EUROPEAN COMMISSION
DIRECTORATE-GENERAL
INFORMATICS
Directorate D - Digital Services
Trans-European Services

European Commission

ECAS Client Installation and Configuration Guide - Basic For WebLogic Server 10.3 and above

Date:	11/05/2017
Version:	4.16.1
Authors:	Dominique Laurent, Michaël Manalis, Dévid Verfaillie, Grégory Dony, Olivier Peters
Revised by:	
Approved by:	
Public:	
Reference Number:	

TABLE OF CONTENTS

TABLE OF FIGURES

Document History

Version	***Author***	***Date***	***Comment***	***Mod. Pages***
1.3.0	lauredo	13/02/2007	Reflected changes in ECAS client 1.3.0	
1.8.5	ackxyyo manalmi	02/07/2008	Skimmed version of the Developer Kit Entirely revamped for version 1.8.3	All
1.8.5	ackxyyo	28/10/2008	reSubmitPosts	19, 23, 36
1.8.6	ackxyyo	12/01/2009	Change CVS to SVN Release 1.8.6	2, 3
1.8.6	ackxyyo	09/02/2009	Added reference to Gateway	53
1.8.7	lauredo	13/03/2009	1. Modified the links to point to the IAM wiki and the ECAS Forge instead of CITnet 2. Removed the *serverName* property in the basic configuration 3. Added the *excludedContextPaths* paragraph 4. Explained why the validateUrl should never be configured to go through a reverse proxy 5. Added the *UserDetailsExtraGroupHandlerIntf* section 6. Updated the logging section 7. Added the *Listen Address* and *Frontend Host* explanation in the serverName section 8. Updated Weblogic Security Debug FAQ 9. Added new FAQs	3, 11, 16, 17, 21, 26, 29, 36, 37, 56, 66
	duponbn	18/03/2009	Review	3, 16, 42, 56, 57, 61, 67
1.9.0	lauredo	02/06/2009	1. Behavioural change: conventional name for configuration files 2. New constants for employeeType and domain in validation responses 3. New HttpRedirector interface	
	verfade	05/06/2009	Typos, lay-out, small additions.	All
1.9.0	verfade	22/06/2009	Changes for v. 1.9.	
1.9.0	verfade	21/08/2009	Moved advanced documentation to [ECAS-ADV]. Changed *ecas-test* to *ecas-demo*.	
1.9.0	ackxyyo	09/09/2009	Review for v. 1.9 Split chapter 4 into multiple 3 chapters	All
1.9.1	verfade	15/11/2009	Changes for v. 1.9.1.	3
1.10.0	guittri	16/03/2010	Changes for v. 1.10.0. Extract "What's new" in a new document	iv, 3, 4
1.11.0	verfade	10/05/2010	Changes for v. 1.11.0.	
1.13.0	verfade	03/06/2010	Changes for v. 1.13.0.	
1.13.3	lauredo	17/09/2010	Changes for v. 1.13.3.	

Version	*Author*	*Date*	*Comment*	*Mod. Pages*
1.14.0	lauredo	08/11/2010	Updated the URLs for ECAS and CITnet	2
1.15.0	verfade	08/12/2010	Changes for v. 1.15.0.	
1.16.1	verfade	07/02/2011	Changes for v. 1.16.0 and 1.16.1.	
1.18.1	verfade	01/06/2011	Changes for v. 1.18.1.	
1.20.0	lauredo	21/06/2011	Fixed CITnet links to use webgate.ec.europa.eu	
1.20.1	verfade	19/07/2011	Changes for v. 1.20.1.	
3.1.0	donydgr	08/06/2012	Update URLs for ECAS Changes for v. 3.1.0	2
3.6.0	lauredo	13/12/2013	Changes for v.3.6.3 Added the new TESTA URL.	2
3.9.0	catizmi	04/09/2014	Changes for v.3.9.0 Fixed styles in section 4.3	5-11
3.9.1	catizmi	11/09/2014	Changes for v.3.9.1	
3.11.1	catizmi	11/12/2014	Changes for v.3.11.1	
3.11.2	catizmi	18/12/2014	Changes for v.3.11.2, general improvements	All
4.3.0	catizmi	12/06/2015	Changes for v.4.3.0	
4.3.1	lauredo	17/06/2015	Changes for v.4.3.1	
4.3.2	catizmi	29/07/2015	Changes for v.4.3.2	
4.3.4	lauredo	13/10/2015	Changes for v.4.3.4	
4.4.1	lauredo	01/12/2015	Changes for v.4.4.1	
4.5.1	lauredo	07/01/2016	Changes for v.4.5.1	
4.5.2	lauredo	21/01/2016	Changes for v.4.5.2	
4.6.0	lauredo	08/02/2016	Changes for v.4.6.0	
4.6.1	catizmi	01/04/2016	Added a note to §6.5 about ecas-demo	18
4.8.0	lauredo	18/07/2016	Changes for v.4.8.0	
4.10.1	lauredo	25/10/2016	Changes for v.4.10.1	
4.12.0	lauredo	15/12/2016	Changes for v.4.12.0	
4.12.1	lauredo	19/01/2017	Changes for v.4.12.1	
4.13.0	lauredo	13/02/2017	Changes for v.4.13.0	
4.14.0	lauredo	08/03/2017	Changes for v.4.14.0	
4.15.0	lauredo	20/03/2017	Changes for v.4.15.0	
4.16.1	lauredo	11/05/2017	Changes for v.4.16.1	

Reference Documents

Code	***Title***
[ECAS-NEWS]	ECAS Client What's New
[ECAS-BASIC]	ECAS Client Installation and Configuration Guide – Basic (= this document)
[ECAS-ADV]	ECAS Client Installation and Configuration Guide – Advanced
[ECAS-TECH]	ECAS Technical Guide
[GATEWAY]	ECAS Gateway (Peek for SSO)
[SIGNATURE]	ECAS Signature
[ECAS-FORGE]	https://webgate.ec.europa.eu/CITnet/confluence/display/IAM/ECAS+Forge All the above-mentioned documents are available at this location.

Contact:

EC-IAM-SERVICE-DESK@ec.europa.eu
DIGIT-ECAS-DEVELOPMENT@ec.europa.eu

1. INTRODUCTION

1.1. What is ECAS

ECAS stands for **E**uropean **C**ommission **A**uthentication **S**ervice.

ECAS is based on the Central Authentication Service (CAS) version 2 developed at Yale University[1].

It is an authentication service to protect Web-based applications.

