webMethods Suite Security Infrastructure Guide

Reference Guide to SIN

Version 9.5 SP1

November 2013

Security Infrastructure

This document applies to SIN Version 9.5 SP1.

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

Document ID: SINEXT-REF-95SP1-20130929

Table of Contents

Preface v
1 Formatting Conventions 1
    Monospace Font 2
    Italics 2
    Bold 2
    Double Quotation Marks 3
2 Samples 5
    JAAS Configuration 6
    log4j Configuration File 6
    SSXLoginModule Configuration Template 7
3 Integrated Authentication Framework Samples 19
    IAF Attribute File for Active Directory (ADSI) Authentication 20
    IAF Attribute File for Internal User Repository Authentication 24
    IAF Attribute File for LDAP User Repository Authentication 28
    IAF Attribute File for Active Directory Using LDAP Interface Authentication 32
    IAF Attribute File for Operating System Authentication 36
4 Troubleshooting 41
    Troubleshooting Integrated Authentication Framework 42
    Troubleshooting sagssxauthd2 43
    Turning on SIN Logging 43
    Using a Specific log4j for Logging Information 44
    Verifying the JAAS Configuration in Tomcat 46
    Running the Test Servlet 46
    If All Other Things Fail... 47
5 Frequently Asked Questions 49
    What is SSX RMI Service? 50
    What Should I Know About the Modules' Configuration 52
    How Do I Use HTTPS with the XmlServerLoginModule 52
6 Deprecated Login Modules 55
    CentraSiteServerLoginModule 56

Preface

This document contains lists of code samples, troubleshooting tips, and FAQ. The information addresses programmers who are developing security modules for their applications.

The information is organized under the following headings:

Samples

Integrated Authentication Framework Samples

Troubleshooting

Frequently Asked Questions

Deprecated Login Modules

See *Formatting Conventions* for a list of typographical conventions used in this documentation.

1 Formatting Conventions

- Monospace Font 2
- Italics 2
- Bold 2
- Double Quotation Marks 3

The following formatting conventions are used in this documentation:

- Monospace Font
- Italics
- Bold
- Double Quotation Marks

Monospace Font

- Pieces of programming `code`, also HTML or XML source
- Names of `classes`, `methods` and `properties` in OO programming, for example C++
- Names of `attributes`, `elements` and `entities` in the SGML/XML space
- Inline `error texts` returned by the system
- Names of `variables` whose values are specified by the user (for example environment variables)
- Names of `parameters` whose values are specified by the user or program, usually as `keyword=value` pair
- KEYS that can be pressed on the keyboard (uppercase, smaller font)

Italics

- *`Placeholder`* for values specified by the user or known only at runtime
- Reference to a *book*, *online manual*, *section* of book or manual; reference to a *location on internet*. If hyperlinked, the browser default will be used
- *File* and *directory* names, *path* names
- Text to be *emphasised*

Bold

- Command **buttons**; names of **menus** on the menu bar; names of **commands** opened from the menu bar
- **Titles** of windows and dialog boxes; **labels** on tabs, fields, check boxes, radio butons, lists
- **Items** selectable from lists in a GUI environment

Double Quotation Marks

- “Quotations”
- "Cited values" of parameters, attributes, element contents etc.

2 Samples

- JAAS Configuration 6
- log4j Configuration File 6
- SSXLoginModule Configuration Template 7

This chapter provides sample configuration files and code listings for the different `LoginModules` and authentication scenarios provided by SIN.

The samples are organized under the following headings:

JAAS Configuration

Following is a sample JAAS configuration:

```
/** Login Configuration for user, group, and role information **/
ApplicationContext {
   com.softwareag.security.jaas.login.module.SSXLoginModule required
    template_section=OS;
   com.softwareag.security.jaas.login.XmlServerLoginModule required
    XMLSERVER_URL="http://localhost:53305/CentraSite/CentraSite";
};
```

log4j Configuration File

Following is a sample log4j configuration file:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE log4j:configuration SYSTEM "log4j.dtd">

<log4j:configuration xmlns:log4j="http://jakarta.apache.org/log4j/">

  <appender name="Console" class="org.apache.log4j.ConsoleAppender">
    <param name="Target" value="System.out"/>
    <layout class="org.apache.log4j.PatternLayout">
     <param name="ConversionPattern" value="%d{ABSOLUTE} [%t] %-5p %c %x - %m%n"/>
    </layout>
  </appender>

  <root>
    <priority value ="INFO" />
    <appender-ref ref="Console" />
  </root>

  <!-- Infos for the security - set level to DEBUG if needed. -->
  <logger name="com.softwareag.security">
    <level value="DEBUG"/>
  </logger>
```

```
</log4j:configuration>
```

SSXLoginModule Configuration Template

Following is the template configuration file that is distributed with the SSXLoginModule.

You can overwrite all parameters in the JAAS configuration file, leaving out the prefix.

Important: Do not overwrite the authType.

```
# OS Section

# The type of the user db or service against which the authentication
# will be attempted.
# Possible values: os, ldap, adsi, iaf
OS.authType=os

# The log file name for the logging of the user DB library. The global
# 'ssx_userdb_global_errors.log' file (which located in the default temp
# directory) will be used if the log ability is turned on and the logging to
# the specified logfile is not possible.
#OS.nativeLogFile=SIN_SSX.log
OS.logCallback=true

# The valid value range is between 1 and 6.
# If 0 or not defined than there will be no logging.
#OS.nativeLogLevel=2

# The time in seconds till the user will be valid in the cache after
# a successful authentication.
OS.cacheTime=12

# The size of the authenticated user cache.
OS.cacheSize=4

# The time in seconds till the user authentication will be denied
# after the 'denyCount' is reached.
OS.denyTime=4

# The number of the unsuccessful authentication after that user
# gets into the deny cache.
OS.denyCount=3

# Always include local groups.
OS.winCheckLocalGroups=0

# Always include local groups.
```

```
OS.useLogonUseron2000=1

# Impersonate the userdb accesses.
OS.noImpersonation=0

# Default group to be automatically included for all requests
# that return any groups
# OS.defaultGroup=DefGroup

# Default domain name. Use this in case the domain parameter
# is not supplied.
# OS.defaultDomain=MyDomain

# Unix only! Explicit path of the privileged daemon process
# Needs to be specified, if the executable "sagssxauthd2"
# is not in the current working directory.
# OS.authDaemonPath=/tmp/sagssxauthd2

#If NOT the automatic domain name should be used to compose
#the canonical user id (SSXGetCanonicalUserId_A/W),
#specify this part of the ID here.
#OS.canonicalDomainName

#When authenticating on Windows and no domain is
#specified, Windows will try
# - to authenticate a local user
#and if this fails, Windows will try
# - to authenticate the user in the currently logged in domain.
#If this is automatic lookup is not desired, that is, only the
#local users shall be auhtenticated, set this variable.
#Valid Values: 0, 1
#Default: 0
#OS.winNoDefaultDomain=

################################################################
# LDAP Section

#This is a sample properties file for the case
#when authType is ldap and the user database is OpenLDAP.

#Specifies the authentication type.

#Is Required: Yes
#Valid values: {"os", "ldap", "adsi", "iaf", "saf"}
#No default value
LDAP.authType=ldap

#Specifies which server type will be used.
#Use only when authType is ldap.

#Is Required: No
#Valid values: {"ActiveDirectory", "SunOneDirectory", "OpenLdap"}
```

```
#Default value: "OpenLdap"
LDAP.serverType=OpenLDAP

#Property name that denotes a user entry.
#Use only when authType is ldap.

#Is Required: No
#Valid values: (attribute name according to LDAP conventions)
#No default value
LDAP.userIdField=cn

#Enumeration of LDAP objectclasses that the user entries use in
#the target LDAP server.
#Use only when authType is ldap.

#Is Required: No
#Valid values: (Comma separated list of objectclass names,
# according to LDAP conventions)
#Default value:
# depending on serverType:
# OpenLdap:
# "top,person"
# SunOneDirectory:
# "top,person,organizationalperson, inetorgperson"
# ActiveDirectory:
# "top,person,organizationalPerson,user"
LDAP.personObjClass=inetOrgPerson

#Enumeration of LDAP objectclasses that the group entries use in
#the target LDAP server.
#Use only when authType is ldap.

#Is Required: No
#Valid values: (Comma separated list of objectclass names,
# according to LDAP conventions)
#Default value:
#   depending on serverType:
#   OpenLdap:
#   "top,groupOfUniqueNames"
#   SunOneDirectory:
#   "top,groupofuniquenames"
#   ActiveDirectory:
#   "top,group"
LDAP.groupObjClass=groupOfUniqueNames

#Property name that denotes a group entry.
#Use only when authType is ldap.

#Is Required: No
#Valid values: (attribute name according to LDAP conventions)
#Default value: cn
LDAP.groupIdField=cn
```

```
#Property name of a user entry that points to the group that
#the user is member of.
#Use only when authType is ldap.

