

# Natural

## Natural Availability Server

Version 9.3.1

September 2025

This document applies to Natural Version 9.3.1 and all subsequent releases.

Specifications contained herein are subject to change and these changes will be reported in subsequent release notes or new editions.

Copyright © 1992-2025 Software GmbH, Darmstadt, Germany and/or its subsidiaries and/or its affiliates and/or their licensors.

The name Software AG and all Software GmbH product names are either trademarks or registered trademarks of Software GmbH and/or its subsidiaries and/or its affiliates and/or their licensors. Other company and product names mentioned herein may be trademarks of their respective owners.

Detailed information on trademarks and patents owned by Software GmbH and/or its subsidiaries is located at <https://softwareag.com/licenses>.

Use of this software is subject to adherence to Software GmbH's licensing conditions and terms. These terms are part of the product documentation, located at <https://softwareag.com/licenses> and/or in the root installation directory of the licensed product(s).

This software may include portions of third-party products. For third-party copyright notices, license terms, additional rights or restrictions, please refer to "License Texts, Copyright Notices and Disclaimers of Third-Party Products". For certain specific third-party license restrictions, please refer to section E of the Legal Notices available under "License Terms and Conditions for Use of Software GmbH Products / Copyright and Trademark Notices of Software GmbH Products". These documents are part of the product documentation, located at <https://softwareag.com/licenses> and/or in the root installation directory of the licensed product(s).

Use, reproduction, transfer, publication or disclosure is prohibited except as specifically provided for in your License Agreement with Software GmbH.

**Document ID: ATUX-NHA-931-20250930**

## Table of Contents

Natural Availability Server .....	v
1 About this Documentation .....	1
Document Conventions .....	2
Online Information and Support .....	2
Data Protection .....	3
2 Introduction .....	5
Components .....	6
The Server .....	7
The Client Web Application .....	7
3 Installing Natural Availability Server .....	9
4 Quick Start .....	11
5 Configuring the Natural Availability Server .....	13
Basic Configuration .....	14
6 Advanced Configuration .....	19
Encrypting Passwords .....	20
Configuration File Option: Class Path .....	21
7 Configuring the User Interface .....	23
8 Deploying Natural Availability Server on Application Server .....	25
9 Building a Docker Image .....	27
10 User Interface Manual .....	29
11 Troubleshooting .....	31



---

# Natural Availability Server

---

<b>Introduction</b>	What is the Natural Availability Server?  The concept, the architecture, and the components.
<b>Installing Natural Availability Server</b>	How to install the Natural Availability Server.
<b>Quick start</b>	Minimal settings to run the Natural Availability Server – not for production.
<b>Configuring the Natural Availability Server</b>	A walkthrough of the configuration parameters for Natural Availability Server.
<b>Advanced Configuration</b>	Insight into advanced application configuration.
<b>Configuring the User Interface</b>	Application configuration properties that are related to the user interface.
<b>Running the Natural Availability Server on Application Server</b>	How to deploy Natural Availability Server on Tomcat® and Wildfly® application servers.
<b>Building a Docker Image</b>	How to create a Docker image for Natural Availability Server.
<b>User Interface Manual</b>	All about the Natural Availability Server user interface for the end user.
<b>Troubleshooting</b>	In case you get stuck.

---

# 1

## About this Documentation

---

■ Document Conventions .....	2
■ Online Information and Support .....	2
■ Data Protection .....	3

## Document Conventions

---

Convention	Description
<b>Bold</b>	Identifies elements on a screen.
Monospace font	Identifies service names and locations in the format <i>folder.subfolder.service</i> , APIs, Java classes, methods, properties.
<i>Italic</i>	Identifies:  Variables for which you must supply values specific to your own situation or environment. New terms the first time they occur in the text. References to other documentation sources.
Monospace font	Identifies:  Text you must type in. Messages displayed by the system. Program code.
{ }	Indicates a set of choices from which you must choose one. Type only the information inside the curly braces. Do not type the { } symbols.
	Separates two mutually exclusive choices in a syntax line. Type one of these choices. Do not type the   symbol.
[ ]	Indicates one or more options. Type only the information inside the square brackets. Do not type the [ ] symbols.
...	Indicates that you can type multiple options of the same type. Type only the information. Do not type the ellipsis (...).

## Online Information and Support

---

### Product Documentation

You can find the product documentation on our documentation website at <https://documentation.softwareag.com>.

