



SecureAware 3.0

Technical Whitepaper

Version 200608281436

Introduction	2
Additional Information	3
Architecture.....	4
System Architecture	4
Application Architecture.....	5
System Requirements	7
Security.....	9
Server Configuration.....	9
Configuration Samples	12
Microsoft SQL Server 7, 2000 or 2005.....	12

Copyright © 2006 Neupart A/S. All rights reserved.

The author of this documentation is Neupart A/S. All information herein including text and graphics belongs to Neupart A/S unless stated otherwise and is protected by copyright laws in Denmark and international agreements.

Permission to quote this documentation in its entire form or partly is given under the premises that no changes are made and that information about this copyright is clearly stated on all copies. No material may be copied or distributed without explicit approval of Neupart A/S. Neupart A/S preserves the right to - at any time and without warning - make changes and/or improvements in the products mentioned.

Names of other companies and their products are or can be registered trademarks or trademarks that belong to their owners. Neupart and SecureAware logo and the name "SecureAware" are trademarks belonging to Neupart A/S.

The documentation is provided "as is", without warranty of any kind, express or implied, including but not limited to the warranties of merchantability, fitness for a particular purpose and non-infringement. In no event shall the authors or copyright holders be liable for any claim, damages or other liability, whether in an action of contract, tort or otherwise, arising from, out of or in connection with the documentation or the use or other dealings in the documentation. The documentation including graphics could contain inaccuracies or typographic errors. Furthermore there are no guarantees regarding results achieved by using this information.

All rights not explicitly mentioned herein are preserved.

Introduction

SecureAware is an information security intranet. The solution provides information security to your business and as a security manager, you get efficient tools to manage information security. Computer users in your organization receive necessary information, knowledge and build required awareness.

SecureAware 3.0

Technical Whitepaper

Complete Solution

SecureAware is a complete solution as we bundle our application together with all needed systems e.g. web server and database pre-configured. There is no need for any additional software licenses besides a server with a supported operating system.

Built using Industry Standards

SecureAware is built using widely accepted technology, standards and components such as

- Java J2EE 1.4 and Java Servlet Engine
- SQL RDBMS Database
- Extensible Markup Language (XML)
- HyperText Markup Language (HTML)
- Macromedia Flash Movies (SWF)
- Adobe Portal Document Format (PDF)

Additional Information

Web

Neupart has a website with further information. It is available in English, Danish and German. Find it using

<http://www.neupart.com/>

Email

Email for contacting sales: sales@neupart.com

Email for technical support: support@neupart.com

SecureAware 3.0

Technical Whitepaper

Architecture

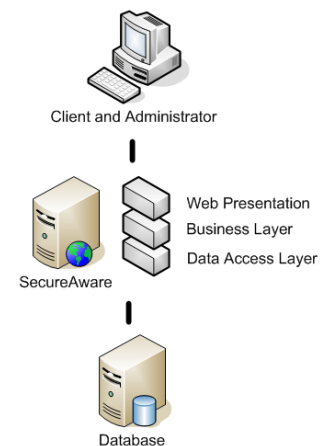
System Architecture

Web Application

SecureAware is a pure web application where the data is stored on a central server and users access it with a thin web client. SecureAware is installed with all required software embedded, like java runtime, a web server with embedded servlet support, and an in-memory database that is started together with the web server used.

SecureAware consists of multiple layers, each with a specific task. The presentation layer builds the screens, the business logic controls the application, the data access layer provides uniform access to the data, and the database layer stores all data.

The building database can be exchanged with a dedicated database server like MySQL, Microsoft SQL server or Oracle, either on the same server or located on an external server or cluster. Using an external database can normally be achieved by changing a few lines in the configuration file and providing a JDBC version 3.0 driver.



Open Source Components

The SecureAware application is bundled with several open source components, all with a strong foundation in the community and very well tested and used.

The database used is HSQLDB, found at <http://hsqldb.org/>

The servlet server used is Tomcat, found at <http://jakarta.apache.org/tomcat/>

The data access layer is Hibernate, found at <http://www.hibernate.org/>

License Issues

All the open source components have licenses that allow SecureAware to be sold as commercial software, and run by customers without license issues.