This guide explains how to install the ECAS Client for WebLogic version 9.2 and above in order to protect your applications.

Although a detailed knowledge of CAS and ECAS internals is not mandatory to proceed with the client installation, the reader is warmly invited to at least get familiar with the ECAS basics. Please refer to [ECAS-TECH] for an introduction to the ECAS technical aspects.

1.2. ECAS Client installation

This guide is divided in two parts:

- the first (chapters 4, 5 "Installing the EcasIdentityAsserterV2" and 6 in this document) deals with a basic, no-fuss ECAS Client installation with a minimal configuration which works. It is a good starting point to get acquainted with ECAS;
- the second ([ECAS-ADV]) explains how to extend the basic configuration to suit your needs.

2. BILL OF MATERIALS

The ECAS Client for WebLogic 10.3 is delivered as a WebLogic Identity Assertion Provider V2 (called *EcasIdentityAsserterV2*).

You can download the ECAS Client from the [ECAS-FORGE] on the I&AM Wiki: https://webgate.ec.europa.eu/CITnet/confluence/display/IAM/Downloads-WebLogic.

An ECAS Client release is composed of the following files:

1) The ECAS Client JAR *ecas-weblogic-10.3-authprovider-4.16.1.jar*: the EcasIdentityAsserterV2 for WebLogic Server 10.3 and above
2) The demo application *ecas-demo.ear*
3) The *security.properties* file: a customized resource bundle to provide descriptions for the EcasIdentityAsserterV2 provider specific page in the WebLogic admin console[2]
4) The patched log4j JAR *log4j-1.2.15.jar* (we patched the broken manifest file; more info at https://issues.apache.org/bugzilla/show_bug.cgi?id=44370)
5) The example log4j configuration file *log4j.xml* (to be adapted to your environment)
6) The basic installation guide *[ECAS-BASIC]* (the document you are currently reading)

[1] See http://www.jasig.org/cas for more information.

[2] From WLS 9 and above, the custom security provider MBean attribute descriptions are not displayed anymore. This is a pity because having an inline help next to each field can be most helpful. If you want those descriptions to appear again, you have to add our customised resource bundle to the Admin Console folder of your server. See 5.2 Copy the files into your WebLogic Server instance.

7) The advanced installation guide *[ECAS-ADV]*

8) The technical guide *[ECAS-TECH]*

9) Other documentation files about specific aspects such as *[GATEWAY]*, *[SIGNATURE]*…

It is important you download the *ecas-demo.ear*. This demo application is protected by ECAS and will be frequently used in this guide to illustrate various ECAS Client features.

Should you be interested in the source code, it is available in CITnet Subversion at https://webgate.ec.europa.eu/CITnet/svn/ecas-public/clients/java/tags/.

If you have any comment or question after reading this documentation, please share it on the CITnet ECAS forum (https://webgate.ec.europa.eu/CITnet/jforum/forums/show/35.page) or drop us an email at DIGIT-ECAS-DEVELOPMENT@ec.europa.eu.

3. IMPORTANT NOTES

3.1. About the URLs

This document mentions links to CITnet (the European Commission's Collaborative IT network) which hosts the ECAS project and its forge.

CITnet URL is https://webgate.ec.europa.eu/CITnet/.

The ECAS servers are accessible through all reverse proxies at the Commission.

Hence please adapt the ECAS links mentioned in this document to your environment and your location:

- From within the European Commission, please use either:
  - https://ecas.cc.cec.eu.int:7002/cas (direct, internal access only)
  - https://ecas.ec.europa.eu/cas (proxied, internal and external access)
  - Or https://www.cc.cec/cas (proxied, internal access only)
  - Or https://intragate.ec.europa.eu/cas (proxied, internal access only)
  - Or https://webgate.ec.europa.eu/cas (proxied, internal and external access)
- From outside and for civil servants, please use:
  - https://intracomm.ec.europa.eu/cas (Officials only, proxied, external access only)
- From outside and for trusted contractors, please use
  - https://webgate.ec.europa.eu/cas (proxied, internal and external access)
  - Or https://ecas.ec.europa.eu/cas (proxied, internal and external access)
- From other European Institutions using the TESTA II network, please use:
  - https://webgate.ec.testa.eu/cas (proxied, only via the TESTA II network)
  - Or https://ecas.ec.testa.eu/cas (proxied, only via the TESTA II network)

3.2. About the version numbers

This document mentions version numbers for the ECAS Client files.

Please adapt those version numbers to the latest recommended versions.

At the time of writing (11/05/2017), the ECAS Client version is 4.16.1**.**

3.3. Accessing the WebLogic server administration console

Important note:

After you install the ECAS Client, all security constraints will be using ECAS for authentication, including WebLogic Server Administration console.

If you access it by /console, ECAS authentication will occur and you will be denied access to the console unless your Commission user ID is member of the "Administrators" group in WebLogic server.

However, WebLogic console will still be accessible if you access it by its direct login page: /console/login/LoginForm.jsp, e.g. http://localhost:7001/console/login/LoginForm.jsp

See [ECAS-ADV], chapter "Accessing the WebLogic console" for more information about accessing the console without ECAS authentication.

4. BEFORE YOU START

This chapter provides a few (strongly advised) recommendations and a set of mandatory settings intended to fine-tune your WebLogic server.

Unless specified otherwise, each section defines one or more JVM options that must be added to the server start-up script(s)[3].

For example:

```
set EXTRA_OPTIONS=...add section-specific options here...

set JAVA_OPTIONS=%JAVA_OPTIONS% %EXTRA_OPTIONS%
```

Note that on UNIX systems `%JAVA_OPTIONS%` is to be understood as `$JAVA_OPTIONS`.

Please refer to the Oracle documentation on how to configure startup parameters for a WebLogic Server instance on your target platform.

4.1. WebLogic server version requirements

The current versions of the ECAS client were tested on WebLogic Server 10.3.0, 10.3.1, 10.3.2, 10.3.4, 10.3.5, 10.3.6, 12.1.2, 12.1.3 and 12.2.1 with either Java 5, 6, 7 or 8.

It is recommended to use WebLogic Server 10.3.6 or greater.

4.2. Network timeout recommendations

4.2.1. DNS cache time-to-live

Specify the following properties (in seconds) to avoid a "cache forever" policy for DNS name lookups:

```
-Dsun.net.inetaddr.ttl=60
-Dsun.net.inetaddr.negative.ttl=5
```

4.2.2. Connection timeout

Specify the following properties (in milliseconds) to avoid threads from hanging forever when establishing a connection to or reading information from a host:

```
-Dsun.net.client.defaultConnectTimeout=60000
-Dsun.net.client.defaultReadTimeout=60000
```

References: Oracle's Networking Properties for Java 6, Java 7 and Java 8.

