



# Ariba Buyer Cluster Testing Using Microsoft Cluster Services on Windows 2000

*IBM/Ariba International Competency Center*

**Prepared By: Joe Meerscheidt**

**Shailaja Gilela**

**December 14, 2001**

Document Name: buyer\_706w2k\_cluster\_121401.pdf



e-business

# Notices

---

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to a product, program, or service (whether IBM or vendor) is not intended to state or imply

This document may refer to products that are not currently available. IBM makes no commitment to make available any unannounced products referred to IBM's business and technical judgment.

## Trademarks and Service Marks

IBM is a registered trademark of IBM in the United States.

The following terms are trademarks of the IBM Corporation in the United States and/or other countries:

DB2 UDB  
IBM Netfinity

The following terms are trademarks or copyrights of their respective organizations, as follows:

Ariba Buyer is a registered trademark of Ariba Inc.

Oracle is a registered trademark of Oracle Corporation.

Netscape Navigator™ is a trademark owned by Netscape Communications Corporation.

Microsoft, Windows, Windows NT, Microsoft Office 97, Microsoft Visual Basic, Microsoft SQL Server, and Microsoft Internet Explorer© are copyrights of Microsoft Corp.

Netscape Communicator™ is a trademark owned by Netscape Communications Corporation.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries.

Pentium is a trademark and Intel is a registered trademark of Intel Corporation.

LoadRunner is a registered trademark of Mercury Interactive.

Other company, product, or service names may be the trademarks or service marks of others

© Copyright International Business Machines Corporation 2000. All rights reserved.

Note to U.S. Government Users – Documentation related to restricted rights – Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp.

## Table of Contents

|  |    |
|--|----|
| <b>Chapter 1: High Availability Cluster Test Overview</b> .....      | 4  |
| <b>Objective</b> .....   | 4  |
| <b>Roles and Responsibilities</b> .....                              | 4  |
| <i>SPC Technical Consultant:</i> .....                               | 4  |
| <i>IBM/Ariba ICC:</i> .....  | 4  |
| <i>Ariba Inc:</i> .....  | 4  |
| <b>ClusterProven Criteria</b> .....                                  | 5  |
| <i>Graceful Recovery Implies:</i> .....                              | 5  |
| <b>What is “MSCS”(Microsoft Cluster Server)?</b> .....               | 5  |
| <b>Cluster Terminology</b> .....                                     | 5  |
| <i>Active/Active and Active/Passive configurations</i> .....         | 6  |
| <i>Quorum Disk:</i> .....  | 6  |
| <i>Cluster Group:</i> .....  | 7  |
| <i>Disk Sharing:</i> .....   | 7  |
| <b>Ariba Buyer Application/ MSCS Test Environment Overview</b> ..... | 7  |
| <i>Hardware</i> .....  | 8  |
| <i>Software</i> .....  | 9  |
| <b>Configuring Resources For Ariba Buyer</b> .....                   | 9  |
| <i>Editing theTibco.pm file</i> .....                                | 10 |
| <i>Editing the Parameters.table file</i> .....                       | 10 |
| <i>Ariba Buyer as NT Service</i> .....                               | 11 |
| <b>Chapter 2: Tests</b> .....  | 13 |
| <b>Testing Procedure / Results</b> .....                             | 13 |
| <i>Sample Test I</i> .....   | 13 |

|  |    |
|--|----|
| <b>Sample Test II</b> .....  | 13 |
| <b>Chapter 3: Results</b> .....  | 15 |
| <b>Remove public network cable from one of the nodes : Passed</b> .....      | 15 |
| <b>All network connections failed : Passed</b> .....                         | 15 |
| <b>Shutdown, Poweroff, Reset, Shutdown and Restart a node : Passed</b> ..... | 15 |
| <b>Ariba Buyer Application Software failure : Failed</b> .....               | 16 |

# Chapter 1: Overview

## Objective

The purpose of this document is to describe the High Availability Cluster testing of Ariba Buyer 7.06 using MicroSoft Cluster Server features. This document explains the tests that were performed and the results achieved. The purpose of this test was to demonstrate the failover of the Ariba Buyer Application only. There was no attempt to ensure data or database failover at this time. The tests included:

- Loss of Network Connectivity

- Loss of Server

- Loss of Application

## Roles and Responsibilities

### **SPC Technical Consultant:**

- Setup and configure high availability hardware, OS and middleware

- Provide first level of support for IBM hardware, OS, and middleware

### **IBM/Ariba ICC:**

- Installation of application and configures for cluster environment

- Executes and manages validation testing

- Provides additional applications, load utilities and test data.

