconds
Default time out of the TCP/IP connection.

numRetryConnection =: default value= " 3"
The number of connection attempts before declaring the service unreachable.

<decisionEnginesPeers>

```
<serviceconf>
        <splitbrainassassin>
                <decisionEnginesPeers>
```

<peer>

```
<serviceconf>
        <splitbrainassassin>
                <decisionEnginesPeers>
                        <peer
```

This section, by necessity must be present 2 times, describes the HealthCheck parameters of the Decision Engine Peers services.

enable =: default value= "true"
Enables or disables the section.

Description =: default value=""
Description of the peer.

URL =: default value=""
URL for connecting to the peer service (normally https://___private__:54445/)

splitBrainTimeDetection=: default value= "params:splitBrainTimeDetection"
The waiting time after an event of failure before setting up the outOfOrder notification files.

Createconnectiontimeout =): default value= "params:createConnectionTimeOut"
Default time out of the TCP/IP connection.

numRetryConnection =: default value= "params:numRetryConnection"
The number of connection attempts before declaring the service unreachable.

healthCheckUriPath =: default value= " /HealthCheck?decisionEngine=SCDEGroup"
The path of health check including the group name of the cluster to observe.
The name of the Cluster group DE is determined by the Query String:

- ?decisionEngine=SCDEGroup

```
<decisionEnginesPeers>
        <peer enable="true"
                description="Sys 001"
                URL="https://oneprivate:54445/"
                createConnectionTimeOut="5000"
                numRetryConnection="3"
                healthCheckUriPath="/HealthCheck?decisionEngine=SCDEGroup"/>
        <peer enable="true"
                description="Sys 002"
                URL="https://quorumprivate:54445/"
                createConnectionTimeOut="5000"
                numRetryConnection="3"
                healthCheckUriPath="/HealthCheck?decisionEngine=SCDEGroup"/>
```

<notification>

```
<serviceconf>
        <splitbrainassassin>
                <decisionEnginesPeers>
                        <notification
```

enable =: default value= "true"
Enables or disables the section.

description =: default value=""
Description of the service for which the notification acts on.

fileName=: default value=""
Name of the notification file to manage. If the parameter is entered in relative form, it will be made absolute with the working directory
Ex.:
“lib/notificationDir/outOfOrder.systemsMonitorGroup”

Result:
“(LBL_HOME)/lib/notificationDir/outOfOrder.systemsMonitorGroup”

18 LBL ®Management Console

CLI Command Line

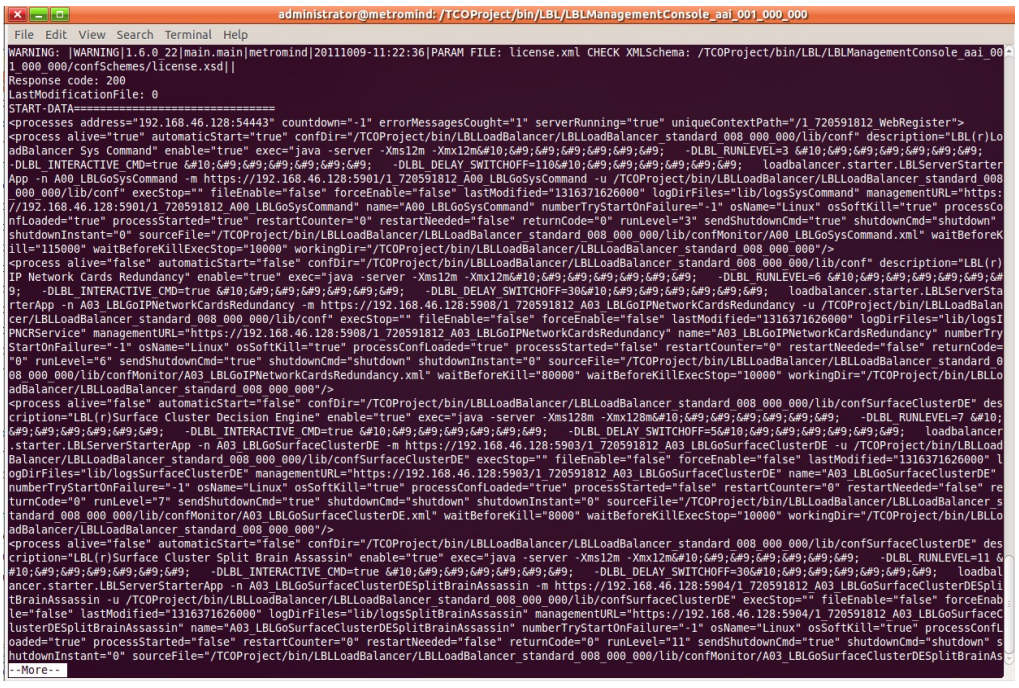
In the installation directory of OPLON® Management Console two executables are available:

- OPLON®ManagementConsole_start.jar
- LBLManagementConsole_startCLI.jar.

The first if executed starts the graphical console, the second allows performing operations directly from Command Line.

For example if a query of the LBL WebRegister to know what processes are being administered is desired, the command to run would be the following:

```
Java -jar LBLManagementConsole_startCLI.jar -L admin -P adminadmin -h
        roadubuntubenchmonitor:54443 -u "/WebRegister?command=processes"
```



The result is the same regardless of the OS being used - Linux/Unix systems or MS Windows.

