KERNEL AUTHENTICATION & AUTHORIZATION FOR J2EE (KAAJEE) VERSION 8.0.749

FOR WEBLOGIC (WL) VERSIONS 12.2 AND HIGHER

RELEASE NOTES

March 2022

This page is left blank intentionally.

Contents

March 2022   Kernel Authentication and Authorization Java (2) Enterprise Edition (KAAJEE)   iii
Release Notes
Version 8.749 on WebLogic 12.2 and higher

This page is left blank intentionally.

1. Introduction

1.1. Orientation

Going forward significant KAAJEE Classic, KAAJEE SSOWAP and KAAJEE Security Provider release objects (JARs, EARs, WARs, ZIP) will correspond to a patch denomination in FORUM (e.g., in "kaajee-8.0.749.jar" where "8.0.749" is the patch number).

1.2. About KAAJEE, KAAJEE Classic

Kernel Authentication & Authorization for Java 2 Enterprise Edition (KAAJEE) for Web-based Health*e*Vet applications on WebLogic 12.2 and higher, such as Pharmacy Re-Engineering (PRE), provides user authentication to grant access to applications, and retrieves VistA security keys for application authorization. KAAJEE takes advantage of the current user store in VistA to authenticate users in new Health*e*Vet applications. Considered an interim solution, KAAJEE will be enhanced to provide the connection to the Department of Veterans Affairs (VA) Enterprise user authentication (sign-on) system in the future. KAAJEE and Fat-client Kernel Authentication and Authorization (FatKAAT) use VistALink as a connectivity solution from Java 2 Platforms, Enterprise Edition (J2EE) applications to Mumps (M)/VistA. WebLogic Updates Project.

KAAJEE 1.2 is the successor to KAAJEE 1.1, which was originally released to support the BEA WebLogic Server 9.2 platform. Version 1.2 of KAAJEE now supports Oracle WebLogic Server 10.3.6 and higher, and is fully Fortify and TRM compliant.

Like KAAJEE 1.2, KAAJEE Classic 8.0.749 provides a custom Authentication Provider "plug-in" for BEA WebLogic J2EE servers, including a set of web forms for web applications that allow web applications to authenticate and authorize an end-user using a Kernel system as the source of authentication and authorization. A form is presented upon protected resource request, gathering user A/V codes and presenting a list of pre-determined stations. Such functionality will be referred to as "KAAJEE Classic", as to denote original implementation of a complete authentication process tied to VistA. Since VA has moved towards two-factory authentication mechanisms, a successor in KAAJEE family has sprung – Single Sign-On Web Application Plugin, SSOWAP. It is important to denote the differences between KAAJEE Classic and KAAJEE SSOWAP, notwithstanding User Interface similarities, the authentication mechanisms and deployment requirements are very different.

1.3. Distribution

KAAJEE Classic 8.0.749 implements all current features of KAAJEE 1.2, except for those related to the persistent cookie handling and support. Included in the distribution are the following:

- KAAJEE 8.0.749 jar file and its dependencies
- Folder of Login web forms for Forms Authentication to drop into a J2EE Web application
- JavaDoc Application Programming Interface (API) documentation for the user manager classes
- Deployment Descriptor examples
- Complete source code for the KAAJEE Classic implementation
- Sample Web Application

/kaajee-8.0.749 contains the following sub-folders:

- /dd_examples -- This folder contains sample deployment descriptors.
- /doc -- This folder contains a readme.txt file.
- /jars -- This folder contains the KAAJEE JAR file and related JSP files.
- /javadoc -- This folder contains the KAAJEE javadocs.
- /samples -- This folder contains the KAAJEE Sample Web application and its dependencies
- /samples
-     /sso-ccow -- This folder contains elements to make the KAAJEE Sample Web application Single Sign-On (SSO CCOW) enabled.
- /src -- This folder contains the KAAJEE Java source code.

1.4. For More Information

For more information on any new feature discussed in the Release Notes, please see the KAAJEE documentation set.

This page is left blank intentionally.

2. New Features / Changes for KAAJEE Version 8.0.749

2.1. Upgraded Deployment Descriptors:

All KAAJEE deployment descriptors have been upgraded as recommended by the WebLogic 12.2 and higher documentation. It is recommended that consuming applications do the same.

For KAAJEE, the sample WebLogic deployment descriptors were changed from referencing DTD based document to a Schema based document.

KAAJEE product line is fully Fortify and TRM compliant. All the dependencies have been updated with the latest TRM-approved versions.

Persistent cookie handling and support has been removed from this release.

2.2. Administrator Security Role for LoginController

The security role assignment has been upgraded in both web.xml and weblogic.xml. There is no more "magic" AUTHENTICATED_KAAJEE_USER role. Change the weblogic.xml file in the KAAJEE server application to use KAAJEE in the run-as-principal-name element. It is important that the "KAAJEE" user account (principal) has administrative privileges. If a different principal is used for this purpose, the weblogic.xml file will have to reflect this change.

However, newer version of WebLogic (12.2) requires a configuration change in order to enable programmatic creation and removal of the users. Please see a new section in the Deployment Guide for details on Configuring Administrative User

The format of identifying the cookie name in the <session-descriptor> in weblogic.xml has changed for WebLogic 12.2. The changes to each document are as follows:

- web.xml:

```
<servlet>
  <servlet-name>LoginController</servlet-name>
  <servlet-
class>gov.va.med.authentication.kernel.servlet.LoginController
  </servlet-class>
  <run-as>
    <role-name>adminuserrole</role-name>
  </run-as>
</servlet>

<security-role>
  <role-name>adminuserrole</role-name>
</security-role>
```

- weblogic.xml:

```
<run-as-role-assignment>
```

```
<role-name>adminuserrole</role-name>
<run-as-principal-name>KAAJEE</run-as-principal-name>
</run-as-role-assignment>

<session-descriptor>
  <cookie-name>kaajeeJSESSIONID</cookie-name>
</session-descriptor>
```

2.3. Sign-On/User Context (SSO/UC)-Enabled

Developers now have the option to make KAAJEE-based applications Single Sign-On/User Context (SSO/UC)-enabled. See the KAAJEE Deployment Guide for the details on how to enable your application. The current implementation of single sign-on is based on the user context of the Clinical Context Object Workgroup (CCOW) standard.

KAAJEE architecture will now allow users to authenticate and sign on to multiple applications that are CCOW-enabled and Single Sign-On/User Context (SSO/UC)-aware using a single set of credentials, which will reduce the need for multiple IDs and passwords.

There is an SSO two-factor authentication implementation - SSO Web Application Plugin– KAAJEE SSOWAP. It is a separate distribution and is not part of this package.

2.4. Section 508 Issue Regarding Session Timeouts

KAAJEE now displays an alert dialogue box, warning the end-user how much time remains before the login session expires. This warning is displayed 30 seconds prior to the expiration of the user's login session. In order to provide this warning, KAAJEE utilizes JavaScript. Therefore, KAAJEE distributes a login.js file, which is exported as part of the login\javascript\ folder.

2.5. Dependency Upgrades

* WebLogic: KAAJEE is now dependent on WebLogic 12.2 and higher.
* VistALink: KAAJEE is now dependent on VistaLink 1.6.7
* Java: KAAJEE is now dependent on Java 1.8

This page is left blank intentionally.