

MARKET EXCHANGE 3.0

CLUSTERING GUIDE

Version 3.0
Date 2/2001

© 2001, Idapta, Inc.

IDAPTA

LEGAL NOTICES

Copyright Notice

Copyright © 2001. Idapta, Inc. All rights reserved.

Proprietary and Confidential. The contents of this document are governed by a license agreement and can only be used in accordance with the terms and conditions described therein. This document may not be copied or distributed in any manner without express written consent from Idapta, Inc. Information is subject to change without notice.

Trademarks

MarketExchange™ is a registered trademark of Idapta, Inc.

BEA® and WebLogic® are registered trademarks of BEA Systems.

Oracle® is a registered trademark of Oracle Corporation.

Sun, Sun Solaris, Solaris, UltraSparc, and Java are registered trademarks of Sun Microsystems, Inc.

Windows NT® is a registered trademark of Microsoft Corporation.

All other products and company names may be trademarks or registered trademarks of their respective manufacturers.

This document contains information that is proprietary to Idapta, Inc. DO NOT DISTRIBUTE without prior written consent.

ABOUT THIS DOCUMENT

Introduction

This document provides information on clustering for system availability, reliability, serviceability, and scalability of MarketExchange 3.0. In addition, it focuses specifically on the recommended operating environment configuration for fail-over and scalability and clustering instructions for MarketExchange 3.0. It also provides references to Web sites for additional reading.

Disclaimer:

This installation documentation includes Universal Resource Locator (URL) references to third-party software installation documentation. Idapta confirmed, as of the publication date of this document, that the URL references contained herein correspond to Idapta's installation instructions in this document. The location and content of the third-party software installation documentation may change after the publication date of this document. The reader of this document shall:

1. Confirm that the publication date of the referenced third-party software installation documentation is on or before the publication date of this document; and
2. Promptly notify Idapta of any contrary finding.

Idapta neither publishes nor controls the third-party software installation documentation referenced herein, and Idapta disclaims any and all liability for the accuracy of any information contained therein.

Audience

The primary audiences for this document are technical users and managers who work with the MarketExchange application. Specifically:

System Integrators

System Administrators

Assumptions

The BEA WebLogic clusterable application server license party has been purchased by the customer.

WebLogic will provide JDBC session support.

Oracle Hot Standby will provide database support.

WebLogic and MarketExchange are installed in a non-clustered environment.

Compatibility/Consistency with Other Idapta Products

For clustering compatibility, all servers will run MarketExchange 3.0.

Clustering will not be backward-compatible with MarketExchange versions prior to 3.0.

Further Information

For information or comments regarding this document, contact:

Susan W. Mordenti, Director
Technical Publications
1700 Promenade II
1230 Peachtree St. NE
Atlanta, GA 30309
Phone: 404-214-2658
e-Mail address: susanm@idapta.com

CONTENTS

Legal Notices	ii
Copyright Notice	ii
Trademarks	ii
About This Document	iii
Introduction	iii
Audience	iii
Assumptions	iv
Compatibility/Consistency with Other Idapta Products	iv
Further Information	iv
Chapter 1: Introduction	1
What Is a Cluster?	1
Key Features Added	1
Fail-Over	1
Scalability and Load Balancing	3
Where to Find More Information	3
Chapter 2: Implementing WebLogic Clustering	5
Required Applications	5
Procedures	5

INTRODUCTION

What Is a Cluster?

A *cluster* is a group of servers that work together, providing a more powerful and reliable application platform than can be obtained with a single server. A cluster appears to its clients as a single server but is, in fact, a group of servers acting as one.

Key Features Added

Clustering adds two key features:

- Fail-over
- Scalability and load balancing

The following sections describe these features.

Fail-Over

Fail-over is achieved through a cluster that uses the redundancy of multiple servers to insulate clients from failures. The same service can then be provided on multiple servers in the cluster. If one server fails, another can take over. The ability to fail-over from a failed server to a functioning server increases the availability of the application to clients.

MarketExchange 3.0 provides 99.9% availability for fail-over.