#Is Required: No
#Valid values: (attribute name according to LDAP conventions)
#Default value:
# depending on serverType:
# OpenLdap:
# "ou"
# SunOneDirectory:
# NULL
# ActiveDirectory:
# "memberOf"
LDAP.personGrpAttr=ou

#Property name of a group entry that points to users (members)
#Use only when authType is ldap.

#Is Required: No
#Valid values: (attribute name according to LDAP conventions)
#Default value:
# depending on serverType:
# OpenLdap:
# "uniqueMember"
# SunOneDirectory:
# "uniqueMember"
# ActiveDirectory:
# "member"
LDAP.groupPrsAttr=uniqueMember

#Seconds how long auth. user remains in cache.

#Is Required: No
#Valid values:
# 0 - No cache
# Min: 1, Max: No limit
#Default value: 180
LDAP.cacheTime=12

#Specify the max. number of cached users that have been successfully
#authenticated. When the cache overflows, the oldest entry is removed.

#Is Required: No
#Valid values:
# 0 - No cache
# Min: 1, Max: No limit
#Default value: 300
LDAP.cacheSize=4

#Time (in seconds) how long to ignore any further authentication
```

```
#requests for a particular User-Id.

#Is Required: No
#Valid values:
# Min: 1, Max: No limit
#Default value: 100
LDAP.denyTime=4

#Number of invalid logon attempts.

#Is Required: No
#Valid values:
# Min: 1, Max: No limit
#Default value: 3
LDAP.denyCount=3

#Specifies an output file for logging.

#Is Required: No
#Valid values: (Valid log file path)
#No default value
LDAP.logCallback=true

#Specifies the log level.

#Is Required: No
#Valid values:
# 0 - No logging
#   Min: 1
#   Max: 6
#No default value
#LDAP.nativeLogLevel=6

#Default group to be automatically included for all requests
#that return any groups
#Is Required: No

# LDAP.defaultGroup=DefGroup

#BaseBindDN where to find the users.
#Is Required: Yes
#and should contain the most detailed DN to find the users

# LDAP.personBindDn=ou=User,o=Org,dc=mycom,dc=com

#BaseBindDN where to find the groups.
#Is Required: Yes
#and should contain the most detailed DN to find the groups

# LDAP.groupBindDn=ou=Groups,o=Org,dc=mycom,dc=com

#Attribute name of the password.
```

```
#Required when changeing the password
#Is Required: Not always
#Default value:
# depending on serverType:
# OpenLdap:
# "userPassword"
# SunOneDirectory:
# "userPassword"
# ActiveDirectory:
# "unicodePwd"

# LDAP.passwdField=userPassword

#Allow to pass a complete BaseBindDN
#via the domain parameter.
#Is Required: No
#Valid values: 0, 1

# LDAP.allowdomainasbasebinddn=0

#Allow to specify which fields to search for as properties
#of a user entry
#Is Required: No
#Valid values: string, for example: "cn,sn,description"

# LDAP.personPropAttr

#Allow to specify which fields to search for as properties
#of a group entry
#Is Required: No
#Valid values: string, for example: "cn,description"

# LDAP.groupPropAttr

#Allow to use the special secure authentication using SASL,
#providing the directory supports this mechanism.
#Is Required: No
#Valid values: 0, 1 (default: 0)

# LDAP.ldapSaslBind

#Allow to switch from a non-secure connection to a TLS connection,
#providing the directory supports this mechanism.
#of a group entry
#Is Required: No
#Valid values: 0, 1 (default: 0)

# LDAP.ldapStartTls

#By default, the first "dc=" occurrence within the distinguished name
#name string denotes the domain name.
#If additional abbreviations want to be defined, one can use
```

```
#the following 2 parameter.
#Example:  Short="RnD;Admins;board"
#      with ↩
Long="ou=Rnd,ou=user,dc=mycom,dc=com;ou=Administrators,dc=mycom,dc=com;ou=VIP,dc=mycom,dc-com"
#LDAP.ldapDomainShort
#LDAP.ldapDomainLong

#If NOT the automatic domain name should be used to compose
#the canonical user id (SSXGetCanonicalUserId_A/W),
#specify this part of the ID here.
#LDAP.canonicalDomainName

#Three algorithms are supported to find the groups of a user:
#"ru", recurse up: take the group pointer from the user entry
#                  and continue to search up for all groups
#                  found
#"rd", recurse down: search for all groups that have the
#                    user as member (no recursion)
#"cp", computed property: use a special field in the user
#                         entry to find all groups
#                         --> computedGroupProp retuired
#Default: "ru"
#LDAP.resolveGroups

#If resolveGroup is set to "cp", this parameter must provide
#the field name to look for in the user entry that denotes
#the user groups
#Default: None
#LDAP.computedGroupProp=

#If the LDAP connection is protected by SSL/TLS, this
#parameter must be set.
#Valid Values: 0, 1
#Default: 0
#LDAP.ldapSSLConnection=1

################################################################
# ADSI Section

#Specifies the authentication type.

#Is Required: Yes
#Valid values: {"os", "ldap", "adsi", "iaf", "saf"}
#No default value

ADSI.authType=adsi

#Specifies the name of the AD Forest.

#Is Required: No, but should be specified
#Example: "dc=mycom,dc=com" (with a possible domain called "dc=eur,dc=mycom,dc=com")
#No default value
```

```
#ADSI.adsiForestDn

#Seconds how long auth. user remains in cache.

#Is Required: No
#Valid values:
# 0 - No cache
# Min: 1, Max: No limit
#Default value: 180
ADSI.cacheTime=12

#Specify the max. number of cached users that have been successfully
#authenticated. When the cache overflows, the oldest entry is removed.

#Is Required: No
#Valid values:
# 0 - No cache
# Min: 1, Max: No limit
#Default value: 300
ADSI.cacheSize=4

#Time (in seconds) how long to ignore any further authentication
#requests for a particular User-Id.

#Is Required: No
#Valid values:
# Min: 1, Max: No limit
#Default value: 100
ADSI.denyTime=4

#Number of invalid logon attempts.

#Is Required: No
#Valid values:
# Min: 1, Max: No limit
#Default value: 3
ADSI.denyCount=3

#Specifies an output file for logging.

#Is Required: No
#Valid values: (Valid log file path)
#No default value
#ADSI.nativeLogFile=SIN_SSX.log
ADSI.logCallback=true

#Specifies the log level.

#Is Required: No
#Valid values:
# 0 - No logging
#   Min: 1
#   Max: 6
#No default value
#ADSI.nativeLogLevel=6

#In case the scope for the node to access users needs to be limited,
#one can specify a particular subtree:
#Example: "ou=user,ou=Rnd,dc=mycom,dc=com"
#ADSI.adsiPersonBindDn

#In case the scope for the node to access groups needs to be limited,
#one can specify a particular subtree:
#Example: "ou=groups,ou=Rnd,dc=mycom,dc=com"
#ADSI.adsiGroupBindDn

#By default, the first "dc=" occurrence within the distinguished name
#name string denotes the domain name.
#If additional abbreviations want to be defined, one can use
#the following 2 parameter.
#Example:  Short="RnD;Admins;board"
#      with   ↩
Dn="ou=Rnd,ou=user,dc=mycom,dc=com;ou=Administrators,dc=mycom,dc=com;ou=VIP,dc=mycom,dc-com"
#ADSI.adsiDomainShort
#ADSI.adsiDomainDn

#If NOT the automatic domain name should be used to compose
#the canonical user id (SSXGetCanonicalUserId_A/W),
#specify this part of the ID here.
#ADSI.canonicalDomainName

#Three algorithms are supported to find the groups of a user:
#"ru", recurse up: take the group pointer from the user entry
#                  and continue to search up for all groups
#                  found
#"rd", recurse down: search for all groups that have the
#                    user as member (no recursion)
#"cp", computed property: use a special field in the user
#                         entry to find all groups
#                         --> computedGroupProp retuired
#Default: "ru"
#ADSI.resolveGroups

#If resolveGroup is set to "cp", this parameter must provide
#the field name to look for in the user entry that denotes
#the user groups
#Default: None
#ADSI.computedGroupProp=

################################################################
# IAF Section

#Specifies the authentication type.
```

```
#Is Required: Yes
#Valid values: {"os", "ldap", "adsi", "iaf", "saf"}
#No default value
IAF.authType=iaf

#Seconds how long auth. user remains in cache.

#Is Required: No
#Valid values:
# 0 - No cache
# Min: 1, Max: No limit
#Default value: 180
IAF.cacheTime=12

#Specify the max. number of cached users that have been successfully
#authenticated. When the cache overflows, the oldest entry is removed.

#Is Required: No
#Valid values:
# 0 - No cache
# Min: 1, Max: No limit
#Default value: 300
IAF.cacheSize=4

#Time (in seconds) how long to ignore any further authentication
#requests for a particular User-Id.

#Is Required: No
#Valid values:
# Min: 1, Max: No limit
#Default value: 100
IAF.denyTime=4

#Number of invalid logon attempts.

#Is Required: No
#Valid values:
# Min: 1, Max: No limit
#Default value: 3
IAF.denyCount=3

#Specifies an output file for logging.