[3] One way to do it is by editing `setDomainEnv.cmd` (or `.sh`) and adding properties into `JAVA_OPTIONS`.

4.3. Security requirements

4.3.1. *Enable strong cryptography*

Install the Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files.

1) Download the ZIP file corresponding to the version of the Java platform used by your WebLogic server.

Java version	Downloads
JDK/JRE 6	Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files 6
JDK/JRE 7	Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files 7
JDK/JRE 8	Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files 8

2) Extract and install the JAR files.

Extract the JAR files from the downloaded ZIP file.

Copy them into the `%JAVA_HOME%\jre\lib\security` (or `$JAVA_HOME/jre/lib/security`) directory and overwrite the existing files.

For example: extract the JAR files into `D:\Oracle\Middleware\jdk1.8.0_72\jre\lib\security`

4.3.2. *Disable SSLv3*

To avoid SSLv3 vulnerabilities such as the SSL POODLE attack, specify the following options:

```
-DUseSunHttpHandler=true
-Dhttps.protocols="TLSv1"
-Dweblogic.security.SSL.enableJSSE=true
-Dweblogic.ssl.JSSEEnabled=true
-Dweblogic.security.SSL.protocolVersion=TLS1
```

where:

- `-DUseSunHttpHandler=true` instructs WebLogic to use `java.net` for HTTPS Client connections (i.e. use the JVM, not `weblogic.net` packages). This option is not mandatory.
- `-Dhttps.protocols="TLSv1"` instructs the JVM to use for HTTPS Client connections at least TLSv1.0 i.e. TLS version 1.0 or better (including TLSv1.1 for Java 6 and TLSv1.2 for Java 7).
  N.B. If your server is running on Java 8, you must specify the TLS versions explicitly and use the new `jdk.tls.client.protocols` property:

  ```
  -Dhttps.protocols="TLSv1,TLSv1.1,TLSv1.2"
  -Djdk.tls.client.protocols="TLSv1,TLSv1.1,TLSv1.2"
  ```
- `-Dweblogic.security.SSL.enableJSSE=true` and `-Dweblogic.ssl.JSSEEnabled=true` instruct WebLogic Server to use JSSE for the TLS layer implementation instead of the obsolete Certicom implementation (i.e. using the JVM implementation instead of the deprecated native implementation from Certicom)[4].
- `-Dweblogic.security.SSL.protocolVersion=TLS1` instructs WebLogic Server to use for Server connections at least TLSv1.0 (including TLSv1.1 for Java 6 and TLSv1.2 for Java 7).

[4] These instructions are mandatory for WLS 10.3 versions but are not needed for WebLogic Server 12 where Certicom is already removed.

References:
Oracle's Instructions to disable SSL v3.0 in Oracle JDK and JRE
Oracle’s Configuring SSL for WebLogic Server 10.3.6, 11g R1, 12.1.2, 12.1.3

4.3.3. *Disable SSLv3 for WebLogic-generated Web Service clients*

For Web service clients running on your WebLogic server, in addition to the properties mentioned in the previous section, you need to specify:

```
-Dweblogic.wsee.client.ssl.usejdk=true
```

If you are running WebLogic 10.3.0 you also need the following:

```
-Djava.protocol.handler.pkgs=com.sun.net.ssl.internal.www.protocol
-Dssl.SocketFactory.provider=com.sun.net.ssl.internal.SSLSocketFactoryImpl
```

where:

- `-Djava.protocol.handler.pkgs=com.sun.net.ssl.internal.www.protocol` instructs the JVM to use Sun's reference implementation of HTTPS protocol.
- `-Dssl.SocketFactory.provider=com.sun.net.ssl.internal.SSLSocketFactoryImpl` instructs the JVM to use the SunJSSE provider.

4.3.1. *Enforce TLS secure renegotiation*

To avoid the TLS renegotiation vulnerability, upgrade the Java platform used by your WebLogic server.

Java version	Release supporting secure TLS renegotiation
JDK/JRE 6	Update 22
JDK/JRE 5.0	Update 26
SDK/JRE 1.4.2	Update 28

Reference: Oracle's Transport Layer Security (TLS) Renegotiation Issue Readme

5. Installing the EcasIdentityAsserterV2

You will have to install the ECAS Identity Asserter V2 only once for your WebLogic domain.

5.1. Download ECAS Client release

Download from the [ECAS-FORGE] all the files from the latest release.

5.2. Copy the files into your WebLogic Server instance

1) Copy the ECAS Client JAR (for example *ecas-weblogic-10.3-authprovider-4.16.1.jar*) into your domain library, e.g. `%BEA_HOME%/%DOMAIN%/lib`[5]
2) If not already present, create a `classes` directory under: `%BEA_HOME%/%WL_SERVER%/server/lib/consoleapp/webbapp/WEB-INF`[6]
3) Copy *security.properties* into: `%BEA_HOME%/%WL_SERVER%/server/lib/consoleapp/webbapp/WEB-INF/classes`

Note: if you want to use Remote EJBs and ECAS, you have to install the ECAS JAR at the level of your server classpath rather than in your domain library folder. In such a case, you can modify your WebLogic startup script CLASSPATH and add the ECAS client JAR in it.