- Resolves all application related problems

### **Ariba Inc:**

- Review and approve results

- Review Setup/Configuration

## ClusterProven Criteria

An application that meets the technical criteria below can be validated on a xSeries or Netfinity server and registered as “ClusterProven” if the application:

Can swith over to an alternate cluster node when the primary system becomes unavailable

Gracefully recovers in case of a hardware or software failure

### Graceful Recovery Implies:

- } Maintains application availability in the event of failure
- } Failure recovery with minimal impact to application availability at end-user level
- } Data integrity preserved during failover - doesn't imply guard against data loss.

## What is “MSCS”(Microsoft Cluster Server)?

MSCS Software is a built-in feature of Windows 2000 Advanced Server Edition. MSCS ships a 2-node clustering package with this edition. A server cluster is a group of independent servers running Cluster service and working collectively as a single system. Server clusters provide high-availability, scalability and manageability to resources and applications by grouping multiple servers and also enable users and administrators to access certain resources of the servers, or nodes, as a single system rather than as separate computers.

The purpose of server clusters is to preserve client access to applications and resources during failures and planned outages. If one of the servers in the cluster is unavailable due to failure or maintenance, resources and applications move to another available cluster node.

A single point of failure is any component in the environment that would block data or applications if failed. We tested the application failover for the following components in the cluster group.

- Node
- Shared Disk (Quorum Disk)
- Network
- Network Adapter

## Cluster Terminology

**Resources:**

Resources are logical or physical units that Cluster service can bring online and take offline, manage in a cluster, host on only one node at a time, and move between nodes. Examples of logical cluster resources include network names and IP devices and network interfaces.

**Resource Groups:**

In a cluster, typically these resources can be grouped into functional units, called resource groups, for the specific purpose of providing a service such as a file and print share, Web service etc. The relationship among resources in a resource group is called resource dependency. For example, a network name needs to resolve to an IP address, therefore the network name is dependent upon the IP address.

**Virtual Servers:**

Virtual servers build on the concept of resources and resource groups. A virtual server in a cluster is the application service entity that network clients see.

**Failover:**

Failover is the moving of cluster services from one node to another. Cluster service supports two types of failover : **Active/active and Active/passive**. MSCS uses software “heartbeats” to detect failed applications or servers. In the event of a server failure, it employs a “shared nothing” clustering architecture that automatically transfers ownership of resources, in this case the disk drive and IP addresses, from a failed server to a surviving server. It then restarts the failed server’s workload on the surviving server. During this time the user may experience a couple of minutes delay connecting to the server. But it is transparent to the users, since the application restarts at exactly the same IP address.

**Active/Active and Active/Passive configurations**

MSCS supports both active/active and active/passive cluster configurations. The term *active/active* refers to a cluster with at least one virtual server running on each node. When an application is running on Node 1, Node 2 does not need to remain idle waiting for Node 1 to fail. Node 2 can run its own cluster-aware applications (or another instance of the same application) while providing failover capabilities for resources on Node 1. An active/active cluster node must be sized appropriately to handle the load of both nodes (in the event of a failover).

The term *active/passive* refers to failover cluster configurations in which one cluster node is actively processing requests for a clustered application while the other cluster node simply waits for the active node to fail. An active/passive configuration is more costly in terms of price/performance because one server sits idle most of the time. It is appropriate for business-critical systems since the application can use the full power of another server in case of a failure.

**Quorum Disk:**

In the event of failure, Windows Clustering maintains a primary node, and replicates changes for the cluster registry by means of a shared disk. This disk is called the “quorum drive”.