The Command Line returns the result on standard output with the following format:

```
Response code: 200
LastModificationFile: 0
START-DATE================================
<processes address= "192.168.46.128 :54443" countdown= " -1" errorMess ...
..
…
      ….... ShutdownInstant= "0"
sourceFile="/TCOProject/bin/LBLLoadBalancer/LBLLoadBalancer_standard_008_00
0_000/lib/confMonitor/Z99_LBLPluginNetworkCheckWithCmdStart.xml"
waitBeforeKill="10000" waitBeforeKillExecStop="10000"
workingDir="lib/plugin"/>
</processes>

END-DATA==================================
```

The controller responds with two parameters of great importance.
The first, the Response code, has value as the HTTP response code, 200 is taken to mean OK.
The second, LastModificationFile, indicates in the case of downloading a file the date of last modification. This value is very important for the management of the concurrence in case of operations of download and upload where this value is controlled by the server.

If running the CLI without any parameter the help of commands is returned:

```
$ java -jar LBLManagementConsole_startCLI.jar
....others service info...
Usage: -L login -P password -h hostaddress:portNumber -u uriPath [-f file -
l lastModificationObject]
```

In the face of an incorrect (server) command the following is returned
Ex.:

```
    $java -jar LBLManagementConsole_startCLI.jar -L admin -P adminadmin -h
roadubuntubenchmonitor:54443 -u "/WebRegister?command="
```

```
Response code: 404
START-MSG================================
001 Usage:
/WebRegister?command=rel
/WebRegister?command=env
/WebRegister?command=stat
/WebRegister?command=gc
/WebRegister?command=processes
/WebRegister?command=processStop&target=processName
/WebRegister?command=processStart&target=processName
```

```
/WebRegister?command=processStart&target=processName
/WebRegister?command=services&target=processName
/WebRegister?command=infoXmlConfig&target=processName&subTarget=serviceName
/WebRegister?
command=downloadXmlConfig&target=processName&subTarget=serviceName
/WebRegister?
command=uploadXmlConfig&target=processName&subTarget=serviceName
/WebRegister?
command=downloadXsdConfig&target=processName&subTarget=serviceName
/WebRegister?
command=listBackupXmlConfig&target=processName&subTarget=serviceName
/WebRegister?
command=downloadBackupXmlConfig&target=processName&subTarget=serviceName&fi
le=fileName
/WebRegister?command=infoXmlProfile&target=processName
/WebRegister?command=downloadXmlProfile&target=processName
/WebRegister?command=downloadXsdProfile
/WebRegister?command=uploadXmlProfile&target=processName
/WebRegister?command=listBackupXmlProfile&target=processName
/WebRegister?
command=downloadBackupXmlProfile&target=processName&file=fileName
/WebRegister?command=listLogFiles&target=processName
/WebRegister?command=downloadLogFile&target=processName&file=fileName
/WebRegister?
command=downloadLastLinesLogFile&target=processName&file=fileName&lines=num
Lines
/WebRegister?command=services
/WebRegister?command=serviceReinit&target=serviceName
END-MSG==============================
```

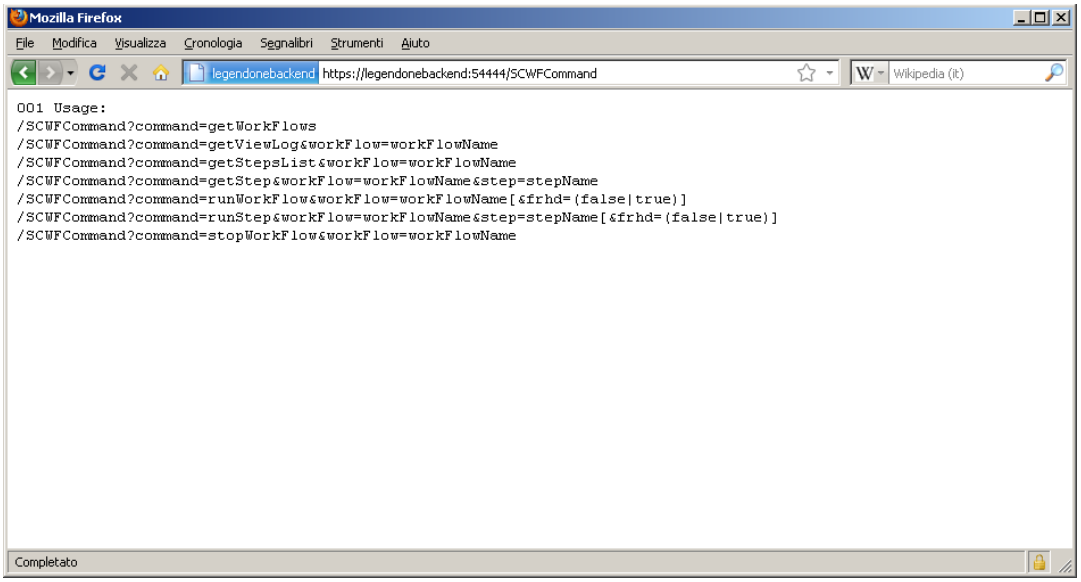
The response code is set to 404, not found, and the “application” message " is described between tags START_MSG===... and END-MSG===.....

