IBM Identity Manager Command Line Interface Adapter

White paper

9/4/2012
IBM

First Edition (January 23, 2011)

This edition applies to version 5.1 of Tivoli Identity Manager and to all subsequent releases and modifications until otherwise indicated in new editions.

Revision History

Version	Date	Created by	Short Description
2.1	Sep 4 2012	Amita Lele	ISIM 6.0 changes – expanded test operation

Table of Contents

Prerequisites

In an effort to create a simple and concise white paper, it is assumed that the readers of this paper have understanding and knowledge in the following topics:

- IBM Identity Manager
- Identity Manager adapters high level architecture
- Identity Manager adapters customization
- Identity Manager adapters detailed profile customization
- Tivoli Directory Integrator connectors and assembly lines
- LDAP schema: attributes and object classes

Refer to the proper documentation for background understanding of the above topics. For your convenience, the *Custom Adapter Developer's Guide* is included with the Command Line Interface adapter package.

Introduction

The Command Line Interface (CLI) adapter is an IBM Identity Manager adapter designed primarily for applications and resources that do not offer a programmable API. Without an API to the managed resource, an adapter developer is forced to issues command lines from the Java code to accomplish the required user management tasks.

Built on the TDI framework, the CLI adapter provides all the programming modules needed by an adapter: TDI connector and assembly lines. Instead of calling a set of APIs to perform a specific operation (i.e. create account), the CLI adapter will execute a command script locally on the operating system where the adapter is running. The adapter will pass the attributes as parameters to the command script. The command will perform the requested operation (i.e. create account) and returns the status.

- *The CLI adapter package provides a sample CLI profile and a complete set of command line batch files. The sample profile and command line batch files illustrate a fully functional adapter. It's highly recommended that you review the content of the* ***CLI Sample adapter****. The CLI Sample adapter will be referenced throughout this document.*

High Level Design

As with any Identity Manager adapter that is built on the TDI framework, the CLI adapter consists of a profile, assembly lines and a connector. The diagram in Figure 1 shows the data flow between the Identity Manager and the adapter. The red boxes identify the profile and the command line scripts: the two modules that will require changes for every deployment of the adapter. The CLI assembly lines and the CL connector are static and will not change.

The CLI assembly lines and the CL connector have no knowledge of the managed resource. The functionality of the connector is to run a command line script for a particular operation. The assembly lines will map all the attributes defined by the CLI profile. Through the profile, the adapter is configured to execute a specific command script per operation. One or more operation can be configured to use the same command script.

The command line script will contain all the logic and the proper methods needed to communicate with the managed resource. The command line script performs the requested operation (i.e. create account) and returns the status.

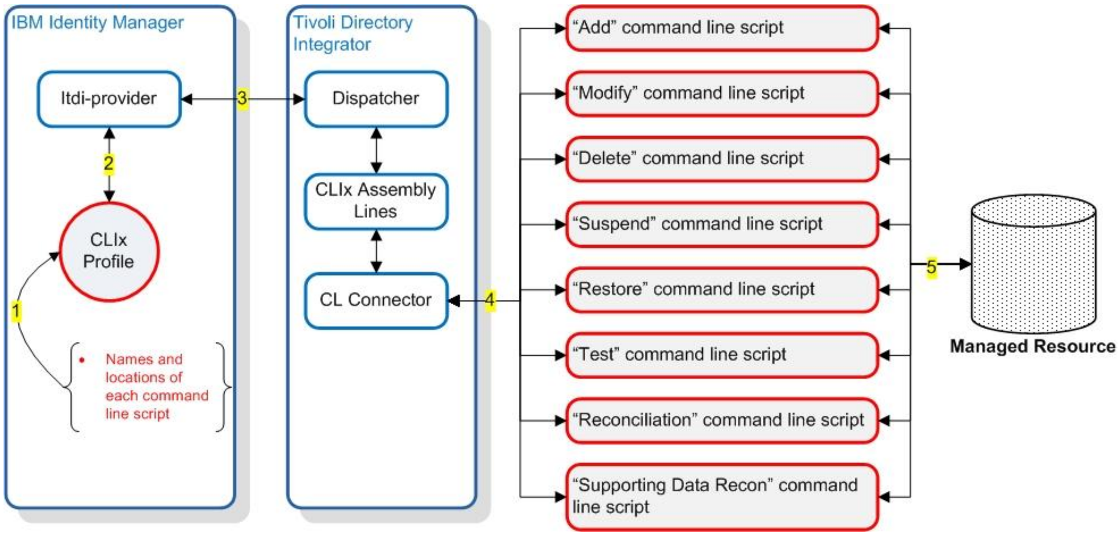

IBM Identity Manager
Itdi-provider
2
CLIx Profile
1
Names and locations of each command line script
3
Tivoli Directory Integrator
Dispatcher
CLIx Assembly Lines
CL Connector
4
"Add" command line script
"Modify" command line script
"Delete" command line script
"Suspend" command line script
"Restore" command line script
"Test" command line script
"Reconciliation" command line script
"Supporting Data Recon" command line script
5
Managed Resource

Figure 1 - High Level Design

Components Detail

As described in the *High Level Design* section and *Figure 1*, two components of the CLI adapter must be changed (or created) in order to deploy your custom CLI adapter: a profile and a set of command line scripts.