### Product Training

You can find helpful product training material on our Learning Portal at <https://learn.software-ag.com>.

### Tech Community

You can collaborate with Software GmbH experts on our Tech Community website at <https://tech-community.softwareag.com>. From here you can, for example:



- Browse through our vast knowledge base.
- Ask questions and find answers in our discussion forums.
- Get the latest Software GmbH news and announcements.
- Explore our communities.
- Go to our public GitHub and Docker repositories at <https://github.com/softwareag> and <https://hub.docker.com/publishers/softwareag> and discover additional Software GmbH resources.

## Product Support

Support for Software GmbH products is provided to licensed customers via our Empower Portal at <https://empower.softwareag.com>. Many services on this portal require that you have an account. If you do not yet have one, you can request it at <https://empower.softwareag.com/register>. Once you have an account, you can, for example:

- Download products, updates and fixes.
- Search the Knowledge Center for technical information and tips.
- Subscribe to early warnings and critical alerts.
- Open and update support incidents.
- Add product feature requests.

## Data Protection

---

Software AG products provide functionality with respect to processing of personal data according to the EU General Data Protection Regulation (GDPR). Where applicable, appropriate steps are documented in the respective administration documentation.



## 2 Introduction

---

■ Components .....	6
■ The Server .....	7
■ The Client Web Application .....	7

Natural Availability Server is a front-end application that provides a Natural emulator in a modernized Angular© and REST-based web application.

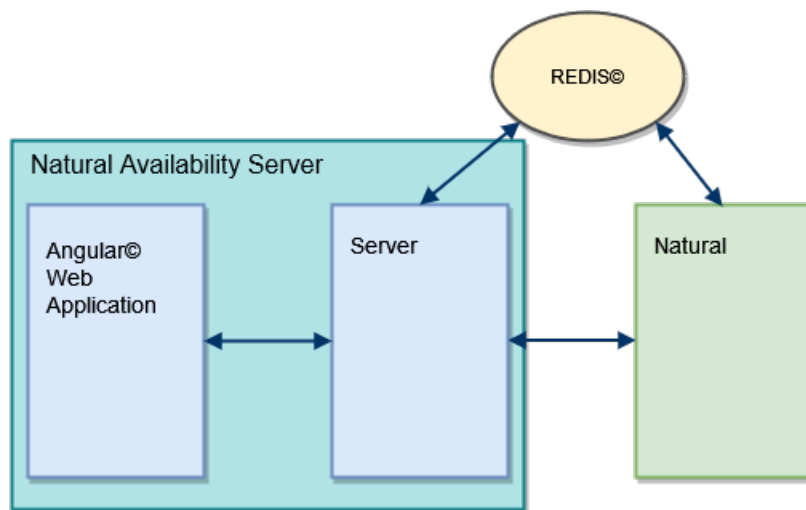
The Availability Server provides capabilities that enable Natural online applications to be operated in a scalable and highly available environment. Based on the availability and SLAs of the underlying infrastructure and the selected architecture, Natural online applications can be configured to achieve system availability of more than 99.99%.

The session context of Natural sessions can be externalized and stored inside a session store, e.g. a REDIS© State Server. This enables the usage of multiple Natural server instances or Linux containers and to place them behind load-balancers. Subsequent requests can be handled by any instance of the Natural Availability Server and any started Natural processes, as session context can be stored and loaded for each request from a distributed clustered in-memory State Server, e.g. REDIS©. For more information on High Availability for Natural, see *Natural/HA (High Availability)*.

## Components

---

The following high-level diagram illustrates the main components of the Natural Availability Server:



The Natural Availability Server has a client-server architecture.

The application is a Web frontend for Natural sessions, that provides High Availability capabilities.

## The Server

---

The server covers following tasks:

- Maintaining a session between the web application and the Natural host
- Synchronizing the data between the web application and the Natural host
- Serving the web user requests and to update the Natural host accordingly

There are two modes to run the server:

- [Non-HA Mode](#)
- [HA Mode](#)

### Non-HA Mode

In this mode the server keeps a session open between the web application and the natural host.

The session is created when the user logs in to the web application and it closes when the user disconnects, or when the program ends.

### HA Mode

When HA mode is enabled, each request can be served by any of the servers in the cluster.

All session data is read from the shared session store and stored back for each request.

The Natural session could possibly be closed after each request and a new Natural session will be created for the next request.

Currently, the Natural Availability Server is supporting only REDIS® as a shared session store.