Commander :URL for remote access

The integration of other instruments of control in the OPLON®Commander modules is possible through the modes described thus far.
The use of OPLON®Commander remote controls is possible through the following URL:

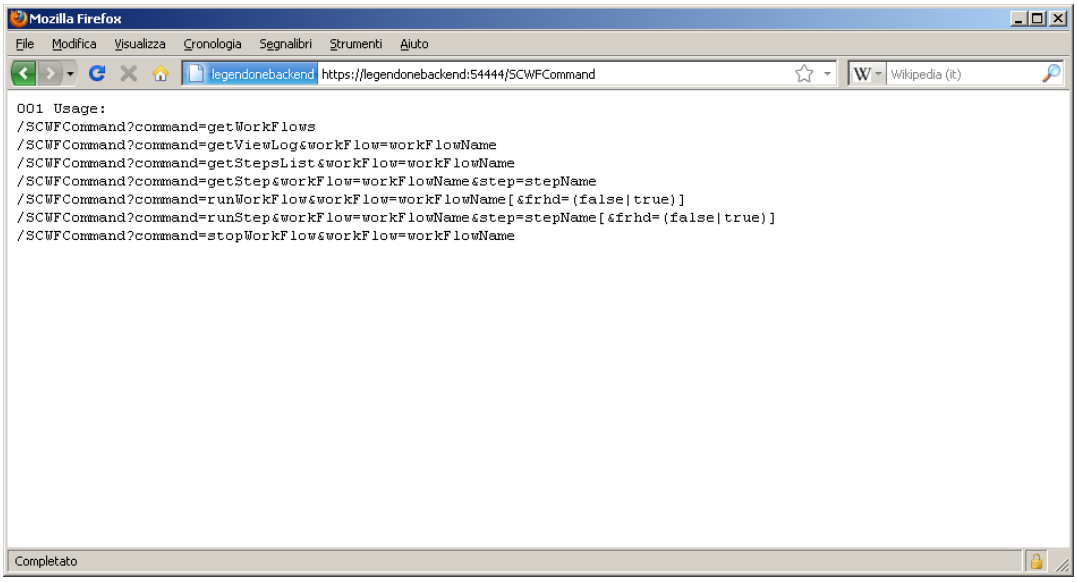
Oplon® Commander Decision Engine:

- Https://hostnamebackend:54445/SCDECommand



Oplon® Commander Work Flow

- Https://hostnamebackend:54444/SCWFCommand



By carrying over the same parameters in SDK calls or CLI it is possible to integrate other tools in a simple and effective manner.

SDK Software Development Kit

In the installation directory of OPLON® Management Console an executable is made available:

- LBLManagementConsole_startCLI.jar.

This file allows operations directly from the Command Line, and can be further used as a class library for use within programs written in Java.

The use of the library occurs from the characteristics of JAVA builds. Therefore libraries required for compilation and use of the generated program must be included.

For this purpose, below is an example of a simple program that returns the XML file iproxy.xml :

```
import managementconsole.starter.LBLManagementConsoleCLI;

/**
 * Request LBL Service Management command from java program
 * @author TCOProject(tm)
 * @version 1.0 Created on 23-set-2011, 12.00.17
 */
public class LBLTestRequestFromJavaProgram {

   /** copyright */
  public static final String COPYRIGHT = "LBL and TCOProject are trademarks of F.Pieretti";

  public static void main(String[] argv) {
       LBLManagementConsoleCLI programInterface = new LBLManagementConsoleCLI();
      StringBuilder result =
         programInterface.goCommandGET("admin",
                                       "adminadmin",
                                       "legendonebackend:54443",
                                       "/WebRegister?command=downloadXmlConfig&target=A10_LBLGo&subTarget=iproxy");
     // out result...
     System.out.println(result);

     // get a last modification object for next post a modified file...
     System.out.println("LastModificationObject="+programInterface.getLastModificationObject());
      // response code
     System.out.println("ResponseCode="+programInterface.getResponseCode());
   }
}
```

For the compilation of this program proceed as follows from the directory of the management console:

javac -classpath LBLManagementConsole_startCLI.jar LBLTestRequestFromJavaProgram.java

For the start of the program:

java -classpath LBLManagementConsole_startCLI.jar LBLTestRequestFromJavaProgramm

The class *LBLManagementConsoleCLI* offers 4 public methods to enable execution of all the operations:

```
          /**
           * Command get
           * @param login login
           * @param password password
           * @param hostAddress hostname:port
           * @param URIPathParams uripath params ex. /WebRegister?command=processes
           * @return value or message
           */
```

```
public StringBuilder goCommandGET(String login,
                                  String password,
                                  String hostAddress,
                                  String URIPathParams)

/**
 * Command post a file
 * @param login login
 * @param password password
 * @param hostAddress hostname:port
 * @param URIPathParams uripath params ex. /WebRegister?command=processes
 * @param file file to post
 * @param lastModificationObject last modification object
 * @return value or message
 */
public StringBuilder goCommandPOST(String login,
                                   String password,
                                   String hostAddress,
                                   String URIPathParams,
                                   String file,
                                   long lastModificationObject)

/**
 * Return a last modification date or 0L if not applicable
 * @return last modification date or 0L if not applicable
 */
public long getLastModificationObject()

/**
 * HTTP response code
 * @return HTTP response code
 */
public int getResponseCode()
```

19

OPLON® Autentication

Oplon®Secure Access rel. 9 introduces delegated authentication in all communications between sensitive components.
The implementation is gained from the experience in mission-critical datacenter to increase security and decrease the possibility of unauthorized use of procedures within the datacenter (s). Programming automation functions via remote shell or netsh, even with encrypted passwords and centrally logged access, expose modern, virtualization-based datacenters to serious danger. Today's possibility of loading an entire operating system inside virtualizers in Personal Computers drastically compromises security by giving the possibility to remotely perform highly dangerous procedures if performed to cause damage.

