

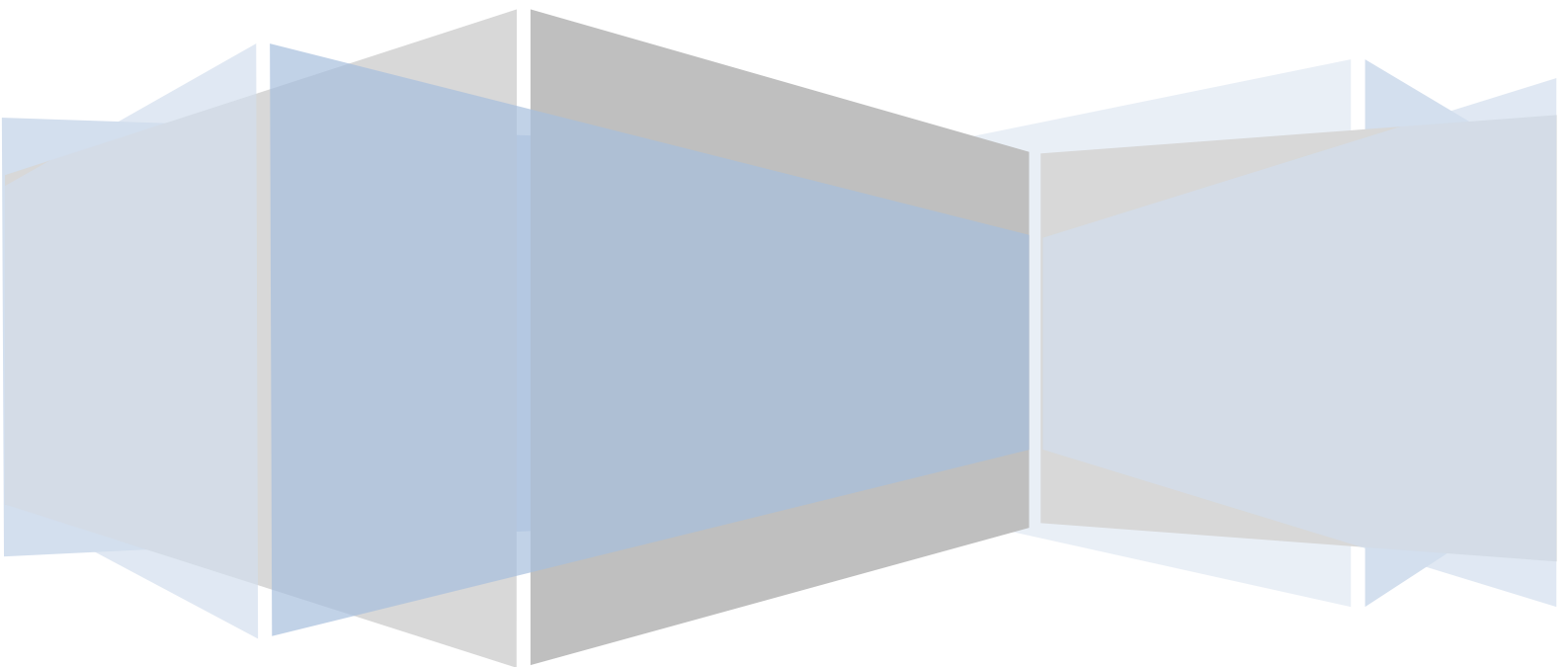
SSO Plugin

Installation for BMC Midtier

J System Solutions

<http://www.javasystemsolutions.com>

Version 3.4



Introduction.....	5
Installation.....	6
Copying the files to the Midtier.....	6
Logging.....	6
Matching the SSO username to an AR System User form entry.....	6
Most common configuration.....	7
Detailed overview of the username matching process.....	7
Alias username by User form query.....	7
Typical use case.....	8
User aliasing and Open ID.....	8
Automation.....	8
Automatically SSO enable accounts.....	8
Actions available when a user has no SSO account.....	8
Redirect user to login page.....	9
Redirect user to manual NTLM login page.....	9
Create account in User form.....	9
Raise an incident in BMC ITSM.....	9
Authentication methods.....	10
RSA Access Manager (ClearTrust) / Siteminder.....	10
Mixing RSA Access Manager, Siteminder or X509 with built-in AD or IIS.....	11
Integrating with an Active Directory.....	11
Creating the service account.....	11
Enabling large Kerberos token support in Tomcat.....	12
Running multiple Midtiers behind a load balancer.....	12
Windows Vista, 7, 2008 and AES 256 bit encryption.....	12
Built-in Kerberos/NTLM for Active Directory.....	13
Built-in Kerberos/NTLM, custom settings.....	13
Selecting supported protocols.....	13
Configuring NTLM.....	13
Advanced options.....	14
Configuring Kerberos.....	14
Advanced options.....	15
Manually creating a service account (Computer object).....	15
Manually configuring a Service Principal Name.....	15
Untrusted domains and the krb5.conf file.....	16
Identical service accounts.....	16
Creating a keytab.....	16
Ports and firewalls.....	17

Built-in Active Directory and load balancing / F5 / VIP.....	17
Using IIS and built-in authentication.....	17
Windows authentication performed by IIS.....	18
Configuring IIS.....	18
Configuring Tomcat.....	19
Large Kerberos tokens.....	19
Configuring SSO Plugin.....	19
Securing IIS and Tomcat.....	20
Open ID.....	20
Multiple Midtier configurations.....	20
How it works.....	21
X509 client certificates (CAC).....	21
Example SSL configuration.....	21
Configuring SSO Plugin.....	22
Installing SSO on BMC Remedy Knowledge Management.....	23
Automated installation.....	23
JSP patches.....	23
Manual installation.....	23
Manually logging in to RKM.....	24
What to do if you reconfigure the Midtier or SSO Plugin.....	24
Enabling RKM logging.....	24

Introduction

This document covers:

- Installation and configuration of SSO Plugin for BMC Midtier.
- Installation and configuration of SSO for BMC RKM 7.2/7.5.

Separate documents are available for other BMC components (ie AR System, Dashboards).