#Is Required: No
#Valid values: (Valid log file path)
#No default value
#IAF.nativeLogFile=SIN_SSX.log
IAF.logCallback=true

#Specifies the log level.
```

```
#Is Required: No
#Valid values:
# 0 - No logging
#   Min: 1
#   Max: 6
#No default value
IAF.nativeLogLevel=6

#Specify the local code page to be used
#for converting strings from the
#IAF wire protocol (UTF8) to a local string
#Default: Unix, Win: ISO-8859-1, mainframe: IBM-037
#IAF.localCodePage=

#Specify the local code page to be used
#for converting strings from the
#IAF wire protocol (UTF8) to a local string
#Default: Unix, Win: sagssxtomcrypt, mainframe: SSXCTC
#IAF.cryptLib=

#Directory where to load dynamically the libraries:
# - broker[32][.dll|.so|.sl]
# - sagssxtomcrypt | SSXCTC (s. IAF.cryptLib)
#IAF.homeDir=
```

3 Integrated Authentication Framework Samples

- IAF Attribute File for Active Directory (ADSI) Authentication 20
- IAF Attribute File for Internal User Repository Authentication 24
- IAF Attribute File for LDAP User Repository Authentication 28
- IAF Attribute File for Active Directory Using LDAP Interface Authentication 32
- IAF Attribute File for Operating System Authentication 36

This chapter provides sample IAF related attribute files for commonly used authentication scenarios.

The sample attribute files are organized as follows:

IAF Attribute File for Active Directory (ADSI) Authentication

Replace the content of the attribute file you want to configure with the sample code below:

Note: Provide information about your environment by configuring setting in the following attributes (in the `SSX ADSI` section): `SERVERHOST` and `ADSIFORESTDN`.

```
****************************************************************
* Attribute file for IAF server.
****************************************************************

*    IAFnnn

DEFAULTS = BROKER

  BROKER-ID                 = IAFnnn
  RUN-MODE                  = IAF
  TRANSPORT                 = TCP-SSL

  AUTOLOGON                 = YES
  CALLABLE-RPC-SERVICES     = NO
  CLIENT-NONACT             = 99
  ICU-CONVERSION            = NO
  DYNAMIC-MEMORY-MANAGEMENT = YES
  NUM-WORKER                = 5
  TRACE-LEVEL               = 0


DEFAULTS = IAF

** IAF Service parameters: *************************************************
  IAF_LISTENADDRESS           = localhost
                                * the IAF servers own name, will be copied
                                * into each IAF Token (future use)
  IAFVALIDTIME                = 300
                                * default time the tokens are valid (in secs)
 * LOCALCODEPAGE              =
                                * default code page: ISO-8859-1, rsp. IBM-037
                                * used as input to
                                * MultiByteToWideChar/WideCharToMultiByte
                                * on Windows and
                                * iconv on Unix
```

```
** IAF Delegation:
 * IAFVERIFYDELEGATEDUSER     =
                                * YES/NO, default: YES
                                * verify if delegated userid really exists
 * IAFDELEGATEDAUTHUSER       =
                                * technical user id
 * IAFDELEGATEDAUTHDOMAIN     =
                                * domain of technical user
 * IAFDELEGATEDAUTHPASS       =
                                * encrypted password of technical user
 * IAFDELEGATEDCERTPATH       =
                                * file name to decrypt techn. user password
 * IAFDELEGATEDAUTHTIMEJITTER =
                                * allow +/- secs. difference between
                                * SSX program and IAF server

** SSX configuration patameters: ********************************************

***************************** SSX Common ************************************
  AUTHTYPE                    = ADSI
                               * Native authentication type (OS, INTERNAL, LDAP, ↩
ADSI)
  VALIDTIME                   = 0
                                * how long (secs) should user remain in cache
                                * 0=disabled
  DENYTIME                    = 60
                                * deny access for 60 secs after
                                * <denycont> false authentications
  DENYCOUNT                   = 0
                                * 0=deactivate, else no. of invalid auths
                                * before waiting <denytime> secs.
  MAXCACHEDUSERS              = 100
        * no. of successful auth'ed users
  * LOGFILE                   = LOG_FILE_PATH
        * log file path
  * LOGLEVEL      = 6
        * 0 - 6: set log level
  * DEFAULTDOMAIN             = defaultDomain
        * The default domain name


***************************** SSX OS ****************************************

  * AUTHDPATH      = DAEMON_PATH
        * Unix only! Explicit path of the privileged
        * daemon process.
  * UNIXADDMACHINENAME    = true/false
        * Machine name is added before users and groups i.e.
        * machine_name\user.
  * DEFAULTGROUP              = default_group
        * Any group can be used. Specify a default group name
        * here that should be returned with any of the group
        * results which are returned by repository manager.
  * WINNOIMPERSONISATION      = true/false
        * that specifies whether any data access should be made
        * under the impersonated userid of the logged in user
        * (false), or whether all access are made under the
        * account of the running process (true)

***************************** SSX INTERNAL ***************************************

 * INTERNALREPOSITORY         = INTERNAL_REPO_PATH
         * path for the file with internal users
***************************** SSX LDAP ***************************************

 * SERVERHOST                 = localhost
                                * where takes place the auths
 * SERVERPORT                 = 389
                                * port of server

** AUTHTYPE=LDAP only:
 * LDAPSERVERTYPE             = OpenLDAP
                                * use some predefined fields with
                                * "ActiveDirectory", "OpenLdap"(default),
                                * "SunOneDirectory", "Tivoli",
                                * "Novell" or "ApacheDS"
 * LDAPPERSONBINDDN           = "ou=people,dc=myorg,dc=com"
                                * node where to find the users
 * LDAPGROUPBINDDN            = "ou=groups,dc=myorg,dc=com"
        * node where to find the groups
 * LDAPUSERIDFIELD            = cn
                                * name of the user id field
 * LDAPGROUPIDFIELD           = cn
                                * name of the group id field
 * LDAPPERSONOBJECTCLASS      = "top,person"
        * user object class
 * LDAPGROUPOBJECTCLASS       = "top,groups"
        * group object class
 * LDAPPERSONGRPATTR          = memberOf
        * Property name of a user entry that points
        * from a user entry to the group that the user
        * is member of.
 * LDAPGROUPUSRATTR           = member
        * Property name of a group entry which points from the
        * group to the users (members).
 * LDAPALLOWDOMAINASBASEBINDDN = true
        * If this boolean field is “true” or “1”, the parameter
        * “domainname” will be interpreted as a BaseBindDN
        *(example: “ou=People,dc=myorg,dc=com”. Note that if
        * no explicit domain
 * LDAPCONNECTIONPEROPERATION = true
        * whether the LDAP connection should be created and closed
        * per method call (true), or whether the connection should
        * stay open until the user handle is closed
 * LDAPPERSONPROPERTYATTR     = "cn,displayName,description,mail,telephoneNumber, ↩

        * Defines the property names that can be accessed for a user
        * entry. The value is a comma separated list, which contains
        * the property name. When all of the specified properties do
        * not exist or are binary properties any user result list
        * will be empty.
 * LDAPGROUPPROPERTYATTR      = "cn,description"
        * Defines the property names that can be accessed for a group
        * entry. The value is a comma separated list, which contains
        * the property name. When all of the specified properties do
        * not exist or are binary properties any group result list will
        * be empty.
 * LDAPSSLCONNECTION          = true
        * the denoted ldap connection (serverHost and serverPort) is a
        * secured (over SSL/TLS) connection to an LDAP server
 * FOLLOWREFERRALS            = false
        * Whether the SSX must follow referrals or not. true/false
 * REFSERVERBINDINGTYPE       = same_creds
        * What kind of binding during referral following.
        * same_creds - use same credential for authentication to
        * the next LDAP server. no_creds use anonymous binding to the
        * next server
 * REFERRALHOPSCNT            = 1
        * Count of the referral hops. If this parameter is not specified
        * the count  is unlimited

***************************** SSX ADSI ***************************************


 SERVERHOST                   = <server_host>
                                * where takes place the auths
 * SERVERPORT                 = 389
                                * port of server

 LDAPSERVERTYPE               = ActiveDirectory

 ADSIFORESTDN                 = "DC=ad,DC=<organization>"
                                  * name of ADS forest
                                  * CAREFUL: do not mix with domain name
 * ADSIPERSONBASEBINDDN       = "dc=myusers,dc=com"
          * Specifies a BindDN that is used to access
          * a user. Note that this is only useful when
          * all users that are accessed are found under
          * in the same node
 * ADSIGROUPBASEBINDDN       = "dc=mygroups,dc=com"
                                  * Specifies a BindDN that is used to access
          * a group. Note that this is only useful when
          * all groups that are accessed are found under
          * in the same node.
```

```
DEFAULTS = TCP
   PORT = 11971


DEFAULTS = SSL
   PORT = 11958
   VERIFY-CLIENT = NO
   KEY-FILE = "..\..\Etc\IAFAppKey.pem"
   KEY-PASSWD = IAFAppKey
   KEY-STORE = "..\..\Etc\IAFAppCert.pem"
** TRUST-STORE = "..\..\Etc\IAFCaCert.pem"
*
```

IAF Attribute File for Internal User Repository Authentication

Replace the content of the attribute file you want to configure with the sample code below:

