



FirstSpirit™

Unlock Your Content

FirstSpirit™ HighAvailability FirstSpirit™ Version 5.2

Version	1.0
Status	RELEASED
Date	2015-06-17
Department	FS-Core
Copyright	2015 e-Spirit AG
File name	HIAV51EN_FirstSpirit_HighAvailability

e-Spirit AG
Stockholmer Allee 24
44269 Dortmund | Germany

T +49 231 . 477 77-0
F +49 231 . 477 77-499

info@e-spirit.com
www.e-spirit.com

e-Spirit

Table of contents

1	Introduction	2
1.1	Task.....	2
1.2	Summary.....	2
2	Functions	3
2.1	Overview	3
2.2	Data storage level	3
3	Supported HA systems	4
4	Installation	4
4.1	Architecture	4
4.2	FirstSpirit Server	5
4.3	Files	5
4.3.1	Control scripts.....	5
4.3.2	Configuration files for standard resource type.....	6
4.3.3	Configuration files for a specific resource type.....	6
4.4	Starting FirstSpirit Server	7
5	Configuration note for Veritas Storage Foundation HA	7
5.1.1	Introduction.....	7
5.1.2	Configuration of the resource group	8
6	Legal notices	10



1 Introduction

1.1 Task

FirstSpirit™ supports use in high-availability (HA) environments. The basic idea of HA support is that an environment that monitors FirstSpirit Server and allows FirstSpirit Server to be started on another computer in the event of a hardware or software failure is provided via a commercial high-availability solution (e.g. Veritas Cluster Service). In the event of a failure, Cluster Service detects the server or service outage and initiates restart of the service on the standby system.

1.2 Summary

This document describes how FirstSpirit support of high-availability environments works, which systems are supported and how FirstSpirit has to be configured to take advantage of HA support.



This document is provided for information purposes only. e-Spirit may change the contents hereof without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. e-Spirit specifically disclaims any liability with respect to this document and no contractual obligations are formed either directly or indirectly by this document. The technologies, functionality, services, and processes described herein are subject to change without notice.



2 Functions

2.1 Overview

FirstSpirit high-availability support provides the following functions:

- Mechanisms for software-controlled starting and stopping of FirstSpirit Server.
- Mechanisms that check whether FirstSpirit Server is running and is in an active operating state.
- Configuration isolation.
- State saving exclusively to a hard drive and/or database, which ensures that FirstSpirit Server starts up again after an outage.

2.2 Data storage level

FirstSpirit writes its data to both a file system and one or more databases. Suitable storage measures are provided to ensure that no incompletely written data is used after an error occurs (e.g. server outage):

- 1) It is assumed that the database in a high-availability environment both has high availability and operates completely using transactions.
- 2) If data is stored in a file system, the situation is not quite as simple because FirstSpirit Server cannot operate transactionally on a file system. To prevent data from not being saved or only being partially saved to the hard drive in the event of a system failure, the individual FirstSpirit project and user data items are saved internally as revisions during each save operation.
- 3) Only the FirstSpirit configuration data is written, as individual XML or text files, directly to the file system. In this case, the integrity of the data is ensured using internal backup files. The time span during which changes to the configuration files can be lost depends on the setting for the `cyclicSaveTime` parameter defined in the FirstSpirit Server configuration file (`fs-server.conf`) (default value is 60 s).

Conclusion: In the event of hardware failures, data integrity when writing to an external SQL database is ensured if the external SQL database is designed to be failsafe, i.e. uninterrupted using hot standby when switching to other hardware.

Data integrity is also ensured when writing FirstSpirit project and user data to the Berkeley DB and when writing FirstSpirit configuration files to the file system (to the extent this is technically possible, see above).