The JSS support website can be found here:

<http://www.javasystemsolutions.com/jss/support>

There's a video available to assist with installing the SSO Plugin and it can be found at

<http://www.javasystemsolutions.com/jss/video>

Installation

The installation process involves copying files to the Midtier and then selecting an authentication method, which in some cases, will involve further configuration of Tomcat.

Copying the files to the Midtier

To install the SSO Plugin on the Midtier, please follow these steps:

1. Copy the contents of the mt directory into the root Midtier directory. i.e. the contents of mt into the Midtier directory that *contains* the WEB-INF directory.
2. Restart Midtier.
3. If you are using IBM Websphere 7, use WAS to ensure the `com.ibm.ws.jsp.jdkSourceLevel` custom property is set to 14 or 15 on the web extension file or the custom WebContainer. This tells Websphere that the application was compiled for Java 1.5+.
4. Go to the SSO Plugin status page by pointing your browser at <http://path-to-midtier/arsys/jss-sso/index.jsp>. You will be presented with a status page. The password field in the left navigation is used to enable configuration and accepts the Midtier configuration password.
5. Review the [Authentication methods](#) section below to learn how to configure the required SSO implementation.
6. Click Configuration and enter the appropriate details. Press set configuration. If the Midtier is configured correctly then you may be advised to restart it again. Please take note of any errors and/or warnings that are displayed.
7. You can now test the SSO configuration by clicking on the Test SSO link in from the SSO Plugin status page. This will attempt to perform an SSO login to the authentication server and report any errors. If the test is successful when you can click on the Midtier Home link in the navigation and you should be taken directly to the Midtier Homepage without being asked to login.
8. If SSO fails then review the troubleshooting document or contact JSS support.

We have identified a possible bug in the AR System which will sometimes prevent our test facility working. It will manifest itself in a message stating that the Shared Key or IP Address is not correct. If this is the case, check this is still the case by visiting the Midtier Homepage. Please also review the information log in the SSO Administration Console within AR System.

Logging

This enables the Midtier SSO logging which writes to the Tomcat stdout file. We recommend you select Information for production use, debugging when configuring the SSO Plugin, and trace when you're trying to resolve an issue. Trace will generate a lot of logging including low level Kerberos debugging, and is required by JSS to resolve issues.

Matching the SSO username to an AR System User form entry

For SSO to work correctly, not only is an entry in the User form with a blank password required, but the case of the SSO username must match that of the entry in the User form. For many people, this won't be the case, and the SSO Plugin provides a range of functionality to resolve this problem.

Most common configuration

While the following may seem complicated, the most common configuration is:

1. Set Case sensitivity to match case insensitively.
2. Set User domain to Strip domain, because users don't tend to store domain names in the User form Login Name field.
3. Set the When user has no valid User form entry control to Redirect to login page.

Detailed overview of the username matching process

The following illustrates the process of deciding how to match an SSO user with the AR System User form and proceed to login:

1. If User domain is set to 'strip domain' is set then a Windows domain name will be stripped from the SSO username. If the username a distinguished name (DN) then this feature extracts the cn or uid value (ie X from "cn=X, ou=Y, o=Z.." or "uid=X, ou=Y, o=Z..").
2. If Case sensitivity is set to convert to upper or lower then username is modified.
3. If user aliasing is enabled, execute query against User form and if a user is returned, login with that user. If no match then fail.
4. If match case insensitively is selected then search for a user. If no match, continue.
5. If User domain is set to Try matching either way, search for a user entry with or without the Windows domain name. If no match then fail.
6. Test the User form to see if an entry exists with the Login Name set to the username (or domain name and username if User Domain is set to "User and domain"). If a valid SSO enabled entry exists then proceed to login, otherwise run the action selected when a user has no valid User form entry.

Alias username by User form query

This allows you to run a query against the User form to return a Login Name using the username (and optionally domain name) as part of the query.

When writing the query, you can use the following place holders which will be replaced with real values.

1. \$SSO_USER\$: The SSO username.
2. \$SSO_DNS_DOMAIN\$: The Windows DNS domain name (if using built-in AD integration).
3. \$SSO_NB_DOMAIN\$: The Windows NetBIOS domain name (if using built-in AD).

For example, user dkellest is logged into the Windows domain CORP (dns: corp.mydomain.com) then the values are as follows:

1. \$SSO_USER\$: dkellest
2. \$SSO_DNS_DOMAIN\$: corp.mydomain.com
3. \$SSO_NB_DOMAIN\$: CORP

If you want to pass the value returned from the Windows authentication system (i.e. user@domain or DOMAIN\user) to \$SSO_USER\$, do not enable *Remove domain part*.

Typical use case

1. If the User form holds the SSO usernames in field 526870912 then you may wish to set the alias query to '526870912' = "\$SSO_USER\$". When this query is executed against the User form, the \$SSO_USER\$ string is replaced with the username, and the value for the Login Name field is returned. This value is then used to connect to AR System.
2. If you have a policy of storing all SSO accounts in the format user@dnsdomain within field 526870912 on the User form, use the following query: '526870912' = "\$SSO_USER@\$SSO_DNS_DOMAIN\$".

Please note:

1. **Do not use field 117, known as the "authentication field"**. This has been reserved by BMC and AR System behaves differently when passing information to AREA plugin when this field is populated.
2. If there is no User form entry returned through user aliasing, the SSO authentication request is rejected. No user provisioning will take place.
3. The legacy \$SSO_DOMAIN\$ variable contains the NetBIOS domain.

User aliasing and Open ID

The user aliasing feature is used for the configuration of Open ID. When using Open ID, the \$SSO_DOMAIN\$ place holder is used to hold the Open ID Provider, and \$SSO_USER\$ is used for the Open ID Identifier.

Configuring Open ID is described in more detail in the Open ID section of the document.

Automation

This section contains features that improve on boarding when a user does have an account in the AR System User form.

Automatically SSO enable accounts

If a User account has a password set, it is not SSO enabled. This is one of the most common failure points during the SSO process: a user has a password assigned.

When a user has passed through the SSO process and has a password set in the User form, this option will remove the password, SSO enabling the account and avoiding a failed SSO login.

Please note, this does not mean the user can login manually with a blank password. Logins with a blank password are prohibited when SSO is enabled on the AR System.