Business-continuity and Disaster-Recovery procedures impose the need to write many scripts that interact at various levels of the datacenter often with administrator / root privileges. A simple malicious operation of reversing storage replicas, launched for example by a cloned operating system, can cause incalculable damage.

In this regard, all the password files of the Oplon®Secure Access rel. 9 are encrypted and can only be used by the location and operating system on which they originated. Any cloning of the system or theft of the authentication files makes the files unreadable and therefore any operation not possible.

This document relates only to configuring user authentication and delegated authentication. For the installation of Oplon®Secure Access Components refer to the installation documents of the individual products.

Introduction

The implementation of Oplon®Secure Access Login authentication Rel 9 is based on 4 fundamental points:

1. Encryption of authentication repositories.
2. Origin of creation of authentication files.
3. Separation of user and delegated authentication.
4. Autonomy with respect to the infrastructure to be managed.

The first (1) point is very important as the authentication files are encrypted to allow their visibility or readability with only the tools made available by the OPLON®S.A.A.I platform.

The second (2) point poses a significant improvement in safety. All files are unreadable, even by Oplon®Secure Access Tools, if they are copied or if they are used on cloned virtual or

physical machines.

The third point (3) separates the management of human users from delegated users or OPLON®S.A.A.I systems. automation systems such as OPLON®Surface Cluster Decision Engine and OPLON®Surface Cluster Work Flow that can interact on multiple levels of data centers.

The fourth point (4) highlights how the authoritative system must be completely autonomous from the infrastructure it has to manage. The authorization delegated to the action is therefore completely disconnected for example from Directory Server which may not be available at the time of the fail-over procedures.

Scenario

The delegated authentication scenario is a set of actions that must be performed within the datacenter on multiple infrastructure levels such as Storage Area Network, Database, Directory server, application server and on multiple operating platforms, physical and virtual: Different operating systems, systems of different virtualization. A typical scenario of Business-continuity and Disaster-recovery environments.

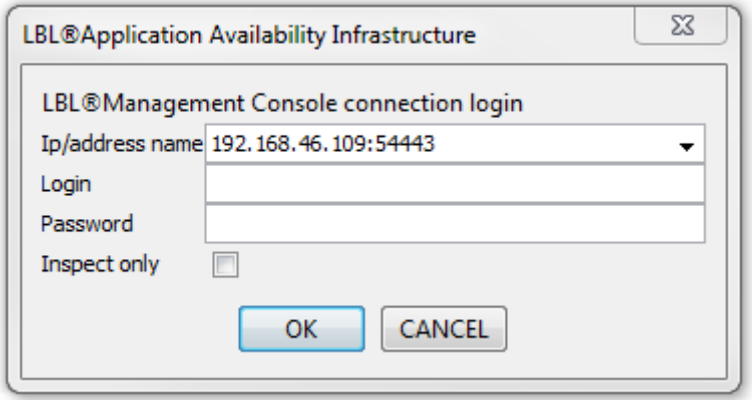
OPLON®Surface Cluster Work Flow allows you to perform the necessary operations through different architectural layers by invoking a remote operation called Remote Workflow Command (RWC from now on) in one step. An RWC operation allows you to run an entire Workflow or a Workflow starting from a designated step or just one step.

By way of example, here is a RWC where # LBL_ADDRESS_RWC_TARGET # is the variable that contains the address on which to perform the operation while selfTestNormalEnd is the step from which to start the execution of the workflow. For further information see the manual LBL_SurfaceCluster_Installation.pdf.