5.3. Activate the EcasIdentityAsserter V2

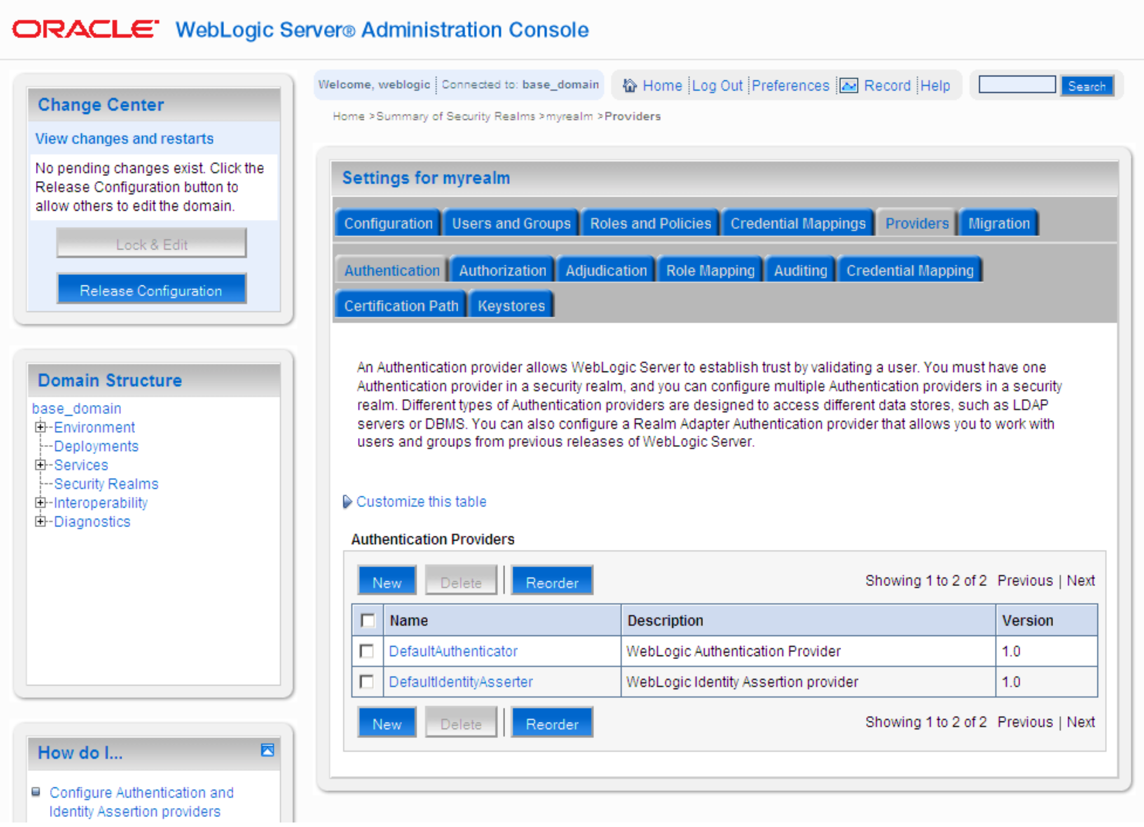
1) Restart WebLogic Server
2) Open WebLogic Server console in a browser[7]
3) Navigate to “your domain” > “Security Realms” > “myrealm” > “Providers”
4) Click on “Authentication”

[5] Where `%BEA_HOME%` is the directory where you installed WebLogic Server and `%DOMAIN%` is your WebLogic domain. For instance: `D:\bea\user_projects\domains\mydomain\lib`. On UNIX, `%BEA_HOME%` is to be understood as `$BEA_HOME`.

[6] Where `%BEA_HOME%` is the directory where you installed WebLogic Server and `%WL_SERVER%` is your WebLogic server name. For instance: `D:\bea\wlserver_10.3\server\lib\consoleapp\webapp\WEB-INF\classes`. On UNIX, `%BEA_HOME%` is to be understood as `$BEA_HOME`.

[7] By default, the console is available at http://YourServerName:7001/console.

5) Click on “Lock & Edit” and press “New”



No pending changes exist. Click the Release Configuration button to allow others to edit the domain.

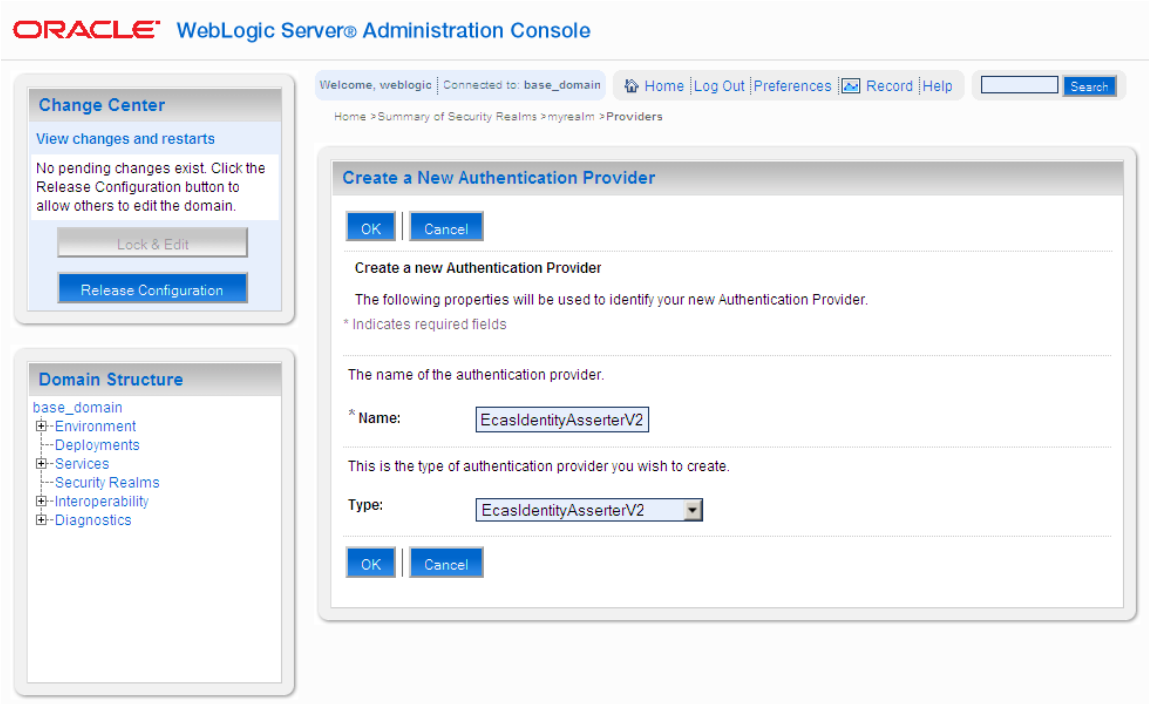
An Authentication provider allows WebLogic Server to establish trust by validating a user. You must have one Authentication provider in a security realm, and you can configure multiple Authentication providers in a security realm. Different types of Authentication providers are designed to access different data stores, such as LDAP servers or DBMS. You can also configure a Realm Adapter Authentication provider that allows you to work with users and groups from previous releases of WebLogic Server.

Authentication Providers

□	Name	Description	Version
□	DefaultAuthenticator	WebLogic Authentication Provider	1.0
□	DefaultIdentityAsserter	WebLogic Identity Assertion provider	1.0

Figure 1 - Authentication Providers

6) Fill in the name “EcasIdentityAsserterV2” and, for the type, select the “EcasIdentityAsserterV2” option from the dropdown list



No pending changes exist. Click the Release Configuration button to allow others to edit the domain.

