oplon®

OPLON®
Secure Access
Rel. 10.11.3
Reference Guide

OPLON®
Release 10.11.3
Document Rev. 4.0.0
Date 24 agosto 2024

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

Table of Contents

Descrizione del prodotto e versioni distribuite

Oplon®Secure Access è un insieme di strumenti progettati per aumentare la disponibilità dei servizi applicativi garantendo la massima sicurezza ottenibile con la tecnologia allo stato dell'arte.

Le scelte di progetto sono basate su attenti studi preliminari e da esperienze significative nell'ambito dell'elaborazione concorrente iniziate nel 1986 che hanno permesso l'implementazione di un modello di riferimento sul quale poggia l'intero sistema. Oplon®Secure Access deriva da una lunga esperienza maturata in numerosi progetti mission-critical che hanno permesso di far acquisire alla soluzione le caratteristiche di semplicità ed affidabilità tipiche di questo settore. Oplon®Secure Access comprende diversi prodotti e funzionalità rilasciate in distribuzioni commerciali:

1
Configurazione & Setup

Note prima dell'installazione

I prodotti OPLON® sono destinati ad ambienti mission critical pertanto solo personale che ha effettuato il corso ed ha superato l'esame è autorizzato a certificare l'installazione e manutenzione dei prodotti in esercizio. Tutte le persone certificate sono dotate di attestato di partecipazione ai corsi e superamento della prova di esame rilasciati da OPLON NETWORKS.

Prima di procedere all'installazione della componente server è consigliato eseguire l'installazione della componente OPLON®Management Console per poter effettuare le operazioni di configurazione.

Per l'installazione della componente OPLON®Management Console fare riferimento al manuale OPLON_ManagementConsole_Installation.pdf disponibile nei supporti di memorizzazione forniti con il prodotto o attraverso download da area riservata.

Per installare i prodotti OPLON® è obbligatorio possedere le licenze necessarie. Queste si possono richiedere a OPLON NETWORKS attraverso una e-mail indirizzata a info@oplon.net, oppure rivolgendosi ad un distributore autorizzato.

Ulteriori informazioni si possono trovare sul sito
www.oplon.net

Preparazione all'installazione: directory tree

Il file di prodotto è un unico file in formato compresso con il nome così composto:

LBLLoadBalancer_aai_999_999_999.zip

Dopo il nome del prodotto,LBLLoadBalancer, nella seconda parte del nome viene indicata a la distribuzione contenuta (es Platform, Standard, Enterprise, datawarehouse etc.) mentre le parti numeriche indicano rispettivamente la release, la sottorelease e la build.

Una volta che il file è stato decompresso la struttura delle directory ed il contenuto è simile all'esempio seguente.
La posizione della directory: “LBLLBLLoadBalancer _aai_999_999_999” identifica la home directory del prodotto,di seguito indicata come (LBL_HOME). La variabile di ambiente LBL_HOME, necessaria per il corretto funzionamento dei prodotti indica la directory suddetta.

```
├──LBLLoadBalancer_aai_999_999_999   (LBL_HOME)
├───interceptors              (classi dinamiche di cattura ed elaborazione delle informazioni)
```

```
|    ├──dwhInterceptors      (classi dinamiche di trattamento del database di storicizzazione DWH)
|    └──rewriteclasses       (classi dinamiche di trattamento e rewriting dei dati)
├──legacyBin                (utility tipizzate per Sistema Operativo)
|    ├──AIX
|    ├──DatabasesScript      (Script database networked)
|    ├──HP-UX
|    ├──Linux
|    ├──SunOS
|    └──Windows
├──lib               (Directory delle configurazioni comuni a tutti i servizi)
|    ├──conf          (File di configurazione del MONITOR)
|    ├──confMonitor   (Directory con i profili di lancio dei processi del MONITOR)
|    ├──confSchemes   (Directory con gli scheme XSD delle configurazioni)
|    ├──extLib        (Directory per librerie esterne, mail, dbclient etc.)
|    ├──logs          (logs del MONITOR)
|    ├──notificationDir      (Directory notifica OutOfOrder servizi o file di geolocalizzazione)
|    ├──plugin               (default directory plugin)
|    |   └──lblplugin
|    ├──webroo               (directory root servizi OPLON®Monitor WebConsole)
|    |   ├──webapps          (home directory web del Monitor)
|    |   ├──webappscon       (file di configurazione applicazioni web del Monitor)
|    |   └──websecurity      (certificati digitali di default per gestione HTTPS dei server Web)
|    |       └──certificate
|    ├──webroot_remotebatch          (home directory Remote Batch HTTP server)
|    ├──webroot_statisticbrokerweb   (home directory Statistic Web Cache HTTP server)
|    ├──webroot_surfaceclusterde     (home directory Surface Cluster Decision Engine  HTTP server)
|    ├──webroot_surfaceclusterwf     (home directory Surface Cluster Work Flow HTTP server)
|    └──webroot_syscommand           (home directory SysCommand  HTTP server)
├──procsProfiles                          (Directory home dei processi gestiti dal MONITOR)
|    ├──A00_LBLGoSysCommand               (home dir Sys Command)
|    ├──A01_LBLIPGeolocalizationDownloader  (home Geo Localization downloader)
|    ├──A03_LBLGoIPNetworkCardsRedundancy   (home IP Network Card Redundancy)
|    ├──A03_LBLGoSurfaceClusterDE           (home Surface Cluster Decision Engine)
|    ├──A03_LBLGoSurfaceClusterDESplitBrainAssassin (home Surface Cluster Split Brain Assassin)
|    ├──A03_LBLGoSurfaceClusterWF           (home Surface Cluster Work Flow)
|    ├──A03_LBLGoSurfaceClusterWFRemoteBatch(home Surface Cluster Remote Batch)
|    ├──A05_LBLGoDNSManager                 (home DNS Manager)
|    ├──A05_LBLGoStatisticsWebCache         (home Statistic Boroker Web Cache
|    ├──A10_LBLGo                           (home LoadBalancer)
|    ├──Z99_LBLPluginHTTPCheckWithCmdStart  (home Plugin di Health Check proattivo HTTP)
|    └──Z99_LBLPluginNetworkCheckWithCmdStart      (home Plugin di Health Check proattivo TCP)
├──resources
└──security
    └──certificate
```

Al primo start altre directory verranno automaticamente inizializzate. Per una completa panoramica delle directory vedere OPLON®Secure Access Reference Guide

NOTA Il contenuto della cartella procProfiles varia a seconda della distribuzione

Variabili ambiente: file lblsetenv

Per poter eseguite OPLON®ADC sono necessarie 2 variabili d'ambiente:

LBL_HOME:= e' la variabile d'ambiente che deve contenere la posizione della home directory OPLON®ADC.

LBL_JAVA_HOME:= e' la variabile d'ambiente che deve contenere la posizione della home directory "java".

Ogni shellscript prima di eseguire qualsiasi comando controlla l'esistenza delle variabili di

ambiente.
Oltre all'esistenza delle variabili ambiente, in LBL_HOME è verificatsa l'esistenza del file (LBL_HOME)/lib/LBLLoadBalancer.jar, mentre in LBL_JAVA_HOME, l'esistenza del file (LBL_JAVA_HOME)/java (Unix/Linux) e (LBL_JAVA_HOME)/java.exe (MS Windows).

Per poter agevolmente eseguire i primi start e si è appena eseguito l'unzip del prodotto, si possono trovare 2 file per sistemi operativi MS Windows oppure Unix/Linux: lblsetenv.bat e lblsetenv.sh.

Editare il file del sistema operativo ospite, togliere il commento dalle ultime 2 righe e inserire le posizioni assolute delle home directory di OPLON®Traffic Mometizer e JAVA.

es. Unix/Linux DA:

```
#!/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.tcoproject.com
#
#                     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
```

A:

```
#!/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.tcoproject.com
#
#                     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
```

NOTA: La porzione di “os optimization” imposta, nei sistemi Unix-Linux, i parametri necessari all'utilizzo di OPLON®ADC.
La variabile LBL_NET_DEV elenca le interfacce di rete utilizzate da LBL mentre LBL_MULTICAST_DEV elenca le interfacce di rete su cui deve essere abilitato il multicast (necessario solo nelle interfacce di heart-beat delle versioni Standard ed Enterprise)

es. MS Windows DA:

```
@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.tcoproject.com
:
:                     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
```

A:

```
@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.tcoproject.com
:
:                     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
```

Per effettuare l'impostazione delle variabili ambiente nella sessione di lavoro corrente

eseguire per Unix/Linux:

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

per MS Windows eseguire:

```
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
```

ATTENZIONE: Questa impostazione di variabile è locale alla sessione di lavoro. Per rendere permanenti le impostazioni agire sui file o le configurazioni del sistema operativo o sugli script di lancio automatico.

Internet protocol: IPv4 IPv6

Oplon®Secure Access è stato realizzato per utilizzare indifferentemente i protocolli IPv4 ed IPv6.

La modalità di utilizzo degli indirizzi IPv4 ed IPv6 è molto semplice ed è basata sulle raccomandazioni W3C,

Gli indirizzi IPv4 devono essere utilizzati con la notazione decimale classica es.:

192.168.43.100

Gli indirizzi IPv6 devono essere utilizzati con la notazione esadecimale separata da : e compresa tra parentesi quadre es.:

[fdd4:3c3f:aaaa::99]

Tutte le definizioni valide di rappresentazione IPv6 sono contemplate.

Impostazione libreria java E-Mail

Alla rilevazione di anomalie OPLON®Monitor ha la possibilità di spedire una e-mail di notifica. Per questo motivo e' necessario eseguire il download delle API all'indirizzo :

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

e spostare la libreria mail.jar in :

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

OPLON®ADC all'avvio verifica l'esistenza della libreria. Se non viene rilevata la libreria l'esecuzione termina con un messaggio di errore.

Impostazione librerie JavaDB

OPLON®ADC storicizza le statistiche in un Database Relazionale. Nell'installazione di default utilizza JavaDB distribuito direttamente nel JDK. Se non diversamente specificato OPLON®ADC utilizza JavaDB in modalità embedded. In ogni caso OPLON®ADC già in fase di installazione richiede le seguenti librerie per poter essere eseguito:

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

Queste librerie vengono distribuite direttamente all'interno del JDK e dopo l'installazione si possono trovare in:

Unix/Linux:
(LBL_JAVA_HOME)/db/lib

Windows:
%LBL_JAVA_HOME%\db\lib

Per rendere visibili le librerie a OPLON®ADC è sufficiente copiare le librerie in (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

OPLON®ADC all'avvio verifica l'esistenza delle librerie. Se non vengono rilevate termina con un messaggio di errore.

ATTENZIONE

Nella distribuzione JDK 64 bit windows potrebbe non essere presente JavaDB. Essendo JavaDB completamente Java è sufficiente prendere le librerie necessarie dalla distribuzione JDK 32 bit.

Start parameters

E' possibile eseguire lo start di OPLON®ADC utilizzando il parametro “-u” (URL) per indicare la posizione dei file parametri. Questo valore viene impostato automaticamante in presenza del parametro confDir nel paragrafo <process>. E' possibile copiare nella posizione di load anche solo alcuni dei file necessari, gli altri verranno caricati dalla zona comune di avvio o all'interno della libreria.

Di seguito alcuni esempi esempi di start (risultato dell'impostazione del parametro confDir nel paragrafo <process>:

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

Questa caratteristica è molto importante in quanto si possono creare diverse release di configurazione in luoghi diversi dando la possibilità di provare nuove configurazioni ed eventualmente tornare alle precedenti in caso di necessità.
Sono stati aggiunti tre paramentri allo start dei processi associati al servizio di management es.:.

-m identifica il servizio univoco es.: -m https://localhost:5900/1_1313t197_A10_LBLGo
-n e' il nome del processo es.: -n A10_LBLGo
-pm è l'URL del processo OPLON®Monitor che lancia il processo es.: -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>
```

Check release and updates

Allo start dei processi Oplon®Secure Access verifica la release nel sito www.tcoproject.com. I dati spediti verso il sito non contengono dati sensibili ma riportano solamente:

rel=99.99.99; license:127123163*****; IP=99999999; RL=9

- rel: E' la release e la versione del prodotto
- license: è la parte distintiva della licenza in utilizzo
- IP: è un digest di controllo
- RL: è il Run Level LBL

Il controllo di release è disattivabile attraverso il parametro “-ncu” allo startup dei processi come evidenziato di seguito nel profilo di lancio. Nel caso venga disattivato o tale messaggio non possa pervenire a www.tcoproject.com TCOGROUP SRL non potrà fornire indicazioni proattive di patch o segnalazioni urgenti relative alla sicurezza.

```
es.:
<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>
```

Per disattivare la verifica della release anche allo start di LBL Monitor impostare il parametro -ncu (NoCheckUpdate), anche allo start iniziale (bacth o service).

es. 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®Secure Access Monitor start definitions

OPLON®ADC è dotato di un sistema di monitoring dei processi di bilanciamento o/e

collegati al check delle risorse. Il “Monitor” quindi è il primo processo a partire ed è colui che lancia il/i programma/i con i servizi specifici.
I parametri di lancio del programma “Monitor” sono quindi:

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

in modalità debug sono:

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

Il parametro -DLBL_INTERACTIVE_CMD=true istruisce il Monitor ad avviarsi con una console interattiva.

Il batch di lancio del programma iniziale (Monitor) è definito in (LBL_HOME) e può essere eseguito in modalità interattiva con line command:

Unix/Linux:
go.sh

Windows:
go.bat

Oppure in modalità batch:

Unix/Linux:
go.sh false

Windows:
go.bat false

Questa funzionalità viene sfruttata durante lo start di OPLON®ADC in modalità Daemon(Unix/Linux) e Services (MSWindows) all'avvio del Sistema Operativo.

Oplon®Secure Access Definizioni opzionali di avvio

E' possibile indicare delle definizioni allo start dei servizi *OPLON*®. Queste definizioni vengono utilizzate dal programma per cambiare alcuni comportamenti come ad esempio il livello di log delle operazioni. Se ne può trovare un esempio nei commenti iniziali in (LBL_HOME)/lib/confMonitor/LBLGo.xml. Nel capitolo ***confMonitor*** verranno spiegati i singoli paragrafi che descrivono lo start di un processo.

```
<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>
```

Tabella definizioni di debug/trace abilitazione/disabilitazione funzionalità di base:

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®LoadBalancer 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

Definition	Description
LBL_DEBUG_MONITOR_HEADER	for http header monitor debugging
LBL_DEBUG_REWRITING	abilita il trace del rewriting durante i test delle condizioni e il caricamento delle variabili
LBL_DEBUG_ROW_REWRITING	abilita il trace del rewriting dei frammenti di stream prima e dopo le modifiche ATTENZIONE: questo flag se abilitato esegue un log molto voluminoso, abilitare in fase di debug delle regole
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	Da utilizzarsi con attivazione delle definizoni [A]
LBL_DEBUG_STATISTIC_CACHE_SIZE	[A] log dell'occupazione delle statistiche nella cache di primo livello
LBL_DEBUG_IPROXY_SESSIONPOOL_SIZE	[A] log dell'occupazione in bytes del 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] occupazione degli oggetti sessioni attive nella tabella sessioni
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	scadenza della sessione applicativa componenti libreria 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 messaggi management service
LBL_PERMIT_UNCOMMITTED_DATA_FORWARDING	Abilita il forwarding di parti di HTTP BODY non completate. Valore di default a false.
LBL_KEEP_ALIVE_EVALUATION	Abilita la valutazione dell'entity Keep-Alive per il mantenimento delle connessioni. Valore di default a false.
LBL_SURVIVOR_TUNNELS_PERC	Se > 0 è la percentuale di tunnel riservati in caso di eccesso di utilizzo delle risorse da parte di un gruppo/dominio/uripath. La sua attivazione avviene con queste discriminanti: a) il numero di tunnel deve essere >=100 b) devono esserci almeno due gruppo/dominio/uripath c) questa definizione deve essere >0
DB_UNDEF	Se true imposta i valori a null con valore “undef” Valore di default a true.
LBL_WHEN_WANT_FORWARD_CLIENT_CERTIFICATES_ANYWAY	se true e c'è una richiesta di un certificato client e il parametro è a want comunque eseguire il forward del certificato, default è false
VXLAN	Default: vxlan, docker, vethe

Configurazione indirizzo IP OPLON®Monitor

Per poter utilizzare la management Console si deve impostare l'indirizzo IP su cui il processo Monitor instaura il listening per le richieste provenienti dalla management console.

I file di configurazione del processo MONITOR di MANAGEMENT sono contenuti in (LBL_HOME)/lib/conf

L'unico file da modificare è il file: (LBL_HOME)/lib/conf/procsProperties.xml nel parametro "LBL_GLOBAL_ADDRESS_MANAGEMENT" che normalmente è impostato a "localhost" come evidenziato.

```
<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>
```

Impostare il valore della variabile con l'indirizzo IP scelto nella rete di management.

Una volta impostato l'indirizzo ip, le impostazioni di queste variabili si eseguono via Management Console

Definizioni connessioni SSL Client

Protocolli SSL

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

Definizioni connessioni SSL (listeners & endpoints)

I listener e gli endpoint possono avere diversi protocolli SSL e diverse configurazioni delle ciphersuite

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

Per eseguire dei test di rete Oplon®Secure Access mette a disposizione una serie di strumenti di base.
L'utilizzo dei test è direttamente utilizzabile con le librerie delle distribuzioni una volta effettuato l'unzip sia lato OPLON® servizi server sia lato OPLON®Management Console.

L'accesso ai test è possibile, da linea di comando, utilizzando sia le librerie della Management Console, che le librerie di una qualsiasi distribuzione di LBLloadbalance.

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®Secure Access server side distributions (OPLON®LoadBalancer, 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
```

I comandi e l'esecuzione sono uguali sia se si sta utilizzando la libreria LBLManagementConsole_lib.jar sia se si sta utilizzando la libreria LBLLoadBalancer.jar.
Per questo motivo di seguito verrà utilizzata a titolo di esempio LBLLoadBalancer.jar.

I comandi disponibili, visualizzabili attraverso l'esecuzione senza parametri, sono i seguenti:
mail (indispensabile mail.jar nel classpath)
multicastReceiver
multicastSender
ping
tcpClient
tcpServer
udpReceiver
udpSender

Ogni comando mette a disposizione il proprio help come nell'esempio di seguito:

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 questo caso con le istanze OPLON®LoadBalancer Standard o Enterprise edition in esecuzione otterremmo un risultato simile a questo:

```
(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
```

Le informazioni visualizzate sono il risultato dello scambio delle informazioni tra due istanze OPLON®LoadBalancer Enterprise. In questo caso, essendo il protocollo multicast, si può notare che vengono “catturati” tutti i messaggi di heart-beat provenienti da diversi nodi con le loro caratteristiche.

Lasciando il programma in esecuzione in multicastReceiver possiamo spostarci in un altro nodo e generare dei messaggi multicast con il tool multicastSender:

```
(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
```

Se il multicast è stato abilitato correttamente nel nodo in ricezione dovremmo ottenere il seguente output:

```
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
```

Come si può notare i messaggi vengono visualizzati assieme agli output provenienti dai nodi OPLON®. Questo non comporta nessun malfunzionamento dei nodi che ricevono i messaggi di test in quanto ogni messaggio contiene un fingerprint che, se non validato, viene scartato.

Un altro esempio di utilizzo può essere un test di performance di rete:

```
(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 un altro nodo è possibile eseguire la funzione client:

```
(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-44LLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLL...
 len read=1029 Buffer to forward=1024|
wk-38LLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLL...
 len read=1029 Buffer to forward=1024|
wk-33LLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLL...
 len read=1029 Buffer to forward=1024|
wk-16LLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLL...
 len read=1029 Buffer to forward=1024|
wk-39LLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLL...
 len read=1029 Buffer to forward=1024|
wk-37LLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLL...
```

Il risultato lato tcpServer è il seguente:

```
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

Il file procProperties a livello di OPLON®Monitor permette di definire dei nomi con un valore da utilizzare nei file di configurazione di tutti i processi gestiti.

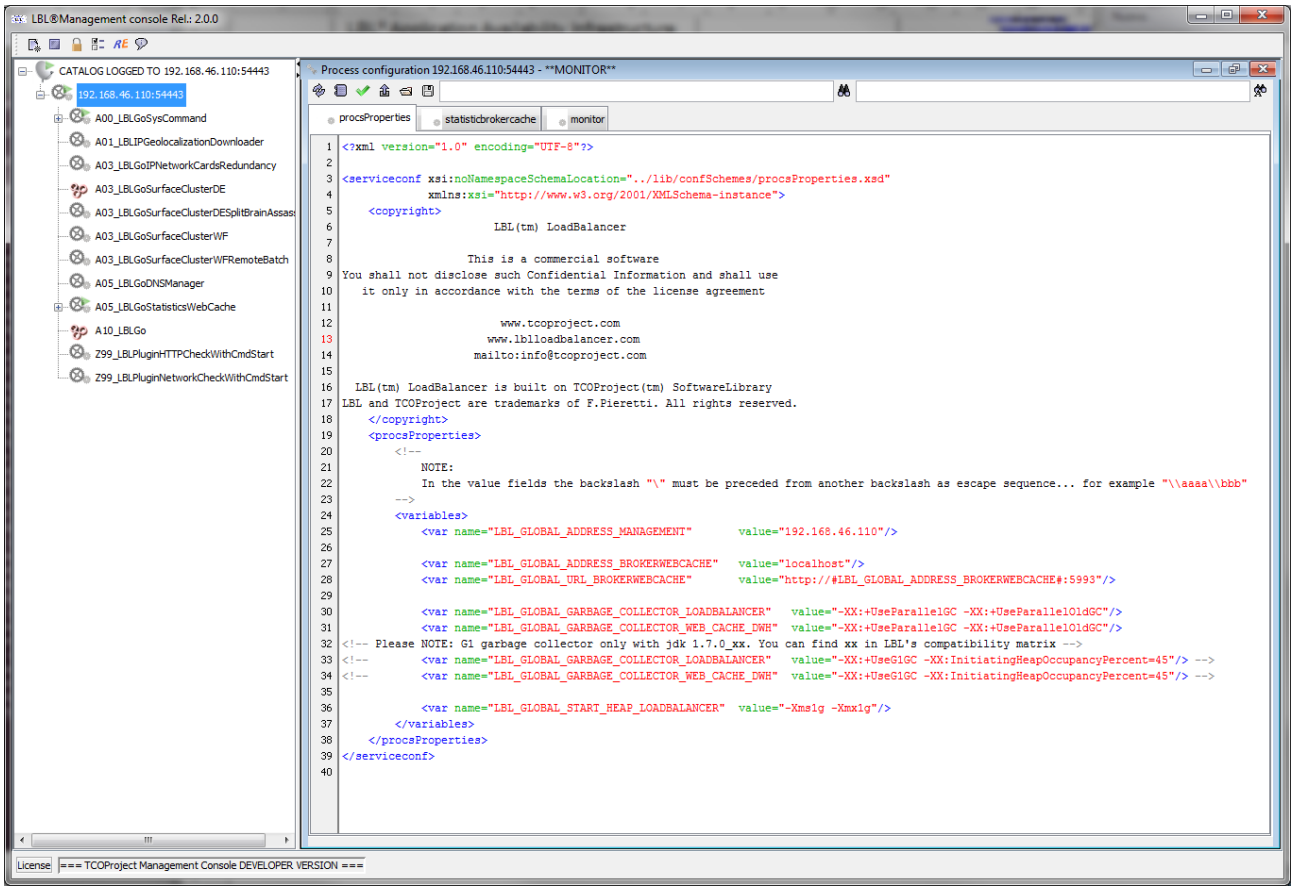


Illustration 1: file procProprties di OPLON®Monitor via management console

La struttura è molto semplice in quanto è sufficiente indicare nel paragrafo <variables> le variabili attribuendo loro un nome ed un valore. E' anche possibile utilizzare, all'interno del valore, variabili precedentemente impostate così da creare nuove variabili composte.
Le variabili, identificate dal paragrafo <var>, possono essere utilizzate in tutti i file di configurazione dei processi. In questo caso, procProperties a livello di OPLON®Monitor, le variabili possono essere utilizzate anche a livello di profilo di lancio del processo (es.: A10_LBLGo).

Consigliamo di dichiarare a questo livello delle variabili con un nome che identifichi la loro posizione, cioè a livello di OPLON®Monitor, utilizzando ad esempio il suffisso “LBL_GLOBAL_...” in modo da non sovrascriverle con le dichiarazioni di variabili a livello di processo, cosa peraltro possibile se si rendesse necessario.

La gerarchia di impostazione è quindi:
1) sequenza impostazione delle variabili all'interno del procProperties OPLON®Monitor
2) sequenza impostazione delle variabili all'interno del procProperties dei singoli processi gestiti da OPLON®Monitor

Il file procProperties non viene propagato negli altri nodi ed è tipico o dell'istanza OPLON®Monitor oppure del singolo processo. “procProperties” è infatti utilizzato per rendere i file di configurazione simili per differenti nodi dove le variabili dichiarate servono a tipizzare la configurazione del singolo nodo.

NOTA: Le variabili utilizzate all'interno dei file di profilo dei processi (es.: A10_LBLGo.xml) verranno utilizzate durante il caricamento delle caratteristiche del processo da parte di OPLON®Monitor. Nel caso si modificasse una variabile a questo livello per farla acquisire nuovamente è necessario rieseguire l'upload del profilo di lancio del processo con conseguente riavvio del processo stesso con le nuove definizioni.

Il file è suddiviso nei seguenti paragrafi:

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

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

name=: valore di default=””
E' il nome della variabile da dichiarare. Il nome non deve contenere #. Se si dichiara un nome variabile già esistente questi verrà sovrascritto in base alla gerarchia di impostazione.

value=: valore di default=””
E' il valore che assume la variabile. Il valore può anche essere composto da variabili precedentemente create es.:

```
<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>
```

Nei file di configurazione è possibile quindi sostituire in tutto o in parte il valore di una proprietà.

es. iproxy.xml:

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

Oplon®Secure Access Process specific procProperties

Per questo paragrafo valgono le stesse considerazioni fatte per il paragrafo a livello procProperties OPLON®Monitor. Si accede a queste variabili attraverso le proprietà del singolo processo.

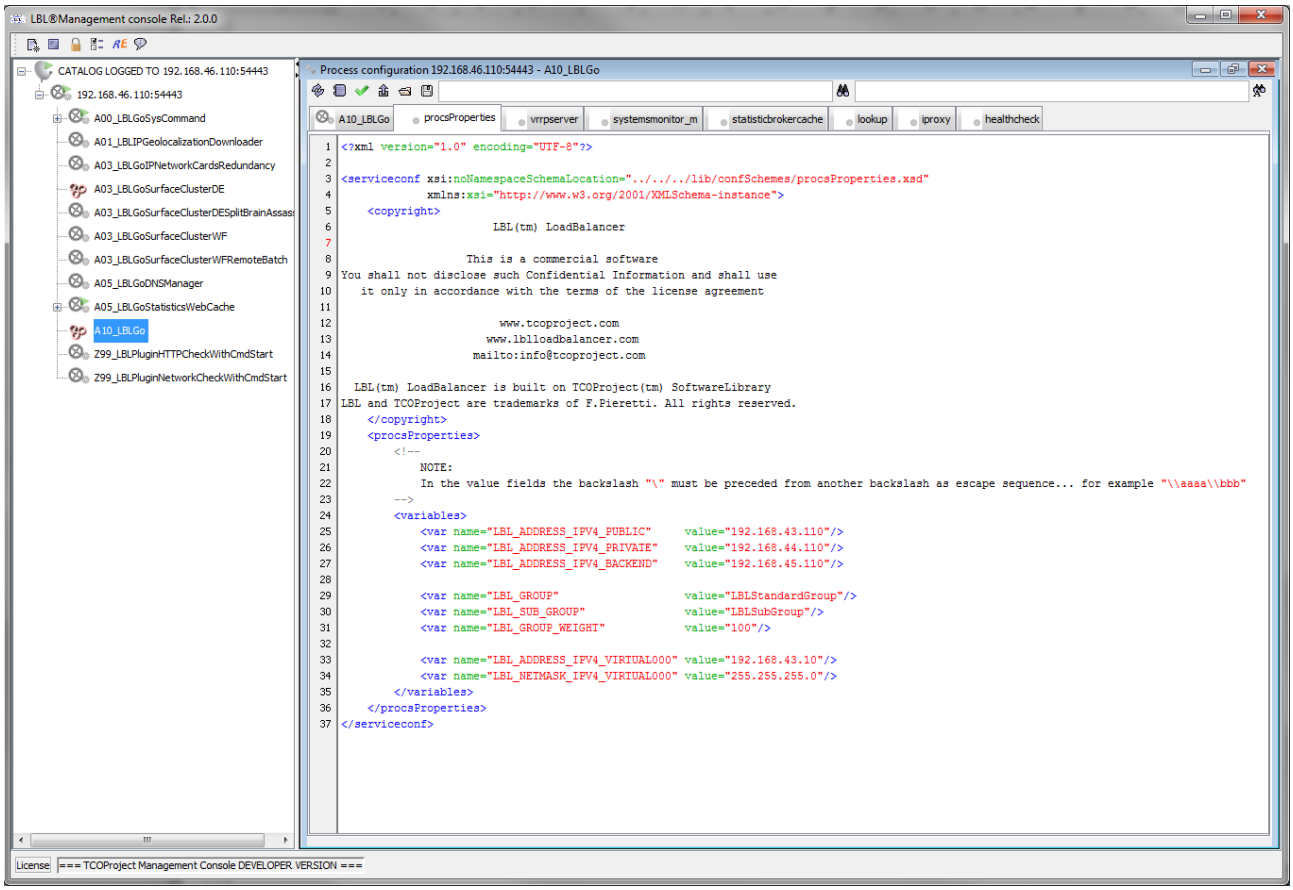


Illustration 2: variabili di processo (A10_LBLGO)

Queste variabili saranno disponibili, a differenza delle variabili dichiarate a livello OPLON®Monitor, per il solo processo a cui appartengono. Attraverso il reinit a caldo dei servizi all'interno dei processi è possibile aggiornare le configurazioni con i nuovi valori.

La struttura del file è identica alla struttura del file procProperties globale OPLON®Monitor.

Oplon®Secure Access System environment variable

There are 2 lbl system environment variable preallocated and useable in procProperties files:

LBL_SYS_LBL_HOME = LBL HOME directory.
Tipically /TCOProject/bin/LBL/LBL_MONITOR/

LBL_SYS_JAVA_HOME = LBL JAVA HOME directory.
Tipically /TCOProject/bin/Java/jdk_1.7_25/

NOTE: the variable ends with /

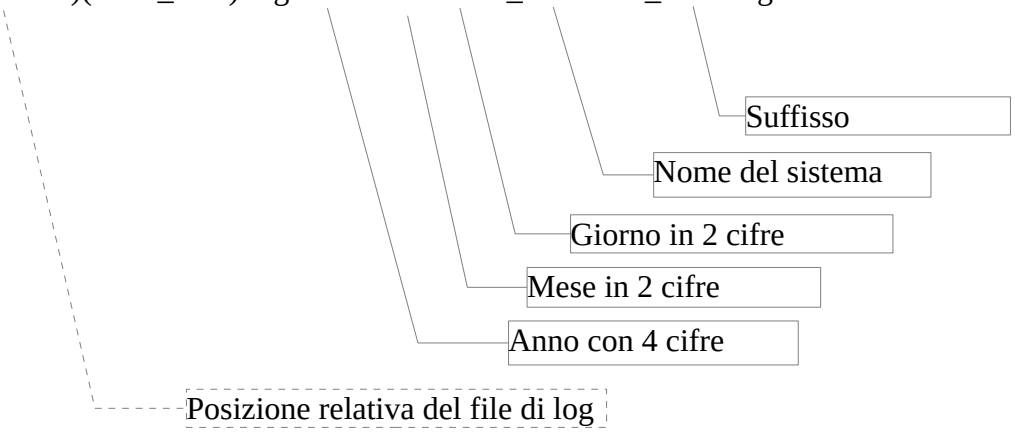
Logfile

OPLON®S.A.A.I. ha un sofisticato sistema di logging che permette agli operatori, o ai sistemi di monitoring, di identificare in maniera semplice le anomalie di funzionamento.
Il log è asincrono e rispetta la sequenza temporale delle operazioni.
La caratteristica di “log-rotation” mantiene la storia degli eventi per un numero predefinito di giorni cancellando i file più vecchi. In questo modo si diminuiscono i tempi di manutenzione del sistema e si rendono stabili le risorse usate nei sistemi di memorizzazione di massa.
Tutti messaggi di log inoltre, sono memorizzati nella tabella database SYSLOG_EVENT ed è quindi possibile centralizzare la gestione degli eventi.

Ogni processo LBL A.A.I. Memorizza il proprio log nella relativa (LBL_HOME)/procsProfiles/XXX_procName/logs.
Il monitor memorizza il proprio log in :(LBL_HOME)/lib/logs.

La struttura del nome dei file di log è così composta:

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



Il file di log è in formato testuale con elementi separati da un pipe (|).
Il tracciato è il seguente:

- Tipologia del messaggio:
  DEBUG=Messaggio di debug
  WARNING=Messaggio di avvertimento, non è un errore
  ERROR=Messaggio di errore, potenziale problema da analizzare specie se persistente
  FATAL=Errore molto grave: stop forzato di un thread oppure shutdown inaspettato dell'istanza
- JVM version
- Nome dell'unità di elaborazione che ha generato il messaggio
- Nome dell'host che ha generato il messaggio
- Data espressa in millisencondi dalla mezzanotte del gennaio 1970 UTC
- Data espressa in forma leggibile
- Messaggio
- Numero di ripetizioni consecutive del medesimo messaggio

NOTA Il “Messaggio” (elemento 6) può contenere dei newline ed è sempre terminato da un pipe(|).
All'interno degli elementi l'escape sequence per proteggere il carattere pipe “|” è il back slash “\”

esempio:

```
|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 possiede un sistema di monitoring dei processi legati alle funzionalità dei singoli processi/servizi. Il sistema di monitoring è il primo programma a partire e gestisce l'intero ciclo di vita degli altri processi. Dal Monitor si possono avviare, fermare e visualizzare i file di log.
Il Monitor al suo start verifica il contenuto della directory
(LBL_HOME)/lib/confMonitor
e per tutti i file con estensione xml ne legge il contenuto. Se il contenuto ha un tracciato di tipo <processconf> lo interpreta e, se il contenuto lo prevede, mette in esecuzione il processo.

Il file è suddiviso nei seguenti paragrafi:

```
<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>
```

Di seguito riportiamo il significato dei paragrafi per poi andare ad approfondire, paragrafo per paragrafo, ogni singolo aspetto di configurazione.

- <**processconf**> E' il paragrafo che identifica una configurazione di processo del Monitor.
- <**copyright**> Paragrafo che contiene i marchi e i diritti legali. Tutti i files, documenti e programmi sono protetti da marchio commerciale e non possono essere rimossi.
- <**properties**> Questo paragrafo identifica il nome univoco del processo all'interno della tabella processi del Monitor. Il suo parametro name deve essere il nome del file senza estensione. es.: <properties name="A10_LBLGo"> nel caso il file si chiamasse A10_LBLGo.xml
- <**process**> E' il paragrafo che contiene i parametri di start del processo. Al suo interno possono essere descritti più starters per quanti sono i sistemi operativi per cui questo processo è stato progettato di funzionare (vedi parametri paragrafi <start>)
- <**warningMessages**> In questo paragrafo possono essere filtrati i messaggi provenienti dallo standard output e dallo standard error del processo in esecuzione. Il Monitor per ogni messaggio indicato ne eseguirà il log con priorità "|WARNING|".
- <**errorMessages**> In questo paragrafo possono essere filtrati i messaggi provenienti dallo standard output e dallo standard error del processo in esecuzione. Il Monitor per ogni messaggio indicato ne eseguirà il log con priorità "|ERROR|".
- <**restartMessages**> In questo paragrafo possono essere filtrati i messaggi provenienti dallo standard output e dallo standard error del processo in esecuzione. Il Monitor per ogni messaggio indicato ne eseguirà il log con priorità "|ERROR|" e provocherà il restart del processo. Questo paragrafo è particolarmente utile per reinizializzare i processi ancora attivi ma che per qualche motivo non sono operativi come ad esempio in concomitanza con "OutOfMemoryError".
- <**alwaysNotifyMessages**> In questo paragrafo possono essere filtrati i messaggi provenienti dallo standard output e dallo standard error del processo in esecuzione. Il Monitor per ogni messaggio indicato in questo paragrafo eseguirà comunque una notifica via e-mail o via post HTTP. In questo paragrafo normalmente vengono filtrati quei messaggi di avvenuto ripristino come ad esempio READY AGAIN.

NOTA E' importante tenere in considerazione che, essendo il nome del processo univoco per ogni istanza di un Monitor, il nome del file xml deve riportare sempre il parametro name del paragrafo XML <properties>.
Se per esempio il file si chiama "LBLFileName.xml" il primo capitolo contenente i parametri di start deve chiamarsi "<properties name="LBLFileName">"

<processconf>

Contiene le configurazioni del processo gestito dal Monitor

<properties>

<processconf>

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

name="Nome Processo"
Il nome del processo deve corrispondere al nome del file xml che lo descrive.

<process>

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

Il paragrafo process contiene i parametri di comportamento generale del processo.

enable=: valore di default="false"
Se true indica che il file descriptor del processo è abilitato allo start automatico

description=: valore di default=""
E' la descrizione del processo.

start=: valore di default="automatic"
Se "automatic" il processo verrà avviato avviato automaticamente allo start del Monitor. Se manual l'avvio è delegato all'operatore.

numberTryStartOnFailure=: valore di default="3"
E' il numero massimo di restart se il processo termina inaspettatamente oppure viene eseguito un restart dal Monitor per messaggio di failure indicato sulla sezione <restartMessages>. Se impostato a -1 il Monitor esegue il restart infinite volte. Questa funzionalità è molto utile per eseguire shellscript di controllo che devono essere rilanciati ciclicamente. Il Monitor riesegue il restart del processo ogni 10" se non indicato diversamente.

waitBeforeKill=: valore di default="5000"
E' il tempo massimo di attesa dopo il comando di shutdown controllato di un processo. Oltrepassato questo limite se il processo non è terminato regolarmente il Monitor esegue un "kill" (Signal 15 su Unix/Linux/BSD). Il restart è regolato dai parametri "numberTryStartOnFailure" e "start".

waitBeforeKillOnFailure=: valore di default="5000"
E' il tempo massimo di attesa dopo un evento di failure. Oltrepassato questo limite se il processo non è terminato il Monitor esegue un "kill" (signal 15 su Unix/Linux/BSD). Il restart è regolato dai parametri "numberTryStartOnFailure" e "start".

managementService=: valore di default="true"
Questo parametro indica se il processo ha a sua disposizione il servizio di management. Normalmente se impostato a true identifica un processo LBL A.A.I. e verrà quindi comunicata all'avvio la porta dinamica di management per la gestione del processo.

confDir=: valore di default=""
E' la posizione della directory del file di configurazione di un processo LBL. Questo valore viene passato all'applicazione tramite il parametro -u (vedi paragrafo Start parameters di questo manuale).

runLevel=: valore di default=”-1”
E' il runLevel di un processo LBL. Se -1 il processo non è di tipo LBL. Questo valore viene sostituito al valore %LBL_EXEC_DEFINES% indicato nel paragrafo <exec>

waitBeforeRestart=: valore di default=”3000”
E' il tempo di attesa prima di un nuovo riavvio (restart) del servizio.Il restart e' regolato dai parametri “numberTryStartOnFailure” e “start”.

sysCommand=: valore di default=”false”
Se true caratterizza il processo come processo di esecuzione comandi di sistema.

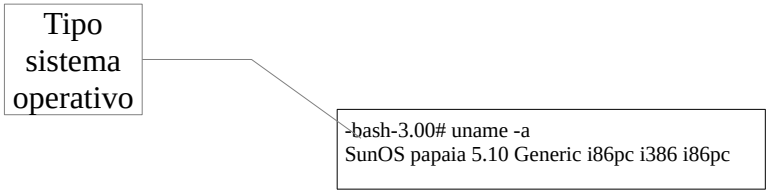
waitBeforeKillExecStop=: valore di default=”10000”
E' il tempo massimo di attesa dopo l'avvio del programma che esegue lo stop del comando eseguito nel paragrafo <exec> . Oltrepassato questo limite se il processo non è terminato regolarmente il Monitor esegue un “kill” (Signal 9 su Unix/Linux/BSD).

sendShutdownCmd=: valore di default=”true”
E' il parametro che abilita la spedizione del comando di shutdown via standard output descritto nel parametro schutdownCmd.

<start>

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

Questo paragrafo identifica per tipo di sistema operativo i parametri del processo da gestire.
<start> ha un unico parametro:
osName=: *valore di default=*””
Identificativo del sistema operativo. Con il comando uname -a è possibile rilevare il tipo di sistema operativo.



Per MS Windows il nome è “Windows”.

osSoftKill=: *valore di default=*”false”
Indica se Java esegue un comando di kill irrevocabile oppure segnala al processo, con oppurtuno signal, l'intenzione di interrompere il processo. Questa indicazione viene valutata dal gestore dei processi durante la fase di stop di un processo in caso di failure e successivo restart. Il valore viene valutato anche al termine di uno stop del processo programmato. Il parametro "osSoftKill" ha una impostazione di default a "false". Alla data attuale i sistemi operativi basati su "Linux" e SunOS (Solaris/OpenSolaris) eseguono un SoftKill (kill -15)

mentre sui sistemi basati su Microsoft Windows eseguono un kill irrevocabile.
=: valore di default=”shutdown”
E' il comando scritto sullo standard input del processo in esecuzione. Questo comando può quindi essere catturato dal programma in esecuzione per comandare uno shutdown graceful (elegante) del processo in esecuzione.

<env>

```
<processconf>
        <properties>
                <process>
                        <start osName=”XX”>
                                <env>
```

valore di default=””
Questo paragrafo può essere ripetuto più di una volta e identifica un parametro di ambiente (variabile ambiente) da impostare o variare rispetto il processo padre (il Monitor).
Ad esempio se si volesse effettuare una variazione o impostazione al CLASSPATH ($CLASSPATH su Unix/Linux, %CLASSPATH% su MS Windows) il suo contenuto dovrebbe essere: “CLASSPATH=lib;lib\LBLLoadBalancer.jar” in ambiente MS Windows e “CLASSPATH=lib:lib/LBLLoadBalancer.jar” in ambienti Unix/Linux.
Il processo in gestione eredita dal Monitor le alte variabili d'ambiente.

Es:

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

<workingDir></workingDir>

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

valore di default=””
Questo paragrafo identifica la directory di lavoro del processo. Se il valore è vuoto la directory di lavoro è quella del Monitor.

<exec></exec>

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

valore di default=””
E' il nome del processo che deve essere gestito dal Monitor.
Esempi:

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

oppure

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

oppure

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

oppure

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

oppure

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

<execStop></execStop>

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

valore di default=""

E' il nome del processo che deve eseguire uno shutdown graceful del processo lanciato nel paragrafo <exec>. Se il questo paragrafo non è stato inserito o è vuoto non eseguirà alcun comando di stop. In tal caso eseguirà un softKill se previsto e alla fine del tempo concesso eseguirà un kill del processo (signal 9 per linux/unix)

es. di comando di shutdown :

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

<warningMessages>

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

In questo paragrafo possono essere filtrati i messaggi provenienti dallo standard output e dallo standard error del processo in esecuzione. Il Monitor per ogni messaggio indicato ne eseguirà il log con priorità "|WARNING|".

<message> </message>

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

message=: valore di default=""

Il valore identifica il contenuto del messaggio da filtrare. Il valore è case-sensitive.

<errorMessages>

In questo paragrafo possono essere filtrati i messaggi provenienti dallo standard output e dallo standard error del processo in esecuzione. Il Monitor per ogni messaggio indicato ne eseguirà il log con priorità "|ERROR|".

<message> </message>

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

message=: valore di default=""
Il valore identifica il contenuto del messaggio da filtrare. Il valore è case-sensitive.

<restartMessages>

In questo paragrafo possono essere filtrati i messaggi provenienti dallo standard output e dallo standard error del processo in esecuzione. Il Monitor per ogni messaggio indicato ne eseguirà il log con priorità "|ERROR|" e provocherà il restart del processo. Questo paragrafo è particolarmente utile per reinizializzare i processi ancora attivi ma che per qualche motivo non sono operativi come ad esempio in concomitanza con "OutOfMemoryError".

<message> </message>

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

message=: valore di default=""
Il valore identifica il contenuto del messaggio da filtrare. Il valore è case-sensitive.

<alwaysNotifyMessages>

In questo paragrafo possono essere filtrati i messaggi provenienti dallo standard output e dallo standard error del processo in esecuzione. Il Monitor per ogni messaggio indicato in questo paragrafo eseguirà comunque una notifica via e-mail o via post HTTP. In questo paragrafo normalmente vengono filtrati quei messaggi di avvenuto ripristino come ad esempio READY AGAIN.

<message> </message>

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

message=: valore di default=""
Il valore identifica il contenuto del messaggio da filtrare. Il valore è case-sensitive.

3 OPLON®Monitor monitor.xml

Questo file di parametri descrive il servizio di monitoring di OPLON®ADC.
Il file è composto dai paragrafi :

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

Mentre il paragrafo <params> descrive i parametri dell'embedded application server il secondo, <notifications> descrive i parametri necessari al monitor per inviare email od eseguire delle post http in caso di eventi anomali o nel caso un operatore volesse notificare al centro di controllo delle note attraveso la WebConsole. I paragrafi <serviceconf>, <copyrigth> e <sysobserver> hanno le stesse funzioni in tutti i file di configurazione e si rimanda quindi a (LBL_HOME)/lib/conf/iproxy.xml per la loro spiegazione.

```
<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>

I parametri che seguono impostano il comportamento del servizio di monitoring

<params>

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

frequency=: valore di default=”10000” UM=Millisecondi
E' la frequenza di verifica cambiamenti di stato nei processi o parametri.

monitorConfDir=: valore di default=”lib/confMonitor”
E' la directory dei file di profilo per il lancio delle applicazioni.

managementPortRangeMin=: valore di default=”5850”
E' il l'inizio del range di porte da assegnare dinamicamente ai processi per il loro management.

managementPortRangeMax=: valore di default=”5990”
E' il la fine del range di porte da assegnare dinamicamente ai processi per il loro management.

address=: valore di default=”localhost”
Bind internal application server web di console
Il valore, impostato di default a localhost, dovrebbe essere impostato per delicatezza del servizio con la rete di management. In mancanza di una rete di management si consiglia di impostarlo con la rete di backend.

addressToConnect=: valore di default=”value in address”
Questo valore, se diverso da “address”, indica l'indirizzo a cui la management console deve riferirsi per poter accedere ai servizi di gestione. Questo è particolarmente utile in quei casi dove l'indirizzo “address” su cui LBL Monitor esegue il bind non sia accessibile direttamente. Se LBL Monitor viene ad esempio installato in AWS Amazon EC2, per poter accedere dall'esterno è necessario impostare address=”0.0.0.0” e addressToConnect con il valore dell'indirizzo pubblico. In questa maniera è possibile dalla propria postazione accedere ai servizi di management.

port=: valore di default=”54443”
E' la porta su cui risponde il servizio.

backlog=: valore di default=”20”
E' il numero massimo di incoming connections livello socket dove il Sistema Operativo permette questa impostazione.

OPLON®ADC ha comunque un suo sistema di gestione delle richieste di connessione nel caso il Sistema operativo non permettesse di sfruttare questa funzionalità.

reuseAddress=: valore di default=”true”
E' il corrispondente parametro socket SO_REUSEADDR.

concurrentWorkers=: valore di default=”20”
E' il numero iniziale di richieste evadibili contemporaneamente dalla WebConsole.

maxConcurrentWorkers=: valore di default=”100”
E' il numero massimo di richieste evadibili contemporaneamente dalla WebConsole.

healthCheckContextPath=: valore di default=”/HealthCheck”
E' il path di healthcheck di attività. Questo valore normalmente non viene mai modificato a meno di uso già presente in altre applicazioni.

webAppsDir=: valore di default=”lib/webroot/webapps”
Directory home applicazione web

webAppsConfDir=: valore di default=”lib/webroot/webappsconf”
Directory configurazione applicazioni web

webSecurityDir=: valore di default=”lib/webroot/websecurity”
Directory configurazione applicazioni web

certificateURL=: valore di default=”/certificate/serverkeys”
Se impostato indica l'indirizzo http da cui prelevare il certificato.

keyStore=: valore di default=”JKS”
Indica il tipo di keystore SSL da cui prelevare il certificato. Normalmente se si utilizza il keystore della JVM deve essere impostato a “JKS” se si utilizza un keystore OpenSSL si deve impostare “PKCS12”.

keyStorePassword=: valore di default=”defaultpwd”
Passoword per l'accesso al keystore.

alias=: valore di default=”lblcert”
E' l'identificativo del certificato all'interno del keystore.

aliasPassword=: valore di default=”defaultpwd”
E' la password per poter accedere al certificato contenuto nel keystore.

keyManagerFactory=: valore di default=”SunX509”
Indica il modulo di interpretazione del certificato. Normalmente impostato a “SunX509”

SSLContextVersion=: valore di default=”SSLv3”
Indica la versione del protocollo SSL. Normalmente impostato a “SSLv3” per JVM keystore oppure “TLS” per OpenSSL.

Di seguito i parametri relativi ai comandi di sistema
<!-- Systems command parameters -->

sysCommandTimeOut=:valore di default=”10000” UM=Millisecondi
Indica il tempo necessario per dichiarare in time-out un comando di sistema. Se il comando oltrepassa questo limite viene eseguito un comando di abort e successivamente viene rilasciato il controllo all'applicazione.

sysCommandCheckRate=:valore di default=”300” UM=Millisecondi
E' la frequenza del controllo dello stato del comando di sistema.

sysCommandRemoteURL=:default=”https://localhost:5992/SysCommand”
E' l'URL del servizio per eseguire i comandi di sistema

Statistic broker web cache parameters →

statisticBrokerWebCacheURL=:default=”http://localhost:5993”
E' l'URL del servizio per richiedere le statistiche

Di seguito i parametri di interscambio informazioni con i servizi →

maxSizeInBuffer=:default=”10485760” UM=Bytes
E' la dimensione massima del buffer in input interscambio informazioni con i servizi.

dateFormat=:default=”dd/MM/yyyy HH:mm:ss:SSSS”
E' il formato della data dei dati statistici provenienti dal Statistic Broker Web Cache e dagli altri servizi

dateFormatDate=:default=”dd/MM/yyyy”
E' il formato della data che esprime i valori giorno mese anno utilizzata per il display

dateFormatTime=:default=”HH:mm:ss”
E' il formato della data che esprime i valori ore minuti secondi utilizzata per il display

delimiter=:default=”|” (pipe)
E' il carattere che delimita i campi durante lo scambio di informazioni.

<surfaceClusterDecisionEngines>

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

Questo paragrafo descrive le caratteristiche di collegamento ai servizi OPLON®Surface Cluster Decision Engine. I parametri servono a stabilire la connessione ai servizi e quindi poter agire centralmente sia per la verifica sia per le azioni da intraprendere.

<instance>

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

Il paragrafo instance descrive i parametri di connessione di una istanza OPLON®Surface Cluster Decision Engine.

enable=:default=”false” UM=boolean
Abilita o disabilita l'interpretazione di questo paragrafo di istanza.

surfaceClusterURL=:default=”https://localhost:54445”
URL per la connessione ai servizi Decision Engine

description=:default=”Surface Cluster Decision Engine”
Descrizione dell'istanza Decision Engine

healthCheckUriPath=:default=”/HealthCheck”
URIPath servizio di HealthCheck

commandUriPath=:default=”/SCDECommand”
URIPath servizio di comando web service

<surfaceClusterWorkFlows>

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

Questo paragrafo descrive le caratteristiche di collegamento ai servizi OPLON®Surface Cluster Work Flow.
I parametri servono a stabilire la connessione ai servizi e quindi poter agire centralmente sia per la verifica sia per le azioni da intraprendere.

<instance>

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

Il paragrafo instance descrive i parametri di connessione di una istanza OPLON®Surface Cluster Work Flow

enable=:default=”false” UM=boolean
Abilita o disabilita l'interpretazione di questo paragrafo di istanza.

surfaceClusterURL=:default=”https://localhost:54444”
URL per la connessione ai servizi Work Flow.

description=:default=”Surface Cluster Work Flow”
Descrizione dell'istanza Work Flow.

healthCheckUriPath=:default=”/HealthCheck”
URIPath servizio di HealthCheck

commandUriPath=:default=”/SCWFCommand”
URIPath servizio di comando web service

<notifications>

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

E' il paragrafo che descrive le modalità di notifica di eventi da OPLON®ADC verso l'esterno. Al momento sono previsti due tipi di notifica: via email e via post HTTP.