```
****************************************************************
* Attribute file for IAF server.
****************************************************************

*    IAFnnn

DEFAULTS = BROKER

  BROKER-ID                 = IAFnnn
  RUN-MODE                  = IAF
  TRANSPORT                 = TCP-SSL

  AUTOLOGON                 = YES
  CALLABLE-RPC-SERVICES     = NO
  CLIENT-NONACT             = 99
  ICU-CONVERSION            = NO
  DYNAMIC-MEMORY-MANAGEMENT = YES
  NUM-WORKER                = 5
  TRACE-LEVEL               = 0


DEFAULTS = IAF

** IAF Service parameters: *************************************************
  IAF_LISTENADDRESS           = localhost
                                * the IAF servers own name, will be copied
                                * into each IAF Token (future use)
  IAFVALIDTIME                = 300
                                * default time the tokens are valid (in secs)
 * LOCALCODEPAGE              =
                                * default code page: ISO-8859-1, rsp. IBM-037
                                * used as input to
                                * MultiByteToWideChar/WideCharToMultiByte
                                * on Windows and
                                * iconv on Unix

** IAF Delegation:
 * IAFVERIFYDELEGATEDUSER     =
                                * YES/NO, default: YES
                                * verify if delegated userid really exists
 * IAFDELEGATEDAUTHUSER       =
                                * technical user id
 * IAFDELEGATEDAUTHDOMAIN     =
                                * domain of technical user
 * IAFDELEGATEDAUTHPASS       =
                                * encrypted password of technical user
 * IAFDELEGATEDCERTPATH       =
                                * file name to decrypt techn. user password
 * IAFDELEGATEDAUTHTIMEJITTER =
                                * allow +/- secs. difference between
                                * SSX program and IAF server

** SSX configuration patameters: *******************************************

***************************** SSX Common ***********************************
  AUTHTYPE                    = INTERNAL
                               * Native authentication type (OS, INTERNAL, LDAP, ↩
ADSI)
  VALIDTIME                   = 0
                                * how long (secs) should user remain in cache
                                * 0=disabled
  DENYTIME                    = 60
                                * deny access for 60 secs after
                                * <denycont> false authentications
  DENYCOUNT                   = 0
                                * 0=deactivate, else no. of invalid auths
                                * before waiting <denytime> secs.
  MAXCACHEDUSERS              = 100
        * no. of successful auth'ed users
  * LOGFILE                   = LOG_FILE_PATH
        * log file path
  * LOGLEVEL      = 6
        * 0 - 6: set log level
  * DEFAULTDOMAIN             = defaultDomain
        * The default domain name


***************************** SSX OS ***************************************

  * AUTHDPATH      = DAEMON_PATH
        * Unix only! Explicit path of the privileged
        * daemon process.
  * UNIXADDMACHINENAME    = true/false
        * Machine name is added before users and groups i.e.
        * machine_name\user.
  * DEFAULTGROUP              = default_group
        * Any group can be used. Specify a default group name
        * here that should be returned with any of the group
        * results which are returned by repository manager.
  * WINNOIMPERSONISATION      = true/false
        * that specifies whether any data access should be made
        * under the impersonated userid of the logged in user
        * (false), or whether all access are made under the
        * account of the running process (true)

***************************** SSX INTERNAL ***************************************

    INTERNALREPOSITORY          = ".\ssx_user.properties"
         * path for the file with internal users
***************************** SSX LDAP ***************************************

 * SERVERHOST                 = localhost
                                * where takes place the auths
 * SERVERPORT                 = 389
                                * port of server

** AUTHTYPE=LDAP only:
 * LDAPSERVERTYPE             = OpenLDAP
                                * use some predefined fields with
                                * "ActiveDirectory", "OpenLdap"(default),
                                * "SunOneDirectory", "Tivoli",
                                * "Novell" or "ApacheDS"
 * LDAPPERSONBINDDN           = "ou=people,dc=myorg,dc=com"
                                * node where to find the users
 * LDAPGROUPBINDDN            = "ou=groups,dc=myorg,dc=com"
        * node where to find the groups
 * LDAPUSERIDFIELD            = cn
                                * name of the user id field
 * LDAPGROUPIDFIELD           = cn
                                * name of the group id field
 * LDAPPERSONOBJECTCLASS      = "top,person"
        * user object class
 * LDAPGROUPOBJECTCLASS       = "top,groups"
        * group object class
 * LDAPPERSONGRPATTR          = memberOf
        * Property name of a user entry that points
        * from a user entry to the group that the user
        * is member of.
 * LDAPGROUPUSRATTR           = member
        * Property name of a group entry which points from the
        * group to the users (members).
 * LDAPALLOWDOMAINASBASEBINDDN = true
        * If this boolean field is “true” or “1”, the parameter
        * “domainname” will be interpreted as a BaseBindDN
        *(example: “ou=People,dc=myorg,dc=com”. Note that if
        * no explicit domain
 * LDAPCONNECTIONPEROPERATION = true
        * whether the LDAP connection should be created and closed
        * per method call (true), or whether the connection should
        * stay open until the user handle is closed
 * LDAPPERSONPROPERTYATTR     = "cn,displayName,description,mail,telephoneNumber, ↩

        * Defines the property names that can be accessed for a user
        * entry. The value is a comma separated list, which contains
        * the property name. When all of the specified properties do
        * not exist or are binary properties any user result list
        * will be empty.
 * LDAPGROUPPROPERTYATTR      = "cn,description"
        * Defines the property names that can be accessed for a group
        * entry. The value is a comma separated list, which contains
        * the property name. When all of the specified properties do
        * not exist or are binary properties any group result list will
        * be empty.
 * LDAPSSLCONNECTION          = true
        * the denoted ldap connection (serverHost and serverPort) is a
        * secured (over SSL/TLS) connection to an LDAP server
 * FOLLOWREFERRALS            = false
        * Whether the SSX must follow referrals or not. true/false
 * REFSERVERBINDINGTYPE       = same_creds
        * What kind of binding during referral following.
        * same_creds - use same credential for authentication to
        * the next LDAP server. no_creds use anonymous binding to the
        * next server
 * REFERRALHOPSCNT            = 1
        * Count of the referral hops. If this parameter is not specified
        * the count  is unlimited

***************************** SSX ADSI ***************************************


 SERVERHOST                   = eur.ad.sag
                                * where takes place the auths
 * SERVERPORT                 = 389
                                * port of server

 LDAPSERVERTYPE               = ActiveDirectory

 ADSIFORESTDN                 = "DC=ad,DC=sag"
                                  * name of ADS forest
                                  * CAREFUL: do not mix with domain name
 * ADSIPERSONBASEBINDDN       = "dc=myusers,dc=com"
          * Specifies a BindDN that is used to access
          * a user. Note that this is only useful when
          * all users that are accessed are found under
          * in the same node
 * ADSIGROUPBASEBINDDN        = "dc=mygroups,dc=com"
                                  * Specifies a BindDN that is used to access
          * a group. Note that this is only useful when
          * all groups that are accessed are found under
          * in the same node.


DEFAULTS = TCP
   PORT = 11971


DEFAULTS = SSL
   PORT = 11958
   VERIFY-CLIENT = NO
   KEY-FILE = "..\..\Etc\IAFAppKey.pem"
   KEY-PASSWD = IAFAppKey
   KEY-STORE = "..\..\Etc\IAFAppCert.pem"
** TRUST-STORE = "..\..\Etc\IAFCaCert.pem"
*
```

IAF Attribute File for LDAP User Repository Authentication

Replace the content of the attribute file you want to configure with the sample code below:

Note: Provide information about your environment by configuring setting in the following attributes (in the `SSX LDAP` section): `SERVERHOST`, `LDAPPERSONBINDDN` and `LDAPGROUPBINDDN`.