Actions available when a user has no SSO account

The standard BMC SSO specification has no provision for users who are not SSO enabled within the AR System – i.e. if they don't have a correctly configured entry in the User form (no account, or an account with a password, etc.), the user is presented with an ARERR623 page. We don't believe this is desirable, so in this scenario, a number of actions can be performed.

The SSO Plugin provides a range of options to deal with the scenario where an SSO user has no valid entry in the User form. This forms an important part of integrating the SSO Plugin into a corporate environment where there are thousands of users, and many leaving/joining on a daily basis.

The configuration page contains a section marked “When SSO user has no valid User form entry” and the options are described below.

Redirect user to login page

Users who do not have an SSO enabled account in the User form will be redirected to the Midtier login form. This is the most obvious improvement to the user experience and requires no additional configuration.

Redirect user to manual NTLM login page

This alternative login page is available if SSO Plugin has been configured with Built-in authentication that includes NTLM. It allows a user to enter their **Windows** login details, which are validated against the Active Directory before allowing the login.

This functionality allows you to remove the BMC AREA LDAP plugin because, by doing so, you reduce the number of components to configure in AR System, and the SSO Plugin configuration is more comprehensive and easier than the BMC AREA LDAP plugin.

Typically, users may choose this option if they do not need to provide the ability for users to login to the Windows User Tool with their Windows credentials.

Create account in User form

When a user has no account in the User form, a new entry can be created from a template entry. When this option is selected, the name of an existing User form entry will be required to be used as a template for new entries. The new entry will be created by copying all fields from the template entry, replacing the login name with the SSO user name from the request.

If a user has an existing entry but it's not correctly configured then they will be redirected to the login page.

Raise an incident in BMC ITSM

If the user doesn't have an SSO enabled account, an incident will be raised and the user will be redirected to a page with the incident number. To configure this, you will need to enter the summary, notes, urgency and select the person ID to be associated with the new incident. The person ID must be verified before the setup is submitted.

When entering the notes and summary, you can use the `$$SSO_USER$`, `$$SSO_DNS_DOMAIN$` and `$$SSO_NB_DOMAIN$` place holders (as per user aliasing queries) to include the SSO credentials within the text.

Authentication methods

There are a number of ways to integrate the product into your network and each are described in detail in this document, however an overview follows:

1. Built-in Kerberos/NTLM for Active Directory, also known as Windows authentication without IIS.
2. Built-in Kerberos/NTLM, custom settings. This allows for advanced configurations when (1) is not suitable.
3. Windows authentication performed by IIS.
4. Using a third party authentication system such as RSA Access Manager (ClearTrust), Siteminder, or retrieving the username from a header/cookie.
5. OpenID, which requires two new fields in the User form and user aliasing configured.
6. Generic REMOTE_USER, a JAAS plugin or X509 client certificates.

When you enable built-in, IIS or OpenID authentication, the plugin will apply a patch to the Midtier web.xml file, adding the contents of the web.xml.patch (supplied with the product). If your Midtier can not write to the web.xml file, an error will be reported when the form is submitted. In this case, you will need to apply the patch manually (the instructions are in the web.xml.patch file).

Please note, most servlet engines (such as Tomcat) detect changes to the web.xml file and will restart the Midtier application. If the patch has been applied, you will need to restart Tomcat after submitting the form. Failure to do so will result in some odd errors as the Midtier can not 'survive' a restart without Tomcat being restarted.

Windows Authentication - implementation choices

If you're intending to implement Windows Authentication - so users can SSO into the Midtier *if they are logged into a Windows Domain* - then you need to decide whether to implement options 1 or 3.

If you don't have an IIS front end, use option 1.

If you've already got IIS installed as a front end to Midtier, use option 3 - please see the 'Configuring IIS and SSO Plugin' video on our website for an interactive installation guide.

RSA Access Manager (ClearTrust) / Siteminder

To enable these types of integrations, simply select the relevant entry in the web interface. We also recommend you install the vendor JAAS plugin.

Typically, this involves placing the vendor jar files in the Midtier WEB-INF/lib directory (or the Tomcat server classpath), modifying the Midtier web.xml file as per the vendor's instructions and mapping the JAAS plugin to the /arsys/home path.

RSA Access Manager and Siteminder have a timeout after which browser requests (to the Midtier) will be directed to the SSO login page. Therefore, the timeout needs to be in sync with the AR System Midtier session timeout or a situation can arise where a user is (still) logged into the Midtier but are not logged into the SSO environment. It is advised that the following points are kept in mind:

- Configure RSA Access Manager or Siteminder to protect the paths /arsys/home, /arsys/forms and /arsys/apps. These paths are the minimum required and a full set can be found in the web.xml.patch file (within the installation set), where each filter-mapping declaration defines a path to be protected.
- Ensure the Midtier and RSA Access Manager/Siteminder session timeouts are identical.

Mixing RSA Access Manager, Siteminder or X509 with built-in AD or IIS

You can mix an integration with RSA AM, Siteminder (or any other technology that uses a JAAS plugin) and/or X509 client certificates with built-in AD or IIS. To perform this integration, follow these steps:

1. Configure the built-in AD or IIS integration as per the documentation and test to ensure it works correctly.
2. Install the Siteminder/RSA AM JAAS plugin within the Midtier web application or configure Tomcat's SSL port to enable X509 client certificates.

The built-in AD and IIS filters will detect if a request has passed through and been validated by the vendor JAAS plugin or whether an X509 client certificate is present, skipping any further authentication if this is the case.

When performing this type of integration, you will typically need to consider how you map the Siteminder/RSA AM/X509 username and the built in AD or IIS username to the User form. This can be challenging so consider how it will be done in advance and consider enabling user aliasing functionality.

Integrating with an Active Directory

The product contains an implementation of the Microsoft Integrated Windows Authentication protocol. This allows users to open a browser, commonly IE, and navigate to the Midtier without being prompted to login.

There are a number of pre-requisites to this integration:

1. Creating the service account: This is a **computer** account in the Active Directory used by SSO Plugin to validate SSO tokens.
2. Enabling large Kerberos token support in Tomcat: Tomcat must be specifically configured to allow large Kerberos tokens to avoid some users receiving authentication errors.

The sections below provide an overview of these tasks, plus other useful information that should be reviewed before implementation.