**Note:** In HA mode, there is an option to set a timeout in the Natural host. The host will wait for a new request before closing the session. This option can be used for performance reasons.

## The Client Web Application

---

The client web application is a modern, web-based frontend that emulates Natural sessions.

More information on configuring the web application behavior can be found in the [User Interface Manual](#).



# 3

## Installing Natural Availability Server

---

Natural Availability Server is installed using the Software AG installer and is part of the Natural Product suite.

The installer for Natural Availability Server can be found under **Natural Products** in the installation tree.

The product requires a specific license file to be provided as part of the installation process. If you don't have one, please contact the Software AG sales department.

After Natural Availability Server is installed, a directory */NaturalAvailabilityServer* is placed under the */Software AG* installation directory.





## 4 Quick Start

---



**Note:** This section is for running the Natural Availability Server on an embedded Tomcat server. It should not be used for production.

After Natural Availability Server is installed, a directory */NaturalAvailabilityServer* is placed under the */Software AG* installation directory.

Follow the steps below to run the Natural Availability Server after installation.

1. Configure the connectivity to Natural (WebIO server) in the *application.properties* file under */NaturalAvailabilityServer/conf*.

In case the default connectivity parameters need to be adapted to your environment, refer to the explanation in the section [Configuring Connectivity to Natural Host](#) to learn how to update the *application.properties*.

2. Configure SSL-related parameters in the *application.properties* file:

- For connection with SSL, set the following properties:

```
com.softwareag.natural.chimera.connection.ssl=true  
com.softwareag.natural.chimera.connection.certificatePath[0]=<cert path>
```

For details, see section [Configuring Connectivity to Natural Host](#).

- For a connection without SSL, set the following properties:

```
com.softwareag.natural.chimera.connection.ssl=false
```



**Caution:** For development only. Not recommended for production!

3. Run `startup.sh` from the directory */NaturalAvailabilityServer/bin*.

These steps will run the Natural Availability Server using its embedded Tomcat®.

To connect to the Natural program from the Angular© web application browse to *http://localhost:8080*

# 5

## Configuring the Natural Availability Server

---

■ Basic Configuration .....	14
-----------------------------	----

This section covers all you need for a complete configuration of Natural Availability Server.

## Basic Configuration

---

After installation, the *application.properties* file will be located under the following directory:

*<installation location>/NaturalAvailabilityServer/conf*

This file supplies all the details required for running the Natural Availability Server, including the host's connectivity data, the type of sessions to be created (HA or not), and all other configurations.

- [Configuring Connectivity to Natural Host](#)
- [Configuring Connectivity to Natural Using SSL](#)
- [Setting the Logging Properties](#)
- [Setting the License Location](#)
- [Setting the Availability Mode](#)
- [Setting the REDIS Session Store Properties](#)

### Configuring Connectivity to Natural Host

The Natural Availability Server can connect in a secure way via SSL protocol, or in a less secure way without SSL.



**Caution:** The non-SSL connection is not recommended for production use and should only be used for development.

To connect to the Natural host, configure the connection parameters:

Property name	Description	Required	Default and Optional Values
com.softwareag.natural.chimera.connection.hostname	Hostname/IP address where the Natural instance is running	No	localhost
com.softwareag.natural.chimera.connection.port	Port the Natural Web I/O Interface daemon is listening on	No	2900
com.softwareag.natural.chimera.connection.ssl	Define if the Natural Web I/O Interface daemon is using an SSL connection	No	False
com.softwareag.natural.chimera.connection.parameters[0]	Define the parameters to start the Natural application  Could be more than one parameters	No	Empty value

Property name	Description	Required	Default and Optional Values
	The next parameter should start using 'parameters[1]' then 'parameters[2]' and so on		
com.softwareag.natural.chimera.connection.certificatePath[0]	Path for the server certificate used by Natural Web I/O Interface daemon  Location:  <i>\$NWODIR/\$NWONODE/server.cert.crt</i>  of the Natural Host machine	Only for SSL connection	-
com.softwareag.natural.chimera.connection.application	The Application script to call to start the session.	No	nwo.sl

### Configuring Connectivity to Natural Using SSL

For quick starting out-of-the-box, the Natural Availability Server is configured by default to use non-SSL communication to the Natural Web I/O server.

