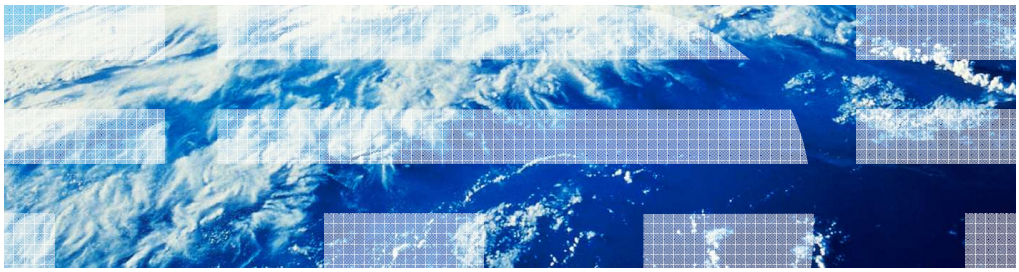
IBM WebSphere Application Server V8.5.0.2 Liberty Profile

Thread identity synchronization for J2C connectors and Java EE web applications



You can synchronize application server and z/OS thread identities for J2C connectors and Java EE web applications in the IBM WebSphere® Application Server V8.5.0.2 Liberty Profile.

Thread identity synchronization

- Security feature for managing application access to native resources
  - Example: file systems, databases
- "Syncs" the Java EE runAs identity with the native thread
  - Native resources are accessed using the security credentials of the runAs identity instead of the server identity
  - RunAs identity = the authenticated identity of the user
- Available only on the z/OS® platform
- Ported to Liberty profile from WebSphere Application Server full profile
- Configured in server.xml with the syncToOSThread element

Thread identity synchronization lets you manage application access to native resources (for example a file in the file system, a network socket, or a connection to a co-located subsystem like DB2®). You can control which security credentials are used to access these resources.

Under normal operating conditions, all native resources are accessed using the security credentials and privileges of the server process -- or more specifically, the identity associated with the server process. For example, if an application attempts to read a file from the native file system, the server's identity must have sufficient read permission to that file in order for the application to read it.

With thread identity synchronization, native resources accessed by the application are accessed using the security credentials and privileges of the JAVA EE runAs identity, not the server's identity. The Java EE runAs identity is the authenticated identity of the logged-in user.

Considering the file example again, if an application attempts to read a file from the native file system while thread identity synchronization is enabled, then the runAs identity (in other words, the end user) must have sufficient read permission to that file in order for the application to read it.

As a Liberty Profile administrator, you can control which native resources an application is permitted to access on behalf of the end user. The feature is available on the z/OS platform only. The feature has been available for WebSphere Application Server Full Profile since version 5. As of 8.5.0.2, it is now available for the Liberty Profile. You enable thread identity synchronization with the syncToOSThread element in the server.xml file for the WebSphere Application Server Liberty Profile.

Attributes for syncToOSThread

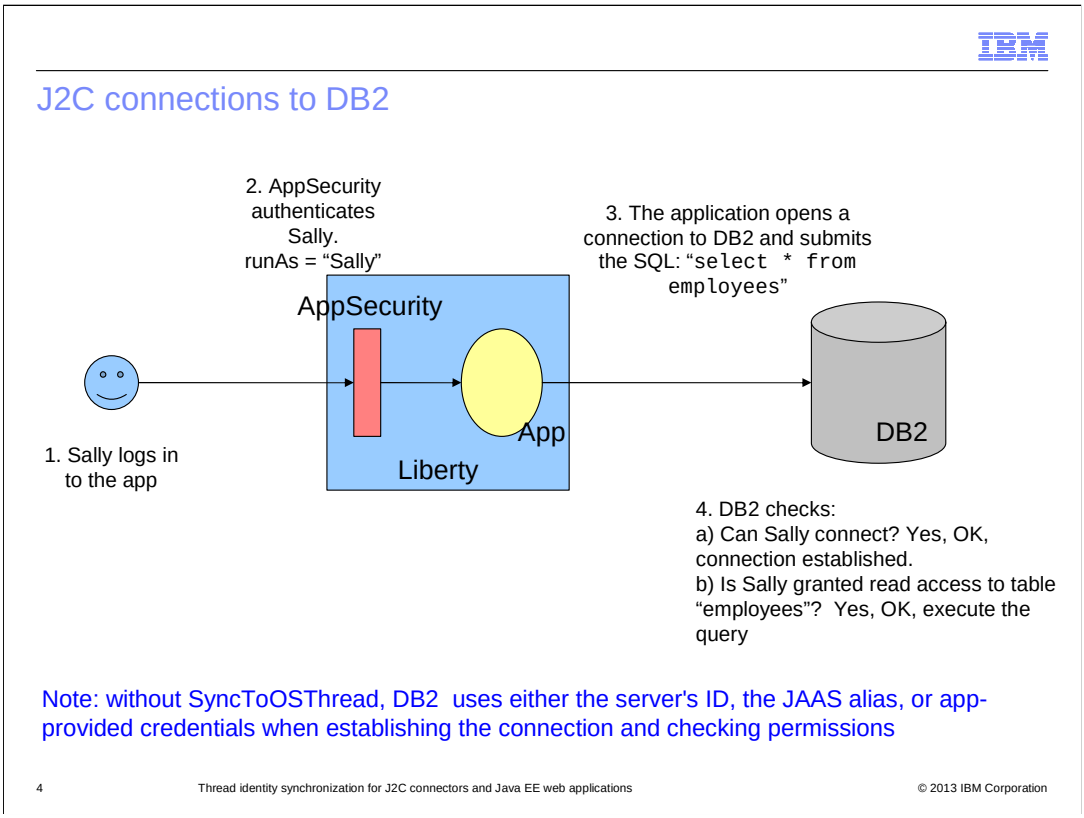
- For J2C connectors
  - The connection is established using the security credentials and privileges of the runAs identity
  - Attribute: j2cEnabled set to true
- For Java EE web applications
  - The application request runs with the runAs identity synced to the thread
  - Attribute: appEnabled set to true