Create a New Authentication Provider

Create a new Authentication Provider

The following properties will be used to identify your new Authentication Provider.

* Indicates required fields

The name of the authentication provider.

* Name: EcasIdentityAsserterV2

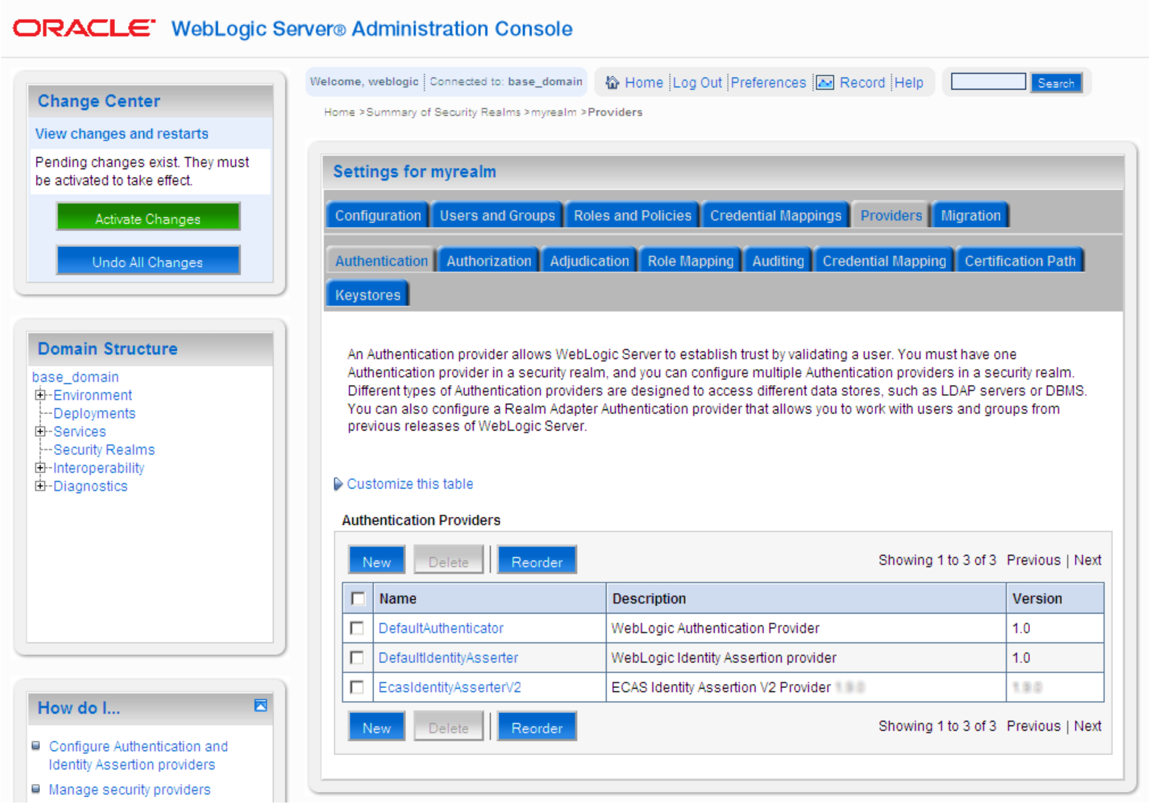
This is the type of authentication provider you wish to create.

Type: EcasIdentityAsserterV2

Figure 2 - Create a New Authentication Provider

Note: if you do not see the “EcasIdentityAsserterV2” in the list, it means that you forgot to copy the ECAS Identity Asserter V2 JAR (*ecas-weblogic-10.3-authprovider-4.16.1.jar*) into your domain library (see 5.2) and to restart WebLogic Server afterwards.

7) Click “OK”, the newly created "EcasIdentityAsserterV2" should be visible



Pending changes exist. They must be activated to take effect.

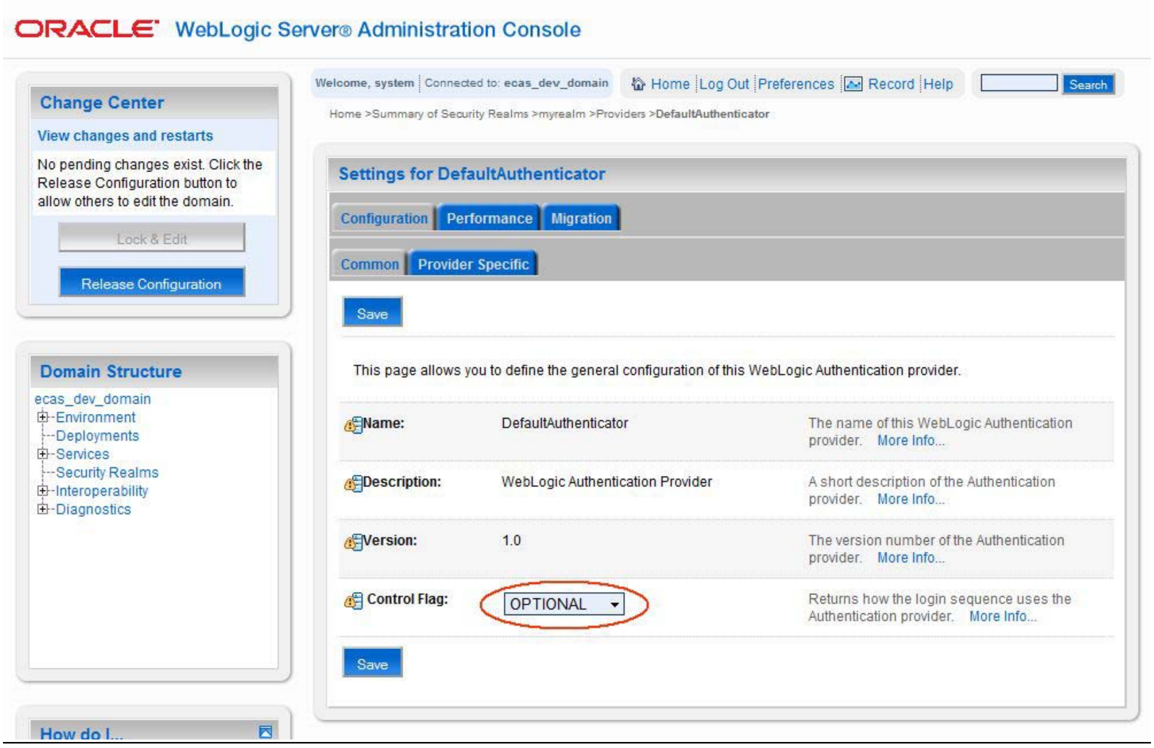
An Authentication provider allows WebLogic Server to establish trust by validating a user. You must have one Authentication provider in a security realm, and you can configure multiple Authentication providers in a security realm. Different types of Authentication providers are designed to access different data stores, such as LDAP servers or DBMS. You can also configure a Realm Adapter Authentication provider that allows you to work with users and groups from previous releases of WebLogic Server.