3 Supported HA systems

Systems **supported** at present:

- Veritas Storage Foundation HA on Solaris 9 and 10

Systems **not supported** at present:

- Microsoft Windows Cluster Server:
Fundamentally speaking, support would be possible but has not been implemented as of yet due to the relatively limited distribution of this solution.
- High-Availability Linux Project:
Support is not planned for version 1.x; depending on reliability and the extent of distribution, support could be realized for release 2.

4 Installation

4.1 Architecture

FirstSpirit Server consists of two processes in the operating system:

1. Java wrapper: This process is started from an executable file adapted to the respective operating system. It then calls Java VM and monitors its operation using internal control files and ping calls over its own API. In the event of faults within Java VM, it is ended and the Java wrapper is also ended using a corresponding exit code.
2. Java VM: This process is started automatically by the Java wrapper and contains the actual FirstSpirit Server instance as a pure Java application. Among other components, a web server is integrated into the Java application.

The cluster manager monitors and controls the Java wrapper process using four control scripts (see following sections) and uses them to obtain information about the state of FirstSpirit Server.



4.2 FirstSpirit Server

Initially, FirstSpirit Server is installed on a computer under the control of the cluster manager as described in the FirstSpirit installation instructions. The following parameters should be written down during installation:

- Path where FirstSpirit Server was installed (default: `/opt/firstspirit5`)
- URL of the FirstSpirit Server start page (default: http server on port 8000)
- Unix user account in which FirstSpirit Server was installed (default: `fs5`)

FirstSpirit Server has to be stopped after being installed to begin the installation from a defined start state in the cluster server.

Then it is integrated into the cluster server, which is described below:

4.3 Files

The following files are provided with the high-availability module.

4.3.1 Control scripts

- **online**: Starts FirstSpirit Server on the failover system.
- **monitor**: Checks whether FirstSpirit Server is active.
- **offline**: Executes a controlled shutdown of FirstSpirit Server.
- **clean**: Ends FirstSpirit Server by terminating the process.

The control scripts require the following respective parameters to be run:

- **FSDIR**: Path where FirstSpirit Server was installed, e.g. `/opt/firstspirit5`
- **URL**: FirstSpirit Server start page, e.g. <http://fs5.mydomain.net:8000>
- **FSUSER**: Unix user account where FirstSpirit Server was installed, e.g. `fs5`

The parameters are passed to the command line in the following order when manually calling the scripts (for testing). Example call:

```
monitor /opt/firstspirit5 http://fs5.mydomain.net:8000 fs5
```



After installing the control scripts in Veritas Cluster Server, the parameters are either passed using the configuration file `firstspirit.cfg` or assigned to the control scripts in the cluster manager's graphical interface. Which of the two procedures is used depends on which resource type is defined for FirstSpirit. The following two procedures exist:

4.3.2 Configuration files for standard resource type

This configuration is recommended if just one instance of FirstSpirit Server is to be managed using the cluster manager.

FirstSpirit Server can be defined as a standard resource using the cluster manager's graphical interface. In this case, the four control scripts are assigned to a newly created instance of the standard resource type with the name "FirstSpirit Server" and the provided configuration file `firstspirit.cfg` is copied to the directory where the control scripts are located. The three parameters for the control scripts (see Chapter 4.3.1, page 5) have to be adapted to the installation in this configuration file.

You can find additional notes on configuring the cluster manager in Chapter 5 starting on page 7.

4.3.3 Configuration files for a specific resource type

This configuration is recommended if multiple FirstSpirit Server instances are to be managed using the cluster manager.

The `FIRSTspiritAgent`, `FIRSTspirit.xml` and `FIRSTspiritTypes.cf` definition files are provided for defining a new "FirstSpirit Server" resource type in the cluster manager. Then an instance is created for each FirstSpirit Server from this resource type. Then the three parameters of the control scripts (see Chapter 4.3.1, page 5) have to be entered in the cluster manager's graphical interface.

You can find additional notes on configuring the cluster manager in Chapter 5 starting on page 7.



