**ACI Migration Tool**

By Stephen W. Cote

***Overview***

This document describes the use of an ACI migration tool to copy, cache, and migrate ACIs between environments. The tool operates on a specified OU, and will correct System Role DN references in ACI principals.

***Installation***

Note: The tool is only needed when migrating multiple ACIs between environments. After it is used and the results tested, it should be removed and the stashed properties file should be deleted.

On the primary SDI node, create the following directory:

* /opt/IBM/isim6ldap

Unzip the isim6ldap.zip file to the specified directory:

* unzip /ims\_share/ims\_software/other\_software/isim6ldap.zip –d /opt/IBM/isim6ldap

***Uninstallation***

Delete the /opt/IBM/isim6ldap directory.

***Configuring***

At present, each ISIM root, and any organization unit, for which ACIs will be migrated must be configured separately per environment. For example, a company that includes two organizational units, DEV and QA, below the PROLIFICS organization. The tool does not crawl for ACIs, so each OU must be migrated separately.

**Example Configuration**

The following is an example configuration of a development environment with an organization structure of PROLIFICS as the Organization and DEV and QA as Organizational Units under PROLIFICS

* sh config.sh dev "localhost:389" cn=root \*\*\*\*\* erglobalid=00000000000000000000,ou=Prolifics,o=prolifics,c=us "o=prolifics,c=us" "(objectclass=\*)"
* sh config.sh dev.ou.dev "localhost:389" cn=root \*\*\*\*\* erglobalid=7210675883683313286,ou=orgChart,erglobalid=00000000000000000000,ou=Prolifics,o=prolifics,c=us "o=prolifics,c=us" "(objectclass=\*)"
* sh config.sh dev.ou.qa "localhost:389" cn=root \*\*\*\*\* erglobalid=7210810483879394904,ou=orgChart,erglobalid=00000000000000000000,ou=Prolifics,o=prolifics,c=us "o=prolifics,c=us" "(objectclass=\*)"

***Syntax***

To run the tool, use the following syntax from the isim6ldap/ext directory.

java -cp .:../lib/bcprov-jdk16-140.jar:../lib/commons-codec-1.5.jar:../lib/objects-0.0.1-SNAPSHOT.jar:../lib/util-0.0.1-SNAPSHOT.jar:../lib/commons-cli-1.2.jar:../lib/log4j-1.2.16.jar:./isim6ldap.jar com.prolifics.isim.ldap.LdapShell

The following options may be used:

* -sourceEnv: The name of the environment to use as a source. E.G.: dev.ou.dev
* -targetEnv: The name of the environment to use as a target. E.G.: dev.ou.qa
* -argFile: The path to a file containing a list of pre-configured arguments.
  + ./dev.commands.txt
* -test: Specify to test a persistence operation, such as –updateAcis
* -cloneAci: Copies the aci specified by –aciName in –sourceEnv to –cloneName in –targetEnv, and clears any principals.
* -migrateAci: Copies the aci specified by –aciName in –sourceEnv to –targetEnv, and reassigns System Role DNs. If the System Role DN is not found in the target system, an error is raised and the operation fails.
* -migrateAci: Copies all ACIs from –sourceEnv to –targetEnv, and reassigns System Role DNs. If the System Role DN is not found in the target system, an error is raised and the operation fails.
* -backup: Copies all ACIs for –sourceEnv and/or –targetEnv into the cache/backup directory. This copy is kept for error recovery.
* -cache: Copies all ACIs for –sourceEnv and/or –targetEnv into the cache/ directory. This copy is used to persist a local copy of changes until –updateAcis is used.
* -updateAcis: Replaces the eracl attribute in –targetEnv with the values currently held in the cache.

***Backup and Cache***

**Backup**

Backups are created with the –backup option. For example, to backup the ACIs for the DN configured on dev.ou.dev, use the following command:

java -cp .:../lib/bcprov-jdk16-140.jar:../lib/commons-codec-1.5.jar:../lib/objects-0.0.1-SNAPSHOT.jar:../lib/util-0.0.1-SNAPSHOT.jar:../lib/commons-cli-1.2.jar:../lib/log4j-1.2.16.jar:./isim6ldap.jar com.prolifics.isim.ldap.LdapShell –targetEnv dev.ou.dev -backup

In the event of an unforeseen error, the backup copy can be copied into the cache location for a specified environment, and the –updateAcis command used to restore the original configuration.

**Cache**

All operations to add or remove principals, or clone or migrate ACIs, takes place on the cached copy. Before any operation, make sure the cache is refreshed with the latest version of the ACIs.