In this section we will describe each component; however it's ***imperative*** to have an understanding of the adapters' profiles. Refer to *Custom Adapter Developer's Guide* for detailed information on profiles.

Command Line Scripts

A command line script is a text file that contains commands that can be executed on the OS command prompt.

The CLI adapter command line script file must adhere to the following:

- Parse the command line arguments as attribute value pairs
- Set the status of the request and redirect the standard output to standard error

The CLI adapter was tested using the following command line scripts:

- Windows DOS batch files
- UNIX Bourne
- bash

➔ Other scripting languages may work with the CLI adapter; however, this release of the adapter was certified on the above list only.

➔ Java or C/C++ programs can be called from the command line script. It may be necessary to develop your solution in Java or C/C++ due to the managed resource limitations. In that case, the command line script files will be used only to pass the arguments to your program.

Parsing the command line arguments

The CLI adapter calls the command line script with the following syntax:

scriptname -attribute1 val1 –attribute2 val2 –attribute3 val3 …

For example, the CLI Sample adapter ADD script (clix-add.bat) is called as follow:

```
CLIx\clix-add.bat -s:erSampleCmdServer ACME -s:erServiceUid admin -s:erPassword test -erUid johnd -erPassword
changeme -erSampleFirstName John -erSampleLastName Doe -erSampleUserLocation DEPT-A -erSampleMemberOf
101 -erSampleMemberOf 102 -erSampleMemberOf 103 -erSampleUserRole Developer
```

It's the responsibility of the command line script to parse the command line arguments and retrieve the attributes and their values.

- Note the following from the above call to clix-add.bat command line script:
  - ➢ The "s:" prefix for some attributes on the command line. These attributes are designated as service attributes on the profile. The purpose of service attributes is for the command line script to distinguish them from account attributes. Service attributes belong to the service object class and can be used by the command line script to connect to the resource or as configuration items to determine command line behavior.
  - ➢ Attribute erSampleMemberOf is repeated three times on the command line. This is done since attribute erSampleMemberOf is a multi-valued attribute.
- Review the CLI Sample command line batch files. The files are well commented and describe how to retrieve attributes and return status back to the adapter.

Search result (reconciliation)

The search (reconciliation) command line script file must output the search results to the standard output. The output data must have the following syntax:

```
objectclass=objectclass-name
$dn=eruid=value
eruid=value
attribute-name=value
 <repeat for every object>
```

Basically, the output of the script is a dump of the objects that is known to the adapter (accounts and supporting data). Each entry must begin with the key word **objectclass** and its value. The object class value is the name defined in the adapter schema file. Section *Schema.dsml file* describes the schema object classes in more detail.

$dn and **eruid** are required. The $dn indicated the naming (unique) attribute of the object. The “eruid” must be used as the $dn for all account objects. The remaining of each entry is a list of attributes and their values. For multi value attribute, repeat the attribute on a separate line with its value. Below is an example of the CLI Sample adapter reconciliation command line script:

```
objectclass=erSampleCmdAccount
$dn=eruid=jdoe
eruid=jdoe
ersamplefirstname=John
ersamplelastname=Doe
eraccountstatus=0
ersampleuserlocation=West Coast
ersampleuserrole=Manager
ersamplememberof=101
ersamplememberof=103
objectclass=erSampleCmdAccount
$dn=eruid=msmith
eruid=msmith
ersamplefirstname=Mike
ersamplelastname=Smith
eraccountstatus=0
ersampleuserlocation=East Coast
ersampleuserrole=Supervisor
ersamplememberof=101
ersamplememberof=102
objectclass=erSampleCmdGroups
$dn=erSampleGroupId=101
erSampleGroupId=101
erSampleGroupName=Development
objectclass=erSampleCmdGroups
$dn=erSampleGroupId=102
erSampleGroupId=102
erSampleGroupName=Support
```

Return status and error message of the command line

The exit status of the command line script determines the status of the request initiated from the Identity Manager server. There are only two statuses: success or failure, no warning status is currently supported by the CLI adapter.

Success status: simply exit the command line script with a value of 0.

Error status: echo the error message to the standard error and exit the command line script with a value of 1. An example from the clix-add.bat command line script of the CLI Sample adapter:

```
echo %DATE% %TIME% [add request] error: missing ersamplefirstname >> CLIx\logs\trace.log
echo “missing first name” 1>&2
set status=1
exist %status%
```

The Profile

The CLI profile defines the service type in the Identity Manager server. This section describes the components needed to create a CLI profile. It's highly recommended to view the CLI Sample adapter profile (SampleCmdProfile) while reading this section. The SampleCmdProfile.jar profile is located under the SampleCLI folder of the adapter package.

More detailed information is provided in the CLI Sample Adapter section of this document.

