

<!--

DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS HEADER.

Copyright 2010 Sun Microsystems, Inc. All rights reserved.

The contents of this file are subject to the terms of either the GNU General Public License Version 2 only ("GPL") or the Common Development and Distribution License("CDDL") (collectively, the "License"). You may not use this file except in compliance with the License. You can obtain

a copy of the License at <https://glassfish.dev.java.net/public/CDDL+GPL.html>

or [glassfish/bootstrap/legal/LICENSE.txt](https://glassfish.dev.java.net/public/legal/LICENSE.txt). See the License for the specific

language governing permissions and limitations under the License.

When distributing the software, include this License Header Notice in each file and include the License file at [glassfish/bootstrap/legal/LICENSE.txt](https://glassfish.dev.java.net/public/legal/LICENSE.txt).

Sun designates this particular file as subject to the "Classpath" exception

as provided by Sun in the GPL Version 2 section of the License file that accompanied this code. If applicable, add the following below the License Header, with the fields enclosed by brackets [] replaced by your own identifying information: "Portions Copyrighted [year] [name of copyright owner]"

Contributor(s):

If you wish your version of this file to be governed by only the CDDL or only the GPL Version 2, indicate your decision by adding "[Contributor] elects to include this software in this distribution under the [CDDL or GPL

Version 2] license." If you don't indicate a single choice of license, a recipient has the option to distribute your version of this file under either the CDDL, the GPL Version 2 or to extend the choice of license to its licensees as provided above. However, if you add GPL Version 2 code and therefore, elected the GPL Version 2 license, then the option applies only if the new code is made subject to such option by the copyright holder.

-->

```
<config name="default-config" dynamic-reconfiguration-enabled="true" >
  <http-service>
    <access-log/>
    <virtual-server id="server" network-listeners="http-listener-1,
http-listener-2" >
      </virtual-server>
    </http-service>
  </iio-service>
```

```

    <orb use-thread-pool-ids="thread-pool-1" />
    <iiop-listener id="orb-listener-1" address="0.0.0.0" port="$
{IIOP_LISTENER_PORT}" />
    <iiop-listener id="SSL" address="0.0.0.0" port="$
{IIOP_SSL_LISTENER_PORT}" security-enabled="true" >
        <ssl cert-nickname="s1as" />
    </iiop-listener>
    <iiop-listener id="SSL_MUTUALAUTH" address="0.0.0.0" port="$
{IIOP_SSL_MUTUALAUTH_PORT}" security-enabled="true">
        <ssl cert-nickname="s1as" client-auth-enabled="true" />
    </iiop-listener>
</iiop-service>
    <admin-service system-jmx-connector-name="system" type="server">
    <!-- JSR 160 "system-jmx-connector" -->
    <jmx-connector address="0.0.0.0" auth-realm-name="admin-realm"
name="system" port="{JMX_SYSTEM_CONNECTOR_PORT}" protocol="rmi_jrmp">
    </jmx-connector>
    <!-- JSR 160 "system-jmx-connector" -->
    <das-config dynamic-reload-enabled="true" autodeploy-enabled="true"
autodeploy-dir="{com.sun.aas.instanceRoot}/autodeploy" />
    </admin-service>
</web-container>
    <session-config>
        <session-manager>
            <manager-properties/>
            <store-properties />
        </session-manager>
        <session-properties />
    </session-config>
</web-container>
    <ejb-container session-store="{com.sun.aas.instanceRoot}/session-store">
    <ejb-timer-service></ejb-timer-service>
    </ejb-container>
</mdb-container />
    <jms-service type="LOCAL" default-jms-host="default_JMS_host"
addresslist-behavior="priority" >
    <jms-host name="default_JMS_host" host="%%%HOST_NAME%%%" port="$
{JMS_PROVIDER_PORT}" admin-user-name="%%%JMS_PROVIDER_USERID%%%" admin-
password="%%%JMS_PROVIDER_PASSWORD%%%" />
    </jms-service>
    <log-service file="{com.sun.aas.instanceRoot}/logs/server.log" use-
system-logging="false" log-to-console="false" log-rotation-limit-in-
bytes="2000000">
        <module-log-levels />
    </log-service>
    <security-service>
    <!-- All administrative tasks within any server instance use this
admin-realm by default. -->
        <auth-realm name="admin-realm"
classname="com.sun.enterprise.security.auth.realm.file.FileRealm">