Authentication Providers

□	Name	Description	Version
□	DefaultAuthenticator	WebLogic Authentication Provider	1.0
□	DefaultIdentityAsserter	WebLogic Identity Assertion provider	1.0
□	EcasIdentityAsserterV2	ECAS Identity Assertion V2 Provider [redacted]	[redacted]

Figure 3 – Newly created Authentication Provider

8) Browse all the other Authentication Providers, verify that their “Control Flag” is set to “OPTIONAL” (not “REQUIRED”)[8] and change it if it's not the case.
For example, “Security” > “Realms” > “myrealm” > “Providers” > “Authentication Providers” > “DefaultAuthenticator”: “Control Flag” must be set to “OPTIONAL”


ORACLE WebLogic Server® Administration Console
Welcome, system | Connected to: ecas_dev_domain
Home | Log Out | Preferences | Record | Help
Search
Home >Summary of Security Realms >myrealm >Providers >DefaultAuthenticator
Change Center
View changes and restarts
Lock & Edit
Release Configuration
Domain Structure
ecas_dev_domain
Environment
Deployments
Services
Security Realms
Interoperability
Diagnostics
How do I...
Configuration
Performance
Migration
Common
Provider Specific
Save

No pending changes exist. Click the Release Configuration button to allow others to edit the domain.

Settings for DefaultAuthenticator

This page allows you to define the general configuration of this WebLogic Authentication provider.

Name:	DefaultAuthenticator	The name of this WebLogic Authentication provider. More Info...
Description:	WebLogic Authentication Provider	A short description of the Authentication provider. More Info...
Version:	1.0	The version number of the Authentication provider. More Info...
Control Flag:	OPTIONAL	Returns how the login sequence uses the Authentication provider. More Info...

Figure 4 - Set the DefaultAuthenticator to OPTIONAL

Important note:

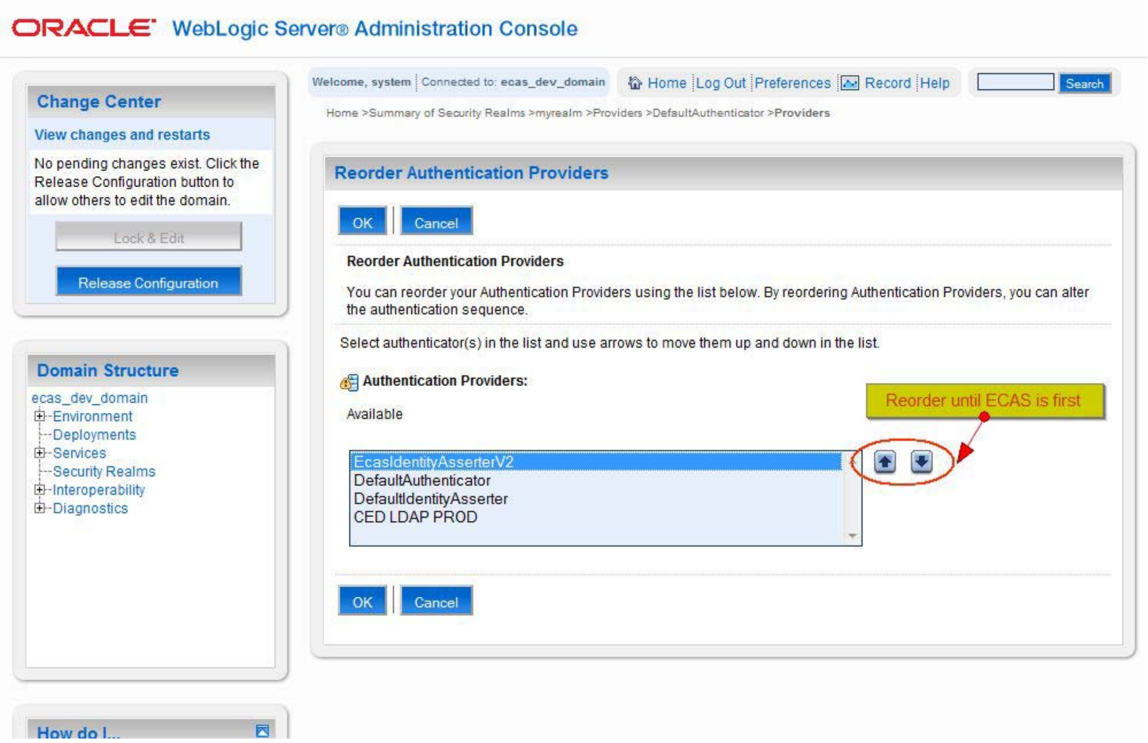
Do NOT let the “Control Flag” of the DefaultAuthenticator on “REQUIRED” or else you will not be able to validate ECAS tickets as the Default Authentication Provider has no knowledge of ECAS whatsoever.

9) Click on “Save”

[8] This is especially true for the DefaultAuthenticator.

10) Back on the Authentication Providers screen, click on “Reorder”

11) Re-order the authentication providers so that "EcasIdentityAsserterV2" is at the top of the list



ORACLE WebLogic Server® Administration Console

Welcome, system | Connected to: ecas_dev_domain

Home > Summary of Security Realms > myrealm > Providers > DefaultAuthenticator > Providers

Change Center

View changes and restarts

No pending changes exist. Click the Release Configuration button to allow others to edit the domain.

Domain Structure

ecas_dev_domain

Reorder Authentication Providers

Reorder Authentication Providers

You can reorder your Authentication Providers using the list below. By reordering Authentication Providers, you can alter the authentication sequence.

Select authenticator(s) in the list and use arrows to move them up and down in the list.