For example, to refresh the cache for the DN configured on dev.ou.dev, use the following command:

java -cp .:../lib/bcprov-jdk16-140.jar:../lib/commons-codec-1.5.jar:../lib/objects-0.0.1-SNAPSHOT.jar:../lib/util-0.0.1-SNAPSHOT.jar:../lib/commons-cli-1.2.jar:../lib/log4j-1.2.16.jar:./isim6ldap.jar com.prolifics.isim.ldap.LdapShell –targetEnv dev.ou.dev -cache

***Bulk Generating ACI Definitions***

Use the –argFile option to specify the location of a text file containing command definitions. Command definitions are JAXB exports of the Command entities.

### JavaScript to Generate Bulk Commands

To generate these definitions, use the following JavaScript statements with any program capable of evaluating ad-hoc script. If one is not available, use the one located on whitefrost.com by clicking on the right-most tool icon and then clicking the 'Active Source' tab.

/// Clones are copies of ACIs without system role assignments

var aClones = [

"New ACI Name///Existing ACI Name",

];

/// Reports are an array of Report ACI clones with a third key for grouping

var aReports = [

"Run ACI Report///Default ACI for Access Control Information (ACI) Report: Grant Run to Auditor Group///audit"

];

function getCloneAciScript(sSrc, sTarg, sPref, aList,bX){

var aOut = [];

for(var i = 0; i < aList.length;i++){

var aP = aList[i].split("///");

var sSrcName = aP[1];

var sTargName = aP[0];

if(!bX) aOut.push("java -cp .:../lib/bcprov-jdk16-140.jar:../lib/commons-codec-1.5.jar:../lib/objects-0.0.1-SNAPSHOT.jar:../lib/util-0.0.1-SNAPSHOT.jar:../lib/commons-cli-1.2.jar:../lib/log4j-1.2.16.jar:./isim6ldap.jar com.prolifics.isim.ldap.LdapShell -sourceEnv " + sSrc + " -targetEnv " + sTarg + " -cloneAci -cloneName \"" + sPref + " " + sTargName + "\" -aciName \"" + sSrcName + "\"");

else{

aOut.push("<command>"

+ "<arguments><name>sourceEnv</name><value><![CDATA[" + sSrc + "]]></value></arguments>"

+ "<arguments><name>targetEnv</name><value><![CDATA[" + sTarg + "]]></value></arguments>"

+ "<arguments><name>cloneAci</name></arguments>"

+ "<arguments><name>cloneName</name><value><![CDATA[" + sPref + " " + sTargName + "]]></value></arguments>"

+ "<arguments><name>aciName</name><value><![CDATA[" + sSrcName + "]]></value></arguments>"

+ "</command>");

}

}

return aOut.join("\n");

}

function addPrincipal(sName, sTarg, sAci, bX){

if(!bX) return "java -cp .:../lib/bcprov-jdk16-140.jar:../lib/commons-codec-1.5.jar:../lib/objects-0.0.1-SNAPSHOT.jar:../lib/util-0.0.1-SNAPSHOT.jar:../lib/commons-cli-1.2.jar:../lib/log4j-1.2.16.jar com.prolifics.isim.ldap.LdapShell -targetEnv " + sTarg + " -addPrincipal -principalName \"" + sName + "\" -aciName \"" + sAci + "\"";

else{

return "<command>"

+ "<arguments><name>targetEnv</name><value><![CDATA[" + sTarg + "]]></value></arguments>"

+ "<arguments><name>addPrincipal</name></arguments>"

+ "<arguments><name>principalName</name><value><![CDATA[" + sName + "]]></value></arguments>"

+ "<arguments><name>aciName</name><value><![CDATA[" + sAci + "]]></value></arguments>"

+ "</command>";

}

}

function addReportGroup(sName, sTarg, sRepName,sPref,aList,bX){

var aRep = [];

for(var i = 0; i < aList.length;i++){

var aP = aList[i].split("///");

if(aP.length < 3) continue;

if(aP[2] != sRepName) continue;

aRep.push(addPrincipal(sName, sTarg,sPref + " " + aP[0],bX));

}

return aRep.join("\n");

}

function buildClones(){

var aBuff = [];

/// Clone an ACI from the 'dev' environment into the 'dev.ou.dev' environment and prefix the new ACI name with 'DEV'

aBuff.push(getCloneAciScript("dev","dev.ou.dev","DEV",aClones,1);

/// Clone the Report ACIs back into 'dev'

aBuff.push(getCloneAciScript("dev","dev ","DEV",aReports,1);

return aBuff.join("\n");

}

function buildPrincipals(){

var aBuff = [];

aBuff.push(addPrincipal("ISIM Group Name","dev.ou.dev","DEV New ACI Name",1));

///

aBuff.push(addReportGroup("ISIM Group Name","dev","audit","DEV",aReports,1));

return aBuff.join("\n");

}

buildClones();

buildPrincipals();

Save all of the output data into a text file. **Do not pretty print the XML**.