```
****************************************************************
* Attribute file for IAF server.
****************************************************************

*    IAFnnn

DEFAULTS = BROKER

  BROKER-ID                 = IAFnnn
  RUN-MODE                  = IAF
  TRANSPORT                 = TCP-SSL

  AUTOLOGON                 = YES
  CALLABLE-RPC-SERVICES     = NO
  CLIENT-NONACT             = 99
  ICU-CONVERSION            = NO
  DYNAMIC-MEMORY-MANAGEMENT = YES
  NUM-WORKER                = 5
  TRACE-LEVEL               = 0


DEFAULTS = IAF

** IAF Service parameters: *************************************************
```

```
  IAF_LISTENADDRESS           = localhost
                                * the IAF servers own name, will be copied
                                * into each IAF Token (future use)
  IAFVALIDTIME                = 300
                                * default time the tokens are valid (in secs)
 * LOCALCODEPAGE              =
                                * default code page: ISO-8859-1, rsp. IBM-037
                                * used as input to
                                * MultiByteToWideChar/WideCharToMultiByte
                                * on Windows and
                                * iconv on Unix

** IAF Delegation:
 * IAFVERIFYDELEGATEDUSER     =
                                * YES/NO, default: YES
                                * verify if delegated userid really exists
 * IAFDELEGATEDAUTHUSER       =
                                * technical user id
 * IAFDELEGATEDAUTHDOMAIN     =
                                * domain of technical user
 * IAFDELEGATEDAUTHPASS       =
                                * encrypted password of technical user
 * IAFDELEGATEDCERTPATH       =
                                * file name to decrypt techn. user password
 * IAFDELEGATEDAUTHTIMEJITTER =
                                * allow +/- secs. difference between
                                * SSX program and IAF server

** SSX configuration patameters: *******************************************

***************************** SSX Common ***********************************
  AUTHTYPE                    = LDAP
                               * Native authentication type (OS, INTERNAL, LDAP, ↩
ADSI)
  VALIDTIME                   = 0
                                * how long (secs) should user remain in cache
                                * 0=disabled
  DENYTIME                    = 60
                                * deny access for 60 secs after
                                * <denycont> false authentications
  DENYCOUNT                   = 0
                                * 0=deactivate, else no. of invalid auths
                                * before waiting <denytime> secs.
  MAXCACHEDUSERS              = 100
        * no. of successful auth'ed users
  * LOGFILE                   = LOG_FILE_PATH
        * log file path
  * LOGLEVEL      = 6
        * 0 - 6: set log level
  * DEFAULTDOMAIN             = defaultDomain
        * The default domain name
```

```
***************************** SSX OS ***************************************

  * AUTHDPATH      = DAEMON_PATH
        * Unix only! Explicit path of the privileged
        * daemon process.
  * UNIXADDMACHINENAME    = true/false
        * Machine name is added before users and groups i.e.
        * machine_name\user.
  * DEFAULTGROUP              = default_group
        * Any group can be used. Specify a default group name
        * here that should be returned with any of the group
        * results which are returned by repository manager.
  * WINNOIMPERSONISATION      = true/false
        * that specifies whether any data access should be made
        * under the impersonated userid of the logged in user
        * (false), or whether all access are made under the
        * account of the running process (true)

***************************** SSX INTERNAL ***************************************

 * INTERNALREPOSITORY         = INTERNAL_REPO_PATH
        * path for the file with internal users
***************************** SSX LDAP ***************************************

  SERVERHOST                  = <server_host>
                                * where takes place the auths
  SERVERPORT                  = 389
                                * port of server

** AUTHTYPE=LDAP only:
  LDAPSERVERTYPE              = OpenLDAP
                                * use some predefined fields with
                                * "ActiveDirectory", "OpenLdap"(default),
                                * "SunOneDirectory", "Tivoli",
                                * "Novell" or "ApacheDS"
  LDAPPERSONBINDDN            = "ou=users,ou=<organization>,o=<organization>"
                                * node where to find the users
  LDAPGROUPBINDDN             = "ou=groups,ou=<organization>,o=<organization>"
        * node where to find the groups
  LDAPUSERIDFIELD             = uid
                                * name of the user id field
  LDAPGROUPIDFIELD            = cn
                                * name of the group id field
  LDAPPERSONOBJECTCLASS       = "top,person,organizationalPerson,inetOrgPerson"
        * user object class
  LDAPGROUPOBJECTCLASS        = "top,groupOfUniqueNames"
        * group object class
 * LDAPPERSONGRPATTR          = memberOf
        * Property name of a user entry that points
        * from a user entry to the group that the user
        * is member of.
  LDAPGROUPUSRATTR            = uniqueMember
        * Property name of a group entry which points from the
        * group to the users (members).
 * LDAPALLOWDOMAINASBASEBINDDN = true
        * If this boolean field is “true” or “1”, the parameter
        * “domainname” will be interpreted as a BaseBindDN
        *(example: “ou=People,dc=myorg,dc=com”. Note that if
        * no explicit domain
 * LDAPCONNECTIONPEROPERATION = true
        * whether the LDAP connection should be created and closed
        * per method call (true), or whether the connection should
        * stay open until the user handle is closed
 * LDAPPERSONPROPERTYATTR     = "cn,displayName,description,mail,telephoneNumber, ↩

        * Defines the property names that can be accessed for a user
        * entry. The value is a comma separated list, which contains
        * the property name. When all of the specified properties do
        * not exist or are binary properties any user result list
        * will be empty.
  LDAPGROUPPROPERTYATTR       = objectClass
        * Defines the property names that can be accessed for a group
        * entry. The value is a comma separated list, which contains
        * the property name. When all of the specified properties do
        * not exist or are binary properties any group result list will
        * be empty.
 * LDAPSSLCONNECTION          = true
        * the denoted ldap connection (serverHost and serverPort) is a
        * secured (over SSL/TLS) connection to an LDAP server
  FOLLOWREFERRALS             = false
        * Whether the SSX must follow referrals or not. true/false
 * REFSERVERBINDINGTYPE       = same_creds
        * What kind of binding during referral following.
        * same_creds - use same credential for authentication to
        * the next LDAP server. no_creds use anonymous binding to the
        * next server
 * REFERRALHOPSCNT            = 1
        * Count of the referral hops. If this parameter is not specified
        * the count  is unlimited
   RESOLVEGROUPS              = rd


***************************** SSX ADSI ***************************************


 * SERVERHOST                 = localhost
                                * where takes place the auths
 * SERVERPORT                 = 389
                                * port of server
 * ADSIFORESTDN               = "dc=myorg,dc=com"
                                  * name of ADS forest
                                  * CAREFUL: do not mix with domain name
 * ADSIPERSONBASEBINDDN       = "dc=myusers,dc=com"
```

```
          * Specifies a BindDN that is used to access
          * a user. Note that this is only useful when
          * all users that are accessed are found under
          * in the same node
 * ADSIGROUPBASEBINDDN        = "dc=mygroups,dc=com"
                                  * Specifies a BindDN that is used to access
          * a group. Note that this is only useful when
          * all groups that are accessed are found under
          * in the same node.


DEFAULTS = TCP
   PORT = 11971


DEFAULTS = SSL
   PORT = 11958
   VERIFY-CLIENT = NO
   KEY-FILE = "..\..\Etc\IAFAppKey.pem"
   KEY-PASSWD = IAFAppKey
   KEY-STORE = "..\..\Etc\IAFAppCert.pem"
** TRUST-STORE = "..\..\Etc\IAFCaCert.pem"
*
```

IAF Attribute File for Active Directory Using LDAP Interface Authentication

Replace the content of the attribute file you want to configure with the sample code below:

Note: Provide information about your environment by configuring setting in the following attributes (in the `SSX LDAP` section): `SERVERHOST`, `LDAPPERSONBINDDN` and `LDAPGROUPBINDDN`.

```
****************************************************************
* Attribute file for IAF server.
****************************************************************

*    IAFnnn

DEFAULTS = BROKER

  BROKER-ID                 = IAFnnn
  RUN-MODE                  = IAF
  TRANSPORT                 = TCP-SSL

  AUTOLOGON                 = YES
  CALLABLE-RPC-SERVICES     = NO
  CLIENT-NONACT             = 99
  ICU-CONVERSION            = NO
  DYNAMIC-MEMORY-MANAGEMENT = YES
  NUM-WORKER                = 5
  TRACE-LEVEL               = 0


DEFAULTS = IAF

** IAF Service parameters: *************************************************
  IAF_LISTENADDRESS           = localhost
                                * the IAF servers own name, will be copied
                                * into each IAF Token (future use)
  IAFVALIDTIME                = 300
                                * default time the tokens are valid (in secs)
 * LOCALCODEPAGE              =
                                * default code page: ISO-8859-1, rsp. IBM-037
                                * used as input to
                                * MultiByteToWideChar/WideCharToMultiByte
                                * on Windows and
                                * iconv on Unix

** IAF Delegation:
 * IAFVERIFYDELEGATEDUSER     =
                                * YES/NO, default: YES
                                * verify if delegated userid really exists
 * IAFDELEGATEDAUTHUSER       =
                                * technical user id
 * IAFDELEGATEDAUTHDOMAIN     =
                                * domain of technical user
 * IAFDELEGATEDAUTHPASS       =
                                * encrypted password of technical user
 * IAFDELEGATEDCERTPATH       =
                                * file name to decrypt techn. user password
 * IAFDELEGATEDAUTHTIMEJITTER =
                                * allow +/- secs. difference between
                                * SSX program and IAF server

** SSX configuration patameters: *******************************************

***************************** SSX Common ***********************************
  AUTHTYPE                    = LDAP
                               * Native authentication type (OS, INTERNAL, LDAP, ↩
ADSI)
  VALIDTIME                   = 0
                                * how long (secs) should user remain in cache
                                * 0=disabled
  DENYTIME                    = 60
                                * deny access for 60 secs after
                                * <denycont> false authentications
  DENYCOUNT                   = 0
                                * 0=deactivate, else no. of invalid auths
                                * before waiting <denytime> secs.
  MAXCACHEDUSERS              = 100
        * no. of successful auth'ed users
  * LOGFILE                   = LOG_FILE_PATH
        * log file path
  * LOGLEVEL      = 6
        * 0 - 6: set log level
  * DEFAULTDOMAIN             = defaultDomain
        * The default domain name