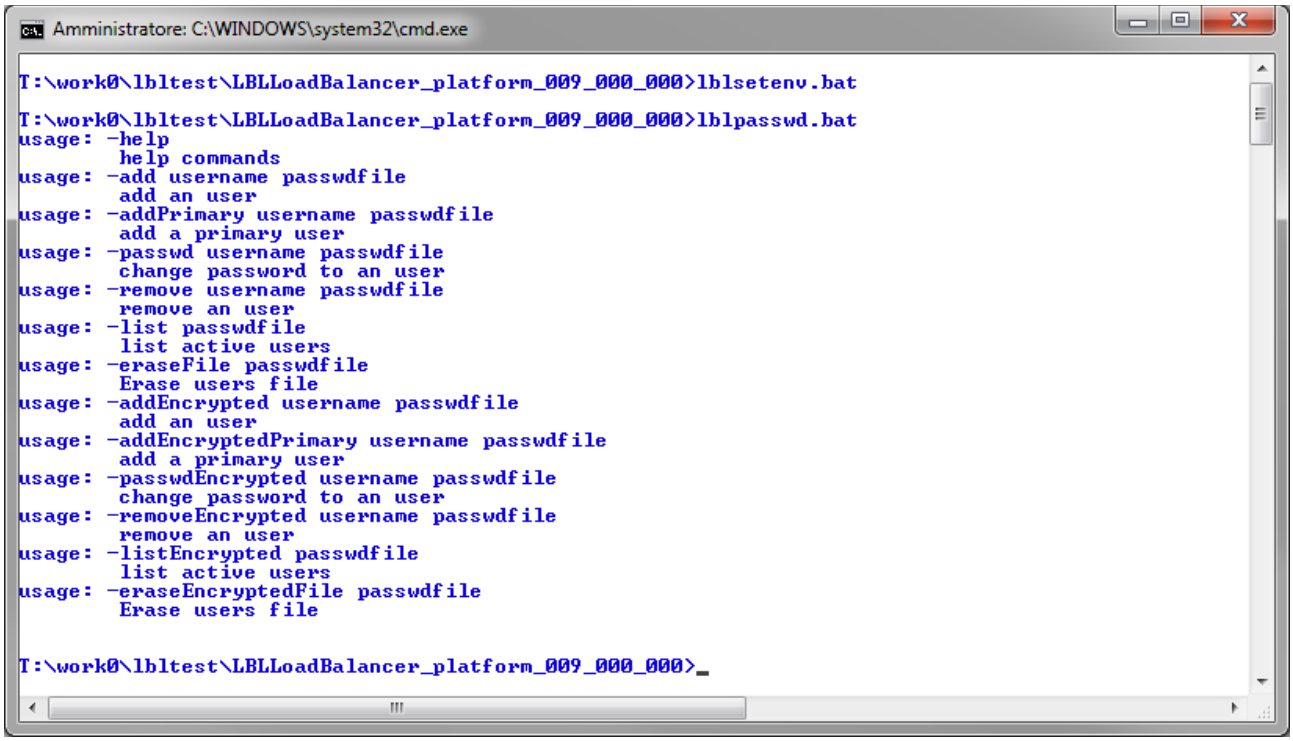
```
<step enable="true"
      name="rwcTest"
      description="Self Test ab end"
      waitBeforeExecute="1000"
      sysCommandTimeOut="600000"
      evaluateReturnCode="true"
      command="@RWC hostName=#LBL_ADDRESS_RWC_TARGET# workFlow=selfTestNormalEnd"
      maxRetry="3"
      gotoWhenFalse="abEnd">
      <returncode value="0"
                  description="ok"
                  result="true"/>
</step>
```

As you can see, in the parameters there is no authorization element such as Login or Password. In fact, the authorizations are contextual to the person or automaton that starts the workflow and are propagated to the remote nodes that will have to perform the operation. Remote nodes based on their authorization repository will check if the request comes from a trusted source and if so they will perform the operation.

User permissions
User authorizations are the authorizations that are used by people who are going to work with the OPLON®S.A.A.I .. tools. For example, when starting OPLON® Management Console, a user authoritative login password is requested:



User permissions are set via the lblpasswd (.bat or .sh) command from the (LBL_HOME) directory. The first thing to do is to set the environment variables via lblsetenv.bat / .sh and then run lblpasswd.bat / .sh



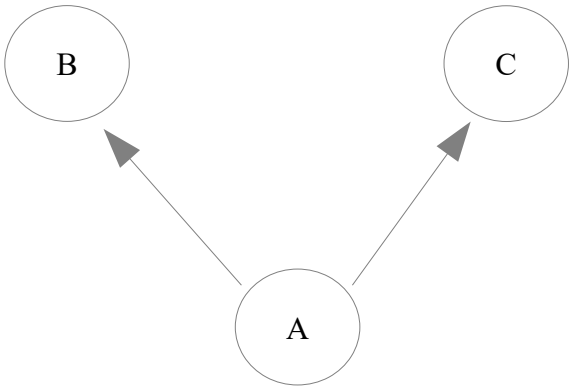
The commands for setting encrypted and non-encrypted passwords are equivalent for compatibility reasons (From Oplon®Secure Access Rel 9 all password repositories are encrypted). For example, the -add and -addEncrypted command perform the same operation.

Delegated permissions
Delegated authentication is used to authorize an operation that must cross multiple layers

and where the final execution layer does not necessarily know the user or automaton that performed the first operation.

For example, we can assume an organization that needs to have three sites with different execution permissions depending on the location from which you want to perform the operation.

Suppose the authorization policies allow site A to be able to perform operations on site B and C but site B and C can only perform operations on themselves.