Schema.dsml file

The schema.dsml must contain at least two object classes: account object class and the service object class. Other object classes for service groups and supporting data are optional and can be defined.

Account object class:

The account object class will always contain the following four attributes:

- erUid
- erPassword
- erAccountStatus
- erLastAccessDate

erUid is the only required attribute and it must be used as the unique identifier on the managed resource. These attributes are already defined in the Identity Manager schema and ***should not be*** defined in the adapter schema file. They can be only included in the account object class.

The ***erAccountStatus*** attribute is used by the Identity Manager server to determine if an account is active or inactive. A value of 0 (zero) indicates that the account is active; a value of 1 indicates that the account is inactive. The adapter must return a value of 0 or 1.

Service object class:

The service object class defines the communication used by the itdi-provider on the Identity Manager server and the adapters. There are few required attributes that are already defined in the Identity Manager schema and must be included in the CLI service object class:

Attributes that are already defined in the Identity Manager schema

- erServiceName
- description
- erITDIurl

Attributes needed for CLI adapter must be defined in the adapter schema.

- Command Line script locations

Expanded Test Request attributes (ISIM 6.0- already defined in the Identity Manager schema)

- erAdapterLastStatusTime
- erAdapterResourceStatusMsg
- erAdapterPlatform
- erAdapterAccount
- erAdapterUpTime
- erAdapterMemory
- erAdapterTdiVersion
- erAdapterProfileVersion

- erAdapterDispatcherVersion
- erAdapterVersion
- erAdapterResourceStatus
- erAdapterResourceVersion
- erAdapterConnectorVersion

- erServiceName is the unique identifier for a service object instance on the Identity Manager.

CustomLabels.properties file:

The CustomLabels.properties file contains labels for attributes and profile definitions. More detailed information is provided in the CLI Sample Adapter section of this document.

Account form xml file

An account form file is not required when importing an adapter profile into Identity Manager. Initially, the profile will may not contain an account form. Identity Manager will create a default form with all the attributes in the object class. Use the Identity Manager Form designer to fix the form and extract the xml source.

By convention, the file name is the account object class name with the xml extension. For example the CLI sample adapter account form file is: erSampleCmdAccount.xml

Note that the service .def file contains the actual name of the account form file name.

Service form xml file

A service form file is not required when importing an adapter profile into Identity Manager. Initially, the profile will may not contain a service form. Identity Manager will create a default form with all the attributes in the object class. Use the Identity Manager Form designer to fix the form and extract the xml source.

By convention, the file name is the service object class name with the xml extension. For example the CLI sample adapter service form file is: erSampleCmdService.xml

Note that the service .def file contains the actual name of the service form file name.

service.def file

The service .def file is the piece that brings all the profile components together. More detailed information is provided in the CLI Sample Adapter section of this document.

CLI Sample Adapter

The CLI adapter package provides a comprehensive sample adapter. The sample adapter can be deployed on any Windows platform. The adapter manages a virtual application.

- ***Make sure that you have access to the CLI Sample profile jar file (SampleCmdProfile.jar) and the CLI sample adapter command line batch files while reading this section.***

The managed application

The application that the CLI sample adapter manages has users, roles and groups. A user can have a role and may belong to one or more group.

User data

- User name (unique identifier) – required
- User password - required
- First name - required
- Last name – required
- Account status (active or not active) – optional, if not supplied it defaults to active.
- User location – optional: a text field that can contain any value
- User role – optional: a single value field that can contain any valid defined role name.
- User group membership – optional: a multi value field that can contain one or more group ID from the list of valid groups.

Each user is stored in its own text file with the following text format (*username.dat*):

```
objectclass=erSampleCmdGroups
$dn=eruid=jdoe
eruid=jdoe
ersamplefirstname=John
ersamplelastname=Doe
eraccountstatus=0
ersampleuserlocation=West Coast
ersampleuserrole=Manager
ersamplememberof=Administrators
ersamplememberof=Support
```

Roles data

- Role name

All Roles data is stored in one text file with the following format (roles.support):

```
objectclass=erSampleCmdRoles
$dn=erSampleRoleName=Scrum master
erSampleRoleName=Scrum master
objectclass=erSampleCmdRoles
$dn=erSampleRoleName=Team leader
erSampleRoleName=Team leader
objectclass=erSampleCmdRoles
$dn=erSampleRoleName=Security officer
erSampleRoleName=Security officer
```

Group data

- Group ID
- Group name

All Groups data is stored in one text file with the following format (groups.support):

```
objectclass=erSampleCmdGroups
$dn=erSampleGroupId=101
erSampleGroupId=101
erSampleGroupName=Development
objectclass=erSampleCmdGroups
$dn=erSampleGroupId=102
erSampleGroupId=102
```

```
erSampleGroupName=Support
objectclass=erSampleCmdGroups
$dn=erSampleGroupId=103
erSampleGroupId=103
erSampleGroupName=Documentation
objectclass=erSampleCmdGroups
$dn=erSampleGroupId=104
erSampleGroupId=104
erSampleGroupName=Marketing
```

➔ Note that the data (users and supporting data) is stored in the same format as the reconciliation syntax. This is done intentionally to simplify the reconciliation command line script file.