<email>

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

enable=: valore di default=”false”
Abilita o disabilita la notifica

interval=: valore di default=”600000” UM=Millisecondi
E' l'intervallo di tempo tra un notifica e l'altra. Superato questo valore se persiste ancora un problema il Monitor esegue la notifica con il primo nuovo messaggio proveniente dal processo in monitoring. Tutti i messaggi di errore o di restart filtrati dai profili di lancio (LBL_HOME)/lib/confMonitor/*.xml possono causare una notifica.

from=: valore di default=”” UM=email address
E' l'email address che identifica il sito o il centro di controllo. Se è sottoscritto il supporto con notifica con TCOProject® verificare le clausole di attivazione del servizio.

fromLogin=: valore di default=”” UM=email address
E' il valore del login del server SMTP. Spesso coincide con il “from”

fromPassword=: valore di default=””
E' il valore della password del server SMTP se richiesta.

to=: valore di default=”” UM=email address/es
Identifica l'indirizzo/i di destinazione della notifica. Se si vuole inserire più di un indirizzo di destinazione è sufficiente spaziare un indirizzo dall'altro con uno o più spazi.

cc=: valore di default=”” UM=email address/es
Identifica l'indirizzo/i di destinazione come copia conoscenza della notifica. Se si vuole inserire più di un indirizzo di destinazione è sufficiente spaziare un indirizzo dall'altro con

uno o più spazi.

bcc=: valore di default="" UM=email address/es
Identifica l'indirizzo/i di destinazione come copia conoscenza nascosta della notifica. Se si vuole inserire più di un indirizzo di destinazione è sufficiente spaziare un indirizzo dall'altro con uno o più spazi.

comment=: valore di default=""
E' un commento libero alfanumerico.

<property>

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

Questo paragrafo, che può essere ripetuto per quanti parametri sono necessari, è l'esatta espressione delle proprietà con cui si può parametrizzare la libreria JavaMail utilizzata da OPLON®ADC e di cui è necessario il download e installazione prima dell'utilizzo come descritto nel manuale di installazione di OPLON®ADC.

name=: valore di default=""
E' il nome della proprietà che si vuole valorizzare.

value=: valore di default=""
E' il valore della proprietà.

OPLON®ADC viene distribuito con alcuni parametri già preimpostati in modo da minimizzare lo sforzo di configurazione e normalmente è sufficiente modificare solo due dei parametri preimpostati.

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

Questa proprietà deve assumere il valore di true se il server SMTP richiede la password per poter effettuare l'invio della email.

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

Questa proprietà normalmente è impostata a smtp come protocollo per poter inviare le email.

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

Questa proprietà identifica il nome del server su cui è attestato il servizio smtp.

<http>

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

<http

Invio notifica via HTTP

enable=: valore di default=”false”
Abilita o disabilita la notificare

interval=: valore di default=”600000” UM=Millisecondi
E' l'intervallo di tempo tra un notifica e l'altra. Superato questo valore se persiste ancora un problema il Monitor esegue la notifica con il primo nuovo messaggio proveniente dal processo in monitoring. Tutti i messaggi di errore o di restart filtrati dai profili di lancio (LBL_HOME)/lib/confMonitor/*.xml possono causare una notifica.

postURL=: valore di default=”” UM=URL
E' l'URL a cui indirizzare le notifiche tramite la funzione POST HTTP.

comment=: valore di default=””
E' un commento libero alfanumerico.

4 OPLON®SysCommand syscommand.xml

Il servizio descritto in questo profilo è deputato ad eseguire i comandi di sistema. Il comando viene criptato per ragioni di sicurezza e inviato a questo servizio per l'esecuzione. Dalla versione 7.0 questo servizio è protetto doppiamente sia nel protocollo applicativo, con calcolo di un digest, sia con Autenticazione e Autorizzazione e trasmissione SSL.

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

Il file si trova in
(LBL_HOME)/procsProfiles/A00_LBLGoSysCommand/conf/syscommand.xml

<syscommand>

Contenitore per la parametrizzazione del servizio.

<params>

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

address=: valore di default=”localhost”
Il valore deve rimanere per motivi di sicurezza impostato su localhost. Solo in casi particolari può assumere valori differenti.

port=: valore di default=”5992”
E' la porta su cui risponde il servizio.

backlog=: valore di default=”20”
E' il numero massimo di incoming connections livello socket dove il Sistema Operativo

permette questa impostazione. OPLON®ADC ha comunque un suo sistema di gestione delle richieste di connessione nel caso il Sistema operativo non permettesse di sfruttare questa funzionalità.

reuseAddress=: valore di default=”true”
E' il corrispondente parametro socket SO_REUSEADDR.

contextPath=: valore di default=”/SysCommand”
E' il context path su cui viene impostato il servizio.

certificateURL=: valore di default=”/certificate/serverkeys”
Se impostato indica l'indirizzo http da cui prelevare il certificato.

keyStore=: valore di default=”JKS”
Indica il tipo di keystore SSL da cui prelevare il certificato. Normalmente se si utilizza il keystore della JVM deve essere impostato a “JKS” se si utilizza un keystore OpenSSL si deve impostare “PKCS12”.

keyStorePassword=: valore di default=”defaultpwd”
Passoword per l'accesso al keystore.

alias=: valore di default=”lblcert”
E' l'identificativo del certificato all'interno del keystore.

aliasPassword=: valore di default=”defaultpwd”
E' la password per poter accedere al certificato contenuto nel keystore.

keyManagerFactory=: valore di default=”SunX509”
Indica il modulo di interpretazione del certificato. Normalmente impostato a “SunX509”

SSLContextVersion=: valore di default=”SSLv3”
Indica la versione del protocollo SSL. Normalmente impostato a “SSLv3” per JVM keystore oppure “TLS” per OpenSSL.

timeOut=: valore di default=”1500” UM=Millesimi di secondo
Il servizio vrrpserver è un servizio HTTP1.0/1.1 e questo valore ne indica il timeout di connessione.

timeOutFactor=: valore di default=”300”
E' il fattore moltiplicativo del time out.

timeOutOpenSocket=: valore di default=”5000” UM=Millisecondi
E' il timeout alla connessione del socket.

timeOutContConnection=: valore di default=”30000” UM=Millisecondi
Questo parametro identifica il timeout durante una connessione HTTP 1.1 tra una lettura consistente di una header HTTP e l'altra. Questa modalita' permette di usufruire dell'ottimizzazione delle connessioni/disconnessioni di HTTP 1.1 e adattarsi all'utilizzo a

volte non del tutto canonico dei client. Il valore -1 disabilita la funzionalità.

tcpNoDelay=: valore di default=”true”
Abilita/Disabilita l'algoritmo di Nagle per controllare la bufferizzazione dei dati

tcpKeepAlive=: valore di default=”true”
Abilita/Disabilita SO_KEEPALIVE nelle connessioni con i client

concurrentWorkers=: valore di default=”10”
E' il numero iniziale di workers per la risoluzione delle richieste di connessione

maxConcurrentWorkers=: valore di default=”20”
E' il numero massimo di workers per la risoluzione delle richieste di connessione

webAppsDir=: valore di default=”lib/webroot_syscommand/webapps”
Directory home applicazione web

webAppsConfDir=: valore di default=”lib/webroot_syscommand/webappsconf”
Directory configurazione applicazioni web

webSecurityDir=: valore di default=”lib/webroot_syscommand/websecurity”
Directory configurazione applicazioni web

5 OPLON®IPGeoLocDownloader iplocalizationdownloader.xml

Il file si trova in:
(LBL_HOME)/procsProfiles/A10_A01_LBLIPGeolocalizationDownloader/conf/
iplocalizationdownloader.xml

Il processo di download del repository aggiornato per la geolocalizzazione degli ip avviene attraverso questo processo. Una volta impostato e avviato aggiorna periodicamente dal sito tcoproject.com le nuove associazioni ip<->località.

La struttura del file è la seguente:

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

<iplocalizationdownloader>

Contiene i parametri per il download del database di geolocalizzazione

<params>

downloadDir=: valore di default=”lib/notificationDir”
E' la directory in cui sarà reso disponibile il file di repository

ipLocalizationFileName=: valore di default=”LBLGeoIPCountry.geo”
E' il nome del file una volta eseguito il download

downloadURL=: valore di
default=”http://www.tcoproject.com/registration/ipgeolocalization/
LBLGeoIPCountry.geo.gz”
E' l'URL dal quale verrà eseguito il download. Questo parametro deve essere modificato nel caso si voglia centralizzare il download a livello di datacenter.

user=: valore di default=”empty”
Utente basic authentication.

password=: valore di default=”empty”
Password basic authentication.

fileMaxSize=: valore di default=”52428800”
E' l'ampiezza massima del file che può essere caricata in memoria. Questo valore è per cautelarsi da caricamenti troppo elevati del repository in memoria.

proxyAddress=: valore di default=”empty”
Se l'accesso al sito tcoproject.com deve transitare da un proxy questo valore deve essere valorizzato con l'indirizzo del proxy.

proxyPort=: valore di default=”0”
Se l'accesso al sito tcoproject.com deve transitare da un proxy questo valore deve essere valorizzato con la porta del proxy.

proxyUser=: valore di default=”empty”
Se l'accesso al sito tcoproject.com deve transitare da un proxy con basic authentication questo valore deve essere valorizzato con l'utente del proxy.

proxyPassword=: valore di default=”empty”
Se l'accesso al sito tcoproject.com deve transitare da un proxy con basic authentication questo valore deve essere valorizzato con la password del proxy.

expireDays=: valore di default=”30” UM=days
E' il tempo di attesa di scaricamento degli aggiornamenti del repository.

keepGzip=: valore di default=”false”
Se true il risultato del download viene nuovamente trasformato in .geo.gz per essere reso disponibile in maniera centralizzata.

trace=: valore di default=”false”
Se true imposta il trace del downloader.

6 OPLON®ADC iproxy.xml

Le impostazioni delle politiche di loadbalancing sono concentrate in un unico file (LBL_HOME)/procsProfiles/A10_LBLGo/conf/iproxy.xml

Il file è suddiviso nei seguenti paragrafi:

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

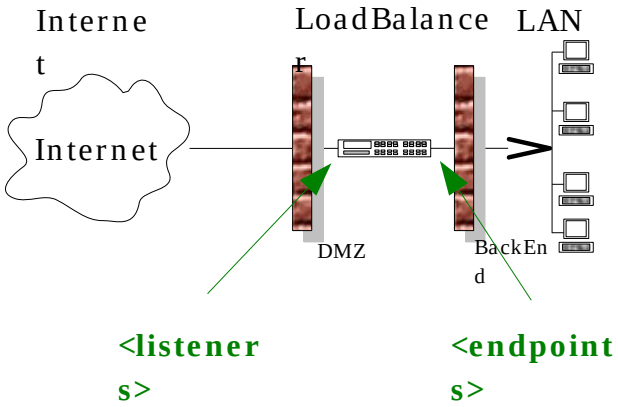
```
            <rewriteManagement>
                <rewriteHeaderRule>
                </rewriteHeaderRule>
                <rewriteBodyRule>
                </rewriteBodyRule>
            </rewriteManagement>
            <endpoints>
               <endPointsGrouping>
                   <virtualDomain>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

Di seguito riportiamo il significato dei paragrafi per poi andare ad approfondire, paragrafo per paragrafo, ogni singolo aspetto di configurazione.

<serviceconf> Essendo OPLON® ADC un sistema basato su servizi questo paragrafo ne delimita le specifiche di configurazione.

<copyright> Paragrafo che contiene i marchi e i diritti legali. Tutti i files, documenti e programmi sono protetti da marchio commerciale e non possono essere rimossi.

<iproxy> E' il paragrafo sul quale è disegnata la strategia di bilanciamento. Per bilanciamento di carico si intende un sistema che riceve in input delle richieste che vengono smistate su più destinazioni come nell'esempio di seguito riportato:



<listeners> Come da disegno questo paragrafo andrà ad elencare su quali indirizzi / interfacce di rete si accettano le connessioni da smistare agli endpoints posti nel backend.

<endPointsGroupingParams> Il port forwarding di diversi protocolli possono rendere necessarie delle impostazioni TCP personalizzate. Questo paragrafo permette di impostare i valori più significativi del protocollo TCP per il trattamento layer 4 (port Forwarding).

<params> Definisce i parametri del bilanciatore come l'ampiezza dei buffer di interscambio dei dati, il numero di thread concorrenti, i time-out e quanto serve a gestire la trasmissione e il forwarding delle informazioni.

<idSessionsManagement> E' il paragrafo dove vengono identificati gli elementi che determinano la “sessione” a livello 7 HTTP1.0/1.1. In questo paragrafo è possibile inoltre indicare a OPLON® ADC di generare e gestire autonomamente la sessione (ADC managed sessions) e di gestirla autonomamente.

<dosAddressesQuarantineList> E' il paragrafo dove vengono identificati gli indirizzi ipv4 ipv6 che non devono essere valutati nella funzionalità DoS Address in Quarantine del modulo DoS Attack Prevention.

<rewriteManagement> Paragrafo dove vengono descritte le regole di rewriting sia della componete HTTP HEADER sia dei contenuti del BODY HTTP durante il forwarding delle informazioni.

<endpoints> Paragrafo che indica le regole di instradamento delle richieste di servizio

<sysobserver> Paragrafo obbligatorio per le versioni OPLON® ADC Standard ed Enterprise Edition descrive le associazioni fisico logiche per comunicare con gli altri servizi. Normalmente non deve essere modificato se non in casi in cui si vogliano virtualizzare più bilanciatori indipendenti all'interno della stessa JVM.

<listeners>

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

In questo paragrafo sono elencati su quali indirizzi / interfacce di rete sono accettate le connessioni da smistare agli endpoints posti nel backend.

<bind>

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

Il sottoparagrafo <bind> specifica il comportamento di ogni singolo listener e può contenere i seguenti parametri:

description=: valore di default=”empty”
Descrizione del listener.

listenType=: valore di default=”NAT"
Può assumere tre diversi valori: STATIC, DAL, NAT.

- **STATIC**: Associazione rigida indirizzo/interfaccia di rete.
  Questa associazione, già presente nella versione 1.1, fornisce una associazione fissa tra listener e indirizzo IP. Nel caso si dovessero verificare modifiche di indirizzo o malfunzionamenti della interfaccia di rete OPLON® ADC cessa di erogare il servizio per quell'indirizzo di rete. Questa modalità è il DEFAULT se non indicato esplicitamente.

- **NAT** Network Adapter Translation.
  Permette ad OPLON® ADC di adattare i listeners sullo stesso indirizzo anche se durante il run-time questo indirizzo è stato spostato da una interfaccia ad un'altra.

  es. di parametrizzazione di un listener NAT:

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

- **DAL** Dynamic Address Listen.
  La funzionalità DAL permette ad OPLON® ADC di specificare una interfaccia di rete, fisica o logica, e la posizione di uno degli indirizzi IP associati. Al modificarsi dell'indirizzo IP OPLON® ADC rileverà la variazione ed effettuerà un nuovo bind con il nuovo indirizzo chiudendo il precedente. Questa funzionalità e' particolarmente utile in quei casi ove l'indirizzo di rete viene riassegnato dinamicamente, DHCP, oppure per creare temporaneamente dei "bridge" su reti ADSL offrendo un unico punto d'ingresso ai servizi del back-end.
  es. di parametrizzazione di un listener DAL:

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

NOTA: non confondere la funzionalità NAT (Network Address Translation) con Network Adapter Translation.
Un gateway (OPLON®LoadBalancer) per sua natura nasconde gli indirizzi degli end-point in gestione.

address=: valore di default=”no default, valore obbligatorio” UM=address list
Indica l'indirizzo TCP/IP sia come nome sia come indirizzo numerico. Se indirizzo numerico indicare:

Per Ipv4 es.: 192.168.43.100
Per Ipv6 es.: [fdd4:3c3f:aaaa::]

E' possibile inoltre indicare più indirizzi IP all'interno di questo parametro. Per ogni indirizzo IP verrà istanziato un listener con la/le porta/e indicate nel parametro port. es.:

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

port=: valore di default=”no default, valore obbligatorio” UM=integer range
Porta sulla quale accettare le connessioni. E' possibile indicare non solo una singola porta ma anche un “range” di porte. La sintassi per poter effettuare un listen su un range di porte è il seguente:

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

Questa espressione indica di istanziare un listener sulla porta 22, i listener dalla porta 70 alla 1000 e dall'8000 al 9000, il listener 30 e 60.

portForwarding=: valore di default=”false”
Questo indicatore serve ad attivare la funzionalità di forwarding simmetrico delle porte associate alla richiesta di servizio. Se attivato il forwarding della richiesta di servizio sarà ribaltato nel backend alla stessa porta della richiesta.

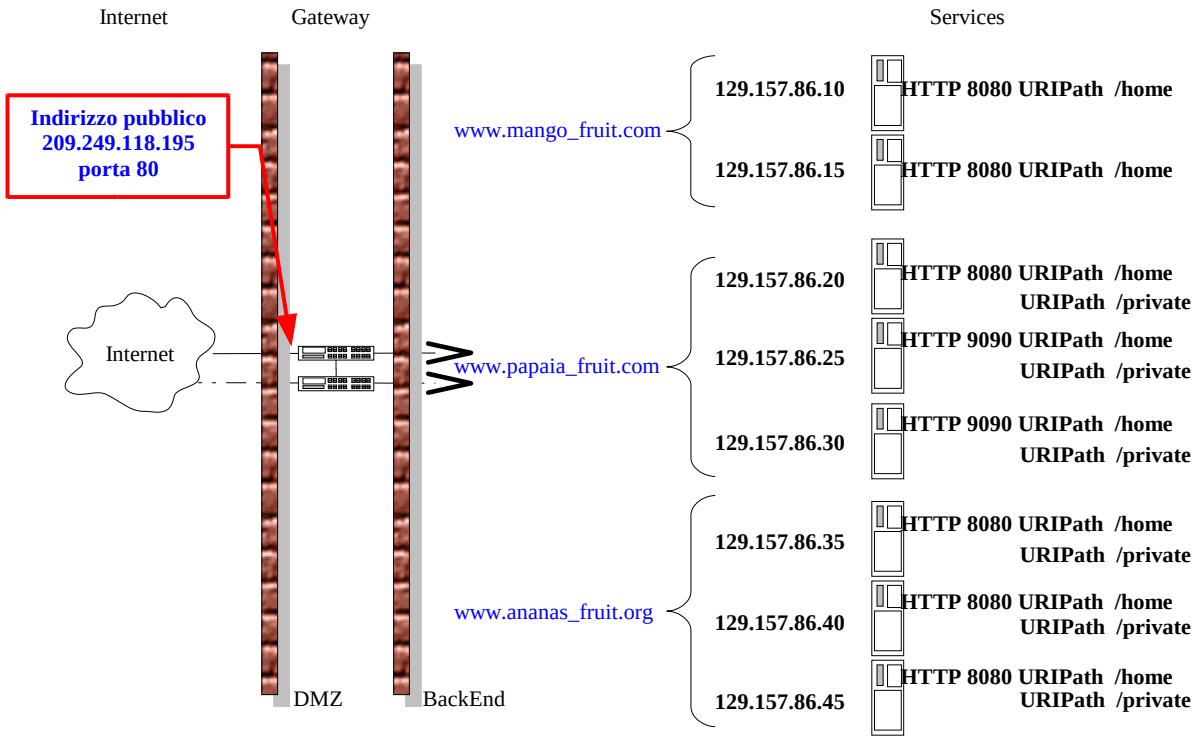
esempi di listeners ed endPointsGroupingParams

```
<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=: valore di default=”false”
Questo parametro è valutato solo dalla versione OPLON®Enterprise Edition. Se “true” il l'address viene propagato ai nodi federati come elemento per effettuare e determinare lo stato del nodo nel GRID.

enableVirtualDomain=: valore di default=”false”
Se “true” il motore di bilanciamento sceglierà tra gli endpoint solamente quelli che avranno una corrispondenza con il dominio (nome host) richiesto dal client. Questo parametro è applicabile sia a livello L7 HTTP/HTTS sia a livello 4 OSI TCP se il protocollo utilizzato è HTTP. A layer 4 HTTPS può essere valutato se la componente ADC funge da terminatore di sessione SSL.

Per fare un esempio possiamo pensare ad un service provider che fornisce servizi a più clienti ognuno con un proprio nome di dominio.



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

Volendo distribuire servizi differenti a questi nomi di dominio utilizzando lo stesso indirizzo porta TCP/IP di accettazione delle richieste è possibile parametrizzare OPLON® ADC attraverso il VirtualDomain, nel modo di seguito descritto.

L' esempio di virtualizzazione di domini può essere così sintetizzato.
L'endPointsGrouoping non è impostato e quindi OPLON® ADC ne assume il default. Gli endPointsGrouping e i virtualDomain possono essere utilizzati contemporaneamente.

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=: valore di default=”true”
Abilita o disabilita il listener. Questo parametro permette di tenere nel file di configurazione il listener con i parametri impostati ma disabilitarlo durante il run-time.

backlog=: valore di default=”2048”
E' il numero di connessioni che il listener può accettare prima di rifiutare la connessione da parte del client.

reuseAddress=: valore di default=”true”
E' il corrispondente parametro socket SO_REUSEADDR.

osiLayer=: valore di default=”7”
Indica la modalità di interpretazione del flusso dati. I valori ammessi sono: 2, 4, 7.
Questo parametro è associato al parametro “protocol” che identifica il tipo di protocollo es.: ftp, http, etc.. OPLON® ADC attualmente interpreta a livello “7” il protocollo HTTP1.0/1.1 e può eseguire il tunneling a livello “4”del TCP e dell'UDP.

protocol=: valore di default=”http”
Indica il protocollo del flusso dati.
Questo parametro è associato al parametro “osiLayer” ed imposta i parametri di connessione tipici del protocollo stesso. Valori predefiniti di protocollo sono: http; ftpCMD; ftpDATA smtp; imap; udp, telnet, ssh, rdp-session-session-affinity, rdp-nosession-affinity, ORAC, DBMS.
E' possibile definire dei protocolli personalizzati, ed associarne i parametri attraverso il paragrafo <endPointsGroupingParams>.
Altri comportamenti possono essere personalizzati attraverso il parametro “endPointGrouping” e il relativo paragrafo come descritto di seguito.

soLinger=: valore di default=”false”
Se true.viene gestito il soLingerTime all'atto della chiusura dei socket.

soLingerTime=: valore di default=”0”
Se impostato a 0 e soLingher è a “true” la chisura del socket propaga un RST TCP.

endPointsGrouping=: valore di default=”default”
Per poter fornire servizi eterogenei in alta affidabilità con una gestione centralizzata OPLON®LoadBalancer mette a disposizione due tipi di aggregazione che permettono la massima flessibilità e partizionamento delle risorse fisico-logiche:
End-Points Grouping e Domains Virtualization.
L' End-Points Grouping e il Domains Virtualization sono le caratteristiche che razionalizzano e aggregano in diverse maniere l'associazione tra Punto/i di Richiesta e Risorse di Servizi.
Con l' End-Points Grouping possiamo assegnare dei nomi simbolici ai listener (Punti di accettazione richieste) e associare ad essi delle risorse di back-end (End-Points)

es.:
LISTENER

```
<bind listenType="NAT"
  address="lblservice" port="5151"
  endPointsGrouping="ftpservers"
  ... altri parametri vedi 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>
```

Aggiungendo all'interno dell'endPointsGrouping ulteriori end-points questi verranno utilizzati solo dai listeners con lo stesso endPointsGrouping.
Aggiungendo altri endPointsGrouping con nomi differenti si possono creare isole di servizi assolutamente separate e utilizzabili solo attraverso i propri listener/s.
L'End-Points Grouping è utilizzabile sia su layer 7 HTTP/S sia su layer 4 OSI (port forwarding). A livello 4 OSI (port forwarding) questa funzionalità viene utilizzata per aggregare servizi dello stesso tipo es.: ftp, smtp etc. .
Su layer 7 HTTP/S l'endPointsGrouping viene utilizzato solamente come raggruppamento, non vengono presi in considerazione i parametri TCP associati utilizzati solo sul layer 4 TCP (port forwarding) o UDP.

transport=: valore di default=”tcp”
Indica il protocollo di trasporto.
Attualmente i protocolli di trasporto possibili sono: tcp; udp.

transportSessionAffinity=: valore di default=”false”
La possibilità di Session Affinity anche a livello di trasporto su layer 4 è possibile

impostando questo parametro a “true”. La sessione sarà distinguibile dall'indirizzo del client che richiede i servizi.

transportSessionAffinityType=: valore di default=”IP” (“IP”, “SESSION”)
A layer 4 TCP e con protocollo HTTP è possibile utilizzare le funzionalità di session management basate su IP oppure su un nome di profilo di management :
Se basate su IP, il default, la sessione viene mantenuta attraverso l'utilizzo dell'IP del client.
Se basate su SESSION viene valutata l'IdSessionsManagerName per velutare l'instradamento alla stessa maniera del layer 7 HTTP

xForwardedFor=: valore di default=”true”
Se impostato a true gestisce l'entity X-Forwarded-For. L'entity X-Forwarded-For e' caraterizzato con un valore comma+space separated ed indica da sinistra verso destra l'indirizzo IP della richiesta fino all'ultimo attraversamento.

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

forwardHeaderCerts=: valore di default=”false”
Se impostato a false esegue la pulizia dei certificati nell'header o comunque esegue la pulizia di certificati non provenienti dal Socket.

h2Bridge=: valore di default=”false”
Se true il listener potrà intaurare una connessione con protocollo HTTP2 e mantenere gli endpoint con protocollo HTTP1.1. Tutte le regole di rewrite e comportamenti di routing applicativo degli endpoint rimangono inalterati. Vedere anche SSLApplicationProtocols.

h2MaxConcurrentStream=: valore di default=”5”
È il numero di connessioni contemporanee parallele massimo (numero di tunnel) che verrà utilizzato per colloquiare con gli endpoint relativi ad una connessione HTTP2.

h2InitialWindowUpdateSize=: valore di default=”3153920”
In alcuni casi il client HTTP2 non esegue il window_update iniziale lasciando l'initial window size di default a 65535 (+1). Serve a impostare il valore di upload iniziale del flow control per permettere l'upload. Nel caso il client incrementi il valore di default con window_update non viene preso in considerazione.

distinguishSingleConnection=: valore di default=”false”
Se impostato a true per ogni connessione TCP inserisce nel database statistico l'informazione LBLCOLOR all'interno del campo COOKIES della tabella L4_TCP_TCPSSL.
Questo valore è univoco rispetto alla connessione e permette di distinguere l'evoluzione della stessa.

tcpInterceptorPrimerCapture=: valore di default=”true”
Al primo innesco della connessione da parte del client (doPrimerFromClient), in dipendenza del protocollo, è possibile escludere la lettura del primo pacchetto proveniente dal client attraverso il parametro tcpInterceptorPrimerCapture=”false”.
Questa funzionalità deve essere disabilitata in tutti i casi in cui si sta eseguendo il rewriting

di protocolli che non prevedono un innesco da parte del client (es.: telnet). Il parametro tcpInterceptorPrimerCapture non ha alcun effetto se non viene esplicitamente utilizzata una classe di rewriting TCP.

tcpInterceptorClassPath=: valore di default=”true”
E' il path di caricamento della classe interceptor definita in tcpInterceptorClass. Questo path viene aggiunto al classpath della JVM in esecuzione.

tcpInterceptorClass=: valore di default=”null”
Indica la classe per intercettare i pacchetti TCP. Questa classe è una estensione della classe : loadbalancer.rewriter.LBLTCPRewriteInterceptorAbstr.

Le distribuzioni contengono già una classe template disponibile in:

(LBL_HOME)/interceptors/rewriteclasses/LBLTCPRewriteInterceptorLogging.java

Le classi contenute in questa directory possono essere compilate attraverso i tool: compile.bat o compile.sh presenti nella directory stessa. Il modo di utilizzo delle classi interceptor template è descritto all'interno delle classi stesse. Questa classe implementa 3 metodi per poter intervenire sia sul flusso sia nell'instradamento.

```
public void doPrimerFromClient(LBLTCPRewriteInterceptorFragment tcpFragment)

public void doPacketFromClient(LBLTCPRewriteInterceptorFragment tcpFragment)

public  void  doPrimerFromEndpoint(LBLTCPRewriteInterceptorFragment  tcpFragment)

public void doPacketFromEndpoint(LBLTCPRewriteInterceptorFragment tcpFragment)
```

Il primo metodo viene richiamato dopo il primo innesco del client che intraprende la richiesta, il secondo metodo viene richiamato ad ogni pacchetto che transita dal client verso l'endpoint, il terzo metodo viene richiamato al primo pacchetto che transita dall'endpoint ed il quarto metodo ad ogni pacchetto che transita dall'endpoint verso il client dopo il primo.

NOTA: I due metodi doPacketFromClient e doPacketFromEndpoint possono essere utilizzati anche contemporaneamente in quanto il flusso è full-duplex.

Al primo innesco (doPrimerFromClient), in dipendenza del protocollo, è possibile escludere la lettura del primo pacchetto proveniente dal client attraverso il parametro tcpInterceptorPrimerCapture=”false”. Questa funzionalità deve essere disabilitata in tutti i casi in cui si sta eseguendo il rewriting di protocolli che non prevedono un innesco da parte del client (es.: telnet). Il parametro tcpInterceptorPrimerCapture non ha alcun effetto se non viene esplicitamente utilizzata una classe di rewriting TCP.

Di seguito un elenco delle funzioni messe a disposizione dalla classe
LBLTCPRewriteInterceptorFragment
Il frammento `tcpFragment` passato nei metodi call-back permette di accede a diverse funzionalità di controllo e modifica di flusso

```
	/**
	* 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/”
E' il path di caricamento della classe interceptor definita in udpInterceptorClass. Questo path viene aggiunto al classpath della JVM in esecuzione.

udpInterceptorClass=: valore di default=”null”
Indica la classe per intercettare il pacchetto dati UDP.
Questa classe è una estensione della classe :
loadbalancer.rewriter.LBLUDPRewriteInterceptorAbstr.

Le distribuzioni contengono già una classe template disponibile in
(LBL_HOME)/interceptors/rewriteclasses/LBLUDPRewriteInterceptorLogging.java

Le classi contenute in questa directory possono essere compilate attraverso i tool: compile.bat o compile.sh presenti nella directory stessa. Il modo di utilizzo delle classi interceptor template è descritto all'interno delle classi stesse. Questa classe implementa 2 metodi per poter intervenire sia sul flusso sia nell'instradamento. La classe deve implementare i seguenti metodi:

```
/**
 * Abstract method called after receive UDP Packet
 * @param streamFragment string fragment before replace
 */
public abstract void doAfterReceivedUDPPacket(LBLUDPRewriteInterceptorFragment udpFragment);


/**
 * Abstract method called after send UDP Packet
 * @param streamFragment string fragment before replace
 */
public abstract void doAfterSendUDPPacket(LBLUDPRewriteInterceptorFragment udpFragment);
```

E' possibile escludere la lettura del pacchetto UDP di ritorno dal servizio modificando il timeout di attesa portandolo a -1. Il valore di default dell'attesa del pacchetto di ritorno del servizio è 5000 millisecondi:

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

L'udpFragment contiene i metodi per l'accesso al pacchetto UDP:

```
        /**
         * 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=””
Numerico che indica la coda nfqueue contenente i pacchetti da e per il frontend.
Il parametro e' usato per il bilanciamento a layer2

layer2QueueOut=: valore di default=””
Numerico che indica la coda nfqueue contentente i pacchetti da e per il backend.
Il parametro e' usato per il bilanciamento a layer2

SSL=: valore di default=”false”

Indica se il listener funzionerà come terminatore di trasmissione cifrata SSL. Se impostato a true sono necessari i parametri sotto riportati.

SSLSNI=: valore di default=”false”
Se il valore è impostato a true il listener accetta connessioni TLS SNI. In questo caso è possibile dichiarare una lista di keystore nel paragrafo <keystoresSNI> che può contenere diversi certificati digitali.

SSLSNIDefaultCertificateEnable=: valore di default=”false”
Se true, nel caso in cui il client non sia abilitato SNI verrà utilizzato il dominio (host name) presente nel parametro SSLSNIDefaultCertificateDomainName.

SSLSNIDefaultCertificateDomainName=: valore di default=””
Se SSLSNIDefaultCertificateEnable impostato a true, il parametro deve contenere il nome host che verrà utilizzato per client non SNI compliant. es.: www.myhostname.com

SSLContextVersion=: valore di default=”SSLv3”
Indica la versione del protocollo SSL. Normalmente impostato a “SSLv3” per JVM keystore oppure “TLS” per OpenSSL.

sslSessionCacheSize=: valore di default=”0” UM=entry
E' il numero di entry nella tabella SSL cache session

sslSessionCacheTimeout=: valore di default=”86400” UM=secondi
E' il timeout della sessione SSL nella cache SSL se non utilizzata.

certificateURL=: valore di default=””
Se impostato indica l'indirizzo http da cui prelevare il certificato.

certificateURIPath=: valore di default=””
Se impostato indica il percorso da cui prelevare il certificato.

keyStore=: valore di default=”JKS”
Indica il tipo di keystore SSL da cui prelevare il certificato.
Normalmente se si utilizza il keystore della JVM deve essere impostato a “JKS” se si utilizza un keystore OpenSSL si deve impostare “PKCS12”.

keyStorePassword=: valore di default=”defaultpwd”
Password per l'accesso al keystore.

keyManagerFactory=: valore di default=”SunX509”
Indica il modulo di interpretazione del certificato. Normalmente impostato a “SunX509”

alias=: valore di default=”lblcert”
E' l'identificativo del certificato all'interno del keystore.

aliasPassword=: valore di default=”defaultpwd”
E' la password per poter accedere al certificato contenuto nel keystore.

needClientAuthentication=: valore di default=”false” UM=lista di valori
Se impostato a “true” e trasmissione SSL viene richiesto al client il suo certificato di identificazione.
Se impostato a “false” non viene richiesto al client alcun certificato.
Se il listener L7 HTTP/S viene impostato a “want” in presenza del certificato questi viene richiesto al client altrimenti continua.
Il valore “want” è normalmente utilizzato con una autenticazione client relativa ad un URIPath specifico.
Per questa esigenza verificare i parametri “needClientCert” e “needClientCertMessage” nei paragrafi: <endPointsGrouping>; <virtualDomain>; <endp>.

trustAllCertificates=: valore di default=”false”
Se questo valore viene impostato a true non viene effettuata la verifica attraverso CA o truststore del certificato. Utile in fase di test o se si vuole effettuare il forwarding di tutti i certificati attraverso reencryption.

checkClientCertificateValidity=: valore di default=”false”
Se questo valore viene impostato a true viene effettuata la verifica sulle date di validità del certificato.

forwardClientCertificateChainDepth=: valore di default=”1”
E' la profondità di forwarding della catena di certificati. Se associato al al parametro “forwardClientPemCertificateToEndpoint” impostare questo valore in maniera consapevole dei 2KB circa occorrenti per ogni certificato che viene trasferito nell'HEADER HTTP verso il servizio.

forwardClientCertificateToEndpoint=: valore di default=”false”
Se questo valore viene impostato a true le informazioni relative al certificato client vengono trasferite verso il servizio.

Le informazioni trasferite al servizio in forma di entity sono:

```
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
```

La parte finale dell'entity identificata da un _ (underscore) indica la profondità della catena di certificati. La profondità della catena di certificati è impostabile attraverso il parametro forwardClientCertificateChainDepth.

forwardClientPemCertificateToEndpoint=: valore di default=”false”
Se impostato a true la catena di certificati provenienti dal client vengono codificati in pem e quindi ricodificati in endoded UTF-8 per essere trasferiti nell'header HTTP verso il servizio.
L'entity nell'HEADER HTTP assume un valore simile a quanto di seguito riportato:

```
x-fwdcertencodedpem_0: -----BEGIN+CERTIFICATE-----
%0D%0AMIIFJzCCBA%2BgAwIBAgIJAMpwGvFCWYDHMA0GCSqGSIb3DQEBBQUAMIG9MQswCQYDVQQ
GEwJJVDEOMAwGA1UECBMFSXRhbHkx%0AFDASBgNVBAcTC0FiYW5vIFRlcm1lM
.... ///// certificato ////
VPNuyd%2FAqN%2BGZz%2BREwdyDPrHQMTt%2BtIVT1bWmmcPfACj0LvuNBt6gyrelICZzX1f7rG
%0AWEy3kZzTgOxAjjChvUaobvy9hGbmb4r4FCT8%0D%0A-----END+CERTIFICATE-----
%0D%0A
```

La dimensione di ogni singolo certificato è di circa 2KB. Il numero di certificati della certificate-chain trasferiti agli endpoint è relativo al parametro “forwardClientCertificateChainDepth”.

trustCertificateURL=: valore di default=””
Se impostato indica l'indirizzo http da cui prelevare il certificato considerato trust.

trustCertificateURIPath=: valore di default=””
Se impostato indica il percorso da cui prelevare il certificato considerato trust.

trustKeyStore=: valore di default=”JKS”
Indica il tipo di keystore SSL trust. Normalmente se si utilizza il keystore della JVM deve essere impostato a “JKS” se si utilizza un keystore OpenSSL si deve impostare “PKCS12”.

trustKeyStorePassword=: valore di default=”defaultpwd”
Password per l'accesso al keystore trust.

trustKeyManagerFactory=: valore di default=”SunX509”
Indica il modulo di interpretazione del certificato. Normalmente impostato a “SunX509”.

netInterface=: valore di default=””
Se “listenType” impostato a “DAL” (Dynamic Address Listen) indica il nome dell'interfaccia di rete in cui eseguire il listen.
Es.: “eth0”

subInterface=: valore di default=”0”
Se “listenType” impostato a “DAL” (Dynamic Address Listen) indica il numero dell'alias dell'interfaccia di rete in cui eseguire il listen.
Es.: “eth0:1” il valore sarà “1”

monitorTimer=: valore di default=”10000” UM=Millisecondi
Se “listenType” impostato a “DAL” (Dynamic Address Listen) indica in Millisecondi il tempo di rilevazione delle modifiche di impostazione dell'interfaccia durante il run-time.

doubleIncomingQueues=: valore di default=”false”
Questo parametro abilita il listener ad utilizzare una coda privata di connessioni in entrata.

Questa funzionalità è stata studiata appositamente per la funzionalità di “network freezing” in quanto permette di “accumulare” le connessioni entranti in una coda e non direttamente sui thread di forwarding evitando effetti “fionda” al ripristino dell'operatività ed eccessivo carico da parte dei servizi di backend.

doubleIncomingQueuesHighWater=: valore di default=”5000”
Questo parametro imposta il limite massimo di richieste di connessione inseribili nella coda prima che il sistema cominci a segnalare un eccessivo riempimento. Superato il doppio di questo valore le connessioni entranti verranno chiuse e non più inserite.

cipherSuites=: valore di default=””
Il parametro imposta le ciphersuites da utilizzare per la comunicazione con il backend.

S**SLApplicationProtocols**=: valore di default=””
In questo parametro si possono elencare i protocolli abilitati su comunicazioni ALPN. I valori ammessi “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=: valore di default=”true”
Se true utilizza i chipersuite nell'ordine indicato per i listener SSL/TLS

SSLProtocols=: valore di default=””
Il parametro imposta i protocolli SSL da utilizzare per la comunicazione.

<keystoresSNI>

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

Quando il parametro SSLSNI è impostato a “true”, la sezione keystoresSNI contiene un elenco di keystore SSL con i certificati digitali associati ai nomi host gestiti nei paragrafi virtualDomain. In questa sezione possono essere inseriti un numero illimitato di keystore per lo stesso listener (indirizzo porta). Ogni keystore può contenere più certificati con lo stesso alias password associati a nomi host (domain) descritti nei CN (common name) dei singoli

certificati. E' possibile avere diversi certificati con diverse password utilizzando semplicemente più keystore.

<keystore>

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

Impostazioni relative al singolo keystore. Un keystore può contenere un numero illimitato di certificati digitali con diversi CN, che devono avere la stessa password. Per utilizzare password differenti è sufficiente aggiungere più keystore.

enable=: valore di default=”true”
Abilita disabilita l'utilizzo del keystore.

description=: valore di default=””

certificateURIPath=: valore di default=””
Indica il percorso del keystore che contiene i certificati. Se in relativo, il percorso è a partire da LBL_HOME

keyStore=: valore di default=”JKS”
Indica il tipo di keystore SSL da cui prelevare il certificato.
Normalmente se si utilizza il keystore della JVM deve essere impostato a “JKS”.

keyStorePassword=: valore di default=”defaultpwd”
Password per l'accesso al keystore.

aliasPassword=: valore di default=””
E' la password del certificato contenuto nel keystore. Tutti i certificati contenuti nello stesso keystore devono avere la stessa password. Certificati con password diverse possono essere gestiti utilizzando più keystore.

keyManagerFactory=: valore di default=”SunX509”
Indica il modulo di interpretazione del certificato. Normalmente impostato a “SunX509”

Esempio:

```
<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
||
```

Esempi <bind>

Alcuni esempi di listeners <bind>

1. hostname
   <bind address="lblservice" port="5050" enable="true"/>

2. network interface
   <bind netInterface="eth0" port="5454" enable="true"/>

3. hostname, terminatore SSL 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"/>
```

4. hostname, terminatore SSL 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"/>
```

5. hostname, terminatore SSL keystore OpenSSL + forward di certificato e 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"/>
```

Se si utilizza un browser per provare questo listener è necessario importare nel browser il certificato già presente nella distribuzione nella directory: (LBL_HOME)/security/certificate/mycert.p12

6. hostname, NAT (Network Adapter Translation)

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

7. hostname, NAT (Network Adapter Translation), Virtual Domain

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

8. 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"/>
```

9. 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"/>
```

10. interface, DAL (Dynamic Address Listen)

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

11. esempio di 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"/>
```

12. esempio di listener UDP

```
<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>
```

Paragrafo per la gestione dei parametri TCP/UDP di un particolare gruppo di servizi, o protocollo.

<endPointsGrouping>

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

endPointsGroupingName=: valore di default=””
E' il nome del gruppo a cui associare i parametri di trasmissione.
Se il nome del gruppo inizia con due pound (##) il riferimento è in relazione con il protocollo e non con il nome del gruppo. Questa funzionalità permette di creare parametri utilizzabili su più gruppo a parità di protocollo utilizzato. E' possibile citare più volte lo stesso nome e in ordine di inserimento verranno applicate per differenza le assegnazioni sui parametri. Questa funzionalità è utile se si vuole modificare ad esempio il comportamento dei protocolli inseriti di default da OPLON®LoadBalancer all'avvio. E' sufficiente quindi impostare il nome del protocollo e indicare solo i parametri che si vogliono modificare. Gli altri parametri rimarranno invariati.

templateName=: valore di default=””
E' il nome del gruppo di parametri preso a template dal quale si vogliono impostare i parametri per differenza sul nuovo gruppo/protocollo.
Da questo gruppo/protocollo vengono presi tutti i valori che non sono presenti sul nuovo gruppo di endpoints. In altre parole se si volessero prendere a riferimento i parametri del protocollo ##http e modificare solo il “clientTimeOut” è sufficiente indicare nell'endPointGrouping:

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

I parametri non citati avranno il valore di default del protocollo “##http”.

clientTimeOut=: valore di default=”1500” UM=Millesimi di secondo
E' il timeOut TCP della connessione verso il client

clientTimeOutFactor=: valore di default=”300”
E' il fattore moltiplicativo del client timeOut

clientTimeOutContConnection=: valore di default=”1500” UM=Millisecondi
Questo parametro identifica il timeout durante una connessione HTTP 1.1 tra una lettura consistente di una header HTTP e l'altra.
Questa modalità permette di usufruire dell'ottimizzazione delle connessioni/disconnessioni di

HTTP 1.1 e adattarsi all'utilizzo a volte non del tutto canonico dei client.
Il valore -1 disabilita la funzionalità.

clientTcpNoDelay=: valore di default=”true”
Abilita/Disabilita l'algoritmo di Nagle per controllare la bufferizzazione dei dati

clientTcpKeepAlive=: valore di default=”true”
Abilita/Disabilita SO_KEEPALIVE nelle connessioni con i client

endPointTimeOut=: valore di default=”1500” UM=Millesimi di secondo
E' il timeOut TCP della connessione verso gli endPoints

endPointTimeOutFactor=: valore di default=”300”
E' il fattore moltiplicativo degli endPoints timeOut

endPointTimeOutContConnection=: valore di default=”-1” UM=Millisecondi
Questo parametro identifica il timeout durante una connessione HTTP 1.1 tra una lettura consistente di una header HTTP e l'altra. Questa modalità permette di usufruire dell'ottimizzazione delle connessioni/disconnessioni di HTTP 1.1 e adattarsi all'utilizzo a volte non del tutto canonico dei servizi. Il valore -1 disabilita la funzionalità.

endPointNumRetryConnection=: valore di default=”10”
Numero di tentativi di connessione prima di dichiarare OutOfOrder un end point

endPointWaitPerRetryConnection=: valore di default=”350” UM=Millisecondi
Tempo di attesa per ogni tentativo (endPointNumRetryConnection) di connessione prima di ritentare la connessione.

endPointCreateConnectionTimeOut=: valore di default=”5000” UM=Millisec.
Time-out del tentativo di connessione all'end-point

endPointTcpNoDelay=: valore di default=”true”
Abilita/Disabilita l'algoritmo di Nagle per controllare la bufferizzazione dei dati

endPointTcpKeepAlive=: valore di default=”true”
Abilita/Disabilita SO_KEEPALIVE nelle connessioni con gli endPoints

sessionTimeOut=: valore di default=”1800000” UM=Millisecondi
Se il load balancer è impostato per gestire le sessioni applicative (stiky-session o loadBalancer managed session) questo valore indica il time-out delle sessioni nella tabella sessioni. Se la sessione in tabella non viene più utilizzata per un periodo di tempo superiore a questo valore vengono cancellati i riferimenti di instradamento della sessione.

sessionRefreshRateFactor=: valore di default=”5” UM=Minuti
Se il load balancer è impostato per gestire le sessioni applicative (stiky-session o loadBalancer managed session) questo valore indica il valore per il calcolo della frazione del tempo di sessionTimeOut dell'algoritmo di touch-session sui nodi.
Il calcolo da effettuare è il seguente:

1800000/5 = 360000 (Refresh Rate in Millisecondi)
360000/1000 = 360 (Refresh Rate in sescondi)
360/60 = 6 (Refresh Rate in minuti)

Quindi se il sessionTimeOut=1800000 l'algoritmo di refresh delle sessioni nelle cache aggiorna le sessioni modificate entro 6 minuti. Più è grande il sessionRefreshRateFactor più piccola sarà la finestra di verifica e aggiornamento delle informazioni di sessione nei nodi. Più è piccolo il valore risultante più aumenta il traffico nella rete privata.

sessionTouching=: valore di default=”true”
L'impostazione della politica di touch a false "sessionTouching=false" può essere utile nei casi in cui si vuole creare una sessione legata all'indirizzo TCP in un periodo di tempo definito e non deve essere rinnovata se avviene uno scambio di dati.
Ad esempio se si vuole utilizzare il protocollo RDP e non associare il server al client ma si vuole mantenere coerenza durante la fase di negoziazione.

sendHTTPResponseOnTimeout=: valore di default=”false” UM=boolean
Se impostato a true al timeout del servizio risponde con i return code 408 e 504.

enable=: valore di default=”true”
Abilita disabilita il paragrafo <endPointsGrouping>.

Esempi

Di seguito Un esempio di <endPointsGroupingParams> e <relativi 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>
```

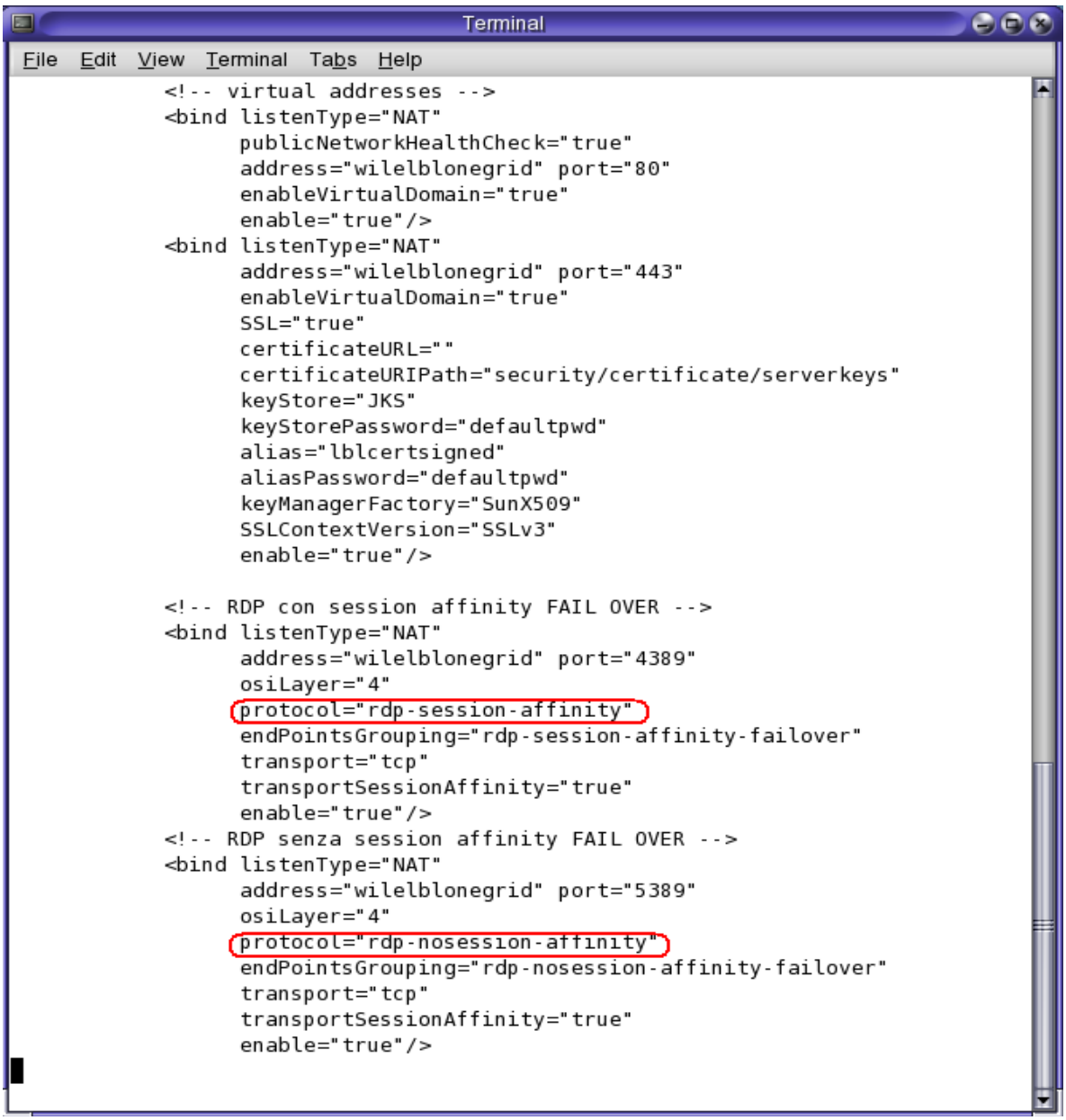
OPLON®LoadBalancer ha già preimpostati i parametri per i più diffusi protocolli:

- HTTP/S
- FTP (command e data)
- RDP senza gestione della sessione
- RDP con gestione della sessione
- Telnet
- SSH
- ORACLE E ORACLE RAC
- Database generic
- UDP generic

Per assegnare uno di questi protocolli ad un listener è sufficiente indicare sul paragrafo <bind> nel valore del campo 'protocol' uno dei valori di seguito riportati in tabella:

Protocollo	Valore da attribuire a <bind protocol=”XXX”)
HTTP/S	http
FTP command	ftpCMD
FTP data	ftpDATA
RDP con gestione della sessione	rdp-session-affinity
RDP senza gestione della sessione	rdp-nosession-affinity
Telnet	telnet
SSH	ssh
ORAC Listener DB Oracle e Oracle RAC	ORAC
DBMS Connessioni a Database	DBMS
UDP generico	udp

Esempio di utilizzo dei parametri di protocollo sul paragrafo <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
```

Questo paragrafo contiene i parametri generali di gestione delle comunicazioni.

NOTA Alcuni dei parametri di questo paragrafo non sono non presi in considerazione nel caso siano presenti paragrafi <endPointsGroupingParams>.
In tal caso sono validi i valori espressi nei rispettivi <endPointsGrouping> in sostituzione dei medesimi del paragrafo <params>.

cipherSuites=: valore di default=””
deprecated: use cipherSuitesListeners instead
Il parametro imposta le ciphersuites da utilizzare per la comunicazione con il backend.