**************************** SSX OS ***************************************

  * AUTHDPATH      = DAEMON_PATH
        * Unix only! Explicit path of the privileged
        * daemon process.
  * UNIXADDMACHINENAME    = true/false
        * Machine name is added before users and groups i.e.
        * machine_name\user.
  * DEFAULTGROUP              = default_group
        * Any group can be used. Specify a default group name
        * here that should be returned with any of the group
        * results which are returned by repository manager.
  * WINNOIMPERSONISATION      = true/false
        * that specifies whether any data access should be made
        * under the impersonated userid of the logged in user
        * (false), or whether all access are made under the
        * account of the running process (true)

**************************** SSX INTERNAL ***************************************

 * INTERNALREPOSITORY         = INTERNAL_REPO_PATH
        * path for the file with internal users
**************************** SSX LDAP ***************************************

  SERVERHOST                  = <server_host>
                                * where takes place the auths
  SERVERPORT                  = 389
                                * port of server

** AUTHTYPE=LDAP only:
  LDAPSERVERTYPE              = ActiveDirectory
                                * use some predefined fields with
                                * "ActiveDirectory", "OpenLdap"(default),
                                * "SunOneDirectory", "Tivoli",
                                * "Novell" or "ApacheDS"
  LDAPPERSONBINDDN            = "dc=eur,dc=ad,dc=<organization>"
                                * node where to find the users
  LDAPGROUPBINDDN             = "DC=ad,DC=<organization>"
        * node where to find the groups
  LDAPUSERIDFIELD             = cn
                                * name of the user id field
  LDAPGROUPIDFIELD            = cn
                                * name of the group id field
  LDAPPERSONOBJECTCLASS       = "top,person,organizationalPerson,user"
```

```
        * user object class
  LDAPGROUPOBJECTCLASS        = "top,group"
        * group object class
  LDAPPERSONGRPATTR           = memberOf
        * Property name of a user entry that points
        * from a user entry to the group that the user
        * is member of.
 * LDAPGROUPUSRATTR           = member
        * Property name of a group entry which points from the
        * group to the users (members).
  LDAPALLOWDOMAINASBASEBINDDN = true
        * If this boolean field is “true” or “1”, the parameter
        * “domainname” will be interpreted as a BaseBindDN
        *(example: “ou=People,dc=myorg,dc=com”. Note that if
        * no explicit domain
 * LDAPCONNECTIONPEROPERATION = true
        * whether the LDAP connection should be created and closed
        * per method call (true), or whether the connection should
        * stay open until the user handle is closed
 * LDAPPERSONPROPERTYATTR    = "cn,displayName,description,mail,telephoneNumber, ↩

        * Defines the property names that can be accessed for a user
        * entry. The value is a comma separated list, which contains
        * the property name. When all of the specified properties do
        * not exist or are binary properties any user result list
        * will be empty.
  LDAPGROUPPROPERTYATTR       = member
        * Defines the property names that can be accessed for a group
        * entry. The value is a comma separated list, which contains
        * the property name. When all of the specified properties do
        * not exist or are binary properties any group result list will
        * be empty.
 * LDAPSSLCONNECTION          = true
        * the denoted ldap connection (serverHost and serverPort) is a
        * secured (over SSL/TLS) connection to an LDAP server
  FOLLOWREFERRALS             = false
        * Whether the SSX must follow referrals or not. true/false
 * REFSERVERBINDINGTYPE       = same_creds
        * What kind of binding during referral following.
        * same_creds - use same credential for authentication to
        * the next LDAP server. no_creds use anonymous binding to the
        * next server
 * REFERRALHOPSCNT            = 1
        * Count of the referral hops. If this parameter is not specified
        * the count  is unlimited

***************************** SSX ADSI ****************************************


 * SERVERHOST                 = localhost
                                * where takes place the auths
 * SERVERPORT                 = 389
```

```
                                * port of server
 * ADSIFORESTDN               = "dc=myorg,dc=com"
                                  * name of ADS forest
                                  * CAREFUL: do not mix with domain name
 * ADSIPERSONBASEBINDDN       = "dc=myusers,dc=com"
          * Specifies a BindDN that is used to access
          * a user. Note that this is only useful when
          * all users that are accessed are found under
          * in the same node
 * ADSIGROUPBASEBINDDN        = "dc=mygroups,dc=com"
                                  * Specifies a BindDN that is used to access
          * a group. Note that this is only useful when
          * all groups that are accessed are found under
          * in the same node.


DEFAULTS = TCP
   PORT = 11971


DEFAULTS = SSL
   PORT = 11958
   VERIFY-CLIENT = NO
   KEY-FILE = "..\..\Etc\IAFAppKey.pem"
   KEY-PASSWD = IAFAppKey
   KEY-STORE = "..\..\Etc\IAFAppCert.pem"
** TRUST-STORE = "..\..\Etc\IAFCaCert.pem"
*
```

IAF Attribute File for Operating System Authentication

Replace the content of the attribute file you want to configure with the sample code below:

```
****************************************************************
* Attribute file for IAF server.
****************************************************************

*    IAFnnn

DEFAULTS = BROKER

  BROKER-ID                 = IAFnnn
  RUN-MODE                  = IAF
  TRANSPORT                 = TCP-SSL

  AUTOLOGON                 = YES
  CALLABLE-RPC-SERVICES     = NO
  CLIENT-NONACT             = 99
  ICU-CONVERSION            = NO
  DYNAMIC-MEMORY-MANAGEMENT = YES
  NUM-WORKER                = 5
  TRACE-LEVEL               = 0


DEFAULTS = IAF

** IAF Service parameters: *************************************************
  IAF_LISTENADDRESS           = localhost
                                * the IAF servers own name, will be copied
                                * into each IAF Token (future use)
  IAFVALIDTIME                = 300
                                * default time the tokens are valid (in secs)
 * LOCALCODEPAGE              =
                                * default code page: ISO-8859-1, rsp. IBM-037
                                * used as input to
                                * MultiByteToWideChar/WideCharToMultiByte
                                * on Windows and
                                * iconv on Unix

** IAF Delegation:
 * IAFVERIFYDELEGATEDUSER     =
                                * YES/NO, default: YES
                                * verify if delegated userid really exists
 * IAFDELEGATEDAUTHUSER       =
                                * technical user id
 * IAFDELEGATEDAUTHDOMAIN     =
                                * domain of technical user
 * IAFDELEGATEDAUTHPASS       =
                                * encrypted password of technical user
 * IAFDELEGATEDCERTPATH       =
                                * file name to decrypt techn. user password
 * IAFDELEGATEDAUTHTIMEJITTER =
                                * allow +/- secs. difference between
                                * SSX program and IAF server

** SSX configuration patameters: *******************************************

***************************** SSX Common ***********************************
  AUTHTYPE                    = OS
                               * Native authentication type (OS, INTERNAL, LDAP, ↩
ADSI)
  VALIDTIME                   = 0
                                * how long (secs) should user remain in cache
                                * 0=disabled
  DENYTIME                    = 60
                                * deny access for 60 secs after
                                * <denycont> false authentications
  DENYCOUNT                   = 0
                                * 0=deactivate, else no. of invalid auths
                                * before waiting <denytime> secs.
  MAXCACHEDUSERS              = 100
```

```
        * no. of successful auth'ed users
  * LOGFILE                   = LOG_FILE_PATH
        * log file path
  * LOGLEVEL      = 6
        * 0 - 6: set log level
  * DEFAULTDOMAIN             = defaultDomain
        * The default domain name


***************************** SSX OS ***************************************

  * AUTHDPATH      = DAEMON_PATH
        * Unix only! Explicit path of the privileged
        * daemon process.
  * UNIXADDMACHINENAME    = true/false
        * Machine name is added before users and groups i.e.
        * machine_name\user.
  * DEFAULTGROUP              = default_group
        * Any group can be used. Specify a default group name
        * here that should be returned with any of the group
        * results which are returned by repository manager.
  * WINNOIMPERSONISATION      = true/false
        * that specifies whether any data access should be made
        * under the impersonated userid of the logged in user
        * (false), or whether all access are made under the
        * account of the running process (true)

***************************** SSX INTERNAL ***************************************

 * INTERNALREPOSITORY          = INTERNAL_REPO_PATH
         * path for the file with internal users
***************************** SSX LDAP ***************************************

 * SERVERHOST                 = localhost
                                * where takes place the auths
 * SERVERPORT                 = 389
                                * port of server