### **Cluster Group:**

When you install the Microsoft Cluster Service (MSCS), the default cluster group that is created contains an IP address and Network Name (Cluster Name) for cluster administration. In Windows 2000 and beyond it will also put the physical disk resource that is used as the Quorum disk in the cluster group as well. Additional resources should not be added to this group.

The Cluster Group contains resources related to cluster administration only. If you add additional resources to the group, the ability to administer the cluster may be hindered. Failure of any additional resources within the Cluster Group may trigger failover or inability to successfully administer the cluster.

### **Disk Sharing:**

Both nodes in the cluster were connected to SSA, but only one node could own the Quorum disk at a time. Since the Quorum disk was configured as part of the Cluster Group the node actively serving the application owns this Quorum disk. The quorum drive has two responsibilities:

- It replicates the cluster registry to all other nodes in the server cluster. The cluster registry is stored in the following location on each node:

**%SystemRoot%\Cluster\Clusdb**

- If there is a loss of communication between the nodes in the cluster, a "Split Brain" scenario occurs. In this situation, the owner of the quorum disk resource becomes the only owner of the cluster and all the resources. The owner then makes the resources available for clients. When the node that owns the quorum disk does not function correctly, the surviving nodes arbitrate to take ownership of the device. In a single cluster, there is a local quorum resource. In the event of a hardware failure of the quorum disk, it may be necessary to designate another device as the quorum disk.

## **Ariba Buyer Application/ MSCS Test Environment Overview**

The testing environment was a 2-node cluster type, configured with the Ariba Buyer application as the application cluster using MSCS 2000 Advanced Server Edition. The application cluster consisted of two X350 servers and one FastT200 storage array and the test plan was tested in "Active/Passive" mode. Each node had 2 network adapters; one adapter configured as the public network for client connections, the other for private communications (heartbeat) between cluster nodes. The network was Ethernet based running TCP/IP. The Quorum disk was configured as a shared disk on the FAStT200 and was also part of the cluster group. An Netfinity 7000 was configured as a Database Server and connected to both nodes in cluster via a CISCO switch. IBM DB2 UDB V7.2 was installed on this DB Server and configured with a small demo database. For more information refer to the 'Ariba Buyer Database Configuration Guide'. The Ariba Buyer application was configured as part of Cluster group and installed separately on both nodes in the same directory structure, so that it could be started on either node during a failover. was 'C:\Ariba\CoreServer' on both nodes.



## Hardware

xSeries 350 900 MHZ 4-way w/4Gig of memory (Main Cluster Server), 3 - 9.1G disk drives

xSeries 350 902 MHZ 4-way w/4Gig of memory (Backup Cluster Server), 3 - 9.1 disk drives

Netfinity 7000/M10 500 MHZ 4-way w/4Gig of memory (Database Server), 4 - 9.1G disk drives