SSLProtocols=: valore di default=””
deprecated: use SSLProtocolsListeners instead
Il parametro imposta i protocolli SSL da utilizzare per la comunicazione con il backend.

cipherSuitesListeners=: valore di default=””
Cipher suites SSL generali utilizzati dai listener

SSLProtocolsListeners=: valore di default=””
Protocolli SSL generali utilizzati dai listener

cipherSuitesEndpoints=: valore di default=””
Cipher suites SSL generali utilizzati dagli endpoint

SSLProtocolsEndpoints=: valore di default=””
Protocolli SSL generali utilizzati dai listener

SSLAlertExpirationDays=: valore di default=”30”
Numero di giorni di anticipo rispetto alla scadenza dei certificati per notificare ogni 24 ore l'imminente scadenza

analyzeHeaderAttackPrevention=: valore di default=”false”
Internal use only

userAgentShortCircuit=: valore di default=””
Se presente, sovrascrive lo user agent di default dell'ADC nelle risposte HTTP che non terminano su un endpoint.

rewriteHeaderRules=: valore di default=””
Regole di rewrite header HTTP applicate su tutte le risorse, è possibile escludere queste regole selettivamente sulle risorse interessate con il modificatore NOME_REGOLA:NOP

rewriteBodyRules=: valore di default=””

Regole di rewrite body HTTP applicate su tutte le risorse, è possibile escludere queste regole selettivamente sulle risorse interessate con il modificatore NOME_REGOLA:NOP

endPointsHealthCheckSetOutOfOrder=: valore di default=”false”
Se true forza l'health check degli endpoint

clientH2Bridge=: valore di default=”false”
Se true abilita il bridge HTTP2 con connessioni client HTTP2 e lato server HTTP

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

clientSSLApplicationProtocols=: valore di default=”” values:”h2 http/1.1 undef”
In questo parametro si possono elencare i protocolli abilitati su comunicazioni ALPN

```
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"/>
```

endPointSSLUseCipherSuitesOrder=: valore di default=”true”
Se true utilizza i chipersuite nell'ordine indicato per gli endpoint SSL/TLS

endPointSSLApplicationProtocols=: valore di default=”” values:”h2 http/1.1 undef”
In questo parametro si possono elencare i protocolli abilitati su comunicazioni ALPN

```
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"/>
```

monitorTimer=: valore di default=”10000” UM=Millisecondi
E' il valore espresso in Millisecondi di monitoring dello stato del sistema di load balancing.

uniqueContextID=: valore di default=”NoVRRPHost”
Valore utilizzato nelle installazioni OPLON®ADC Platform Edition per rendere univoci i valori di traffico registrati nei database.

Il parametro deve essere impostato manualmente nel caso più istanze fossero presenti nello stesso sistema operativo.
Il valore risultante nel database, per distinguere la provenienza, è così composto:
uniqueContextID+local host address+ hash(uniqueContextID+local host address)

es. default: NoVRRPHost-192.168.41.169-1983464450

es. con variazione del valore di default:

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

risultato: 001-192.168.41.169-1983464450

reloadConfHour=: valore di default=”2”
Valore da -1 a 23, se -1 il reload temporizzato e' disattivato, altrimenti se != da -1 all'ora indicata verifica l'esistenza del file reloadConf in
        procsProfile/PROC_NAME/notificationDir/reloadConf
e se esistente, esegue il reload a caldo temporizzato della configurazione cancellando il file

concurrentSessions=: valore di default=”50”
E' i numero di connessioni servibili contemporaneamente iniziali

maxConcurrentSessions=: valore di default=”200”
E' il numero massimo di connessioni servibili contemporaneamente.
Il sistema implementa la funzionalità Survivor Tunnel per preservare risorse in caso un endpoints-group ecceda nell'utilizzo. Il sistema riserva il 10% dei tunnel, con un massimo di 500 tunnel, in queste condizioni:
A) Il numero di tunnel massimi è superiore a 99;
B) Il numero di endpoints-gouping è superiore a 1.

NOTA: Per endpoints-grouping si intende l'insieme tra: gruppo, dominio, uripath.

In un ambiente dove ci sono più endpoints-group, se un servizio supera il 90% delle risorse tunnels, per quello specifico endpoint-group non verranno più accettate le richieste per poter lasciare tunnels liberi per agli altri servizi.

listenersPriority=: valore di default=”5”
Priorità dei listeners (produttori nel gestore di code delle richieste di connessione).

protocolsResolverPriority=: valore di default=”5”

Priorità dei risolutori di protocollo (consumatori nel gestore di code delle richieste di connessione).

expandConcurrentSessionsFactor=: valore di default=”10.0” UM=%
E' il fattore di crescita con cui vengono istanziate nuove sessioni di connessione fino al raggiungimento del “maxConcurrentSessions”.

survivorTunnelsPercent=: valore di default=”30.0” UM=%
E' la percentuale soglia di tunnel occupati dallo stesso servizio “gruppo/dominio/uripath”. Al superamento di questa soglia tutti gli indirizzi IP all'interno dei tunnel che appartengono allo stesso “gruppo/dominio/uripath” vengono messi in quarantena. Associato al parametro “dosAttackPreventionWashingMachine“ permette di liberare risorse per gli altri servizi anche con attacchi DoS/DDoS violenti. **Valori minori o uguali a 0 disattivano la funzionalità**.

La WasingMachine è abilitata a gestire l'indirizzo solo se **isDosAttackPreventionOnlyClose** è true

Sia la WashingMachine che il SurvirvistTunnelsPercent si attivano in presenza della licenza DoS/DDoS e se dosAttackPreventionDetection è impostato a false.

Condizioni di attivazione della washing machine:
dosAttackPreventionWashingMachine = dosAttackPrevention=true &&
!dosAttackPreventionDetection=false

Condizioni di attivazione survivalTunnelsPercent:
survivalTunnelsPercent = dosAttackPrevention=true &&
!dosAttackPreventionDetection &&
dosAttackPreventionWashingMachine=true

highWaterWarningLevel=: valore di default=”10.0” UM=%
E' la percentuale soglia di connessioni in attesa di essere prese in carico da un esecutore del pool di forwarding. Se viene superata questa soglia il sistema avvisa con un messaggio specifico (yellow-alert) l'avvenuto superamento del limite. Se il limite viene superato vengono istanziate inoltre nuove sessioni di forwarding fino al raggiungimento del “maxConcurrentSessions”.

highWaterDangerLevel=: valore di default=”70.0” UM=%
E' la percentuale soglia di connessioni in attesa di essere prese in carico da una sessione di forwarding. Se viene superata questa soglia il sistema avvisa con un messaggio specifico (red-alert) l'avvenuto superamento del limite. Se il limite viene superato vengono istanziate altre nuove sessioni di forwarding fino al raggiungimento del “maxConcurrentSessions”. Con OPLON®DoS Attack Prevention function se viene raggiunto questo limite il sistema esegue le procedure DDoS Congestion Resolver©.

dosAttackPrevention=: valore di default=”false” boolean
Se impostato a true vengono applicate le regole di prevenzione di un attacco informatico di tipo DoS (Denial of Service). dosAttackPrevention interviene in modalità diverse a seconda

del tipo di attacco:
1- Richieste multiple provenienti dallo stesso indirizzo IP
2- Richieste multiple provenienti da diversi indirzzi IP

Nel primo caso al superamento di un numero definito di richieste contemporanee, impostabile con il parametro “dosMaxTunnelForClientAddress”, queste vengono cancellate comprese le eventuali richieste pendenti nelle code interne.
Nel secondo caso al superamento della soglia highWaterDangerLevel tutte le richieste in corso e/pendenti nelle code vengono cancellate.

In entrambi i casi prima della cancellazione delle richieste pendenti se connessione in HTTP/S viene spedita una pagina di cortesia al client che ne ha fatto richiesta.

NOTA L'attivazione di questa funzionalità è subordinata alla presenza della licenza specifica di DoS Attack Prevention.

dosAttackPreventionDetection=: valore di default=”false” boolean
Se true quando avviene il detect di un attacco DoS/DDoS viene segnalato ma non bloccato. È molto utile all'attivazione della funzionalità per determinare gli indirizzi da mettere in whitelist.

dosAttackPreventionOnlyClose=: valore di default=”true”
Alla rilevazione di un attacco DoS l'azione di default è chiudere i canali che si sono identificati come minaccia. Per alcuni protocolli è possibile ritornare l'indicazione della temporanea impossibilità di raggiungere il servizio con una pagina di cortesia. Questa indicazione potrebbe però essere utilizzata per perfezionare l'attacco e quindi è disabilitata per default.

dosAttackPreventionWashingMachine=: valore di default=”false” boolean
Se impostato a true tutti gli indirizzi in quarantena vengono inoltrati in una coda separata dagli indirizzi non in quarantena e viene eseguito un reset TCP (RST).

La WasingMachine è abilitata a gestire l'indirizzo solo se **isDosAttackPreventionOnlyClose** è true

Sia la WashingMachine che il SurvirvistTunnelsPercent si attivano in presenza della licenza DoS/DDoS e se dosAttackPreventionDetection è impostato a false.

Condizioni di attivazione della washing machine:
dosAttackPreventionWashingMachine = dosAttackPrevention=true &&
!dosAttackPreventionDetection=false

Condizioni di attivazione survivalTunnelsPercent:
survivalTunnelsPercent = dosAttackPrevention=true &&
!dosAttackPreventionDetection &&
dosAttackPreventionWashingMachine=true

dosDDoSPutEndPointsOutOfOrder=: valore di default=”false” boolean

During a red alert event, if true DDoS attack prevention places the endpoints to Outoforder.

dosMaxTunnelForClientAddress=: valore di default=”100” UM=richieste nel tunnel
Indica la soglia come numero di richieste servite contemporaneamente per singolo indirizzo IP di provenienza. Se questa soglia viene superata tutte le connessioni instaurate provenienti dallo stesso IP verranno istantaneamente cancellate. Se HTTP/S, a tutte le connessioni in attesa nella coda di richieste verrà invece risposto con il messaggio di cortesia (dosCMessage) oppure ridirezionate in base al parametro (dosRedirect).

dosMaxTunnelForClientSubnet_255_255_255_0=: valore di default=”255” UM=richieste nel tunnel
Indica la soglia come numero di richieste servite contemporaneamente per la stessa subnet (C) di provenienza. Se questa soglia viene superata tutte le connessioni instaurate provenienti dalla stessa subnet verranno istantaneamente cancellate. Se HTTP/S, a tutte le connessioni in attesa nella coda di richieste verrà invece risposto con il messaggio di cortesia (dosCMessage) oppure ridirezionate in base al parametro (dosRedirect).

dosMaxTunnelForClientSubnet_255_255_0_0=: valore di default=”500” UM=richieste nel tunnel
Indica la soglia come numero di richieste servite contemporaneamente per la stessa subnet (B) di provenienza. Se questa soglia viene superata tutte le connessioni instaurate provenienti dalla stessa subnet verranno istantaneamente cancellate. Se HTTP/S, a tutte le connessioni in attesa nella coda di richieste verrà invece risposto con il messaggio di cortesia (dosCMessage) oppure ridirezionate in base al parametro (dosRedirect).

dosMaxTunnelForClientSubnet_255_0_0_0=: valore di default=”2000” UM=richieste nel tunnel
Indica la soglia come numero di richieste servite contemporaneamente per la stessa subnet (A) di provenienza. Se questa soglia viene superata tutte le connessioni instaurate provenienti dalla stessa subnet verranno istantaneamente cancellate. Se HTTP/S, a tutte le connessioni in attesa nella coda di richieste verrà invece risposto con il messaggio di cortesia (dosCMessage) oppure ridirezionate in base al parametro (dosRedirect).

dosAddressQuarantineTime=: valore di default=”1800000” UM=milliseconds
All'individuazione di un attacco proveniente da un singolo indirizzo ip questo viene automaticamente posto in quarantena impedendo l'accesso ai servizi. Il tempo di quarantena è determinato da questo valore (30'). Superato questo tempo è data nuovamente la possibilità al client di accedere ai servizi. Se questo valore è impostato a 0 o < di 0 la funzionalità di quarantena viene disabilitata.

dosCMessage=: valore di default=”messageServicesOverload.html”
Indica la pagina di cortesia da esporre in caso di attivazione del DoS Attack Prevention. Se Se si vuole modificare la pagina di cortesia è sufficiente modificare il file: (LBL_HOME)/resources/html/messageServicesOverload.html oppure posizionarlo per singolo processo come ad esempio :(LBL_HOME)/procsProfiles/A10_LBLGoStandardHA/resources/html. Una volta posizionato il file è possibile modificarlo anche durante il runtime. Questo parametro viene preso in considerazione se dosAttackPreventionOnlyClose è impostato a false.

dosCaptchaCMessage=: valore di default=”messageServicesCaptchaOverload.html”
Indica la pagina di cortesia da esporre in caso di attivazione del DoS Attack Prevention con conferma del Captcha Input per verificare l'utilizzo da parte di una persona e non da un automa. Se si vuole modificare la pagina di cortesia è sufficiente modificare il file: (LBL_HOME)/resources/html/messageServicesCaptchaOverload.html oppure posizionarlo per singolo processo come ad esempio :(LBL_HOME)/procsProfiles/A10_LBLGoStandardHA/resources/html. Una volta posizionato il file è possibile modificarlo anche durante il runtime. Questo parametro viene preso in considerazione se dosAttackPreventionOnlyClose è impostato a false.

dosCaptchaGoCMessage=: valore di default=”messageServicesCaptchaGoOverload.html”
NOTA: QUESTO VALORE È OBSOLETO, NON UTILIZZATO IN QUESTA RELEASE
Indica la pagina di cortesia da esporre dopo aver impostato il captcha corretto in caso di attivazione del DoS Attack Prevention. Se si vuole modificare la pagina di cortesia è sufficiente modificare il file: (LBL_HOME)/resources/html/ messageServicesCaptchaGoOverload.html oppure posizionarlo per singolo processo come ad esempio:(LBL_HOME)/procsProfiles/A10_LBLGoStandardHA/resources/html. Una volta posizionato il file è possibile modificarlo anche durante il runtime. Questo parametro viene preso in considerazione se dosAttackPreventionOnlyClose è impostato a false.

dosRedirect=: valore di default=””
Se valorizzato indica l'URI a cui ridirigere la richiesta in caso di attivazione del DoS Attack Prevention. es.: http://www.caugthinfo.com/
Se questo valore viene impostato ha priorità rispetto al parametro dosCMessage. Questo parametro viene preso in considerazione se dosAttackPreventionOnlyClose è impostato a false.

rewriteHeaderRules=: valore di default=””
Elenco dei nomi delle regole di rewriting dell'HEADER HTTP (layer 7 HTTP/S) da applicare a tutti gli endpoint.
E' possibile indicare più regole separate da uno o più spazi.
es.:

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

Le regole verranno applicate, se le condizioni lo permettono, in sequenza.

;**LAST**
Per ogni nome regola è possibile inoltre indicare il parametro LAST che nel caso la regola venga eseguita determina lo stop dell'applicazione delle restanti regole della sequenza.
Per indicare il parametro è sufficiente mettere dopo il nome il “;” e il nome del parametro.

es.

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

Tra il nome e il/i parametri non devono esserci spazi. In questo caso se la regola redirSSLloginWhenNoSSL viene applicata la regola proxyTo non verrà mai eseguita.

;**ALWAYS**
Il parametro ALWAYS indica che la regola viene sempre eseguita
Per indicare il parametro è sufficiente mettere dopo il nome il “;” e il nome del parametro.

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

In questo caso la regola proxyTo viene eseguita a prescindere dall'esecuzione della regola redirSSLloginWhenNoSSL.

;**NOP**
Il parametro NOP indica che la regola non deve essere eseguita.

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

In questo caso la regola redirSSLloginWhenNoSSL non viene eseguita. Il parametro NOP è utile per escludere l'esecuzione di regole generali.

;FINAL
;ALWAYS-FINAL
;LAST_FINAL
;NOP-FINAL
Le operazioni con nome o estensione FINAL hanno la stessa funzionalità di quelle già esistenti con la differenza che vengono eseguite alla fine. In pratica, anche se descritte a livello globale ADC, GROUP DOMAIN o ENDPOINT vengono eseguiti, nell'ordine indicato, alla fine di tutte le altre regole descritte a livelli sottostanti.

Sequenza impostazione regole:
ADC (globale)
GROUP
DOMAIN
ENDPOINT

Un esempio di concatenazione di regole:
AAAA**;ALWAYS-FINAL** BBBB CCCC DDDD**;FINAL** EEEE

L'ordine di esecuzione sarà:
BBBB CCCC EEEE AAAA;ALWAYS DDDD

rewriteBodyRules=: valore di default=””
Elenco dei nomi delle regole di rewriting del BODY HTTP (layer 7 HTTP/S) da applicare a tutti gli endpoint. E' possibile indicare più regole separate da uno o più spazi.
es.:
```
rewriteBodyRules="addTrademarkParam absoluteToRelative echoRewriteBody"
```

Le regole verranno applicate, se le condizioni lo permettono, in sequenza.

;**LAST**

Per ogni nome regola è possibile inoltre indicare il parametro LAST che nel caso la regola venga eseguita determina lo stop dell'applicazione delle restanti regole della sequenza
Per indicare il parametro è sufficiente mettere dopo il nome il “;” e il nome del parametro.
es.

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

Tra il nome e il/i parametri non devono esserci spazi. In questo caso se la regola addTrademarkParam viene applicata le regole absoluteToRelative echoRewriteBody non verranno mai eseguite.

;ALWAYS
Il parametro ALWAYS indica che la regola viene sempre eseguita.

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

In questo caso la regola echoRewriteBody viene eseguita a prescindere dall'esecuzione della regola addTrademarkParam.

;**NOP**
Il parametro NOP indica che la regola non deve essere eseguita.

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

;FINAL
;ALWAYS-FINAL
;LAST_FINAL
;NOP-FINAL
Le operazioni con nome o estensione FINAL hanno la stessa funzionalità di quelle già esistenti con la differenza che vengono eseguite alla fine. In pratica, anche se descritte a livello globale ADC, GROUP DOMAIN o ENDPOINT vengono eseguiti, nell'ordine indicato, alla fine di tutte le altre regole descritte a livelli sottostanti.

Sequenza impostazione regole:
ADC (globale)
GROUP
DOMAIN
ENDPOINT

Un esempio di concatenazione di regole:
AAAA**;ALWAYS-FINAL** BBBB CCCC DDDD**;FINAL** EEEE

L'ordine di esecuzione sarà:
BBBB CCCC EEEE AAAA;ALWAYS DDDD

clientDefaultTimeOut=: valore di default=”1500” UM=Millesimi di secondo
E' il timeOut TCP della connessione verso il client

clientDefaultTimeOutFactor=: valore di default=”300”
E' il fattore moltiplicativo del client timeOut

clientDefaultTcpNoDelay=: valore di default=”true”

Abilita/Disabilita l'algoritmo di Nagle per controllare la bufferizzazione dei dati

clientReceiveBufferSize=: valore di default=”-1”
E' l'ampiezza del buffer del socket in ricezione del client

clientSendBufferSize=: valore di default=”-1”
E' l'ampiezza del buffer del socket in spedizione al client

endPointDefaultTimeOut=: valore di default=”1500” UM=Millesimi di secondo
E' il timeOut TCP della connessione verso gli endPoints

endPointDefaultTimeOutFactor=: valore di default=”300”
E' il fattore moltiplicativo degli endPoints timeOut

endPointDefaultTcpNoDelay=: valore di default=”true”
Abilita/Disabilita l'algoritmo di Nagle per controllare la bufferizzazione dei dati

endPointReceiveBufferSize=: valore di default=”-1”
E' l'ampiezza del buffer del socket in ricezione dall'endPoint

endPointSendBufferSize=: valore di default=”-1”
E' l'ampiezza del buffer del socket in spedizione all'endPoint

healthCheckContextPath=: valore di default=”/LBLHealthCheck”
E' il path di healthcheck di attività del sistema di bilanciamento. Questo valore normalmente non viene mai modificato a meno di uso già presente in altre applicazioni. Se questo valore viene modificato è necessario modificarlo anche in “systemsmonitor_m.xml”, “iproxy.xml” e in “healthcheck.xml”.

defaultCType=: valore di default=”text/html”
Valore inserito nell'header proveniente dall'endPoint nel caso sia stata abilitata la correzione del mime type (content type)

clientCorrectionCType=: valore di default=”false”
Abilita la correzione del mime type (content type) nell'header proveniente dall'endPoint in spedizione verso client.

clientCType=: valore di default=”Content-Type: +defaultCType”
Valore da attribuire all'header proveniente dall'endPoint in spedizione verso il client nel caso di correzione di mime type (Content type)

endPointCorrectionCType=: valore di default=”false”
Abilita la correzione del mime type (content type) nell'header del client in spediione verso l'endpoint.

maxPureStreamLength=: valore di default=”99999999” UM=bytes
E' la lunghezza massima di uno stream del body HTTP in assenza di dichiarazione esplicita della lunghezza. Questa comportamento, tipico delle prime implementazioni HTTP 1.0, è

spesso associato alla non valutazione dell'entity HTTP “connection: keep-alive”. A questo scopo la valutazione di questo parametro anche in OPLON®ADC è impostata a “false”di default (vedi parametro keepAliveEvaluationHTTP10).

keepAliveEvaluationHTTP10=: valore di default=”false”
Questo parametro abilita e disabilita la valutazione dell'entity HTTP “connection: keep-alive”. Nelle prime implementazioni HTTP 1.0 questo parametro è stato a lungo non valutato e ancora oggi la maggior parte degli application server HTTP 1.0 o proxy server non valutano questo parametro. A questo scopo anche OPLON®ADC si adegua a questo comportamento impostando di default a false la valutazione di questo parametro.

endPointCType=: valore di default=”Content-Type:+defaultCType”
Valore da attribuire all'header del client in spedizione verso l'endpoint nel caso di correzione di mime type (Content type)

monitorTimerStatSessions=: valore di default=”120000” UM=Millisecondi
E' il valore espresso in Millisecondi e determina la granularità temporale delle informazioni storicizzate nel database.

monitorTimerActivePoolQueueConsumer=: valore di default=”100” UM=mill
Millesimi di secondo di attesa per ogni ciclo di monitoraggio del sistema iProxy active pool queue consumer

monitorTimerActiveRoutingSessions=: valore di default=”8000” UM=mill
Millesimi di secondo di attesa per ogni ciclo di monitoraggio del sistema iProxy active routing sessions

fileNameForceIncomingConnectionToWait=:
valore di default=”forceIncomingConnectionToWait” UM=mill
Nome file notifica wait incoming connections per freeze connessioni.

monitorIncludeAddressStatSessions=: valore di default=”false”
Valore che indica al motore statistico di includere l'informazione del client address nelle informazioni di sessione. Il valore impostato a true in alcune circostanze con numero elevato di utenti può portare ad un elevato numero di record nella tabella SESSION_ACTIVITY.

notificationDir=: valore di default=”lib/notificationDir”
Indica la directory dove notificare a OPLON®ADC un outOfOrder di un servizio o un gruppo di servizi di backend. E' possibile comandare dall'esterno un out-of-order di un end-point o un raggruppamento di end-point. A tal proposito nel file iproxy.xml e' possibile assegnare dei nomi associativi a gruppi/domini fino al singolo end-point.
Se vengono creati dei file nella nuova directory "notificationDir" con prefisso “outOfOrder” seguiti dal nome simbolico ("outOfOrder.nomeSimbolico") l'end-point o il dominio oppure il gruppo di end-point si metteranno in stato di out-of-order fino alla cancellazione del file.
Nella zona (LBL_HOME)/lib/plugin si possono trovare 2 sorgenti di esempio di health check e disabilitazione/abilitazione di servizi di backend comandati dall'esterno.

udpWaitTimeSessionInitialization=: valore di default=”3000”

E' il tempo massimo di attesa per i datagrammi senza associazione di sessione. Il valore e' impostato di default a 3000 millisecondi (3 secondi).

monitorTimerStatUdpSessions=: valore di default=”10000” UM=millisecondi
Millesimi di secondo di attesa per ogni ciclo di monitoraggio della lista sessioni di sincronizzazione UDP.

cMessage=: valore di default=”messageNoEndPoint.html”
Nome del file in formato html contenente il messaggio di cortesia per il contesto al quale è associato. Al momento del caricamento dei parametri, allo start del processo di bilanciamento, viene verificata l'effettiva esistenza del file in sequenza nelle seguenti directory:

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

Nel caso il file non sia esistente viene segnalato al momento dell'avvio e assegnato il valore di default:

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

emptyGroupDomainEval=: default value=”false” UM=Boolean
È ora possibile avere un Gruppo/Dominio senza endpoint o con tutti gli endpoint impostati su enable=“false”. In questo caso è possibile utilizzare cMessage personalizzati e regole di riscrittura (senza redirect/shortcircuite/displaceto). Questa opzione è attiva solo se “virtualDomain=”true“” è presente nell'elenco.

Se si desidera attivare questa opzione, è necessario impostare un parametro in “iproxy base params”, “endpointsGrouping” o “virtualDomain”:

emptyGroupDomainEval="true"

Quando si imposta emptyGroupDomainEval =true, il paragrafo minimo per impostare un gruppo/dominio senza endpoint è il seguente:

```
<endPointsGrouping groupName="GrpName" enable="true"
rewriteHeaderRules="MyRule, MyRule1">
        <virtualDomain enable="true">
                <endp  enable="false"/>
        </virtualDomain>
</endPointsGrouping>
```

A proposito di redirect, shortcircuit e dispalce a un altro gruppo, se l'endpoint che istruisce un redirect o uno shortcircuit è disabilitato dal runtime (disable/outOfOrder/Drain) restituisce la pagina di errore 404. Questo è logico perché se un operatore disabilita (disable/outOfOrder/Drain) si potrebbe pensare che non voglia più eseguire il reindirizzamento o il cortocircuito.

Di seguito è riportata una tabella riassuntiva che descrive quando le regole vengono eseguite rispetto allo stato degli endpoint e alla loro funzione.

L7 HTTP RewriteRule exec	Service	Redirect	shortcircuit	Displace to group
endPoint enable="false"	REQUEST	no execution	no execution	no execution
disable/outOfOrder/Drain	REQUEST	no execution	no execution	no execution

L2/L3/L4 TCP/UDP RewriteRule exec	Service	Redirect	shortcircuit	Displace to group
endPoint enable="false"	no execution	no execution	no execution	no execution
disable/outOfOrder/Drain	no execution	no execution	no execution	no execution

initialDim=: valore di default=”68000” UM=Byte
Dimensione iniziale della cache della cache per singolo tunnel.

redimFactor=: valore di default=”1”
E' il fattore adattativo per portare la cache fino alla massima dimensione espressa dal parametro “maxDim”

flushFactor=: valore di default=”65535”
Fattore di flushing del traffico bufferizzato.

maxDim=: valore di default=”68000”
Dimensione massima della cache per singolo tunnel.

I quattro valori initialDim, redimFactor, flushFactor, maxDim servono a calcolare la dimensione della cache di IO e ottimizzare ed accelerare le operazioni di forwarding.

Il dimensionamento della cache è importante perché determina l'utilizzo della banda e quindi il throughput.
Il valori di default sono impostati per ottenere il minimo dispendio di risorse.

OPLON®ADC inizializza per ogni thread 4 buffer: 2 per lo streaming proveniente dal client verso i servizi (endpoint); 2 per i valori di ritorno. I buffer utilizzano la tecnologia COW (Copy On Write) per le modifiche dei valori passanti.

InitialDim determina la dimensione iniziale dei buffer, maxDim determina la dimensione massima del buffer. Normalmente lo streaming viene bufferizzato e, al raggiungimento della soglia di flushing, flushFactor, lo stream viene inoltrato.
Questo meccanismo non è valido nelle fasi di lettura delle HEADER HTTP.
Le HEADER HTTP, infatti, per necessità di routing devono venire lette e bufferizzate per intero per essere analizzate e interpretate.
Questo significa che le HEADER HTTP non possono essere più ampie, in byte, del parametro maxDim. Nel caso in cui il client o l'endpoint utilizzassero delle HEADER più

corpose, è possibile agire sul maxDim tenendo presente che, la lettura dell'HEADER non avviene per blocchi, e quindi un'applicazione che ne facesse un uso pesante , in termini di volume, impiegherebbe inevitabilmente più tempo di altre applicazioni analoghe con HEADER meno voluminose.
Nel caso in cui un servizio o un client oltrepassassero questo valore durante il runtime OPLON®ADC reagirebbe interrompendo la connessione e generando un messaggio di errore con “Buffer grows to much!” con l'indicazione di dove è avvenuto lo sbordamento.

Nei casi in cui non è possibile applicare il flushing, ad esempio nella lettura delle HEADER HTTP, quando il valore initialDim raggiunge il suo limite viene ampliato attraverso il parametro redimFactor. Questo parametro amplia il buffer di IO per blocchi fino al raggiungimento del maxDim. Superato anche il maxDim viene chiusa la connessione e generato l'errore di cui sopra.

Dagli elementi sopra descritti ne deriva che il fattore di flushing, flushFactor, per essere efficace deve essere minore dell'initialDim.

Dai test effettuati sui diversi sistemi operativi i valori che garantiscono un ottimo rapporto di throughput/occupazione di risorse durante la produzione sono:

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

In questo caso si ottiene anche un considerevole miglioramento nella scalabilità e dell'affidabilità nel tempo facendo lavorare OPLON®ADC a capacità finite e quindi predeterminate all'avvio del servizio diminuendo le possibilità di out of memory durante il runtime per effetto dell'ampliamento dei buffer dinamici.

A parità di allocazione di buffers alzando quindi il flushFactor si otterrà un miglioramento nel throghput e una diminuzione dell'utilizzo delle CPU. Abbassando il flushFactor, fino anche a 1 byte alla volta, si diminuirà l'utilizzo della banda e si alzerà l'utilizzo della CPU che sarà impegnata a elaborare blocchi di informazioni con dimensioni minori.

Prendendo a riferimento un maxDim di 68000 il calcolo della memoria massima, occupata dai buffers di IO è il seguente:

- maxConcurrentSessions * maxDim * 4 = MAX MEMORY FOR IO BUFFERS

es:

- 5000 threads * 68000 * 4 = 1.360,00 MB (1,3GB)

Con una JVM dimensionata a 2GB rimangono circa 500 MB disponibili per mantenere le tabelle di routing. Circa 800.000 sessioni stiky.

udpMaxDim=: valore di default=”65535” UM=Byte
Dimensione dimensione massima che può assumere il datagramma.
Questo parametro è importante per il dimensionamento della memoria. Solo se viene utilizzato un listener di tipo “udp” i buffer vengono allocati.

Se esiste almeno un listener di tipo udp vengono preallocati:

- maxConcurrentSessions*2*udpMaxDim bytes.

Per 2000 Thread il calcolo risulta quindi essere il seguente:

- 2000*2*65535=262.140.000 (250 MB)

File outOfOrder

Attraverso il file outOfOrder è possibile dichiarare fuori servizio un gruppo di endpoint/servizi.

```
<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"/>
```

Come si può notare nella sezione <endPointsGrouping> sono stati assegnati 2 nomi simbolici "pippo" e "pluto", nella sezione <virtualDomain> sono stati assegnati altri due nomi simbolici "paperino" e "minni" e solo negli <endp> con uriPath="/TCOProject" sono stati associati i nomi "qui", "quo" e "qua".

Se durante l'esecuzione verrà creato un file nella nuova directory:

(LBL_HOME)/lib/notificationDir/outOfOrder.quo

solo i due end-point con uriPath="/TCOProject" verranno posti in out of order

Se durante l'esecuzione viene creato un file nella nuova directory:

(LBL_HOME)/lib/notificationDir/outOfOrder.minni

tutti gli end-point con quel dominio verranno posti in out of order

Se durante l'esecuzione viene creato un file nella nuova directory:

(LBL_HOME)/lib/notificationDir/outOfOrder.pippo

tutti gli end-point con quel gruppo verranno posti in out of order

Questo metodo permette ai sistemisti di utilizzare qualsiasi strumento per effettuare i check dei servizi e solo creando/cancellando un file abilitare o disabilitare interi gruppi di servizi di backend.

<idSessionsManagement>

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

Questo paragrafo identifica e parametrizza la gestione della sessione applicativa.

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

<idSessions>

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

E' possibile creare diversi contesti di determinazione della sessione e quindi associarli, attraverso il nome, ad un gruppo di servizi, dominio fino ad un gruppo di URIPath in dipendenza dei servizi da gestire.
Esso è composto da diversi <id> che identificano le entità determinanti la sessione

enable=: valore di default=”true”
Abilita o disabilita questo gruppo di parametri

name=: valore di default=”default”
E' il nome dello specifico gruppo identificatore di sessione.
Attraverso questo parametro si identificano i diversi contesti di determinazione della sessione associabili a diversi gruppi di servizi.
Il valore automatico se inesistente è “default” che viene associato a tutti i servizi se non indicato puntualmente.

lblSessionGeneration=: valore di default=”true”
Abilita disabilita la creazione e gestione della sessione da parte di OPLON® ADC (Loadbalancer managed session).

caseSensitive=: valore di default=”true”
Se case sensitive i tag di sessione, nomi di identificatore di sessione, vengono valutati considerando diversamente lettere maiuscole e minuscole.
Se impostato a “false” non vengono valutate differenze tra lettere minuscole e maiuscole.
Ad esempio JSESSIONID è, nel caso caseSensitive=”false”, uguale a jSessionId.

lblTagSession=: valore di default=”LBLSESSIONID”

Nome del cookie di sessione se lblSessionGeneration=true

httpOnly=: valore di default=”false”
Imposta l'attributo httpOnly nel cookie di sessione se lblSessionGeneration=true

secure=: valore di default=”false”
Imposta l'attributo secure nel cookie di sessione se lblSessionGeneration=true

maxDelaySessionPropagation=: valore di default=”3000” UM=Millisecondi
Parametro utilizzato nella versione Enterprise di OPLON® ADC. Indica l'attesa massima di un nodo del cluster paritetico prima di dichiarare non trovata una sessione.

<id>

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

Nella gestione della session applicativa identificano le entità determinanti la sessione.

context=: valore di default=””
contiene il valore significativo (entity) nell'header HTTP che identifica la sessione.
Se si vuole creare una regola per la prima stringa HTTP deve essere valorizzato a "FIRST-LINE":
es.
<id context="FIRST-LINE" tag="JSESSIONID" path="firts_line" enable="true"/>

tag=: valore di default=””
E' il nome del valore da ricercare es.: JSESSIONID. Il valore è case-sensitive cioè “JSESSIONID” è diverso da “jsessionid” e ancora diverso da “JsessionId”.

caseSensitive=: valore di default=”<idSessions caseSensitive”
Se case sensitive i tag di sessione, nomi di identificatore di sessione, vengono valutati considerando diversamente lettere maiuscole e minuscole. Se impostato a “false” non vengono valutate differenze tra lettere minuscole e maiuscole.
Ad esempio, nel caso caseSensitive=”false”, JSESSIONID è uguale a jSessionId.
Il valore viene ereditato dal paragrafo <idSessions> ma può essere impostato per differenza.

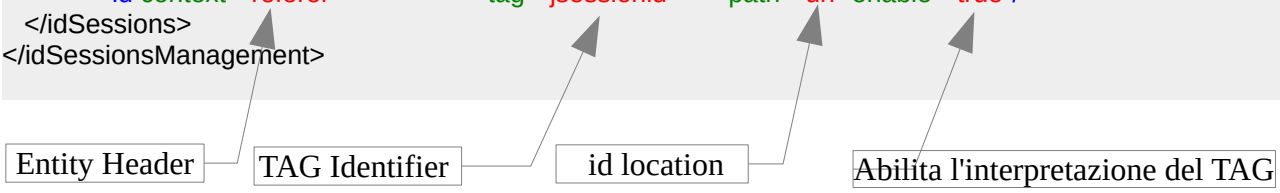
path=: valore di default=””
puo' assumere i seguenti valori: “first_line”, “header_fields”, “url”.

- header_fields
  ricerca il valore con l'espressione nome=valore
- first_line
  ricerca il valore in una uri /aaa/bbb?nome=valore
- url
  ricerca il valore in una url http://www.aaa. .../ccc?nome=valore

enable=: valore di default=”true”
Abilita o disabilita un elemento di identificazione di sessione.

Esempio <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>
```



Nel caso venga abilitata la funzionalità di generazione e gestione delle sessioni da parte di OPLON® ADC altri eventuali <id> presenti nel paragrafo <idSessions> non verranno valutati e la gestione delle sessioni sarà completamente demandata a OPLON® ADC.

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

Per facilitare le operazioni di impostazione sono stati creati due profili di management delle sessioni.
Il primo non gestisce alcuna sessione; il secondo invece gestisce la sessione generata da OPLON® con i parametri di default.
Ovviamente e' possibile creare nuovi profili con nomi diversi e con ulteriori nuovi parametri.
I due profili automaticamente generati sono:

- "nosessions" che equivale a:
  <idSessions name="nosessions">
  </idSessions>
- "autosessions" che equivale a:
  <idSessions name="autosessions" lblSessionGeneration="true">
  </idSessions>

Nelle distribuzioni OPLON® Standard HA ed OPLON® Enterprise HA il profilo “nosessions” è stato associato all'endpoint /HealthCheck che deve solo determinare lo stato di attività senza nessun controllo o generazione di sessione.

<dosAddressesQuarantineList>

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

Questo paragrafo identifica le regole per escludere indirizzi dalla funzionalità DoS Attack Prevention e nello specifico la funzione DoS Address In Quarantine.

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

<address>

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

Il paragrafo indica, attraverso espressioni regolari, quali sono gli indirizzi da escludere dalla gestione della quarantena. Nel caso eccedano nell'utilizzo delle risorse sono comunque segnalati con dei messaggi di warning.

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

Il paragrafo sopra descritto può anche essere utilizzato nella seguente maniera essendo il valore una espressione regolare.

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

<cacheControl>

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

Questo paragrafo identifica le regole di inserimento dell'entity “cache-control” in base al contenuto “content-type” alla risposta del server di backend.
Questa sezione serve a descrivere per raggruppamenti di “content-type” gli entity da inserire nell'HEADER di risposta del server.
Il cache control, identificato il “content-type” nell'HEADER di risposta dell'application server, elimina gli eventuali “entity” preesistenti in base alla lista degli entity da inserire, e quindi inserisce tutti gli “entity” presenti nel paragrafo.
Questa funzionalità permette ai proxy ed ai client di sfruttare, qualora vi aderiscano, la specifica “rfc2616” per controllare i sistemi di cache in response.
Riferimento rfc2616: Cache-control Mechanisms, Cache-Control ed in particolare cache-response-directive.

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

<cacheControlId>

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

name=: valore di default=””
E' il nome identificativo del gruppo di content-type per i quali devono essere inseriti gli entity nell'Header di riposta dal servizio di backend verso il client che ne ha fatto richiesta.

enable=: valore di default=”true”
Abilita o disabilita questo gruppo content-type.

<contentType>

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

value=: valore di default=””
E' il valore per cui un match detemina un inserimento dell'Header con l'entity indicato nel paragrafo <entity>. I contenType possono essere anche ripetuti.

<entity>

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

name=: valore di default=””
E' nome dell'entity che deve essere inserito. Di norma è “cache-control”.

value=: valore di default=””
E' il valore per che verrà attribuito all'entity.
es.:
“max-age=300, must-revalidate

Esempio <cacheControl>

Segue un esempio di parametrizzazione del paragrafo cacheControl

```
...
    <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"/>
     ...
```

Di seguito il risultato nell'HEADER di risposta prima e dopo la modifica

RESPONSE HEADER PRIMA DELL'INSERZIONE DELL'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 DOPO L'INSERZIONE DELL'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

NOTA I browser più recenti eseguono questo controllo in automatico verificando con gli entity "etag" e "last-modify" l'ultima modifica effettuata nell'oggetto presente sul server. Se l'oggetto non è stato modificato viene rispedita la riposta con valore di ritorno 304 (Not modified) e il browser quindi preleva dalla propria cache l'oggetto senza rileggerlo nuovamente dalla rete.

<sslCertificatesManagement>

```
<serviceconf>
    <iproxy>
        <sslCertificatesManagement>
```

Questo paragrafo e' il contenitore dei certificati digiltali client utilizzati dall'ADC per le connessioni ai server di backend che usano la SSL client authentication.

```
<sslCertificatesManagement>
    <SSLCerts
        name=”... enable=”...>
    </SSLCerts>
    <SSLCerts>
        ...
        ...
</sslCertificatesManagement>
```

<SSLCerts>

```
<serviceconf>
    <iproxy>
        <sslCertificatesManagement>
            <SSLCerts
```

name=: valore di default=””
E' il nome identificativo delle caratteristiche della gestione certificati nel backend.

enable=: valore di default=”true”
Abilita o disabilita questo paragrafo.

SSLContextVersion=: valore di default=”SSLv3”
Indica la versione del protocollo SSL. Normalmente impostato a “SSLv3” per JVM keystore oppure “TLS” per OpenSSL.

certificateURL=: valore di default=””
Se impostato indica l'indirizzo http da cui prelevare il certificato client.

certificateURIPath=: valore di default=””
Se impostato indica il percorso da cui prelevare il certificato client.

keyStore=: valore di default=”JKS”
Indica il tipo di keystore SSL da cui prelevare il certificato client. Normalmente se si utilizza il keystore della JVM deve essere impostato a “JKS” se si utilizza un keystore OpenSSL si deve impostare “PKCS12”.

keyStorePassword=: valore di default=”defaultpwd”
Password per l'accesso al keystore.

keyManagerFactory=: valore di default=”SunX509”
Indica il modulo di interpretazione del certificato. Normalmente impostato a “SunX509”

enableClientCertificate=: valore di default=”false”
Se impostato a true e trasmissione SSL viene abilitato l'utilizzo del certificato client.

trustAllCertificates=: valore di default=”false”
Se questo valore viene impostato a true non viene effettuata la verifica attraverso CA o truststore del certificato. Utile in fase di test o se si vuole utilizzare il certificato del servizio al solo fine di eseguire una trasmissione criptata.

trustCertificateURL=: valore di default=””
Se impostato indica l'indirizzo http da cui prelevare il certificato considerato trust.

trustCertificateURIPath=: valore di default=””
Se impostato indica il percorso da cui prelevare il certificato considerato trust.

trustKeyStore=: valore di default=”JKS”
Indica il tipo di keystore SSL trust. Normalmente se si utilizza il keystore della JVM deve essere impostato a “JKS” se si utilizza un keystore OpenSSL si deve impostare “PKCS12”.

trustKeyStorePassword=: valore di default=”defaultpwd”
Passoword per l'accesso al keystore trust.

trustKeyManagerFactory=: valore di default=”SunX509”
Indica il modulo di interpretazione del certificato. Normalmente impostato a “SunX509”.

Esempio <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>
```

Per essere utilizzati negli endpoint è sufficiente includere nei paragrafi <endPointsGrouping> oppure <virtualDomain> oppure direttamente sull'endpoint il parametro SSLCert.

```
<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>
```

In questo paragrafo vengono descritte le regole di rewriting sia della componete HTTP HEADER sia dei contenuti del BODY HTTP durante il forwarding delle informazioni.
OPLON®ADC implementa nativamente un sistema avanzatissimo di rewriting coniugando sia la semplicità espressiva dell'XML sia la possibilità di utilizzare, anche congiuntamente all'XML, il linguaggio JAVA attraverso estensione di classi appositamente studiate allo scopo.
Questo strumento mette a disposizione degli ingegneri un potente strumento con possibilità infinite non solo di rewriting ma anche di integrazione con altre piattaforme come sistemi di SSO o di fatturazione del traffico.

All'interno di questo paragrafo possono coesistere due tipi di sottoparagrafi:

- <rewriteHeaderRule> descrivere regole di rewriting sulle componenti HEADER HTTP
- <rewriteBodyRule> descrive le regole di rewriting sulle componenti BODY HTTP

I paragrafi <rewriteHeaderRule> e <rewriteBodyRule> possono essere ripetuti più volte all'interno del paragrafo <rewriteManagement> e descrivono dei template di comportamento da applicare successivamente a livello di Domain <virtualDomain> fino al singolo <endp>.

La struttura generale del paragrafo <rewriteManagement> è di seguito riportata:

```
<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
```

Il paragrafo <rewriteHeaderRule> descrive una regola di rewriting applicabile a livello di HEADER HTTP. Questo paragrafo può essere ripetuto tante volte quante sono le regole da applicare anche in diversi contesti di forwarding. I parametri relativi a questo paragrafo descrivono le caratteristiche della regola che si vuole esprimere e le condizioni di base.

name=: valore di default=”null”
E' il nome della regola. Questo valore, associato ad uno o più domini <virtualDomain> fino al singolo <endp>, permette di applicare questa regola su più contesti.
Se il valore non è stato valorizzato la regola non viene caricata.

enable=: valore di default=”true”
Abilita o disabilita questa regola

flow=: valore di default=”BOTH” valori disponibili: REQUEST|RESPONSE|BOTH
Identificazione del flusso al quale si vuole applicare la regola.
Se viene indicato il flusso di REQUEST la regola viene applicata per il flusso client--->LBL--->endp.
Se indicato RESPONSE la regola viene applicata per il flusso client<---LBL<---endp.

Se indicato BOTH la regola viene applicata sia per le richieste in entrata sia per le risposte dei servizi verso i client.

httpMethod=: valore di default="" valori disponibili: Metodi HTTP: POST, GET...
Nei flussi di REQUEST identifica il metodo HTTP utilizzato
es.: POST GET PATCH
Se non impostato o valore vuoto viene eseguito su tutti i metodi .
Questo valore nei casi di flow="BOTH" verrà verificato solo nei flussi di REQUEST e quindi può convivere con il parametro responseCode.

responseCode=: valore di default="-1"
Nei flussi di RESPONSE identifica il codice di ritorno HTTP e per il quale la regola verrà applicata. Questo valore nei casi di flow="BOTH" verrà verificato solo nei flussi di RESPONSE e quindi può convivere con il parametro httpMethod.

caseSensitive=: valore di default="false"
Questo valore descrive le modalità di matching delle regole di rewriting (espressioni regolari).
Questo valore viene propagato anche ai propri sottoparagrafi dove potrà comunque essere modificato per eccezione.

httpInterceptorClassPath=:default="interceptors/"
E' il path di caricamento della classe interceptor definita in httpInterceptorClass. Questo path viene aggiunto al classpath della JVM in esecuzione.

httpInterceptorClass=: valore di default=""
Indica la classe per intercettare i pacchetti dati HTTP.
In questo parametro è possibile indicare una classe che implementa la classe

- loadbalancer.rewriter.LBLHTTPInterceptorHeaderAbstr

Questa viene messa a disposizione dalla piattaforma OPLON®ADC per poter intervenire nell'HEADER sia prima che dopo l'applicazione delle regole descritte attraverso questo paragrafo.
Le distribuzioni contengono già una classe template disponibile in
(LBL_HOME)/interceptors/rewriteclasses/LBLHTTP.... .java
Le classi contenute in questa directory possono essere compilate attraverso i tool: compile.bat o compile.sh presenti nella directory stessa.
Il modo di utilizzo delle classi interceptor template è descritto all'interno delle classi stesse.

In ogni interceptor class è possibile utilizzare due metodi richiamati nell'inizializzazione della classe ed alla terminazione della classe. Questi metodi servono ad allocare risorse o inizializzare valori. Il metodo di terminazione della classe serve a deallocare risorse durante i reinit o alla terminazione gracefull del processo di bilanciamento.

```
@Override
public void interceptorInit() {
        /* initialization code */
}

@Override
public void interceptorEnd() {
```

```
		/* termination code */
}
```

LBLHTTPInterceptorHeaderAbstr

La classe "loadbalancer.rewriter.LBLHTTPInterceptorHeaderAbstr" mette a disposizione 4 metodi che identificano anche i quattro momenti fondamentali del flusso dati:

```
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());
	}
}
```

Ogni metodo mette a disposizione l'oggetto LBLHTTPInterceptorHeaderStreamFragment con il quale è possibile agire sul frammento dati durante il flusso (in questo caso l'intera HEADER HTTP).

LBLHTTPInterceptorHeaderStreamFragment contiene già moltissimi metodi per agire nell'HTTP sia in interrogazione dei valori sia in modifica controllata. Una trattazione completa con alcuni esempi è possibile trovarla sul manuale "OPLON®ADC Content Rewriting".

LBLHTTPInterceptorHeaderAbstr è la classe da cui sono derivate le classi interceptor, contiene alcuni metodi di utilità per poter eseguire le operazioni più comuni come quelle di trasformazione dei certificati digitali in diversi formati e/o verifica dei valori significativi come il fingerprint od il serial number.

Segue un elenco dettagliato dei metodi delle classi:
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>
```

Il valore all'interno del paragrafo <requestURLMatches> identifica una espressione regolare. Questa è usata per verificare la corrispondenza con l'URL richiesta, comprensiva di URI Params e Query String.
Se questa condizione è soddisfatta la regola viene applicata.
Se questo paragrafo non è valorizzato o inesistente tutte le URL richieste verranno considerate valide.
Il valore dell'URL request è in formato senza encoding UTF-8 (gli spazi sono rappresentati da spazi e non dalla loro rappresentazione URL encoded).

Esempio:
<requestURLMatches>^/viewProcessProperties.html\?
process=A05_LBLGoDNSManager$</requestURLMatches>

<mimeType>

```
<serviceconf>
    <iproxy>
        <rewriteManagement>
            <rewriteHeaderRule>
                <mimeType
```

Questo paragrafo, che può essere ripetuto, è un'altra condizione per cui la regola potrà essere applicata nel flusso.

enable=: valore di default=”true”
Abilita o disabilita questo paragrafo

value=: valore di default=””
Valore del mimeType con cui è possibile applicare la regola.

Esempi:
"text/html"
"text/css".

<variables>

```
<serviceconf>
    <iproxy>
        <rewriteManagement>
            <rewriteHeaderRule>
                <variables>
```

Questo paragrafo permette di caricare attraverso diverse fonti delle variabili da utilizzarsi all'interno della regola per comporre o modificare i valori e renderli disponibili durante il rewriting.

La creazione di una variabile avviene utilizzando il sottoparagrafo <var> che può essere ripetuto più volte.

es.:
```
<var varName="MY_VAR_FROM_ENTITY"       name="Connection"
                                        from="ENTITY">
</var>
```

In questo caso è stata creata la variabile MY_VAR_FROM_ENTITY prelevando il valore dall'entity "Connection" dell'HEADER durante il passaggio.
Il valore contenuto in MY_VAR_FROM_ENTITY potrebbe essere valorizzato durante il flusso dati con "Keep-alive" oppure "Close" o quanto previsto dal protocollo HTTP.

Nell'esempio seguente abbiamo introdotto anche un modificatore con espressione regolare.