SecureAware 3.0

Technical Whitepaper

Neupart does not charge for the open source components themselves, but for the application running within these components.

No changes have been made to the open source components used without giving the fixes and changes back to the community as required by the licenses.

All open source licenses are included on the distribution media. Please see your media or the web sites referenced above for reading these.

Component Exchangeability

The use of layers gives the ability to exchange components by modifying configuration files and exchanging system files.

- Java runtime engine (default version 1.4)
- Database server (default HSQLDB)
- Web server with servlet engine (default Apache Tomcat)

Please contact Neupart A/S before modifying the configuration files. Modifying the SecureAware configuration without explicit authorization by Neupart A/S invalidates support.

Application Architecture

Front-end

The SecureAware front-end consists of 4 servlets running within the Tomcat servlet engine. The 4 servlets are Main for dynamic content and management, Report for generation of PDF, RTF and SAF reports and files, Screen Saver which handles web service requests from the screen saver clients, and the File that serves all static content like images, text files and JavaScript.

Static files are served through a servlet coded by Neupart, due to performance reasons as well as security.

The browser renders the HTML and displays the interface to the user.

SecureAware 3.0

Technical Whitepaper

Within the interface there are buttons and links. When the user presses one of these to e.g. delete a policy, this information is send as a HTML POST form to the servlet engine and parsed by relevant servlet.

Backend

The backend is running inside the web server, and is based on the Servlet standard and java objects. The backend checks all requests and based on the URL and session parameters it selects the correct user, portal, language and page to display. Each page contains a layout and areas where modules are located. Modules can be navigation elements like menus and links, or presentation elements like text, images, forms and editors.

The backend is organized into the portal framework, modules, session handling, cache, and database access. This division makes it easy to extend and customize.

Localization

The SecureAware data structures are separated 100% from any language specific text. The data structures contain labels in the form of e.g. /SA/POLICY/NAME, which are then replaced within the data access layer as the final step before sending the data to the browser.

The replacement text is taken from the database tables containing the English, Danish, Swedish, Norwegian and German texts depending on the language selected in the user interface for the particular users session.

This enables Neupart to add new languages to the system by translating text in a few database tables, as well as being able to fix textual errors without modifying application code.

Further the content is divided into application content (labels, text on buttons, help texts etc.), and content that the user is able to change like names of standards, rules etc.

The content is stored with a version number so SecureAware is able to separate content generated by different providers like Neupart and the customer. This ensures that content is not overwritten when the application is upgraded and that all changes can be revoked. In a future version the end user will be able to access and select a text between the stored versions in the database.

System Requirements

Server Hardware

Minimum requirements for the server:

Free memory for SecureAware at server at least 256MB; 512MB recommended.

Disk space: At least 300MB free; 600MB recommended.

3 GHz Pentium 4 or Xeon CPU or similar is recommended.

The hardware requirements are defined for a single portal system. The SecureAware server is able to operate multiple portals (completely isolated installations) from the same code base, but these portals extend the requirements for disk space and available memory.

For each extra portal you should add:

100MB free memory for SecureAware

50MB free disk space

Please note that the current Java engine is not able to use more than 2GB of memory, so the standard installed setup has a maximum configuration of approximate 10 extra SecureAware portals on a single server with 2.5 GB memory.

The build-in database loads all data into memory when the SecureAware is started, and this can lead to a delay before the first page is shown when the database file grows.

When operating a multi portal server we recommend that you use an external database, either installed on the same server or on a remote server. If the database is installed on the same server, a dual processor server will improve performance.

When using an external the memory required for each portal will be 30MB, enabling a single server to operate more portals.

Server Performance Considerations

The most important performance parameter is available memory, secondary the disk system, and then the server processors.

Remember to allocate more memory to SecureAware (see relevant section below) to utilize the memory, if available. This can easily be done with the SecureAware manager.

The entire application and server environment is also threaded, and can therefore utilize multiple CPUs. Having an SMP server with slow CPUs is better than a single fast CPU.

SecureAware 3.0

Technical Whitepaper

For most purposes though the minimum requirements are adequate enough for small sized installations (a few hundred users), but it all depends on the usage of the system. 10 users hitting the system hard every 2 minutes can make the system perform worse than 5000 users only using the system once a month.