** AUTHTYPE=LDAP only:
 * LDAPSERVERTYPE             = OpenLDAP
                                * use some predefined fields with
                                * "ActiveDirectory", "OpenLdap"(default),
                                * "SunOneDirectory", "Tivoli",
                                * "Novell" or "ApacheDS"
 * LDAPPERSONBINDDN           = "ou=people,dc=myorg,dc=com"
                                * node where to find the users
 * LDAPGROUPBINDDN            = "ou=groups,dc=myorg,dc=com"
        * node where to find the groups
 * LDAPUSERIDFIELD            = cn
                                * name of the user id field
 * LDAPGROUPIDFIELD           = cn
                                * name of the group id field
 * LDAPPERSONOBJECTCLASS      = "top,person"
        * user object class
 * LDAPGROUPOBJECTCLASS       = "top,groups"
        * group object class
 * LDAPPERSONGRPATTR          = memberOf
        * Property name of a user entry that points
        * from a user entry to the group that the user
        * is member of.
 * LDAPGROUPUSRATTR           = member
        * Property name of a group entry which points from the
        * group to the users (members).
 * LDAPALLOWDOMAINASBASEBINDDN = true
        * If this boolean field is “true” or “1”, the parameter
        * “domainname” will be interpreted as a BaseBindDN
        *(example: “ou=People,dc=myorg,dc=com”. Note that if
        * no explicit domain
 * LDAPCONNECTIONPEROPERATION = true
        * whether the LDAP connection should be created and closed
        * per method call (true), or whether the connection should
        * stay open until the user handle is closed
 * LDAPPERSONPROPERTYATTR     = "cn,displayName,description,mail,telephoneNumber, ↩

        * Defines the property names that can be accessed for a user
        * entry. The value is a comma separated list, which contains
        * the property name. When all of the specified properties do
        * not exist or are binary properties any user result list
        * will be empty.
 * LDAPGROUPPROPERTYATTR      = "cn,description"
        * Defines the property names that can be accessed for a group
        * entry. The value is a comma separated list, which contains
        * the property name. When all of the specified properties do
        * not exist or are binary properties any group result list will
        * be empty.
 * LDAPSSLCONNECTION          = true
        * the denoted ldap connection (serverHost and serverPort) is a
        * secured (over SSL/TLS) connection to an LDAP server
 * FOLLOWREFERRALS            = false
        * Whether the SSX must follow referrals or not. true/false
 * REFSERVERBINDINGTYPE       = same_creds
        * What kind of binding during referral following.
        * same_creds - use same credential for authentication to
        * the next LDAP server. no_creds use anonymous binding to the
        * next server
 * REFERRALHOPSCNT            = 1
        * Count of the referral hops. If this parameter is not specified
        * the count  is unlimited
 * RESOLVEGROUPS              = rd
        * Resolve goups algorithm ru, rd, cp

**************************** SSX ADSI **************************************
```

```
 * SERVERHOST                 = localhost
                                * where takes place the auths
 * SERVERPORT                 = 389
                                * port of server
 * ADSIFORESTDN               = "dc=myorg,dc=com"
                                  * name of ADS forest
                                  * CAREFUL: do not mix with domain name
 * ADSIPERSONBASEBINDDN       = "dc=myusers,dc=com"
          * Specifies a BindDN that is used to access
          * a user. Note that this is only useful when
          * all users that are accessed are found under
          * in the same node
 * ADSIGROUPBASEBINDDN        = "dc=mygroups,dc=com"
                                  * Specifies a BindDN that is used to access
          * a group. Note that this is only useful when
          * all groups that are accessed are found under
          * in the same node.


DEFAULTS = TCP
   PORT = 11971


DEFAULTS = SSL
   PORT = 11958
   VERIFY-CLIENT = NO
   KEY-FILE = "..\..\Etc\IAFAppKey.pem"
   KEY-PASSWD = IAFAppKey
   KEY-STORE = "..\..\Etc\IAFAppCert.pem"
** TRUST-STORE = "..\..\Etc\IAFCaCert.pem"
*
```

4 Troubleshooting

- Troubleshooting Integrated Authentication Framework 42
- Troubleshooting sagssxauthd2 43
- Turning on SIN Logging 43
- Using a Specific log4j for Logging Information 44
- Verifying the JAAS Configuration in Tomcat 46
- Running the Test Servlet 46
- If All Other Things Fail... 47

This chapter details on ways of troubleshooting SIN.

The information is organized under the following headings:

Troubleshooting Integrated Authentication Framework

- If you cannot start the IAF service, proceed with the following check-list:

  1. Make sure that the following service and daemon are running:

     - **Windows**
       Software AG Integrated Authentication Framework

     - **UNIX**
       iafsrv

     You can start the `iafsrv` daemon from `Software AG_directory`***/bin/iafdstart***.

  2. Make sure that the ports are available.

  3. Make sure that the configuration name is correct. The attribute file name has to be the same as the directory name including an extension .attr. The broker-id has to be the same as the directory name as well.

- If you cannot see IAF integrated within SMH administration web user interface, see Installing System Management Hub over an Integrated Authentication Framework Installation

- If you see Integrated Authentication Framework Service but you cannot do anything else, check the user in the following registry key: HKEY_LOCAL_MACHINE\SOFTWARE\Software AG\System Management Hub\Products\Integrated Authentication Framework\Users. The user in the registry key must be the same as the user that you used to log on the system.

- There is a patch by HP, that has influence on the stack size and requires larger stack size to be configured. The patch ID is PHCO_33173. The fixes introduced with PHCO_33173 require more local storage on the stack. Multithreaded applications that use the default stack size (of 64K for PA and 256K for IPF) or a smaller custom size, may fail with SIGBUS after PHCO_33173 is installed.

  To prevent this, increase the thread stack size manually by at least 64K. For applications that use the default stack size, this can be done by installing PHCO_34718 (or a replacement of that) and setting the corresponding environment variable before launching the application.

  You can monitor the actual values of the stack size and iafnuc about the real stack size requirement by using external monitoring tools.

Troubleshooting sagssxauthd2

When you install CentraSite on a network file system (NFS) which is mapped to the local one, the local policies do not allow access rights, such as root or setuid to the remote installation. As a result, the *sagssxauthd2* executable does not work properly despite the properly configured root and setuid rights.

▶ To resolve the issues with the remote *sagssxauthd2* executables

1. Copy the *sagssxauthd2* executable on the local file system.
2. Set its root and setuid rights.
3. To use the *sagssxauthd2* on the remote installation of CentraSite, you must replace the remote executable files in the corresponding directories with symbolic hyperlinks that point to the locally copied executable.

Turning on SIN Logging

SIN uses the *log4j* package for logging data. Ensure that the *log4j* logging level for `com.softwareag.security` is set to DEBUG. If this does not help you to solve the problem yourself, contact Software AG Customer Support.

▶ To set the log level in *log4j* using the property style file:

- Use the following:

```
# Set log level for package com.softwareag.security to DEBUG:
log4j.logger.com.softwareag.security=DEBUG
```

▶ To set the log level in *log4j* using the XML file:

- Use the following:

```
<logger name="com.softwareag.security">
        <level value="DEBUG"/>
    </logger>
```

You can configure Security Infrastructure login modules to log information into an external file on the file system.

Note: It is recommended to use these logging settings to resolve only severe issues or system crashes. These logging settings have impact on the system performance and if you configure the system to log information constantly this leads to reduced overall performance.

To switch on logging, you must include the following properties into the properties list of the first login module of the stack in the login context (JAAS configuration):

```
useLog="true"
logLevel="debug"
logFile="<path_to_the_log_file>"
```

Thus, you enable DEBUG severity logging on all modules that are included in the JAAS configuration context. The result file contains the entire debug information generated during the login process, role management and user repository management.

When you specify the path to the log file, make sure that the directory is not write-protected for the user who executes the Java Virtual Machine. On Unix based operating systems it is recommended to use */tmp* directory.

It is recommended that you switch off the logging after you collect sufficient information about the issues. If you do not change these logging settings, the system keeps logging information in the file which leads to greater file size and reduced overall performance. Alternatively, instead of configuring external logging on Security Infrastructure, you can also check the system logging.

Using a Specific log4j for Logging Information

Setting the *log4j* configuration can be tricky in an Apache context. Tomcat uses *log4j* but it is possible you deploy other web applications that also use *log4j* configuration files. Usually, the *log4j* of the web application that is loaded first is the one that is used. In such cases, you must configure your system to use a specific *log4j* configuration.

Following is a sample development scenario that is valid for webMethods products (for example, CentraSite):

To use a specific *log4j* configuration

1 Provide the *log4j* you want to use.

For example, use the following:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE log4j:configuration SYSTEM "log4j.dtd">

<log4j:configuration xmlns:log4j="http://jakarta.apache.org/log4j/">

  <appender name="Console" class="org.apache.log4j.ConsoleAppender">
    <param name="Target" value="System.out"/>
    <layout class="org.apache.log4j.PatternLayout">
      <param name="ConversionPattern" value="%d{ABSOLUTE} [%t] %-5p %c %x - ↩
%m%n"/>
    </layout>
  </appender>

  <root>
    <priority value ="INFO" />
    <appender-ref ref="Console" />
  </root>

  <!-- Infos for the security - set level to DEBUG if needed. -->
  <logger name="com.softwareag.security">
    <level value="DEBUG"/>
  </logger>

</log4j:configuration>
```

2 Put the file in your Tomcat installation directory under *common/classes*.

3 Modify the path names to the log directories according to your installation.

4 Search in your Tomcat *webapps* to see if there are *log4j.xml* configuration files you do not need.

5 Rename them temporarily.

6 Restart Tomcat.

Note: Debug logging takes time and fills your log files. Remember to switch the logging level back to INFO once you are done.