```
<var varName="MY_VAR_FROM_ENTITY"       name="Connection"
                                        from="ENTITY">
                <regexTag>Keep-alive</regexTag>
                <replaceTo>Close</replaceTo>
</var>
```

In questo caso è stata creata la variabile MY_VAR_FROM_ENTITY prelevando il valore dall'entity "Connection" dell'HEADER durante il passaggio e trasformandone il valore da "Keep-alive" a "Close".
Il risultato finale sarà una variabile MY_VAR_FROM_ENTITY con contenuto "Close" se in precedenza "Connection" conteneva "Keep-alive" altrimenti conterrà il valore dell'entity originale.

E' possibile inoltre utilizzare altre variabili precedentemente create nella stessa regola per formare ulteriori variabili.

es.:
```
<var varName="MY_PARAM"         name="paramName"
                                from="URI_PARAM">
        <regexTag>valueInParamQueryString%MY_VAR_FROM_ENTITY%</regexTag>
                <replaceTo>newValueInVar</replaceTo>
</var>
```

In questo caso il valore del <regexTag> prima di essere utilizzato sarà convertito con il valore della variabile in quell'istante della trasmissione.

<var>

```
<serviceconf>
        <iproxy>
                <rewriteManagement>
                        <rewriteHeaderRule>
                                <variables>
                                        <var
```

I parametri di definizione delle variabili contenuti in <variables> sono:

enable=: valore di default=”true”
Abilita o disabilita questo paragrafo

varName=: valore di default=””
Nome della variabile da utilizzarsi nella regola

name=: valore di default=””
Nome/valore della risorsa da cui ricavare il valore da caricare nella variabile (Se valore costante e Supporta il modificatore %VARIABLE%)

caseSensitive=: valore di default=”caseSensitive del paragrafo rewrite rule”
Questo valore descrive le modalità di matching delle regole di rewriting (espressioni regolari).

from=: valore di default=”” valori:
INNERVAR|ENTITY|URI_PARAM|CONSTANT|COOKIE|VARIABLE
Fonte da cui caricare il valore nella variabile.

INNERVAR

- **REQUEST_HTTP_UR**L
  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"

ATTENZIONE: L'utilizzo di REQUEST_HTTP_HOST_NAME può determinare la risoluzione del nome attraverso DNS.
Se il nome non è associato a nessun indirizzo il suo timeout può causare un forte rallentamento

- **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 not entity found
- **ENTITY_RESPONSE**
  return the value of response entity name or empty if not entity found
- **SSL_CONNECTION_CLIENT**
  true se il client si connette in SSL con LBL
- **SSL_CONNECTION_ENDPOINT**
  true se l'endpoint si connette in SSL con LBL
- **SSL_CONNECTION_REENCRYPTION**
  true se viene eseguita la reencryption SSL (quindi LBL fa la terminazione SSL e si connette verso l'endpoint in SSL)
- **REQUEST_INCOMING_ADDRESS**
  indirizzo locale sul quale e' stata accettata la richiesta di servizio
- **REQUEST_INCOMING_HOST_NAME**
  nome host o indirizzo locale sul quale e' stata accettata la richiesta di servizio

ATTENZIONE: L'utilizzo di REQUEST_INCOMING_HOST_NAME può determinare la risoluzione del nome attraverso DNS.
Se il nome non è associato a nessun indirizzo il suo timeout può causare un forte rallentamento

- **REQUEST_HTTP_SCHEME**
  http o https in base al tipo di connessione del client verso LBL
- **HIGH_WATER_YELLOW_WARNING_REACHED**
  se true è stata superata la soglia Yellow Warning. Indica quindi un carico rilevante ma non ancora critico.
- **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 contemporary settings.
- **TUNNEL_SESSIONS_ACTIVE**
  Instant active tunnels, int format.
- **TUNNEL_SESSIONS_COMMITTED**
  Instant tunnel 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.

NOTA: Questi valori "precaricati" per essere utilizzati come modificatori (%xxx%) devono comunque essere caricati in una variabile locale alla regola.

ENTITY

ENTITY=caricamento della variabile descritta in varName con il valore dell'Entity dell'HEADER HTTP il cui nome è indicato in name.

URI_PARAM
URI_PARAM=caricamento della variabile descritta in varName con il valore del parametro o query string del HEADER HTTP il cui nome è indicato in name.

URI_PARAM_DECODED
URI_PARAM_DECODED=Valore del parametro “name” preso dalla query string del HEADER HTTP, in formato decoded

CONSTANT
CONSTANT=caricamento della variabile descritta in varName con il valore del parametro indicato in name. Solo in questo caso il valore contenuto in name può essere composto da un'altra variabile precedentemente caricata.

COOKIE
COOKIE=caricamento della variabile descritta in varName con il valore del Cookie dell'HEADER HTTP il cui nome è indicato in name.

VARIABLE
VARIABLE=caricamento della variabile descritta in varName con il valore di un'altra variabile il cui nome è indicato in name.

<regexTag></regexTag>
<replaceTo></replaceTo>

```
<serviceconf>
    <iproxy>
        <rewriteManagement>
            <rewriteHeaderRule>
                <variables>
                    <var>
                        <regexTag></regexTag>
                        <replaceTo></replaceTo>
                    </var>
```

I paragrafi <regexTag> e <replaceTo>, non obbligatori ma univoci se esistenti a livello di paragrafo <var>, vengono valorizzati con espressioni regolari per applicare delle modificazioni ai valori originali ottenuti dalle varie fonti.
Questi due valori possono inoltre contenere delle ulteriori variabili di modificazione espresse in precedenti paragrafi <var> inerenti alla stessa regola di rewriting.

In particolare:
<regexTag>
espressione regolare applicata nel valore estratto (Supporta %VARIABLE% e caseSensitive)
</regexTag>
<replaceTo>
valore con cui eseguire il replace (Supporta %VARIABLE% e caseSensitive)
</replaceTo>

Esempio <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 questo paragrafo opzionale possono essere aggiunte altre condizioni di applicazione della regola in aggiunta a quanto specificato in maniera facilitata nel paragrafo <rewriteHeaderRule>.
Questo paragrafo è omnicomprensivo e si possono realizzare condizioni su tutti i valori dell'HEADER provenienti dai flussi.

Questo paragrafo inoltre può essere utilizzato in "AND" od in "OR" rispetto le condizioni descritte nei sottoparagrafi.

enable=: valore di default=”true”
Abilita o disabilita questo paragrafo

operator=: valore di default=”AND” valori: AND|OR
Questo valore descrive le modalità di matching delle condizioni di rewriting.
Se utilizzato in "AND" tutte le condizioni descritte nei sottoparagrafi devono essere soddisfatte affinché la regola venga applicata altrimenti se in "OR" anche se una sola condizione viene soddisfatta la regola viene applicata.

<cond>

```
<serviceconf>
        <iproxy>
                <rewriteManagement>
                        <rewriteHeaderRule>
                                <conditions>
                                        <cond
```

enable=: valore di default=”true”
Abilita o disabilita questo paragrafo

name=: valore di default=””
Nome della condizione

caseSensitive=: valore di default=”caseSensitive del paragrafo rewrite rule”
Questo valore descrive le modalità di matching delle regole di rewriting (espressioni regolari).

eval=: valore di default=”” valori: NOT|""
Il valore espresso in eval se caricato a "NOT" è la negazione del risultato dell'espressione regolare.

from=: valore di default=”” valori: INNERVAR|ENTITY|URI_PARAM|CONSTANT|COOKIE|VARIABLE
Fonte da cui caricare il valore della condizione.

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"

ATTENZIONE: L'utilizzo di REQUEST_HTTP_HOST_NAME può determinare la risoluzione del nome attraverso DNS.
Se il nome non è associato a nessun indirizzo il suo timeout può causare un forte rallentamento

- **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 not entity found
- **ENTITY_RESPONSE**
  return the value of response entity name or empty if not entity found
- **SSL_CONNECTION_CLIENT**
  true se il client si connette in SSL con LBL
- **SSL_CONNECTION_ENDPOINT**
  true se l'endpoint si connette in SSL con LBL
- **SSL_CONNECTION_REENCRYPTION**

true se viene eseguita la reencryption SSL (quindi LBL fa la terminazione SSL e si connette verso l'endpoint in SSL)

- **REQUEST_INCOMING_ADDRESS**
  indirizzo locale sul quale e' stata accettata la richiesta di servizio
- **REQUEST_INCOMING_HOST_NAME**
  nome host o indirizzo locale sul quale e' stata accettata la richiesta di servizio

ATTENZIONE: L'utilizzo di REQUEST_INCOMING_HOST_NAME può determinare la risoluzione del nome attraverso DNS.
Se il nome non è associato a nessun indirizzo il suo timeout può causare un forte rallentamento

- **REQUEST_HTTP_SCHEME**
  http o https in base al tipo di connessione del client verso LBL
- **HIGH_WATER_YELLOW_WARNING_REACHED**
  se true è stata superata la soglia Yellow Warning. Indica quindi un carico rilevante ma non ancora critico.
- **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 contemporary settings.
- **TUNNEL_SESSIONS_ACTIVE**
  Instant active tunnels, int format.
- **TUNNEL_SESSIONS_COMMITTED**
  Instant tunnel committed, (subset of TUNNEL_SESSIONS_ACTIVE)

NOTA: Questi valori "precaricati" per essere utilizzati come modificatori (%xxx%) devono comunque essere caricati in una variabile locale alla regola.

ENTITY
ENTITY=caricamento della variabile descritta in varName con il valore dell'Entity dell'HEADER HTTP il cui nome è indicato in name.

URI_PARAM
URI_PARAM=caricamento della variabile descritta in varName con il valore del parametro o query string del HEADER HTTP il cui nome è indicato in name.

URI_PARAM_DECODED
URI_PARAM_DECODED=Valore del parametro “name” preso dalla query string del HEADER HTTP, in formato decoded

CONSTANT
CONSTANT=caricamento della variabile descritta in varName con il valore del parametro indicato in name. Solo in questo caso il valore contenuto in name può essere composto da un'altra variabile precedentemente caricata.

COOKIE

COOKIE=caricamento della variabile descritta in varName con il valore del Cookie dell'HEADER HTTP il cui nome è indicato in name.

VARIABLE
VARIABLE=caricamento della variabile descritta in varName con il valore di un'altra variabile il cui nome è indicato in name.

<regexTag></regexTag>

```
<serviceconf>
        <iproxy>
                <rewriteManagement>
                        <rewriteHeaderRule>
                                <conditions>
                                        <cond
                                                <regexTag></regexTag>
```

Il paragrafo <regexTag> non obbligatorio ed univoco a livello di paragrafo <cond>, viene valorizzato con espressioni regolari per verificare la condizione.

In particolare:
<regexTag>
espressione regolare applicata nel valore estratto (Supporta %VARIABLE% e caseSensitive)
</regexTag>

<numOperatorTag></numOperatorTag>

```
<serviceconf>
        <iproxy>
                <rewriteManagement>
                        <rewriteHeaderRule>
                                <conditions>
                                        <cond
                                                <numOperatorTag></numOperatorTag>
```

Il paragrafo <numOperatorTag> non obbligatorio ed univoco a livello di paragrafo <cond>, viene valorizzato con espressioni numeriche di confronto.

Questo paragrafo è stato introdotto per facilitare i confronti numerici e viene utilizzato in sostituzione del paragrafo <regexTag>.

Il nuovo paragrafo può avere i seguenti operatori:

- **eq**=equal
- **neq**=not equal
- **gt**=greater than
- **geq**=greater equal
- **lt**=less than
- **leq**=less equal

Esempio <numOperatorTag>:

Di seguito un esempio di utilizzo dove vengono verificate due condizioni. La prima condizione testa le INNERVAR "HIGH_WATER" con un operatore maggiore o uguale al

valore 10, la seconda condizione verifica che l'URIPath richiesta sia differente da /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>
```

L'azione <connectionToCut>dell'esempio può essere sostituita dalle azioni: <displaceEndPointsGrouping> oppure <entities> oppure <redirectTo>.

Esempio <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 questo paragrafo opzionale può essere modificata l'HEADER in qualsiasi suo aspetto se le condizioni ne hanno accertato la consistenza.

In particolare questo paragrafo può sia aggiungere e modificare la richiesta/risposta iniziale, sia cancellare ogni singolo entity (entity è una linea di un HEADER suddivisa in "nome:valore").

enable=: valore di default=”true”
Abilita o disabilita questo paragrafo

entityName=: valore di default=””
Nome dell'entity che si vuole utilizzare (es.: "Connection", "Content-Type", "Accept-Encoding".

Nella HEADER di seguito notiamo una prima riga che identifica la richiesta, mentre le righe che seguono identificano gli Entities.

GET /favicon.ico HTTP/1.1
Host: www.tcoproject.com
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

Per effettuare modifiche alla prima linea è sufficiente (sia per REQUEST che per RESPONSE) impostare:
entityName=”FIRST-LINE”

action=: valore di default=”” valori: remove|change|add
E' l'azione che si vuole intraprendere sull'entity.
E' da notare che:

- "add" non rimuove un entity preesistente,
- "change" modifica il valore dell'entity attraverso le espressioni regolari partendo dal valore preesistente oppure con un valore completamente nuovo proposto dal parametro "value".

value=: valore di default=””
Se esistente si sostituisce al valore dell'entity (Supporta il modificatore %VARIABLE%).

caseSensitive=: valore di default=”caseSensitive del paragrafo rewriteRule”
Questo valore descrive le modalità di matching delle regole di rewriting (espressioni regolari).

Esempio <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
```

Nel caso si voglia ridirezionare la richiesta è possibile inserire questo paragrafo, non obbligatorio ed univoco, con il valore dell'URL obiettivo della ridirezione.
Questo paragrafo è potentissimo perché permette di utilizzare le variabili per comporre l'indirizzo di destinazione utilizzando quindi la stessa richiesta per costruire una nuova URL.

enable=: valore di default=”true”
Abilita o disabilita questo paragrafo

responseCode=: valore di default=”302”
E' il response code con cui viene eseguita la ridirezione. se <=0 viene preso il valore di default 302. I valori utilizzabili sono normalmente 301 e 302.

redirectURL=: valore di default=””
Valore dell'URL su cui ridirezionare.
Questo valore puo' contenere variabili che verranno sostituite (Supporta modificatore %VARIABLE%).

Esempio:

```
<redirectTo responseCode="302"
redirectURL="http://www.tcoproject.com/?http://%MY_REQUEST_HTTP_HOST_NAME% "/>
```

<displaceEndPointsGrouping>

```
<serviceconf>
    <iproxy>
        <rewriteManagement>
            <rewriteHeaderRule>
                <displaceEndPointsGrouping
```

Con questa direttiva è possibile spiazzare in un altro EndPointsGrouping una richiesta proveniente da un listener afferente ad un EndPointsGrouping differente.

Questa funzionalità si ottiene utilizzando le regole di rewriting dell'Header con un nuovo paragrafo che indica l'azione di spiazzamento.

La forma più semplice è di seguito riportata:

```
<rewriteHeaderRule enable="true" flow="REQUEST" name="setEndpGrouping">
    <displaceEndPointsGrouping enable="true" endPointsGrouping="services001"/>
</rewriteHeaderRule>
```

Il paragrafo <displaceEndPointsGrouping> (displace = spiazzamento) indica dove la richiesta deve fare riferimento come endPointsGrouping.

E' possibile utilizzare tutte le forme già previste per la composizione dello spiazzamento come ad esempio le variabili e le condizioni:

```
<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>
```

E' possibile comandare il displace anche da classi interceptor HEADER Java:

```
<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) {
      }
}
```

esempio di Header rewriting:

Questo esempio modifica le richieste di redirect provenienti dai server con protocollo https verso lo stesso indirizzo con protocollo http.

```
<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
```

Questo paragrafo è dedicato al Quality Of Application Service. E' infatti possibile attraverso questo paragrafo decidere arbitrariamente di “tagliare” la connessione e di far eseguire, a seconda dei casi, una close dei canali, un redirect oppure esporre la pagina di cortesia descritta nelle funzionalità DoS Attack prevention.

Di seguito riportiamo i template di utilizzo di questo paragrafo utilizzando sia regole XML sia classi interceptor.

La funzione è disponibile con il modulo DoS Attack prevention attivato con apposita licenza.

L'utilizzo di questa funzionalità è descritto da alcuni esempi dove, al verificarsi di condizioni programmabili, viene attivato il taglio delle connessioni.
Per i dettagli guardare il paragrafo <numOperatorTag>

Di seguito un esempio di utilizzo dove vengono verificate due condizioni. La prima condizione verifica le INNERVAR "HIGH_WATER" con un operatore maggiore o uguale al valore 10, la seconda condizione verifica che l'URIPath richiesta sia diverso da /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>
```

E' possibile utilizzare questa funzionalità anche da classe interceptor come indicato nella seguente pagina. Il risultato finale è identico a quanto espresso in forma XML.

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
```

Il paragrafo <rewriteBodyRule> descrive una regola di rewriting applicabile a livello di BODY HTTP. Questo paragrafo può essere ripetuto tante volte quante sono le regole da applicare anche in diversi contesti di forwarding. I parametri relativi a questo paragrafo descrivono le caratteristiche della regola che si vuole esprimere e le condizioni di base.

name=: valore di default=”null”
E' il nome della regola.
Questo valore, associato ad uno o più domini <virtualDomain> fino al singolo <endp>, permette di applicare questa regola su più contesti. Se il valore non è stato valorizzato la regola non viene caricata.

enable=: valore di default=”true”

Abilita o disabilita questa regola

flow=: valore di default=”BOTH” valori disponibili: REQUEST|RESPONSE|BOTH
Identificazione del flusso al quale si vuole applicare la regola.
Se viene indicato il flusso di REQUEST la regola viene applicata per il flusso:
client--->LBL--->endp.

Se indicato RESPONSE la regola viene applicata per il flusso:
client<---LBL<---endp.

Se indicato BOTH la regola viene applicata sia per le richieste in entrata sia per le risposte dei servizi verso i client.

httpMethod=: valore di default=”” valori disponibili: Metodi HTTP: POST, GET...
Nei flussi di REQUEST identifica il metodo HTTP utilizzato es.: POST GET PATCH, se non impostato o valore vuoto viene eseguito su tutti i metodi.
Questo valore nei casi di flow="BOTH" verrà verificato solo nei flussi di REQUEST e quindi può convivere con il parametro responseCode.

responseCode=: valore di default=”-1”
Nei flussi di RESPONSE identifica il codice di ritorno HTTP e per il quale la regola verrà applicata. Questo valore nei casi di flow="BOTH" verrà verificato solo nei flussi di RESPONSE e quindi può convivere con il parametro httpMethod.

caseSensitive=: valore di default=”false”
Questo valore descrive le modalità di matching delle regole di rewriting (espressioni regolari). Questo valore viene propagato anche ai propri sottoparagrafi dove potrà comunque essere modificato per eccezione.

inspectionOnly=: valore di default=”false”
Se true, si dichiara che la regola non altera il contenuto del body e quindi non verrà modificato il tipo di body a "Transfer-Encoding: chunked" per permetterne la modifica on-the-fly. Questo determina l'utilizzo di classi di inspection body anche su application server che non implementano POST di tipo chunked.

charset=: valore di default=”JVM default charset”
La gerarchia di attribuzione del charset durante il rewriting è la seguente:

- Verifica se nell'entity "Content-Type: text/html; charset=iso-8859-1" esiste l'indicazione del charset da utilizzare
- Se non trovata l'indicazione del charset nell'entity Content-Type viene presa la definizione del charset nel paragrafo <rewriteBodyRule ... charset="iso-8859-1">
- Se il parametro <rewriteBodyRule ... charset="iso-8859-1"> non era stato specificato viene assunto il valore del charset dell'utente di lancio della JVM

Il charset del content rewriting viene acquisito dall'HEADER HTTP attraverso l'entity "Content-Type: text/html; charset=iso-8859-1". Nel caso l'entity "Content-Type" non contenga l'indicazione del charset e' possibile forzare il charset attraverso il parametro "charset" nella regola di rewriting nel paragrafo <rewriteBodyRule> come riportato di

seguito.

```
<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>
```

Nel caso non venga specificato il valore charset nel paragrafo <rewriteBodyRule> verra' assunto di default il charset di start della JVM.

Allo start la JVM acquisisce a default il charset della piattaforma, l'utente da cui e' stato eseguito lo start. Per cambiare il charset di default della JVM e' sufficiente impostare nel profilo di lancio il parametro:

-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=: valore di default=””
In questo parametro è possibile indicare una classe che implementa la classe

- loadbalancer.rewriter.LBLHTTPInterceptorBodyAbstr"

messa a disposizione dalla piattaforma OPLON®ADC per poter intervenire nel BODY sia prima che dopo l'applicazione delle regole descritte attraverso questo paragrafo.

LBLHTTPInterceptorBodyAbstr

La classe "loadbalancer.rewriter.LBLHTTPInterceptorBodyAbstr" mette a disposizione 4 metodi che identificano anche i quattro momenti fondamentali del flusso dati.

```
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());
	}
}
```

Ogni metodo mette a disposizione l'oggetto LBLHTTPInterceptorBodyStreamFragment con il quale è possibile agire sul frammento dati durante il flusso.

LBLHTTPInterceptorBodyStreamFragment contiene già moltissimi metodi per agire nell'HTTP BODY fragment sia in interrogazione dei valori sia in modifica controllata. Una trattazione completa con alcuni esempi è possibile trovarla sul manuale "OPLON®ADC Content Rewriting"

LBLHTTPInterceptorBodyAbstr è la classe da cui sono derivate le classi interceptor.
La classe contiene alcuni metodi di utilità per poter eseguire le operazioni più comuni come quelle di trasformazione dei certificati digitali in diversi formati e/o verifica dei valori significativi come il fingerprint od il serial number.

Segue un elenco dettagliato dei metodi delle classi:
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
 * @return endpoint host address or null  if not found
 */
public String getResponseEndpointAddress()
```

1 Per i metodi riferirsi a LBLHTTPInterceptorHeaderAbstr

```
/**
 * 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)
```

ATTENZIONE

Alcuni application server non supportano il metodo POST con "transfer-encoding: chunked". Ad esempio Apache Tomcat ha risolto questo problema sicuramente dalla versione 6.0.26 con versioni inferiori (es.:6.0.14) non funziona.

Accertarsi che l'application server sia compliant alla raccomandazione: HTTP1.1 transfer-encoding: chunked

<requestURLMatches></requestURLMatches>

```
<serviceconf>
    <iproxy>
        <rewriteManagement>
```

```
                                <rewriteBodyRule>
                                        <requestURLMatches></requestURLMatches>
```

Il valore all'interno del paragrafo <requestURLMatches> identifica una espressione regolare per verificare la corrispondenza con l'URL richiesta comprensiva di URI Params e Query String. Se la condizione è soddisfatta la regola viene applicata.
Se questo paragrafo non è valorizzato o inesistente tutte le URL richieste verranno considerate valide.
Il valore dell'URL request è già in formato senza encoding UTF-8 (gli spazi sono rappresentati da spazi e non dalla loro rappresentazione URL encoded).

```
<requestURLMatches>^/viewProcessProperties.html\?process=A05_LBLGoDNSManager$</requestURLMatches>
```

<mimeType>

```
        <serviceconf>
                <iproxy>
                        <rewriteManagement>
                                <rewriteBodyRule>
                                        <mimeType
```

Questo paragrafo, che può essere ripetuto, è un'altra condizione per cui la regola potrà essere applicata nel flusso.
Nel caso del BODY, a differenza del trattamento dell'HEADER, il mimeType assume una grande importanza perché oltre a definire il tipo di mime type su cui applicare la regola definisce anche la regola di frammentazione consistente del body.
Il BODY infatti può assumere delle dimensioni notevoli ed è quindi necessario frammentarlo in tanti “buffer” (chiamati chunk) per poterlo trasferire attraverso il bilanciatore.
Le modifiche ad opera del rewriter devono essere però apportate in maniera consistente anche quando un valore da modificare è tra due “chunk”.
OPLON®ADC è in grado di ricomporre un frammento in parti consistenti dando i limiti di consistenza per quel particolare mime type.
Per text/html è ad esempio i caratteri “<” e “>” distinguono la consistenza di un blocco.
Per text/css sono le parentesi graffe “{“ e “}” che contraddistinguono un blocco consistente.

enable=: valore di default=”true”
Abilita o disabilita questo paragrafo

value=: valore di default=””
Valore del mimeType con cui è possibile applicare la regola, es.: "text/html" oppure "text/css" etc.

fragmentOpen=: valore di default=””
Questo parametro, assieme al parametro fragmentClose, è importantissimo per determinare un frammento consistente di dati. Infatti il rewriting del body avviene per frammenti consistenti dovendo trattare non l'intero body bensì un “buffer” di forwarding con lunghezza nota. Essendo una notazione xml si devono sostituire alcuni caratteri con i loro tag es.:
per “>” “<” gli equivalenti sono “>” e “<” per le “&” devono essere utilizzati “&”

fragmentClose=: valore di default=””

Questo parametro, assieme al parametro fragmentOpen, è importantissimo per determinare un frammento consistente di dati. Infatti il rewriting del body avviene per frammenti consistenti dovendo trattare non l'intero body bensì un “buffer” di forwarding con lunghezza nota. Essendo una notazione xml si devono sostituire alcuni caratteri con i loro tag es.:
per “>” “<” gli equivalenti sono “>” e “<” per le “&” devono essere utilizzati “&”

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="}"/>
```

Se fragmentOpen o fragmentClose non vengono valorizzati non verrà effettuata nessuna frammentazione e il frame sarà disponibile senza logicità. Questo è molto utile nei casi in cui ad esempio si vogliono effettuare operazioni bulk come la compressione.

<conditions>

```
<serviceconf>
    <iproxy>
        <rewriteManagement>
            <rewriteBodyRule>
                <conditions>
```

Vedere <conditions> del paragrafo<rewriteHeaderRule>

<regexTag></regexTag>

Vedi <replaceTo>.

<replaceTo></replaceTo>

```
<serviceconf>
    <iproxy>
        <rewriteManagement>
            <rewriteBodyRule>
                <regexTag></regexTag>
                <replaceTo></replaceTo>
```

I paragrafi <regexTag> e <replaceTo>, non obbligatori ma univoci se esistenti a livello di paragrafo, vengono valorizzati con espressioni regolari per modificare il contenuto del BODY. Questi due valori possono contenere delle ulteriori variabili di modificazione espresse in precedenti paragrafi <var> inerenti alla stessa regola di rewriting.

In particolare:
<regexTag>
espressione regolare applicata nel valore estratto (Supporta %VARIABLE% e caseSensitive)
</regexTag>

<replaceTo>
valore con cui eseguire il replace (Supporta %VARIABLE% e caseSensitive)
</replaceTo>

Esempi rewriting body

Rewriting del body per portare un riferimento URL da assoluto a relativo:

```
<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>
```

Cambiamento di un valore su un 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>
```

Cambiamento di descrizioni all'interno del 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>
```

Questo paragrafo delimita l'elenco di servizi di back-end. La sua sintassi è:

```
<endpoints>
        <endPointsGrouping>
          <virtualDomain>
                </endp>
                ….
                </endp>
          </virtualDomain>
        </endPointsGrouping>
</endpoints>
```

<endPointsGrouping>

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

```
                    <endpoints>
                        <endPointsGrouping
```

groupName=: valore di default=”default”
E' il nome dei gruppi a cui i servizi appartengono.
Questo valore, associato ad uno o più listener, identifica un gruppo di servizi su cui bilanciare il traffico. E' possibile inserire più gruppi separati da almeno uno spazio come nell'esempio di seguito.

```
<endPointsGrouping groupName=" services   test001    test002 test003" enable="true">
```

In questo modo è possibile ad esempio associare più listeners con differenti parametri ma associati agli stessi servizi.

enable=: valore di default=”true”
Abilita o disabilita questo gruppo di servizi

protocol=: valore di default=””
Se valorizzato cambia per il gruppo il protocollo impostato sul listener.

redirectToHttps=: valore di default=”false”
Se true, in presenza di servizi a layer 7 HTTP/S verifica se la richiesta è in HTTP ed esegue un redirect per tutte le risorse interessate a HTTPS.

sniHostName=: valore di default=””
Nome dell'host utilizzato per le connessioni TLS SNI verso il backend.
Viene utilizzato per bilanciare servizi di backend in SSL che implementano il protocollo Server Name Indication. Se impostato il parametro viene ereditato nei VirtualDomain ed endp contenuti nella sezione endPointsGrouping. ATTENZIONE: Se viene impostato questo valore il parametri address nei paragrafi endp devono essere impostati con il valore numerico (ipv4 e ipv6 notation) e non con un nome host.

sniForwarding=: valore di default=”false”
Se true permette di impostare a SNI degli endpoint e di esguire il forward automatico dell'hortname della richiesta verso il backend con lo stesso valore della provenienza.

S**SLApplicationProtocols**=: valore di default=””
In questo parametro si possono elencare i protocolli abilitati su comunicazioni ALPN. I valori ammessi “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=: valore di default=”true”
Se true utilizza i chipersuite nell'ordine indicato per i listener SSL/TLS

SSLProtocols=: valore di default=””
Il parametro imposta i protocolli SSL da utilizzare per la comunicazione.

associateName=: valore di default=””
In questo parametro si possono associare dei nomi simbolici al gruppo di end-point nella forma :
“nome_simbolico_a nome_simbolico_b nome_simbolico_c “.
Ogni nome simbolico deve essere separato da uno spazio da un altro nome simbolico. Il nome simbolico non può contenere spazi.
Per la trattazione dei nomi associativi fare riferimento a: <params notificationDir=”” />

loadBalancingType=: valore di default=”Adaptative”
Imposta la politica di bilanciamento.
Valori possibili:

- RoundRobin
- Adaptative
- FailOver

idSessionsManagerName=: valore di default=”default”
Imposta il tipo di riconoscimento della sessione a layer 7 HTTP/S.
Questo valore deve essere associato ad un paragrafo <idSessions>. Il valore sarà riportato su tutti i domini e gruppi di URIPath all'interno di questo raggruppamento se non diversamente indicato.

dosMaxConcurrentConnectionsReaction=: valore di default=”false”
Attiva il DDoS congestion resolver service capping, indicato dal parametro maxConcurrentConnections del paragrafo <endp>.

NOTA L'attivazione di questa funzionalità è subordinata alla presenza della licenza specifica di DoS Attack Prevention.

realmLogin=: valore di default=””: valore di default=””
Imposta il login basic authentication del servzio.
Questo login assieme alla password verranno consegnati ai servizi endp, se non specificato diversamente nel domain o nell'endp, come entity HTTP attraverso la forma es.:

- authorization: Basic dXNyMjp1c3Iy

realmPassword=: valore di default=””
Imposta la password basic authentication del servizio.
Questa password assieme al login verranno consegnati ai servizi endp, se non specificato diversamente nel domain o nell'endp, come entity HTTP attraverso la forma es.:

- authorization: Basic dXNyMjp1c3Iy

rewriteHeaderRules=: valore di default=””
Elenco dei nomi delle regole di rewriting dell'HEADER HTTP (layer 7 HTTP/S) da applicare al raggruppamento di endpoint.
E' possibile indicare più regole separate da uno o più spazi.

es.
```
rewriteHeaderRules="redirSSLloginWhenNoSSL proxyTo"
```

Le regole verranno applicate, se le condizioni lo permettono, in sequenza.

;**LAST**
Per ogni nome regola è possibile inoltre indicare il parametro LAST che nel caso la regola venga eseguita determina lo stop dell'applicazione delle restanti regole della sequenza.
Per indicare il parametro è sufficiente mettere dopo il nome il “;” e il nome del parametro.

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

Tra il nome e il/i parametri non devono esserci spazi. In questo caso se la regola redirSSLloginWhenNoSSL viene applicata la regola proxyTo non verrà mai eseguita.

;**ALWAYS**
Il parametro ALWAYS indica che la regola viene sempre eseguita
Per indicare il parametro è sufficiente mettere dopo il nome il “;” e il nome del parametro.

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

In questo caso la regola proxyTo viene eseguita a prescindere dall'esecuzione della regola redirSSLloginWhenNoSSL.

;**NOP**
Il parametro NOP indica che la regola non deve essere eseguita.

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

In questo caso la regola redirSSLloginWhenNoSSL non viene eseguita. Il parametro NOP è utile per escludere l'esecuzione di regole generali.

rewriteBodyRules=: valore di default=””
Elenco dei nomi delle regole di rewriting del BODY HTTP (layer 7 HTTP/S) da applicare al raggruppamento di endpoint. E' possibile indicare più regole separate da uno o più spazi.

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

Le regole verranno applicate, se le condizioni lo permettono, in sequenza.

;**LAST**
Per ogni nome regola è possibile inoltre indicare il parametro LAST che nel caso la regola venga eseguita determina lo stop dell'applicazione delle restanti regole della sequenza
Per indicare il parametro è sufficiente mettere dopo il nome il “;” e il nome del parametro.
es.

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

Tra il nome e il/i parametri non devono esserci spazi. In questo caso se la regola addTrademarkParam viene applicata le regole absoluteToRelative echoRewriteBody non verranno mai eseguite.

;ALWAYS
Il parametro ALWAYS indica che la regola viene sempre eseguita.

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

In questo caso la regola echoRewriteBody viene eseguita a prescindere dall'esecuzione della regola addTrademarkParam.

;**NOP**
Il parametro NOP indica che la regola non deve essere eseguita.

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

In questo caso la regola absoluteToRelative non viene eseguita. Il parametro NOP è utile per escludere l'esecuzione di regole generali.

cMessage=: valore di default=”messageNoEndPoint.html”
Nome del file in formato html contenente il messaggio di cortesia per il contesto al quale è associato.
Al momento del caricamento dei parametri, allo start del processo di bilanciamento, viene verificata l'effettiva esistenza del file nella directory:

- (LBL_HOME)/procsProfiles/A10_LBLGo/resources/html/messageName.html

Nel caso il file non sia esistente viene segnalato al momento dell'avvio e assegnato il valore di default:

- (LBL_HOME)/procsProfiles/A10_LBLGo/resources/html/messageNoEndPoint.html

emptyGroupDomainEval=: default value=”Da params” UM=Boolean
Ora è possibile avere un set di gruppo/dominio senza endpoint o con tutti Endpoint impostati su abilita = "false".

Se si desidera attivare questa opzione, è necessario impostare un parametro nella base iproxy params, endpointsgrouping o virtualdomain:

EmptyGroupDomaineAval = "True"

Il valore predefinito EmptyGroupDomaineval è "falso"

Il paragrafo minimo per impostare un gruppo/dominio senza un endpoint è come segue:

```
<endpointsGrouping GroupName = "GrpName" abilita = "True" RewriteHeaderRules =
                        "MyRule, MyRule1">
                <virtualdomain able = "true">
                        <endp able = "false"/>
                </virtualdomain>
</endpointsgrouping>
```

In questo modo, è possibile impostare temporaneamente gruppi/domini e assegnare loro tutte le caratteristiche
come la pagina di cortesia, le regole di riscrittura e ciò che può essere assegnato a un gruppo di endpoint omogenei

Nelle versioni precedenti, le regole di ristrutturazione venivano sempre eseguite nella "richiesta"
Streaming anche se gli endpoint sono stati disabilitati in base al runtime (disabilita/outoforder/drenaggio)
E anche ora è ancora ad eccezione del "reindirizzamento/scorciatoia"
Endpoint. Tuttavia, non sono stati eseguiti in caso di endpoint completamente disabilitato dalla configurazione (abilita = "false").

Abbiamo diversificato il comportamento per renderlo più flessibile.

Per i reindirizzamenti e le scorciatoie, se l'endpoint istruisce un reindirizzamento o il collegamento è disabilitato dal runtime (disabilita/outoforder/drena) questo non più funziona e restituisce la pagina di errore 404. Questo è logico perché se un operatore disabilita (disabilita /outofter /drena) potresti pensare che non voglia fare
il reindirizzamento o il collegamento più.

Invece, per endpoint diversi da reindirizzamenti o scorciatoie (tutti gli altri),
È stato deciso di eseguire comunque la regola della richiesta perché potrebbe contenere si sposta ad altri gruppi abilitati per l'esecuzione.

Anche se tutti gli endpoint sono disabilitati dalla configurazione (abilita = "false") il il comportamento sarà lo stesso.

Di seguito è riportata una tabella di riepilogo che descrive quando le regole vengono eseguite rispetto al rispetto
allo stato degli endpoint e alla loro funzione.

L7 HTTP RewriteRule exec	Service	Redirect	Shortcut
endPoint enable="false"	REQUEST	no execution	no execution

disable/outOfOrder/Drain	REQUEST	no execution	no execution

L2/L3/L4 RewriteRule exec TCP/UDP	Service	Redirect	Redirect
endPoint enable="false"	no execution	no execution	no execution
disable/outOfOrder/Drain	no execution	no execution	no execution

redirectNoHostsFound=: valore di default="" UM=URL
Se Layer 7 HTTP/S in caso di hosts non disponibili per il raggruppamento "endPointsGrouping-virtualDomain-uriPath " è possibile, in alternativa al messaggio di cortesia, comandare un redirect ad un altro URL. Nel caso questo parametro non sia esistente o valorizzato a "" verrà utilizzato il messaggio di cortesia.

needClientCert=: valore di default="false" UM=boolean
Se Layer 7 HTTP/S questo valore impone a tutto il raggruppamento di domini ed endpoint l'utilizzo del certificato client per poter usufruire dei servizi associati. Per differenza i domini e gli endpoint possono modificare tale valore.

needClientCertMessage=: valore di default="messageNeedClientCert.html"
Nome del file in formato html contenente il messaggio di cortesia per il contesto al quale è associato.
Al momento del caricamento dei parametri, allo start del processo di bilanciamento, viene verificata l'effettiva esistenza del file nella directory:

- (LBL_HOME)/procsProfiles/A10_LBLGo/resources/html/messageName.html

Nel caso il file non sia esistente viene segnalato al momento dell'avvio e assegnato il valore di default:

- (LBL_HOME)/procsProfiles/A10_LBLGo/resources/html/ messageNeedClientCert.html

SSLCert=: valore di default=""
E' il riferimento al certificato digitale da utilizzarsi come "client certificate" da OPLON®LoadBalancer verso gli endpoints di questo gruppo.

cipherSuites=: valore di default=""
Il parametro imposta le ciphersuites da utilizzare per la comunicazione con il backend.

SSLProtocols=: valore di default=""
Il parametro imposta i protocolli SSL da utilizzare per la comunicazione con il backend.

Esempio:

```
<endPointsGroupingSSLProtocols="TLSv1.1 TLSv1.2" enable="true">
```

Solo i protocolli TLSv 1.1 e TLSv 1.2 sono utilizzati per comunicare con il backend

<virtualDomain>

```
<serviceconf>
    <iproxy>
        <endpoints>
            <endPointsGrouping>
                <virtualDomain
```

enable=: valore di default=”true”
Abilita o disabilita questo gruppo di servizi

virtualDomainName=: valore di default=”default”
E' il nome dei domini virtuali a cui i servizi appartengono.
Questo valore identifica un gruppo di servizi omogenei su cui bilanciare il traffico. E' possibile inserire più nomi di dominio associandoli agli stessi servizi endpoint come nell'esempio di seguito:

```
<virtualDomain virtualDomainName="  localhost   www.tcoproject.dev pandorum  127.0.0.1" enable="true"
        rewriteHeaderRules...
```

E' inoltre possibile indicare il nome del dominio con una espressione regolare.

```
<endPointsGrouping enable="true">
        <virtualDomain virtualDomainName="www.pluto.com (.*)\.pippo\.com"
```

protocol=: valore di default=””
Se valorizzato cambia per il gruppo il protocollo impostato sul listener.

redirectToHttps=: valore di default=”false”
Se true, in presenza di servizi a layer 7 HTTP/S verifica se la richiesta è in HTTP ed esegue un redirect per tutte le risorse interessate a HTTPS.

sniHostName=: valore di default=”value in endPointsGrouping:=sniHostName”
Nome dell'host utilizzato per le connessioni TLS SNI verso il backend.
Viene utilizzato per bilanciare servizi di backend in SSL che implementano il protocollo Server Name Indication. Se impostato il parametro viene ereditato negli endp contenuti nella sezione virtualDomain. ATTENZIONE: Se viene impostato questo valore il parametri address nei paragrafi endp devono essere impostati con il valore numerico (ipv4 e ipv6 notation) e non con un nome host.

sniForwarding=: valore di default=”false”
Se true permette di impostare a SNI degli endpoint e di esguire il forward automatico dell'hortname della richiesta verso il backend con lo stesso valore della provenienza.

S**SLApplicationProtocols**=: valore di default=””
In questo parametro si possono elencare i protocolli abilitati su comunicazioni ALPN. I valori ammessi “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=: valore di default=”true”
Se true utilizza i chipersuite nell'ordine indicato per i listener SSL/TLS

SSLProtocols=: valore di default=””
Il parametro imposta i protocolli SSL da utilizzare per la comunicazione.

portRewriting=: valore di default=”true”
Se impostato a true riscrive il numero di porta nell'header proveniente dall'endpoint con il valore della porta richiesta dal client.
Questo comportamento è necessario per bilanciare servizi di backend che rispondono modificando l'header originale della richiesta con il proprio port number. Con IIS questo parametro deve essere impostato a true.

sslRewriting=: valore di default=”true”
Se impostato a true verifica se il listener e' in SSL ed eventualmente riscrive il protocollo da http a https in caso di redirect comandate dal servizio aventi protocollo http.
Questa funzionalità è utile nel caso il listener funga da terminatore ssl e nel backend vi siano dei servizi in chiaro.

SSLCert=: valore di default=”endPointGrouping:=SSLCert”
E' il riferimento al certificato digitale da utilizzarsi come “client certificate” da OPLON®LoadBalancer verso gli endpoints di questo dominio.

associateName=: valore di default=””
In questo parametro si possono associare dei nomi simbolici al gruppo di end-point nella forma:
“nome_simbolico_a nome_simbolico_b nome_simbolico_c “.
Ogni nome simbolico deve essere separato da uno spazio da un altro nome simbolico. Il nome simbolico non può contenere spazi.
Per la trattazione dei nomi associativi fare riferimento a <params notificationDir=”” />

loadBalancingType=: valore di default=”Adaptative”
Imposta la politica di bilanciamento.
Valori possibili:

- RoundRobin
- Adaptative
- FailOver

idSessionsManagerName=: valore di default=”default”
Imposta il tipo di riconoscimento della sessione a layer 7 HTTP/S.
Questo valore deve essere associato ad un paragrafo <idSessions>. Il valore sarà riportato su tutti i gruppi URIPath all'interno di questo raggruppamento se non diversamente indicato.

cacheControl=: valore di default=””
In questo parametro devono essere elencati i filtri preventivamente creati nel paragrafo <cacheControl> riportandone l'id, o gli id, di riferimento per questo dominio.

dosMaxConcurrentConnectionsReaction=: valore di default=”false”
Attiva il DDoS congestion resolver service capping, indicato dal parametro maxConcurrentConnections del paragrafo <endp>.

NOTA L'attivazione di questa funzionalità è subordinata alla presenza della licenza specifica di DoS Attack Prevention.

realmLogin=: valore di default=”realmLogin from endPointsGrouping”
Imposta il login basic authentication del servzio.
Questo login assieme alla password verranno consegnati ai servizi endp, se non specificato diversamente nell'endp, come entity HTTP attraverso la forma es.:

- authorization: Basic dXNyMjp1c3Iy

realmPassword=: valore di default=”realmPassword from endPointsGrouping”
Imposta la password basic authentication del servzio. Questa passoword assieme al login verranno consegnati ai servizi endp, se non specificato diversamente nell'endp, come entity HTTP attraverso la forma es.:

- authorization: Basic dXNyMjp1c3Iy

rewriteHeaderRules=: valore di default=”endPointsGrouping +”
Elenco dei nomi delle regole di rewriting dell'HEADER HTTP (layer 7 HTTP/S) da applicare al dominio. E' possibile indicare più regole separate da uno o più spazi.
es.

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

Le regole verranno applicate, se le condizioni lo permettono, in sequenza.

;**LAST**
Per ogni nome regola è possibile inoltre indicare il parametro LAST che nel caso la regola venga eseguita determina lo stop dell'applicazione delle restanti regole della sequenza.
Per indicare il parametro è sufficiente mettere dopo il nome il “;” e il nome del parametro.

es.

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

Tra il nome e il/i parametri non devono esserci spazi. In questo caso se la regola redirSSLloginWhenNoSSL viene applicata la regola proxyTo non verrà mai eseguita.

;**ALWAYS**
Il parametro ALWAYS indica che la regola viene sempre eseguita
Per indicare il parametro è sufficiente mettere dopo il nome il “;” e il nome del parametro.

es.

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

In questo caso la regola proxyTo viene eseguita a prescindere dall'esecuzione della regola redirSSLloginWhenNoSSL.

;**NOP**
Il parametro NOP indica che la regola non deve essere eseguita.

es.

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

In questo caso la regola redirSSLloginWhenNoSSL non viene eseguita. Il parametro NOP è utile per escludere l'esecuzione di regole generali.

rewriteBodyRules=: valore di default=”endPointsGrouping +”
Elenco dei nomi delle regole di rewriting del BODY HTTP (layer 7 HTTP/S) da applicare al dominio. E' possibile indicare più regole separate da uno o più spazi.

es.

```
rewriteBodyRules="addTrademarkParam absoluteToRelative echoRewriteBody"
```

Le regole verranno applicate, se le condizioni lo permettono, in sequenza.

;**LAST**
Per ogni nome regola è possibile inoltre indicare il parametro LAST che nel caso la regola venga eseguita determina lo stop dell'applicazione delle restanti regole della sequenza
Per indicare il parametro è sufficiente mettere dopo il nome il “;” e il nome del parametro.
es.

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

Tra il nome e il/i parametri non devono esserci spazi. In questo caso se la regola addTrademarkParam viene applicata le regole absoluteToRelative echoRewriteBody non verranno mai eseguite.

;ALWAYS
Il parametro ALWAYS indica che la regola viene sempre eseguita.

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

In questo caso la regola echoRewriteBody viene eseguita a prescindere dall'esecuzione della regola addTrademarkParam.

;**NOP**
Il parametro NOP indica che la regola non deve essere eseguita.

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

In questo caso la regola absoluteToRelative non viene eseguita. Il parametro NOP è utile per escludere l'esecuzione di regole generali.

cMessage=: valore di default=”endPointsGrouping: cMessage”
Nome del file in formato html contenente il messaggio di cortesia per il contesto al quale è associato.
Al momento del caricamento dei parametri, allo start del processo di bilanciamento, viene verificata l'effettiva esistenza del file nella directory:

- (LBL_HOME)/procsProfiles/A10_LBLGo/resources/html/messageName.html

Nel caso il file non sia esistente viene segnalato al momento dell'avvio e assegnato il valore di default:

- (LBL_HOME)/procsProfiles/A10_LBLGo/resources/html/messageNoEndPoint.html

emptyGroupDomainEval=: default value=”Da gruppo” UM=Boolean
Ora è possibile avere un set di gruppo/dominio senza endpoint o con tutti
Endpoint impostati su abilita = "false".

Se si desidera attivare questa opzione, è necessario impostare un parametro nella base iproxy params, endpointsgrouping o virtualdomain:

EmptyGroupDomaineAval = "True"

Il valore predefinito EmptyGroupDomaineval è "falso"

Il paragrafo minimo per impostare un gruppo/dominio senza un endpoint è come segue:

```
<endpointsGrouping GroupName = "GrpName" abilita = "True" RewriteHeaderRules =
                        "MyRule, MyRule1">
                <virtualdomain able = "true">
                        <endp able = "false"/>
                </virtualdomain>
</endpointsgrouping>
```

In questo modo, è possibile impostare temporaneamente gruppi/domini e assegnare loro tutte le caratteristiche
come la pagina di cortesia, le regole di riscrittura e ciò che può essere assegnato a un gruppo di endpoint omogenei

Nelle versioni precedenti, le regole di ristrutturazione venivano sempre eseguite nella "richiesta"
Streaming anche se gli endpoint sono stati disabilitati in base al runtime (disabilita/outoforder/drenaggio)
E anche ora è ancora ad eccezione del "reindirizzamento/scorciatoia"
Endpoint. Tuttavia, non sono stati eseguiti in caso di endpoint completamente

disabilitato dalla configurazione (abilita = "false").

Abbiamo diversificato il comportamento per renderlo più flessibile.

Per i reindirizzamenti e le scorciatoie, se l'endpoint istruisce un reindirizzamento o il collegamento è disabilitato dal runtime (disabilita/outoforder/drena) questo non più funziona e restituisce la pagina di errore 404. Questo è logico perché se un operatore disabilita (disabilita /outofter /drena) potresti pensare che non voglia fare il reindirizzamento o il collegamento più.

Invece, per endpoint diversi da reindirizzamenti o scorciatoie (tutti gli altri),
È stato deciso di eseguire comunque la regola della richiesta perché potrebbe contenere si sposta ad altri gruppi abilitati per l'esecuzione.

Anche se tutti gli endpoint sono disabilitati dalla configurazione (abilita = "false") il il comportamento sarà lo stesso.

Di seguito è riportata una tabella di riepilogo che descrive quando le regole vengono eseguite rispetto al rispetto
allo stato degli endpoint e alla loro funzione.

L7 HTTP RewriteRule exec	Service	Redirect	Shortcut
endPoint enable="false"	REQUEST	no execution	no execution
disable/outOfOrder/Drain	REQUEST	no execution	no execution

L2/L3/L4 RewriteRule exec TCP/UDP	Service	Redirect	Redirect
endPoint enable="false"	no execution	no execution	no execution
disable/outOfOrder/Drain	no execution	no execution	no execution

redirectNoHostsFound=: valore di default=”endPointsGrouping:redirectNoHostsFound”
Se Layer 7 HTTP/S in caso di hosts non disponibili per il raggruppamento “endPointsGrouping-virtualDomain-uriPath “ è possibile, in alternativa al messaggio di cortesia, comandare un redirect ad un altro URL.
Nel caso questo parametro non sia esistente o valorizzato a “” verrà utilizzato il messaggio di cortesia.

needClientCert=: valore di default=”endPointsGrouping:needClientCert” UM=boolean
Se Layer 7 HTTP/S questo valore impone a tutto il raggruppamento di domini ed endpoint l'utilizzo del certificato client per poter usufruire dei servizi associati. Per differenza i domini e gli endpoint possono modificare tale valore.

needClientCertMessage=: valore di default=”endPointsGrouping:needClientCertMessage”
Nome del file in formato html contenente il messaggio di cortesia per il contesto al quale è associato.
Al momento del caricamento dei parametri, allo start del processo di bilanciamento, viene verificata l'effettiva esistenza del file nella directory:

- (LBL_HOME)/procsProfiles/A10_LBLGo/resources/html/messageName.html

Nel caso il file non sia esistente viene segnalato al momento dell'avvio e assegnato il valore di default:

- (LBL_HOME)/procsProfiles/A10_LBLGo/resources/html/ messageNeedClientCert.html

cipherSuites=: valore di default=””
Il parametro imposta le ciphersuites da utilizzare per la comunicazione con il backend.

SSLProtocols=: valore di default=””
Il parametro imposta i protocolli SSL da utilizzare per la comunicazione con il backend.

proxyHost=: valore di default=””
Host name/dominio del servizio esposto internamente
Se valorizzato l'host del servizio viene mascherato con il nome indicato nel virtualDomainName

proxyPort=: valore di default=””
Porta del servizio esposto internamente
Se valorizzato la porta del servizio viene mascherata con la porta presente nell'endpoint.

removePortFromHost=: valore di default=”false”
Se true, rimuove la porta dall'elemento header Host:

proxyUrl=: valore di default=””
URI / contesto del servizio
Permette di mascherare l' URI del servizio con il valore indicato nel parametro uriPath

proxyLocation=: valore di default=”true”
Riscrive il contenuto dell'header Location con coerentemente con i valori indicati nei parametri proxyHost e proxyUrl

proxyDestination=: valore di default=”true”
Riscrive il contenuto dell'header Destination con coerentemente con i valori indicati nei parametri proxyHost e proxyUrl. Usato prevalentemente nel bilanciamento di servizi webDav

proxyCookie=: valore di default=”true”
Riscrive l'attributo path e domain dei cookie coerentemente con i valori indicati nei parametri proxyHost e proxyUrl

proxyRedirect=: valore di default=”false”
Se impostato a true verifica se la richiesta corrisponde al valore impostato in proxyURL e

esegue una ridirezione verso uriPATH.
es.:
uriPath=/images
proxyURL=/service/images
proxyRedirect=”true”
proxyRedirectResponseCode=”308”

se la richiesta è:
/service/images/image180x71.png

In questo caso verrà eseguito un redirect a:
/images/image180x71.png

proxyRedirectResponseCode=: valore di default=”308”
Se proxyRedirect è impostato a true è il response code che verrà utilizzato nel redirect

<endp>

```
<serviceconf>
        <iproxy>
                <endpoints>
                        <endPointsGrouping>
                                <virtualDomain>
                                        <endp
```

address=: valore di default= impostare obbligatoriamente
E' il nome o indirizzo del server di backend che ospita il servizio

dhcp=: valore di default= “false”
Se true verifica il nome host e converte il nome in indirizzo per avere endpoint con indirizzo dinamico come ad esempio succede con AWS EC2 scalable services.

protocol=: valore di default=””
Se valorizzato cambia per il gruppo il protocollo impostato sul listener.

redirectToHttps=: valore di default=”false”
Se true, in presenza di servizi a layer 7 HTTP/S verifica se la richiesta è in HTTP ed esegue un redirect per tutte le risorse interessate a HTTPS.

sniHostName=: valore di default=”value in endPointsGrouping:=sniHostName and virtualDomain:=sniHostName”
Nome dell'host utilizzato per le connessioni TLS SNI verso il backend.
Viene utilizzato per bilanciare servizi di backend in SSL che implementano il protocollo Server Name Indication. ATTENZIONE: Se viene impostato questo valore il parametri address nei paragrafi endp devono essere impostati con il valore numerico (ipv4 e ipv6 notation) e non con un nome host.

N.B.: In ogni caso se il nome dell'host proposto nell'HEADER proveniente dal client è

differente è necessario modificare il nomde dell'host nellheader attraverso una semplice regola di rewrite.

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=: valore di default=”false”
Se true permette di impostare a SNI degli endpoint e di esguire il forward automatico dell'hortname della richiesta verso il backend con lo stesso valore della provenienza.

S**SLApplicationProtocols**=: valore di default=””
In questo parametro si possono elencare i protocolli abilitati su comunicazioni ALPN. I valori ammessi “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=: valore di default=”true”
Se true utilizza i chipersuite nell'ordine indicato per i listener SSL/TLS

redirectTo=: valore di default=””
Con questo parametro è possibile istruire un redirect a fronte di una richiesta di servizio. Attraverso questo parametro infatti OPLON®LoadBalancer risponderà alla richiesta di servizio con un messaggio di ridirezione all'URL indicata nel valore. E' possibile inoltre

bilanciare i redirect in maniera ciclica introducendo piu' riferimenti di redirect per lo stesso gruppo di richieste alla stessa stregua degli end-point abituali. Nella URL è possibile inoltre indicare dei parametri o query string ricordandosi di scrivere le “&” nella notazione xml “&”.

port=: valore di default=”0”
E' la porta a cui risponde il servizio di backend

sslPort=: valore di default=”<endp port>”
E' la porta in cui attestare la connessione ssl se il listener e' in ssl. Come default viene caricato il parametro port.

SSLlistenerEval=: valore di default=”true”
Questo parametro, impostato a true per default, indica come discriminante di connessione ssl, il listener. Se il listener è ssl allora si prenderà la porta ssl altrimenti si prenderà la porta plain. Questo parametro viene valutato anche per instaurare una connessione ssl nel backend o meno.
Usato congiuntamente a SSL il risultato sarà:

CONDIZIONE (consueta):
Listener SSL && SSLlistenerEval==true && SSL=false
RISULTATO:
Connessione di frontend SSL (criptata)
Connessione di backend plain (in chiaro) su porta dichiarata come SSL
NOTA:
SSL Rewriting nei casi di ridirezione viene applicato automaticamente.
E' possibile disabilitarlo attraverso il parametro sslRewriting="false"

CONDIZIONE (consueta):
Listener SSL && SSLlistenerEval==true && SSL=true
RISULTATO:
Connessione di frontend SSL (criptata)
Connessione di backend in SSL su porta SSL

CONDIZIONE (non consueta):
Listener plain && SSLlistenerEval==false && SSL=true
RISULTATO:
Connessione di frontend plain (in charo)
Connessione di backend in SSL su porta SSL
NOTA:
Applicare rewriting su ridirezioni da https a http

CONDIZIONE (non consueta in questa forma):
Listener SSL && SSLlistenerEval==false && SSL=true
RISULTATO:
Connessione di frontend SSL (criptata)
Connessione di backend in SSL su porta SSL

SSL=: valore di default=”false”
Se true indica che se indirizzata la porta sslPort questa attuerà un traffico criptato SSL.

SSLCert=: valore di default=”virtualDomain:=SSLCert”
E' il riferimento al certificato digitale da utilizzarsi come “client certificate” da OPLON®LoadBalancer verso gli endpoints di questo dominio.

uriPath=: valore di default=””
Su layer 7 (HTTP/S 1.0/1.1 è il path che identifica il root path context del servizio.
Su layer 4 (TCP) questo valore deve essere necessariamente impostato a “” o non essere presente.

proxyHost=: valore di default=””
Host name/dominio del servizio esposto internamente
Se valorizzato l'host del servizio viene mascherato con il nome indicato nel virtualDomainName

proxyPort=: valore di default=””
Porta del servizio esposto internamente
Se valorizzato la porta del servizio viene mascherata con la porta presente nell'endpoint.

removePortFromHost=: valore di default=”false”
Se true, rimuove la porta dall'elemento header Host:

proxyUrl=: valore di default=””
URI / contesto del servizio
Permette di mascherare l' URI del servizio con il valore indicato nel parametro uriPath

proxyLocation=: valore di default=”true”
Riscrive il contenuto dell'header Location con coerentemente con i valori indicati nei parametri proxyHost e proxyUrl

proxyDestination=: valore di default=””
Riscrive il contenuto dell'header Destination con coerentemente con i valori indicati nei parametri proxyHost e proxyUrl

proxyCookie=: valore di default=”true”
Riscrive l'attributo path e domain dei cookie coerentemente con i valori indicati nei parametri proxyHost e proxyUrl

portRewriting=: valore di default=”<virtualDomain> portRewriting”
Se impostato a true riscrive il numero di porta nell'header proveniente dall'endpoint con il valore della porta richiesta dal client. Questo comportamento è necessario per bilanciare servizi di backend che rispondono modificando l'header originale della richiesta con il proprio port number. Con IIS questo parametro deve essere impostato a true.

sslRewriting=: valore di default=”<virtualDomain> sslRewriting”
Se impostato a true verifica se il listener e' in SSL ed eventualmente riscrive il protocollo da

http a https in caso di redirect comandate dal servizio aventi protocollo http. Questa funzionalità è utile nel caso il listener funge da terminatore ssl e nel backend vi sono dei servizi in chiaro.

uriPathMatcher=: valore di default=””
Su layer 7 (HTTP/S 1.0/1.1 è possibile utilizzare in alternativa al parametro "uriPath" il parametro "uriPathMatcher" con l'espressione regolare applicata all'URL della richiesta.

Una possibile applicazione e' ad esempio instradare le richieste di contenuti statici verso server dedicati a tali contenuti mentre le richieste a contenuti dinamici verranno instradate agli application server.

E' possibile gestire differenti politiche di sessione per i singoli endp non gestendo ad esempio la sessione per i contenuti statici e gestendo invece la sessione per i contenuti dinamici.

Di seguito alcuni esempi:

Es.