Once you have reviewed this section, skip to either [Built-in Kerberos/NTLM for Active Directory](#).

Creating the service account

The script is called set-service-account.cmd and is included in the installation files. Copy it to your Active Directory, run it, and you can almost certainly accept the default options. It will create a computer called JSS-SSO-SERVICE - **note down the password it generates!**

The script also asks you for the hostnames on which the **Midtier Tomcat server** will be running - please provide both the hostname and the fully qualified hostname (i.e. myserver and myserver.domain.com) to the script.

If you do not wish to run our script, refer to the section [Manually creating a service account \(Computer object\)](#).

You must create a separate service account for each deployment of SSO Plugin. Name them appropriately, ie JSS-SSO-DEV, JSS-SSO-UAT, JSS-SSO-PRD1, JSS-SSO-PRD2, etc. This means you must create a service account for EVERY instance of a Midtier using a service account.

Enabling large Kerberos token support in Tomcat

Kerberos tokens are sent by the browser. By default, Tomcat has a hard coded limit of 4Kb for an HTTP header, and if the Kerberos token exceeds 4Kb then Tomcat returns status code 400 without passing the request to the Midtier. The standard BMC Tomcat distribution has been known to have 8Kb set, which is inadequate.

Open the Tomcat server.xml file (in the conf directory) and look for the HTTP connector:

```
<Connector port="8080" protocol="HTTP/1.1"
```

and add a maxHttpHeaderSize attribute, which is given a value in bytes (6500 is almost 64Kb):

```
<Connector port="8080" protocol="HTTP/1.1" maxHttpHeaderSize="65000"
```

Restart Tomcat and check the Midtier still works as expected.

Running multiple Midtiers behind a load balancer

If you are running multiple Midtiers, you require a separate service account for each instance. However, when creating the accounts, only one can have the Service Principal Names (required for Kerberos) associated with it.

Therefore, you must use "Built-in Kerberos/NTLM, custom settings" to specify the NTLM protocol to use the unique service account, and the Kerberos protocol to use the service account configured with the Service Principal Names.

For example, if you have three Midtiers (M1, M2, M3) in production then follow these steps:

1. Create three service accounts (JSS-SSO-P, JSS-SSO-P2, JSS-SSO-P3) and assign the SPNs (hostnames of the load balancer) to JSS-SSO-P.
2. On all Midtiers, set the Kerberos service account to JSS-SSO-P.
3. Set the NTLM service account to JSS-SSO-P on M1, JSS-SSO-P2 on M2 and JSS-SSO-P3 on M3.

If you are in any doubt then please assistance then please verify your configuration with JSS.

Windows Vista, 7, 2008 and AES 256 bit encryption

Without a patch from Sun, due to US export rules, the standard Java Virtual Machine does not support 256bit encryption and can not decode AES 256bit tokens. AES 256bit tokens are often generated by IE when using a Windows 2008 Domain Controller and a Windows Vista, 7 or 2008 client.

The SSO Plugin warns users if AES256 is not supported by displaying a message when the setup page is submitted, on the status page, and by writing a warning in the Tomcat logs when Midtier starts.

To enable AES 256bit support, you need to download and install the Sun "Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files", currently available from <http://java.sun.com/javase/downloads>.

Installing the patch is easy: You unzip two jar files and place them in the JRE lib/security directory.

For your information, the Java Security documentation includes the following explanation:

"The JCE framework within JDK includes an ability to enforce restrictions regarding the cryptographic algorithms and maximum cryptographic strengths available to applications. Such restrictions are specified in "jurisdiction policy files". The jurisdiction policy files bundled in Java SE limits the maximum key length. Hence, in order to use

AES256 encryption type, you will need to install the JCE crypto policy with the unlimited version to allow AES with 256-bit key."

Built-in Kerberos/NTLM for Active Directory

This integration requires just four pieces of information to configure both Kerberos and NTLM support. The integration requires a Computer account to be created in the AD and we supply a script to automate the process.

Please note, if you are using an IBM JDK then you must set up [Built-in Kerberos/NTLM, custom settings](#) and configure a krb5.conf file.

You are required to enter the following information:

1. Fully qualified DC hostname: This fully qualified hostname of the Active Directory/Domain Controller. You can provide a comma separated list of AD hostnames for failover support.
2. Realm/domain: This is the fully qualified name of the Windows domain.
3. Computer account name and password of the computer created from the script.

(1) Discovering the hostname of your Domain Controller

Open a command prompt and type:

```
echo %LOGONSERVER%
```

It will report the name with two backslashes, ie \\server2k3. You do not need the backslashes. The **fully qualified** hostname of the DC must be added, so look it up using nslookup:

```
nslookup adhostname
```

(2) Discovering the fully qualified domain

Open a command prompt and type:

```
net config workstation
```

The fully qualified domain name is printed by *Windows Domain DNS Name*.

Please note: While it is uncommon, the Windows Domain DNS Name is not correct. The value should start with the Windows Domain name, and if it does not, remove the string before this value. For example, if the Windows Domain DNS Name is ds.javasystemsolutions.local, and the Windows Domain is JAVASYSTEMSOLUTIONS, the correct value for SSO Plugin javasystemsolutions.local.

Built-in Kerberos/NTLM, custom settings

This is the advanced configuration for built-in Kerberos/NTLM. In most cases, you should be able to use Built-in Kerberos/NTLM for Active Directory.

Selecting supported protocols

The custom settings allow you to support one or both of the protocols. You should consider both Kerberos and NTLM in production. It's unlikely you will be able to allow Kerberos only due to NTLM being a common protocol on Windows networks, even when administrators have attempted to insist on Kerberos only.

Configuring NTLM

You are required to enter the following information:

1. DC hostname(s): This must be the fully qualified hostname of the Active Directory/Domain Controller. You can provide a comma separated list of AD hostnames for failover support.
2. Domain name: The name of the Windows domain – this is what you can see when you login to your machine.
3. Computer account name and password of the computer created from the script.

Press the set configuration button, restart the Midtier if prompted and go to the Test SSO page. If NTLM fails, you will get some indication of what's wrong – you can also review the Tomcat logs or contact JSS for assistance.