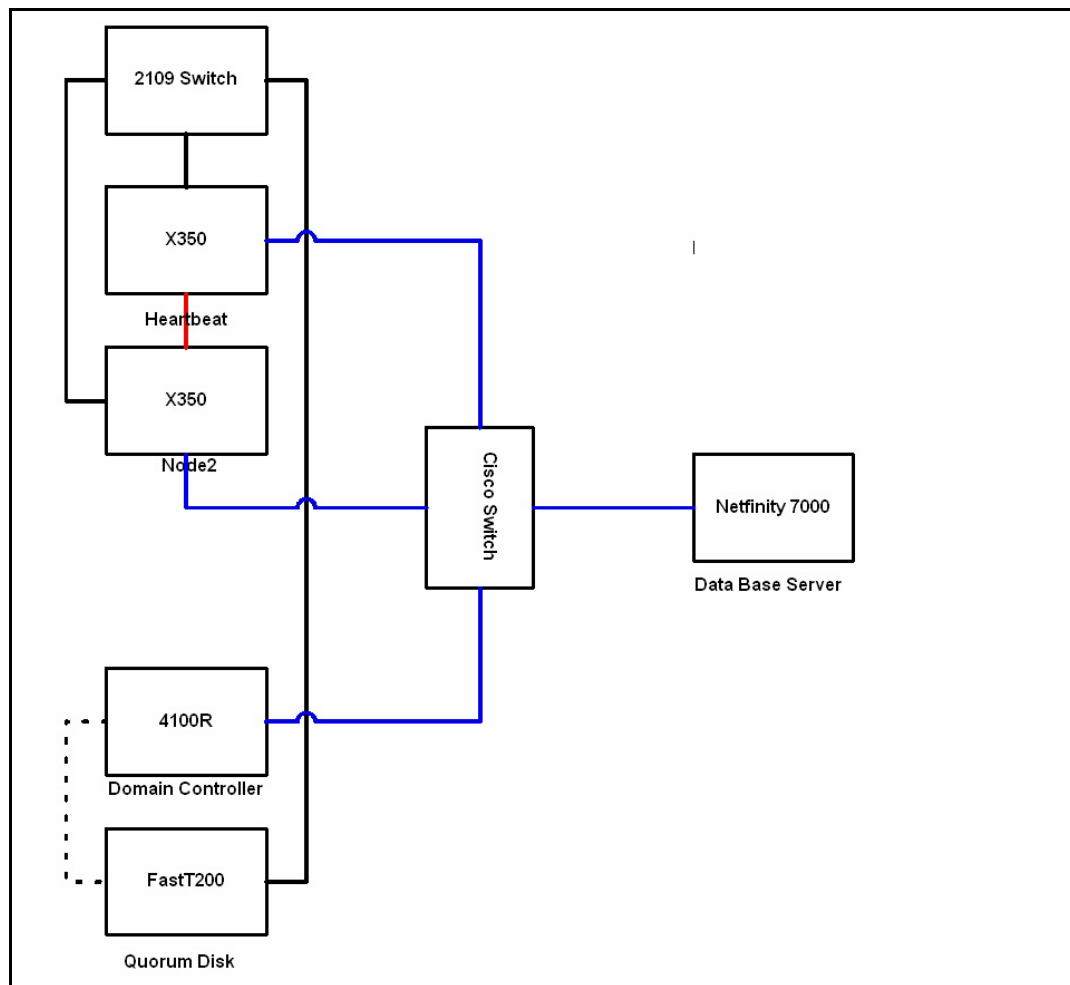
Netfinity 4000R 864 MHZ 2-way w/512Meg of memory (Domain/FastT Controller), 1 - 9.1G disk drive

FastT200 HA Shared Drives/Quorum Drive, 9 - 18.2G disk Drives

XSeries 220 Model 8645-3AX, 2-way, 866 MHZ, with 2 G of memory:

- 1 LoadRunner controller/host
- 2 36.4G disk drives

Cisco Catalyst 2900 series XL 100 mbs ethernet switch



## Software

The Clustered systems hosting Ariba Buyer 7.06 were installed with Windows/2000 Advanced Server /SP2 with MSCS, SUN JRE 1.3.1 and WebSphere 3.5.4. The Database was installed with DB2 UDB 7.2.

## Configuring Resources For Ariba Buyer

There are 3 steps involved to configure Resources for the Ariba Buyer application after the successful installation of Ariba Buyer 7.06 on both nodes.

- Create a script named as 'AribaIni.cmd' to resolve the conflict of ownership of nodes and configure the resource as a "Generic Application" using Cluster Administrator tool.
- Edit the Ariba Buyer Application configuration files Tibco.pm and Parameters.table to allow the use of a virtual address.
- Setup the Ariba Buyer application as an NT service labeled as 'AribaApp' and configure it as a cluster resource and make it dependent on the resource 'AribaIni'.

### AribaIni.cmd Script

This script contains 2 lines of code to start the Ariba Buyer application command utility 'resetDatabaseOwner' as shown below. This command must run before starting the application server in order to release Ariba buyer application lock mechanism. (To prevent accidentally erasing the data, Ariba Buyer prevents you from running 'initdb' against the database if you have already initialized from another computer).

```
cd C:\Ariba\Coreserver
```

```
start resetDatabaseOwner
```

Then from the Cluster Administrator -> Resources -> Add new Resource -> Choose as Generic Application. For the Parameters ->

```
Commandline - C:\Ariba\CoreServer\AribaIni.cmd
```

```
Current Directory - C:\Ariba\CoreServer
```

And check the box for option 'Allow application to interact with Desktop'.