However, if you want to connect using SSL communication you need to explicitly enable SSL in the following property and change the parameter to `true`.

```
com.softwareag.natural.chimera.connection.ssl=true
```

The public certificate of the Natural Web I/O server needs to be available to enable SSL communication. This value is configured in the following property:

```
com.softwareag.natural.chimera.connection.certificatePath[0]
```

Note that a fully valid certification path is required to create a secure connection to the host. If the certificate used in the Natural Web I/O server has an unknown intermediate certificate, it should also be defined in the certificate path by assigning a different index, as in the following example:

```
com.softwareag.natural.chimera.connection.certificatePath[1]
```

## Setting the Logging Properties

These are the properties that are used to define the log level for each section of the application:

Property name	Description	Required	Default and Optional Values
logging.level.root	Default log level for the Application		INFO
logging.level.org.springframework.web	Log level for Web requests		INFO
logging.level.org.springframework.security	Log level for Spring security		DEBUG

In addition to the properties that can be defined in the *application.properties* file, you can customize the logging output by updating the file *logback-spring.xml* located in the directory *<installation dir>/NaturalAvailabilityServer/conf*. For more information about this configuration please visit <https://logback.qos.ch/manual/index.html> site and follow the configuration link.

## Setting the License Location

The following property defines the location of the license file:

```
com.softwareag.natural.web.frontend.server.config.licensePath
```

This property expects the fully qualified or relative path to the file.

In case this property is not set, there are optional possibilities to place the product license file:

1. If the SAG environment variable is defined, the application will try to find the license file under:

```
$SAG/common/conf  
$SAG/NaturalAvailabilityServer/conf
```

2. If these two directories do not contain the license file, the application will attempt to find the license file in the classpath.
3. The server will stop with an error message if the license file is not found.

Commonly, the name of the expected license file is *nha<version number>.xml*.

However, this name is not used if the license property is set in the *application.properties* file.

## Setting the Availability Mode

The following property sets the mode in which the Natural Availability Server will run.

```
com.softwareag.natural.web.frontend.server.config.sessionStoreType
```

There are two modes available:

1. **memory**

This is *NON-HA* mode.

`memory` indicates that the session data is saved in the memory and not in a session store.

2. **redis**

This is *HA* mode.

`redis` indicates that the session data is saved to the Redis@ session store.

The following section will explain how to define the properties if option REDIS is selected.

## Setting the REDIS Session Store Properties

The following properties allow the configuration of the REDIS service location, access credentials, and the use of SSL.

Property name	Description	Required	Default and Optional Values
<code>spring.data.redis.host</code>	Host name for the REDIS server	Required if <code>sessionStoreType</code> is defined as REDIS	
<code>spring.data.redis.port</code>	Port on which the REDIS server is listening	Required if <code>sessionStoreType</code> is defined as REDIS	
<code>spring.data.redis.ssl.enabled</code>	Set to true if the REDIS instance is using an SSL connection	Required if <code>sessionStoreType</code> is defined as REDIS	
<code>spring.data.redis.username</code>	The username that should be used to connect to REDIS	Required if <code>sessionStoreType</code> is defined as REDIS	
<code>spring.data.redis.password</code>	Password to connect to REDIS if defined  See also section <a href="#">Encrypting Passwords</a> below	Required if <code>sessionStoreType</code> is defined as REDIS	





# 6

## Advanced Configuration

---

■ Encrypting Passwords .....	20
■ Configuration File Option: Class Path .....	21

## Encrypting Passwords

---

Each password parameter should be encrypted in the `application.properties` file.

- [Procedure](#)
- [Example](#)

### Procedure

Follow these steps to encrypt a password:

1. Decide on which password to encrypt.
2. Run `encryptPassword.sh` under the `/bin` directory with the following command:

```
encryptPassword.sh --encryption-password <JASYPT_ENCRYPTOR_PASSWORD> <passphrase>
```

The `JASYPT_ENCRYPTOR_PASSWORD` property is the key to encrypt the passphrase. It can be any string combination.

3. Place the value received from running the shell script as the password value. This value should be wrapped in `'ENC()'` string to mark the fact that this password is encrypted.
4. In a running testing or production environment, the `JASYPT_ENCRYPTOR_PASSWORD` should be set as an environment variable.

### Example

In the following example, the required password for the `spring.data.redis.password` property is `'pass1'`, and the `JASYPT_ENCRYPTOR_PASSWORD` property was decided to be `'pass2'`.

Run the shell script as follows:

```
encryptPassword.sh --encryption-password pass2 pass1
```

The output received from the shell script could be `AAAABBBB44444FFFBB`.