4.4 Starting FirstSpirit Server

After the "FirstSpirit Server" resource is defined, FirstSpirit Server can be started using the cluster manager's graphical interface.

In the event of problems, notes on the cause of errors can be found in the following FirstSpirit Server log files: `/opt/firstspirit5/log/fs-wrapper.log` and `/opt/firstspirit5/log/fs-server.log`.

If there are false alarms, the timeout parameters for the wrapper should be increased. These are modified in the file `/opt/firstspirit5/conf/fs-wrapper.conf`. In particular, the following parameters are critical: `wrapper.cpu.timeout` and `wrapper.ping.timeout`.

5 Configuration note for Veritas Storage Foundation HA

5.1.1 Introduction

VERITAS Storage Foundation HA consists of the following individual products:

- VERITAS Volume Manager
- VERITAS File System
- VERITAS Cluster Server

While VERITAS Volume Manager and VERITAS File System are used for making a shared (or, more specifically, switchable) storage system available, Cluster Server is used for managing the cluster's resources.

On one hand, resources include low-level resources that are managed fully automatically by VERITAS Cluster Server, such as

- Network card
- IP address
- Hard disk volume or volume group
- Mount point

Fully automatic in this context means that the failure of one such resource is not just detected, but is handled as well. This means it can be moved to another cluster server.



Additionally, complex resources such as applications can also be described in VERITAS Cluster Server. In order to be able to monitor an application and react to an outage, the following functions have to be made available by the application:

- Start script
- Stop script
- Clean script with "hard" termination of the process
- Monitor script for monitoring the application

These four control commands put VERITAS Cluster Server in a position to shift any application from one cluster node to another if the application has the following properties:

1. State saving only to a hard drive and/or database
2. Configuration isolation (see below)

5.1.2 Configuration of the resource group

A so-called resource group is defined in VERITAS Cluster Server using a graphical interface (VERITAS Cluster Explorer). A resource group is a hierarchical dependency tree, where the application (i.e. FirstSpirit in this case) has its root. All of the resources that the application needs are arranged below the root node. In the case of FirstSpirit, this is just the IP addresses of the FirstSpirit Server instances and the file system (depending on the DB: see below).

VERITAS Cluster Server then monitors each of the resource group's resources. As soon as one resource fails, the hierarchical dependency tree is checked from the failed node to the root to determine whether a critical resource is affected. A critical resource is considered to be a resource that has to be available at all times to be able to operate services established on it. In the event of an error, a critical resource is swapped to the failover system by the cluster server.