### Editing theTibco.pm file

```
# This routine returns the first part of the hostname because Tibco
# has problems with the . character in their .ini files.
sub getHost ()
{

    # RELEASE_HOST is defined since release will have multiple build
    # on one machine.
    my $releaseHost = $ENV{'ARIBA_TIBCO_REPOSITORY_SERVER'};
    if ($releaseHost) {
        return $releaseHost;
    }
    my $thisHostName = $ENV{'ARIBA_DYNAMIC_HOSTNAME'};
    if (!defined($thisHostName)) {
        #$thisHostName = Sys::Hostname::hostname(); (original line is commented out)
        $thisHostName = "w2kcluster";
    }
    #my @host = split(/\./,$thisHostName); (original line is coomented out)
    my @host = $thisHostName;
    return $host[0];
}
```

### Editing the Parameters.table file

The Parameters.table file must have one element changed for MSCS to work. The hostname under the System.Nodes section must be modified so that it is not host specific. This must be changed to the virtual address (i.e. 10.1.16.111), so that whichever node is serving the application will be considered the host. The change is illustrated below.

```
Nodes = {
    Node1 = {
        ClassName = ariba.server.ormserver.ORMSNode;
        HTTPServerPort = 3377;
        Host = localhost;
```

```
Port = 7733;
};
NodeAuth = { Port = 6146;};
NodeLog = { Port = 6201;};
};
```

## Ariba Buyer as NT Service

### Windows/200 Environment Variables Setup

BuyerPath

C:\Ariba\CoreServer (This is my buyer core server path)

CTIBGOPATH

```
%BuyerPath%\tibco\rv\bin;%BuyerPath%\tibco\Repository\bin;%BuyerPath%\tibco\adapters30\sap.3.0\bin;%BuyerPath%\tibco\adapters30\peoplesoft.3.0\ps80\bin;%BuyerPath%\tibco\adapters30\peoplesoft.3.0\ps75\bin;%BuyerPath%\tibco\MessageBroker\bin;%BuyerPath%\weblogic5.1.0\bin;%BuyerPath%\ariba\peoplesoft\bin
```

TIB\_JAVA\_HOME

```
%BuyerPath%\jre\Nt\1.3.0
```

TIB\_REPO\_JAVA\_JAR

```
%TIB_JAVA_HOME%\lib\rt.jar
```

PATH

```
%CTIBGOPATH%;%BuyerPath%\jre\Nt\1.3.0\bin; (Add this into your path)
```

### Commands:

Then Run the below Buyer utilities from the CoreServer directory as follows:

```
C>Ariba/CoreServer/tibcontservice –installreposer
```

TIB/Repo service would be created by using the above command, you could verify this from Start -> Programs -> Administrative Tools -> Services program.

After this run the below commands to make Ariba buyer application as Service.

```
C:>Ariba/CoreServer/ntserviceinstall –program AribaBuyer –nodename Node1
```

```
C:>Ariba/CoreServer/ntserviceinstall –program AribaMonitor –dependency Repo-Default
```

Now you could able to see ‘AribaMonitor’ as a service from Windows Administrator tool Services program. From now we could able to start and stop Ariba Buyer application by starting and stopping AribaMonitor service. There was no need to start/stop TIB/Repo service since it was already defined as dependency on AribaMonitor.

To uninstall TIB/Repo service and AribaMonitor service simply run the below commands:

‘tibcontservice –uninstallreposer ’ and for the application ‘ntserviceinstall –uninstall’. For more information refer the manual ‘Ariba buyer application Administration on Windows NT’.

**Configuring NT Service as Cluster Resource:**

From the Clustor Administrator tool, Resources -> Add New Resource ->

The properties should be set as:

General -> Name : AribaApp

Generic Service

Possible Owners -> X350L01, X350L02

Check the box for the option ‘Run this resource in a separate Resource Monitor’.

Dependencies -> AribaIni

Parameters -> AribaMonitor

Registry Info -> Nothing

After this AribaApp also became part of Cluster Group.