Delegated authentication and authorization relies on an authoritative repository other than the user repository. In order to use and modify the delegated repository it is necessary to use the utility (LBL_HOME) /lbldelegatedpasswd.bat/.sh.

lbldelegatedpasswd.bat / .sh uses the same commands that are available to lblpasswd.bat / .sh but the treatment of the repositories by the programs that interpret it is slightly different. The repository must contain a delegated user who is considered primary. To set up the primary user, simply perform:

(LBL_HOME) /llbldelegatedpasswd.bat -addPrimary myPrimary

If you have already installed the Oplon®Secure Access Component surely the error message will be reported:

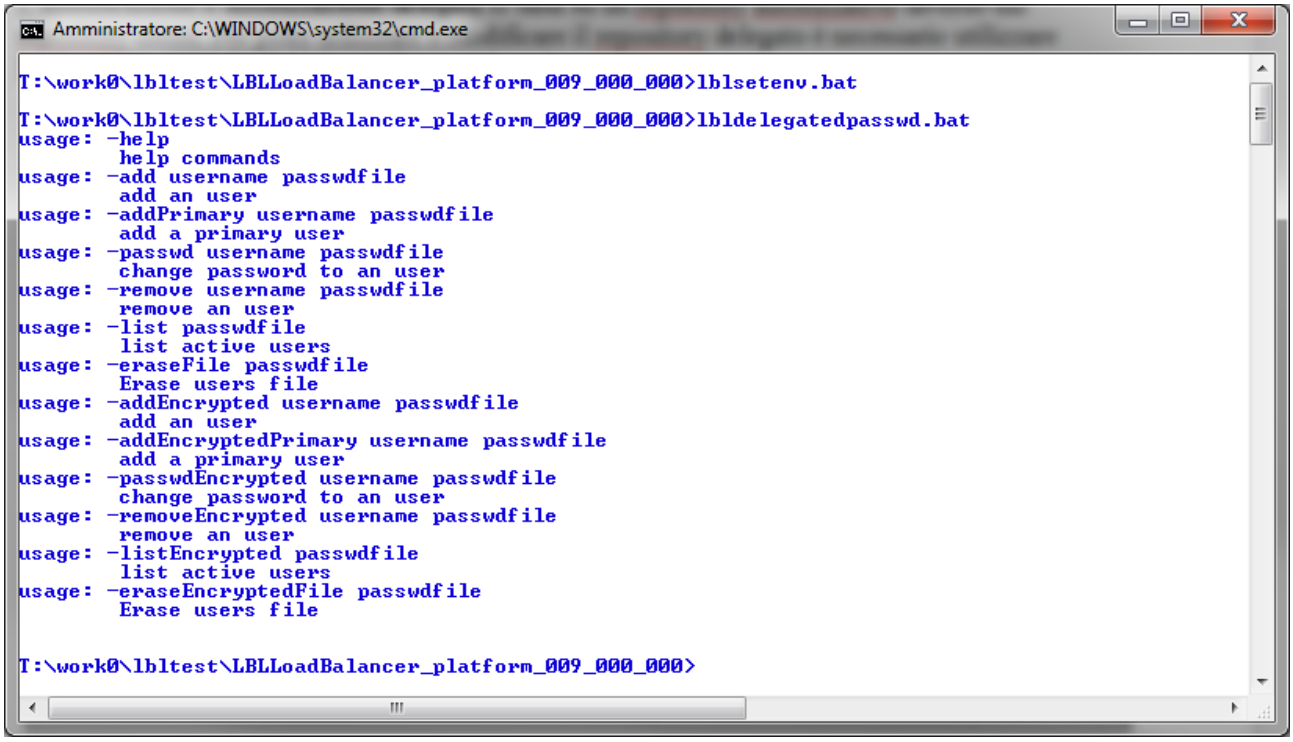
(LBL_HOME) /llbldelegatedpasswd.bat -addPrimary myPrimary
Primary user already exists!

The message is due to the presence of a primary user already set in the repository during the initial setup with the command (LBL_HOME) /lblinit.bat/.sh

The primary user proposed by the command (LBL_HOME) /lblinit.bat/.sh can be viewed using the command:

```
(LBL_HOME)/lbldelegatedpasswd.bat -list
delegated primary
```

(LBL_HOME)>



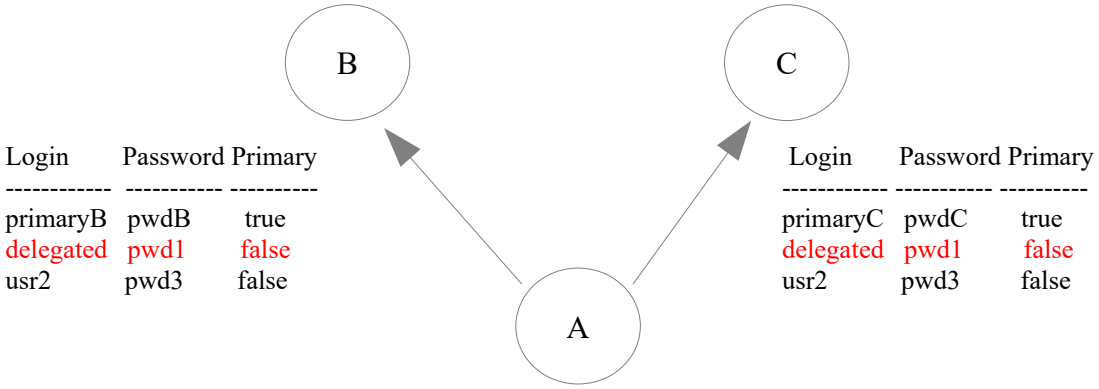
In summary, the delegated authentication repository must have a single user inside it that has been defined as primary. The primary user will be used by the "client" program, that is the one who requests to do a remote operation, proposing his credentials. The "server" program, ie the command target, must have a user with the same credentials (login and password). The "server" program does not necessarily have to have the same user declared primary, but must contain him in its list of delegated users.