NTLM and multiple domains

If your domains are in a trusted relationship then you only need configure the SSO Plugin to authenticate against one of the domains. The Domain Controller should be able to authenticate users connecting from any other domain where that other domain is trusted. If the domains are in an untrusted relationship then we recommend you configure Kerberos.

Advanced options

There are a number of advanced options when enabling Permit NTLM:

- Fixed DC IP: In the unlikely event that the hostname of the Domain Controller does not resolve to an IP address, enter the IP address.
- DC fail timeout (secs): If there is a connection failure to the DC, it is marked as failed for the period of time defined in this field. The default, 30 seconds, should suffice but the option is configurable.

Configuring Kerberos

Kerberos requires two separate elements to the configuration, each of which can be configured in two ways (providing four possible ways to configure the product!):

1. The location of the Kerberos Domain Controller (KDC). This is configured by providing the hostname of the KDC and a Kerberos realm, or by configuring a [krb5.conf](#) file (please note, when using an IBM JDK, you must set up the krb5.conf file).
2. A mechanism of authenticating with the KDC. This is configured by providing service account credentials (as required for NTLM), or a keytab file created using the [ktpass](#) program.

We recommend you use with the easiest configuration: a service account, a hostname of the KDC and a Kerberos realm. *This configuration is the easiest for end users, and in many cases, no further configuration will be required.*

Whatever configuration you choose, you must configure a Service Principal Name (SPN) or the browsers will not send a Kerberos token to the Midtier. The set-service-account.cmd script (described above) will create the SPNs and this can also be performed manually (see below) – we recommend you use the script.

Assuming you use the easiest configuration you are required to enter the following information:

1. KDC address: Enter the fully qualified hostname of your Kerberos Domain Controller, which is probably the hostname of the Active Directory.
2. Kerberos realm: This is **not the Windows Domain**, but usually the fully qualified Windows domain. You can out this value by opening a command prompt and typing:

```
net config workstation
```

and looking for the value of *Windows Domain DNS Name*. If in doubt, contact your network administrator as Kerberos **will not work** without the correct Kerberos realm.

3. Computer account name and password of the computer created from the script.

Advanced options

There are a number of advanced options when enabling Permit Kerberos:

- Use ticket cache: This is an internal feature of the Kerberos module and you should not enable it without JSS advice.

Manually creating a service account (Computer object)

If you don't want to run our `set-service-account.cmd` script then you will need to configure the Active Directory service account manually. Please follow these steps:

1. Using the Active Directory Users and Computes tool, create a new **computer** object called JSS-SSO-SERVICE. **If you choose any other name, it must be in upper case.**
2. The AD tool provides no way to set the password on the computer account. There are two ways to do this:
 1. Use the `netdom` command in the Microsoft Support tools. The syntax is as follows (/pd:* means, prompt for password):

```
netdom resetpwd /s:domaincontroller /ud:DOMAIN\JSS-SSO-SERVICE /pd: *
```

2. A small script can be used to set the password. Please edit the LDAP path appropriately and run from a command prompt:

```
echo GetObject('LDAP://CN=JSS-SSO-SERVICE,CN=Computers,DC=testdomain,DC=local').setPassword('new-password'); > temp.js
cscript //E:jscript temp.js
del temp.js
```

3. Create the service principal name (see below).

Manually configuring a Service Principal Name

We recommend you use the `set-service-account.cmd` script provided to setup the Service Principal Names (SPN).

For the Midtier to be able to authenticate clients using Kerberos, an SPN must be configured on the Domain Controller. The `setspn.exe` tool is used by the administrators to create an SPN which maps the Midtier host to a service account in the Active Directory.

To find out the fully qualified hostname of the Active Directory, ping it from the command prompt (you will see the hostname and `sudo apt-get update` fully qualified hostname).

We assume that:

- The Windows domain is called DEVELOPMENT,
- The domain's fully qualified name is development.javasystemsolutions.com,
- The Midtier is running on a machine with the hostname midtier.javasystemsolutions.com,
- The service account username is JSS-SSO-SERVICE.

Here is an example of how to use `setspn` - **you must add both the hostname and the fully qualified hostname of the Midtier!**

```
setspn.exe -A HTTP/midtier.javasystemsolutions.com DEVELOPMENT\JSS-SSO-SERVICE
setspn.exe -A HTTP/midtier DEVELOPMENT\JSS-SSO-SERVICE
```

You can check to see if the SPN has been added by using the -L option, which lists the SPNs for a computer or user account:

```
setspn.exe -L DEVELOPMENT\JSS-SSO-SERVICE
```

Please note, a hostname should only ever be declared against one user account - to declare it against multiple users will confuse Active Directory.

Untrusted domains and the krb5.conf file

Users with multiple Windows Domains that are in an **untrusted** relationship will need to configure a krb5.conf file (an example is provided with the installation). If your Domain Controllers are in a trusted relationship then the KDC for domain A should be able to authenticate users for domain B, and vice versa, so the krb5.conf isn't required unless you require some of the advanced Kerberos configuration options.

If you need to use a krb5.conf file, there are two ways in which the product can connect to the Kerberos Domain Controller: Using identical service accounts in each Kerberos Domain or through a keytab.

Identical service accounts

Only one service account name and password can be configured with Kerberos, hence keytabs are popular as they allow different accounts to be configured for each domain. However, if you create the same account in each domain, where each account has the same password, you can connect to each KDC with the service account and remove the need for a keytab.

Creating a keytab

This is an advanced configuration: we recommend you use a service account.

Users who do not wish to store service account credentials with the Midtier can use a keytab. This is created with the ktpass program, and plenty of examples are available online, however it is briefly covered below.

Following on from the SPN example above, a keytab can then be created as follows:

```
ktpass -princ
HTTP/midtier.javasystemsolutions.com@DEVELOPMENT.JAVASYSTEMSOLUTIONS
.COM -out JSS-SSO-SERVICE.keytab -mapuser JSS-SSO-SERVICE -pass
service_account_password -ptype KRB5_NT_PRINCIPAL -crypto RC4-HMAC-
NT
```

(Note, the realm - DEVELOPMENT.JAVASYSTEMSOLUTIONS.COM - has to be in upper case.)