Every user on the system reserves a minor part of the server's memory to hold status and session information, so the memory usage increases with the number of concurrent sessions that are in use on the server. If you are using the recommend setting the system should be able to handle more than 5000 concurrent sessions.

SCSI based disk systems normally performs better than ATA based, and multiple disk systems controlled by a RAID controller normally performs better than a single disk system.

When separating parts of the SecureAware application on more than one server, the network traffic can be a limiting factor. Ensure that the path between the servers are short and that the needed bandwidth can be ensured.

Server Software

For the standard installation the only requirement is a supported operating system:

Windows XP/2000, Windows Server 2000/2003, or Redhat Linux 7 or higher.

Client Software

For clients the browser requirements are:

- MS Internet Explorer 5.5 or higher
- Firefox 1 or higher

For the animated end user e-learning, a flash player version 4 or higher is required.

Unsupported

The following server software is known to work with SecureAware, but is unsupported and not guaranteed to work.

Databases

MySql 4.x

SecureAware 3.0

Technical Whitepaper

Virtual Machines

IBM JRE 1.4, BEA JRockit 1.4

Security

Authentication and Security Roles

In SecureAware we use a single database module, where the username and password of users allowed access are contained in the database. If required the handling of user authentication can be performed by an external LDAP provider, like Microsoft Active Directory, or by a front end web server like Microsoft Internet Information Server. Read more about the security modes and configuration in the separate SecureAware security document.

Data Consistency

All commands that update the database are contained within transactions. Should any part of the update fail for unknown reasons, the entire update is rolled back so that the data structures within the database are protected.

Application Updates

Neupart constantly monitors for security updates in the components used within SecureAware. Updates to SecureAware include these component updates.

Server Configuration

Files and Directory Layout

The important directories are depicted below

Within the bin directory the startup/shutdown files are located as well as the files needed for installing/uninstalling SecureAware as a service.

The jre directory contains a Sun JRE for Intel processors on Linux or Windows (depending on installation). It is possible to run SecureAware on different platforms by replacing this JRE with one for the target platform, but this is not supported by Neupart.

SecureAware 3.0

Technical Whitepaper

The webapps directory contains the actual SecureAware software. The conf directory contains all Apache Tomcat configurations, the windows/database directory contains the database files, and the log directory contains all server logs.

The temp and work directory contains temporary data while the server is running.

Open Ports

SecureAware is configured to run as an http server on port 8080. This setting is configurable within the file “server.xml” in the conf folder. This port is the only port that is required to be accessible from the outside (seen from the server). Tomcat itself uses port 8005 for the shutdown signaling, but it should be open from the outside.

If a front-end web server is used the redirection is default running on port 8009.

All Tomcat ports are movable by altering the XML configuration files. Changes to Tomcat configuration are not guaranteed to be preserved when upgrading SecureAware.

Webserver Port Configuration

The Tomcat servlet server acts as web server. It is configured to run off port 8080 as default, so that SecureAware can operate besides Apache and MS Internet Information Server on the same physical server. It is possible to configure SecureAware to be accessible from a different port, e.g. port 80 within the server.xml file.

Memory Configuration

The Java Virtual Machine gets a finite amount of memory allocated to run the application in. This amount is specified on startup and is not the amount of physical RAM in the server. It is recommended that the amount of RAM given to the JVM is maximum half the sizes of the physical RAM.

SecureAware 3 ships with approximately 100 MB allocated for heaps and stacks. For all production installation this should be changed within the manager right after installation, to at least 256 MB.

Increasing the amount of memory can increase the performance of SecureAware and is recommended for machines with large amounts of physical RAM.

SecureAware 3.0

Technical Whitepaper

Changes to the configuration are not guaranteed to be preserved when upgrading SecureAware.

Accessing SecureAware via Proxy

When setting up a proxy for accessing SecureAware it is important to not modify the URL pattern after the hostname and port. The URL pattern is used by SecureAware to identify what data resources should be fetched and send to the browser.

It is ok to proxy map hostname and ports like

```
proxy.external.net:80 -> realserver.internal.net:8080
```

Be aware that the pages in SecureAware are dynamically generated and that caching is handled inside tomcat, so any proxy should be set to allow tomcat to control the caching.

