



**AGETOR®**

**AGETOR® Security Service**

**User guide**

**Contents**

1	AGETOR® Security service .....	1
2	Service command line .....	1
3	Configuration file .....	1
4	Permission.....	3
5	Local security instance .....	3

## 1 AGETOR® Security service

ADK 2.0 provide you a simple Security service. The service is compatible with the Broker security model, which means that it can be used in order to enable the Broker security.

- ☞ This is not a replacement of the Extraware security service, which, opposite the new simple security service, allows adding and removing users and permissions via a graphical interface. Instead this service is an alternative to the Extraware security service for instance if you do not need Extraware, but you want to enable the Broker security.

The configuration of the service is done via an XML configuration file.

## 2 Service command line

The service accepts two extra parameters in addition to the parameters accepted by a standard ORB services.

parameter	explanation	Default
--config:<configfile>	The configuration file where the user rights and permission are stored	\${INSIDE_HOME}/conf/security.cfg
--log:<logfile>	The log file	\${INSIDE_HOME}/log/simplesecurity.log

A few words about the log file. The service tries to maintain a clean log file. This means that you get all the unformatted output (which is not needed in the log) to the System out, while only the relevant information is wrote to the log. This means that if you configure the service in the ServiceRunner, then set the ServiceRunner log option to another file e.g.

```

<PROGRAM
    RPORT="10"
    CLASS="dk.bording.inside.security.SimpleSecurityService"
    TYPE="orb"
    DESCRIPTION="mainsecurity"
    LOGFILE="mainsecurity.out"
>
    <PARAM NAME="config " VALUE="mainsecurity.cfg" />
    <PARAM NAME="log" VALUE="mainsecurity.log" />
</PROGRAM>
    
```

## 3 Configuration file

The configuration file is an XML file where you can configure user rights and permissions.

The users are configured via the XML element: USER. In this element you can configure the:

- The user name via the attribute NAME.
- The user password via the attribute PASSWORD.

- The user environment via the attribute ENV.
- The user permission via the attribute PERMISSION.

Example:

```
<USER
    NAME="userid"
    PASSWORD="userpassword"
    ENV="inside"
    PERMISSION="1:2:3"
/>
```

The question numbers are configured via the XML element: QUESTION. In this element you can configure the:

- The question number via the attribute QNO.
- The question environment via the attribute ENV.
- The question permission via the attribute PERMISSION.

If the ENV attribute is omitted then question number is valid for any environments.

Example:

```
<QUESTION
    QNO="2101"
    ENV="inside"
    PERMISSION="1:2:3"
/>
```

For example this could be the configuration in order to access the Extraware menu and lookup service.

```
<SECURITY>
```

```
<USER NAME="anonymous" ENV="extraware" PASSWORD="" PERMISSION=":0" />
<USER NAME="user1" ENV="extraware" PASSWORD="user1" PERMISSION=":0:1:2:5" />
```

```
<!-- menu 213 -->
```

```
<QUESTION QNO="213" ENV="extraware" PERMISSION=":5" />
```

```
<!-- menu 212 -->
```

```
<QUESTION QNO="212" ENV="extraware" PERMISSION=":1:0:5:2" />
```

```
<!-- lookup 461 -->
```

```
<QUESTION QNO="461" ENV="extraware" PERMISSION=":1:0:5:2" />
```

```
<!-- lookup 460 -->
```

```
<QUESTION QNO="460" ENV="extraware" PERMISSION=":0" />
```

&lt;/SECURITY&gt;

## **4 Permission**

A number represents the permissions. When a user and the question have the same number, the user is enabled to invoke a method with the corresponding question number.

## **5 Local security instance**

If you need security identification but you do not need a remote service then the class can also be used in your code just like all the other java classes. Refer to the SimpleSecurityService API for extra information.