Data storage location

Under the CLI Sample adapter installation directory, there will be sub-directories used for data storage. Each sub-directory resembles an instance of the resource. These instances are presented on the Identity Manager server as service instances. The service definition of the CLI Sample adapter includes a "Managed Server name" attribute. This value of the "Managed Server name" is the sub-directory name under the CLI Sample adapter installation directory. All data for that server is stored under the directory.

See section *Installing the CLI Sample adapter* for visual description of the data storage location.

Adapter service type (profile)

The first step of adapters' development is to identity how to present the managed resource data on the Identity Manager server.

CLI Sample schema (schema.dsml file)

- Use the schema.dsml file of the CLI Sample profile jar file (SampleCmdProfile.jar) for reference.

  Define the user data: Refer to object class "*erSampleCmdAccount*" defined in the schema.dsml file of the SampleCmdProfile.jar profile.

Resource field	Adapter attribute name
User name	erUid
User password	erPassword
First name	erSampleFirstName
Last name	erSampleLastName
Account status	erAccountStatus
User location	erSampleUserLocation
User role	erSampleUserRole
Group membership	erSampleMemberOf

Table 1 - CLI Sample adapter users attributes

➔ erUid, erPassword and erAccountstatus are built-in Identity Manager attributes. They *must not be* defined in the schema.dsml file, only included in the object class definition.

Define the roles data: The purpose of the role data is to present all the valid roles on the Identity Manager server. The role data will be defined as supporting data. Refer to object class "*erSampleCmdRoles*" defined in the schema.dsml file of the SampleCmdProfile.jar profile.

Resource field	Adapter attribute name
Role name	erSampleRoleName

Table 2 - CLI Sample adapter role attributes

Define the groups data: The purpose of the group data is to present all the valid groups on the Identity Manager server. The group data will be defined as supporting data. Refer to object class "*erSampleCmdGroups*" defined in the schema.dsml file of the SampleCmdProfile.jar profile.

Resource field	Adapter attribute name
Group ID	erSampleGroupId
Group name	erSampleGroupName

Table 3 - CLI Sample adapter groups attributes

Service connection definition: The service connection data is used by the Identity Manager server to communicate with the adapter. Refer to object class "*erSampleCmdService*" defined in the schema.dsml file of the SampleCmdProfile.jar profile.

Adapter service attribute name	Required	Description and usage
erServiceName	Yes	Service name. A short descriptive name of the service instance. Note: do not include any forward or backward slashes in the service name.
description	No	A description of the service instance. Recommended.
erITDIurl	Yes	The location of the Dispatcher service. Not required in the object class, but must be supplied. The service.def file can be used to set a default value.
erSampleCmdServer	No	The managed resource name (IP address or server) that is managed by the CLI Sample adapter. Used to demonstrate service attributes.
erServiceUid	No	The administrator ID of the managed resource. Used to demonstrate service attributes.
erPassword	No	Password of the administrator ID of the managed resource. Used to demonstrate service attributes.
erSampleAddCommand	Yes	Location and name of the command line script to execute for ADD request. For all the command line scripts, If the full path is not specified, then the location is relative to the adapters' solutions directory.
erSampleModCommand	Yes	Command line script to execute for Modify request.
erSampleDelCommand	Yes	Command line script to execute for Delete request.
erSampleSearchCommand	Yes	Command line script to execute for Reconciliation request.
erSampleSupportDataCommand	Yes	Command line script to execute for Supporting Data Reconciliation request.
erSampleSuspendCommand	Yes	Command line script to execute for Suspend request.
erSampleRestoreCommand	Yes	Command line script to execute for Restore request.
erSampleTestCommand	Yes	Command line script to execute for Test request.
erSampleDisableALCache	No	Dispatcher parameter: see the *Custom Adapter Developer's Guide*
erSampleALFileSystemPath	No	Dispatcher parameter: see the *Custom Adapter Developer's Guide*
erSampleMaxConnectionCount	No	Dispatcher parameter: see the *Custom Adapter Developer's Guide*
Expanded Test Request attributes(ISIM6.0)		
erAdapterLastStatusTime	No	Date/Time of last expanded test response

erAdapterResourceStatusMsg	No	Managed resource status
erAdapterPlatform	No	Adapter installation platform
erAdapterAccount	No	Account running the Adapter binary
erAdapterUpTime	No	Adapter uptime
erAdapterMemory	No	Adapter memory usage
erAdapterTdiVersion	No	TDI version
erAdapterProfileVersion	No	Adapter profile version
erAdapterDispatcherVersion	No	Dispatcher version
erAdapterVersion	No	Adapter version
erAdapterResourceStatus	No	Managed resource status
erAdapterResourceVersion	No	Managed resource version
erAdapterConnectorVersion	No	Managed resource status message when the resource is not available.