Deploying on other Servlet engines

Deploying SecureAware on other J2EE servers or servlet engines should be possible with no internal changes in SecureAware, only the deployment scripts for the target should be added.

The deployment of SecureAware in version 3 is done without packing the application into an archive, which increases the overall performance of the application. If required by other engines the application can be placed inside a WAR archive, but special care should be taken.

SecureAware is installed in the root of the tomcat server, which means that all URL references in the application is relative to the root “/”. If the application is packed into a WAR archive it will normally be deployed into a virtual subfolder called “/sa/” if the name is “sa.war” – which will not work. The default placement should be overwritten by the deployment scriptures so the application is deployed in the root.

Configuration Samples

Microsoft SQL Server 7, 2000 or 2005

The SecureAware standard installation contains all files that are required to operate the application on the Microsoft database, even the JDBC driver.

Database installation

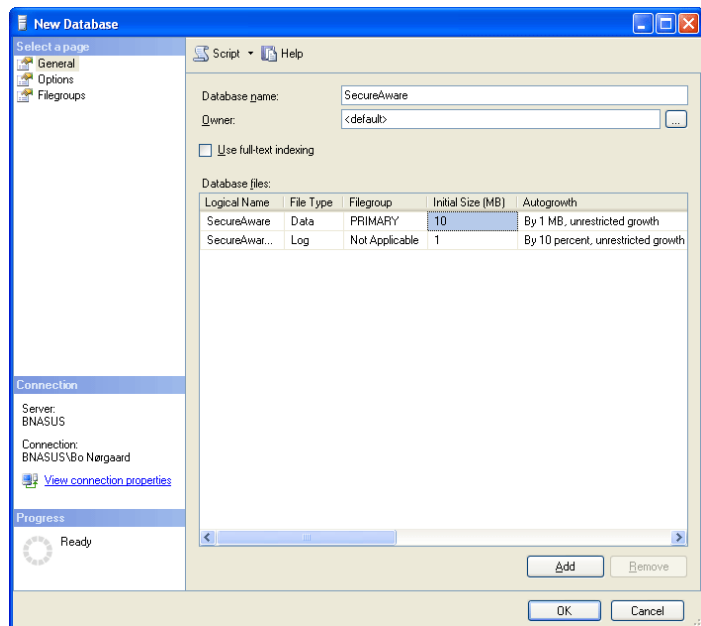
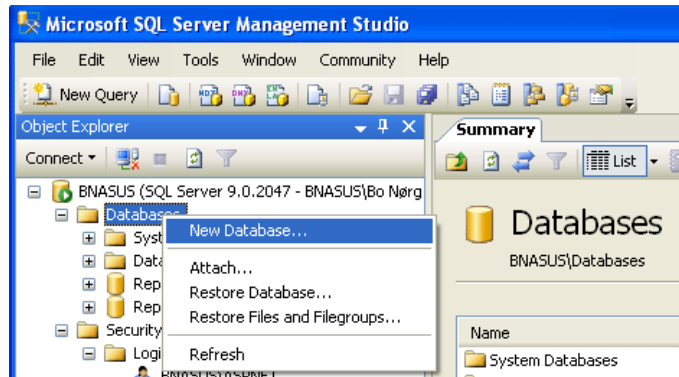
Ensure that the database is installed and working, and if the SecureAware server and the SQL Server are not located on the same server, do ensure that the connection port (default 1433) is not blocked by firewalls.

Create a database

Create a new database called "SecureAware" in your SQL server. Tables and content will be created by SecureAware on installation and upgrades.

On Microsoft SQL server 2005 you start the SQL server management studio application, right click the databases item in the object explorer tree view, then select the option "New Database..."

In the new database dialog, enter "SecureAware" as the database name, and change the Initial size of the data file to 10MB which will fit the initial installation. .