Using the above configuration, you would store the keytab in the Midtier (we recommend under WEB-INF) and configure the SSO Plugin by providing:

- The full path to the keytab.
- The service principal, which is [HTTP/midtier.javasystemsolutions.com@DEVELOPMENT.JAVASYSTEMSOLUTIONS.COM](http://www.javasystemsolutions.com)

(You do not have to use such a long SPN name!)

Ports and firewalls

The NetBIOS protocol makes use of TCP ports 135-139, 445 and ports above 1024 (source <http://support.microsoft.com/kb/832017>) therefore your Midtier must have access to these ports on the Domain Controller. If you've got a firewall between the Midtier and Windows Domain Controller, ensure the ports are open.

Built-in Active Directory and load balancing / F5 / VIP

When using a load balancer / F5 / VIP with a group of web servers, there are extra steps to configure Kerberos through the load balancer hostname.

Kerberos relies on a Service Principal Name being present in the Active Directory, mapping a hostname to a service account. Setting up SPNs has been documented above but the golden rule is as follows: An SPN for a hostname can only exist once; if it exists more than once, it is ignored.

In a situation where there are multiple web servers, each with a service account, it is impossible to set up an SPN for a load balancer hostname against each service account, Assume JSS-SSO-P1/2 are two computer service accounts that are configured with two web servers, the following is invalid:

```
setspn -A HTTP/lbhostname.mydomain.com JSS-SSO-P1
setspn -A HTTP/lbhostname.mydomain.com JSS-SSO-P2
```

The solution is to create a separate service account for Kerberos only, and configure Kerberos independently of NTLM on each web server. The Kerberos account can be a normal user account and assuming it is called JSS-SSO-KERB, the Active Directory administrator can enable Kerberos as follows:

```
setspn -A HTTP/lbhostname.mydomain.com JSS-SSO-KERB
```

To re-configure each web server, select built-in Kerberos/NTLM, leave the NTLM set up as is (ie JSS-SSO-P1 on one web server, and JSS-SSO-P2 on the other) and configure both with the JSS-SSO-KERB service account in the Kerberos setup.

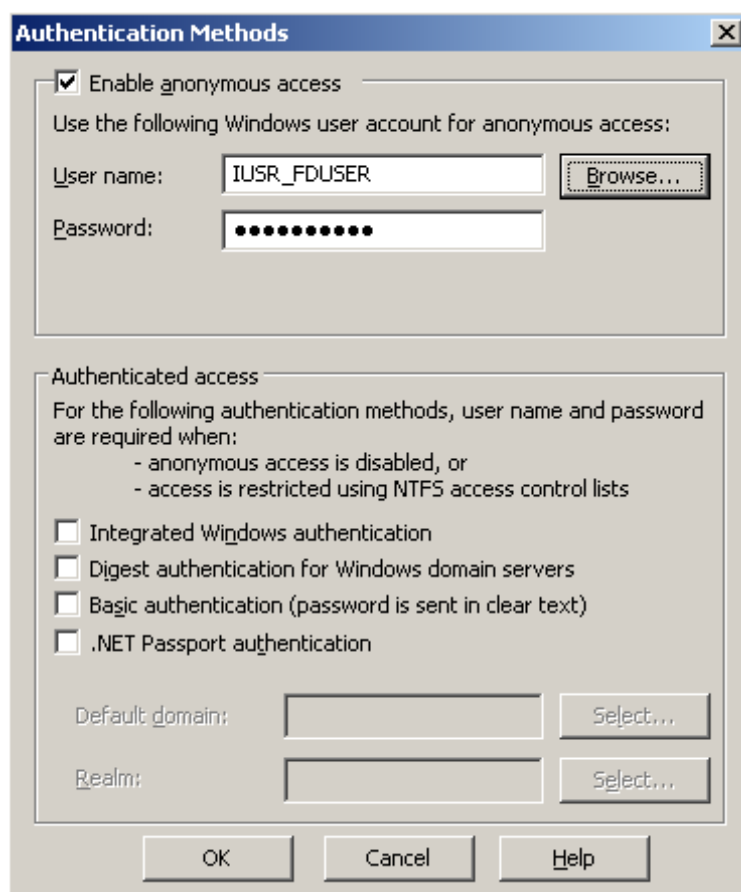
Using IIS and built-in authentication

If you require an IIS front end then we recommend you do not use built-in authentication and instead use Windows authentication performed by IIS.

If you're using a single Tomcat instance, and are not engaging in software load balancing, then you don't need to use an IIS front end with built-in authentication. The BMC Midtier installer will configure IIS if it's present, and while we do not recommend this configuration, it is possible to use Internal Windows Authentication with IIS.

In order to do this, you must ensure IIS is not configured to perform any authentication. This is done by configuring the IIS website authentication to anonymous only:

1. Open the Windows Control Panel.
2. Open Administrative Tools.
3. Open the IIS management console.
4. Locate Websites → Default website → jakarta, right click and select Properties.
5. Locate the Directory Security tab and click Edit in 'Authentication and Access Control'.
6. Ensure 'Enable anonymous access' is checked, and the 'Authenticated access' check boxes are unchecked. The following dialog box shows the configuration:



Windows authentication performed by IIS

Please do not run the `set-service-account.cmd` script if you're using an IIS front end - it's not required and may lead to IIS being unable to authenticate users.

There are a number of steps to perform in order to correctly configure Windows authentication performed by IIS that involve changes to IIS, Tomcat's `server.xml` file and the `workers.properties` file that configures `mod_jk`, the software that connects IIS to Tomcat. These changes are detailed below.

Configuring IIS

IIS must be configured to perform Integrated Windows Authentication (IWA) and anonymous. The anonymous access is required for WUT Data Visualisation fields which are displayed in an IE component that can not perform IWA.

Make the changes to IIS as follows:

1. Open the Windows Control Panel.
2. Open Administrative Tools.
3. Open the IIS management console.
4. Locate Websites → Default website → jakarta, right click and select Properties.
5. Locate the Directory Security tab and click Edit in 'Authentication and Access Control'.
6. Ensure 'Enable Anonymous Authentication' and 'Integrated Windows Authentication' are checked.