Table 4 - CLI Sample adapter service attributes

➔ erServiceName, description, erITDIurl , erServiceUid and erPassword are Identity Manager attributes. They *must not be* defined in the schema.dsml file.

CLI Sample labels (CustomLabels.properties file)

- Use the CustomLabels.properties file of the CLI Sample profile jar file (SampleCmdProfile.jar) for reference.

  The CustomLabels.properties file contains labels for attributes and profile definitions. The best way to describe the file is to include a portion the CLI Sample adapter file:

```
# service and account profile labels
SampleCmdProfile=Sample Command-line profile
SampleCmdProfileDesc=Sample Command-line profile
erSampleCmdService=Sample Command-line service
erSampleCmdAccount=Sample Command-line account
SampleCmdAccount=Sample Command-line account

ersamplefirstname=First name
ersamplelastname=Last name
ersampleuserlocation=User location
ersamplememberof=Group membership
ersampleuserrole=User role
```

Note the SampleCmdProfile**Desc** property. The SampleCmdProfile is the profile name. We append a “Desc” to constructs a property that the Identity Manager will look for to display when listing profile types. Also included above:

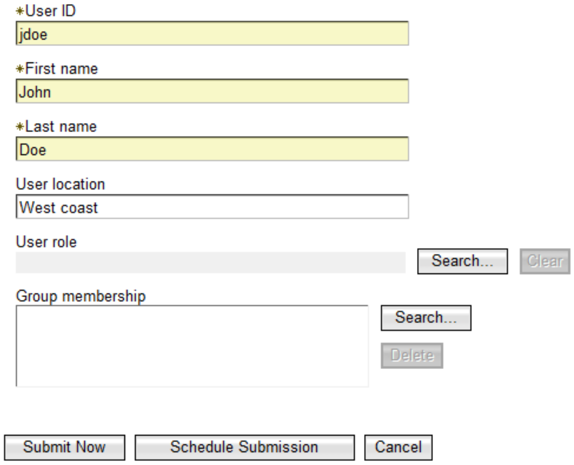
service profile name (SampleCmdProfile)
account profile name (SampleCmdAccount)
service object class name (erSampleCmdService)
account object class name (erSampleCmdAccount)

In addition, all attributes from the account and service objects that are defined in the schema.dsml file will appear in the CustomLabels.properties file.

CLI Sample account form (erSampleCmdAccount.xml)

- Use the erSampleCmdAccount.xml file of the CLI Sample profile jar file (SampleCmdProfile.jar) for reference.

The account form was created using the Identity Manager Form designer. Most of the fields are plain text widgets with the exception of role (erSampleUserRole) and group membership (erSampleMemberOf) attributes. For these two attributes, a search widget is used to list values from the supporting data.



*User ID
jdoe

*First name
John

*Last name
Doe

User location
West coast

User role

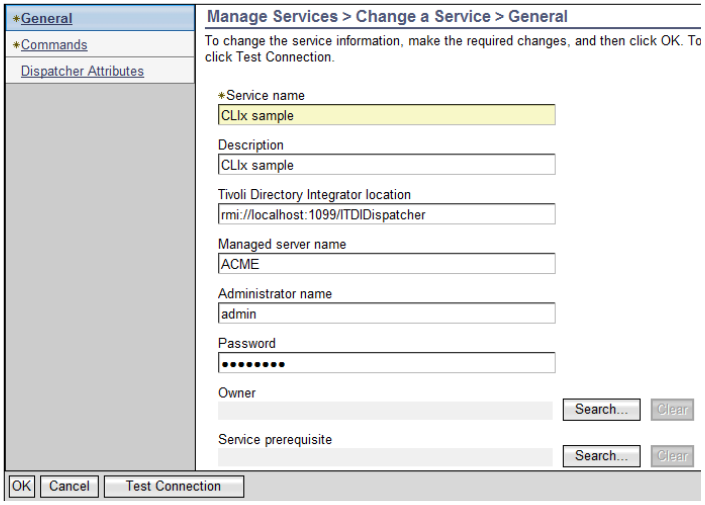
Group membership

Figure 2 - CLI Sample adapter account form

CLI Sample service form (erSampleCmdService.xml)

- Use the erSampleCmdService.xml file of the CLI Sample profile jar file (SampleCmdProfile.jar) for reference.

The service form of the CLI Sample adapter consists of three panels: General or connection data, Command line scripts information and Dispatcher parameters. For ISIM 6.0 there is an additional tab “Status and Information”.



Manage Services > Change a Service > General

To change the service information, make the required changes, and then click OK. To click Test Connection.