The following figure shows how fail-over is accomplished when three nodes or servers are available in the system. In networks, a node is a processing location. Every node has a unique network address, sometimes called a Data Link Control (DLC) address or Media Access Control (MAC) address. If Server A fails, Server B and Server C handle the fail-over. Server A, after

restarted or fixed, is then brought back into the network. This entire process is transparent to users.

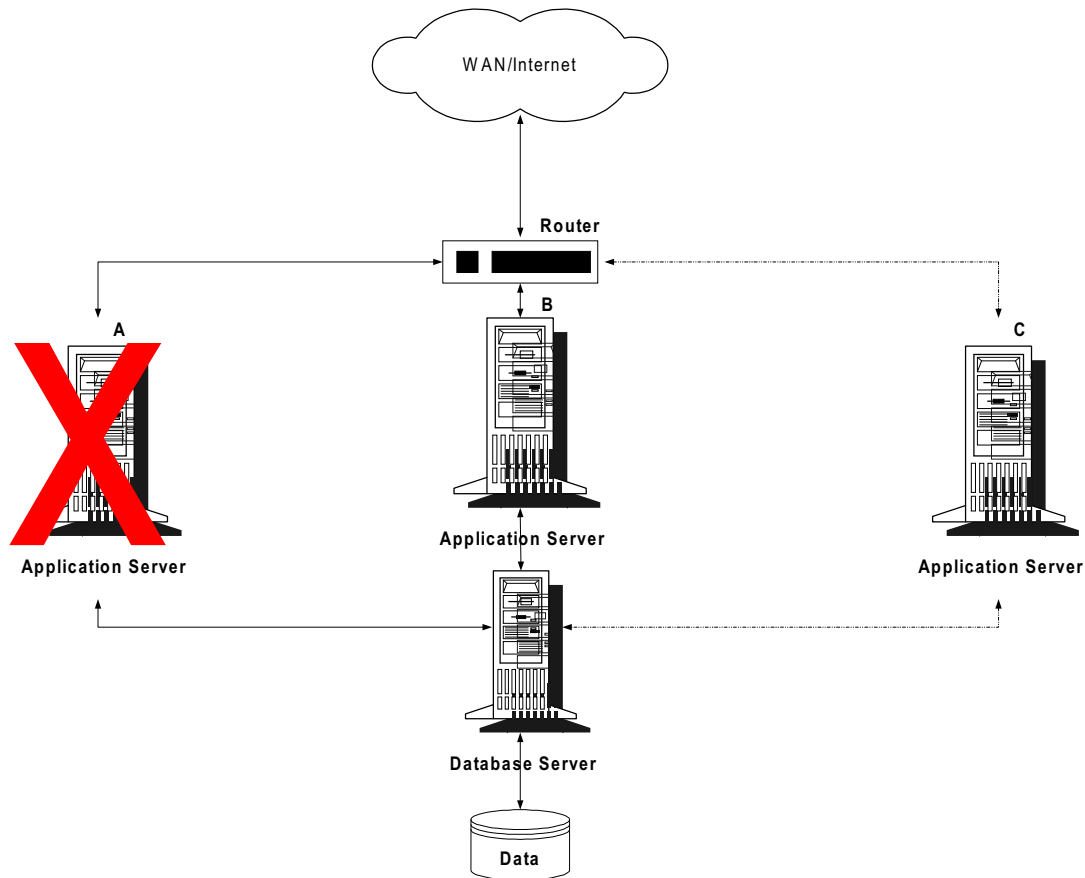


Figure 1: Fail-Over Configuration with Three Nodes

Fail-over support allows for:

- HTTP servlet fail-over

HTTP session state fail-over clustering provides clustering for servlets and JSPs by replicating the HTTP session state.

- Object level fail-over

Object clustering supports EJBs, such as stateful and stateless EJBs.

Scalability and Load Balancing

Because additional users can be supported with a cluster of servers, clustering increases *scalability*. Currently, Idapta recommends BEA WebLogic configuration standards.