Briefly the Resources Table shown as below:

| Name              | State  | Owner   | Group         | Resource Type       |
|-------------------|--------|---------|---------------|---------------------|
| Cluster IPAddress | Online | X350L01 | Cluster Group | IP Address          |
| Cluster Name      | Online | X350L01 | Cluster Group | Network Name        |
| Disk Q            | Online | X350L01 | Cluster Group | Physical Disk       |
| Disk R            | Online | X350L02 | Disk Group1   | Physical Disk       |
| AribaApp          | Online | X350L01 | Cluster Group | Generic Service     |
| AribaIni          | Online | X350L01 | Cluster Group | Generic Application |

## Chapter 2: Tests

### Testing Procedure / Results

The testing procedure is outlined in the ClusterProven and Advanced ClusterProven on xSeries Validation Process. According to the process, the application failover was tested in Active/Passive mode.

shutdown a node

poweroff a node,

hit the reset button on one of the nodes

shutdown and restart a node

remove private network cables from both nodes.

Stop the Ariba Buyer application

The Application availability was also monitored from Mercury Interactive Load Runner for almost more than 10 hours with 10 users. All tests had results as expected

#### Sample Test I

X350L01 is up and running and owns the cluster resources. The Ariba buyer application is up and running on X350L01 as well. From a client machine one user is logged on and entering a Requisition, When a failure occurs the user is automatically logged off for a couple of minutes until the second node owns the resources. There will be a delay of approximately 5 to 7 minutes while the cluster services are moved to the other node and the application restarts. When the user logs back on with the same username the user should see the requisition in composing mode.

#### Sample Test II

The Ariba Buyer application will be executed using Mercury Interactive Load Runner. Load Runner will be running 10 users simultaneously for approximately 10 hours. During this period node failures will occur and the resource group will be moved from one node to other. Whenever these failures occur LoadRunner will log 'time out' errors until the application is back up and running at which time LoadRunner will be able to reconnect the users to the application.



## Chapter 3: Results

This section covers the results of all Ariba Buyer Application clustering tests that were done at SPC Center. All test results have been reviewed and approved by Jason Ding at Ariba Inc.

### **Remove public network cable from one of the nodes : Passed**

If one network connection between nodes failed, both servers are still available and communicating over the other network link. No failover occurred, current processing continued and operation of the cluster continued with no failover of resources.

### **All network connections failed : Passed**

If all network connections between nodes failed then both servers were still alive, the servers would arbitrate for the quorum disk. Ariba buyer application and other cluster resources moved to the server that owned the quorum disk. MSCS on the server that did not own the quorum disk would shutdown, and retry later. Client requests from clients who were still able to connect to the server would continue to be serviced. Users experience approximately 5 to 7 minutes delay while connection to the server is re-established and Ariba buyer application startup process completes.

### **Shutdown, Poweroff, Reset, Shutdown and Restart a node : Passed**

In all cases when the host node hardware fails, or the operating system crashes, all heartbeat messages to the passive(standby) node cease. The passive node would then use the quorum disk to discover that the first host has failed. The passive node would then bring the resources of the failed node online, start the Ariba buyer application and become the active node. Here again, users experience approximately 5 to 7 minutes delay while connection to the server is re-established and Ariba buyer application startup process completes



## **Ariba Buyer Application Software failure : Failed**

In order to use the cluster functionality, any applications or resources started in the cluster and automatically monitored by the cluster need to stay up. If it fails depending on the configuration, the cluster either restarts or will switch over to other node. Since the Ariba Buyer application was configured as a service and was also part of a cluster group the window which started the application was also monitored by the cluster. When the Ariba Buyer Application failure happened it tried to restart the application but the window where the application was started didn't stay up, so the Ariba Buyer Application did not start. To make the Ariba buyer application "Cluster-Aware" the Ariba buyer application needs to implement their own resource DLLs to communicate with the Cluster Service and then use the cluster API to request and update cluster information.

A cluster-aware applications can:

- Report status upon request to the Resource Monitor
- Responds to requests to be brought online or to be taken offline gracefully
- Respond more accurately to IsAlive and LooksAlive requests.
- Also supports Active/Active failover and failback capability.