This value should be set as property in the *application.properties* as follows:

```
spring.data.redis.password=ENC(AAAABBBB44444FFFBB)
```

This process must be done for each password used in the *application.properties* file.

The same `JASYPT_ENCRYPTOR_PASSWORD` property must be used for all encrypted properties.

When subsequently accessing the server that runs the application (for production or testing), an environment variable must be set for the key `JASYPT_ENCRYPTOR_PASSWORD` with the value `pass2`.

## Configuration File Option: Class Path

An alternative option to supply the configuration files (*application.properties* and *license file*) is by adding a resource directory to the `NaturalAvailabilityServer` class path in the servlet container:

- For WildFly®
- For Tomcat®

### For WildFly®

A module needs to be defined in order to add files to the class path:

1. Create a directory that contains the resource files.
2. Run the following command to add the module. The module name should stay as defined in the command:

```
/opt/jboss/wildfly/bin/jboss-cli.sh -c -command="module add
-name=natural-availability-server-configuration --resource-delimiter=,
--resources=<path to folder created>/application.properties, <path to folder
created>/nha<version-number>.xml"
```

### For Tomcat®

Configure the location of the *application.properties* file:

1. In the *application.properties*, set the `licensePath` property pointing to the location of the license file.
2. Create the following file under Tomcat® installation:

```
<tomcat installation>/conf/Catalina/myHost/NaturalAvailabilityServer.xml
```

... where *myHost* is the host name.

3. In this file *NaturalAvailabilityServer.xml*, add the following element:

```
<Context> <Environment name=" spring_config_location" value="<location of
application.properties>/application.properties" type="java.lang.String"/>
</Context>
```



# 7

## Configuring the User Interface

---

In addition to the aforementioned, the following properties are also part of the application.properties file.

However, these properties are related to the user interface and the Angular© web application:

Property name	Description	Required	Default and Optional Values
com.softwareag.natural.chimera.fields.input.trim	Defines if and how to trim the field's content when the screen is loaded. The trimChar property contains the value to trim.	No	Optional values:  none (default) Left Right both
com.softwareag.natural.chimera.fields.input.trimChar	The character to remove from the field's content.  <b>Note:</b> When defining a space character,	Yes, if the trim property is set.	

Property name	Description	Required	Default and Optional Values
	please set the value to " <code>\ </code> " (backslash and space).		
<code>com.softwareag.natural.client.config.selectOnFocus</code>	Define if focusing on an input field will select the full content of the field.	No	false
<code>com.softwareag.natural.client.config.themesList</code>	Define the list of themes available in the client UI. The themes define the look and feel of the application. There is an option to create a new CSS theme file and add it to the list.	No	This is the default theme list:  <code>emulator.css</code> <code>theme_dark.css</code> <code>theme_light.css</code>

# 8

## Deploying Natural Availability Server on Application Server

---

Natural Availability Server supports the following servlet containers:

- Tomcat®
- WildFly®

The `NaturalAvailabilityServer.war` can be found in the following directory:

*/NaturalAvailabilityServer/lib*

Follow these steps to run Natural Availability Server on a web server:

1. Configure the *application.properties* file (under `<installation location>\NaturalAvailabilityServer\conf`) following the instructions in the section [Configuring the Natural Availability Server](#).
2. Encrypt the passwords in the *application.properties* following the instructions in the section [Encrypting Passwords](#).
3. Place the *application.properties* (and, if needed, the license file) in a location visible to the `NaturalAvailabilityServer`.

There are two alternatives for this step:

- Add the file(s) to the *.war* before deploying the *.war*:

Navigate to the folder of the *NaturalAvailabilityServer.war* file.

Add the following directory structure *WEB-INF/classes* to your current directory.

Under *WEB-INF/classes* add the configuration files (*application.properties* and *nha<version>.xml*)

Run the command: `jar -u -f NaturalAvailabilityServer.war`

`WEB-INF/classes/application.properties WEB-INF/classes/nha93.xml` (assuming 9.3 is the version).

- Or ...

Follow the instructions described in the section [Advanced Configuration](#).

4. Copy the *.war* to the deployment folder in the web server:

Web Server	Directory
Tomcat®	<i>/Webapp</i>
WildFly®	<i>/standalone/deployments</i>

5. Start the web server application.
6. The Angular© web application will be available under the following URL:

*http://<server location>:<default port of web server>*



# 9

## Building a Docker Image

---

The docker image can be configured for the supported web servers.