```

```

        <property name="file" value="\${com.sun.aas.instanceRoot}/config/
admin-keyfile" />
        <property name="jaas-context" value="fileRealm" />
    </auth-realm>
    <!-- All administrative tasks within any server instance use this
admin-realm by default. -->
    <auth-realm name="file"
classname="com.sun.enterprise.security.auth.realm.file.FileRealm">
        <property name="file" value="\${com.sun.aas.instanceRoot}/config/
keyfile" />
        <property name="jaas-context" value="fileRealm" />
    </auth-realm>
    <auth-realm name="certificate"
classname="com.sun.enterprise.security.auth.realm.certificate.CertificateRe
alm">
    </auth-realm>
    <jacc-provider name="default" policy-
provider="com.sun.enterprise.security.provider.PolicyWrapper" policy-
configuration-factory-
provider="com.sun.enterprise.security.provider.PolicyConfigurationFactoryIm
pl">
        <property name="repository" value="\${com.sun.aas.instanceRoot}/
generated/policy" />
    </jacc-provider>
    <jacc-provider policy-
provider="com.sun.enterprise.security.jacc.provider.SimplePolicyProvider"
name="simple" policy-configuration-factory-
provider="com.sun.enterprise.security.jacc.provider.SimplePolicyConfigurati
onFactory" />
        <audit-module name="default"
classname="com.sun.enterprise.security.Audit">
            <property name="auditOn" value="false" />
        </audit-module>
        <message-security-config auth-layer="SOAP">
            <!-- turned off by default -->
        </provider-config class-
name="com.sun.xml.wss.provider.ClientSecurityAuthModule" provider-
id="XWS_ClientProvider" provider-type="client">
            <request-policy auth-source="content" />
            <response-policy auth-source="content" />
            <property name="encryption.key.alias" value="s1as" />
            <property name="signature.key.alias" value="s1as" />
            <property name="dynamic.username.password" value="false" />
            <property name="debug" value="false" />
        </provider-config>
        <provider-config class-
name="com.sun.xml.wss.provider.ClientSecurityAuthModule" provider-
id="ClientProvider" provider-type="client">
            <request-policy auth-source="content" />
            <response-policy auth-source="content" />

```

```

        <property name="encryption.key.alias" value="s1as" />
        <property name="signature.key.alias" value="s1as" />
        <property name="dynamic.username.password" value="false" />
        <property name="debug" value="false" />
        <property name="security.config" value="$
{com.sun.aas.instanceRoot}/config/wss-server-config-1.0.xml" />
    </provider-config>
    <provider-config class-
name="com.sun.xml.wss.provider.ServerSecurityAuthModule" provider-
id="XWS_ServerProvider" provider-type="server">
        <request-policy auth-source="content" />
        <response-policy auth-source="content" />
        <property name="encryption.key.alias" value="s1as" />
        <property name="signature.key.alias" value="s1as" />
        <property name="debug" value="false" />
    </provider-config>
    <provider-config class-
name="com.sun.xml.wss.provider.ServerSecurityAuthModule" provider-
id="ServerProvider" provider-type="server">
        <request-policy auth-source="content" />
        <response-policy auth-source="content" />
        <property name="encryption.key.alias" value="s1as" />
        <property name="signature.key.alias" value="s1as" />
        <property name="debug" value="false" />
        <property name="security.config" value="$
{com.sun.aas.instanceRoot}/config/wss-server-config-1.0.xml" />
    </provider-config>
</message-security-config>
</security-service>
    <transaction-service automatic-recovery="true" tx-log-dir="$
{com.sun.aas.instanceRoot}/logs" />
    <monitoring-service>
    <module-monitoring-levels />
    </monitoring-service>
    <diagnostic-service />
    <java-config debug-options="-Xdebug -
Xrunjdpw:transport=dt_socket,server=y,suspend=n,address=9009" system-
classpath="" classpath-suffix="">
        <jvm-options>-XX:MaxPermSize=192m</jvm-options>
        <jvm-options>-client</jvm-options>
        <jvm-options>-XX:+UnlockDiagnosticVMOptions</jvm-options>
        <jvm-options>-XX:+LogVMOutput</jvm-options>
        <jvm-options>-XX:LogFile=${com.sun.aas.instanceRoot}/logs/jvm.log</
jvm-options>
        <jvm-options>-Djava.endorsed.dirs=${com.sun.aas.installRoot}/
modules/endorsed${path.separator}${com.sun.aas.installRoot}/lib/endorsed</
jvm-options>
        <jvm-options>-Djava.security.policy=${com.sun.aas.instanceRoot}/
config/server.policy</jvm-options>
        <jvm-options>-Djava.security.auth.login.config=$