```
<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"/>
```

Es.

```
<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 questi due frammenti si può notare anche il parametro "uriPathContextForSession". Questo parametro serve in caso LBL debba gestire la sessione per attribuire il path di contesto non possibile altrimenti. Nel caso non venga gestita la sessione oppure la sessione venga gestita dal servizio questo valore può non essere valorizzato.

uriPathContextForSession=: valore di default=””
Parametro per gestire la sessione ed attribuire il path di contesto in presenza di uriPathMatcher.
Nel caso non venga gestita la sessione oppure la sessione venga gestita dal servizio questo valore può non essere valorizzato.

sequence=: valore di default=” ” (space)
OPLON®LoadBalancer determina automaticamente le topologie di instradamento attraverso un algoritmo di definizione dei percorsi. Normalmente la determinazione automatica dei percorsi è la migliore soluzione ma volendo intervenire manualmente è possibile attribuire

delle sequenze arbitrarie attraverso questo parametro.
E' possibile ispezionare il risultato delle gerarchie di percorso e delle sequenzeattraverso il log allo start o al reinit del servizio di bilanciamento e instradamento.

Di seguito un esempio abbreviato dell'output della generazione delle gerarchie di instradamento:

```
|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=: valore di default=”false”
In caso di siti paritetici distribuiti geograficamente è possibile indicare i più vicini endpoint relativamente allo stesso servizio.
Le richieste saranno instradate verso i siti con near=true e solo se tutti gli endpoint dichiarati near risultano non raggiungibili verranno allora selezionati gli endpoint non near, cioè far.

maxConcurrentConnections=: valore di default=”0”
Se > 0 verifica se viene superata la soglia delle connessioni concorrenti per questo endpoint.
Il sistema inoltre conteggia tutti i limiti per quel virtual domain e notifica anche se la somma delle connessioni per quel virtual domain viene superata.
Se in possesso di licenza DoS Attack Prevention al superamento della soglia di connessioni

per questo contesto (endpointsgrouping, domain, uripath) le connessioni vengono tagliate per non indurre fenomeni di overload nel backend.
Attenzione: questo parametro, se > 0, si attiva e reagisce solo se dosMaxConcurrentConnectionsReaction è impostato a true sul gruppo o sul dominio, altrimenti non viene preso in considerazione.

uriPathContext=: valore di default=”uriPath”
Su layer 7 (HTTP/S 1.0/1.1) è il path che identifica il root path context del servizio se viene utilizzato il rewriter con cambiamento del contesto. Nel caso particolare questo valore è utilizzato per valorizzare il path del cookie generato da LBL con il valore del nuovo contesto. Quindi questo valore verrà utilizzato solo se è stata generata una regola di cambiamento di contesto (es.: da /training a /papaia) e si genera la sessione da LBL. Questo valore in tutti gli altri casi non verrà mai utilizzato.

healthCheck=: valore di default=”true”
Abilita/disabilita la funzionalità di healtcheck dell'endpoint. Questa funzionalità è utile per alcuni protocolli che utilizzano delle porte variabili.

downtimeOutOfOrder=: valore di default=”0”
Il parametro downtimeOutOfOrder serve a mettere temporaneamente fuori servizio l'endpoint in caso di timeout udp.
IMPORTAMTE: Utilizzare il nuovo parametro con un healthCheck = "false". Se impostiamo healthCheck su "true", l'endpoint, dopo un timeout, non tornerà più online.
Esempio:
<endp SSL = "false" address = "192.168.56.208" healthCheck = "false" downtimeOutOfOrder = "40000" port = "4501" associateName = "VPN000" enable = "true" />

enable=: valore di default=”true”
Abilita o disabilita questo servizio

idSessionsManagerName=: valore di default=”default”
Imposta il tipo di riconoscimento della sessione a layer 7 HTTP/S.
Questo valore deve essere associato ad un paragrafo <idSessions>. Il valore sarà riportato sul gruppo di endpoint con stesso URIPath. Se stessi URIPath vengono assegnati erroneamente a più gruppi di riconoscimento sessione verrà visualizzato un errore e corretto con il nome della prima assegnazione. L'errore sarà disponibile sul log.

associateName=: valore di default=””
In questo parametro si possono associare dei nomi simbolici al gruppo di end-point nella forma “nome_simbolico_a nome_simbolico_b nome_simbolico_c “. Ogni nome simbolico deve essere separato da uno spazio da un altro nome simbolico. Il nome simbolico non può contenere spazi.
Per la trattazione dei nomi associativi fare riferimento a <params notificationDir=”” />

loadBalancingType=: valore di default=”Adaptative”
Imposta la politica di bilanciamento.
Valori possibili:

- RoundRobin
- Adaptative
- FailOver

cMessage=: valore di default=”virtualDomain: cMessage”
Nome del file in formato html contenente il messaggio di cortesia per il contesto al quale è associato.
Al momento del caricamento dei parametri, allo start del processo di bilanciamento, viene verificata l'effettiva esistenza del file nella directory:

- (LBL_HOME)/procsProfiles/A10_LBLGo/resources/html/messageName.html

Nel caso il file non sia esistente viene segnalato al momento dell'avvio e assegnato il valore di default:

- (LBL_HOME)/procsProfiles/A10_LBLGo/resources/html/messageNoEndPoint.html

Questo parametro deve essere omogeneo per uriPath. In caso contrario verrà comunque uniformato al valore del primo endPoint con lo stesso uriPath.

redirectNoHostsFound=: valore di default=”virtualDomain:redirectNoHostsFound”
Se Layer 7 HTTP/S in caso di hosts non disponibili per il raggruppamento “endPointsGrouping-virtualDomain-uriPath “ è possibile, in alternativa al messaggio di cortesia, comandare un redirect ad un altro URL.
Nel caso questo parametro non sia esistente o valorizzato a “” verrà utilizzato il messaggio di cortesia.

needClientCert=: valore di default=”virtualDomain:needClientCert” UM=boolean
Se Layer 7 HTTP/S questo valore impone a tutto il raggruppamento di domini ed endpoint l'utilizzo del certificato client per poter usufruire dei servizi associati.
Per differenza i domini e gli endpoint possono modificare tale valore.

needClientCertMessage=: valore di default=”virtualDomain:needClientCertMessage”
Nome del file in formato html contenente il messaggio di cortesia per il contesto al quale è associato.
Al momento del caricamento dei parametri, allo start del processo di bilanciamento, viene verificata l'effettiva esistenza del file nella directory:

- (LBL_HOME)/procsProfiles/A10_LBLGo/resources/html/messageName.html

Nel caso il file non sia esistente viene segnalato al momento dell'avvio e assegnato il valore di default:

- (LBL_HOME)/procsProfiles/A10_LBLGo/resources/html/messageNeedClientCert.html

cipherSuites=: valore di default=””
Il parametro imposta le ciphersuites da utilizzare per la comunicazione con il backend.

SSLProtocols=: valore di default=””
Il parametro imposta i protocolli SSL da utilizzare per la comunicazione con il backend.

realmLogin=: valore di default=””: valore di default=”realmlogin from virtualDomain”
Imposta il login basic authentication del servizio.
Questo login assieme alla password verranno consegnati ai servizi endp come entity HTTP attraverso la forma es.:

authorization: Basic dXNyMjp1c3Iy

realmPassword=: valore di default=”realmPassword from virtualDomain”
Imposta la password basic authentication del servizio.
Questa password assieme al login verranno consegnati ai servizi endp come entity HTTP attraverso la forma es.:

authorization: Basic dXNyMjp1c3Iy

rdPool=: default value="null"
Remote Desktop Pool è l'indirizzo della subnet utilizzato come identificatore del pool

rdTarget=: default value="null"
Remote Desktop Target Server è l'indirizzo reale di quel server che fa riferimento all'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

Il sistema è stato progettato per consentire l'accesso simultaneo sia a risorse in pool che a risorse dedicate a uno specifico utente oa gruppi di utenti. In pratica le due funzioni coesistono contemporaneamente dando la massima flessibilità.

rewriteHeaderRules=: valore di default=”endPointsGrouping + virtualDomain +”
Elenco dei nomi delle regole di rewriting dell'HEADER HTTP (layer 7 HTTP/S) da applicare all'endpoint.
E' possibile indicare più regole separate da uno o più spazi.
es.:

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

Le regole verranno applicate, se le condizioni lo permettono, in sequenza.

;**LAST**
Per ogni nome regola è possibile inoltre indicare il parametro LAST che nel caso la regola venga eseguita determina lo stop dell'applicazione delle restanti regole della sequenza.
Per indicare il parametro è sufficiente mettere dopo il nome il “;” e il nome del parametro.

es.

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

Tra il nome e il/i parametri non devono esserci spazi. In questo caso se la regola redirSSLloginWhenNoSSL viene applicata la regola proxyTo non verrà mai eseguita.

;**ALWAYS**
Il parametro ALWAYS indica che la regola viene sempre eseguita
Per indicare il parametro è sufficiente mettere dopo il nome il ";" e il nome del parametro.

es.

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

In questo caso la regola proxyTo viene eseguita a prescindere dall'esecuzione della regola redirSSLloginWhenNoSSL.

;**NOP**
Il parametro NOP indica che la regola non deve essere eseguita.

es.

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

In questo caso la regola redirSSLloginWhenNoSSL non viene eseguita. Il parametro NOP è utile per escludere l'esecuzione di regole generali.

rewriteBodyRules=: valore di default="endPointsGrouping + virtualDomain +"
Elenco dei nomi delle regole di rewriting del BODY HTTP (layer 7 HTTP/S) da applicare all'endpoint. E' possibile indicare più regole separate da uno o più spazi.
es.:

```
rewriteBodyRules="addTrademarkParam absoluteToRelative echoRewriteBody"
```

Le regole verranno applicate, se le condizioni lo permettono, in sequenza.

;**LAST**
Per ogni nome regola è possibile inoltre indicare il parametro LAST che nel caso la regola venga eseguita determina lo stop dell'applicazione delle restanti regole della sequenza
Per indicare il parametro è sufficiente mettere dopo il nome il ";" e il nome del parametro.
es.

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

Tra il nome e il/i parametri non devono esserci spazi. In questo caso se la regola addTrademarkParam viene applicata le regole absoluteToRelative echoRewriteBody non verranno mai eseguite.

;ALWAYS
Il parametro ALWAYS indica che la regola viene sempre eseguita.

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

In questo caso la regola echoRewriteBody viene eseguita a prescindere dall'esecuzione della regola addTrademarkParam.

;**NOP**
Il parametro NOP indica che la regola non deve essere eseguita.

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

In questo caso la regola absoluteToRelative non viene eseguita. Il parametro NOP è utile per escludere l'esecuzione di regole generali.

healthcheck<endpoints>

Nella versione Standard ed Enterprise di OPLON® ADC l'endpoint di seguito riportato deve essere incluso obbligatoriamente affinchè OPLON® ADC possa eseguire l'healthcheck dei nodi:

```
<!-- 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>
```

Esempio <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>
```

Esempi redirect

Esempio di configurazione con redirect su diverse 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"/>
        ...
```

NOTA-1: Il comando HTTP di "redirect" per sua natura non può memorizzare nessun cookie e per questo motivo,in questa particolare condizione, non e' possibile assegnare in automatico un identificatore di sessione. Il round-robin sugli end-point di redirection sarà quindi eseguito ogni volta che viene invocato il gruppo/uri specificato.

NOTA-2: LBL(tm)LoadBalancer non esegue nessun "health check" autonomo del punto di ridirezione perché potrebbe non essere raggiungibile, ad esempio un indirizzo esterno protetto dal firewall.
E' possibile utilizzare la directory "(LBL_HOME)\lib\notificationDir" e il tag "associateName" per notificare l'indisponibilità del servizio. A tale scopo può essere utilizzato un plugin esterno opportunamente adattato al caso.

NOTA-3: Se al parametro "redirectTo" vengono aggiunti anche i parametri "address" e "port" verrà eseguito l'health check all'indirizzo porta specificati come per gli altri end-point.

Esempio di configurazione con redirect su diverse URL e nome associativo per interazione esterna non raggiungibile (healthCheck non possibile):

```
<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"/>
        ...
```

Esempio di configurazione con redirect su diverse URL , nome associativo e address e port per health-check automatico della risorsa:

```
<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"/>
      ...
```

Esempio di configurazione con richiesta del certificato SSL su un contesto (URIPath) definito:

```
<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 questo caso è necessario avere a disposizione il certificato client solo per il contesto "/mySecureContext".

<sysobserver>

```
<serviceconf>
        <iproxy>
                <sysobserver>
```

OPLON® ADC è un sistema basato su servizi che comunicano tra loro attraverso code di messaggi. Questo paragrafo permette di associare un servizio ad un nome.
Infatti OPLON® ADC può essere virtualizzato e contenere nella stessa JVM più istanze indipendenti di bilanciatori.

<service>

```
<serviceconf>
        <iproxy>
                <sysobserver>
                        <service
```

name=: valore di default=””
E' il nome del servizio logico.

id=: valore di default=””
E' il nome del servizio dell'istanza virtualizzata.

7 OPLON®ADC healthcheck.xml

(LBL_HOME)/procsProfiles/A10_LBLGo/conf/healthcheck.xml

Questo file di parametri permette di verificare lo stato di attività dell'istanza.
Normalmente il non necessita di essere modificato a meno che la porta di default “5991” su localhost dovesse già essere occupata da qualche altro servizio.

La struttura del file healthcheck.xml è la seguente:

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

<serviceconf>

<healthcheck>

<params>

```
<serviceconf>
    <healthcheck>
        <params
```

Il paragrafo contiene i parametri di configurazione del servizio di healthcheck

address=: valore di default=”localhost”
Il valore deve essere corrispondente a localhost.
I nodi loadbalancer accedono a questo servizio passando attraverso lo strato di bilanciamento verificando contemporaneamente lo stato della rete e del servizio di bilanciamento stesso.

port=: valore di default=”5991”
E' la porta su cui risponde il servizio.

NOTA: Questo è l'unico parametro da modificare solo nel caso esista già un servizio attivo sulla stessa porta.

reuseAddress=: valore di default=”true”
E' il corrispondente parametro socket SO_REUSEADDR.

timeOut=: valore di default=”1500” UM=Millesimi di secondo
Il servizio healthCheck è un servizio HTTP1.0/1.1 e questo valore ne indica il timeout di connessione.

timeOutFactor=: valore di default=”300”
E' il fattore moltiplicativo del time out.

tcpNoDelay=: valore di default=”true”
Abilita/Disabilita l'algoritmo di Nagle per controllare la bufferizzazione dei dati

concurrentWorkers=: valore di default=”20”
E' il numero iniziale di workers per la risoluzione delle richieste di connessione.

maxConcurrentWorkers=: valore di default=”100”
E' il numero massimo di workers per la risoluzione delle richieste di connessione.

healthCheckContextPath=: valore di default=”/LBLHealthCheck”
E' il path di healthcheck di attività del sistema di bilanciamento.
Questo valore normalmente non viene mai modificato a meno già in uso in altre applicazioni. Se questo valore viene modificato è necessario modificarlo anche in “systemsmonitor_m.xml”, “iproxy.xml” e in “healthcheck.xml”.

webAppsDir=: valore di default=”lib/webroot_healtcheck/webapps”
Directory home applicazione web

webAppsConfDir=: valore di default=”lib/webroot_healtcheck/webappsconf”
Directory configurazione applicazioni web

webSecurityDir=: valore di default=”lib/webroot_healtcheck/websecurity”
Directory configurazione applicazioni web

<sysobserver>

```
<serviceconf>
        <healthcheck>
                <sysobserver>
```

OPLON® ADC è un sistema basato su servizi che comunicano tra loro attraverso code di messaggi. Questo paragrafo permette di associare un servizio ad un nome.

Infatti OPLON® ADC può essere virtualizzato e contenere nella stessa JVM più istanze indipendenti di bilanciatori.

<service>

```
<serviceconf>
        <healthcheck>
                <sysobserver>
                        <service
```

name=: valore di default=””
E' il nome del servizio logico.

id=: valore di default=””
E' il nome del servizio dell'istanza virtualizzata.

8 OPLON®ADC lookup.xml

(LBL_HOME)/procsProfiles/A10_LBLGo/conf/lookup.xml

Il servizio di lookup fornisce a OPLON® ADC la funzionalità di identificazione automatica del nodo.
Questa funzionalità viene fornita attraverso servizi di multicast propagati attraverso la rete privata.
La rete privata deve essere utilizzata in maniera esclusiva dai nodi OPLON® ADC.

```
<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=: valore di default=”multicast”
Valori possibili: multicast, udp
Definisce il protocollo di comunicazione dei nodi del cluster.
In caso di cluster geografici, o di reti che non supportano il protocollo multicast, si possono impostare i nodi del cluster ad utilizzare il protocollo udp.
In questo caso tutti i nodi del cluster devono essere dichiarati nel paragrafo <peersInstances>

Esempio:

```
<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=: valore di default=””
Quando impostato, le comunicazioni tra i nodi del cluster vengono crittografate ed il valore del parametro definisce la chiave di encryption, che deve essere impostata con lo stesso valore in tutti i nodi del cluster.

group=: valore di default=”LBLDistributionGroup”
E' il nome del gruppo di nodi.
Più gruppi di nodi possono coesistere all'interno della stessa rete privata. Questo nome è il discriminante per distinguere i messaggi provenienti dallo stesso gruppo di bilanciatori. I messaggi provenienti da altri altri gruppi verranno scartati. A seconda della distribuzione OPLON® il nome di default può assumere il valore LBLEnterpriseGroup oppure LBLStandardGroup.

subGroup=: valore di default=”LBLSubGroup”

E' il nome del sottogruppo dei nodi.
Questo parametro è normalmente utilizzato nella distribuzione Standard di OPLON®LoadBalancer per la gestione del Disaster Recovery.
Impostando il sottogruppo (subGroup) del nodo/i installato sul sito di DR con un nome diverso dal/i sottogruppi dei nodi del sito principale non vengono propagate le informazioni di instradamento delle sessioni da e verso i nodi con differenti sottogruppi. Infatti in una infrastruttura con gestione del DR il/i nodi posti nel sito di DR instradano le richieste di servizio verso i propri servizi di backend mentre devono mantenere comuni i servizi di Virtual IP (VIP) e quanto serve alla gestione del cluster.

groupWeight=: valore di default=”100”
E' il peso di questo bilanciatore.
Questo valore, nella versione Standard di OPLON®LoadBalancer determina l'assunzione a master di uno dei nodi. Se i nodi hanno lo stesso peso OPLON®LoadBalancer calcola dei numeri casuali e chi ottiene il risultato più alto prende l'iniziativa di diventare master.
Dalla versione 6.0 è possibile usare a scopi DR nodi Standard Edition ed Enterprise Edition per il take over degli indirizzi del sito principale.

interfaceAddress=: valore di default=””
E' l'indirizzo/nome host dell'interfaccia della rete privata

address==: valore di default=”228.5.6.7”
E' l'indirizzo multicast di ricetrasmissione

port=: valore di default=”6789”
E' la porta multicast di ricetrasmissione

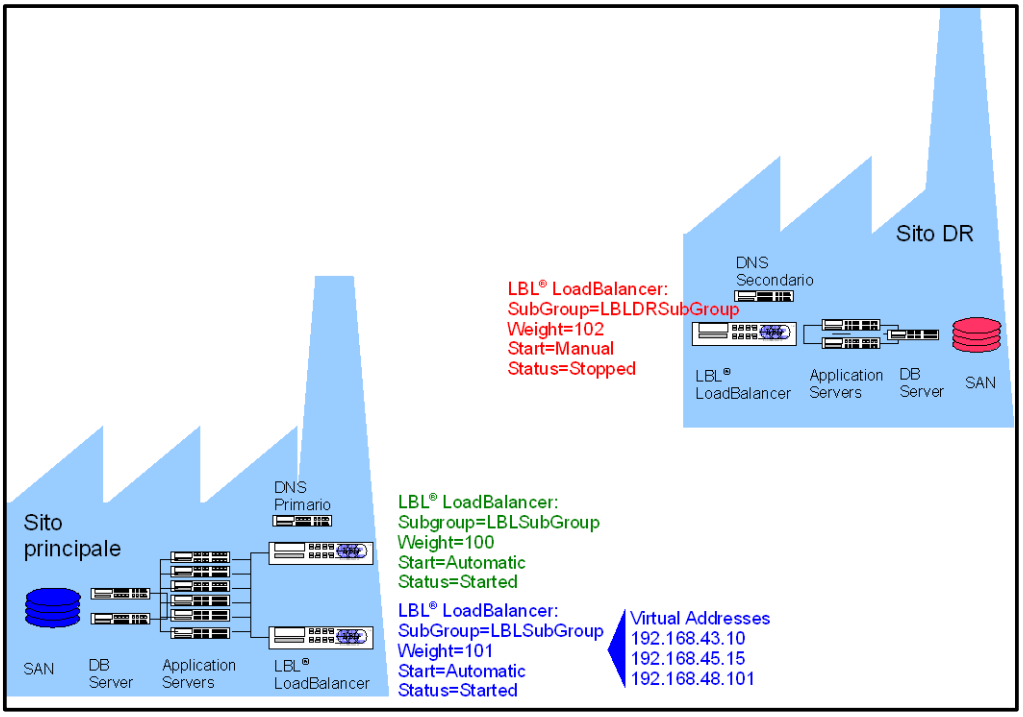
timeToLive=: valore di default=”1”
E' il tempo di vita del pacchetto multicast

timeOut=: valore di default=”-1” UM=Millisecondi
Timeout in millesimi di secondo di attesa di un pacchetto in ricezione.
Valore di default -1 (no time out)

datagramLength=: valore di default=”1024”

Lunghezza del datagramma

Esempio di utilizzo del sottogruppo in uno scenario di Business Continuity:



<peersInstances>

```
<serviceconf>
        <lookup>
            <peersInstances>
```

In questo paragrafo sono dichiarati tutti i nodi del cluster di bilanciamento nel caso in cui venga utilizzato il protocollo udp per la comunicazione tra i nodi (lookup).

<peer>

```
<serviceconf>
        <lookup>
            <peersInstances>
                <peer
```

Il tag peer descrive un nodo del cluster. Il tag è utilizzato nel caso il protocollo di lookup sia udp.

enable=: valore di default=”true”
Abilita o disabilita questo paragrafo.

description=: valore di default=””
E' descrizione del nodo a cui il paragrafo si riferisce

address==: valore di default=””
E' l'indirizzo del nodo a cui inviare i messaggi

port=: valore di default=””
E' la porta upd di ricezione

timeToLive=: valore di default=”1”
E' il tempo di vita del pacchetto.

timeOut=: valore di default=”-1” UM=Millisecondi
Timeout in millesimi di secondo di attesa di un pacchetto in ricezione.
Valore di default -1 (no time out)

<redundantLookupInterfaces>

```
<serviceconf>
        <lookup>
                <redundantLookupInterfaces>
```

Questo paragrafo opzionale contiene le interfacce ridondate private dove vengono attestati i servizi di lookup paralleli a quanto descritto sul paragrafo params. Si possono inserire in questo paragrafo più interfacce che lavorano contemporaneamente per poter eseguire un corretto routing multicast in caso di failure di una scheda di rete privata.

<interface>

```
<serviceconf>
        <lookup>
                <redundantLookupInterfaces>
                        <interface
```

enable=: valore di default=”true”
Abilita o disabilita questo paragrafo.

description=: valore di default=””
E' l'indirizzo/nome host dell'interfaccia della rete privata

interfaceAddress=: valore di default=””
E' l'indirizzo/nome host dell'interfaccia della rete privata

address==: valore di default=”228.5.6.7”
E' l'indirizzo multicast di ricetrasmissione

port=: valore di default=”6789”
E' la porta multicast di ricetrasmissione

timeToLive=: valore di default=”1”
E' il tempo di vita del pacchetto multicast

timeOut=: valore di default=”-1” UM=Millisecondi
Timeout in millesimi di secondo di attesa di un pacchetto in ricezione.
Valore di default -1 (no time out)

<peersInstances>

```
<serviceconf>
        <lookup>
            <peersInstances>
```

In questo paragrafo sono dichiariati tutti i nodi del cluster di bilanciamento nel caso in cui venga utilizzato il protocollo udp per la comunicazione tra i nodi (lookup).

<peer>

```
<serviceconf>
        <lookup>
            <peersInstances>
                <peer
```

Il tag peer descrive un nodo del cluster. Il tag è utilizzato nel caso il protocollo di lookup sia udp.

enable=: valore di default=”true”
Abilita o disabilita questo paragrafo.

description=: valore di default=””
E' descrizione del nodo a cui il paragrafo si riferisce

address==: valore di default=””
E' l'indirizzo del nodo a cui inviare i messaggi

port=: valore di default=””
E' la porta upd di ricezione

timeToLive=: valore di default=”1”
E' il tempo di vita del pacchetto.

timeOut=: valore di default=”-1” UM=Millisecondi
Timeout in millesimi di secondo di attesa di un pacchetto in ricezione.
Valore di default -1 (no time out)

Esempio redundantLookupInterfaces protocollo multicast

```
 <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>
```

Esempio redundantLookupInterfaces protocollo udp

```
<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>
```

9 OPLON®ADC vrrpserver.xml

(LBL_HOME)/procsProfiles/A10_LBLGo//conf/vrrpserver.xml

Il sistema Virtual Routing Redundancy Protocol permette ai Gateway di aggiornare automaticamente le informazioni necessarie al mantenimento delle sessioni e determinare lo stato di funzionamento durante il run-time.

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

<serviceconf>

<vrrpserver>

<params>

```
<serviceconf>
        <vrrpserver>
                <params
```

address=: valore di default=”localhost”
Il valore deve essere corrispondente ad un indirizzo o nome host afferente alla rete privata.

port=: valore di default=”5991”
E' la porta su cui risponde il servizio.

reuseAddress=: valore di default=”true”
E' il corrispondente parametro socket SO_REUSEADDR.

timeOut=: valore di default=”1500” UM=Millesimi di secondo
Il servizio vrrpserver è un servizio HTTP1.0/1.1 e questo valore ne indica il timeout di connessione.

timeOutFactor=: valore di default=”300”
E' il fattore moltiplicativo del time out.

tcpNoDelay=: valore di default=”true”
Abilita/Disabilita l'algoritmo di Nagle per controllare la bufferizzazione dei dati

concurrentWorkers=: valore di default=”20”
E' il numero iniziale di workers per la risoluzione delle richieste di connessione

maxConcurrentWorkers=: valore di default=”100”
E' il numero massimo di workers per la risoluzione delle richieste di connessione

webAppsDir=: valore di default=”lib/webroot_vrrp/webapps”
Directory home applicazione web

webAppsConfDir=: valore di default=”lib/webroot_vrrp/webappsconf”
Directory configurazione applicazioni web

webSecurityDir=: valore di default=”lib/webroot_vrrp/websecurity”
Directory configurazione applicazioni web

10 OPLON®ADC systemsmonitor_m.xml

(LBL_HOME)/procsProfiles/A10_LBLGo/conf/systemsmonitor_m.xml

Questo file parametri descrive i parametri per gestire gli indirizzi virtuali.
Dalla versione 7.0 è possibile indicare più indirizzi e più schede di rete per ottenere ridondanza hardware locale dell'indirizzo virtuale stesso.

```
<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=:valore di default=”systemsMonitorGroup”
Nome logico del sistema di virtualizzazione indirizzi.

notificationDir=:valore di default=”lib/notificationDir”
Directory di notifica eventi esterni.
Creando un nome file con il nome impostato nel systemsMonitorGroup con estensione “outOfOrder” è possibile far dichiarare tutto il servizio di bilanciamento fuori servizio.

es.:

```
# touch  outOfOrder.systemsMonitorGroup
```

monitorTimer=:valore di default=”6000” UM=Millisecondi
Temporizzazione di healthcheck in Millisecondi

systemLostTime=:valore di default=”monitorTimer*2” UM=Millisecondi
Tempo necessario per dichiarare un sistema OutOfOrder

lostTransactionTimeOut=:valore di default=”15000” UM=Millisecondi
Indica il tempo necessario per dichiarare perse le transazioni provenienti dall'HeartBeat non ancora pervenute e quindi disporre un riallineamento totale (BULK-LOAD SESSIONS).

dropTimeFederatedSystemOutOfOrder=:default=”86400000” UM=Millisecondi
Indica il tempo necessario per eliminare dalla tabella dei sistemi federati un sistema in OutOfOrder. Il valore di default è relativo a 1 giorno in millisecondi.

sysCommandTimeOut=:valore di default=”10000” UM=Millisecondi
Indica il tempo necessario per dichiarare in time-out un comando di sistema.
Se il comando oltrepassa questo limite viene eseguito un comando di abort e successivamente viene rilasciato il controllo all'applicazione.

sysCommandCheckRate=:valore di default=”300” UM=Millisecondi
E' la frequenza del controllo dello stato del comando di sistema.

sysCommandRemoteURL=:default=”https://localhost:5992/SysCommand”
E' l'URL del servizio per eseguire i comandi di sistema

createConnectionTimeOut=:default=”5000” UM=Millisecondi
E' il timeout impostato sui messaggi ICMP (ping) e connect TCP per verificare lo stato della rete.
Questo valore viene utilizzato anche come default se non impostato diversamente sui singoli punti di healthCheck (paragrafi: <publicNetworkHealthCheckPolicy> e

<backendNetworkHealthCheckPolicy>)

stayMasterAfterFailOverEvent=:default=”true”
Questo parametro permette di cambiare le condizioni di aumento della priorità durante le fasi di fail-over degli indirizzi virtuali. Impostato a “false” l'aumento della priorità avviene se l'indirizzo virtuale esiste e non presenta attività di bilanciamento (SND). Impostato a “true l'aumento di priorità avviene sia per la mancanza dell'indirizzo virtuale sia per la mancanza di attività di bilanciamento (OR). Questo parametro è stato studiato per le installazioni in CLOUD dove gli indirizzi virtuali possono non essere residenti sulle stesse istanze di LBL. Questo parametro impostato a “true” è normalmente utilizzato con l'indicazione dell'indirizzo virtuale “esterno” da verificare nelle fasi di fail-over. L'indirizzo virtuale esterno è caratterizzato dall'avere valorizzati a “” i parametri device=”” e 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>

Il paragrafo contiene la definizione dell'indirizzo virtuale

<virtualAddress>

```
<serviceconf>
    <systemsmonitor_m>
        <virtualAddressesMgr>
            <virtualAddress
```

enable=:valore di default=”true”
Abilitazione disabilitazione indirizzo virtuale

description=:valore di default=””
Descrive l'indirizzo virtuale

staticAddress=:valore di default=””
E' il l'indirizzo statico che serve da quorum per determinare la raggiungibilità della rete. Quando viene utilizzata la ridondanza hardware è necessario disporre di un indirizzo associato al device che si vuole utilizzare per verificare la raggiungibilità della rete. Questo parametro viene utilizzato come impostazione della scheda di rete, per verificare l'esistenza dell'indirizzo virtuale e quindi trarne le conclusioni.
Se non impostato non può essere utilizzata la caratteristica di fail over locale delle schede di rete.
Per IPv6 la rappresentazione deve essere effettuata tra parentesi quadrate [fdd4:3c3f:aaaa::99].

staticNetmask=:valore di default=”255.255.255.0”
E' il netmask in cifre (es.: 255.255.255.0) utilizzato durante il set dell'indirizzo descritto su <staticAddress>.
Se l'indirizzo è riferito al protocollo IPv6 il valore è determinato dalla precisione che si vuole ottenere. Es: per 64 l'impostazione dell'indirizzo sarà fdd4:3c3f:aaaa::99/64

address=:valore di default=””
E' il l'indirizzo virtuale in cifre (es.: 192.168.43.10) .
Per IPv6 la rappresentazione deve essere effettuata tra parentesi quadrate [fdd4:3c3f:aaaa::99].

netmask=:valore di default=”255.255.255.0”
E' il netmask in cifre (es.: 255.255.255.0) dell'indirizzo virtuale
Se l'indirizzo è riferito al protocollo IPv6 il valore è determinato dalla precisione che si vuole ottenere es: per 64 l'impostazione dell'indirizzo sarà fdd4:3c3f:aaaa::99/64

healthCheckPort=:valore di default=””
E' la porta su cui eseguire il test di healthCheck. Se “” l'health check non viene eseguito. Questo valore è molto importante perché determina lo stato di attività non solo dell’indirizzo IP ma anche del sistema di bilanciamento e instradamento.

healthCheckSSL=:valore di default=”false”
Se impostato a true esegue un healch check HTTP instaurando una connessione cifrata.

healthCheckUriPath=:valore di default=”/LBLHealthCheck”

E' il path di healthcheck di attività del sistema di bilanciamento. Questo valore normalmente non viene mai modificato a meno di uso già presente in altre applicazioni. Se questo valore viene modificato è necessario modificarlo anche in “systemsmonitor_m.xml”, “iproxy.xml” e in “healthcheck.xml”.

<virtualInterface>

```
<serviceconf>
        <systemsmonitor_m>
                <virtualAddressesMgr>
                        <virtualAddress>
                                <virtualInterface
```

Dalla versione 7.0 questo paragrafo può essere duplicato per gestire la ridondanza hardware delle interfacce di rete. In questo caso i parametri definiti con l'iniziale “static” (es.: staticDevice) indicano i parametri da utilizzarsi per l'indirizzo locale di quorum descritto nel precedente paragrafo.
Su MS Windows i parametri saranno uguali mentre per Unix/Linux saranno diversi perchè indicheranno 2 alias distinti per l'indirizzo statico a determinazione del quorum e un altro alias per l'indirizzo virtuale. (vedi esempi)

staticDevice=:valore di default=””
Parametro utilizzato per la ridondanza hardware delle schede di rete è l'identificatore dell'alias associato ad un device fisico per Unix/Linux (es.: e1000g4 / eth4) e del device fisico per MS Windows

```
es.: PCI\\VEN_10EC&amp;DEV_8139&amp;SUBSYS_813910EC&amp;REV_10\
\4&amp;2E98101C&amp;0&amp;10F0
```

Questo parametro non deve essere impostato nel caso non vi sia la ridondanza hardware delle schede di rete.

staticDeviceName=:valore di default=””
E' il nome del device.
Su MS Windows deve assumere il valore del nome visibile in Risorse di rete. Fare riferimento al manuale di installazione OPLON®LoadBalancer Standard&Enterprise Installation per maggiori dettagli di installazione.
Questo parametro non deve essere impostato nel caso non vi sia la ridondanza hardware delle schede di rete.

device=:valore di default=””
E' l'identificatore dell'alias associato ad un device fisico per Unix/Linux e del device fisico per MS Windows.
Per ottenere questo nome si utilizzeranno “devcon.exe” per MS Windows e “ip addr” per i sistemi Unix/Linux.
Fare riferimento al manuale di installazione OPLON® ADC Standard&Enterprise Installation per maggiori dettagli di installazione e impostazione. Se impostato a “” assieme a deviceName indica un indirizzo in gestione esterno all'installazione (es.: CLOUD).

deviceName=:valore di default=””
E' il nome del device.
Su MS Windows deve assumere il valore del nome visibile in Risorse di rete. Fare

riferimento al manuale di installazione OPLON® ADC Standard&Enterprise Installation per maggiori dettagli di installazione. Se impostato a “” assieme a device indica un indirizzo in gestione esterno all'installazione (es.: CLOUD).

Di seguito alcune possibili configurazioni:
MS Windows:

```
<virtualInterface device="PCI\\VEN_8086&amp;DEV_1019&amp;SUBSYS_80F71043&amp;REV_00\
\4&amp;3B3CB9B1&amp;0&amp;0818"
  deviceName="LBLPublic">
```

Linux**:**

```
<virtualInterface device="eth1:0"
                  deviceName="eth1:0">
```

Solaris/OpenSolaris (7,8,9,10)

```
<virtualInterface device="iprb2:1"
                   deviceName="iprb2:1">
```

ATTENZIONE MOLTEPLICI NETWORK CARDS VIRTUALI
Se si hanno a disposizione più network cards nella rete pubblica per ridondanza o per gestione degli indirizzi allungare i tempi di shutdown di OPLON®LoadBalancer per permettere al bilanciatore di disallocare gli indirizzi
Per effettuare questa parametrizzazione è sufficiente inserire il parametro -DLBL_DELAY_SWITCHOFF in fase di startup su A10_LBLGo.xml
es.: -DLBL_DELAY_SWITCHOFF=60

Il valore di default è 40”

Esempio di file di configurazione MS Windows con ridondanza hardware di rete

```
<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_813910EC&amp;REV_10\\4&amp;2E98101C&amp;0&amp;10F0"
                          staticDeviceName="LBLPublic"
                          device="PCI\\VEN_10EC&amp;DEV_8139&amp;SUBSYS_813910EC&amp;REV_10\\4&amp;2E98101C&amp;0&amp;10F0"
                          deviceName="LBLPublic">
        </virtualInterface>
        <virtualInterface staticDevice="PCI\\VEN_10EC&amp;DEV_8139&amp;SUBSYS_813910EC&amp;REV_10\\4&amp;2E98101C&amp;0&amp;18F0"
                          staticDeviceName="LBLBackendMonitor"
                          device="PCI\\VEN_10EC&amp;DEV_8139&amp;SUBSYS_813910EC&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>
```

Esempio di file di configurazione Solaris senza ridondanza hardware di rete

```
<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>
```

Esempio di file di configurazione Solaris con ridondanza hardware di rete

```
<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>

Questo paragrafo indica quali sono gli indirizzi di rete da utilizzare per verificare lo stato della rete pubblica.
Inserire almeno 3 indirizzi, il numero consigliato è 5 indirizzi. ATTENZIONE: se tutti gli indirizzi di health check non sono raggiungibili, il virtual IP viene tolto ed il sistema si pone

in out of order per lasciare libero il sistema gemello di poter rilevare l'indirizzo.

<publicNetwork>

```
<serviceconf>
    <systemsmonitor_m>
        <virtualAddressesMgr>
            <virtualAddress>
                <publicNetworkHealthCheckPolicy>
                    <publicNetwork
```

Questo paragrafo esplicita l'indirizzo di rete da utilizzare per verificare lo stato della rete pubblica

address=:valore di default=””
Indirizzo su cui effettuare i test di verifica rete pubblica

port=:valore di default=””
Se valorizzato con valore > 0 l'health check avverrà attraverso TCP/IP con una connessione e conseguente disconnessione. Se non valorizzato verrà eseguito un health check attraverso ICMP (ping).

description=:valore di default=””
Descrizione dell'indirizzo di verifica.

createConnectionTimeOut=:default=”5000” UM=Millisecondi
E' il timeout impostato sui messaggi ICMP (ping) Connect TCP per verificare lo stato della rete.
Se non impostato viene utilizzato come default il valore impostato al paragrafo <systemsmonitor_m> <params>

Di seguito una possibile configurazione:

```
<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>
```

ATTENZIONE: Se non vengono impostate delle regole di HealtHCheck il responso sarà sempre positivo.
Assicurarsi di aver impostato almeno 5 regole di health check per scheda fisica o per logica applicativa.

<backendNetworkHealthCheckPolicy>

Questo paragrafo indica quali sono gli indirizzi di rete da utilizzare per verificare lo stato della rete di backend.
Inserire almeno 3 indirizzi, il numero consigliato è 5 indirizzi. ATTENZIONE: se tutti gli indirizzi di health check non sono raggiungibili, il virtual IP viene tolto ed il sistema si pone in out of order per lasciare libero il sistema gemello di poter rilevare l'indirizzo.

<backendNetwork>

```
<serviceconf>
    <systemsmonitor_m>
        <virtualAddressesMgr>
            <virtualAddress>
                <backendNetworkHealthCheckPolicy>
                    <backendNetwork
```

Questo paragrafo esplicita l'indirizzo di rete da utilizzare per verificare lo stato della rete di backend.

address=:valore di default=””
Indirizzo su cui effettuare i test di verifica rete di backend

port=:valore di default=””
Se valorizzato con valore > 0 l'health check avverrà attraverso TCP/IP con una connessione e conseguente disconnessione. Se non valorizzato verrà eseguito un health check attraverso ICMP (ping).

description=:valore di default=””
Descrizione dell'indirizzo di verifica.

healthCheckTimeOut=:default=”4000” UM=Millisecondi
E' il timeout impostato sui messaggi ICMP (ping) per verificare lo stato della rete.
Se non impostato viene utilizzato come default il valore impostato al paragrafo <systemsmonitor_m> <params>

Di seguito una possibile configurazione:

```
<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>
```

ATTENZIONE: Se non vengono impostate delle regole di HealtHCheck il responso sarà sempre positivo.
Assicurarsi di aver impostato almeno 5 regole di health check per scheda fisica o per logica applicativa.

11 OPLON® syslog.xml

(LBL_HOME)/procsProfiles/XXX_procName/conf/syslog.xml

Questo file non è proposto di default nella directory (LBL_HOME)/procsProfiles/XXX_procName/conf infatti normalmente non deve essere modificato e viene quindi distribuito all'interno della libreria/programma LBLLoadBalancer.jar.
Per modificarlo è sufficiente copiarlo dall'interno della libreria nella directory (LBL_HOME)/procsPriles/XXX_procName/conf. Oplon®Secure Access lo acquisirà e non prenderà più in considerazione l'originale all'interno della libreria.

```
<serviceconf>
   <copyright>
   </copyright>
        <syslog>
                <params>
                </params>
        </syslog>
</serviceconf>
```

<syslog>

<params>

```
<serviceconf>
        <syslog>
                <params
```

logDir=:valore di default=”logs”
E' la directory dove verranno creati e utilizzati i files di log. Questo valore può assumere sia valori relativi al processo in esecuzione che assoluti a livello di filesystem.

logFileSuffix=:valore di default=”LBLLog.txt”
E' il completamento del nome del file di log es.: YYYYMMDD_hostname_LBLLog.txt

logRotationHistoryDays=:valore di default=”15” UM=Giorni
Indica il tempo di ritenzione dei file di log. Oltrepassato il numero dei giorni indicati il file

viene cancellato.

logMessageDeduplication=:valore di default=”20” UM=numero di messaggi uguali
Dalla versione 7.0 e' stata introdotta la deduplica dei messaggi di log. A parità di messaggio consecutivo non viene eseguito immediatamente il log. Il log viene eseguito solo se arriva temporalmente un messaggio diverso oppure se si supera il numero di messaggi uguali indicati da questo parametro. A tal fine è stato aggiunto un nuovo campo alla fine del messaggio che indica il numero di messaggi ripetuti prima di quell'output.
es.:

```
 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
```

12 OPLON® statisticbrokercache.xml

Questo file descrive il servizio preposto a collezionare le statistiche di 1° e 2° livello.
Le statistiche di 1° livello vengono collezionate in memoria attraverso un servizio asincrono rispetto i servizi.
Con una temporizzazione, definibile in un parametro, la cache di 1° livello viene portata nella memoria di massa permanente definita cache di 2° livello. Con un altro servizio la cache di 2° livello viene spedita ad un processo completamente separato che ne colleziona, in maniera transazionale, nuovamente i valori in un area di memoria di massa permanente pronta per la storicizzazione su DataBase Relazionale. Il file descrive quindi con i parametri il servizio di caching delle statistiche di 1° e 2° livello.

```
<serviceconf>
   <copyright>
   </copyright>
        <statisticbrokercache>
                <params>
                </params>
        </statisticbrokercache>
</serviceconf>
```

<statisticbrokercache>

<params>

```
<serviceconf>
        <statisticbrokercache>
                <params
```

frequency=:valore di default=”10000” UM=Millisecondi
E' la frequenza di aggiornamento delle statistiche e passaggio da una cache ad un'altra.

maxFilesCacheHistory=:valore di default=”8640” UM=Numero di files
E' il numero di files massimo di file di cache oltre il quale viene cancellato il più vecchio. Normalmente la cache è popolata con 1 file e solo nel caso non sia disponibile il servizio di cache di 3° livello la directory trattiene i files fino al raggiungimento di questo numero che equivale a 1 giorno di lavoro pieno diviso per il parametro frequency. In altre parole se la frequenza di svuotamento della cache di 1° livello in memoria è di 10” e per 24 ore c'è stato

traffico di rete il numero risultante di file sarà 8640. Normalmente quindi verranno spediti 8640 file verso la cache di 3° livello in un giorno di 24 ore in cui non c'è stata soluzione di continuità.

dateFormat=:default=”dd/MM/yyyy HH:mm:ss:SSSS”
E' il formato della data dei dati statistici provenienti dal Statistic Broker Web Cache

delimiter=:default=”|” (pipe)
E' il carattere che delimita i campi durante lo scambio di informazioni.

statisticCacheHistoryDir=:valore di default=”lib/statisticCacheHistory”
E' la directory della cache di 2° livello..

remoteServerURL=:default=”http://localhost:5993/updateStatistic”
URL di upload statistiche per spostamento su cache di 3° livello. La cancellazione della cache di 2° livello avviene solo ad avvenuta scrittura definitiva della cache di 3° livello nel servizio che conferma con apposito messaggio.

SSL= default=”false”
Indica se il servizio di upload statistiche è in SSL.

description= default=”LBL(r)LoadBalancer statistic web cache manager”
Indica la descrizione del client HTTP per la spedizione delle statistiche verso il servizio web cache di 3° livello.

13 OPLON® statisticbrokerwebcache.xml

Questo file descrive il servizio preposto a collezionare le statistiche di 3° e 4° livello.
Le statistiche di 1° livello vengono collezionate in memoria attraverso un servizio asincrono rispetto i servizi di bilanciamento. Con una temporizzazione, definibile in un parametro, la cache di 1° livello viene portata su memoria di massa permanente definita cache di 2° livello. Con un altro servizio la cache di 2° livello viene spedita ad un processo completamente separato che ne colleziona, in maniera transazionale, nuovamente i valori in un area di memoria di massa permanente, 3° livello, pronta per la storicizzazione su DataBase Relazionale, 4° livello.
Il file descrive quindi con i parametri il servizio di caching delle statistiche di 3° e 4° livello.

```
<serviceconf>
   <copyright>
   </copyright>
        <statisticbrokerwebcache>
                <params>
                </params>
        </statisticbrokerwebcache>
</serviceconf>
```

<statisticbrokerwebcache>

<params>

```
<serviceconf>
        <statisticbrokerwebcache>
                <params
```

address=: valore di default=”localhost”
Il valore deve essere corrispondente a localhost. Infatti gli altri loadbalancer accedono a questo servizio passando attraverso lo strato di bilanciamento verificando contemporaneamente lo stato della rete e del servizio di bilanciamento stesso.

port=: valore di default=”5993”
E' la porta su cui risponde il servizio. Questo è l'unico parametro da modificare nel caso esista già un servizio attivo sulla stessa porta.

timeOut=: valore di default=”1500” UM=Millesimi di secondo
Il servizio healthCheck è un servizio HTTP1.0/1.1 e questo valore ne indica il timeout di connessione.

timeOutFactor=: valore di default=”300”
E' il fattore moltiplicativo del time out.

tcpNoDelay=: valore di default=”true”
Abilita/Disabilita l'algoritmo di Nagle per controllare la bufferizzazione dei dati

concurrentWorkers=: valore di default=”20”
E' il numero iniziale di workers per la risoluzione delle richieste di connessione.

maxConcurrentWorkers=: default=”100”
E' il numero massimo di workers per la risoluzione delle richieste di connessione.

contextPath=: default=”/updateStatistic”
E' il path di healthcheck di attività. Questo valore normalmente non viene mai modificato a meno di uso già presente in altre applicazioni. Se questo valore viene modificato è necessario modificarlo anche in “systemsmonitor_m.xml”.

webAppsDir=: default=”lib/webroot_statisticbrokerweb/webapps”
Directory home applicazione web

webAppsConfDir=: default=”lib/webroot_statisticbrokerweb/webappsconf”
Directory configurazione applicazioni web

webSecurityDir=: valore di default=”lib/webroot_healtcheck/websecurity”
Directory configurazione applicazioni web

frequency=:valore di default=”10000” UM=Millisecondi
E' la frequenza di aggiornamento delle statistiche

statisticBrokerCacheFrequency=:valore di default=”10000” UM=Millisecondi
E' la frequenza di svuotamento della cache per allineamento temporale con il processo di creazione cache di 1° e 2° livello.

timeLimitObsoleteStatisticSnap=:valore di default=”60000” UM=Millisecondi
E' il limite temporale per cui il log all'interno del database non viene più considerato come attuale per una visualizzazione istantanea.

timeLimitObsoleteStatisticSnapSessions=:valore di default=”420000” UM=Millisecondi
E' il limite temporale per cui il log all'interno del database non viene più considerato come attuale per una visualizzazione istantanea della statistica sessioni.

maxFilesCacheHistory=:valore di default=”8640” UM=Numero di files
E' il numero di files massimo di file di cache oltre il quale viene cancellato il più vecchio. Normalmente la cache è popolata con 1 file e solo nel caso non sia disponibile il servizio di

persistenza del Database la directory trattiene i files fino al raggiungimento di questo numero che equivale a 1 giorno di lavoro pieno diviso per il parametro frequency. In altre parole se la frequenza di svuotamento della cache di 1° livello in memoria è di 10” e per 24 ore c'è stato traffico di rete il numero risultante di file sarà 8640. Normalmente quindi verranno storicizzati 8640 file sul database in un giorno di 24 ore in cui non c'è stata soluzione di continuità.

statisticCacheHistoryDir=:valore di default=”lib/statisticWebCacheHistory”
E' la directory della cache di 3° livello..

dateFormat=:default=”dd/MM/yyyy HH:mm:ss:SSSS”
E' il formato della data dei dati statistici provenienti dal Statistic Broker Cache (processo di bilanciamento)

delimiter=:default=”|” (pipe)
E' il carattere che delimita i campi durante lo scambio di informazioni.

maxEmbeddedDBSize=:default=”5368709120” UM=byte
Nella configurazione con DB Embedded (JavaDB embedded) questo parametro indica la dimensione massima del database sull'unità di memoria di massa. Al superamento di questa soglia viene segnalato attraverso il log il problema e quindi il DB viene ricreato. Questa misura è per prevenire l'insorgenza di esaurimento delle risorse di sistema del nodo che contiene anche il processo di bilanciamento di carico. La considerazione da cui è stata tratta questa funzionalità è che essendo un DB embedded, cioè senza la possibilità di interrogazione delle statistiche dall'esterno, i dati contenuti sono da considerarsi puramente statistici e quindi non critici. Questo parametro non viene preso in considerazione in tutti gli altri casi di storicizzazione delle statistiche su DB networked.

DBInitialConnectionPool=:default=”5”
Numero iniziale di connessioni al database.

DBMaxConnectionPool=:default=”5”
Numero massimo di connessioni al database. Superato questo limite verra' comunque aperta una nuova connessione che cesserà al termine della richiesta e non verrà riutilizzata dal connection pool.

DBDriver=:default=”org.apache.derby.jdbc.EmbeddedDriver”
E' il driver per gestire il Database Relazionale.

DBProtocol=:default=”jdbc:derby:”
E' il protocollo della stringa di connessione al database

DBName=:default=”lib/LBLDBStatistics”
E' il nome del database. Assieme al DBProtocol determina la stringa di connessione al DB. Un altro esempio di DBName potrebbe essere “//trantor:1527/S:\javaDB\LBLDBStatistics” che identifica un servizio DB Networked sul sistema trantor alla porta 1527.