Authentication Providers:

Available

EcasIdentityAsserterV2
DefaultAuthenticator
DefaultIdentityAsserter
CED LDAP PROD

Reorder until ECAS is first

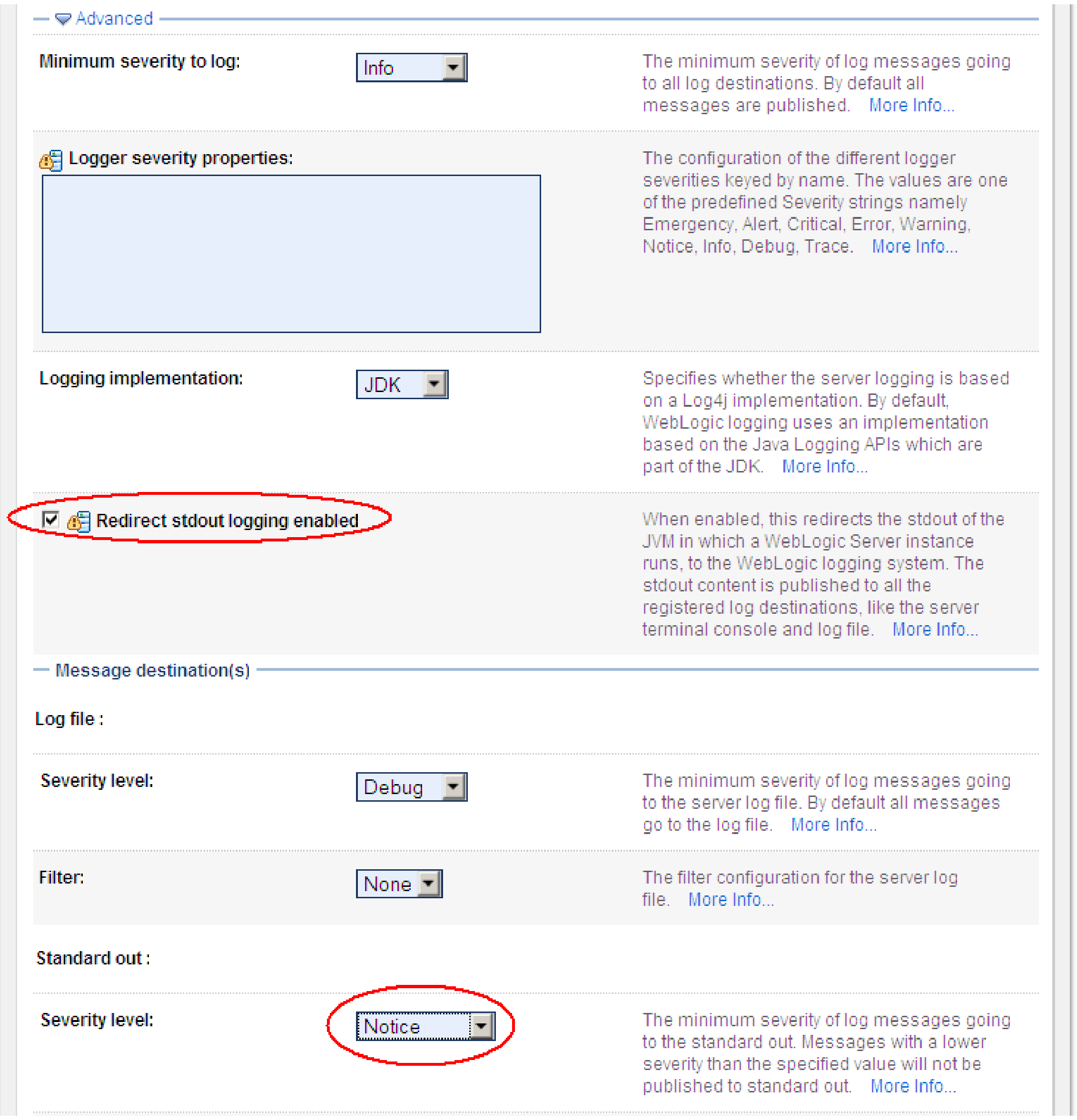
Figure 5 - Reorder Authentication Providers

12) Activate the changes (a server restart may be required).

13) Turn on WebLogic logging (optional)

If you **want** to see the logging of the Ecas Identity Asserter V2 on the standard output[9], you need to turn on WebLogic logging using WebLogic Administration Console.

In your domain, navigate to “Environment” > “Servers” > “your server” > “Logging” > “General” > “Advanced” and check the box “Redirect sdtout logging enabled” and set the “Severity level” either on "Debug", "**Info**" or "Notice".



Advanced

Minimum severity to log: Info

The minimum severity of log messages going to all log destinations. By default all messages are published. More Info...

Logger severity properties:

The configuration of the different logger severities keyed by name. The values are one of the predefined Severity strings namely Emergency, Alert, Critical, Error, Warning, Notice, Info, Debug, Trace. More Info...

Logging implementation: JDK

Specifies whether the server logging is based on a Log4j implementation. By default, WebLogic logging uses an implementation based on the Java Logging APIs which are part of the JDK. More Info...

☑ Redirect stdout logging enabled

When enabled, this redirects the stdout of the JVM in which a WebLogic Server instance runs, to the WebLogic logging system. The stdout content is published to all the registered log destinations, like the server terminal console and log file. More Info...

Message destination(s)

Log file :

Severity level: Debug

The minimum severity of log messages going to the server log file. By default all messages go to the log file. More Info...

Filter: None

The filter configuration for the server log file. More Info...

Standard out :

Severity level: Notice

The minimum severity of log messages going to the standard out. Messages with a lower severity than the specified value will not be published to standard out. More Info...

Figure 6 – Configure WebLogic Logging

14) Click on "Save"

15) Restart the WebLogic Server

[9] The standard output (`stdout`) is the command/shell prompt from which you launched WebLogic Server. On Unix, it may be a `nohup.out` file, depending on the way your server was launched.

6. Protect your web application

For each application you want to be protected by ECAS, you need to configure the deployment descriptors (*web.xml* and *weblogic.xml*) and configure the ECAS Client.

For this section you should inspect the sample application *ecas-demo*, install it and try it. Use the various files provided in this section as examples and adapt them to match your own needs.

The important files involved in the ECAS Client mechanism are:

1. web.xml
2. weblogic.xml
3. an error page such as: error.jsp
4. a logout page such as: logout.jsp
5. the ECAS Client configuration file (if any)
6. the logging configuration

6.1. Configure the ECAS Client

The ECAS Client for WebLogic can be configured in different places.