```

```

{com.sun.aas.instanceRoot}/config/login.conf</jvm-options>
  <jvm-options>-
Dcom.sun.enterprise.security.httpsOutboundKeyAlias=s1as</jvm-options>
  <jvm-options>-Djavax.net.ssl.keyStore=${com.sun.aas.instanceRoot}/
config/keystore.jks</jvm-options>
  <jvm-options>-Djavax.net.ssl.trustStore=$
{com.sun.aas.instanceRoot}/config/cacerts.jks</jvm-options>
  <jvm-options>-Djava.ext.dirs=${com.sun.aas.javaRoot}/lib/ext$
{path.separator}${com.sun.aas.javaRoot}/jre/lib/ext${path.separator}$
{com.sun.aas.instanceRoot}/lib/ext</jvm-options>
  <jvm-options>-Djdbc.drivers=org.apache.derby.jdbc.ClientDriver</
jvm-options>
  <jvm-options>-DANTLR_USE_DIRECT_CLASS_LOADING=true</jvm-options>
  <jvm-options>-
Dcom.sun.enterprise.config.config_environment_factory_class=com.sun.enterpr
ise.config.serverbeans.AppserverConfigEnvironmentFactory</jvm-options>
  <jvm-options>-Dosgi.shell.telnet.port=6666</jvm-options>
  <jvm-options>-Dosgi.shell.telnet.maxconn=1</jvm-options>
  <jvm-options>-Dosgi.shell.telnet.ip=127.0.0.1</jvm-options>
  <jvm-options>-Dfelix.fileinstall.dir=${com.sun.aas.installRoot}/
modules/autostart/</jvm-options>
  <jvm-options>-Dfelix.fileinstall.poll=5000</jvm-options>
  <jvm-options>-Dfelix.fileinstall.debug=1</jvm-options>
  <jvm-options>-Dfelix.fileinstall.bundles.new.start=true</jvm-
options>
  <jvm-options>-
Dorg.glassfish.web.rfc2109_cookie_names_enforced=false</jvm-options>
  <jvm-options>-XX:NewRatio=2</jvm-options>
  <jvm-options>-Xmx512m</jvm-options>
</java-config>
  <availability-service availability-enabled="true">
    <web-container-availability availability-enabled="true" http-session-
store-pool-name="jdbc/hastore" persistence-frequency="web-method"
persistence-scope="session" persistence-store-health-check-enabled="false"
persistence-type="ha" sso-failover-enabled="false"/>
    <ejb-container-availability availability-enabled="true" sfsb-ha-
persistence-type="ha" sfsb-persistence-type="file" sfsb-store-pool-
name="jdbc/hastore"/>
    <jms-availability availability-enabled="false"/>
  </availability-service>

  <network-config>
    <protocols>
      <protocol name="http-listener-1" security-enabled="false">
        <http header-buffer-length-in-bytes="8192" max-
connections="250" timeout="30" default-virtual-server="server" server-
name="" xpowered-by="true" send-buffer-size="8192">
          <file-cache enabled="true" max-age="30" max-cache-
size="10485760" max-files-count="1024"/>
        </http>

```

```

        </protocol>
        <protocol security-enabled="true" name="http-listener-2">
            <http header-buffer-length-in-bytes="8192" max-
connections="250" timeout="30" default-virtual-server="server" server-
name="" xpowered-by="true" send-buffer-size="8192">
                <file-cache enabled="true" max-age="30" max-cache-
size="10485760" max-files-count="1024"/>
            </http>
            <ssl ssl3-enabled="false" cert-nickname="s1as" />
        </protocol>
    </protocols>
    <network-listeners>
        <network-listener address="0.0.0.0" port="{HTTP_LISTENER_PORT}"
protocol="http-listener-1" transport="tcp" name="http-listener-1" thread-
pool="http-thread-pool" enabled="true"/>
        <network-listener address="0.0.0.0" port="{
HTTP_SSL_LISTENER_PORT}" protocol="http-listener-2" transport="tcp"
name="http-listener-2" thread-pool="http-thread-pool" xpowered-by="true"
enabled="true"/>
        <thread-pool min-thread-pool-size="2" max-thread-pool-size="5"
thread-increment="1" max-queue-size="4096"/>
    </network-listeners>
    <transports>
        <transport name="tcp" acceptor-threads="1" max-connections-
count="4096"/>
    </transports>
</network-config>
<thread-pools>
    <thread-pool name="http-thread-pool" />
    <thread-pool max-thread-pool-size="200" idle-thread-timeout-in-
seconds="120" name="thread-pool-1" />
</thread-pools>
<group-management-service fd-protocol-max-tries="3" fd-protocol-timeout-in-
millis="2000" merge-protocol-max-interval-in-millis="10000" merge-protocol-
min-interval-in-millis="5000" ping-protocol-timeout-in-millis="5000" vs-
protocol-timeout-in-millis="1500"/>
    <management-rules enabled="true" />
<system-property name="HTTP_LISTENER_PORT" value="38080"/>
    <system-property name="HTTP_SSL_LISTENER_PORT" value="38181"/>
    <system-property name="JMS_PROVIDER_PORT" value="37676"/>
    <system-property name="IIOP_LISTENER_PORT" value="33700"/>
    <system-property name="IIOP_SSL_LISTENER_PORT" value="33820"/>
    <system-property name="IIOP_SSL_MUTUALAUTH_PORT" value="33920"/>
    <system-property name="JMX_SYSTEM_CONNECTOR_PORT" value="38686"/>
</config>

```