Login	Password	Primary	Login	Password	Primary
delegated	pwd1	true	primaryB	pwdB	true
usr1	pwd2	false	delegated	pwd1	false
usr2	pwd3	false	usr2	pwd3	false

In this case the delegated action of node A can be performed towards node B as B contains a user with the same login and password even if he is not primary in B. Node B cannot perform any delegated action towards node A as although it contains the "delegated" login with credentials equal to node A, B will perform an action towards node A exposing its credentials with the "primaryB" login which is not present in node A and which will therefore refuse the operation.
The technique is very simple and therefore very effective. This technique also allows, in situations where a stringent security policy is not required, to perform self-consistent

configurations with a few settings.

If we go back to our initial example and fully implement it below a possible authoritative scheme.

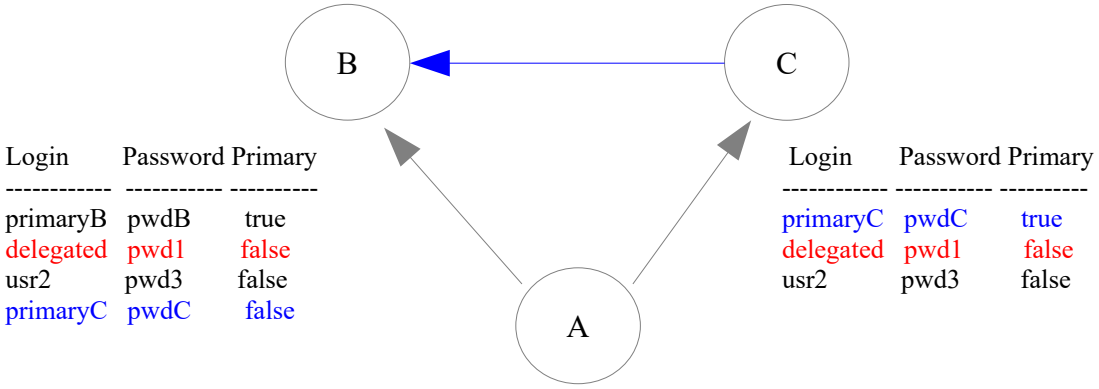


Login	Password	Primary
primaryB	pwdB	true
delegated	pwd1	false
usr2	pwd3	false

Login	Password	Primary
primaryC	pwdC	true
delegated	pwd1	false
usr2	pwd3	false

Login	Password	Primary
delegated	pwd1	true
usr1	pwd2	false
usr2	pwd3	false

With this authoritative scheme, only site A is able to carry out cross-site operations while sites B and C can only carry out operations within them. If at a later time we want to enable operations between site C and site B, the operation will be very simple by adding the primary C user to site B.



Login	Password	Primary
primaryB	pwdB	true
delegated	pwd1	false
usr2	pwd3	false
primaryC	pwdC	false

Login	Password	Primary
primaryC	pwdC	true
delegated	pwd1	false
usr2	pwd3	false

Login	Password	Primary
delegated	pwd1	true
usr1	pwd2	false
usr2	pwd3	false

External user permissions

User permissions can be easily integrated with external authentication systems through the use of java plugins.
In this case, in response to a login request, LBL authenticates the user through the external authentication system. In case of failure of the authentication system, LBL proceeds with the authentication of the user using the internal repository.

Java plugin
In order to interface LBL with an external authentication system, it is necessary to write a java plugin, which implements the login method of the abstract class TCOAASLoginAbstr.

public abstract TCOAASUser login (String login, String password);

The login method returns an object of type TCOAASUser which contains the user's characteristics.

```
package tco.lib.aas;

/*
 * Copyright 1998, by OPLON NETWORKS SRL
 * www.oplon.net
 * All rights reserved.
 *
 * This software is the confidential and proprietary information
 * of OPLON NETWORKS . ("Confidential Information").
 * You shall not disclose such Confidential Information and shall use
 * it only in accordance with the terms of the license agreement
 * you entered into with OPLON NETWORKS
 */

/**
 * User profile
 * @author OPLON NETWORKS
 * @version 1.0 Created on 23-mag-2014, 13.02.41
 */
public class TCOAASUser
{
	/** copyright */
	public static final String COPYRIGHT="TCOProject is a trademark, All rights reserved";

	/** login */
	private final String login;

	/** user roles */
	private final String roles;

	public TCOAASUser(String login, String roles) {
		this.login = login;
		this.roles = roles;
	}

	@Override
	public String toString() {
		return " Login="+login+" rules="+roles;
	}
```

```
	/**
	 * user roles
	 * @return the roles
	 */
	public String getRoles() {
		return roles;
	}

	/**
	 * login
	 * @return the login
	 */
	public String getLogin() {
		return login;
	}
}
```

An example of an LDAP plugin can be found in

```
../LBL_HOME/security/aas/LBLAALLoginInternalTest.java
```