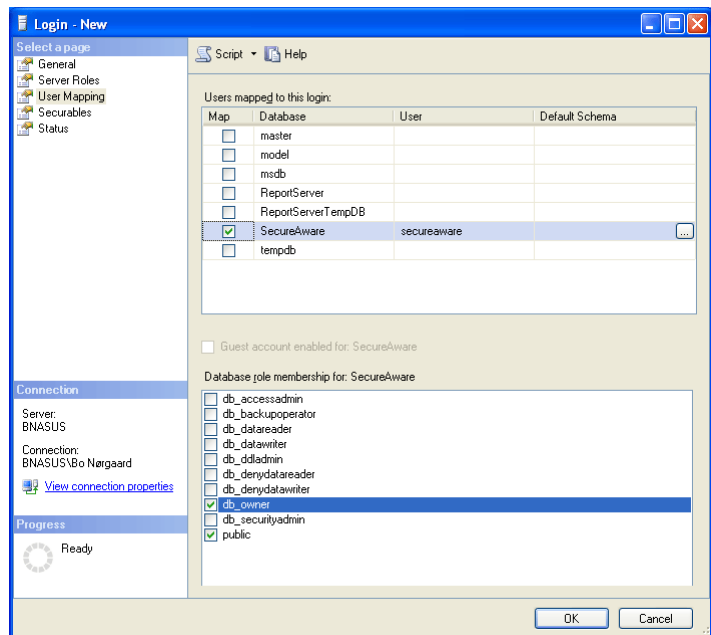
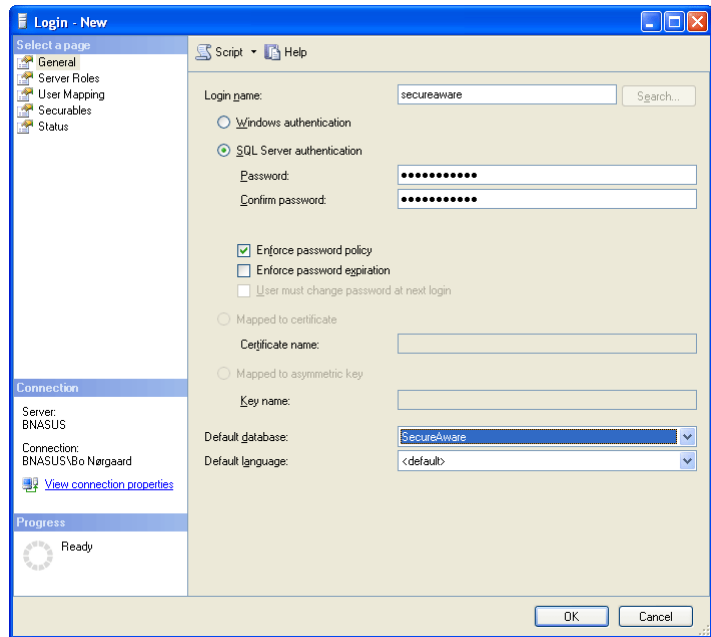
Create a user for SecureAware

For the SecureAware database connection we need a database user account. In this sample we use a user called “secureaware” with the password “secureaware” (which is not recommended for production systems). The database user should have database owner rights during installations and upgrades, but only needs select, update and delete rights during normal operation.

In MS SQL 2005 the user is also created in the management studio application. In the security folder you right click the login item and select to create a new login.

In the new login dialog you enter “secureaware” as the login name. The new user should then be configured to use SQL server authentication, which lets you enter “secureaware” as password. You need to disable the password expiration feature or this will require regular configuration changes. You can set the default database to be the SecureAware database.

Then you map the user to the SecureAware database, and when mapped you add the role db_owner which ensures that the login user is able to create and maintain the data model (create and alter table rights).

