

# High Availability Whitepaper



# ELIXIR REPERTOIRE

Integrated Business Intelligence Suite

# High Availability with Elixir

## High Availability

---

*High-availability clusters (also known as HA Clusters or Failover Clusters) are computer clusters that are implemented primarily for the purpose of providing high availability of services which the cluster provides. They operate by having redundant computers or nodes which are then used to provide service when system components fail. Normally, if a server with a particular application crashes, the application will be unavailable until someone fixes the crashed server. HA clustering remedies this situation by detecting hardware/software faults, and immediately restarting the application on another system without requiring administrative intervention, a process known as Failover. As part of this process, clustering software may configure the node before starting the application on it. For example, appropriate filesystems may need to be imported and mounted, network hardware may have to be configured, and some supporting applications may need to be running as well.*

(Source from [http://en.wikipedia.org/wiki/High-availability\\_cluster](http://en.wikipedia.org/wiki/High-availability_cluster))

Elixir Technology provides a combination of load-balancing software and enterprise ready Repertoire Server to fulfill your organization's Reporting, ETL and Interactive Dashboard needs.

## Elixir Repertoire Server

---

Elixir Repertoire Server provides a scalable reporting (Elixir Report Designer) and Enterprise Transformation and Loading (Elixir Data Designer) solution that grows with your business needs. It scales from small departmental workgroups to enterprise portal deployments. It may be deployed on small single departmental servers run on lower cost Linux, Windows Servers to large enterprise server such multi-processors boxes. It's unique architecture allows you to configure and tune the usage of server resources.

It is written in Java language to provide cross platform functionality and will run on a Java 5 (or later) compliant machine. The server provides a web interface, support for the Elixir Repertoire Remote Designer and an HTTP-based API so that you can build applications in the programming language of your choice and call the server for scheduling, reports, data and dashboard functionality. Elixir Repertoire Server supports repository-based storage and provides secure access to the datasources, reports and dashboards that it manages.

For more information, see <http://www.elixirtech.com/release/Rep7.3.1/pdf/Elixir%20Repertoire%20Server.pdf>

## Elixir Load Balancer

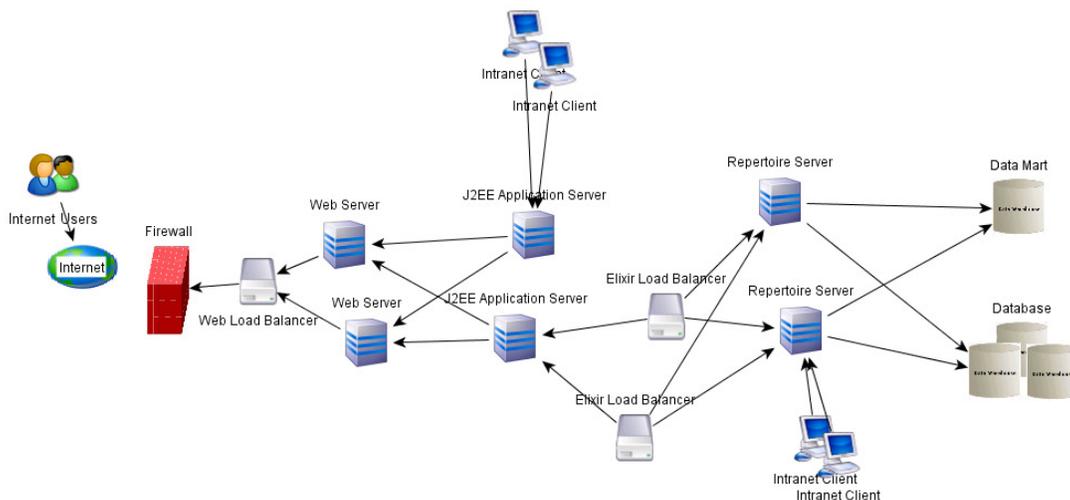
Elixir Load Balancer is a pure Java based software application, it is platform independent and can run on any J2SE enabled OS platforms (> version 1.4). Thus, Elixir Report Server Load Balancer supports a wide range of platforms and operating systems and is able to run on commodity hardware. Therefore, its performance can easily and cost-effectively grow as required.