DBLogin=:default=”null”

E' il login per la connessione al DB. Il valore null indica che non è richiesta autenticazione. OPLON® ADC per default assume come embedded il DB non dando accesso dall'esterno ai listener del DB che possano eseguire SQL Script. Ponendo il DB come networked e cambiando i parametri è possibile indicare in questo parametro il login per effettuare la connessione.

DBPassword=:default=”null”
E' la password per la connessione al DB. Il valore null indica che non è richiesta autenticazione. OPLON® ADC per default assume come embedded il DB non dando accesso dall'esterno ai listener del DB che possano eseguire SQL Script. Ponendo il DB come networked e cambiando i parametri è possibile indicare in questo parametro il login per effettuare la connessione.

DBSchemaName=:default per JavaDB=”APP.” default altri DB=””
E' il nome dello schema che ha generato le tabelle. Per JavaDB (aka derby, cloudscape) il default è “APP.” e viene anteposto a tutte le query. Per gli altri DB il default è “” (vuoto) e quindi il risultato nel nome della tabella sulla query riporta solo il nome

DBDateFormat=:default=”yyyy-MM-dd”
E' il formato della data che esprime i valori giorno mese anno utilizzata nel database

DBTimeFormat=:default=”HH:mm:ss”
E' il formato della data che esprime i valori ore minuti secondi utilizzata nel database

DBSetDateFormat=:default=”” UM=Comando SQL
In alcune circostanze è necessario eseguire un comando SQL per impostare il formato date/time. Questo parametro se impostato viene eseguito nel contesto di sessione del DataBase per determinare le modalità di accesso a questi dati. Sui database rilasciati all'utilizzo questo parametro è utilizzato solo nel caso di Oracle DB ed il suo valore deve essere:

```
ALTER SESSION set NLS_DATE_FORMAT = &amp;apos;YYYY-MM-DD HH24:MI:SS&amp;apos;
```

DBMaxHistoryDays=:default=”2” UM=Giorni
Per valori maggiori di 0 è il numero massimo di giorni storicizzati nel Database. Al cambio di data verranno cancellati i log del traffico avvenuto precedentemente il numero di giorni impostati rispetto la nuova data. Se il valore è minore a 0 (zero) le statistiche non verranno mai cancellate per tutte le tabelle indipendentemente dai valori impostati nei parametri specifici delle tabelle.

DBMaxHistoryDaysSessionActivity=:default=”DBMaxHistoryDays” UM=Giorni
Per valori maggiori di 0 è il numero massimo di giorni storicizzati nel Database tabella SESSION_ACTIVITY. Al cambio di data verranno cancellati i log del traffico avvenuto precedentemente il numero di giorni impostati rispetto la nuova data. Se il valore è minore a 0 (zero) le statistiche non verranno mai cancellate.

DBMaxHistoryDaysPoolQueuesActivity=:default=”DBMaxHistoryDays” UM=Giorni
Per valori maggiori di 0 è il numero massimo di giorni storicizzati nel Database tabella POOL_QUEUES_ACTIVITY. Al cambio di data verranno cancellati i log del traffico

avvenuto precedentemente il numero di giorni impostati rispetto la nuova data. Se il valore è minore a 0 (zero) le statistiche non verranno mai cancellate.

DBMaxHistoryDaysIncomingQueueHighWater=:default=”DBMaxHistoryDays” UM=Giorni
Per valori maggiori di 0 è il numero massimo di giorni storicizzati nel Database tabella INCOMING_QUEUE_HIGHWATER_LEVEL. Al cambio di data verranno cancellati i log del traffico avvenuto precedentemente il numero di giorni impostati rispetto la nuova data. Se il valore è minore a 0 (zero) le statistiche non verranno mai cancellate.

DBMaxHistoryDaysL4TcpTcpSSL=:default=”DBMaxHistoryDays” UM=Giorni
Per valori maggiori di 0 è il numero massimo di giorni storicizzati nel Database tabella L4_TCP_TCPSSL. Al cambio di data verranno cancellati i log del traffico avvenuto precedentemente il numero di giorni impostati rispetto la nuova data. Se il valore è minore a 0 (zero) le statistiche non verranno mai cancellate.

DBMaxHistoryDaysL7HttpHttps=:default=”DBMaxHistoryDays” UM=Giorni
Per valori maggiori di 0 è il numero massimo di giorni storicizzati nel Database tabella L7_HTTP_HTTPS. Al cambio di data verranno cancellati i log del traffico avvenuto precedentemente il numero di giorni impostati rispetto la nuova data. Se il valore è minore a 0 (zero) le statistiche non verranno mai cancellate.

DBMaxHistoryDaysL4Datagram=:default=”DBMaxHistoryDays” UM=Giorni
Per valori maggiori di 0 è il numero massimo di giorni storicizzati nel Database tabella L4_DATAGRAM. Al cambio di data verranno cancellati i log del traffico avvenuto precedentemente il numero di giorni impostati rispetto la nuova data. Se il valore è minore a 0 (zero) le statistiche non verranno mai cancellate.

DBLoadHistorySessionActivity=:default=”true” UM=boolean
Se impostato a false il database non verrà popolato con i valori corrispondenti.

DBLoadHistoryPoolQueuesActivity=:default=”true” UM=boolean
Se impostato a false il database non verrà popolato con i valori corrispondenti.

DBLoadHistoryIncomingQueueHighWater=:default=”true” UM=boolean
Se impostato a false il database non verrà popolato con i valori corrispondenti.

DBLoadHistoryL4TcpTcpSSL=:default=”true” UM=boolean
Se impostato a false il database non verrà popolato con i valori corrispondenti.

DBLoadHistoryL7HttpHttps=:default=”true” UM=boolean
Se impostato a false il database non verrà popolato con i valori corrispondenti.

DBLoadHistoryL4Datagram=:default=”true” UM=boolean
Se impostato a false il database non verrà popolato con i valori corrispondenti.

DBLoadHistorySyslogEvent=:default=”true” UM=boolean

Se impostato a false il database non verrà popolato con i valori corrispondenti.

DBQueryLimit=:default=”5000” UM=Records
E' il numero massimo di records estraibili in una query per evitare overflow.

DBVarcharLimit=:default=”4000” UM=Byte
E' il numero massimo di bytes che verranno inseriti nelle colonne di tipo VARCHAR. Vedere manuale LBL_DBNetworkedConfiguration.pdf per maggiori dettagli su questo valore nelle diverse piattaforme DB.

DBMaxScriptBatch=:default=”200” UM=Byte
E' il numero massimo di script SQL utilizzati nel caricamento batch del database. Alcuni database supportano un numero limitato di batchscript. Alla data JavaDB (aka derby, cloudscape) supporta al massimo 65534 batch script. Fare riferimento al manuale del produttore del database per modificare questo valore.

cookieValueToExtract=:default=”null” UM=Byte
In the L7_HTTP_HTTPS you can create a new column DUMMY varchar(300) to get a cookie value from a cookie in column COOKIES. To obtain the cookie value extraction is simple. In the parameter file statisticbrokerwebcache.xml put a parameter cookieValueToExtract with name of value's 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 set the parameter, run the following script with the new column DUMMY 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);
```

14 OPLON®Traffic Monetizer statisticbrokerwebcachedwh.xml

Questo file descrive il servizio preposto a collezionare le statistiche di 3° e 4° livello im modalita avanzata.
Le statistiche di 1° livello vengono collezionate in memoria attraverso un servizio asincrono rispetto i servizi di bilanciamento. Con una temporizzazione, definibile in un parametro, la cache di 1° livello viene portata su memoria di massa permanente definita cache di 2° livello. Con un altro servizio la cache di 2° livello viene spedita ad un processo completamente separato che ne colleziona, in maniera transazionale, nuovamente i valori in un area di memoria di massa permanente, 3° livello, pronta per la storicizzazione su DataBase Relazionale, 4° livello.
Il file descrive quindi con i parametri il servizio di caching delle statistiche di 3° e 4° livello.

```
<serviceconf>
   <copyright>
   </copyright>
        <statisticbrokerwebcachedwh>
                <params>
                </params>
        </statisticbrokerwebcachedwh>
</serviceconf>
```

<statisticbrokerwebcachedwh>

<params>

```
<serviceconf>
        <statisticbrokerwebcache>
                <params
```

address=: valore di default=”localhost”
Il valore deve essere corrispondente a localhost. Infatti gli altri loadbalancer accedono a questo servizio passando attraverso lo strato di bilanciamento verificando contemporaneamente lo stato della rete e del servizio di bilanciamento stesso.

port=: valore di default=”5993”
E' la porta su cui risponde il servizio. Questo è l'unico parametro da modificare nel caso esista già un servizio attivo sulla stessa porta.

timeOut=: valore di default=”1500” UM=Millesimi di secondo
Il servizio healthCheck è un servizio HTTP1.0/1.1 e questo valore ne indica il timeout di connessione.

timeOutFactor=: valore di default=”300”
E' il fattore moltiplicativo del time out.

tcpNoDelay=: valore di default=”true”
Abilita/Disabilita l'algoritmo di Nagle per controllare la bufferizzazione dei dati

concurrentWorkers=: valore di default=”20”
E' il numero iniziale di workers per la risoluzione delle richieste di connessione.

maxConcurrentWorkers=: default=”100”
E' il numero massimo di workers per la risoluzione delle richieste di connessione.

contextPath=: default=”/updateStatistic”
E' il path di healthcheck di attività. Questo valore normalmente non viene mai modificato a meno di uso già presente in altre applicazioni. Se questo valore viene modificato è necessario modificarlo anche in “systemsmonitor_m.xml”.

webAppsDir=: default=”lib/webroot_statisticbrokerweb/webapps”
Directory home applicazione web

webAppsConfDir=: default=”lib/webroot_statisticbrokerweb/webappsconf”
Directory configurazione applicazioni web

webSecurityDir=: valore di default=”lib/webroot_healtcheck/websecurity”
Directory configurazione applicazioni web

frequency=:valore di default=”10000” UM=Millisecondi
E' la frequenza di aggiornamento delle statistiche

statisticBrokerCacheFrequency=:valore di default=”10000” UM=Millisecondi
E' la frequenza di svuotamento della cache per allineamento temporale con il processo di creazione cache di 1° e 2° livello.

timeLimitObsoleteStatisticSnap=:valore di default=”60000” UM=Millisecondi
E' il limite temporale per cui il log all'interno del database non viene più considerato come attuale per una visualizzazione istantanea.

timeLimitObsoleteStatisticSnapSessions=:valore di default=”420000” UM=Millisecondi
E' il limite temporale per cui il log all'interno del database non viene più considerato come attuale per una visualizzazione istantanea della statistica sessioni.

maxFilesCacheHistory=:valore di default=”8640” UM=Numero di files
E' il numero di files massimo di file di cache oltre il quale viene cancellato il più vecchio. Normalmente la cache è popolata con 1 file e solo nel caso non sia disponibile il servizio di persistenza del Database la directory trattiene i files fino al raggiungimento di questo numero che equivale a 1 giorno di lavoro pieno diviso per il parametro frequency. In altre parole se la frequenza di svuotamento della cache di 1° livello in memoria è di 10” e per 24 ore c'è stato traffico di rete il numero risultante di file sarà 8640. Normalmente quindi verranno storicizzati 8640 file sul database in un giorno di 24 ore in cui non c'è stata soluzione di continuità.

statisticCacheHistoryDir=:valore di default=”lib/statisticWebCacheHistory”
E' la directory della cache di 3° livello..

dateFormat=:default=”dd/MM/yyyy HH:mm:ss:SSSS”
E' il formato della data dei dati statistici provenienti dal Statistic Broker Cache (processo di bilanciamento)

delimiter=:default=”|” (pipe)
E' il carattere che delimita i campi durante lo scambio di informazioni.

maxEmbeddedDBSize=:default=”5368709120” UM=byte
Nella configurazione con DB Embedded (JavaDB embedded) questo parametro indica la dimensione massima del database sull'unità di memoria di massa. Al superamento di questa soglia viene segnalato attraverso il log il problema e quindi il DB viene ricreato. Questa misura è per prevenire l'insorgenza di esaurimento delle risorse di sistema del nodo che contiene anche il processo di bilanciamento di carico. La considerazione da cui è stata tratta questa funzionalità è che essendo un DB embedded, cioè senza la possibilità di interrogazione delle statistiche dall'esterno, i dati contenuti sono da considerarsi puramente statistici e quindi non critici. Questo parametro non viene preso in considerazione in tutti gli altri casi di storicizzazione delle statistiche su DB networked.

DBLoaderInterceptorClassPath=:default=”interceptors/”
E' il path di caricamento della classe interceptor definita in DBLoaderInterceptorClass. Questo path viene aggiunto al classpath della JVM in esecuzione.

DBLoaderInterceptorClass=:default=”loadbalancer.services.statisticbroker.webcachedwh.LBLDBStatisticBrokerBatchStatementsInterceptor”
E' la classe di caricamento del database. Le distribuzioni contengono già una classe template disponibile in

- (LBL_HOME)/interceptors/dwhInterceptors/ LBLDBStatisticBrokerBatchStatementsInterceptorTemplate.java

Le classi contenute in questa directory possono essere compilate attraverso i tool: compile.bat o compile.sh presenti nella directory stessa.

DBInitialConnectionPool=:default=”1”
Numero iniziale di connessioni al database.

DBMaxConnectionPool=:default=”1”
Numero massimo di connessioni al database. Superato questo limite verra' comunque aperta una nuova connessione che cesserà al termine della richiesta e non verrà riutilizzata dal connection pool.

DBDriver=:default=”oracle.jdbc.driver.OracleDriver”
E' il driver per gestire il Database Relazionale.

DBProtocol=:default=”jdbc:oracle:”
E' il protocollo della stringa di connessione al database

DBName=:default=””
E' il nome del database. Assieme al DBProtocol determina la stringa di connessione al DB.

DBLogin=:default=”null”
E' il login per la connessione al DB.

DBPassword=:default=”null”
E' la password per la connessione al DB.

DBDateFormat=:default=”yyyy-MM-dd”
E' il formato della data che esprime i valori giorno mese anno utilizzata nel database

DBTimeFormat=:default=”HH:mm:ss”
E' il formato della data che esprime i valori ore minuti secondi utilizzata nel database

DBSetDateFormat=:default=”” UM=Comando SQL
Comando SQL per impostare il formato date/time. Questo parametro quando impostato viene eseguito nel contesto di sessione del DataBase. Nel caso di Oracle DB il suo valore deve essere:

```
ALTER SESSION set NLS_DATE_FORMAT = &apos;YYYY-MM-DD HH24:MI:SS&apos;
```

DBMaxHistoryDays=:default=”-1” UM=Giorni
Per valori maggiori di 0 è il numero massimo di giorni storicizzati nel Database. Al cambio di data verranno cancellati i log del traffico avvenuto precedentemente il numero di giorni impostati rispetto la nuova data. Se il valore è minore a 0 (zero) le statistiche non verranno mai cancellate per tutte le tabelle indipendentemente dai valori impostati nei parametri specifici delle tabelle.

DBMaxHistoryDaysSessionActivity=:default=”DBMaxHistoryDays” UM=Giorni
Per valori maggiori di 0 è il numero massimo di giorni storicizzati nel Database tabella SESSION_ACTIVITY. Al cambio di data verranno cancellati i log del traffico avvenuto precedentemente il numero di giorni impostati rispetto la nuova data. Se il valore è minore a 0 (zero) le statistiche non verranno mai cancellate.

DBMaxHistoryDaysPoolQueuesActivity=:default=”DBMaxHistoryDays” UM=Giorni
Per valori maggiori di 0 è il numero massimo di giorni storicizzati nel Database tabella POOL_QUEUES_ACTIVITY. Al cambio di data verranno cancellati i log del traffico

avvenuto precedentemente il numero di giorni impostati rispetto la nuova data. Se il valore è minore a 0 (zero) le statistiche non verranno mai cancellate.

DBMaxHistoryDaysIncomingQueueHighWater=:default=”DBMaxHistoryDays” UM=Giorni
Per valori maggiori di 0 è il numero massimo di giorni storicizzati nel Database tabella INCOMING_QUEUE_HIGHWATER_LEVEL. Al cambio di data verranno cancellati i log del traffico avvenuto precedentemente il numero di giorni impostati rispetto la nuova data. Se il valore è minore a 0 (zero) le statistiche non verranno mai cancellate.

DBMaxHistoryDaysL4TcpTcpSSL=:default=”DBMaxHistoryDays” UM=Giorni
Per valori maggiori di 0 è il numero massimo di giorni storicizzati nel Database tabella L4_TCP_TCPSSL. Al cambio di data verranno cancellati i log del traffico avvenuto precedentemente il numero di giorni impostati rispetto la nuova data. Se il valore è minore a 0 (zero) le statistiche non verranno mai cancellate.

DBMaxHistoryDaysL7HttpHttps=:default=”DBMaxHistoryDays” UM=Giorni
Per valori maggiori di 0 è il numero massimo di giorni storicizzati nel Database tabella L7_HTTP_HTTPS. Al cambio di data verranno cancellati i log del traffico avvenuto precedentemente il numero di giorni impostati rispetto la nuova data. Se il valore è minore a 0 (zero) le statistiche non verranno mai cancellate.

DBMaxHistoryDaysL4Datagram=:default=”DBMaxHistoryDays” UM=Giorni
Per valori maggiori di 0 è il numero massimo di giorni storicizzati nel Database tabella L4_DATAGRAM. Al cambio di data verranno cancellati i log del traffico avvenuto precedentemente il numero di giorni impostati rispetto la nuova data. Se il valore è minore a 0 (zero) le statistiche non verranno mai cancellate.

DBLoadHistorySessionActivity=:default=”true” UM=boolean
Se impostato a false il database non verrà popolato con i valori corrispondenti.

DBLoadHistoryPoolQueuesActivity=:default=”true” UM=boolean
Se impostato a false il database non verrà popolato con i valori corrispondenti.

DBLoadHistoryIncomingQueueHighWater=:default=”true” UM=boolean
Se impostato a false il database non verrà popolato con i valori corrispondenti.

DBLoadHistoryL4TcpTcpSSL=:default=”true” UM=boolean
Se impostato a false il database non verrà popolato con i valori corrispondenti.

DBLoadHistoryL7HttpHttps=:default=”true” UM=boolean
Se impostato a false il database non verrà popolato con i valori corrispondenti.

DBLoadHistoryL4Datagram=:default=”true” UM=boolean
Se impostato a false il database non verrà popolato con i valori corrispondenti.

DBLoadHistorySyslogEvent=:default=”true” UM=boolean

Se impostato a false il database non verrà popolato con i valori corrispondenti.

DBQueryLimit=:default=”5000” UM=Records
E' il numero massimo di records estraibili in una query per evitare overflow.

DBVarcharLimit=:default=”4000” UM=Byte
E' il numero massimo di bytes che verranno inseriti nelle colonne di tipo VARCHAR. Vedere manuale LBL_DBNetworkedConfiguration.pdf per maggiori dettagli su questo valore nelle diverse piattaforme DB.

DBMaxScriptBatch=:default=”200” UM=Byte
E' il numero massimo di script SQL utilizzati nel caricamento batch del database.

15 OPLON®Traffic Monetizer statisticbrokerwebcachectrl.xml

(LBL_HOME)/procsProfiles/A05_LBLGoStatisticsWebCacheDWHCTR/conf/statisticbrokerwebcachedwhctr.xml

Il file descrive il gestore delle tabelle di staging ed il loro ciclo di vita.

```
<serviceconf>
   <copyright>
   </copyright>
        <statisticbrokerwebcachedwhctr>
                <params>
                </params>
                <tables>
                    <id/>
                    …
                    …
                </tables>
        </statisticbrokerwebcachedwhctr>
</serviceconf>
```

<statisticbrokerwebcachedwhctr>

<params>

```
<serviceconf>
        <statisticbrokerwebcachedwhctr>
                <params
```

DBInitialConnectionPool=:default=”1”
Numero iniziale di connessioni al database.

DBMaxConnectionPool=:default=”1”
Numero massimo di connessioni al database. Superato questo limite verrà comunque aperta una nuova connessione che cesserà al termine della richiesta e non verrà riutilizzata dal connection pool.

DBDriver=:default=”oracle.jdbc.driver.OracleDriver”
E' il driver per gestire il Database Relazionale.

DBProtocol=:default=”jdbc:oracle:”
E' il protocollo della stringa di connessione al database

DBName=:default=””
E' il nome del database. Assieme al DBProtocol determina la stringa di connessione al DB.

DBLogin=:default=”null”
E' il login per la connessione al DB.

DBPassword=:default=”null”
E' la password per la connessione al DB.

<tables>

```
<serviceconf>
    <statisticbrokerwebcachedwhctr>
        <tables
```

Il paragrafo elenca le tabella di staging gestite da OPLON®Traffic Monetizer.

statement=:default=”{call lbl_admin.consmanager(?,?,?,?)}”
E' lo statement SQL invocato per eseguire il controllo dei motori di aggregazione e del ciclo di vita dei dati di staging.

<id>

```
<serviceconf>
    <statisticbrokerwebcachedwhctr>
        <tables
            <id
```

name=:default=”null”
E' il nome della tabella da gestire

partitionStep=:default=”2”
Numero di minuti di raccolta di una partizione.

flashForward=:default=”5”
Numero di partizioni future

rememberBack=:default=”0”
Numero di giorni di ritenzione delle statistiche analitiche prima della cancellazione

16 OPLON®Traffic Monetizer statisticbrokerwebcacheagr.xml

(LBL_HOME)/procsProfiles/D10_LBLDWH_xx_xxx_AGRxx/conf/
statisticbrokerwebcachedwhagr.xml

Il file imposta un aggregatore di dati.

```
<serviceconf>
   <copyright>
   </copyright>
        <statisticbrokerwebcachedwhagr>
                <params>
                </params>
        </statisticbrokerwebcachedwhagr>
</serviceconf>
```

<statisticbrokerwebcachedwhagr>

<params>

```
<serviceconf>
        <statisticbrokerwebcachedwhagr>
                <params
```

DBInitialConnectionPool=:default=”1”
Numero iniziale di connessioni al database.

DBMaxConnectionPool=:default=”1”
Numero massimo di connessioni al database. Superato questo limite verrà comunque aperta una nuova connessione che cesserà al termine della richiesta e non verrà riutilizzata dal connection pool.

DBDriver=:default=”oracle.jdbc.driver.OracleDriver”
E' il driver per gestire il Database Relazionale.

DBProtocol=:default=”jdbc:oracle:”

E' il protocollo della stringa di connessione al database

DBName=:default=””
E' il nome del database. Assieme al DBProtocol determina la stringa di connessione al DB.

DBLogin=:default=”null”
E' il login per la connessione al DB.

DBPassword=:default=”null”
E' la password per la connessione al DB.

aggregatorID=:default=”null”
Identificatore aggregatore.

table=:default=”null”
Nome tabella da aggregare (es,: L7_HTTP_HTTPS)

partitionStatementStart=:default=”{call lbl_admin.partition_to_consolidate(?,?,?,?,?)}”
Start registration statement

aggregatorStatement=:default=”null”
Aggregatore (es.: {call lbl_l7.cons_l7_fact_01_template(?)})

partitionStatementEnd=:default=”{call lbl_admin.set_partition_consolidated(?,?,?)}”
Sql end statement to call

17 OPLON® Tabelle logging

La persistenza dei dati di traffico nel database statistico viene realizzata attraverso le seguenti tabelle:

- SESSION_ACTIVITY
  analisi dell'utilizzo delle sessioni.
- L7_HTTP_HTTPS
  persistenza delle informazioni di traffico HTTP e HTTPS.
- L4_TCP_TCPSSL
  persistenza delle informazioni di traffico TCP e TCP come terminatore SSL.
- L4_DATAGRAM
  persistenza delle informazioni di traffico UDP e MULTICAST.
- POOL_QUEUES_ACTIVITY
  analisi dell'utilizzo dei pool di risoluzione delle richieste di servizio.
- INCOMING_QHIGHWATER_LEVEL
  analisi dell'utilizzo del riempimento della coda delle connessioni entranti.
- SYSLOG_EVENT
  Memorizzazione messaggi log infrastruttura OPLON®SAAI.

Tabella SESSION_ACTIVITY

Tabella contenente gli snap ad intervalli di 10” dello stato delle sessioni di instradamento.

RECORD_TYPE	Int	6	Tipo di record
VRRP_HOST_NAME	varchar(4000)		Nome dell'host del servzio VRRP. Questo valore associato al VRRP_PORT_NUMBER, definiscono l'istanza di bilanciamento per il consolidamento dei dati di traffico. Se Platform Edition vedere Reference Guide parametro uniqueContextID in iproxy.xml
VRRP_PORT_NUMBER	int		Port number del servzio VRRP. Questo valore associato al VRRP_HOST_NAME, definiscono l'istanza di bilanciamento per il

			consolidamento dei dati di traffico
END_POINTS_GROUPING	varchar(4000)		E' il nome del gruppo di endpoint
DOMAIN_REQUEST	varchar(4000)		E' il dominio associato alla sessione
COMMAND	varchar(4000)		EMPTY
URI_PATH_REQUEST	varchar(4000)		EMPTY
RESPONSE_CODE	int		0
END_POINT_HOST_NAME	varchar(4000)		Nome dell'host dell'endpoint associato alla sessione
END_POINT_PORT_NUMBER	int		Port number dell'host dell'endpoint associato alla sessione
END_POINT_URI_PATH	varchar(4000)		E' l'URIPath di contesto in elaborazione
USER_ID	varchar(4000)		Utilizzo futuro
CLIENT_ADDRESS	varchar(4000)		Il client address riporta l'indirizzo del client che ha richiesto il servizio. es.: 192.168.43.150 Se trasmissione a layer 7 HTTP/S ed e' stata impostata la gestione dell'entity X-Forwarded-For (HEADER HTTP) attraverso il parametro xForwardedFor="true" nel listener in iproxy.xml il valore dell'intera catena IP sara' trasferita. Ovviamente LBL puo' assicurare solo l'ultimo elemento della catena essendo gli altri elementi utili solo a scopo statistico essendo popolati da altri strumenti di infrastruttura come ad esempio i proxy. es.: 192.168.32.115,192.168.41.10,192.168.43.150
THIS_DATE	date		Data uniformata ad orario a 00:00:00
THIS_TIME	time		Ora con data uniformata a 01-01-1970
NUMBER_OF_ACTIVE_SESSIONS	int		E' il numero di sessioni relative alla chiave di raggruppamento (in rosso)

Le chiavi utilizzate sono:

- index K1_SESSION_ACTIVITY ON SESSION_ACTIVITY (THIS_DATE);
- index K2_SESSION_ACTIVITY ON SESSION_ACTIVITY (THIS_TIME);

Tabella L7_HTTP_HTTPS

Questa tabella contiene il traffico del protocollo HTTP e HTTPS (SSL).

RECORD_TYPE	Int	0=HTTP 1=HTTPS	Tipo di record
VRRP_HOST_NAME	varchar(4000)		Nome dell'host del servzio VRRP. Questo valore associato al VRRP_PORT_NUMBER, definiscono l'istanza di bilanciamento per il consolidamento dei dati di traffico. Se Platform Edition vedere Reference Guide parametro uniqueContextID in iproxy.xml
VRRP_PORT_NUMBER	int		Port number del servzio VRRP. Questo valore associato al VRRP_HOST_NAME, definiscono l'istanza di bilanciamento per il consolidamento dei dati di traffico
END_POINTS_GROUPING	varchar(4000)		E' il nome del gruppo di endpoint
DOMAIN_REQUEST	varchar(4000)		E' il dominio richiesto dal client
COMMAND	varchar(4000)		E' il comando HTTP richiesto dal client (GET, POST, etc)
URI_PATH_REQUEST	varchar(4000)		Path riferito al dominio richiesto dal client. Questo valore riporta il risultato della richiesta dopo l'azione di rewriting se esistente. E' l'effettiva richiesta che viene effettuata nel backend.
URI_PATH_REQUEST_ORG	varchar(4000)		Path originale riferito al dominio richiesto dal client prima di eventuali rewriting
CONTENT_TYPE_REQUEST	varchar(4000)		content type in request
CONTENT_TYPE_RESPONSE	varchar(4000)		content type in response
RESPONSE_CODE	int		Response CODE HTTP inviato dal server al client in risposta alla sua richiesta
END_POINT_HOST_NAME	varchar(4000)		Nome dell'host su cui è stata elaborata la richiesta di servzio
END_POINT_PORT_NUMBER	int		Port number dell'host su cui è stata elaborata la richiesta di servzio
END_POINT_URI_PATH	varchar(4000)		E' l'URIPath di contesto su cui è stata elaborata la richiesta

USER_ID	varchar(4000)		La colonna contiene le informazioni dell'utente profilato e autenticato. Le informazioni sono una somma degli elementi caratteristici di una autenticazione BASIC (Basic Authentication) e della autorizzazione derivante da certificato digitale. Questa colonna contiene rispettivamente: Se autorizzazione con certificato digitale il Subject con un valore aggiuntivo riportante il Serial Number del certificato es.: “CN=clientname, OU=clientlob, O=clientcompany, L=clientcountry, ST=clientdistrict, C=IT, SERIAL=1282479557” Se basic authentication “BASIC=usr1” “Digest=usr2” “AWS= usr3” Nel caso siano presenti entrambe le credenziali il valore risultante sara' : “BASIC=usr1, CN=clientname, OU=clientlob, O=clientcompany, L=clientcountry, ST=clientdistrict, C=IT, SERIAL=1282479557”
CLIENT_ADDRESS	varchar(4000)		Il client address riporta l'indirizzo del client che ha richiesto il servizio. es.: 192.168.43.150 Se trasmissione a layer 7 HTTP/S ed e' stata impostata la gestione dell'entity X-Forwarded-For (HEADER HTTP) attraverso il parametro xForwardedFor="true" nel listener in iproxy.xml il valore dell'intera catena IP sara' trasferita. Ovviamente LBL puo' assicurare solo l'ultimo elemento della catena essendo gli altri elementi utili solo a scopo statistico essendo popolati da altri strumenti di infrastruttura come ad esempio i proxy. es.: 192.168.32.115,192.168.41.10,192.168.43.150
USER_AGENT	varchar(4000)		Valore User-agent nell'HTTP HEADER
COOKIES	varchar(4000)		Valore Cookie nell'HTTP HEADER

REFERER	varchar(4000)		Valore Referer nell'HTTP HEADER
URI_PARAMETERS	varchar(4000)		Valore con parametri e/o query string. I parametri, definiti nella URL dal carattere ; e le query string definite daò carattere ? vengono normalizzate in una unico valore separato da &. es.: /aaa/bbb;c1=aaa&c2=bbb?q1=ccc&q2=dddd VALORE 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)		Indirizzo locale in cui è stata accettata la connessione
INCOMING_PORT_NUMBER	int		Porta locale in cui è stata accettata la connessione
THIS_DATE	date		Data uniformata ad orario a 00:00:00
THIS_TIME	time		Ora con data uniformata a 01-01-1970
COUNTER	bigint		Numero di operazioni sommate in questo record (se non sono stati modificati i default è il numero di operazioni effettuate nella sua finestra temporale negli ultimi 10”)
RESPONSE_TIME	bigint		E' il response time del servizio richiesto (Questo valore è relativo al traffico degli ultimi 10”. Per avere un valore medio è necessario dividerlo per il COUNTER). Valore espresso in nanosecondi.
LAP_TIME_A	bigint		Tempo compreso tra la fine della lettura dell'HEADER del client e l'inizio della connessione verso l'endpoint (Questo valore è relativo al traffico degli ultimi 10”. Per avere un valore medio è necessario dividerlo per il COUNTER). Valore espresso in nanosecondi.
LAP_TIME_B	bigint		Tempo di connessione all'endpoint (Questo valore è relativo al traffico degli ultimi 10”. Per avere un valore medio è necessario dividerlo per il COUNTER). Valore espresso in nanosecondi.
LAP_TIME_C	bigint		Tempo tra la connessione avvenuta all'endpoint e l'inizio della lettura dell'HEADER di risposta. Se il client invia il body è il tempo di trasmissione del body dal client all'endpoint.(Questo valore è relativo al traffico degli ultimi 10”. Per

			avere un valore medio è necessario dividerlo per il COUNTER) . Valore espresso in nanosecondi.
LAP_TIME_D	bigint		Tempo di lettura dell'Header dell'endpoint (Questo valore è relativo al traffico degli ultimi 10”. Per avere un valore medio è necessario dividerlo per il COUNTER). Valore espresso in nanosecondi.
LAP_TIME_E	bigint		Tempo tra la fine della lettura dell'Header e la fine dei dati in risposta (Questo valore è relativo al traffico degli ultimi 10”. Per avere un valore medio è necessario dividerlo per il COUNTER). Valore espresso in nanosecondi.
HEADER_LENGTH_FROM_CLIENT	bigint		E' la lunghezza dell'HEADER di richiesta dei client verso gli endpoint (Questo valore è relativo al traffico degli ultimi 10”. Per avere un valore medio è necessario dividerlo per il COUNTER)
BYTES_SENT_FROM_CLIENT	bigint		Totale trasferito compresa l'HEADER dal client verso gli endpoint (Questo valore è relativo al traffico degli ultimi 10”. Per avere un valore medio è necessario dividerlo per il COUNTER)
HEADER_LENGTH_FROM_END_POINT	bigint		E' la lunghezza dell'HEADER di risposta degli endpoint verso i client (Questo valore è relativo al traffico degli ultimi 10”. Per avere un valore medio è necessario dividerlo per il COUNTER)
BYTES_SENT_FROM_END_POINT	bigint		Totale trasferito compresa l'HEADER dagli endpoint verso i client (Questo valore è relativo al traffico degli ultimi 10”. Per avere un valore medio è necessario dividerlo per il COUNTER) relative alla chiave di raggruppamento (in rosso)

Le chiavi utilizzate sono:

- index K1_L7_HTTP_HTTPS ON L7_HTTP_HTTPS (THIS_DATE);
- index K2_L7_HTTP_HTTPS ON L7_HTTP_HTTPS (THIS_TIME);

Tabella L4_TCP_TCPSSL

Come per la tabella precedente questa tabella contiene i dati di traffico relativo alle attività del connettore TCP e TCP come terminatore SSL. La struttura iniziale è molto simile a quella HTTP/S per differenziarsi poi sui dati di traffico

RECORD_TYPE	Int	2=TCP 3=SSL	Tipo di record
VRRP_HOST_NAME	varchar(4000)		Nome dell'host del servzio VRRP. Questo valore associato al VRRP_PORT_NUMBER, definiscono l'istanza di bilanciamento per il consolidamento dei dati di traffico. Se Platform Edition vedere Reference Guide parametro uniqueContextID in iproxy.xml
VRRP_PORT_NUMBER	int		Port number del servzio VRRP. Questo valore associato al VRRP_HOST_NAME, definiscono l'istanza di bilanciamento per il consolidamento dei dati di traffico
END_POINTS_GROUPING	varchar(4000)		E' il nome del gruppo di endpoint
DOMAIN_REQUEST	varchar(4000)		Se WebSocket è l'host richiesto a layer 7
COMMAND	varchar(4000)		Identifica il flusso dei dati: CLIENT_FLOW dal client verso gli endpoint ENDPOINT_FLOW dagli endpoint verso i client WebSocket: WSCLIENT_FLOW dal client verso gli endpoint WSENDPOINT_FLOW dagli endpoint verso i client
URI_PATH_REQUEST	varchar(4000)		Se WebSocket è l'URIPath richiesto a layer 7
RESPONSE_CODE	int		0
END_POINT_HOST_NAME	varchar(4000)		Nome dell'host su cui è stata elaborata la richiesta di servzio
END_POINT_PORT_NUMBER	int		Port number dell'host su cui è stata elaborata la richiesta di servzio
END_POINT_URI_PATH	varchar(4000)		Se WebSocket è l'URIPath richiesto a layer 7
USER_ID	varchar(4000)		If a 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)		Il client address riporta l'indirizzo del client che ha richiesto il servizio.

			es.: 192.168.43.150
COOKIES	varchar(4000)		Valore univoco associato alla connessione. Questo valore viene popolato con il valore LBLCOLOR per identificare la singola connessione a layer 4. Per l'attivazione del popolamento di questo valore è necessario impostare il parametro **distinguishSingleConnection**="**true**" nel paragrafo **<bind>** del file parametri iproxy.xml Se WebSocket contiene i cookie della connessione layer 7 HTTP/S
INCOMING_ADDRESS	varchar(4000)		Indirizzo locale in cui è stata accettata la connessione
INCOMING_PORT_NUMBER	int		Porta locale in cui è stata accettata la connessione
THIS_DATE	date		Data uniformata ad orario a 00:00:00
THIS_TIME	time		Ora con data uniformata a 01-01-1970
COUNTER	bigint		Numero di operazioni sommate in questo record (se non sono stati modificati i default è il numero di operazioni effettuate nella sua finestra temporale negli ultimi 10")
BYTES_FORWARDED	bigint		E' il numero di bytes scambiati in base al flusso (client>endpoint oppure endpoint>client) relative alla chiave di raggruppamento (in rosso)
START_ADV_TIME	bigint		E' il momento in cui viene rilevato il primo buffer/carattere espresso in nanosecondi
END_ADV_TIME	bigint		E' il momento in cui viene eseguito il flush dell'ultimo buffer dello stream espresso in nanosecondi
TOTAL_ADV_TIME	bigint		E' il tempo totale di forwarding dell'informazione (END_ADV_TIME – START_ADV_TIME) espresso in nanosecondi

Le chiavi utilizzate sono:

- index K1_L4_TCP_TCPSSL ON L4_TCP_TCPSSL (THIS_DATE);
- index K2_L4_TCP_TCPSSL ON L4_TCP_TCPSSL (THIS_TIME);

Tabella L4_DATAGRAM

Come per la tabella precedente questa tabella contiene i dati di traffico relativo alle attività del connettore TCP e TCP come terminatore SSL. La struttura iniziale è molto simile a quella HTTP/S per differenziarsi poi sui dati di traffico

RECORD_TYPE	Int	10=UDP 11=MULTICAST	Tipo di record
VRRP_HOST_NAME	varchar(4000)		Nome dell'host del servzio VRRP. Questo valore associato al VRRP_PORT_NUMBER, definiscono l'istanza di bilanciamento per il consolidamento dei dati di traffico. Se Platform Edition vedere Reference Guide parametro uniqueContextID in iproxy.xml
VRRP_PORT_NUMBER	int		Port number del servzio VRRP. Questo valore associato al VRRP_HOST_NAME, definiscono l'istanza di bilanciamento per il consolidamento dei dati di traffico
END_POINTS_GROUPING	varchar(4000)		E' il nome del gruppo di endpoint
DOMAIN_REQUEST	varchar(4000)		EMPTY
COMMAND	varchar(4000)		Identifica il flusso dei dati: sempre vuoto in quanto il flusso è sempre dal client agli endpoint
URI_PATH_REQUEST	varchar(4000)		EMPTY
RESPONSE_CODE	int		0
END_POINT_HOST_NAME	varchar(4000)		Nome dell'host su cui è stata elaborata la richiesta di servzio
END_POINT_PORT_NUMBER	int		Port number dell'host su cui è stata elaborata la richiesta di servzio
END_POINT_URI_PATH	varchar(4000)		EMPTY
USER_ID	varchar(4000)		Utilizzo futuro
CLIENT_ADDRESS	varchar(4000)		Il client address riporta l'indirizzo del client che ha richiesto il servizio. es.: 192.168.43.150
INCOMING_ADDRESS	varchar(4000)		Indirizzo locale in cui è stata accettata la connessione
INCOMING_PORT_NUMBER	int		Porta locale in cui è stata accettata la connessione
THIS_DATE	date		Data uniformata ad orario a 00:00:00

THIS_TIME	time		Ora con data uniformata a 01-01-1970
COUNTER	bigint		Numero di operazioni sommate in questo record (se non sono stati modificati i default è il numero di operazioni effettuate nella sua finestra temporale negli ultimi 10”)
BYTES_FORWARDED	bigint		E' il numero di bytes scambiati

Le chiavi utilizzate sono:

- index K1_L4_DATAGRAM ON L4_DATAGRAM (THIS_DATE);
- index K2_L4_DATAGRAM ON L4_DATAGRAM (THIS_TIME);

Tabella POOL_QUEUES_ACTIVITY

Questa tabella contiene a snap di 10” lo stato di attività dei risolutori di protocollo. La sua interpretazione non può essere utilizzabile come sommatoria delle attività della giornata ma come dato statistico istantaneo nel momento relativo alla data e all'ora riportate sui records. Con questi snap è possibile tracciare su un asse temporale lo stato di attività e quindi i picchi di utilizzazione con uno slice di 10”.
Questa tabella contiene due tipi di snap, uno relativo allo stato di attività dei risolutori di protocollo e l'altro relativo allo stato di “committed” dei risolutori di protocollo.

RECORD_TYPE	Int	4=ACTIVITY 5=COMMITTED	Tipo di record
VRRP_HOST_NAME	varchar(4000)		Nome dell'host del servzio VRRP. Questo valore associato al VRRP_PORT_NUMBER, definiscono l'istanza di bilanciamento per il consolidamento dei dati di traffico. Se Platform Edition vedere Reference Guide parametro uniqueContextID in iproxy.xml
VRRP_PORT_NUMBER	int		Port number del servzio VRRP. Questo valore associato al VRRP_HOST_NAME, definiscono l'istanza di bilanciamento per il consolidamento dei dati di traffico
END_POINTS_GROUPING	varchar(4000)		E' il nome del gruppo di endpoint
DOMAIN_REQUEST	varchar(4000)		E' il dominio in elaborazione sul risolutore
COMMAND	varchar(4000)		EMPTY
URI_PATH_REQUEST	varchar(4000)		EMPTY

RESPONSE_CODE	int		0
END_POINT_HOST_NAME	varchar(4000)		Nome dell'host su cui è stata elaborata la richiesta di servzio
END_POINT_PORT_NUMBER	int		Port number dell'host su cui è stata elaborata la richiesta di servzio
END_POINT_URI_PATH	varchar(4000)		E' l'URIPath di contesto in elaborazione
USER_ID	varchar(4000)		Utilizzo futuro
CLIENT_ADDRESS	varchar(4000)		Il client address riporta l'indirizzo del client che ha richiesto il servizio. es.: 192.168.43.150 Se trasmissione a layer 7 HTTP/S ed e' stata impostata la gestione dell'entity X-Forwarded-For (HEADER HTTP) attraverso il parametro xForwardedFor="true" nel listener in iproxy.xml il valore dell'intera catena IP sara' trasferita. Ovviamente LBL puo' assicurare solo l'ultimo elemento della catena essendo gli altri elementi utili solo a scopo statistico essendo popolati da altri strumenti di infrastruttura come ad esempio i proxy. es.: 192.168.32.115,192.168.41.10,192.168.43.150
THIS_DATE	date		Data uniformata ad orario a 00:00:00
THIS_TIME	time		Ora con data uniformata a 01-01-1970
NUMBER_OF_BUSY_CONSUMER	int		E' il numero di consumatori in attività nell'istante THIS_DATE e THIS_TIME relative alla chiave di raggruppamento (in rosso)

Le chiavi utilizzate sono:

- index K1_POOL_QUEUES_ACTIVITY ON POOL_QUEUES_ACTIVITY (THIS_DATE);
- index K2_POOL_QUEUES_ACTIVITY ON POOL_QUEUES_ACTIVITY (THIS_TIME);

Tabella INCOMING_QHIGHWATER_LEVEL

E' la tabella che registra a snap di 10” il livello di riempimento della coda delle richieste di connessione entranti. Questa tabella è importante perché in relazione al numero di risolutori di protocollo è l'indicatore di un possibile attacco DoS o di risorse insufficienti per elaborare il carico di richieste.

RECORD_TYPE	Int	7	Tipo di record
VRRP_HOST_NAME	varchar(4000)		Nome dell'host del servzio VRRP. Questo valore associato al VRRP_PORT_NUMBER, definiscono l'istanza di bilanciamento per il consolidamento dei dati di traffico. Se Platform Edition vedere Reference Guide parametro uniqueContextID in iproxy.xml
VRRP_PORT_NUMBER	int		Port number del servzio VRRP. Questo valore associato al VRRP_HOST_NAME, definiscono l'istanza di bilanciamento per il consolidamento dei dati di traffico
THIS_DATE	date		Data uniformata ad orario a 00:00:00
THIS_TIME	time		Ora con data uniformata a 01-01-1970
HIGH_WATER	int		E' il numero di richieste di connessione all'interno della coda prima in attesa di essere elaborate
HIGH_WATER_LEVEL	float		E' il fattore di calcolo derivato da: (100*HIGH_WATER)/ACT_SESSIONS
HIGH_WATER_WARNING_LEVEL	float		E' la soglia in % superata la quale viene spedito un messaggio di avvertimento
HIGH_WATER_DANGER_LEVEL	float		E' la soglia in % superata la quale viene spedito un messaggio di pericolo
ACT_SESSIONS	int		E' l'attuale disponibilità di risolutori di protocollo
MAX_CONCURRENT_SESSIONS	int		E' il numero massimo di risolutori di protocollo

Le chiavi utilizzate sono:

- index K1_INCOMING_QUEUE_HL ON INCOMING_QUEUE_HIGHWATER_LEVEL (THIS_DATE);
- index K2_INCOMING_QUEUE_HL ON INCOMING_QUEUE_HIGHWATER_LEVEL (THIS_TIME);

Tabella SYSLOG_EVENT

Questa tabella raccoglie tutti i messaggi provenienti da tutti i processi OPLON®A.A.I.. In ambienti complessi è quindi un utile strumento che, se centralizzato, può essere utilizzato come sistema per rilevare eventuali anomalie di funzionamento.

RECORD_TYPE	Int	12	Tipo di record
VRRP_HOST_NAME	varchar(4000)		E' una stringa separata da @ contenente: host name@monitor management url@server process name@absolute log dir@date YYYYMMDD@hostname_logfileSuffix
VRRP_PORT_NUMBER	int		
COOKIES	varchar(4000)		E' un valore identificativo del messaggio sempre differente ed univoco per sistema es.: ID=”16231623”
INCOMING_ADDRESS	varchar(4000)		Host name
THIS_DATE	date		Data uniformata ad orario a 00:00:00
THIS_TIME	time		Ora con data uniformata a 01-01-1970
REPETITIONS	bigint		Numero di ripetizioni dello stesso messaggio
TIME_SEQUENCE	bigint		Sequenza temporale evento
SEVERITY	varchar(4000)		ERROR | WARNING | DEBUG | FATAL
JAVA_REL	varchar(4000)		Release java
LBL_REL	varchar(4000)		Relelase LBL
MESSAGE_GRP	varchar(4000)		Nome dell'unita di elaborazione che ha generato il messaggio
HOST_ID	varchar(4000)		Nome dell'host che ha generato il messaggio
MESSAGE	varchar(4000)		Messaggio
MONITOR_MNG_URL	varchar(4000)		
MONITOR_PROCESS_NAME	varchar(4000)		

Le chiavi utilizzate sono:

- 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 ** because it still has not been started in-house theater management. Immediately after the start 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**

18 OPLON®ADC TCOHTTPUtils.xml

(LBL_HOME)/lib/tco/resources/net/http/TCOHTTPUtils.xml

Attraverso il file opzionale TCOHTTPUtils.xml è possibile intervenire sul comportamento del keep-alive in dipendenza del contenuto dell'HTTP HEADER.
In particolare è possibile comandare una close del canale al termine della response in base ad un valore contenuto nello user-agent e per response code.

```
<root>
 <copyright>
 </copyright>
 <TCOHTTPUtil>
  <uaKeepAliveEvaluation>
   <userAgent/>
   ...
  </uaKeepAliveEvaluation>
 </TCOHTTPUtil>
</root>
```

<TCOHTTPUtil>

<uaKeepAliveEvaluation>

<userAgent>

```
<root>
        <TCOHTTPUtil>
                <uaKeepAliveEvaluation>
                        <userAgent
```

value=:valore di default=””
E' il valore da ricercare all'interno dello user-agent nell'HTTP header.

ignoreCase=:valore di default=”false”
Influenza il tipo di match all'interno della stringa. Se true il confronto non distiguerà differenza tra i caratteri minuscoli e maiuscoli.

compareType=:valore di default=”0”

Indica il tipo di ricerca all'interno della stringa:

- 0=contains (default)
- 1=equals
- 2=startsWith
- 3=endsWith
- 4=matches - regex compare

returnCode=:valore di default=”999”
E' il return code HTTP al quale deve essere applicato il match dello user-agent. A parità di return-code e del match dello user agent il canale sarà chiuso al temine della spedizione della risposta al client.

Esempio di file TCOHTTPUtils.xml:

```
<?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>
```

19 OPLON®ADC Courtesy message

(LBL_HOME)/procsProfiles/A10_LBLGo/resources/html

Il messaggio di cortesia è importante perché permette di informare gli utenti di un disservizio ed eventualmente comunicare delle azioni alternative.

Il messaggio viene utilizzato da LBL LodBalancer quando il servizio richiesto non può essere instradato in nessun EndPoint quando sono indisponibili a causa di manutenzione o per effettivi problemi.

Il messaggio può essere modificato durante il runtime. Al verificarsi dell'evento OPLON®LoadBalancer prenderà sempre la nuova versione o, in assenza del file, il messaggio di default.

Per poter personalizzare il messaggio e' sufficiente creare la directory.

- (LBL_HOME)/procsProfiles/A10_LBLGo/resources/html

In questa directory dovrà essere creato il file "messageNoEndPoint.html"

Il path risultante dovra' quindi essere:

- (LBL_HOME)/procsProfiles/A10_LBLGo/resources/html/messageNoEndPoint.html

Il file "messageNoEndPoint.html" dovrà essere un file HTML semplice senza referenze esterne (immagini o filmati ad esempio).
Di seguito i parametri che all'interno del file HTML vengono sostituiti con il messaggio appropriato:

- %title% = E' il titolo (caricato con "LBL(r)ADC")
- %msg1% = E' il return code HTTP che viene visualizzato in rosso
- %msg2% = E' l'URIPath che viene visualizzata in nero
  es.: NO HOST AVAILABLE FOR URI PATH

Si consiglia di modificare il form HTML personalizzandolo con un proprio messaggio ma di lasciare, magari non in evidenza, il messaggio originale per capire eventualmente l'origine del problema se non dovesse essere un down-time programmato del servizio ma dovesse

essere un effettivo problema nel servizio di backend.

Nell'area riservata possono essere scaricati dei messaggi di default disponibili in un archivio compresso (zip).

20 OPLON® notificationDir

Con OPLON®S.A.A.I. è possibile intervenire dall'esterno per comunicare eventuali malfunzionamenti dei servizi di backend o per manutenzioni programmate.
In alcune occasioni infatti, pur essendo operativi i servizi posti in bilanciamento, potrebbero indirettamente essere indisponibili per failure o manutenzione delle risorse ad essi associati. Si pensi ad esempio ad applicazioni che afferiscono ad un DataBase. Pur essendo i servizi web attivi il DataBase potrebbe essere fuori servizio e quindi anche i servizi web ad esso associati saranno, di fatto, fuori servizio pur rispondendo correttamente alla connessione.

A tal proposito nel file iproxy.xml è possibile assegnare dei nomi associativi di riferimento a gruppi/domini fino al singolo end-point. Se viene creato un file nella directory "(LBL_HOME)/lib/notificationDir" con nome "outOfOrder.nomeAssociativo" l'end-point o il dominio oppure il gruppo di end-point associati a quel nome si metteranno in stato di out-of-order.

Nel package LBLLoadBalancer_xxxxxxxxx_009_00x_00x.zip sono presenti due esempi di programmi di health check forniti in formato sorgente che eseguono dei controlli custom delle risorse. Entrambi i programmi sono corredati del file di startup process per OPLON®LoadBalancer Monitor anch'esso già presente nella directory (LBL_HOME)/lib/confMonitor e quindi disponibile immediatamente per l'utilizzo. Rimandiamo al manuale OPLON®LoadBalancer How To Use plugins per una completa trattazione dell'utilizzo di questi programmi di esempio.

(LBL_HOME)/lib/plugin/lblplugin/	Descrizione
Z99_LBLPluginHTTPCheckWithCmdStart.xml	Esegue l'healthcheck di un indirizzo HTTP e controlla la connettività e il return code. Allo start ed al cambiamento di stato ready/not-ready e viceversa può lanciare un programma per l'esecuzione di procedure pertinenti all'anomalia rilevata.
Z99_LBLPluginNetworkCheckWithCmdStart.xml	Esegue l'healthcheck di connettività di un indirizzo porta TCP/IP. Allo start ed al cambiamento di stato ready/not-ready e viceversa può lanciare un programma per

	l'esecuzione di procedure pertinenti all'anomalia rilevata.