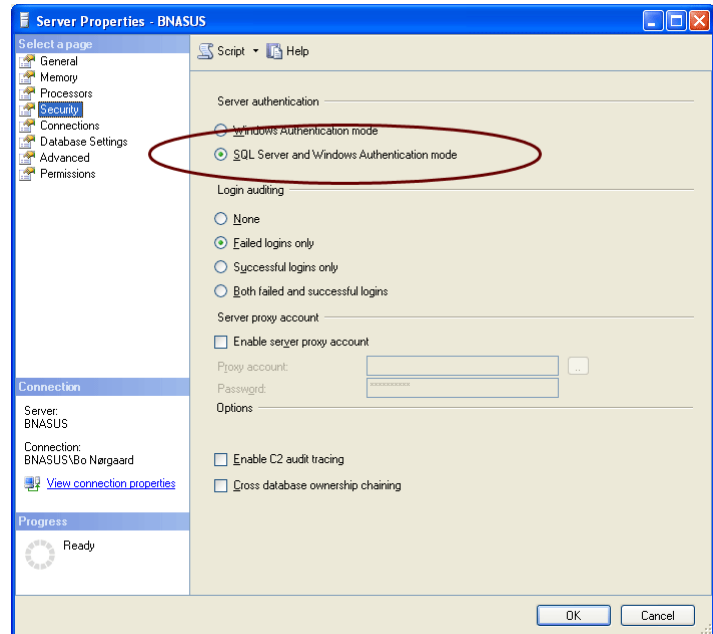


As we set the login to use the SQL server authentication, we need to ensure that the database was installed to allow this authentication method.

Be aware that a default installed Microsoft SQL server only enables Windows integrated security, so you have to manually select to enable the SQL authentication mode during installation.

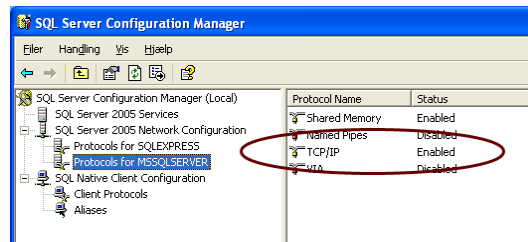
In SQL server 2005 you can change this from within the management studio application, by selecting server properties on the database.

In the Security area you can select to enable both SQL server and Windows authentication mode.



As the SecureAware application uses a JDBC driver that is based on creating a database connection using TCP, we need to ensure that the SQL server enables this. In SQL 2005 the connection settings are controlled in the SQL server configuration manager application, which can be found in the start menu.

A standard installed SQL server 2005 does not enable TCP connections, only the shared memory connection method. And when you enable the TCP connection you can also configure which port is used (default 1433).



Remember that after making changes with the SQL server configuration manager, you need to restart the SQL server service to activate your changes.

SecureAware configuration changes

Stop the SecureAware service before making any changes to the configuration files.

To change the database used you must edit the “cateline.properties” file in the conf folder with a standard text editor like notepad. The first section is the configuration of the in memory database HSQLDB, and should be disabled by placing a # character in front of all the lines.

Section two is the MS SQL server settings and should all be enabled by removing the # character that is in front of all the lines.

Correct the user name in “hibernate.connection.username” and the password in “hibernate.connection.password” to the one created on the SQL server. The connection URL should be changed to match your settings,

```
hibernate.connection.url jdbc:jtds:sqlserver://localhost:1433/secureaware;tds=8.0;lastupdatecount=true
```

The “localhost” is the name of the server running MS SQL, 1433 is the port, and SecureAware is the catalog (database) name. The last parameters are needed by the driver to optimize the communication.

In SecureAware versions above 3.0.4 the configuration was extended to include two database connections (see separate document for upgrade instructions), one for the normal SecureAware data, and one for the document database. You can make both points to the same database if you don't need to store the large binary document objects in a separate place.

If you make two separate databases, you could optimize the SecureAware database for the random access and update of small texts using joins and advanced content filtering. The Document database could be optimized for storing simple tables containing large binary objects.

SecureAware 3.0

Technical Whitepaper

Here is the configuration that would work with the setup we described here, on a SQL server installed on the same machine as SecureAware and both connections pointing at the same database.

```
## MS SQL Server
hibernate.dialect org.hibernate.dialect.SQLServerDialect
hibernate.connection.driver_class net.sourceforge.jtds.jdbc.Driver
hibernate.connection.username secureaware
hibernate.connection.password secureaware
hibernate.default_schema dbo
hibernate.default_catalog secureaware
hibernate.connection.url jdbc:jtds:sqlserver://localhost:1433/secureaware;tds=8.0;lastupdatecount=true

document.dialect org.hibernate.dialect.SQLServerDialect
document.connection.driver_class net.sourceforge.jtds.jdbc.Driver
document.connection.username secureaware
document.connection.password secureaware
document.default_schema dbo
document.default_catalog secureaware
document.connection.url jdbc:jtds:sqlserver://localhost:1433/secureaware;tds=8.0;lastupdatecount=true
```