There are a number of supported Servlet containers.

Wildfly® A Dockerfile that uses a standard Wildfly 29 image with Java17 is provided.

Tomcat® A Dockerfile using a Tomcat 10 image with Java17 is provided.

Embedded A Tomcat 10 embedded in the delivered war base image using Java17.

### > To Create the Docker Image

#### 1 Build the Image

The script *createDockerImage.sh* can be used to package the Natural Availability Server into an image.

Use the following command to create the image:

```
./createDockerImage [options]
```

The following options can be called:

```
--appserver      Name of the application server to use ↵  
("wildfly"|"tomcat"|"embedded") (default "embedded")  
--help          Help for this script
```

#### 2 Run the Image

The script *createDockerImage.sh* automatically creates a suitable docker run command. After generating an image successfully, the docker run command will appear at the end of the build's docker log.

To run the newly created Natural Availability Server as a container, a valid Natural Availability Server license is needed. The generated command will expect the file to be present in the directory */licenses*.

Example for version 9.3.1

```
podman run -dt -p 8080:8080 -v ↵
./licenses/nha93.xml:/opt/softwareag/common/conf/nha93.xml:z \
-v ↵
./embedded/application.properties:/opt/softwareag/NaturalAvailabilityServer/conf/application.properties ↵
natural_availability_server:9.3.1
```

### 3 Run the Web Application

Open a browser on the local machine and navigate to the URL *http://localhost:8080/*.

### 4 SSL

For use with SSL encryption, the configuration files have to be adjusted according to the manuals at *<https://docs.wildfly.org/>* or *<https://tomcat.apache.org/>*.

# 10

## User Interface Manual

---

Make sure server is running before accessing the Natural web application. To access the web application, in the default configuration, browse to:

*http(s)://<server location>:<default port>*

The Natural credentials are required to login to the web application.

The version property in the login screen indicates the current version that you are accessing. If this value is not retrieved, then there is a problem with accessing the Natural Availability Server.

After the credentials are applied, the Natural session starts.



**Example of screen in the Angular web application:**

- The main content of the screen is an emulation of the Natural screen.  
It provides input and output fields.
- On the right side of the screen, the current screen's PF keys are displayed as buttons.  
Additionally, each button expands a tooltip that displays the alternative command.
- The top menu shows the following buttons (left to right):
  - **Logoff**  
Closes the current session with Natural  
Redirects the web application to the login screen
  - **Theme**  
Optional look-and-feel flavors to adapt the UI
  - **Refresh**  
Synchronizes the web application content based on the latest state of the Natural session

# 11 Troubleshooting

---



## Issue:

When connecting to the URL `http://localhost:8080`, I got the message *"This site can't be reached"*.

## Solution:

Ensure that ...

... Natural Availability Server is started and listens to the port 8080.

... The firewall is not blocking this port



## Issue:

The server did not start and I got the following error in the logs

*"redis.clients.jedis.exceptions.JedisConnectionException: Failed to create socket"*.

## Issue:

You have configured the Natural Availability Server to use Redis® as a session store but Redis® cannot be reached.

Check that the following properties are pointing to a working instance of Redis®. You can connect to it from the machine running the Natural Availability Server.

```
# host name for the redis instance
spring.data.redis.host=<your_redis_host_address>
# password to connect to Redis if necessary
#spring.data.redis.password=<xxxxxxx>
# port the Redis server is listening on default is 6379
spring.data.redis.port=6379
# set to true if the redis instance is using SSL connection
spring.data.redis.ssl.enabled=false
# the password that should be used to connect to redis.
#spring.data.redis.password=ENC(encrypted password)
#spring.data.redis.password=clearText
```

```
#the username that should be used to connect to redis.  
#spring.data.redis.username=clearText
```

### Issue:

When trying to connect to a Natural session, I get the message *"Cannot connect to host"*.

### Solution:

Check that the Natural Availability Server can connect to the Natural Web IO (NWO daemon) at the address defined in the following properties :

```
# hostname/IP address where NWO daemon instance is running  
com.softwareag.natural.chimera.connection.hostname=localhost  
# port the NWO daemon is listening on  
com.softwareag.natural.chimera.connection.port=39388  
# define if the NWO daemon is using SSL connection  
com.softwareag.natural.chimera.connection.ssl=false  
# define the parameters to start the Natural application  
Com.softwareag.natural.chimera.connection.parameters[0]
```