*Service name: CLIx sample

Description: CLIx sample

Tivoli Directory Integrator location: rmi://localhost:1099/ITDIDispatcher

Managed server name: ACME

Administrator name: admin

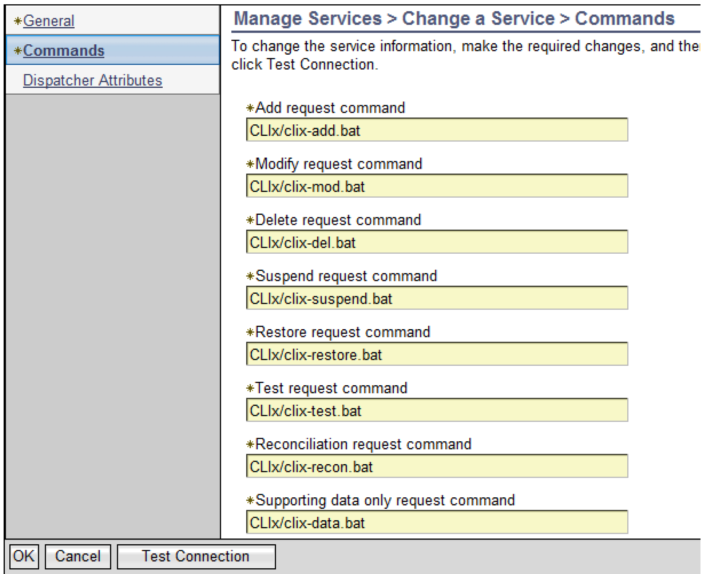
Password: ••••••••

Owner:

Service prerequisite:

Figure 3 - CLI Sample adapter service form panel 1

- “Owner” and “Service prerequisite” are not required. These two fields are Identity Manager specific attributes that can be used by Identity Manager for work flow and dependencies between service instances.



Manage Services > Change a Service > Commands

To change the service information, make the required changes, and the click Test Connection.

*Add request command: CLIx/clix-add.bat

*Modify request command: CLIx/clix-mod.bat

*Delete request command: CLIx/clix-del.bat

*Suspend request command: CLIx/clix-suspend.bat

*Restore request command: CLIx/clix-restore.bat

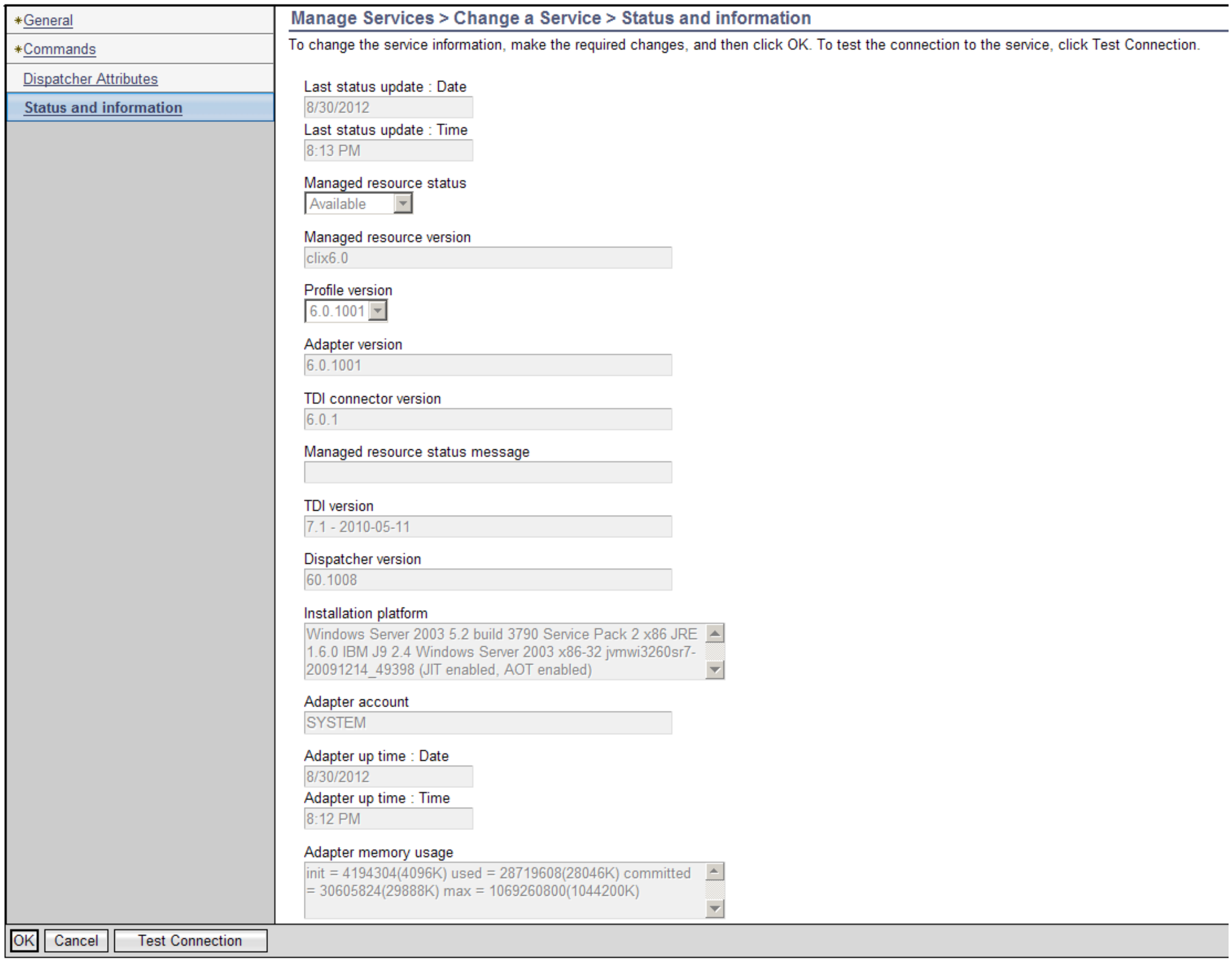
*Test request command: CLIx/clix-test.bat