For more information, see

<http://www.elixirtech.com/ElixirRepertoire/productresource/technicalresource/LoadBalancerDocumentation.pdf>

This article discusses how high availability is achieved using Elixir Load Balancer distributing loads to 2 or more instances of the Elixir Repertoire Servers. It discusses implementation details for setting up load balancing with Elixir Load Balancer. It also looks at some of the features provided by Elixir Load Balancer such as 'server affinity' and safe removal of node. Repertoire Servers supports both hardware load balancer as well as any proven software load balancer like the Apache Web Server.

The network diagram below illustrate a typical enterprise deployment situation consisting of 2 instances of the Repertoire Server, Elixir Load Balancer, Application Server and Standard Web Server



## Server Affinity

---

Server Affinity, a concept in load balancing and clustering, refers to the ability of a load balancer or router to send a user's request to the same server where their session was initiated. It is important to send the user to the same server for each request, otherwise, he will be forced to initiate a new session each time they are sent to a new server.

With Repertoire Server 7.3.x, Server Affinity feature can be configured for all instances deployed within the server farm.

Repertoire Server - <installation\_directory>/config/ERS2.xml

```
<!-- enable or disable Server affinity, the session id will include ip address -->
  <ers:property name="ServerAffinityEnabled">true</ers:property>
```

Once this is completed, the next restart of the Repertoire Server would allow the server ip address to be included in the session cookie – JSESSIONID.

## Dynamic Instance Configurator

---

With Elixir Load Balancer 2.x, any configured Repertoire Server instance can be automatically discovered and dynamically included within the Load Balancer's pool of working servers. This reduces any downtime required to add or delete working servers from the Load Balancer's pool.

The Dynamic Instance Configurator needs to be initially setup in both the Load Balancer and all working Repertoire Servers.

Load Balancer - <Installation\_Directory>/config/ERS2-LoadBalancer.xml

```
<ers:mbean name="ERS2:name=DynamicInstanceConfigurator"
class="com.elixirtech.ers2.lb.DynamicInstanceConfigurator">
  <ers:property name="Port">7001</ers:property>
  <ers:property name="Address">224.5.12.24</ers:property>
  <ers:property name="Timeout">2000</ers:property>
  <ers:property name="BroadcastInterval">10000</ers:property>
</ers:mbean>
```

## Ping Listener

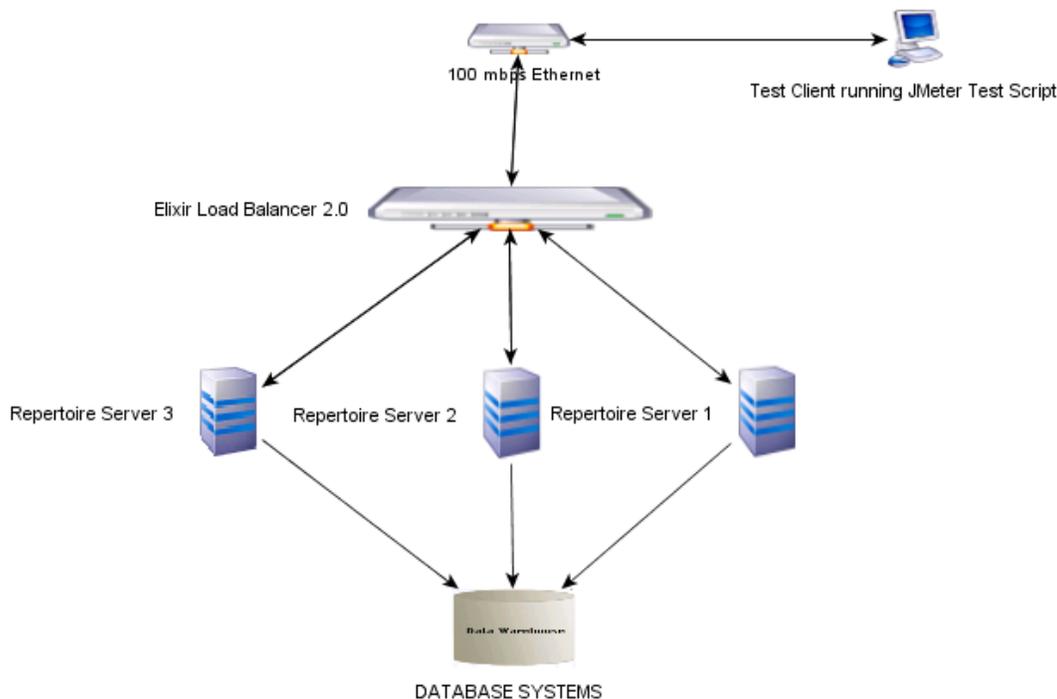
The ping listener, as the name implies, is a mbean listener which would be listening to any ping request coming from the Load Balancer's DynamicInstanceConfigurator.