The LBLAALLoginInternalTest.java class proceeds to load the TCOAASUser object through the Login method, after verifying the credentials via LDAP.
In the event that LDAP responds negatively to login, or if the authorization system fails, the Login method leaves the TCOAASUser object initialized to null. In this case, LBL authenticates through its internal user authentication repository.

```
package aasAA;

/*
 * Copyright 2010, by OPLON NETWORKS
 * Via Savonarola, 217 35137 Padova - Italy.
 * All rights reserved.
 *
 * This software is the confidential and proprietary information
 * of OPLON NETWORKS . ("Confidential Information").
 * You shall not disclose such Confidential Information and shall use
 * it only in accordance with the terms of the license agreement
 * you entered into with OPLON NETWORKS
 */

import java.util.Properties;

import javax.naming.Context;
import javax.naming.NamingException;
import javax.naming.NamingEnumeration;

import javax.naming.directory.Attribute;
import javax.naming.directory.Attributes;
import javax.naming.directory.DirContext;
import javax.naming.directory.SearchResult;
import javax.naming.directory.SearchControls;
import javax.naming.directory.InitialDirContext;

import tco.lib.aas.TCOAASLoginAbstr;
import tco.lib.aas.TCOAASUser;
```

```
/**
 * Test LDAP Login
 * @author TCOGROUP S.R.L.
 * Example of use:
 * @version 1.0 Created on 23-may-2014
 */
public class LBLAASLoginInternalTest extends TCOAASLoginAbstr
{
	/** copyright */
	public static final String COPYRIGHT="TCOProject is a trademark, All rights reserved";

	@Override
	public TCOAASUser login(String login, String password) {
		TCOAASUser returnValue = null;

		String lblUserDnFilter = "(&(mail=%login%))";
		String ldapServerURL = "ldap://192.168.45.207:389/dc=tcoproject,dc=com";
		boolean logged = loginProcess("com.sun.jndi.ldap.LdapCtxFactory", ldapServerURL,
"ou=utenti", lblUserDnFilter, login, password, "gidNumber");
		logDebug("logged:" + logged);
		if (logged) {
			returnValue = new TCOAASUser(login, "");
		}
		return returnValue;
	}

	private boolean loginProcess(String InitialContextFactory,
								String ldapServerUrl,
								String ldapSearchBase,
								String lblUserDnFilter,
								String lblUsername,
								String lblPassword,
								String lblProfileAttribute) {
		boolean ret = false;
		Properties env = new Properties();
		env.put(Context.INITIAL_CONTEXT_FACTORY, InitialContextFactory);
		env.put(Context.PROVIDER_URL, ldapServerUrl);
		env.put(Context.SECURITY_AUTHENTICATION, "none");

		SearchControls searchCtrls = new SearchControls();
		searchCtrls.setReturningAttributes(new String[]{lblProfileAttribute});
		searchCtrls.setSearchScope(SearchControls.SUBTREE_SCOPE);
		DirContext ctx = null;
		try {
			ctx = new InitialDirContext(env);
			NamingEnumeration<SearchResult> answer = ctx.search(
					ldapSearchBase, lblUserDnFilter.replaceAll("%login%",
lblUsername), searchCtrls);

			String fullDN = null;

			if (answer.hasMore()) {
				SearchResult sr = answer.next();
				fullDN = sr.getNameInNamespace();
				logDebug(fullDN);
				if (!answer.hasMore()) {
					// ottenuto un dn unico, faccio login
```

```
                                        ctx.close();
                                        ctx = null;

                                        env.put(Context.SECURITY_AUTHENTICATION, "simple");
                                        env.put(Context.SECURITY_PRINCIPAL, fullDN);
                                        env.put(Context.SECURITY_CREDENTIALS, lblPassword);

                                        ctx = new InitialDirContext(env);
                                        ret = true;

                                        Attributes attrs = sr.getAttributes();
                                        for (NamingEnumeration ae = attrs.getAll(); ae.hasMore();) {
                                                Attribute attr = (Attribute) ae.next();
                                                System.out.print("attribute:<" + attr.getID() + ">=");
                                                /* Print each value */
                                                for (NamingEnumeration e = attr.getAll(); e.hasMore();) {
                                                        logDebug("<" + e.next() + ">");
                                                }
                                        }
                                }
                                ctx.close();
                        }
                } catch (Throwable e) {
                        logError(lblPassword);
                } finally {
                        if (ctx != null) {
                                try {
                                        ctx.close();
                                } catch (NamingException ex) {
                                        /* no actions */
                                }
                        }
                }
                return ret;
        }

        public static void main(String[] args) {
                String lblUser = "valerio.mezzalira@tcoproject.com";
                String lblPasswd = "valerioadmin";
                TCOAASLoginAbstr testLogin = new LBLAASLoginInternalTest();
                TCOAASUser user = testLogin.login(lblUser, lblPasswd);
                if (user!=null) {
                        System.out.println(" ->"+user);
                } else {
                        System.out.println(" ->not authenticated");
                }
        }
}
```

Compilazione

Once written, the plugin java class must be placed in the folder

```
LBL_HOME/security/aas
```

To compile the class, simply use the compile.sh/bat script.

```
$ sudo sh ./compile.sh *.java
```

Configuration

In order for LBL to authenticate using an external plugin it is necessary to modify the _loginAAS.xml file located in:

```
(LBL_HOME)/lib/webroot/webappsconf/_loginAAS.xml
```

Change the className attribute of the authorization tag to the name of the plugin class

(Default value):

```
<contextRoot>
        <authorizations className="tco.lib.aas.TCOAASLoginDefaultImpl"/>
</contextRoot>
```

(template example):

```
<contextRoot>
        <authorizations className="aas.LBLAASLoginInternalTest"/>
</contextRoot>
```