```
<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"/>
    ...
```

Si può notare che nella sezione "endPointsGrouping" sono stati assegnati 2 nomi simbolici "pippo" e "pluto", nella sezione "virtualDomain" sono stati assegnati altri due nomi simbolici "paperino" e "minni" e solo negli "endp" con uriPath="/TCOProject" sono stati associati i nomi "qui", "quo" e "qua".

Se durante l'esecuzione viene creato un file nella nuova directory:

(LBL_HOME)/lib/notificationDir/outOfOrder.quo
solo i due end-point con uriPath="/TCOProject" verranno posti in out of order

Se durante l'esecuzione viene creato un file nella nuova directory:

(LBL_HOME)/lib/notificationDir/outOfOrder.minni
tutti gli end-point con quel dominio verranno posti in out of order

Se durante l'esecuzione viene creato un file nella nuova directory:

(LBL_HOME)/lib/notificationDir/outOfOrder.pippo
tutti gli end-point con quel gruppo verranno posti in out of order

Questo metodo permette ai sistemisti di utilizzare qualsiasi strumento per effettuare i check dei servizi e solamente creando/cancellando un file di abilitare o disabilitare interi gruppi di servizi.

A questa funzionalità si aggiunge la possibilità di porre fuori servizio dal punto di vista logico l'intero servizio di bilanciamento. Questa funzionalità si attiva nella stessa modalità della precedente e cioè creando un file avente come estensione il nome dell'istanza di bilanciamento contenuta nel file parametri systemsmonitor_m.xml:

File parametri:
(LBL_HOME)/lib/conf/systemsmonitor_m.xml

Frammento del file parametri:

```
<systemsmonitor_m>
        <params
                systemsMonitorGroup="systemsMonitorGroup"   <=== Nome
                notificationDir="build/classes/notificationDir"
                monitorTimer="100....
```

Per porre questo servizio in OutOfOrder è sufficiente creare un file con nome:

- outOfOrder.systemsMonitorGroup

Appena l'istanza di bilanciamento rileva l'esistenza del file si pone in OutOfOrder togliendo anche gli indirizzi virtuali. Per ripristinare la situazione è sufficiente eliminare il file e l'istanza di bilanciamento riprenderà il suo ruolo all'interno del pool di bilanciamento.

DISABILITAZIONE DI SERVIZI

OutOfOrder

Il file con prefisso outOfOrder pone un servizio come non più disponibile e ne segnala continuamente l'indisponibilità attraverso il log e, se attivati, mail e/o post HTTP.

disable

Ci sono dei casi tuttavia che necessitano di disabilitazione dell'endpoint senza la segnalazione dell'evento. Questi casi sono molto frequenti in caso di servizi in fail-over o business-continuity, fast disaster recovery o disaster recovery dove la necessità di chiudere preventivamente un canale è molto importante.

Per disabilitare un gruppo di endpoint senza segnalazione dell'evento è sufficiente creare un file vuoto con il prefisso disable.

- es.: (LBL_HOME)/lib/notificationDir/disable.quo

ATTENZIONE: come per l'outOfOrder è possibile disabilitare un intero gruppo di servizi in dipendenza della posizione del nome associativo.

drain

The drain function disable new incoming connections for the endpoints resources associated to the tag.
The existing connections or services will continue to function until their session time expires (layer 4 TCP/UDP layer 7 HTTP/S).
For layer 4 TCP persistent connections without sticky session management, ssh or rdp for example, the connection will continue to work until first disconnection.
After disconnection the client will fail to connect again

21 OPLON® statisticCacheHistory

Con l'introduzione delle statistiche sono stati introdotti dei metodi per controllarne il flusso durante il runtime.

Per poter attivare le esclusioni è sufficiente creare un file nella directory della cache con i seguenti nomi per le rispettive disabilitazioni:

La generazione e popolamento delle statistiche (cache di 1° e 2° Livello)
Nome file per fermare il popolamento delle statistiche

- (LBL_HOME)/procsProfiles/XXX_procName/statisticCacheHistory/stopLoadStat

ATTENZIONE: Lo stop del popolamento delle statistiche a questo livello ferma la generazione anche della cache. I dati generati durante il permanere dello stop non saranno recuperabili

La spedizione delle statistiche dal processo di bilanciamento verso l'application server StatisticsBorokerWebCach.
Nome file per fermare la spedizione delle statistiche verso web cach:

- (LBL_HOME)/procsProfiles/XXX_procName/statisticCacheHistory/stopSendStat

Questo stop non pregiudica la successiva riattivazione e spedizione delle statistiche generate fino a quel momento se entro i limiti di ritenzione delle statistiche in cache di 2° livello impostate di default a 1 giorno solare.

Il trace sul log dei due controlli di flusso per disattivazione totale del servizio statistico
Nome file per fermare il log dei messaggi di avviso dell'avvenuto stop delle statistiche

- (LBL_HOME)/procsProfiles/XXX_procName/statisticCacheHistory/ stopNoLogError

22
OPLON®ADC
forceIncomingConnectionToWait

E' possibile fermare lo svuotamento della coda delle connessioni entranti. Questo evento è molto importante perché permette di congelare le comunicazioni per un periodo di tempo che, per natura, deve essere molto limitato ma che permetta di creare dei punti di sincronizzazione.

Per creare questo evento e quindi bloccare lo svuotamento della coda è sufficiente impostare il file:

- (LBL_HOME)/lib/notificationDir/forceIncomingConnectionToWait

Per far riprendere lo svuotamento della coda delle connessioni entranti è sufficiente cancellarlo.

ATTENZIONE: Questo file deve essere utilizzato in maniera prudente con per tempi brevi. Il mantenere questo file di blocco dello scodamento delle connessioni entranti può causare Out Of Memory

23 OPLON®IPNetworkCardRedundancy ipncr.xml

IP Network Card Redundancy è il servizio che permette di ridondare le interfacce di rete. E' possibile eseguire gli health check sia con responso positivo sia con responso negativo e quindi in base al risultato spostare l'indirizzo in un'altra interfaccia di rete. Questo servizio è disponibile su tutte le versioni OPLON® distribuite per fornire funzionalità di alta affidabilità IP.

E' possibile configurare più gruppi di ridondanza che verranno trattati in maniera distinta. Ogni gruppo di ridondanza è contenuto nel paragrafo <floatingAddress> con proprie caratteristiche.

```
<serviceconf>
   <copyright>
   </copyright>
        <ipncr>
                <params>
                </params>
                <floatingAddressesMgr>
                  <floatingAddress>
                    <floatingInterface>
                    </floatingInterface>
                    ...
                    <healthCheckPolicy>
                      <healthCheck>
                      </healthCheck>
                    </healthCheckPolicy>
                    ...
                    <healthCheckConditionPolicy>
                      <healthCheck>
                      </healthCheck>
                    </healthCheckConditionPolicy>
                    ...
                  </floatingAddress>
                </floatingAddressesMgr>
        </ipncr>
</serviceconf>
```

Esempio tratto da un'installazione Solaris.

```
<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>
```

Facendo riferimento all'esempio precedente notiamo che il floatingAddress 192.168.47.113 può essere associato alle interfacce e1000g1:1 ed e1000g3:1.
Il processo verifica costantemente che l'interfaccia che detiene il floatingAddress sia in grado di eseguire correttamente gli health check descritti nell'apposito paragrafo. Nel caso questi dovessero dare tutti esito negativo, il sistema verifica il funzionamento dell'interfaccia e1000g3:1 ed in caso positivo assegna il floatingAddress a quest'ultima.

E' possibile utilizzare più gruppi di health check per differenziare i test, ad esempio, su reti differenti. Nell'esempio di seguito è stato aggiunto un nuovo gruppo di Health Check. Se un gruppo di Health Check risultasse completamente non responsivo avverrà lo switch sull'interfaccia successiva.

```
 <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>
```

E' possibile inoltre condizionare il set dell'indirizzo in base a condizioni esterne come ad esempio l'esistenza di un servizio. La condizione può essere sia positiva che negativa e quindi l'impostazione dell'indirizzo avverrà se, in condizione positiva, i servizi/indirizzi sono esistenti; in condizione negativa, se i servizi/indirizzi non sono esistenti.
Un esempio di impostazione di condizioni è di seguito riportato:

```
<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>
```

Il questo caso, verificate le condizione di health check, il paragrafo healthCheckConditionPolicy, con parametro positiveCondition=”true”, indica che se almeno uno dei servizi/indirizzi indicati è attivo, il floatingAddress deve essere assegnato ad una delle due interfacce.

Nel caso invece di positiveCondition=”false” l'indirizzo floating verrebbe impostato solo se i servizi/indirizzi ad esso associati non rispondessero agli health check.

NOTA: E' sufficiente un solo servizio/indirizzo attivo per considerare valida la condizione di health check.

<ipncr>

```
<serviceconf>
    <ipncr>
```

<params>

```
<serviceconf>
    <ipncr>
        <params
```

frequency=: valore di default=”10000” UM=Millisecondi
E' la frequenza di verifica cambiamenti di stato nei processi o parametri.

sysCommandTimeOut=:valore di default=”10000” UM=Millisecondi
Indica il tempo necessario per dichiarare in time-out un comando di sistema. Se il comando oltrepassa questo limite viene eseguito un comando di abort e successivamente viene rilasciato il controllo all'applicazione.

sysCommandCheckRate=:valore di default=”300” UM=Millisecondi
E' la frequenza del controllo dello stato del comando di sistema.

sysCommandRemoteURL=:default=”https://localhost:5992/SysCommand”
E' l'URL del servizio per eseguire i comandi di sistema

createConnectionTimeOut=: valore di default=”5000” UM=Millisec.
Time-out del tentativo di connessione

numRetryConnection=: valore di default=”3”
Numero di tentativi di connessione prima di dichiarare OutOfOrder la risorsa

<floatingAddressesMgr>

```
<serviceconf>
        <ipncr>
                <floatingAddressesMgr>
```

Il paragrafo contiene le definizioni dei floatingAddress

<floatingAddress>

```
<serviceconf>
        <ipncr>
                <floatingAddressesMgr>
                        <floatingAddress
```

enable=: valore di default=”false”
Abilita disabilita il caricamento e la valutazione del paragrafo

description=: valore di default=””
Descrizione del floating address.

address=: valore di default=””
Indirizzo del floatingAddress.

netmask=: valore di default=””
Netmask dell'indirizzo floating.

resetOnShutdown=: valore di default=”true”
Se true durante lo shutdown controllato del servizio l'indirizzo viene eliminato da qualsiasi interfaccia di rete descritta nei paragrafi <floatingInterface>. Se false allo shutdown del servizio non verrà intrapresa nessuna azione lasciando le interfacce di rete allo stato attuale. E' da ricordare che allo startup del servizio tutti gli indirizzi floating sono eliminati dalle interfacce di rete prestabilite. L'indirizzo è reimpostato solo dopo l'analisi delle condizioni di health check.

associateFItoHCP=: valore di default=”false”
In una situazione con più schede di rete collegate tra loro attraverso crosscable o switch ed un indirizzo floating per ogni nodo, come nello schema di seguito, e' possibile inserire l'indicatore "associateFItoHCP" (associate Floating Address to Health Check Policy) nel file ipncr.xml. Questo indicatore associa il gruppo di healthcheck alla scheda di rete.

ATTENZIONE: Con crosscable la migrazione all'indietro dell'indirizzo deve essere comandata manualmente con uno stop e start del servizio in caso di ripristino dell'interfaccia fisica.

Impostando associateFItoHCP="true" l'health check sarà associato all'interfaccia in questo modo:

- Prima floating interface a primo gruppo di health check;
- Seconda floating interface a secondo gruppo di healthcheck:
- Terza floating interface a terzo gruppo di healthcheck:

Quello che si ottiene è una migrazione dell'indirizzo floating tra le network card.

Esempio di migrazione a seguito di un failure:

```
Stato A: Tutto ok
=======================================================
NODO A                                                  NODO 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

Stato B: NODO A prima scheda in failure
=======================================================
NODO A                                                  NODO 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
<====================Migrazione

+staticAddress--------------+   <--CrossCable-->  +staticAddress--------------+
192.168.45.101                                    192.168.45.100
```

configurazione 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=: valore di default=”false”
Abilita disabilita il caricamento e la valutazione del paragrafo

device=:valore di default=””
E' l'identificatore dell'alias associato ad un device fisico per Unix/Linux e del device fisico per MS Windows. Per ottenere questo nome si utilizzeranno “devcon.exe” per MS Windows e “ifconfig” per i sistemi Unix/Linux. Fare riferimento al manuale di installazione OPLON®LoadBalancer IP Network Card redundancy Installation per maggiori dettagli di installazione e impostazione.

deviceName=:valore di default=””
E' il nome del device. Su MS Windows deve assumere il valore del nome visibile in Risorse di rete. Fare riferimento al manuale di installazione OPLON® LoadBalancer IP Network Card redundancy Installation per maggiori dettagli di installazione e impostazione.

Di seguito alcune possibili configurazioni:

MS Windows:

```
<floatingInterface
device="PCI\\VEN_8086&amp;DEV_1019&amp;SUBSYS_80F71043&amp;REV_00\
\4&amp;3B3CB9B1&amp;0&amp;0818"
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
```

E' il paragrafo preposto all'health check degli indirizzi per determinare la raggiungibilità della rete dall'interfaccia selezionata. Se tutti gli health check configurati in questo paragrafo dovessero dare esito negativo allora l'indirizzo verrà migrato in un'altra interfaccia di rete. E' anche possibile predisporre più gruppi di healtcheck duplicando questo paragrafo come descritto all'inizio di questo capitolo.

enable=: valore di default=”false”
Abilita disabilita il caricamento e la valutazione del paragrafo

description=: valore di default=””
Descrizione del controllo

positiveCondition=: valore di default=”true”
Cambia il senso del controllo.
Se true e almeno un controllo è risultato positivo allora tutto il paragrafo è positivo.
Se false e almeno un controllo è risultato positivo allora tutto il paragrafo risulta negativo.

<healthCheck>

```
<serviceconf>
        <ipncr>
                <floatingAddressesMgr>
                        <floatingAddress>
                                <healthCheckPolicy>
                                        <healthCheck
```

enable=:default=”false”
Abilita disabilita il caricamento e la valutazione del paragrafo

address=: valore di default=””
E' l'indirizzo su cui viene effettuato l'health check..

port=: valore di default=”0”
E' la porta su cui risponde il servizio di health check. Se <=0 verrà eseguito un check ICMP

SSL=: valore di default=”false”
Se true, esegue l'health check del servizio attraverso una connessione SSL (HTTPS).

uriPath=: valore di default=””
E' l'URIPath su cui risponde il servizio di health check. Se non presente verrà eseguito un health check con una connessione TCP.

numRetryConnection=: valore di default=”default da <params>”
E' il numero di tentativi di default per dichiarare un servizio non più attivo. Se non valorizzato prende il default dal paragrafo <params>.

waitPerRetryConnection=: valore di default=”300” UM=Millisecondi
Tempo di attesa tra un tentativo di connessione e l'altro fino al raggiungimento di numRetryConnection tentativi. Se non valorizzato prende il default dal paragrafo <params>.

createConnectionTimeOut=: valore di default=”default da <params>”
Tempo di attesa prima di dichiarare timeout un tentativo di connessione. Se non valorizzato prende il default dal paragrafo <params>.

<healthCheckConditionPolicy>

```
<serviceconf>
        <ipncr>
                <floatingAddressesMgr>
                        <floatingAddress>
                                <healthCheckConditionPolicy
```

E' il paragrafo preposto all'health check degli indirizzi per determinare la condizione di funzionamento dell'interfaccia di rete. Se tutti gli health check predisposti in questo paragrafo dovessero dare esito negativo allora l'indirizzo verrà impostato in una delle interfacce di rete prestabilite. E' anche possibile predisporre più gruppi di healtcheck duplicando questo paragrafo come descritto all'inizio di questo capitolo. Questo controllo è condizionale e cioè non determina la migrazione dell'indirizzo da una interfaccia ad un'altra bensì determina la possibilità di impostazione dell'indirizzo floating in base a delle condizioni esterne.

enable=: valore di default=”false”
Abilita disabilita il caricamento e la valutazione del paragrafo

description=: valore di default=””
Descrizione del controllo

positiveCondition=: valore di default=”true”
Cambia il senso del controllo.
Se true e almeno un controllo è risultato positivo allora tutto il paragrafo è positivo.
Se false e almeno un controllo è risultato positivo allora tutto il paragrafo risulta negativo.

<healthCheck>

```
        <serviceconf>
                <ipncr>
                        <floatingAddressesMgr>
                                <floatingAddress>
                                        <healthCheckConditionPolicy>
                                                <healthCheck
```

enable=:default=”false”
Abilita disabilita il caricamento e la valutazione del paragrafo

address=: valore di default=””
E' l'indirizzo su cui viene effettuato l'health check..

port=: valore di default=”0”
E' la porta su cui risponde il servizio di health check. Se <=0 verrà eseguito un check ICMP

SSL=: valore di default=”false”
Se true, esegue l'health check del servizio attraverso una connessione SSL (HTTPS).

uriPath=: valore di default=””
E' l'URIPath su cui risponde il servizio di health check. Se non presente verrà eseguito un health check con una connessione TCP.

numRetryConnection=: valore di default=”default da <params>”
E' il numero di tentativi di default per dichiarare un servizio non più attivo. Se non valorizzato prende il default dal paragrafo <params>.

waitPerRetryConnection=: valore di default=”300” UM=Millisecondi
Tempo di attesa tra un tentativo di connessione e l'altro fino al raggiungimento di numRetryConnection tentativi. Se non valorizzato prende il default dal paragrafo <params>.

createConnectionTimeOut=: valore di default=”default da <params>”
Tempo di attesa prima di dichiarare timeout un tentativo di connessione. Se non valorizzato prende il default dal paragrafo <params>.

24
OPLON®AAI WorkFlow surfaceclusterwf

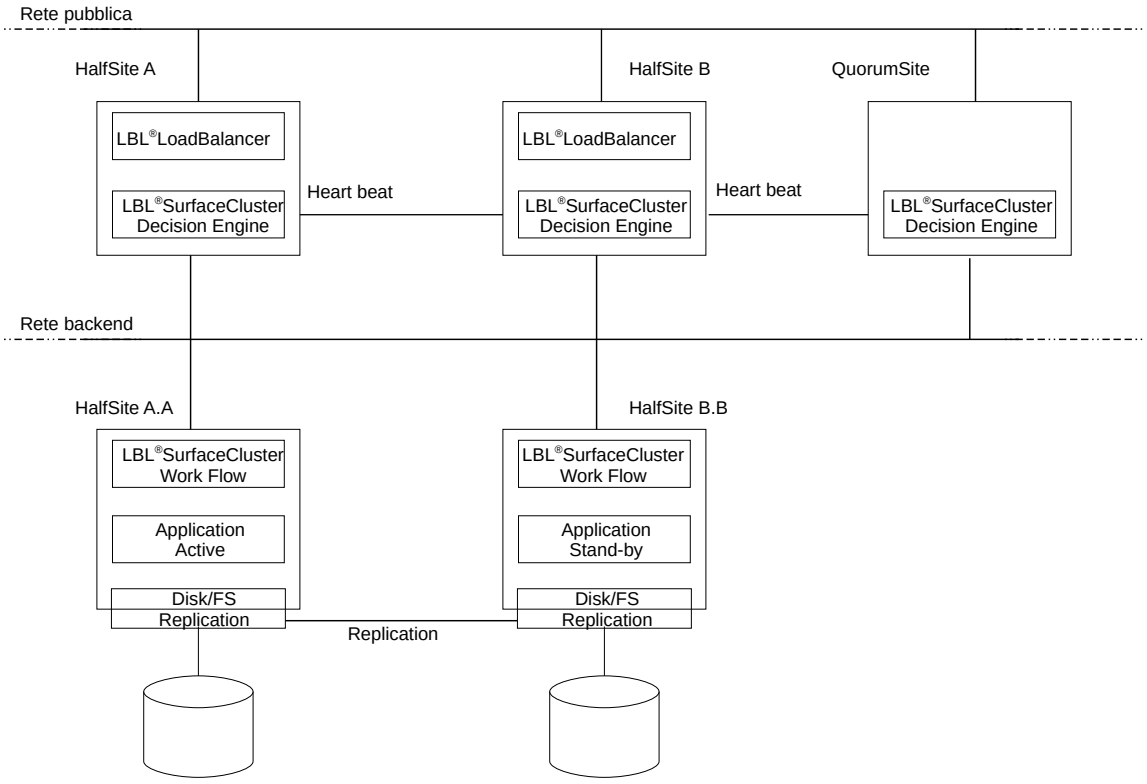
Introduzione

OPLON®Commander introduce un nuovo concetto di alta affidabilità in ambito applicativo andando a ricoprire il ruolo di coordinatore delle attività di un datacenter mission-critical. OPLON®Commander è composto da due moduli principali:

- OPLON®Commander Work Flow
- OPLON®Commander Decision Engine.

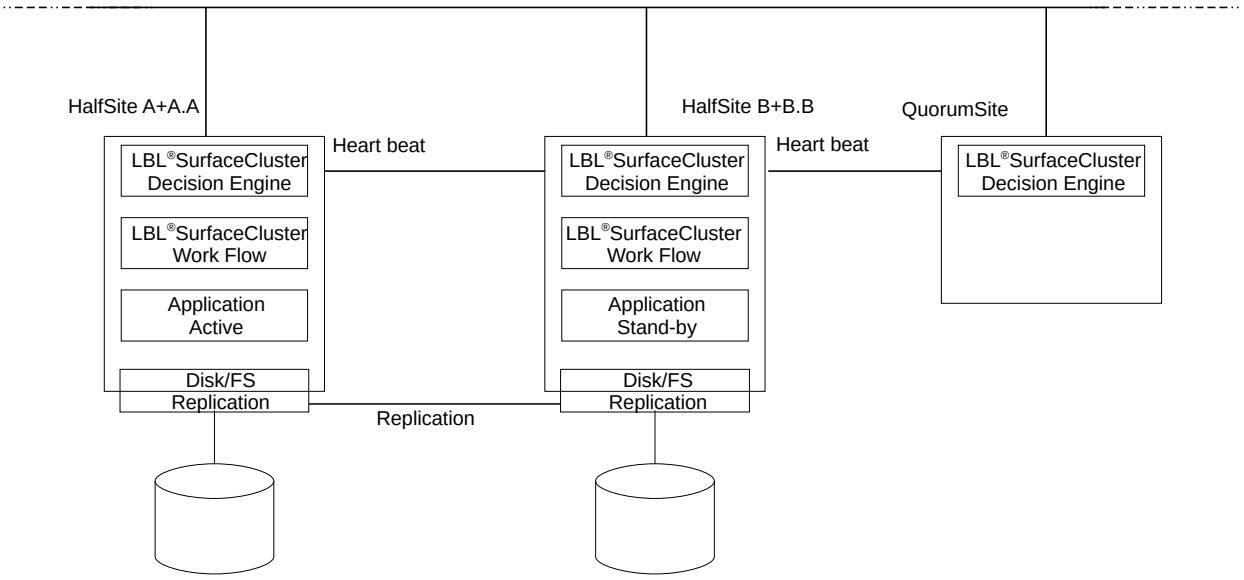
I due moduli sono stati progettati per lavorare in cooperazione, ma nel caso non siano richieste delle operazioni automatiche, è possibile utilizzare solo il componente OPLON®Commander Work Flow.

L'architettura generale può essere riassunta nel seguente schema:

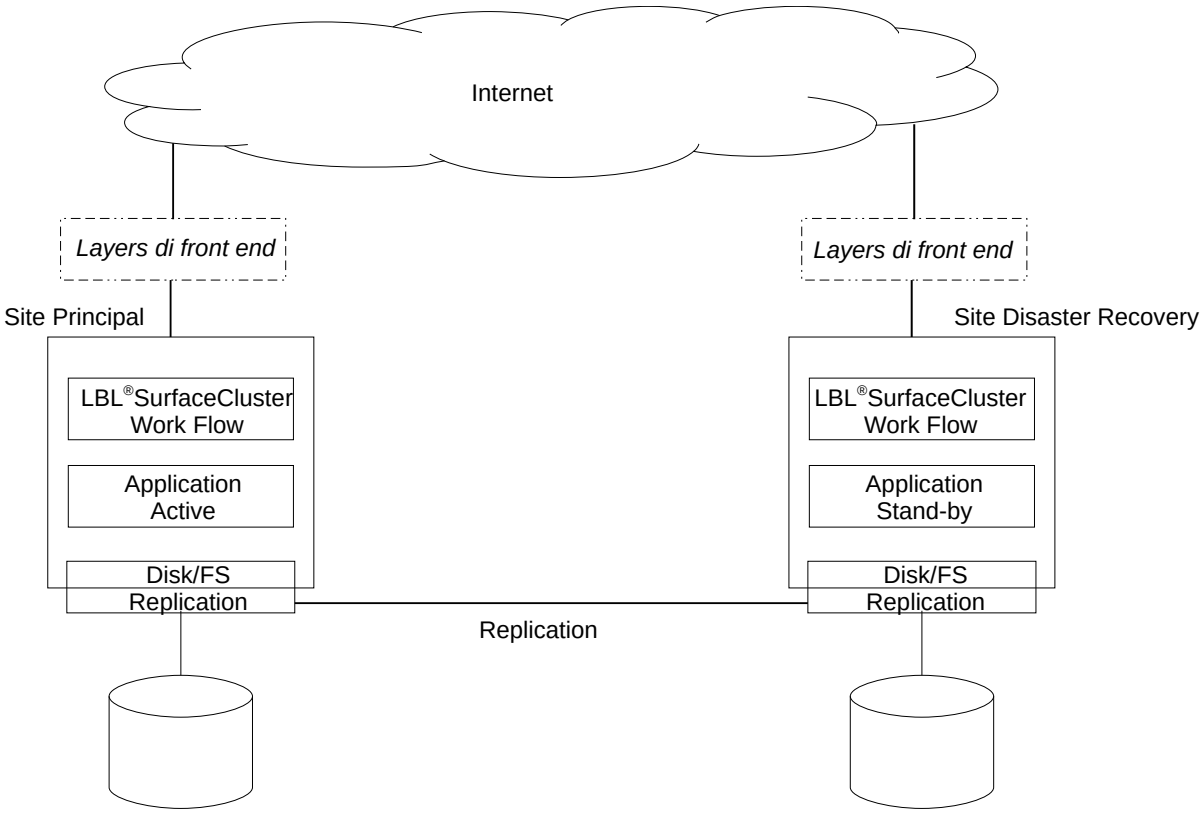


La flessibilità dello strumento può essere espressa in differenti architetture.

Di seguito un altro esempio di architettura possibile:



Di seguito uno scenario di Disaster Recovery:



Singolarmente, il servizio di work flow può essere immaginato come un automatizzatore di procedure. Di fatto il servizio prevede di descrivere all'interno del file XML la logica delle operazioni che verranno poi eseguite in maniera puntuale, e la valutazione del return code permette di verificare il funzionamento dell'operazione e di programmare gli "step" successivi da eseguire.

Il file parametri del servizio di Work Flow descrive quindi le azioni da eseguire e per ogni evento le azioni successive corrispondenti.

OPLON®Commander Work Flow introduce un nuovo paradigma essendo un servizio di Remote Workflow Command (RWC), ed introduce la programmazione strutturata nell'ambito sistemistico.

La struttura del file di configurazione è la seguente

```
<serviceconf>
   <copyright>
   </copyright>
        <surfaceclusterwf>
                    <params>
                    </params>

                    <workflow>
                       <step>
                          <returncode>
                          </returncode>
                       </step>
                        ...
                    </workflow>
                     ...
        </surfaceclusterwf>
</serviceconf>
```

<surfaceclusterwf>

```
<serviceconf>
        <surfaceclusterwf>
```

<params>

```
<serviceconf>
        <surfaceclusterwf>
                <params
```

Il paragrafo contiene i parametri generali di configurazione del servizio di workflow.

address=: valore di default="localhost"
Indirizzo su cui risponde il servizio. Solitamente il servizio normalmente è attestato sulla rete di backend.

port=: valore di default=”54444”
E' la porta su cui risponde il servizio.

backlog=: valore di default=”20”
E' il numero massimo di incoming connections livello socket, dove il Sistema Operativo permette questa impostazione.
Le librerie TCOProject® hanno comunque un proprio sistema di gestione delle richieste di connessione nel caso il Sistema operativo non permettesse di sfruttare questa funzionalità.

reuseAddress=: valore di default=”true”
E' il corrispondente parametro socket SO_REUSEADDR.

concurrentWorkers=: valore di default=”20”
E' il numero iniziale di richieste evadibili contemporaneamente.

maxConcurrentWorkers=: valore di default=”100”
E' il numero massimo di richieste evadibili contemporaneamente.

webAppsDir=: valore di default=”lib/webroot_surfaceclusterwf/webapps”
Directory home applicazione web

webAppsConfDir=: valore di default=”lib/webroot/webappsconf”
Directory configurazione applicazioni web

webSecurityDir=: valore di default=”lib/webroot/websecurity”
Directory configurazione applicazioni web

certificateURL=: valore di default=”certificate/serverkeys”
Se impostato indica l'indirizzo http da cui prelevare il certificato.

keyStore=: valore di default=”JKS”
Indica il tipo di keystore SSL da cui prelevare il certificato. Normalmente se si utilizza il keystore della JVM deve essere impostato a “JKS” se si utilizza un keystore OpenSSL si deve impostare “PKCS12”.

keyStorePassword=: valore di default=”defaultpwd”
Passoword per l'accesso al keystore.

alias=: valore di default=”lblcert”
E' l'identificativo del certificato all'interno del keystore.

aliasPassword=: valore di default=”defaultpwd”
E' la password per poter accedere al certificato contenuto nel keystore.

keyManagerFactory=: valore di default=”SunX509”
Indica il modulo di interpretazione del certificato. Normalmente impostato a “SunX509”

SSLContextVersion=: valore di default=”SSLv3”

Indica la versione del protocollo SSL. Normalmente impostato a “SSLv3” per JVM keystore oppure “TLS” per OpenSSL.

healthCheckUriPath=: valore di default=”/HealthCheck”
E' il path di healthcheck di attività. Questo valore normalmente non viene mai modificato a meno di uso già presente in altre applicazioni.

workFlowCommandUriPath=: valore di default=”/SCWFCommand”
E' il path di al quale risponderà il web service di gestione del Remote Work Flow Command (RWC d'ora in poi).

dateFormat=:default=”dd/MM/yyyy HH:mm:ss:SSSS”
E' il formato della data dei dati statistici provenienti dai web service.

delimiter=:default=”|” (pipe)
E' il carattere che delimita i campi durante lo scambio di informazioni.

maxRecords=:valore di default=”500” UM=Record
Numero massimo di record in risposta ad una richiesta Web Service.

sysCommandTimeOut=:valore di default=”10000” UM=Millisecondi
Indica il tempo necessario per dichiarare in time-out un comando di sistema. Se il comando oltrepassa questo limite viene eseguito un comando di abort e successivamente viene rilasciato il controllo all'applicazione.

sysCommandCheckRate=:valore di default=”300” UM=Millisecondi
E' la frequenza del controllo dello stato del comando di sistema.

sysCommandRemoteURL=:default=”https://localhost:5992/SysCommand”
E' l'URL del servizio per eseguire i comandi di sistema

exclusiveRunWorkFlow=:valore di default=”false” UM=boolean
Questo valore indica se deve essere eseguito un Work Flow alla volta e solo al termine (@STOP_RUN) del workflow potrà essere avviato un nuovo Work Flow.

surfaceClusterWFCommandDir=:valore di default=”surfaceClusterWFCommandDir”
E' la directory base dove andranno cercati gli script/programmi se non diversamente specificato, con un path assoluto, su ogni “step” di lavoro. Se non indicato un indirizzo assoluto il tutto sarà riferito a (LBL_HOME)/procsProfile/XXX_procName/..

<workflow>

```
<serviceconf>
        <surfaceclusterwf>
                <workflow
```

enable=:default=”true”
Se true questo paragrafo è attivo. Se false il paragrafo non viene preso in considerazione

durante il caricamento iniziale.

name=:valore di default=””
E' il nome del work flow. Il nome deve essere univoco a livello di questo file XML ed identifica un insieme di “step”.

description=:valore di default=”workflow: (name)”
E' la descrizione del work flow. Inserire una descrizione breve ma significativa in quanto apparirà poi nelle visualizzazioni.

startName=:valore di default=”primo <step> in sequenza”
E' il primo step ad essere eseguito. Se non specificato prende il primo <step> in sequenza.

startType=:valore di default=”manual”
Indica se il WorkFlow deve essere avviato all'avvio del servizio.
Se “manual” il servizio deve essere avviato: o manualmente dall'operatore, o da un OPLON®Commander Decision Engine.
Se “automatic” il servizio partirà automaticamente all'avvio del servizio.

requiresHumanDecision=:valore di default=”false”
Indica se il work flow una volta avviato avrà un'evoluzione automatica o sarà necessario un intervento umano per avviare uno step alla volta. Può essere utilizzato anche per non far proliferare il numero di work flow con un singolo <step> e mantenere una libreria di componenti.

<step>

```
<serviceconf>
        <surfaceclusterwf>
                <workflow>
                        <step
```

enable=:default=”true”
Se true questo paragrafo è attivo. Se false il paragrafo non viene preso in considerazione durante il caricamento iniziale.

name=:default=””
Nome dello step. Questo parametro è il riferimento univoco in questo paragrafo <workflow>.

description=:default=”workflow: (<workflow name>)-(name)”
Descrizione dello step. Inserire una descrizione sintetica ma esaustiva dell'azione relativa a questo step.

commitWorkFlow=:default=”false” UM=boolean
Indica se questo step è da considerarsi conclusivo.
Può essere utilizzato per segnalare che lo step è in loop su se stesso perché si vuole impegnare quel particolare workflow non dando la possibilità di essere nuovamente utilizzato fino al prossimo riavvio del sistema. Questo flag se impostato a true, ed in presenza di un loop su se stesso con risultato positivo, non esegue ulteriori log evitando di

intasare i log facilitandone la lettura.

waitBeforeExecute=:default=”0” UM=Millisecondi
Tempo di attesa prima di eseguire il comando.

sysCommandTimeOut=:default=”1000” UM=Millisecondi
Tempo di attesa prima di dichiarare il comando non a buon fine.
Questo valore è molto importante perché un comando potrebbe durare anche per ore o giorni, si pensi ad esempio ad un restart di un database con molti terabyte in linea. E' un valore quindi che deve essere sempre ben meditato. Ovviamente se il comando terminasse prima di questo valore l'elaborazione procederebbe.

evaluateReturnCode=:default=”true” UM=Boolean
Indica se deve essere valutato il return code.
Se “false” qualsiasi risultato del comando verrà valutato come positivo. Se “true” verranno valutati i return code delle operazioni. In assenza dei paragrafi <returncode> il returncode 0 (zero) verrà valutato come risultato positivo altrimenti qualsiasi altro valore verrà interpretato come responso negativo.
In presenza anche di un solo paragrafo <returncode> tutti i return code verranno interpretati come responso negativo se non diversamente indicato. Quindi se non venisse indicato il return code 0 (zero) questi verrebbe interpretato come negativo.

command=:default=””
E' il comando che verrà eseguito in questo step.
Può essere indicato un qualsiasi eseguibile (batch compresi .sh/bash etc. oppure .bat nel mondo Microsoft Windows). Se il comando non è indicato con un path assoluto viene aggiunto quanto specificato nel parametro “surfaceClusterWFCommandDir” del paragrafo <params> .
E' possibile definire in questo parametro un comando interno di RWC. Per effettuare un RWC è sufficiente inserire:

- @RWC hostName=localhost workFlow=takeControl

Questa keyword riservata permette di propagare un RWC iniziale su più sistemi andando quindi ad effettuare operazioni articolate su più layer infrastrutturali.

Il comando completo con tutti i parametri è il seguente:
@RWC Remote Workflow command
hostname=obbligatorio
portNumber=facoltativo, default 54444
uriPath=facoltativi, default /SCWFCommand
workFlow=obbligatorio
step=facoltativo, default ""
command=facoltativo, default ""
frhd=facoltativo, default false

Il parametro command può assumere i seguenti valori visualizzabili anche attraverso connessione tramite browser.
es.:

```
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=Numero intero
A fronte di un return code negativo o interpretato per negativo lo step verrà ripetuto per un numero di volte pari a questo parametro. Come si vedrà in seguito nel paragrafo <returncode> è possibile cambiare per eccezione questo valore, a fronte di un particolare return code o annullarlo completamente.

gotoWhenTrue=:default=”next step”
Se il comando ha dato esito positivo e se non indicato diversamente nel paragafo <returncode>, questo valore indica lo step al quale il work flow punterà per eseguire il prossimo step.
Se omesso in automatico verrà assunto come prossimo step quello successivo.

gotoWhenFalse=:default=”next step”
Se il comando ha dato esito negativo e se non indicato diversamente nel paragafo <returncode> questo valore indica lo step al quale il work flow punterà per eseguire il prossimo step.
Se omesso in automatico verrà assunto come prossimo step quello successivo.

<returncode>

```
<serviceconf>
    <surfaceclusterwf>
        <workflow>
            <step>
                <returncode
```

enable=:default=”true”
Se true questo paragrafo è attivo. Se false il paragrafo non viene preso in considerazione durante il caricamento iniziale.

value=:default=”0” UM=Numero intero
E' valore del return code che verrà preso in considerazione per questo step. Questo valore non può essere duplicato nell'ambito dello stesso step.

description=:default=”returncode: (<step> name-value”
Descrizione del return code. Questa descrizione deve essere sintetica ma esauriente per fornire una traccia valida durante la visualizzazione degli step e per una corretta interpretazione umana nel tempo del risultato che si vuole ottenere.

result=:default=”true” UM=Boolean
Se true il valore di ritorno verrà interpretato come positivo e quindi verranno innescati gli step di ritorno positivo se non altrimenti indicato in questo paragrafo.

gotoStep=:default=”next step”
Se esistente indica lo step a cui il work flow punterà per questo return code. Se non indicato e il result è considerato positivo punterà al prossimo step. In ogni caso lo step sarà ripetuto per maxRetry nel paragrafo <step> se non diversamente indicato su questo stesso paragrafo nel parametro retry.

retry=:default=”true” UM=Boolean
Se true l'operazione verrà ripetuta per maxRetry del paragrafo <step> ovviamente nel caso il parametro result di questo return code fosse impostato come risultato negativo “false”. Se false in ogni caso non verranno eseguiti ulteriori tentativi.

waitBeforeRetry=:default=”0” UM=Millisecondi
E' il tempo di attesa prima di ritentare l'operazione se previsto dal parametro retry

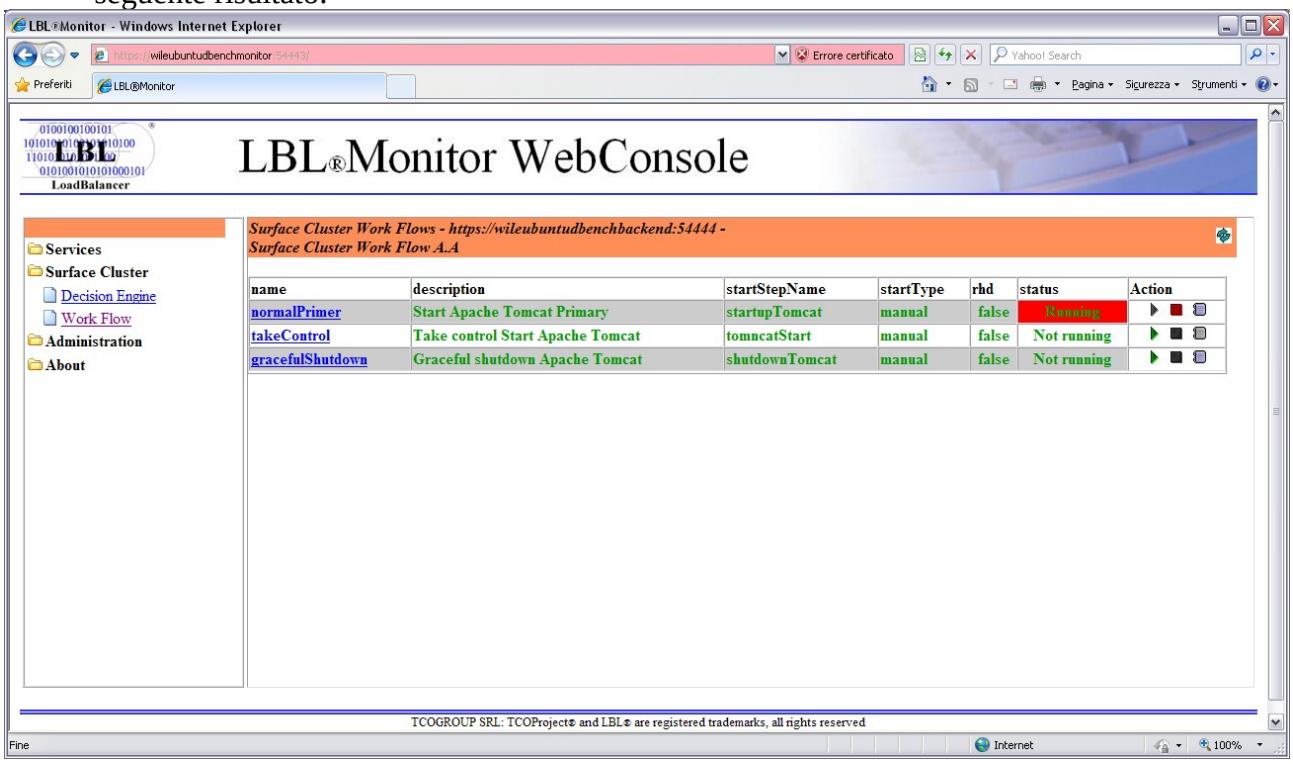
Esempio:start Tomcat

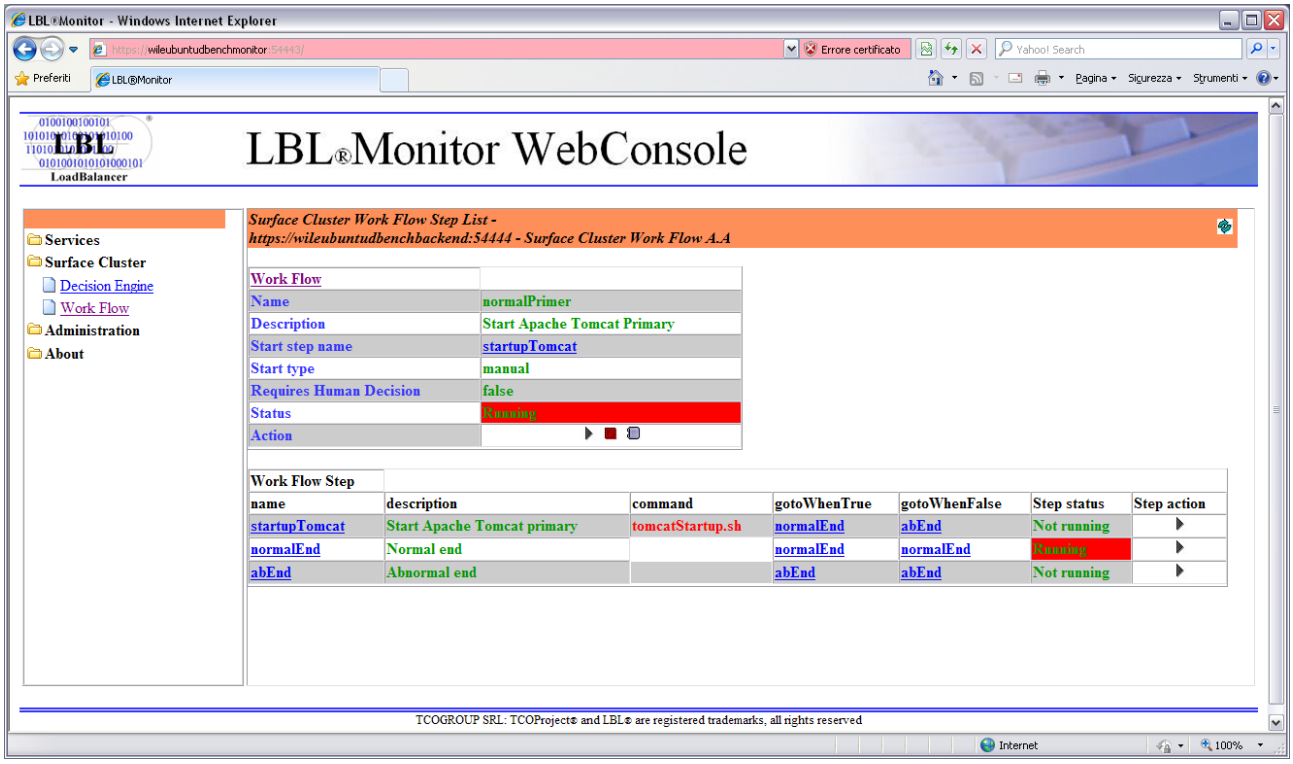
Di seguito un esempio di configurazione di un Work Flow di minima dove viene eseguito lo start di una istanza Tomcat.

```
...
  OPLON(r)ADC 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"
```

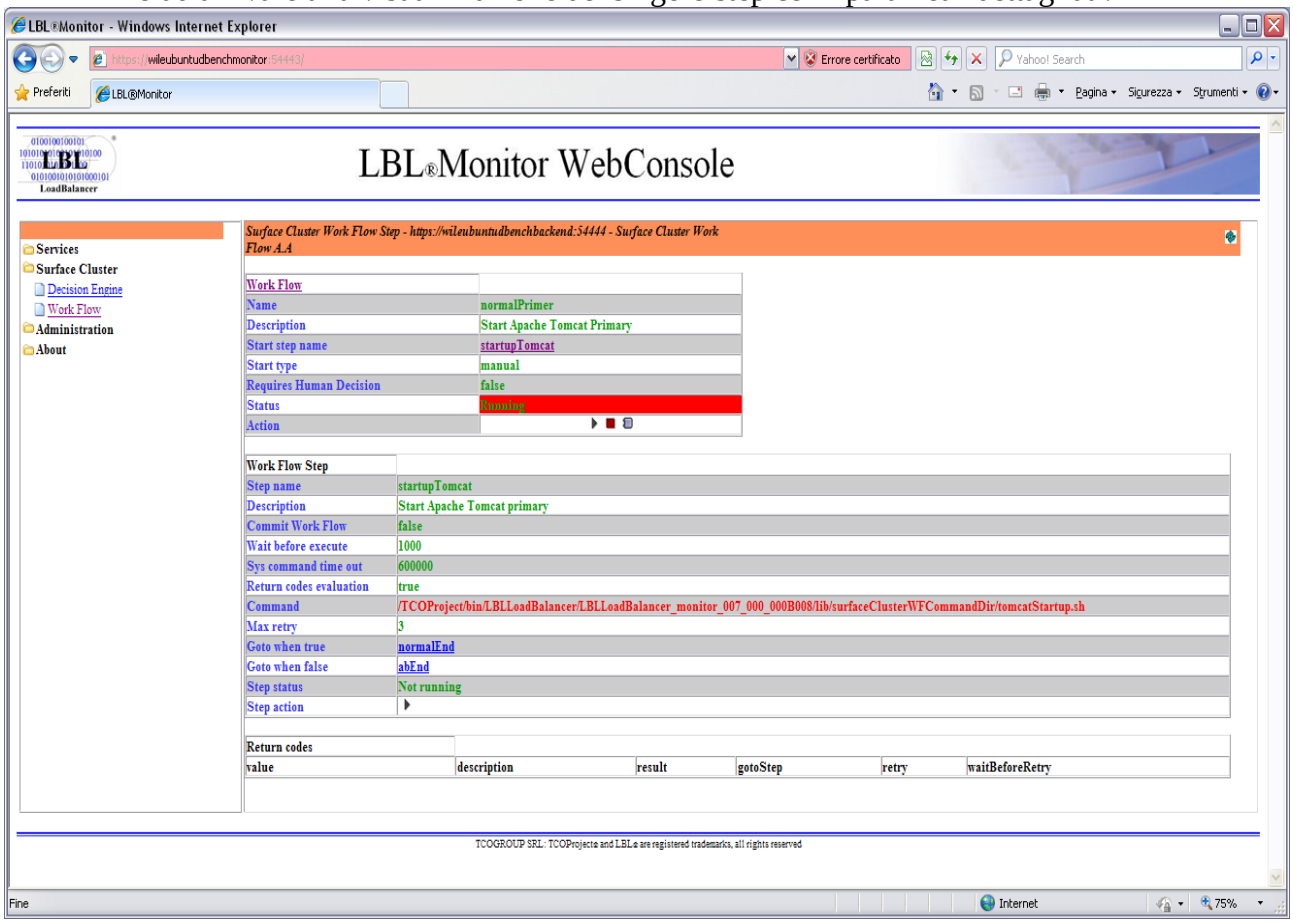
```
            commitWorkFlow="true"
            name="normalEnd"
            description="Normal end"
            waitBeforeExecute="10000"
            evaluateReturnCode="false"
            gotoWhenTrue="normalEnd"
            gotoWhenFalse="normalEnd">
        </step>

        <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"
        ...
            ...
```

La visualizzazione attraverso OPLON®Monitor del Work Flow sopra esposto darà il seguente risultato:





Fino ad arrivare alla visualizzazione del singolo step con i parametri dettagliati:



25
OPLON® WorkFlow Remote Batch

Questo servizio mette a disposizione un esecutore remoto sicuro di batch o eseguibili.

```
<serviceconf>
   <copyright>
   </copyright>
        <remotebatch>
                <params>
                </params>
        </remotebatch>
</serviceconf>
```

<params>

```
<serviceconf>
        <remotebatch>
                <params>
```

address=: valore di default=”localhost”
Il valore, impostato di default a localhost, dovrebbe essere impostato con la rete di monitoring. In mancanza di una rete di monitoring si consiglia di impostarlo con la rete di backend.

port=: valore di default=”5994”
E' la porta su cui risponde il servizio.

backlog=: valore di default=”20”
E' il numero massimo di incoming connections livello socket dove il Sistema Operativo permette questa impostazione.
OPLON®ADC ha comunque un suo sistema di gestione delle richieste di connessione nel caso il Sistema operativo non permettesse di sfruttare questa funzionalità.

concurrentWorkers=: valore di default=”20”
E' il numero iniziale di richieste evadibili contemporaneamente dalla WebConsole.

maxConcurrentWorkers=: valore di default=”100”
E' il numero massimo di richieste evadibili contemporaneamente dalla WebConsole.

healthCheckContextPath=: valore di default=”/HealthCheck”
E' il path di healthcheck di attività. Questo valore normalmente non viene mai modificato a meno di uso già presente in altre applicazioni.

webAppsDir=: valore di default=”lib/webroot_remotebatch/webapps”
Directory home applicazione web

webAppsConfDir=: valore di default=”lib/webroot/webappsconf”
Directory configurazione applicazioni web

webSecurityDir=: valore di default=”lib/webroot/websecurity”
Directory configurazione applicazioni web

certificateURL=: valore di default=”certificate/serverkeys”
Se impostato indica l'indirizzo http da cui prelevare il certificato.

keyStore=: valore di default=”JKS”
Indica il tipo di keystore SSL da cui prelevare il certificato. Normalmente se si utilizza il keystore della JVM deve essere impostato a “JKS” se si utilizza un keystore OpenSSL si deve impostare “PKCS12”.

keyStorePassword=: valore di default=”defaultpwd”
Password per l'accesso al keystore.

alias=: valore di default=”lblcert”
E' l'identificativo del certificato all'interno del keystore.

aliasPassword=: valore di default=”defaultpwd”
E' la password per poter accedere al certificato contenuto nel keystore.

keyManagerFactory=: valore di default=”SunX509”
Indica il modulo di interpretazione del certificato. Normalmente impostato a “SunX509”

SSLContextVersion=: valore di default=”SSLv3”
Indica la versione del protocollo SSL. Normalmente impostato a “SSLv3” per JVM keystore oppure “TLS” per OpenSSL.

File di profilo lancio eseguibili e batch

OPLON®Cluster Remote Batch interpreta i file di profilo contenuti nella directory

- (LBL_HOME)/lib/webroot_remotebatch/webapps/RemoteBatch

Il file di profilo di lancio deve avere il seguente formato:

```
<startBatch>
        <params remoteBatchTimeOut="10000"
                remoteBatchCheckRate="300"
                allowURIParams="false" debug="false"/>
```

```
        <legacyCommand>T:\work0\tmp\vai.bat 11 </legacyCommand>
</startBatch>
```

Essendo interpretato al momento del richiamo, esattamente come una pagina HTML, i valori potranno essere cambiati dinamicamente a caldo.

Il nome del file deve essere uguale al paragrafo principale.
Nel caso sopra descritto il file si dovrà chiamare startBatch ed avere una estensione come: xml; txt, etc. (mime type text/html).

I parametri sono pochissimi e abbastanza intuitivi.
Di seguito una descrizione dettagliata.

Parametri File di profilo

remoteBatchTimeOut=: valore di default "10000"
E' il tempo di attesa prima di rilasciare il processo batch/eseguibile.

remoteBatchCheckRate=: valore di default “300”
E' il tempo di attesa tra un tentativo e l'altro dopo aver aspettato remoteBatchTimeOut. Dopo 3 tentativi il lancio del batch viene dichiarato fallito.

allowURIParams=:valore di default "false"
Se true permette di inserire ulteriori parametri da URI.
Il default è false per motivi di sicurezza. Se non impostato o impostato a false eventuali parametri impostati nell'URIPath verranno ignorati.

debug=: valore di default "false"
Se true esegue un log warning con i valori di start del processo e del risultato ottenuto.

<legacyCommand>=: valore di default "null"
E' il batch da lanciare. Per motivi di sicurezza questo valore non può essere vuoto.