There are two options for using thread identity synchronization on your Liberty server. The first is for J2C connections; the second is for Java EE web applications. These options are not mutually exclusive; you can enable either one or both.

When thread identity synchronization is enabled for J2C connections, the J2C connection is obtained using the security credentials and privileges of the runAs identity. For example, a DB2 connection obtained in this fashion is only able to access the DB2 collections and tables that the runAs identity is authorized to access.

Note that thread identity synchronization for J2C connections only applies to local connections to native subsystems on z/OS that support thread identity -- namely DB2, CICS®, and IMS™.

The second option is to enable thread identity synchronization for Java EE web applications. When applied, the runAs identity is "synced" to the native thread for the duration of the application request. Any native resources accessed by the application are accessed using the runAs identity's security credentials and privileges.



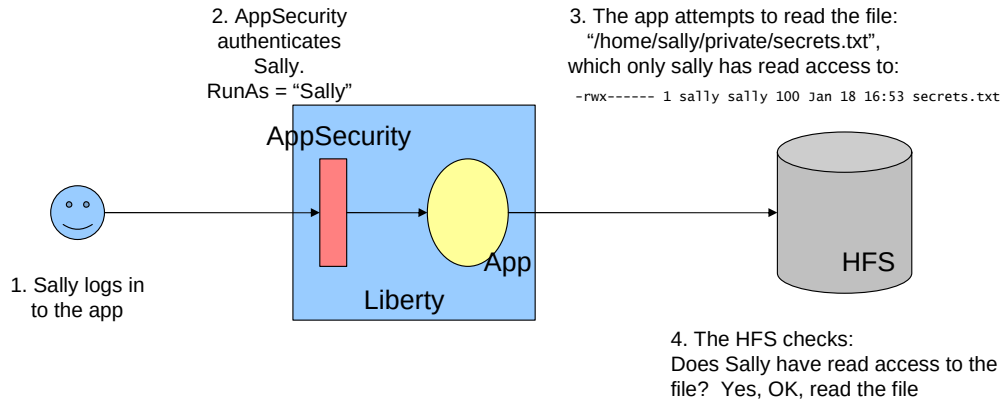
The first scenario deals with thread identity synchronization for J2C connections. In practice, thread identity synchronization for J2C connections is employed much more often than thread identity synchronization for Java EE web applications.

In this scenario, the user, named Sally, logs in to the application hosted by the Liberty server. Sally is authenticated by Liberty's application security layer and her identity is set as the runAs identity for the application request.

As part of the request, the application must read the "employees" table from DB2. Since thread identity synchronization for J2C connections is enabled, the connection is established using the credentials of the runAs identity, which in this case is Sally. DB2 checks whether Sally has permission to open connections; if she does, then the connection is successfully established and returned to the application. The application then issues an SQL query to read all records from the "employees" table. DB2 checks whether Sally is granted read access to the "employees" table; if she is, then DB2 executes the query and returns the results.

Note that normally the DB2 connection is established using either the server's identity, the configured JAAS Alias, or a specific set of credentials provided by the application. With thread identity synchronization, however, the connection is established using the runAs identity. This gives the Liberty administrator fine-grained control over what DB2 resources the application is permitted to access on behalf of the end user.