*Reconciliation request command: CLIx/clix-recon.bat

*Supporting data only request command: CLIx/clix-data.bat

Figure 4 - CLI Sample adapter service form panel 2

- On this panel, all command line scripts are supplied.



Manage Services > Change a Service > Status and information

To change the service information, make the required changes, and then click OK. To test the connection to the service, click Test Connection.

Last status update : Date
8/30/2012

Last status update : Time
8:13 PM

Managed resource status
Available

Managed resource version
clix6.0

Profile version
6.0.1001

Adapter version
6.0.1001

TDI connector version
6.0.1

Managed resource status message

TDI version
7.1 - 2010-05-11

Dispatcher version
60.1008

Installation platform
Windows Server 2003 5.2 build 3790 Service Pack 2 x86 JRE 1.6.0 IBM J9 2.4 Windows Server 2003 x86-32 jvmwi3260sr7-20091214_49398 (JIT enabled, AOT enabled)

Adapter account
SYSTEM

Adapter up time : Date
8/30/2012

Adapter up time : Time
8:12 PM

Adapter memory usage
init = 4194304(4096K) used = 28719608(28046K) committed = 30605824(29888K) max = 1069260800(1044200K)

Figure 5 - CLI Sample adapter service form panel 4

- On this panel, all ‘Expanded Test Request’ attribute are displayed in read-only mode(ISIM 6.0 specific)

CLI Sample service profile definition (service.def)

- Use the service.def of the CLI Sample profile jar file (SampleCmdProfile.jar) for reference. The service.def file is well commented and describes each section of the file in details.

The service .def file is the piece that brings all the profile components together. There are sections in the service.def file.

Service profile definition

This section is at the top of the file. All adapters must use the exact same syntax. The values in bold is what needs to be changed:

<Service erserviceproviderfactory="com.ibm.itim.remoteservices.provider.itdiprovider.ItdiServiceProviderFactory"

```
name="SampleCmdProfile">

<!-- service profile definition -->
<type category="service" location="schema.dsml"
        name="erSampleCmdService" profile="SampleCmdProfile">
        <key>
                <field>erservicename</field>
        </key>
        <form location="erSampleCmdService.xml"/>
</type>
```

SampleCmdProfile: the service profile name.

erSampleCmdService: service object class name from the schma.dsml file.

erSampleCmdService.xml: service form file name.

Account profile definition

The next section defines the account profile.

```
<!-- account profile definition -->
<type category="account" location="schema.dsml"
        name="erSampleCmdAccount" profile="SampleCmdAccount">
        <key>
        <field>eruid</field>
        </key>
        <form location="erSampleCmdAccount.xml"/>
</type>
```

erSampleCmdAccount: account object class name from schema.dsml file.

SampleCmdAccount: the account profile name.

erSampleCmdAccount.xml: account form file name.

Operation definition

This section defines the assembly line name and the operations used for each operation. Multiple operations can use the same assembly line. This section also contains information such as:

- Connector parameters
- Dispatcher parameters
- Attribute defaults
- Service profile specific behaviors.

Additional section

The service.def file contains addition section such as:

- Properties definition
- Attribute Map section
- Service groups definition

For more detailed description of the service.def file is available in the *Custom Adapter Developer's Guide*.

Deploy and run the CLI Sample adapter

The CLI sample adapter is located under the ***SampleCLI*** folder of the adapter package. The content of the folder is as follow:

- SampleCmdProfile.jar – the adapter profile jar file.
- CLIx.zip – the command line script files.

- The command line scripts files of the Sample adapter are Windows dos batch files. They must run on a Windows platform.

Note: A sample UNIX shell command line script file, ***add-req.sh,*** is supplied for reference purposes only. It demonstrates how to parse the command line and set error status.