Repertoire Server(s) – <Installation\_Directory>/config/ ERS2.xml

```
<ers:mbean name="ERS2:name=PingListener" class="com.elixirtech.ers2.ping.Ping">
  <ers:property name="Port">7001</ers:property>
  <ers:property name="Address">224.5.12.24</ers:property >
  <ers:property name="Timeout">30000</ers:property >
</ers:mbean>
```

## Deploying Repertoire Servers with Load Balancer

Here is a typical deployment scenario:



Once the 3 Repertoire Server instances are setup with Dynamic Configuration enabled, it would automatically join the pool of working servers. In an event of a hardware outage, where 1 of the instance has been rebooted or reset, all the subsequent incoming requests would be distributed to the rest of the working servers. When the outage server finally resumes, the Load Balancer would automatically sense its presence via a socket ping and resume its services.

## Load Balancer Administration Console

The Load Balancer 2.0 has a web administration console whereby you can monitor the health of the Load Balancer's working servers.

MX4J/Http Adaptor  
JMX Management Console

Server view | MBean view | Timers | Monitors | Relations | MLet | About

MBean By Domain: Filter: [ ] Query

Domain: com.sun.management

com.sun.management:type=HotSpotDiagnostic	sun.management.HotSpotDiagnostic	Information on the management interface of the MBean	Unregister
---	----------------------------------	--	------------

Domain: ERS2

ERS2.name=ClassLoader	com.elixirtech.ers2.jmx.MBeanClassLoader	Information on the management interface of the MBean	Unregister
ERS2.name=DynamicInstanceConfigurator	com.elixirtech.ers2.lb.DynamicInstanceConfigurator	Information on the management interface of the MBean	Unregister
ERS2.name=MX4JHttpAdaptor	mx4j.tools.adaptor.http.HttpAdaptor	Information on the management interface of the MBean	Unregister
ERS2.name=LBAdmin	com.elixirtech.ers2.lb.LBAdmin	Information on the management interface of the MBean	Unregister
ERS2.name=LoadBalancer	com.elixirtech.ers2.lb.LoadBalancer	Information on the management interface of the MBean	Unregister
ERS2.name=ServerProbe	com.elixirtech.ers2.lb.ServerProbe	Information on the management interface of the MBean	Unregister
ERS2.name=XSLTProcessor	mx4j.tools.adaptor.http.XSLTProcessor	Information on the management interface of the MBean	Unregister

Domain: ERS2Instance

ERS2Instance.name=DI-10	com.elixirtech.ers2.lb.ERS2Instance	Information on the management interface of the MBean	Unregister
ERS2Instance.name=DI-11	com.elixirtech.ers2.lb.ERS2Instance	Information on the management interface of the MBean	Unregister
ERS2Instance.name=DI-5	com.elixirtech.ers2.lb.ERS2Instance	Information on the management interface of the MBean	Unregister