You need to have service principal names (SPNs) setup on the Active Directory for the hostnames running Midtier if you are not using the hostname of the computer. Your AD administrator can create them as follows:

```
setspn -A HTTP/hostname domain\computer
setspn -A HTTP/fully.qualified.hostname domain\computer
```

Configuring Tomcat

To tell Tomcat that IIS is performing Windows Authentication, locate the Tomcat server.xml file, which will be in the Tomcat conf directory. Locate the ajp/13 connector, which looks like this:

```
<Connector port="8009" protocol="AJP/1.3" redirectPort="8443" />
```

and add the following attribute:

```
tomcatAuthentication="false"
```

In this example, it would now look like this:

```
<Connector port="8009" tomcatAuthentication="false" ...
```

Large Kerberos tokens

IE clients can send very large Kerberos tokens which can be too big to be passed between IIS and Tomcat through the mod_jk connector (this is the software that connects the two systems). This will cause browser issues and often only on some machines (as Kerberos tokens contain group information, so if a user is in many groups, the token is likely to be larger than a user who is not).

To rectify this, two files must be modified:

1. The mod_jk workers.properties. You will need to search for this file as it could be in many locations, but is often found near the Apache Tomcat installation (if the BMC installer has been used). Open the file and add the line:

```
worker.X.max_packet_size=16000
```

where X is the name of the worker - you will see many other similar lines from which to copy and edit. The BMC installer sometimes adds this line for you, so if that is the case then set the value to 16000 and ensure you carry out step 2.

2. Locate the Tomcat server.xml file, which will be in the Tomcat conf directory. Locate the ajp/13 'Connector, which looks like this:

```
<Connector port="8009" protocol="AJP/1.3" redirectPort="8443" />
```

and add the following attribute:

```
packetSize="16000"
```

In this example, it would now look like this:

```
<Connector port="8009" packetSize="16000" ...
```

Configuring SSO Plugin

Select the authentication method Windows authentication performed by IIS.

This operation will result in a patch being applied to the Midtier web.xml file (as is the case with built-in authentication). If the web.xml file is patched, a warning message will be displayed when you submit the setup form and you **must restart the Midtier!**

Securing IIS and Tomcat

It is important to secure Tomcat once IIS integration has been completed. This is achieved by turning off the Tomcat HTTP Connector (so all requests must go through IIS) and setting a secret on the Tomcat's AJP connector so it is only accessible by the IIS instance configured.

Turning off Tomcat's AJP connector

Open the server.xml file, look for the following and comment it out by surrounding with `<!--` and `-->` (it is usually located directly above the AJP connector modified above):

```
<Connector port="8080" ...
```

Setting a secret on the AJP connector

Open the workers.properties file and add a secret to the worker:

```
worker.X.secret=mysecretkey
```

Open the Tomcat server.xml file and add this attribute to the AJP connector:

```
<Connector port="8009" packetSize="16000"
request.secret="mysecretkey"
```

Restart both IIS and Tomcat when complete.

Open ID

The SSO Plugin can integrate with Open ID Providers such as Google, Yahoo, MyOpenID, ClaimID, etc. Open ID requires two pieces of information - the Open ID provider and identifier. Please see <http://en.wikipedia.org/wiki/OpenID> for an overview of Open ID.

To configure Open ID, follow these steps:

1. Go to the BMC AR System Developer Studio, open the User form and add two character fields called 'Open ID Provider' (456) and 'Open ID Identifier' (123) - we have included sample field IDs in brackets for the purposes of this guide, you will need to note down the ones assigned by Developer Studio. Arrange the new fields neatly on the User form.
2. Go to the SSO Plugin setup page and select the OpenID authentication method.
3. Select Strip domain in the User domain control.
4. Select Match case-sensitively in the Case sensitivity control.
5. Enable 'Alias username by User form query' and enter the following into 'User matching condition':

```
'123' = "$SSO_DOMAIN$" AND '456' = "$SSO_USER$"
```

6. Submit the configuration.
7. Restart the Midtier if prompted.

Open a new browser, go to the Test SSO page and you will be presented with an Open ID login form. Select your Open ID provider, type in your Open ID and submit the login form. The browser will be redirected to the Open ID provider's login page, and after logging in, the browser will return to the Test SSO page.

If no entry exists in the User form with the correct Open ID Provider and Identifier, they will be provided so you can edit a User entry (don't forget to set a blank password). If you return to the Test SSO page then you should now see the user that has just been activated.

Multiple Midtier configurations

The Midtier configuration is held in the AR System database and hence is shared between all Midtier instances using the same authentication server.

If you require specific Midtier configurations, ie when using built-in authentication and a service account which should be different for each Midtier, the SSO Plugin can create Midtier specific configurations.

Another common requirement is if an MSP wishes to install a Midtier on each customer site, each of which requires a different configuration, all of which connect to a common group of AR System servers hosted on the MSP's own network.

The SSO Plugin supports this functionality via the "Shared configuration" checkbox that is located under the 'Set configuration' button in the setup page. Leave the checkbox unchecked for local configurations.

The local configuration is stored in AR System using the Midtier hostname plus a unique key generated by SSO Plugin and stored in the Midtier config.properties file.

How it works

The configuration is managed by SSO Plugin so you should not have to perform any manual steps in AR System, however we appreciate that administrators may wish to understand how the configuration is stored.

The Midtier configuration is stored in the J System Solutions Midtier Properties form within AR System. There is always one entry that contains the global settings (if upgrading from version 3.2, it is created the first time the configuration is saved), and additional configurations are stored by creating additional entries with one field set to the Midtier server hostname.

X509 client certificates (CAC)

The Midtier SSO Plugin supports X509 client certificates if your webserver has been configured with an SSL connector configured to use a server side certificate. This subject can be challenging and there are many online tutorials, so this document provides a brief guide. JSS support would be happy to help you configure this integration.

Typically, your company will have a server side certificate and you'd configure Tomcat's SSL connector to load the key from a Java Key Store (.jks) file. You will also have a client side certificate (a .p12 file) to load into your browser's certificates.

Please note, X509 is also used by DoD Common Access Cards.

Example SSL configuration

For the purposes of demonstrating the functionality, you can generate your own and we've provided a script called generate-example-client-cert.bat to do this for you. It will create a fake key store and client side certificate. **This script is only for the purposes of demonstration - if your business wishes to use SSL, you should not have to generate these files.**

If you run the script and pass a username, a certificate will be created with the subject:

```
CN=[user],OU=jss,O=office,L=mk,S=bucks,C=uk
```

You can change this subject by editing the script.