Load balancing is distributing workload among multiple servers to optimize the performance based on load balancing algorithms. For the cluster to be fully utilized, load balancing of some form is typically used. Examples of load balancing include Round-Robin and Weight-Based Round-Robin (BEA WebLogic supports Round-Robin as the default).

The load balancing environment should support CPU utilization/optimization load balancing. With this type of load balancing, the system load is allocated according to each node's ability, based on the power and speed of its CPUs.

Where to Find More Information

For more information about clustering, see:

<http://www.weblogic.com/docs51/cluster/overview.html>

For more information about WebLogic 5.1, see:

<http://www.weblogic.com/docs51/install/index.html>

IMPLEMENTING WEBLOGIC CLUSTERING

Required Applications

Before you implement WebLogic clustering, make sure that WebLogic and MarketExchange are already installed (for a non-clustered environment). For more details about WebLogic installation, refer to the documentation at <http://e-docs.bea.com/index.html>.

Procedures

To implement WebLogic clustering, perform the following steps:

Step	Action
1	Go to the WebLogic home directory. In a Unix environment, you can enter: <code>cd \$WL_HOME</code> .
2	Make sure that each server is started on a separate IP address but on the same port. This is necessary to run multiple instances of WebLogic servers, and it can be achieved in two ways. You can either: <ul style="list-style-type: none">• Assign more than one IP address to a single machine, or• Have more than one physical machine
3	Make as many copies of the startME.sh file as the number of desired servers in a cluster.

Step	Action
4	<p>Assign names to the files according to your naming convention.</p> <p>Two possible naming conventions are: startME1.sh, startME2.sh, startME3.sh..., or appending the last three digits of the IP address to the name of the startME.sh file.</p>
5	<p>Make sure that your copies of the startME.sh file are in the proper directories.</p> <p>In the case of a <i>multi-homed machine</i>, all copies of the startME.sh file reside in the WebLogic home directory (i.e., /opt/weblogic).</p> <p>In the case of separate physical machines, a copy of startME.sh resides in the WebLogic home directory on each server (e.g., startME1.sh on server one, and startME2.sh on server two).</p>
6	<p>Edit either startME.sh (for Unix) or startME.cmd (for NT) and add the following parameters:</p> <ul style="list-style-type: none"> • Dweblogic.cluster.multicastAddress=<IP address>, where IP address is the multicast address used by all the servers in the cluster. Make sure this IP address is not being used by any other applications. The value ranges from 224.0.0.0 through 239.255.255.255. • Dweblogic.cluster.enable=true • Dweblogic.system.name=<name>, where name is the user-defined logical name for the specific server in the cluster. • Dweblogic.system.bindAddr=<IP address> <p>This parameter is required only in the case of a multi-homed machine (that is, a machine with more than one IP address assigned).</p> <p>The following example illustrates how to use these parameters.</p> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <p>Example - Typical <i>startME.sh</i> File in a Cluster</p> <pre>java-server-Xnoclassgc-Xms64m-Xmx64m- Dweblogic.system.name=server641- Dweblogic.system.bindAddr=10.1.6.41 Dweblogic.cluster.multicastAddress=237.0.0.7 Dweblogic.cluster.enable=true -classpath \$CLASSPATH IntelligentServer &</pre> </div>

Step	Action
7	Repeat Step 6 for each server.
8	On each server, run the corresponding startME.sh file to start the WebLogic server in the cluster.
9	Do the following: <ol style="list-style-type: none"> a. Add a directory called “mycluster” under the WL_HOME directory. b. Copy WL_HOME/myserver/public_html to WL_HOME/mycluster/.
10	Edit the weblogic.properties file in the WL_HOME directory to change the parameter weblogic.system.listenPort=<portnumber>.
11	Edit the marketexchange.properties file in the WL_HOME directory to change the parameter comos.system.providerUrl = t3://localhost:=<portnumber>. <p><i>NOTE: Make sure the portnumber is not used by any other applications other than WebLogic cluster servers. Also, the port numbers for Step 10 and Step 11 should be the same.</i></p>