Il lancio del batch avverrà tramite richiesta di una URL che riporti il nome del file descrittore da interpretare per il lancio del batch/eseguibile.
E' possibile indicare ulteriori parametri durante il lancio del programma nella URL stessa come nell'esempio di seguito:

- http://localhost:5994/RemoteBatch/startBatch.txt?parameters=2222 3333 4444

In questo caso il lancio del batch sara' cosi' composto nella parte parametri:
T:\work0\tmp\vai.bat 11 2222 3333 4444

ATTENZIONE: Il contenuto dei file batch e le varie composizioni dei parametri sono a cura del batch stesso. Il servizio LBL non controllerà la correttezza del batch o varie composizioni dei parametri che saranno completamente a cura dell'implementatore

26
OPLON® DecisionEngine surfaceclusterde.xml

Il modulo OPLON®Commander Decision Engine è il motore della soluzione Surface Cluster. Il suo scopo è verificare costantemente lo stato delle applicazioni e a fronte di criticità innescare processi (Work Flow) a risoluzione del problema.

La caratteristica principale della soluzione OPLON®Commander è proprio il disaccoppiamento tra decisione automatica e risoluzione del problema. Questa caratteristica rende flessibilissima la soluzione e permette alle persone preposte di tenere sempre sotto controllo la situazione ed eventualmente intervenire con sicurezza anche manualmente agendo direttamente sul modulo di Work Flow.

OPLON®Commander Decision Engine attraverso Health Check incrociati sulla rete pubblica, rete di backend, rete di Heart Beat e vitalità applicativa, è in grado di restituire un quadro complessivo della situazione che gli permette di prendere delle decisioni.

I parametri che andremo a descrivere servono proprio a dare gli elementi fondamentali per poter discriminare una situazione critica da un falso problema. Volutamente la parametrizzazione è stata studiata per essere nel contempo semplice ma efficace dovendo indicare solo i dati essenziali allo scopo.

Lo schema che riportiamo di seguito distingue 4 zone di intervento racchiuse nel paragrafo <decisionEngine>

- < decisionEnginesPeers>
- < healthCheckServicesPolicy>
- < healthCheckPublicPolicy>
- < healthCheckBackendPolicy>

Questi quattro paragrafi rappresentano il panorama che permette al Decision Engine di trarre delle conclusioni ed attuare eventuali interventi di ripristino dell'operatività applicativa.

```
<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>
```

Andremo ora a particolareggiare ogni singolo paragrafo e ogni parametro all'interno del paragrafo. Molti valori sono proposti di default e quindi il risultato finale sarà molto semplice da comporre lasciando ovviamente la strada aperta per le particolarizzazioni implementative che si dovessero presentare.

<decisionEnginesPeers> Racchiude i parametri che servono a dare alla singola istanza OPLON®Commander Decision Engine la conoscenza dell'esistenza delle altre istanze decision engine che concorrono a prendere le decisioni. Ricordiamo che il minimo per garantire una decisione affidabile è 3 (tre) istanze OPLON®Commander Decision Engine.

<healthCheckServicesPolicy> Contiene i profili dei servizi da tenere in osservazione e da attivare in caso di necessità. La sequenza dei profili applicativi determina anche la priorità del servizio rispetto i corrispettivi servizi in stand-by.

<healthCheckPublicPolicy> Contiene le politiche di health check per decretare il funzionamento della rete pubblica. Un buon campione di health check non può essere inferiore a 3 (tre) health check.

<healthCheckBackendPolicy> Contiene le politiche di health check per decretare il funzionamento della rete di backend. Un buon campione di health check non può essere inferiore a 3 (tre) health check. E' utile inserire nel campione anche le istanze OPLON®Commander Work Flow in modo da verificare costantemente anche la raggiungibilità e operatività del servizio che permetterà di intraprendere le azioni correttive.

Un esempio di file di configurazione di un'istanza OPLON®Commander Decision Engine può essere sintetizzata in un'unica pagina dove gli elementi da completare sono veramente molto pochi essendo o già precompilati o di 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"
                                      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
```

Parametri generali del servizio

frequency=: valore di default=”10000” UM=Millisecondi
E' la frequenza di verifica cambiamenti di stato.

address=: valore di default=”localhost”
Il valore, impostato di default a localhost. Il servizio normalmente sarà attestato sulla rete di monitor.

port=: valore di default=”54445”
E' la porta su cui risponde il servizio.

addressHeartBeat=: valore di default=”address”
E' l'indirizzo di HeartBeat. Il Decision Engine può essere raggiunto quindi attraverso due listeners. Questo listener risponderà all'indirizzo attestato sull'HeartBeat.

portHeartBeat=: valore di default=”port+1 (es 54446)”
E' la porta su cui risponde il servizio attestato sulla rete di HeartBeat.

backlog=: valore di default=”20”
E' il numero massimo di incoming connections livello socket dove il Sistema Operativo permette questa impostazione.
Le librerie TCOProject® hanno comunque un proprio sistema di gestione delle richieste di connessione nel caso il Sistema operativo non permettesse di sfruttare questa funzionalità.

reuseAddress=: valore di default=”true”
E' il corrispondente parametro socket SO_REUSEADDR.

concurrentWorkers=: valore di default=”20”
E' il numero iniziale di richieste evadibili contemporaneamente.

maxConcurrentWorkers=: valore di default=”100”
E' il numero massimo di richieste evadibili contemporaneamente.

webAppsDir=: valore di default=”lib/webroot_surfaceclusterde/webapps”
Directory home applicazione web

webAppsConfDir=: valore di default=”lib/webroot/webappsconf”
Directory configurazione applicazioni web

webSecurityDir=: valore di default=”lib/webroot/websecurity”
Directory configurazione applicazioni web

certificateURL=: valore di default=”/certificate/serverkeys”
Se impostato indica l'indirizzo http da cui prelevare il certificato.

keyStore=: valore di default=”JKS”
Indica il tipo di keystore SSL da cui prelevare il certificato. Normalmente se si utilizza il keystore della JVM deve essere impostato a “JKS” se si utilizza un keystore OpenSSL si deve impostare “PKCS12”.

keyStorePassword=: valore di default=”defaultpwd”
Password per l'accesso al keystore.

alias=: valore di default=”lblcert”
E' l'identificativo del certificato all'interno del keystore.

aliasPassword=: valore di default=”defaultpwd”
E' la password per poter accedere al certificato contenuto nel keystore.

keyManagerFactory=: valore di default=”SunX509”
Indica il modulo di interpretazione del certificato. Normalmente impostato a “SunX509”

SSLContextVersion=: valore di default=”SSLv3”
Indica la versione del protocollo SSL. Normalmente impostato a “SSLv3” per JVM keystore oppure “TLS” per OpenSSL.

healthCheckUriPath=: valore di default=”/HealthCheck”
E' il path di healthcheck di attività. Questo valore normalmente non viene mai modificato a meno di uso già presente in altre applicazioni.

decisionEngineCommandUriPath=: valore di default=”/SCDECommand”
E' il path di al quale risponderà il web service di gestione del Web Service di gestione.

sysCommandTimeOut=:valore di default=”10000” UM=Millisecondi
Indica il tempo necessario per dichiarare in time-out un comando di sistema. Se il comando oltrepassa questo limite viene eseguito un comando di abort e successivamente viene rilasciato il controllo all'applicazione.

sysCommandCheckRate=:valore di default=”300” UM=Millisecondi
E' la frequenza del controllo dello stato del comando di sistema.

sysCommandRemoteURL=:default=”https://localhost:5992/SysCommand”
E' l'URL del servizio per eseguire i comandi di sistema

dateFormat=:default=”dd/MM/yyyy HH:mm:ss:SSSS”
E' il formato della data dei dati statistici provenienti dai web service.

delimiter=:default=”|” (pipe)
E' il carattere che delimita i campi durante lo scambio di informazioni.

maxRecords=:valore di default=”500” UM=Record
Numero massimo di record in risposta ad una richiesta Web Service.

maxSizeInBuffer=:valore di default=”10485760” UM=Bytes
Numero massimo di bytes in risposta ad una richiesta Web Service.

statusDir=:valore di default=”lib/surfaceClusterDEStatus”
E' la directory di persistenza degli stati dei servizi (non alterare a mano, viene gestita automaticamente).

notificationDir=:valore di default=”lib/notificationDir”
E' la directory di persistenza degli stati verificata da OPLON®ADC per forzare l'Out Of Order di un servizio posto in bilanciamento.
OPLON®Commander Decision Engine notifica anche su questa directory nel formato interpretato da OPLON®ADC l'Out Of Order del servizio.

numRetryConnection=: valore di default=”3”
E' il numero di tentativi di default per dichiarare un servizio non più attivo.

createConnectionTimeOut=: valore di default=”5000” UM=Millisec.
Time-out del tentativo di connessione.

<decisionEngineMgr>

```
<serviceconf>
    <surfaceclusterde>
        <decisionEngineMgr
```

Questo paragrafo contiene l'elenco di tutti i motori Decision Engine. E' quindi possibile associare ad una stessa istanza più motori di decisione indipendenti

<decisionEngine>

```
<serviceconf>
    <surfaceclusterde>
        <decisionEngineMgr>
            <decisionEngine
```

Questo paragrafo descrive i parametri relativi ad un singolo motore di decisione. In una istanza OPLON®Commander Decision Engine possono essere contenuti più motori indipendenti di decisione.

enable=:default=”true” UM=boolean
Abilita o disabilita l'interpretazione di questo paragrafo nell'istanza.

groupName=: valore di default=”SCDEGroup”
E' il nome del gruppo di Decision Engine. E' importantissimo in quanto i motori di decisione contenuti su un'istanza si distinguono da questo nome.

description=: valore di default=”description: groupName”
E' la descrizione di questo decision engine. Deve essere sintetica ma esaustiva.

frequency=: valore di default=”<params frequency>” UM=Millisecondi
E' la frequenza di verifica cambiamenti di stato. Se non specificata viene assunta la frequenza del paragrafo <params>

firstThinkingTime=: valore di default=”45000” UM=Millisecondi
E' il tempo di attesa di inizializzazione e prime verifiche. Superato questo tempo di attesa il motore decisionale evidenzierà un messaggio. Il motore decisionale non procede comunque anche oltrepassato questo tempo iniziale e attende la totale inizializzazione degli stati.

applicationLostTimeBeforeRestart=:default=”1/2 di applicationLostTime”
UM=Millisecondi
Tempo di attesa prima di dichiarare persa l'applicazione e provare un restart, se previsto. Se viene raggiunto il quorum di switch viene comunque atteso questo tempo prima di eseguire il restart della risorsa definitivamente in avaria. Se trascorso questo tempo la risorsa torna UP non viene intrapresa nessuna azione. L'evento viene registrato su log file, eventualmente segnalato tramite e-mail o post HTTP.
Default 1/2 di applicationLostTime (15 secondi). Se impostato a > di applicationLostTime non viene eseguito nulla.

operator=:default=”OR”
Operatore booleano applicato ai servizi in healthcheck. Se “OR” anche se un solo servizio è in avaria vengono attivate le procedure di switch, se “AND” tutti i servizi devono essere in avaria per attivare le procedure di restart.

applicationLostTime=: valore di default=”30000” UM=Millisecondi
E' il tempo di attesa da quando l'applicazione in verifica è stata dichiarata in stato di down. Questo è un valore molto importante perché superato questo periodo di tempo e l'applicazione risultasse ancora non raggiungibile e verificato il quorum di switch si innescherà la procedura di recovery. I 30” sono il minimo per evitare falsi positivi.

numRetryConnection=: valore di default=”<params numRetryConnection>”
E' il numero di tentativi di default per dichiarare un servizio non più attivo.

createConnectionTimeOut=: valore di default=”<params createConnectionTimeOut>”
UM=Millisecondi
Time-out del tentativo di connessione.

<decisionEnginesPeers>

```
<serviceconf>
    <surfaceclusterde>
        <decisionEngineMgr>
            <decisionEngine>
```

```
<decisionEnginesPeers
```

Il paragrafo contiene le informazioni relative alle altre istanze paritetiche di OPLON®Commander Decision Engine. Il numero minimo di decision engine è di 3 per poter decretare il quorum di swicth.

<peer>

```
<serviceconf>
    <surfaceclusterde>
        <decisionEngineMgr>
            <decisionEngine>
                <decisionEnginesPeers>
                    <peer
```

Questo paragrafo contiene le informazioni di un servizio Decison Engine paritetico.

enable=:default=”true” UM=boolean
Abilita o disabilita l'interpretazione di questo paragrafo nell'istanza.

URL=:default=”” UM=URL W3c
E' l'URL di connessione al servizio paritetico attraverso la rete di HeartBeat.

description=:default=”description peer: URL”
descrizione sintetica ma esaustiva del servizio.

healthCheckUriPath=:default=”<params healthCheckUriPath>”
URIpath che associata all'URL determina il servizio di Health Check dell'istanza paritetica.

decisionEngineCommandUriPath=:default=”<params decisionEngineCommandUriPath>”
URIpath che associata all'URL determina il servizio di Health Check dell'istanza paritetica.

numRetryConnection=: valore di default=”<decisionEngine numRetryConnection>”
E' il numero di tentativi di default per dichiarare un servizio non più attivo.

createConnectionTimeOut=: valore di default=”< decisionEngine createConnectionTimeOut>” UM=Millisecondi
Time-out del tentativo di connessione.

addressProxy=: valore di default=””
E' l'indirizzo del proxy su cui viene effettuata la connect di health check.

ATTENZIONE: non impostare questo valore con un nome ma impostarlo con un indirizzo (es.: 192.168.43.142).

portProxy=: valore di default=”0”
E' la porta del proxy di health check.

<healthCheckServicesPolicy>

```
<serviceconf>
```

```
<surfaceclusterde>
    <decisionEngineMgr>
        <decisionEngine>
            <healthCheckServicesPolicy
```

Questo paragrafo contiene le informazioni generali di un servizio applicativo per poterne verificare la funzionalità e quindi intraprendere le azioni di recovery. La successione di attivazione a fronte di un evento di failure avviene in base alla sequenza di inserimento nel file xml.

description=:default=""
Descrizione generale del servizio posto in alta affidabilità.

operator=:default="<decisionEngine sameParam>"
Operatore booleano applicato ai servizi in healthcheck. Se "OR" anche se un solo servizio è in avaria vengono attivate le procedure di switch, se "AND" tutti i servizi devono essere in avaria per attivare le procedure di restart.

waitTimeAfterNormalPrimer=:default="180000" UM=Millisecondi
E' il tempo di attesa dopo aver avviato il primo servizio disponibile (normal startup). Superato questo tempo il decision engine comincerà a verificare se il servizio è arrivato allo stato di attività. In caso contrario verranno avviate le procedure di recovery. Se l'applicazione va in up prima di questo valore il Decision Engine si mette immediatamente in verifica.

waitTimeAfterFlagBroken=:default="60000" UM=Millisecondi
E' il tempo di attesa dopo l'impostazione del flag persistente per la propagazione agli altri peer.

waitTimeBeforeTakeControl=:default="180000" UM=Millisecondi
E' il tempo di attesa prima di effettuare il take control. Serve a lasciare il tempo per un eventuale graceful shutdown del servizio precedentemente attivo se la risorsa è ancora raggiungibile.

waitTimeAfterTakeControl=:default="900000" (15') UM=Millisecondi
E' il tempo di attesa dopo il take control. Serve ad aspettare la conclusione del completo take control prima di ritornare a verificare lo stato di attività e quindi decidere se è andato a buon fine oppure ritentare con altra risorsa se disponibile. Se l'applicazione va in up prima di questo valore il Decision Engine si mette immediatamente in verifica. Questo tempo è variabile in dipendenza del tipo di recovery e dalla quantità di dati se presente un database.

applicationLostTime=:default="<decisionEngine applicationLostTime>" UM=Mill.
E' il tempo di attesa da quando l'applicazione in verifica è stata dichiarata in stato di down. Questo è un valore molto importante perché superato questo periodo di tempo e l'applicazione risultasse ancora non raggiungibile e verificato il quorum di switch si innescherà la procedura di recovery. I 30" sono il minimo per evitare falsi positivi.

applicationLostTimeBeforeRestart=:default="<decisionEngine applicationLostTimeBeforeRestart>" UM=Millisecondi

Tempo di attesa prima di dichiarare persa l'applicazione e provare un restart se previsto. Se viene raggiunto il quorum di switch viene comunque atteso questo tempo prima di eseguire il restart della risorsa definitivamente in avaria. Se trascorso questo tempo la risorsa torna UP non viene intrapresa nessuna azione, l'evento viene registrato su log file, eventualmente segnalato tramite e-mail o post HTTP.

surfaceClusterWorkFlowCommandUriPath=:default=”/HealthCheck”
E' l'uri path di health check del servizio OPLON®Commander Work Flow afferente a questo applicativo.

surfaceClusterWorkFlowHealthCheckUriPath=:default=”/SCWFCommand”
E' l'uri path di comando del servizio di OPLON®Commander Work Flow afferente a questo applicativo.

normalPrimerWorkFlow=:default=”normalPrimer” UM=Work Flow name
E' il nome del Work Flow che sarà innescato se viene determinato lo startup iniziale del servizio..

gracefulShutdownWorkFlow=:default=”gracefulShutdown”UM=Work Flow name
E' il nome del Work Flow che sarà innescato se viene determinato il down del servizio e immediatamente prima di effettuare l'azione di recovery.

restartWorkFlow=:default=”null” UM=Work Flow name
E' il nome del Work Flow che sarà innescato per tentare di eseguire un restart prima di innescare le procedure di take-over in un altro semisito.
Se non valorizzato non verrà eseguita questa fase.

takeControlWorkFlow=:default=”takeControl”UM=Work Flow name
E' il nome del Work Flow che sarà innescato dopo l'avvio del gracefulShutdownWorkFlow per avviare l'azione di recovery.

<failOverService>

```
<serviceconf>
        <surfaceclusterde>
                <decisionEngineMgr>
                        <decisionEngine>
                        <healthCheckServicesPolicy>
                                <failOverService
```

Questo paragrafo contiene le informazioni di un servizio applicativo posizionato in uno specifico sistema.
Il paragrafo contiene quindi per eccezione rispetto al paragrafo <healthCheckServicesPolicy> tutti i parametri per effettuare health check e recovery dell'applicazione.

enable=:default=”true” UM=boolean
Abilita o disabilita l'interpretazione di questo paragrafo nell'istanza.

description=:default=””

Descrizione puntuale del servizio posto in alta affidabilità. Si consiglia, nei limiti del possibile, di utilizzare la nomenclatura OPLON®Surface Clsuter es.: HalfSite A.A piuttosto che HalfSite B.B con la breve ma esaustiva descrizione del servizio.

operator=:default="<healthChekServicePolicy sameParam"
Operatore booleano applicato ai servizi in healthcheck. Se "OR" anche se un solo servizio è in avaria vengono attivate le procedure di switch, se "AND" tutti i servizi devono essere in avaria per attivare le procedure di restart,

surfaceClusterWorkFlowURL=:default="https://"
E' l'URL dell'istanza OPLON®Commander Work Flow afferente a questo servizio applicativo.

waitTimeAfterNormalPrimer=:default="<healthChekServicePolicy sameParam"
UM=Millisecondi
E' il tempo di attesa dopo aver avviato il primo servizio disponibile (normal startup). Superato questo tempo il decision engine comincerà a verificare se il servizio è arrivato allo stato di attività. In caso contrario verranno avviate le procedure di recovery. Se l'applicazione va in up prima di questo valore il Decision Engine si mette immediatamente in verifica.

waitTimeAfterFlagBroken=:default="<healthChekServicePolicy sameParam"
UM=Millisecondi
E' il tempo di attesa dopo l'impostazione del flag persistente per la propagazione agli altri peer.

waitTimeBeforeTakeControl=:default="<healthChekServicePolicy sameParam"
UM=Millisecondi
E' il tempo di attesa prima di effettuare il take control. Serve a lasciare il tempo per un eventuale graceful shutdown del servizio precedentemente attivo se la risorsa è ancora raggiungibile.

waitTimeAfterTakeControl=:default="<healthChekServicePolicy sameParam"
UM=Millisecondi
E' il tempo di attesa dopo il take control. Serve ad aspettare la conclusione del completo take control prima di ritornare a verificare lo stato di attività e quindi decidere se e' andato a buon fine oppure ritentare con altra risorsa se disponibile. Se l'applicazione va in up prima di questo valore il Decision Engine si mette immediatamente in verifica. Questo tempo è variabile in dipendenza del tipo di recovery e dalla quantità di dati se presente un database.

applicationLostTime=:default="<healthChekServicePolicy sameParam" UM=Millisecondi
E' il tempo di attesa da quando l'applicazione in verifica è stata dichiarata in stato di down. Questo è un valore molto importante perché superato questo periodo di tempo e l'applicazione risultasse ancora non raggiungibile e verificato il quorum di switch, si innescherà la procedura di recovery. I 30" sono il minimo per evitare falsi positivi.

applicationLostTimeBeforeRestart=:default ="<healthChekServicePolicy sameParam"
UM=Millisecondi
Tempo di attesa prima di dichiarare persa l'applicazione e provare un restart se previsto.

surfaceClusterWorkFlowCommandUriPath=:default=”<healthChekServicePolicy sameParam”
E' l'uri path di health check del servizio OPLON®Commander Work Flow afferente a questo applicativo.

surfaceClusterWorkFlowHealthCheckUriPath=:default=”<healthChekServicePolicy sameParam”
E' l'uri path di comando del servizio di OPLON®Commander Work Flow afferente a questo applicativo.

normalPrimerWorkFlow=:default=”<healthChekServicePolicy sameParam”
E' il nome del Work Flow che sarà innescato se viene determinato lo startup iniziale del servizio.

gracefulShutdownWorkFlow=:default=”<healthChekServicePolicy sameParam”
E' il nome del Work Flow che sarà innescato se viene determinato il down del servizio e immediatamente prima di effettuare l'azione di recovery.

restartWorkFlow=:default=”<healthChekServicePolicy sameParam” UM=Work Flow name
E' il nome del Work Flow che sarà innescato per tentare di eseguire un restart prima di innescare le procedure di take-over in un altro semisito. Se non valorizzato non verrà eseguita questa fase.

takeControlWorkFlow=:default=”<healthChekServicePolicy sameParam”
E' il nome del Work Flow che sarà innescato dopo l'avvio del gracefulShutdownWorkFlow per avviare l'azione di recovery.

<healthCheck>

```
<serviceconf>
    <surfaceclusterde>
        <decisionEngineMgr>
            <decisionEngine>
                <healthCheckServicesPolicy>
                    <failOverService>
                        <healthCheck
```

I paragrafi healthCheck afferenti al servizio applicativo posto in fail over, servono a determinare l'attività del servizio applicativo.
Gli applicativi posti in Health Check possono essere anche più di uno e se anche solo uno di essi dovesse dare esito negativo durante il test verrà innescata la verifica di lostApplication time out terminata la quale se il quorum di switch è positivo viene innescata la procedura di recovery.

enable=:default=”true”
Se false il paragrafo non viene preso in considerazione.

description=:default=””

Descrizione sintetica dell'HealthCheck.

address=: valore di default=””
E' l'indirizzo su cui viene effettuato l'health check..

port=: valore di default=”0”
E' la porta su cui risponde il servizio di health check. Se <=0 verrà eseguito un check ICMP

SSL=: valore di default=”false”
Se impostato a true esegue l'health check del servizio attraverso una connessione SSL (HTTPS).

uriPath=: valore di default=””
E' l'URIPath su cui risponde il servizio di health check. Se non presente verrà eseguito un health check con connessione TCP.

numRetryConnection=: valore di default=”default da <params>”
E' il numero di tentativi di default per dichiarare un servizio non più attivo. Se non valorizzato prende il default dal paragrafo <params>.

waitPerRetryConnection=: valore di default=”300” UM=Millisecondi
Tempo di attesa tra un tentativo di connessione e l'altro fino al raggiungimento di numRetryConnection tentativi. Se non valorizzato prende il default dal paragrafo <params>.

createConnectionTimeOut=: valore di default=”default da <params>”
Tempo di attesa prima di dichiarare timeout un tentativo di connessione. Se non valorizzato prende il default dal paragrafo <params>.

<healthCheckPublicPolicy>

```
<serviceconf>
    <surfaceclusterde>
        <decisionEngineMgr>
            <decisionEngine>
                <healthCheckPublicPolicy
```

Paragrafo deputato a verificare la raggiungibilità della rete pubblica attraverso un minimo di 3 HealthCheck su indirizzi e servizi distinti.

<healthCheck>

```
<serviceconf>
    <surfaceclusterde>
        <decisionEngineMgr>
            <decisionEngine>
                <healthCheckPublicPolicy>
                    <healthCheck
```

Indirizzo o servizio da controllare in base ai parametri inseriti.

enable=:default=”false”
Se false il paragrafo non viene preso in considerazione.

description=:default=””
Descrizione sintetica dell'HealthCheck.

address=: valore di default=””
E' l'indirizzo su cui viene effettuato l'health check..

port=: valore di default=”0”
E' la porta su cui risponde il servizio di health check. Se <=0 verrà eseguito un check ICMP

SSL=: valore di default=”false”
Se impostato a true esegue l'health check del servizio attraverso una connessione SSL (HTTPS).

uriPath=: valore di default=””
E' l'URIPath su cui risponde il servizio di health check. Se non presente verrà eseguito un health check con connessione TCP.

numRetryConnection=: valore di default=”default da <params>”
E' il numero di tentativi di default per dichiarare un servizio non più attivo. Se non valorizzato prende il default dal paragrafo <params>.

waitPerRetryConnection=: valore di default=”300” UM=Millisecondi
Tempo di attesa tra un tentativo di connessione e l'altro fino al raggiungimento di numRetryConnection tentativi. Se non valorizzato prende il default dal paragrafo <params>.

createConnectionTimeOut=: valore di default=”default da <params>”
Tempo di attesa prima di dichiarare timeout un tentativo di connessione. Se non valorizzato prende il default dal paragrafo <params>.

<healthCheckBackendPolicy>

```
<serviceconf>
    <surfaceclusterde>
        <decisionEngineMgr>
            <decisionEngine>
                <healthCheckBackendPolicy
```

Paragrafo deputato a verificare la raggiungibilità della rete di backend attraverso un minimo di 3 HealthCheck su indirizzi e servizi distinti.

<healthCheck>

```
<serviceconf>
    <surfaceclusterde>
        <decisionEngineMgr>
            <decisionEngine>
                <healthCheckBackendPolicy>
                    <healthCheck
```

Indirizzo o servizio da controllare in base ai parametri inseriti.

enable=:default=”false”
Se false il paragrafo non viene preso in considerazione.

description=:default=””
Descrizione sintetica dell'HealthCheck.

address=: valore di default=””
E' l'indirizzo su cui viene effettuato l'health check..

port=: valore di default=”0”
E' la porta su cui risponde il servizio di health check. Se <=0 verrà eseguito un check ICMP

SSL=: valore di default=”false”
Se impostato a true esegue l'health check del servizio attraverso una connessione SSL (HTTPS).

uriPath=: valore di default=””
E' l'URIPath su cui risponde il servizio di health check. Se non presente verrà eseguito un health check con connessione TCP.

numRetryConnection=: valore di default=”default da <params>”
E' il numero di tentativi di default per dichiarare un servizio non più attivo. Se non valorizzato prende il default dal paragrafo <params>.

waitPerRetryConnection=: valore di default=”300” UM=Millisecondi
Tempo di attesa tra un tentativo di connessione e l'altro fino al raggiungimento di numRetryConnection tentativi. Se non valorizzato prende il default dal paragrafo <params>.

createConnectionTimeOut=: valore di default=”default da <params>”
Tempo di attesa prima di dichiarare timeout un tentativo di connessione. Se non valorizzato prende il default dal paragrafo <params>.

27
OPLON® WorkFlow Split Brain Assasin

Questo servizio gestisce, in ambienti cluster distribuiti e paritetici (stretch cluster), eventuali possibilità di Split Brain. Il servizio basa la sua logica sulla raggiungibilità di almeno uno dei due nodi OPLON®Commander Decision Engine Peer.

La struttura del file di configurazione rispecchia la semplicità che il servizio deve mantenere:

```
<serviceconf>
   <copyright>
   </copyright>
        <splitbrainassassin>
                <params>
                </params>
                <decisionEnginesPeers>
                    <peer/>
                    <peer/>
                    <notification/>
                    <notification/>
                    ...
                </decisionEnginesPeers>
        </splitbrainassassin>
</serviceconf>
```

<*splitbrainassassin*>

```
<serviceconf>
        <splitbrainassassin>
```

<*params*>

```
<serviceconf>
        <splitbrainassassin>
                <params
```

Il paragrafo <params> descrive il funzionamento generale del servizio:

frequency=: valore di default=”10000” UM=Millisecondi
E' la frequenza di verifica cambiamenti di stato.

splitBrainTimeDetection=: valore di default=”30000” UM=Millisecondi
E' il tempo di attesa dopo un evento di failure prima di impostare i file di notifica outOfOrder.

createConnectionTimeOut=: valore di default=”5000” UM=Millisecondi
Time out di default della connessione TCP/IP.

numRetryConnection=: valore di default=”3”
E' il numero di tentativi di connessione prima di dichiarare non raggiungibile il servizio.

<decisionEnginesPeers>

```
<serviceconf>
        <splitbrainassassin>
                <decisionEnginesPeers>
```

<peer>

```
<serviceconf>
        <splitbrainassassin>
                <decisionEnginesPeers>
                        <peer
```

Questo paragrafo, presente necessariamente per 2 volte, descrive i parametri di HealthCheck dei servizi Decision Engine Peers.

enable=: valore di default=”true”
Abilita o disabilita il paragrafo.

description=: valore di default=””
Descrizione del peer.

URL=: valore di default=””
URL di connessione al servizio peer (normalmente https://___private__:54445/)

splitBrainTimeDetection=: valore di default=”params:splitBrainTimeDetection”
E' il tempo di attesa dopo un evento di failure prima di impostare i file di notifica outOfOrder.

createConnectionTimeOut=:valore di default=”params:createConnectionTimeOut”
Time out della di default della connessione TCP/IP.

numRetryConnection=: valore di default=”params:numRetryConnection”
E' il numero di tentativi di connessione prima di dichiarare non raggiungibile il servizio.

healthCheckUriPath=: valore di default=”/HealthCheck?decisionEngine=SCDEGroup”
E' il path di health check comprensivo del nome gruppo del cluster che si vuole osservare.
Il nome del gruppo Cluster DE è determinato dalla 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=: valore di default=”true”
Abilita o disabilita il paragrafo.

description=: valore di default=””
Descrizione del servizio su cui agisce la notifica.

fileName=: valore di default=””
Nome del file di notifica da gestire. Se il parametro viene valorizzato in relativo sarà reso assoluto con la working directory
es.:
“lib/notificationDir/outOfOrder.systemsMonitorGroup”

Risultato:
“(LBL_HOME)/lib/notificationDir/outOfOrder.systemsMonitorGroup”

28 OPLON® Authentication

Oplon®Secure Access rel. 9 introduce l'autenticazione delegata in tutte le comunicazioni tra le componenti sensibili.
L'implementazione è maturata dall'esperienza in datacenter mission-critical per aumentare la sicurezza e diminuire le possibilità di utilizzi non autorizzati di procedure all'interno del/i datacenter. La programmazione di funzioni di automazioni attraverso remote shell o netsh, anche se con password crittografate e accessi censiti a livello centrale, espongono i datacenter moderni, basati sulla virtualizzazione, in serio pericolo. L'odierna possibilità di caricamento di un intero sistema operativo all'interno di virtualizzatori in Personal Computer compromette drasticamente la sicurezza dando la possibilità di eseguire da remoto procedure altamente pericolose se eseguite per produrre dei danni.

Procedure di business-continuity e di Disaster-Recovery impongono la necessità di scrivere moltissimi script che interagiscono a vari livelli del datacenter spesso con privilegi da amministratore/root. Una semplice operazione dolosa di inversione delle repliche dello storage, lanciata ad esempio da un sistema operativo clonato, può causare danni incalcolabili.

A questo proposito tutti i file delle password delle componenti Oplon®Secure Access rel. 9 sono crittografati ed utilizzabili solo dalla posizione e dal sistema operativo su cui sono stati originati. Eventuali clonazioni del sistema o furto dei file di autenticazione rende i file illeggibili e quindi qualsiasi operazione non possibile.

Questo documento è relativo alla sola configurazione delle autenticazioni utente e delle autenticazioni delegate. Per l'installazione delle componenti Oplon®Secure Access fare riferimento ai documenti di installazione dei singoli prodotti.

Introduzione

L'implementazione dell'autenticazione degli accessi Oplon®Secure Access Rel 9 si basa su 4 punti fondamentali:

1. Crittografia dei repository delle autenticazioni.
2. Origine della creazione dei file di autenticazione.
3. Separazione dell'autenticazione utente e delegata.
4. Autonomia rispetto all'infrastruttura da gestire.

Il primo (1) punto è importantissimo in quanto i file delle autenticazioni sono crittografati

per permetterne la visibilità o leggibilità con i soli strumenti messi a disposizione dalla piattaforma Oplon®Secure Access

Il secondo (2) punto pone un significativo miglioramento nella sicurezza. Tutti i file risultano illeggibili, anche dagli strumenti Oplon®Secure Access, se vengono copiati o se vengono utilizzati su macchine virtuali o fisiche clonate.

Il terzo punto (3) separa la gestione degli utenti umani dagli utenti delegati o da sistemi Oplon®Secure Access di automazione come OPLON®Surface Cluster Decision Engine e OPLON®Surface Cluster Work Flow che possono interagire su più livelli dei datacenter.

Il quarto punto (4) evidenzia come il sistema autoritativo debba essere completamente autonomo dall'infrastruttura che deve gestire. L'autorizzazione delegata all'azione è quindi completamente slegata ad esempio da Directory Server che potrebbero non essere disponibili al momento dell'esecuzione delle procedure di fail-over.

Scenario

Lo scenario dell'autenticazione delegata è un insieme azioni che devono essere eseguite all'interno dei datacenter su più livelli infrastrutturali come Storage Area Network, Database, Directory server, application server e su più piattaforme operative, fisiche e virtuali: Sistemi operativi differenti, sistemi di virtualizzazione differenti. Uno scenario tipico di ambienti di Business-continuity e Disaster-recovery.

OPLON®Surface Cluster Work Flow permette di eseguire le operazioni necessarie attraverso diversi layer architetturali invocando in uno step una operazione remota denominata Remote Workflow Command (RWC da questo momento). Una operazione RWC permette di eseguire un intero Workflow oppure un Workflow partendo da uno step designato o solamente uno step.

Solo a titolo esemplificativo di seguito una RWC dove #LBL_ADDRESS_RWC_TARGET# è la variabile che contiene l'indirizzo su cui effettuare l'operazione mentre selfTestNormalEnd è lo step da cui iniziare l'esecuzione del workflow. Per ulteriori approfondimenti si veda il manuale LBL_SurfaceCluster_Installation.pdf.

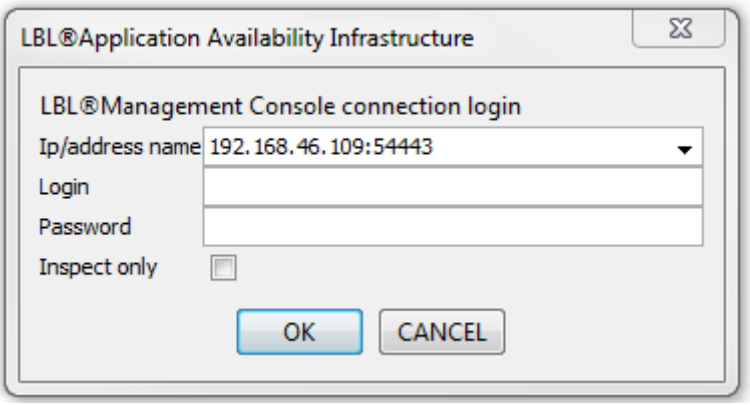
```
<step enable="true"
        name="rwcTest"
        description="Self Test ab end"
        waitBeforeExecute="1000"
        sysCommandTimeOut="600000"
        evaluateReturnCode="true"
        command="@RWC hostName=#LBL_ADDRESS_RWC_TARGET# workFlow=selfTestNormalEnd"
        maxRetry="3"
        gotoWhenFalse="abEnd">
        <returncode value="0"
                        description="ok"
                        result="true"/>
</step>
```

Come si potrà notare nei parametri è assente qualsiasi elemento di autorizzazione come Login o Password. Infatti, le autorizzazioni sono contestuali alla persona o automa che avvia

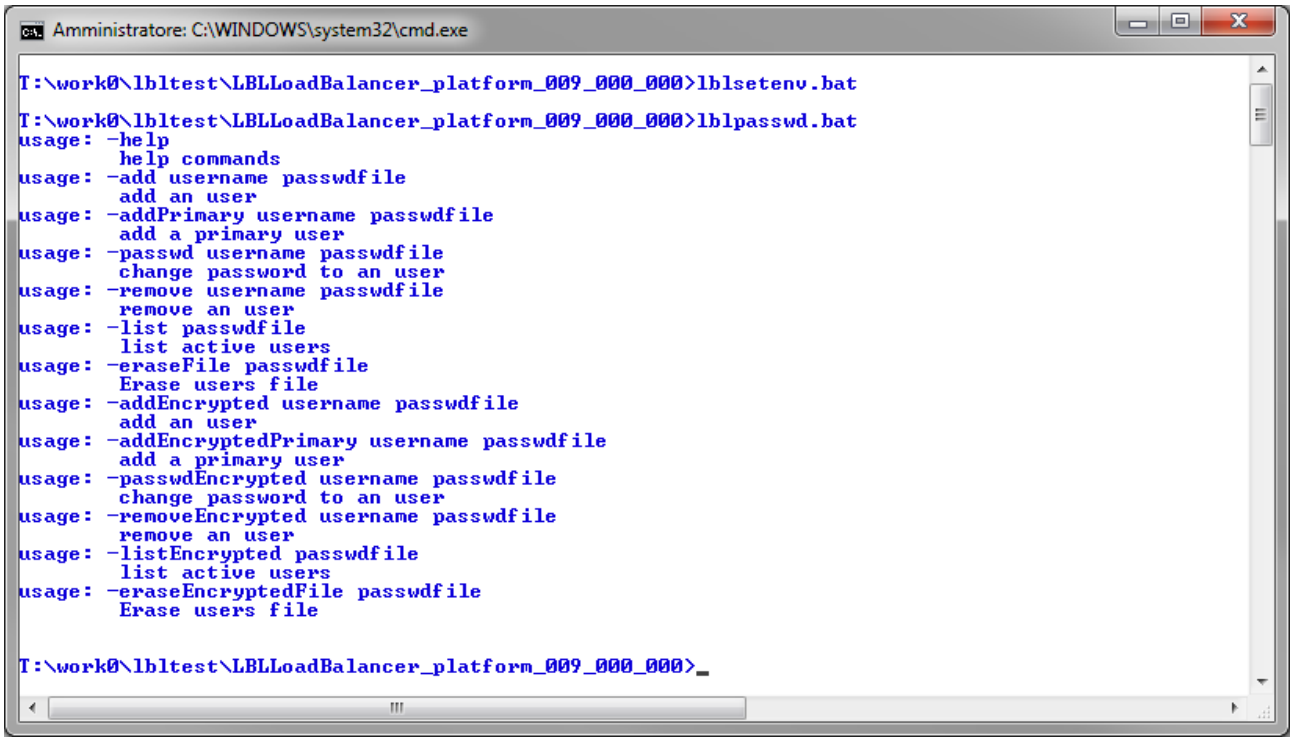
il workflow e vengono propagate ai nodi remoti che dovranno eseguire l'operazione. I nodi remoti in base al loro repository di autorizzazione verificheranno se la richiesta proviene da una fonte sicura ed in tal caso eseguiranno l'operazione.

Autorizzazioni utente

Le autorizzazioni utente sono le autorizzazioni che vengono utilizzate da persone che si accingono ad operare con gli strumenti Oplon®Secure Access. Ad esempio nel momento in cui si avvia OPLON®Management Console viene richiesta una login password autoritativa utente:



L'impostazione delle autorizzazioni utente viene effettuata tramite il comando lblpasswd (.bat o .sh) dalla (LBL_HOME) directory. La prima operazione da effettuare è l'impostazione delle variabili di ambiente tramite lblsetenv.bat/.sh e quindi eseguire lblpasswd.bat/.sh



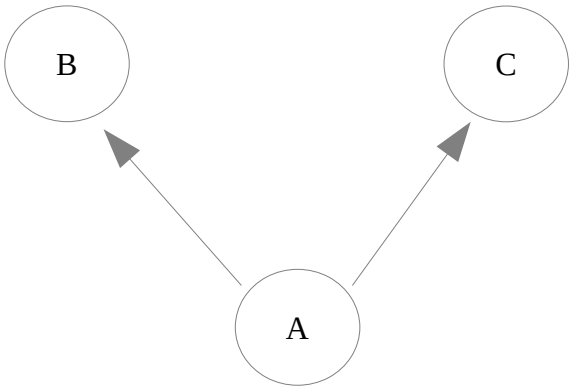
I comandi di impostazione delle password encrypted e non, sono equivalenti per ragioni di compatibilità (Da Oplon®Secure Access rel 9 tutti i repository delle password sono crittografati). Ad esempio il comando -add e -addEncrypted eseguono la stessa operazione.

Autorizzazioni delegate

Le autenticazioni delegate sono utilizzate per autorizzare un'operazione che deve attraversare più layer e dove il layer di esecuzione finale non necessariamente conosce l'utente o l'automa che ha effettuato la prima operazione.

Possiamo ad esempio ipotizzare un'organizzazione che necessiti di avere tre site con diverse autorizzazioni di esecuzione a seconda della posizione da cui si vuole effettuare l'operazione.

Supponiamo che le policy di autorizzazione permettano al sito A di poter effettuare operazioni nel sito B e C ma il sito B e C possano effettuare operazioni solo su loro stessi.



L'autenticazione e autorizzazione delegata si basa su un repository autoritativo diverso dal repository degli utenti. Per poter utilizzare e modificare il repository delegato è necessario utilizzare l'utilty (LBL_HOME)/lbldelegatedpasswd.bat/.sh.

lbldelegatedpasswd.bat/.sh utilizza gli stessi comandi che sono a disposizione di lblpasswd.bat/.sh ma il trattamento del repositori, da parte dei programmi che lo interpretano, è leggermente diverso. Il repository deve contenere un utente delegato che è considerato primario. Per l'impostazione dell'utente primario è sufficiente effettuare:

```
(LBL_HOME)/llbldelegatedpasswd.bat -addPrimary myPrimary
```

Se si è già effettuata l'installazione della componente *Oplon®Secure Access* sicuramente verrà riportato il messaggio di errore:

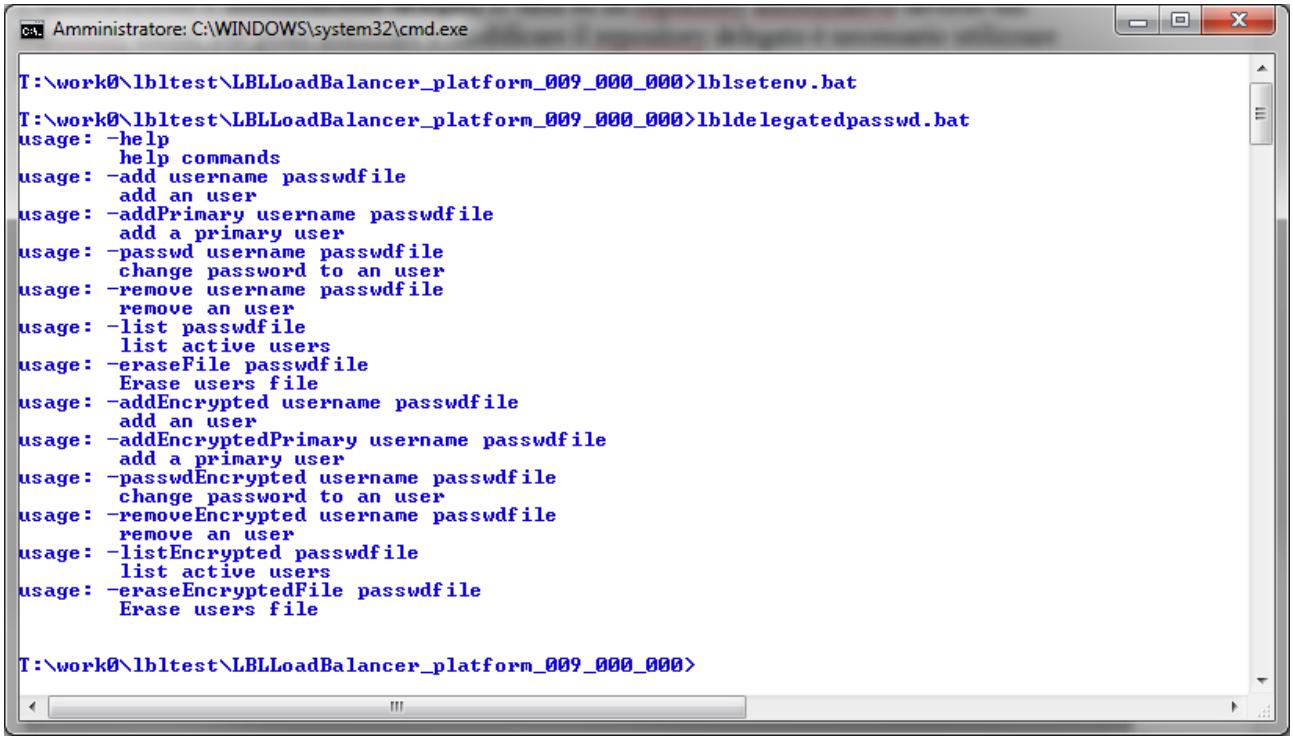
```
(LBL_HOME)/llbldelegatedpasswd.bat -addPrimary myPrimary
Primary user already exists!
```

Il messaggio è dovuto alla presenza di un utente primario già impostato nel repository durante il setup iniziale con il comando (LBL_HOME)/lblinit.bat/.sh

L'utente primario proposto dal comando (LBL_HOME)/lblinit.bat/.sh è possibile visionarlo

tramite il comando:

```
(LBL_HOME)/lbldelegatedpasswd.bat -list
delegated primary

(LBL_HOME)>
```



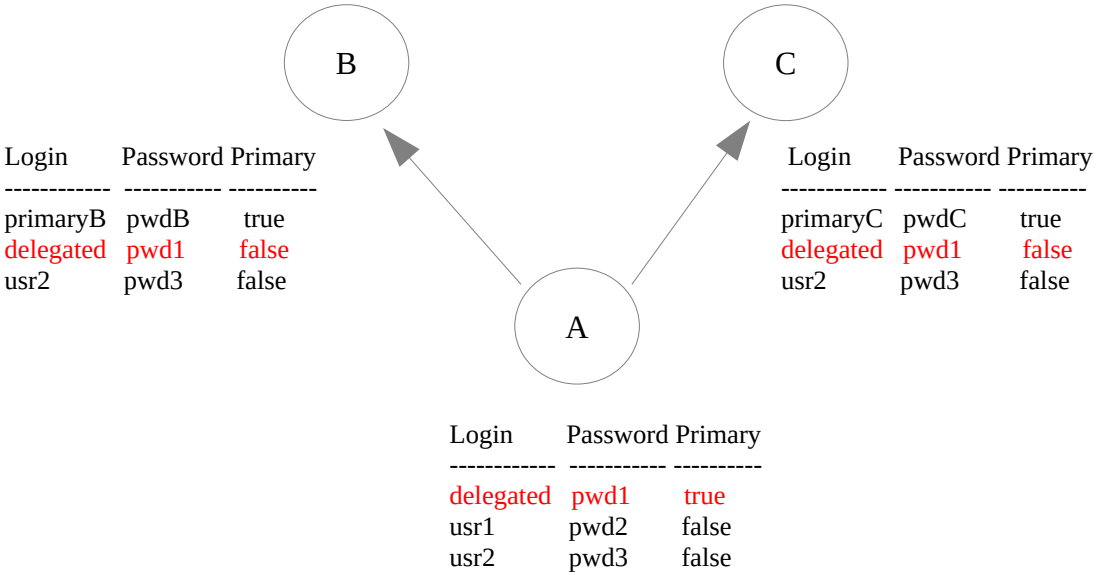
Riassumendo il repository di autenticazione delegata deve avere al suo interno un unico utente che è stato definito come primario. L'utente primario verrà utilizzato dal programma “client”, cioè colui che richiede di fare una operazione remota, proponendo le sue credenziali. Il programma “server”, cioè l'obiettivo del comando, dovrà avere un utente con le stesse credenziali (login e password). Il programma “server” non deve necessariamente avere lo stesso utente dichiarato primario, ma deve contenerlo nella propria lista di utenti delegati.

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 questo caso l'azione delegata del nodo A può essere effettuata verso il nodo B in quanto B contiene un utente con stesso login e password anche se in B non è primario. Il nodo B non può eseguire nessuna azione delegata verso il nodo A in quanto pur contenendo il login

"delegated" con credenziali uguali al nodo A, B effettuerà una azione verso il nodo A esponendo le sue credenziali con il login "primaryB" che non è presente nel nodo A e che quindi rifiuterà l'operazione.
La tecnica è molto semplice e per questo molto efficace. Questa tecnica permette anche, in situazioni dove non è necessaria una stringente policy di sicurezza, di effettuare con poche impostazioni delle configurazioni autoconsistenti.

Se torniamo al nostro esempio iniziale e lo implementiamo completamente di seguito un possibile schema autoritativo.

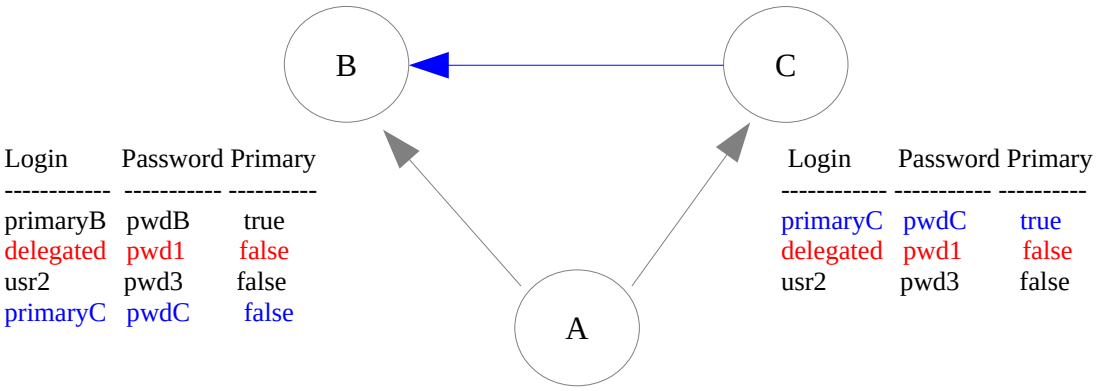


Login	Password	Primary
primaryB	pwdB	true
delegated	pwd1	false
usr2	pwd3	false

Login	Password	Primary
primaryC	pwdC	true
delegated	pwd1	false
usr2	pwd3	false

Login	Password	Primary
delegated	pwd1	true
usr1	pwd2	false
usr2	pwd3	false

Con questo schema autoritativo il solo sita A è in grado di fare operazioni cross site mentre i site B e C possono effettuare solamente operazioni al loro interno. Se in un secondo momento volessimo abilitare delle operazioni tra il site C ed il site B l'operazione risulterà semplicissima aggiungendo al site B l'utente primaryC.



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

Autorizzazioni utente esterne

Le autorizzazioni utente, sono facilmente integrabili con sistemi di autenticazioni esterni tramite l'utilizzo di plugin java.
In questo caso a fronte di una richiesta di login, LBL autentica l'utente attraverso il sistema di autenticazione esterno. In caso di mancata operatività del sistema di autenticazione, LBL procede all'autenticazione dell'utente utilizzando il repository interno.

Plugin Java

Per poter interfacciare LBL con un sistema di autenticazione esterno, è necessario scrivere una plugin java, che implementi il metodo login della classe astratta TCOAASLoginAbstr.

```
public abstract TCOAASUser login(String login, String password);
```

Il metodo login ritorna un oggetto di tipo TCOAASUser che contente le caratteristiche dell'utente.

```
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;
	}
}
```

Un esempio di plugin per LDAP si puo' trovare in

```
../LBL_HOME/security/aas/LBLAALLoginInternalTest.java
```

La classe LBLAALLoginInternalTest.java procede al caricamento dell'oggetto TCOAASUser attraverso il metodo Login, dopo aver accertato le credenziali via LDAP. Nel caso in cui LDAP risponda negativamente alla login, o in caso di mancato funzionamento del sistema di autorizzazione, il metodo Login lascia l'oggetto TCOAASUser inizializzato a null. In questo caso LBL procede all'autenticazione attraverso il suo repository interno di autenticazione utente.

```
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 OPLON NETWORKS
 * 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

Una volta scritta, la classe java di plugin deve essere posizionata nella cartella

```
LBL_HOME/security/aas
```

Per la compilazione della classe, è sufficiente utilizzare lo script compile.sh/bat.

```
$ sudo sh ./compile.sh *.java
```

Configurazione

Affinché LBL esegua l'autenticazione utilizzando un plugin esterno è necessario modificare il file _loginAAS.xml che su trova in:

```
(LBL_HOME)/lib/webroot/webappsconf/_loginAAS.xml
```

Modificare il l'attributo className del tag authorization con il nome della classe plugin

(Default value):

```
<contextRoot>
        <authorizations className="tco.lib.aas.TCOAASLoginDefaultImpl"/>
</contextRoot>
```

(template example):

```
<contextRoot>
        <authorizations className="aas.LBLAASLoginInternalTest"/>
</contextRoot>
```