The script generates two files:

1. user.p12, which should be loaded into your browser.
 1. On Firefox, go to Tools → Options → Advanced → Encryption → View Certificates → Your certificates → press import. Select the file and when prompted for a password, enter password.
 2. On IE, go to Tools → Internet Options → Content → Certificates → Personal → press import, browse to the file → select .p12 in the file types drop down, enter password and press next, next, finish.

2. server.jks, which is placed in the Tomcat conf directory.

To complete the setup, you must enable Tomcat's SSL port by opening the server.xml and adding this connector:

```
<Connector
  clientAuth="true" port="8443" minSpareThreads="5"
  maxSpareThreads="75"
  enableLookups="true" disableUploadTimeout="true"
  acceptCount="100" maxThreads="200"
  scheme="https" secure="true" SSLEnabled="true"
  keystoreFile="${catalina.home}/conf/server.jks"
  keystoreType="JKS" keystorePass="password"
  truststoreFile="${catalina.home}/conf/server.jks"
  truststoreType="JKS" truststorePass="password"
  SSLVerifyClient="require" SSLEngine="on" SSLVerifyDepth="2"
  sslProtocol="TLS"
/>
```

Configuring SSO Plugin

To look for an X509 client certificate, select the appropriate integration type (generic REMOTE_USER, JAAS and X509). You can now choose a number of ways to map a subject back to the User form.

You could choose to implement user aliasing: insert a new field on the User form, enter the certificate subject against a user and let the SSO Plugin match the record when that user accesses Midtier.

You could use the 'strip domain' feature which, in this case, extracts the cn or uid value (ie X from "cn=X, ou=Y, o=Z.." or "uid=X, ou=Y, o=Z..") and matches that against a User form Login Name, or invoke user aliasing.

Installing SSO on BMC Remedy Knowledge Management

The SSO Plugin has been extended to provide SSO for Remedy Knowledge Management (RKM). There is no official SSO interface to RKM, so JSS have developed their own 'hook' into the application in order to trap requests to the RKM homepage and pass them through the SSO Plugin.

Automated installation

The SSO Plugin Midtier interface is able to automatically patch RKM and this can be done through the Midtier SSO status page. However, this will only proceed if the plugin can correctly find the RKM installation.

You must also follow the steps in 'JSP Patches', below.

JSP patches

Whether using the automated or manual installation process, you will also need to follow these steps which are difficult to automate but only need to be performed once:

1. Locate the `getAssignTo.jsp` file within the RKM installation files (ie `C:\Program Files\Apache Software Foundation\Tomcat 5.5\webapps\rkm`), open it and find the following line of code:

```
if (HomeFinder.getDefault().getAppConfig().isRemedyAuthentication())
```

and add the code highlighted in bold below:

```
if (HomeFinder.getDefault().getAppConfig().isNoAuthentication() ||  
HomeFinder.getDefault().getAppConfig().isRemedyAuthentication())
```

Manual installation

The installation involves a couple of changes to the RKM configuration and copying files from a correctly configured Midtier.

Please do not proceed to install SSO on RKM until you are completely satisfied with the SSO configuration for the Midtier.

The installation steps are as follows:

1. Copy the `WEB-INF/lib/jss-sso.jar` file from the Midtier into the same location in the rkm installation. **For example, if your Midtier installation is located at `c:\Program Files\BMC\Midtier` and RKM is installed at `c:\Program Files\BMC\RKM` then for step 1, copy `c:\Program Files\BMC\Midtier\WEB-INF\lib\jss-sso.jar` to `c:\Program Files\BMC\RKM\WEB-INF\lib`.**
2. Copy the relevant `jss-sso-rkm-X.jar` from the `jss-sso/rkm` within the Midtier to the RKM `WEB-INF/lib` directory, where X is 7.2 if RKM 7.2, or 7.5 for RKM 7.5 and above..
3. Locate the following file from the RKM installation:

`WEB-INF/classes/kms/authenticators/SimpleAuthenticator.class`

Rename it by appending `.old` to the filename.

4. Go to the RKM configuration page, i.e. `http://hostname:8080/rkm/configuration`
5. Select Authentication on the Configuration Settings page.
6. Set the Mode to None. This may seem odd but JSS have used the 'None' authentication layer within RKM to implement SSO.
7. Click save.

8. This step is only for those using Windows Authentication. Find the web.xml file in the rkm installation (it's in the WEB-INF directory) and make a backup of it. Open up the original web.xml file and the web.xml.rkm.patch file that's located in the jss-ss0/rkm directory within the Midtier. Copy the contents of the patch file and paste it into the web.xml file at the point below the last </filter> element and before the following <filter-mapping> element. i.e. The non-italic text below is in the web.xml, and the patch is placed at the point highlighted.

```
<filter>
  <filter-name>SystemFilter</filter-name>
  <filter-class>kms.filters.SystemFilter</filter-class>
</filter>
```

The patch goes here.

```
<!-- To use non XDoclet filter-mappings, create a filter-
mappings.xml file that ...
```

9. You must also follow the steps in 'JSP Patches', above.
10. Restart the webserver running RKM.

To integrate with your SSO system, simply point the browser at the usual homepage RKM link, i.e. <http://hostname:8080/rkm/home.jsp>.

Manually logging in to RKM

The SSO Plugin extends the original Remedy authentication scheme and delegates authentication to it in the event an SSO login can not be performed. Therefore, existing manual login functionality is still available.

What to do if you reconfigure the Midtier or SSO Plugin

The Midtier configuration is held in the AR System so restarting RKM will result in the new configuration being loaded.

Enabling RKM logging

If there's a problem then you will need to send JSS the webserver standard out log files (i.e. the Tomcat stdout.log or catalina.out file, in the logs directory). RKM may not be configured to write all log messages to the log file, so to ensure they are, follow these steps:

1. Go to the RKM configuration page, i.e. <http://hostname:8080/rkm/configuration>
2. Select General on the Configuration Settings page.
3. Click the Log Level and set to Debug.
4. Click save.
5. Restart the webserver running RKM.