### Example JavaScript Output

The following is an example of the output . **Do not pretty print the XML.**

<command><arguments><name>targetEnv</name><value><![CDATA[dev.ou.dev]]></value></arguments><arguments><name>addPrincipal</name></arguments><arguments><name>principalName</name><value><![CDATA[ISIM Group Name]]></value></arguments><arguments><name>aciName</name><value><![CDATA[DEV New ACI Name]]></value></arguments></command>

For example, to process the argument file for dev:

java -cp .:../lib/bcprov-jdk16-140.jar:../lib/commons-codec-1.5.jar:../lib/objects-0.0.1-SNAPSHOT.jar:../lib/util-0.0.1-SNAPSHOT.jar:../lib/commons-cli-1.2.jar:../lib/log4j-1.2.16.jar:./isim6ldap.jar com.prolifics.isim.ldap.LdapShell –argFile ./dev.commands.txt

Check the log output each time and verify that source ACIs were found, and that all principals were found. If either were not found, then verify that the named ACI or ISIM Group exists.

***Migrating ACI Definitions***

Use the –migrateAcis option for each organization and organizational unit in each source and target environment. It is recommended to start with one organizational unit, migrate the acis, update the acis, and then verify in both the LDAP and the ISIM Admin Console that the ACIs are searchable and specify participants.

When running the –migrateAci or –migrateAcis option, an error will be thrown if a System Role is not found in the target environment. In this case, verify the ISIM Group is created and try again.

Once the –migrateAci or –migrateAcis operation has succeeded, run the –updateAcis option to persist the change.

For example, to migrate ACIs from the DEV Organizational Unit in Development to the DEV Organizational Unit in QA, use the following:

java -cp .:../lib/bcprov-jdk16-140.jar:../lib/commons-codec-1.5.jar:../lib/objects-0.0.1-SNAPSHOT.jar:../lib/util-0.0.1-SNAPSHOT.jar:../lib/commons-cli-1.2.jar:../lib/log4j-1.2.16.jar:./isim6ldap.jar com.prolifics.isim.ldap.LdapShell –sourceEnv dev.ou.dev –targetEnv qa.ou.dev –migrateAcis

### Testing Updates

Before updating any ACIs, use the –test option to verify that the current state of the target environment will support the ACIs (eg: the System Roles haven’t changed), and that all of the ACIs can be parsed correctly.

For example, to test ACI updates to the DEV Organizational Unit in QA, use the following:

java -cp .:../lib/bcprov-jdk16-140.jar:../lib/commons-codec-1.5.jar:../lib/objects-0.0.1-SNAPSHOT.jar:../lib/util-0.0.1-SNAPSHOT.jar:../lib/commons-cli-1.2.jar:../lib/log4j-1.2.16.jar:./isim6ldap.jar com.prolifics.isim.ldap.LdapShell–targetEnv qa.ou.dev –updateAcis –test

***Update ACIs***

After ACIs have been cloned and principals updated, or the ACIs have been migrated, and after the changes have been tested, the ACIs can be updated in the target environment. The –updateAcis option will replace the eracl attribute values for the specified Organization DN with the values currently held in the cache for that environment.

For example, to update the ACIs to the DEV Organization Unit in QA, use the following:

java -cp .:../lib/bcprov-jdk16-140.jar:../lib/commons-codec-1.5.jar:../lib/objects-0.0.1-SNAPSHOT.jar:../lib/util-0.0.1-SNAPSHOT.jar:../lib/commons-cli-1.2.jar:../lib/log4j-1.2.16.jar:./isim6ldap.jar com.prolifics.isim.ldap.LdapShell–targetEnv qa.ou.dev –updateAcis

After updating ACIs, they should be immediately verified with a direct LDAP inspection of the eracl attribute to verify the values were correctly set, and that the ACIs are visible and with the current principals in the target environment’s ISIM Admin Console.

***Troubleshooting***

* After updating ACIs, if they are updated in the LDAP but not yet displayed in the ISIM console, wait a short period of time for ISIM to update it’s internal cache. Restarting ISIM to force a refresh, though this is not desirable in Production environments.
* After updating ACIs, if the eracl attribute is empty in LDAP or contains invalid data, then delete the cache for the specified environment, copy the backup for that environment into the cache location, and run –updateAcis to restore the original values.