Domain: java.lang

java.lang:type=ClassLoading	sun.management.ClassLoadingImpl	Information on the management interface of the MBean	Unregister
java.lang:type=Compilation	sun.management.CompilationImpl	Information on the management interface of the MBean	Unregister
java.lang:type=GarbageCollector,name=PS_MarkSweep	sun.management.GarbageCollectorImpl	Information on the management interface of the MBean	Unregister
java.lang:type=GarbageCollector,name=PS_Scavenge	sun.management.GarbageCollectorImpl	Information on the management interface of the MBean	Unregister
java.lang:type=Memory	sun.management.MemoryImpl	Information on the management interface of the MBean	Unregister
java.lang:type=MemoryManager,name=CodeCacheManager	sun.management.MemoryManagerImpl	Information on the management interface of the MBean	Unregister
java.lang:type=MemoryPool,name=Code_Cache	sun.management.MemoryPoolImpl	Information on the management interface of the MBean	Unregister
java.lang:type=MemoryPool,name=PS_Eden_Space	sun.management.MemoryPoolImpl	Information on the management interface of the MBean	Unregister
java.lang:type=MemoryPool,name=PS_Old_Gen	sun.management.MemoryPoolImpl	Information on the management interface of the MBean	Unregister
java.lang:type=MemoryPool,name=PS_Perm_Gen	sun.management.MemoryPoolImpl	Information on the management interface of the MBean	Unregister
java.lang:type=MemoryPool,name=PS_Survivor_Space	sun.management.MemoryPoolImpl	Information on the management interface of the MBean	Unregister
java.lang:type=OperatingSystem	com.sun.management.UnixOperatingSystem	Information on the management interface of the MBean	Unregister
java.lang:type=Runtime	sun.management.RuntimeImpl	Information on the management interface of the MBean	Unregister
java.lang:type=Threading	sun.management.ThreadingImpl	Information on the management interface of the MBean	Unregister

One of the important area to monitor is the distribution of the requests to the working servers - TotalClientCount

MBean ERS2Instance:name=DI-5				
Description Information on the management interface of the MBean				
Attributes				
Name	Description	Type	Value	New Value
AllowNewConnections	Attribute exposed for management	boolean	true	<input checked="" type="checkbox"/> true <input type="checkbox"/> false <input type="button" value="set"/>
CurrentClientCount	Attribute exposed for management	int	0	Read-only attribute
Dynamics	Attribute exposed for management	boolean	true	Read-only attribute
Host	Attribute exposed for management	java.lang.String	192.168.1.164	<input type="text" value="192.168.1.164"/> <input type="button" value="set"/>
IP	Attribute exposed for management	java.lang.String	192.168.1.164	Read-only attribute
Port	Attribute exposed for management	int	8080	<input type="text" value="8080"/> <input type="button" value="set"/>
TotalBytesTransmitted	Attribute exposed for management	long	35854	Read-only attribute
TotalClientCount	Attribute exposed for management	long	14	Read-only attribute

With the ability to load balanced multiple instances of the Repertoire Server using the Elixir Load Balancer 2.x, high availability is achieved for all standard Repertoire features like high performance report generation, multiple ETL processes and support for dashboard navigations.

## Summary

---

Elixir Repertoire offers a wide array of functionalities for maximum business impact, while leveraging Java Standard & Enterprise Edition for cross-platform compatibility. Satisfied customers worldwide rely on Elixir Repertoire to improve operations efficiency as well as increase corporate profitability.

For additional information, please contact Elixir at [sales@elixirtech.com](mailto:sales@elixirtech.com).

## About Elixir Technology

---

Elixir Technology provides Integrated Business Intelligence with Elixir Repertoire™ - an award-winning product for Dashboard, Reporting, Data ETL and Scheduling. Supporting "Web 2.0" with RESTful web services architectural approach on SOA, Elixir Repertoire aims to power the new generation enterprise applications with Business Intelligence that is simplified yet powerful, feature-rich yet affordable. With over 600 customers across 50 countries already using its enterprise reporting feature, Elixir Repertoire completes the entire Business Intelligence cycle to realize the full return of investment.

Elixir Technology is headquartered in Singapore with an R&D arm in UK, represented by a worldwide network of partners.

## Contact Us

Elixir Technology Pte Ltd  
50 Armenian Street  
#04-04 Wilmer Place  
Singapore 179938  
Republic of Singapore

Telephone: +65 6532 4300  
Fax: +65 6532 4255

Email: [sales@elixirtech.com](mailto:sales@elixirtech.com)  
Visit us at [www.elixirtech.com](http://www.elixirtech.com)