Verifying the JAAS Configuration in Tomcat

SIN uses JAAS to determine which `LoginModules` to call. The configuration of the JAAS environment may be done by a configuration file that is located in the *conf directory* in the standard installation.

▶ To verify the JAAS configuration

1. Check the file to verify that all paths and URL in it are valid.
2. For UNIX platforms, check if the path to the *ssx auth* daemon is correct and if the executable it points to has the S-bit set.
3. CentraSite uses the `PluggableUI LoginContext`. Ensure that it is set up correctly.

Running the Test Servlet

If the previous steps did not help you to solve your issues with a web application using SIN for authentication and role management, install the *testjaas* web application.

▶ To verify the JAAS configuration using the *Testjaas* web application

1. Download *testjaas.war* from the Software AG Community Website > **Suite Downloads** at **http://techcommunity.softwareag.com/ecosystem/communities/public/webmethods/products/suite/downloads/**.
2. Install the *testjaas.war* in your *tomcat webapps*.
3. Point your browser to *http://yourhost:yourport/testjaas/testjaas* and save the output in a file. You can manually verify the working of the different `LoginContexts` by pointing your browser to *http://yourhost:yourport/testjaas/InputForm.html* and by providing the `LoginContext` and the logon credentials.
4. Save the output in a file.
5. Send the saved files to Software AG Customer Support.

If All Other Things Fail...

If things are still not working for you, send the following information to Software AG Customer Support:

- The *jaas_configuration.properties* file
- The output of the *log4j* that is set to DEBUG logging level for `com.softwareag.security`
- The output of the test servlet if this is applicable for your case.

5 Frequently Asked Questions

- What is SSX RMI Service? 50
- What Should I Know About the Modules' Configuration 52
- How Do I Use HTTPS with the XmlServerLoginModule 52

This chapter provides details on the additional functionality in SIN. Its usage is dependent on the specific security requirements of particular applications, so applying the described configurations is optional for some of the products.

The information is organized under the following headings:

What is SSX RMI Service?

This section details the configurations related to the implementation of RMI in SSX.

Using SSX RMI is optional. Its usage is prompted only in cases when you must invoke the methods of remote SIN objects from different hosts.

SSX RMI on Windows uses the Apache `prunsrv` service application to allow usage of SIN components for authentication as services on remote hosts. The SSX RMI service on UNIX uses the *sagrmisrvc* start script.

For more information on the `prunsrv` service, see ***http://commons.apache.org/daemon/procrun.html***.

The information is organized under the following headings:

- How to Start RMI on Windows
- How To Start RMI on UNIX

How to Start RMI on Windows

Check the *rmisrvc* folder inside SIN distribution for the following files:

1. *install_service.bat* installation script
2. *RmiService.config* file
3. *prunsrv.exe*
4. *prunmgr.exe*
5. Other required files

Following is a list of the required files:

- *sin-common.jar*
- *sin-ssx.jar*
- *log4j.jar*
- *sagssxuserdb2.dll*
- *sagssxuserdbimpl2.dll*

Note: For the correct version number, see *Readme.txt* in your SIN distribution.

To use SSX RMI on Windows

1. Put all required JAR and DLL files into one directory together with the *install_service.bat* installation script.
2. Execute the installation script.

   Note: Executing the script installs the server, but it is not started automatically.

3. Pass the URL of the sample configuration file *RmiService.config* to the service. The only parameter in this file is `port = 31415`. This is the port on which the RMI server is started.

   In the simplest case, the URL can point to a file in the same directory as the JAR and DLL files, but it can also be served centrally by an application server.

You can uninstall the service with `> prunsrv //DS//SAGRMI` and modify the parameters of the installed service with `> prunmgr //ES//SAGRMI`.

How To Start RMI on UNIX

Put the *sagrmisrvc* start script in the usual place for the relevant UNIX (or Linux) distribution.

For example, this will be */etc/rc.d/init.d* on most Linux systems.

Important: Consult the administration manual of your Unix OS for the right place for the*sagrmisrvc* file.

Check the *rmisrvc* folder inside SIN distribution for the following files:

1. The *sagrmisrvc* start script
2. The *rmisrvc.sh* shell script
3. The other required files

Following is a list of the required files:

- *rmisrvc.sh*
- *sin-common.jar*
- *sin-ssx.jar*
- *log4j.jar*
- *libsagssxuserdb2.so*
- *libsagssxuserdbimpl2.so*

Note: For the correct version number, see the *Readme.txt* in the SIN distribution.

To use SSX RMI on UNIX

Put all required files into the *RMISRVC_BASE* directory.

1. The start script runs the *rmisrvc.sh* shell script.
2. Set the environment variable `RMISRVC_BASE` in the *sagrmisrvc* script accordingly and take note of the comments concerning further settings.
3. Ensure that the `authdaemonPath` pointing to the *sagssxauthd2* executable in your SSX installation is set correctly in the *jaas_config.properties* and that its S-bit is set.

   Important: There is a shell script included in the SSX distribution that can be used to perform this step.

What Should I Know About the Modules' Configuration

Keep in mind the following warnings when setting up the JAAS configuration:

- Many programs expect one and only one `SagUserPrincipal` as the result of a successful authentication. However, a different expected behavior cannot be excluded.

  Ensure you configure the `LoginContexts` accordingly.
- Keeping the password in clear text in the `Subject.privateCredentials` may constitute a security risk, depending on how the `Subject` is handled. However, there are use cases where the password needs to be accessible through the `Subject`.

  Ensure you store the password only if needed.

How Do I Use HTTPS with the XmlServerLoginModule

The communication between the `XmlServerLoginModule` and the CentraSite server works also via HTTPS. This is a requirement if the calling program and the CentraSite server are not located on the same physical machine.

Following are the prerequisites for the HTTPS connection:

- The CentraSite Apache web server must be set up to provide an HTTPS port (set up `mod_ssl`).

- The installed server certificate that is to be used by the Apache web server must have the server name as the `subject DName`; for example, for a server that is accessed via *https://myserver.abc.com:53443*, the `Subject DName` must be `myserver.abc.com`.
- The `XMLSERVER_URL` must point to this HTTPS port
- There must be a trust anchor for the client to verify the server certificate. For this, preferably the issuer certificate of the server certificate must be made known to the client application.

  One way of doing this is to import this issuer certificate into the *cacerts* file. This file is located in the *jre/lib/security* directory of the Java installation.

6 Deprecated Login Modules

- CentraSiteServerLoginModule 56

Here is a list of deprecated login modules that are no longer used in Security Infrastructure.

CentraSiteServerLoginModule

This is a `Login Module` for retrieving user roles. It is responsible for the following operations:

- Authenticate a user with supplied credentials against CentraSite Server (Registry/Repository).
- Selecting the roles that are assigned to the authenticated `Principal` from an XML server.
- Adding these roles to the `Subject`.

With the `CentraSiteServerLoginModule` you can retrieve role information from the CentraSite Server. It creates the corresponding `RolePrincipal` objects and adds them in the `Subject`. The `CentraSiteServerLoginModule` requires a user name and a password for authentication. It also supports the usage of both an IAF token and an IAF artifact. The IAF artifact must be presented in `SagCredentials` object as a password, in the corresponding field.

- Parameters for Configuration
- Using SSL / HTTPS

Parameters for Configuration

Set *sin-common.jar*, *sin-xmlserver.jar*, and the Tamino API for Java (*TaminoAPI4J*) in the classpath.

Note: You must have a running CentraSite Server to be able to use the `CentraSiteServerLoginModule`.

Check the list with the parameters' description in the following table:

Parameter	Description	Default Value	Possible Values	Mandatory
xmlserver_url	The URL pointing to the CentraSite Server.	None	Any valid URL	Yes
useIAF	If the IAF artifact is present in the `SagCredentials`, the parameter specifies whether to use it. If set to "false", user name and password will be used for authentication against the XML server.	The default value is "true".	true false	No
usePasswordForIAF	This parameter gets the IAF artifact from the password field. **Note:** The IAF artifact is Base-64 encoded.	The default value is "false".	true false	No

Check the next task for the prerequisites to the secure communication between `CentraSiteServerLoginModule` and the application server.

Using SSL / HTTPS

The communication between the `CentraSiteServerLoginModule` and CentraSite server works via HTTPS. Following are prerequisites for the usage of HTTPS if the calling program and the CentraSite server are not located on the same physical system:

- Set up the CentraSite Apache web server to provide an HTTPS port (set up `mod_ssl`).
- The installed server certificate that is to be used by the Apache web server must have the same server name as the subject Distinguished Name (DN). For example, for a server that is accessed via *https://myserver.abc.com:53443*, the Subject DN has to be *myserver.abc.com*.
- The `xmlserver_url` has to point to this HTTPS port.
- There has to be a trust anchor for the client to verify the server certificate. To do this, the certificate of the issuer of the server certificate must be known to the client application.

  One way of doing this is to import the certificate of the issuer into the *cacerts* file. This file is located in the *jre/lib/security* directory of the Java installation that is used for the client program.

  Another way is to set your own trust store and add to it all trusted certificates.