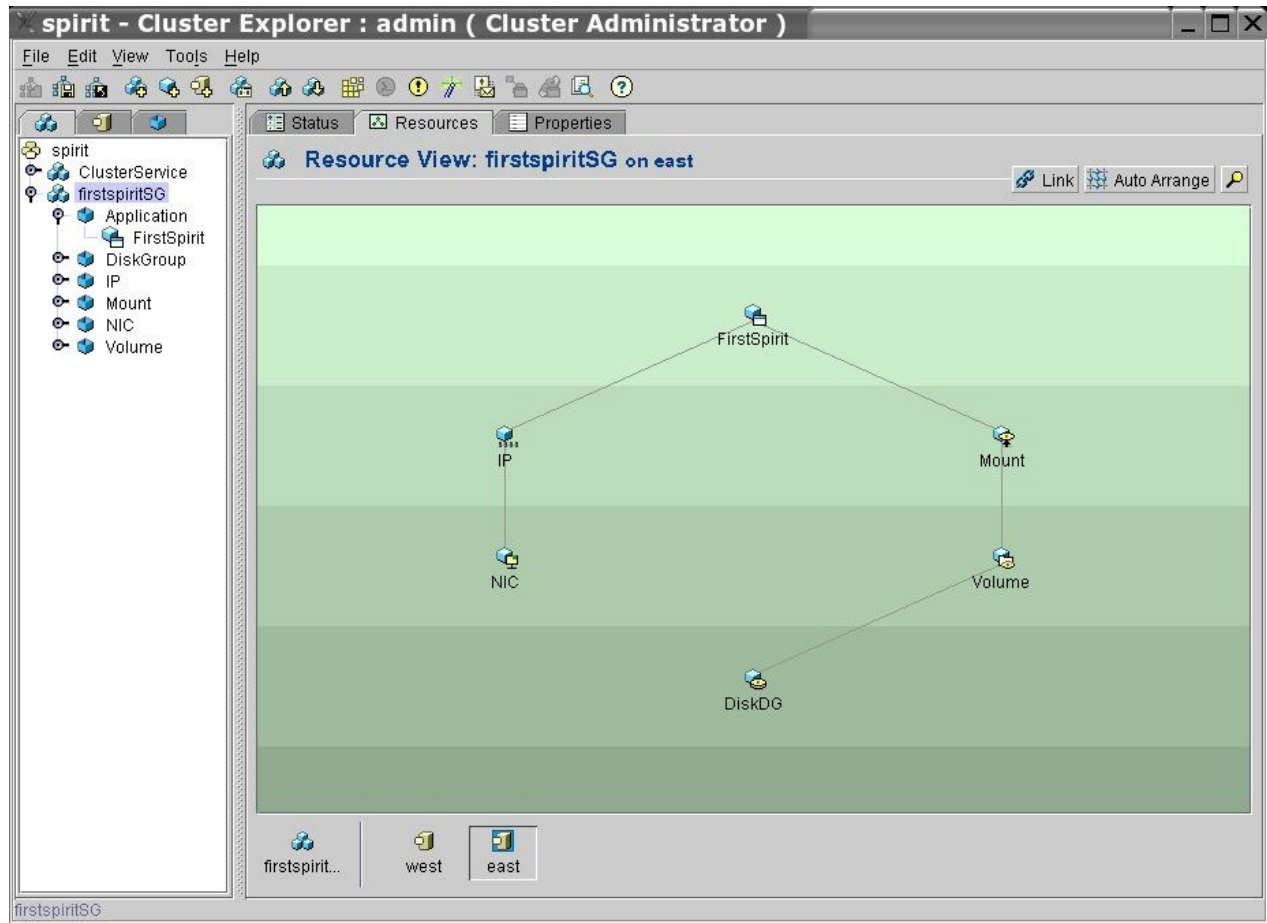
All of the resources, including FirstSpirit Server, are classified as "critical" for a FirstSpirit resource group. Therefore, the failure of one resource always results in the entire resource group transferring to the failover system.

The cluster server behaves as follows in the event of an error:

Starting from the failed resource, the smallest shared, "non-critical" part of the tree is sought. One cannot be found for FirstSpirit since all of the resources are critical. Therefore, the entire resource group is migrated. To accomplish this, the resources are stopped starting from the top of the tree and working down, first using the "Stop" script and then using "Clean" if "Stop" is not successful. The resources are then started in reverse order on the failover system.



The following model of the FirstSpirit resource group has to be created:



In this instance it is important to note that the database in the resource group definition is intentionally NOT taken into account because the database represents its own resource group.

Therefore, a resource group dependence between FirstSpirit and the database (e.g. Oracle) is established in the cluster server. The benefit of this architecture is that the high availability of FirstSpirit and the database are present in entirely separate form. Thus, in a multi-server environment for instance, a simple form of load distribution can be implemented by running the database on a cluster server other than the one FirstSpirit Server is running on.



6 Legal notices

FirstSpirit™ HighAvailability is a product of e-Spirit AG, Dortmund, Germany.

Only a license agreed upon with e-Spirit AG is valid with respect to the user for using the module.

Details regarding potential third-party software products in use not created by e-Spirit AG, their separate licenses and, if applicable, their update information can be found on the start page of every FirstSpirit™ server in the "Legal notices" area.