For a basic configuration of the client, we will use a configuration file[10] that uses a conventional name. By convention, the name of the configuration file is

“`ecas-config-`” + your context-path with slashes replaced by dots + “`.properties`” (or “`.xml`”)

- Where by context-path, we mean the result of `HttpServletRequest#getContextPath()` for your Web application.
- And the slashes (‘/’) in your context-path are replaced by dots (‘.’).

For example, if your application uses the context-path “*/oib/f/budg-app*”, you would have to use the conventional name “*ecas-config-**oib.f.budg-app**.xml*” for your configuration file.

Since you cannot deploy two applications on the same context-path in a domain, your configuration file should be unique per domain.

So create the corresponding file for your application. In the examples that follow, we will call it *ecas-config-**mycontextpath**.xml*.

You may either put this file inside your WAR in `WEB-INF/classes` or directly in the classpath of your domain or server.

[10] The various configuration methods are described in [ECAS-ADV]. In this section, we intentionally limit ourselves to a minimal recommended configuration that works. Alternative configurations methods, such as using a plain-text properties file or embedding an `ecas-config.xml` file in the web archive without using an intermediate file are discussed with their respective benefits and drawbacks in [ECAS-ADV].

Below is an example of a minimal `ecas-config-mycontextpath.xml`. Refer to the next section for more detailed samples of configuration files.

```
<client-config xmlns="https://ecas.ec.europa.eu/cas/schemas/client-config/ecas/" />
```

Figure 7 – Simple ecas-config-mycontextpath.xml file

Where "*mycontextpath*" stands for your application context-path (normalized as specified earlier).

Deploying this file outside the WAR allows you to deploy the same WAR file throughout different environments (dev, test, prod) and only adapt the configuration files with the appropriate information (e.g. dev, test and prod environments just differ by a few properties). This configuration file must be on the domain or server classpath.

6.2. Define a security constraint

Accordingly to the Servlet API, let us define a protected area in the `web.xml` deployment descriptor of the application. All you need to have to trigger ECAS authentication is a `security-constraint` tag:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<web-app xmlns="http://java.sun.com/xml/ns/j2ee"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee
                            http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd"
        version="2.4">
        <description>ECAS v2 minimal</description>
        <display-name>ecasV2-minimal</display-name>

        <!-- This is an example security constraint for a resource that
        requires only authentication but not authorization: -->
        <security-constraint>
                <web-resource-collection>
                        <web-resource-name>ecasV2-minimal</web-resource-name>
                        <description>
                        This is the protected area of the application.
                        </description>
                        <url-pattern>/protected/*</url-pattern>
                </web-resource-collection>
                <auth-constraint>
                        <description>
                        Requires users to be authenticated but
                        does not require them to be authorized.
                        </description>
                        <role-name>*</role-name>
                </auth-constraint>
                <user-data-constraint>
                        <description>
                        Encryption is not required for this area.
                        </description>
                        <transport-guarantee>NONE</transport-guarantee>
                </user-data-constraint>
        </security-constraint>
</web-app>
```

Figure 8 – Example security-constraint tag in web.xml

This sample means that all resources (pages or controllers) with a URL path starting with “`/protected/`” require authentication. In our case, authentication will be done using the ECAS Identity Assertion V2 Provider. Of course, you should adapt the `url-pattern` tag according to your application paths.

6.3. Configure weblogic.xml

Eventually, configure your *weblogic.xml* using the following file as a template:

```
<weblogic-web-app xmlns="http://www.bea.com/ns/weblogic/90"
       xmlns:j2ee="http://java.sun.com/xml/ns/j2ee"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xsi:schemaLocation="http://www.bea.com/ns/weblogic/90
                    http://www.bea.com/ns/weblogic/90/weblogic-web-app.xsd">

       <session-descriptor>
             <cookie-path>/mycontextpath</cookie-path>
       </session-descriptor>

       <!-- We want to allow some resources to be accessed by
             authenticated users who do not possess any role -->
       <container-descriptor>
             <allow-all-roles>true</allow-all-roles>
       </container-descriptor>

       <context-root>/mycontextpath</context-root>

</weblogic-web-app>
```

Figure 9 – weblogic.xml template

The “`cookie-path`” tag must match the context-path of your Web application. Replace the “`/mycontextpath`” value with the correct deployment path of your application.

Be careful: a wrong cookie-path will result in the impossibility to access the protected parts of your application.

This deployment descriptor uses the “`allow-all-roles`” tag to instruct WebLogic Server to consider the special role named “`*`” in `web.xml` as users being only authenticated (without concerns about authorization). Alternatively, you can use a “`security-role-assignment`” tag to assign a role to groups requested from ECAS (i.e. one or more CUD[11] groups). Please see [ECAS-ADV] for more information.

6.4. Logging

You will need to copy our log4j JAR in your WEB-INF/lib directory:

- *log4j-1.2.15.jar*[12]

And you need to provide a valid log4j configuration file in your WEB-INF/classes directory:

- *log4j.xml*

[11] Central User Database: CUD groups are also known as LDAP groups and can be retrieved from the CED LDAP directory at ldap.cc.cec.eu.int on port 389.

[12] This version of the ECAS client has been successfully tested with log4j 1.2.8 up to log4j 1.2.14. If you intend to use version 1.2.15, please use the patched binary we provide because there is a defect in the official binary due to a jar manifest issue. See https://issues.apache.org/bugzilla/show_bug.cgi?id=44370 for details.

6.5. Test your configuration

You can test your configuration by installing and trying the sample application *ecas-demo.ear*.

Note: the ECAS demo application is not cross-platform. Access the [ECAS-FORGE] and make sure you download the distribution for WebLogic.

Once you have downloaded the sample application, follow the instructions in [ECAS-ADV] to configure the `web.xml` and `weblogic.xml` to match your settings. You also need to add a valid `log4j.properties` or `log4j.xml` file in `WEB-INF/classes`. You can do that by changing the path of the log file in the provided `log4j.xml`.

When all the files are modified and saved, just copy the ecas-demo folder to `your_domain/autodeploy`[13] folder and test it by hitting e.g. http://YourServerName:7001/ecas-demo /protected/.

6.6. Allow access to non-Commission users

By default the ECAS Client allows only internal Commission users to access an application.

If you would like non-Commission users such as self-registered, sponsored or interinstitutional users to access your application, please see the chapter on configuration property "*assuranceLevel*" in [ECAS-ADV].

[13] E.g. `%BEA_HOME%/user_projects/domains/mydomain/autodeploy`