Application access to the file system



Note: without SyncToOSThread, the application is not be able to read the file, because the server does not have read-access to it

The second scenario deals with thread identity synchronization for Java EE web applications. In this scenario the user, Sally, wants to read a file from the file system by way of a Liberty application. The file in question has strict permissions applied to it. Only Sally is authorized to read the file.

Again, Sally logs in to the application and is authenticated by the application security layer. Her identity is set as the runAs identity for the application request. Additionally, since thread identity synchronization is enabled for the application, her identity is "synced" to the native thread for the duration of the application request.

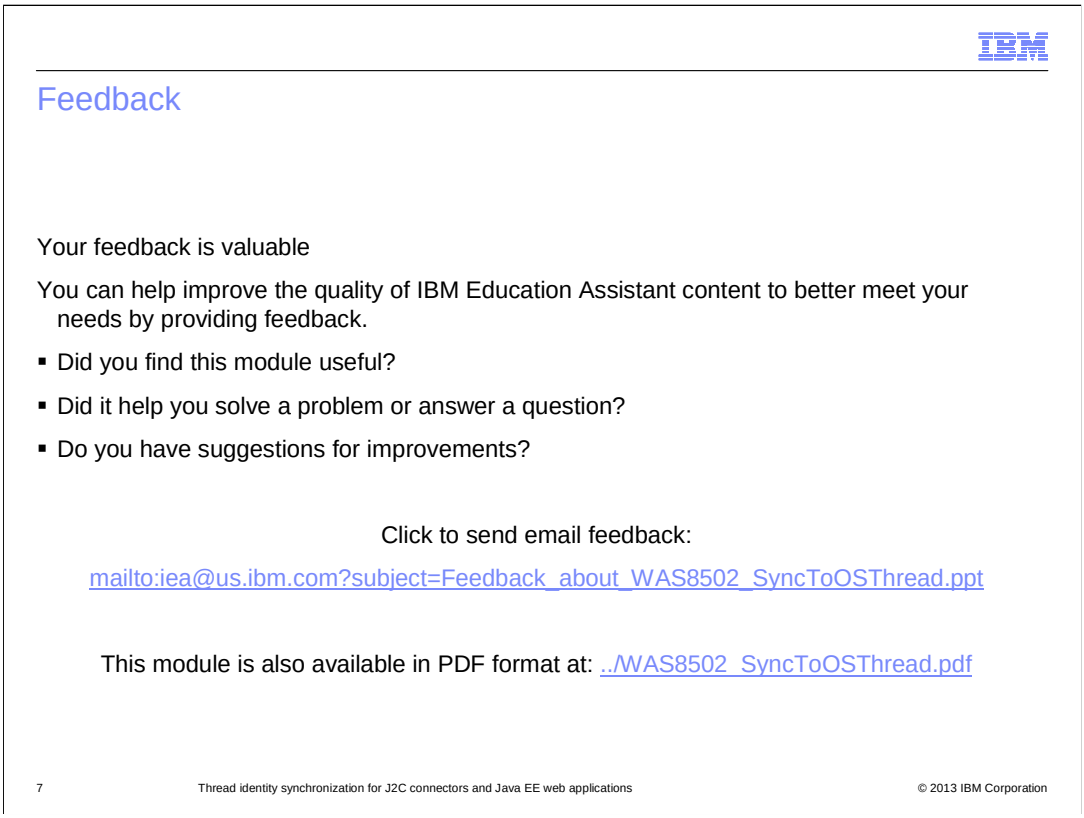
The application then attempts to read the file named "/home/sally/private/secrets.txt". Since Sally's identity is "synced" to the thread, the native file system checks if Sally is authorized to read the file (as opposed to checking the server’s identity). In this case, she is, so the file system permits the application to read the file.

Note that normally the file system is accessed using the server's identity. If the server's identity did not have read permission to the file, then the attempt would fail. With thread identity synchronization, however, the file system is accessed using the runAs identity. Again, this gives you fine-grained control over which file system resources the application is permitted to access on behalf of the end user.

References

- Configuring SyncToOSThread for J2C connectors
  http://www14.software.ibm.com/webapp/wsbroker/redirect?version=phil&product=was-nd-mp&topic=twlp_synctoosthread_j2c
- Configuring SyncToOSThread for Java EE web applications
  http://www14.software.ibm.com/webapp/wsbroker/redirect?version=phil&product=was-nd-mp&topic=twlp_synctoosthread

See these references for additional information about thread identity synchronization.



You can help improve the quality of IBM Education Assistant content by providing feedback.

Trademarks, disclaimer, and copyright information