Installing the CLI Sample adapter

- Install the CLI adapter as described in the adapters installation guide.
- From the "manage service type" option on the Identity Manager, import the SampleCmdProfile.jar profile.
- On the server where the Dispatcher is running, unzip the content of the CLIx.zip file into the adapters' solution directory.
    - A new directory called CLIx will be created.
    - All contents of CLIx.zip must be under the CLIx directory.
    - The CLIx directory contains all the command line script files.
    - The CLIx directory contains all the sub-folders, each representing an instance of the managed resource.
    - Data files (users and supporting data) are stored under the sub-folders

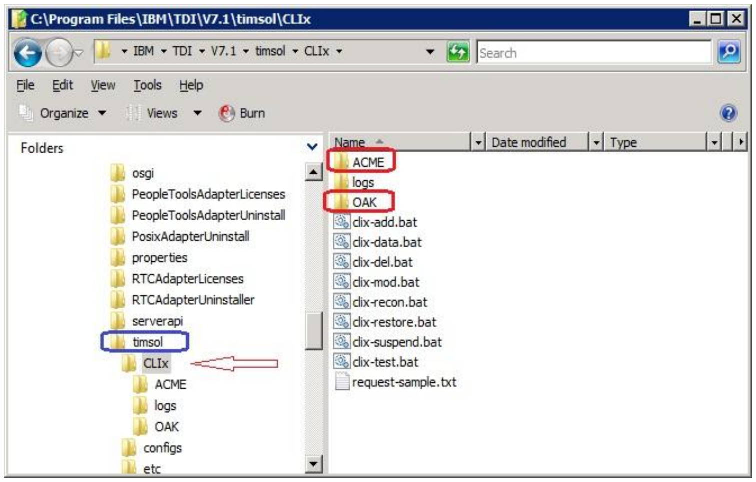


Figure 6 - CLI Sample adapter installation

Figure 5 shows the following:

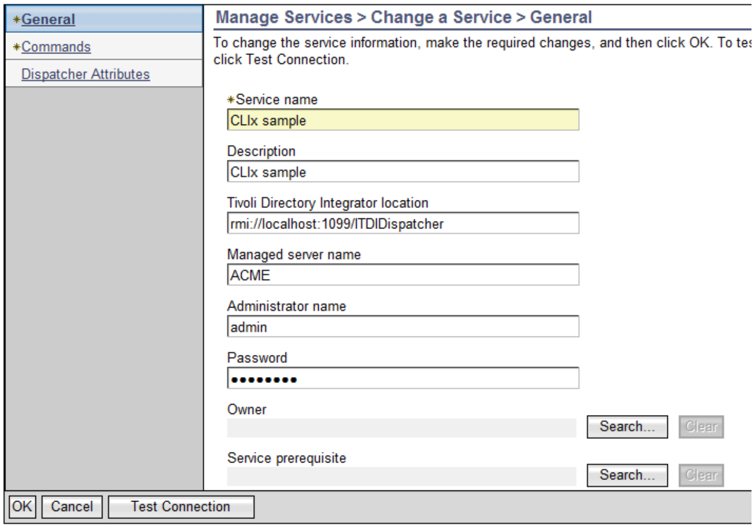
- timsol = the adapters solution directory.

- CLIx = CLI Sample adapter installation folder.
- ACME = an instance of the managed resource.
- OAK = an instance of the managed resource.

Run the CLI Sample adapter

- From the “managed services” on the Identity Manager, create a “Sample Command-line profile” service instance.

The service form has three panels. Supply the information as displayed below (leave the Dispatcher Attributes panel blank for now). Hit the “Test Connection” button to confirm that the adapter can execute the command line script. On ISIM UI the adapter service form is expanded to display the new attributes. There is an additional panel “Status and Information “on Service form for ISIM UI.



Manage Services > Change a Service > General

To change the service information, make the required changes, and then click OK. To tes
click Test Connection.

*Service name
CLIx sample

Description
CLIx sample

Tivoli Directory Integrator location
rmi://localhost:1099/ITDIDispatcher

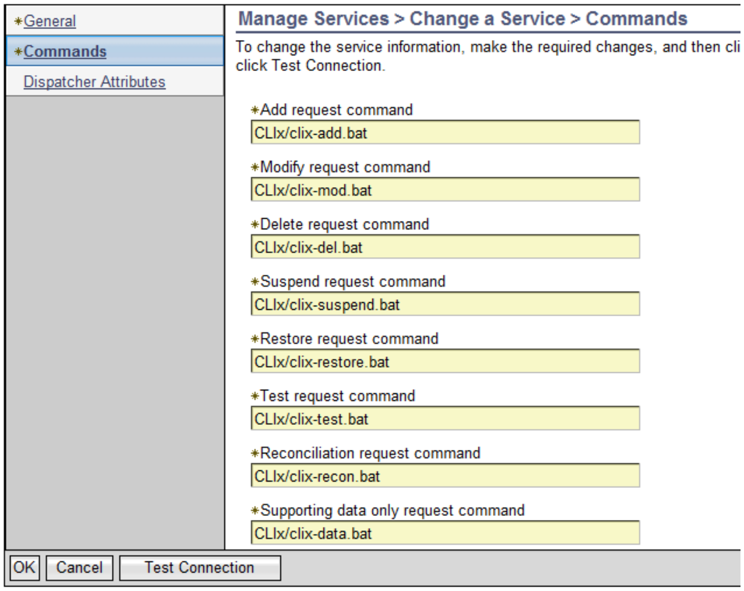
Managed server name
ACME

Administrator name
admin

Password
••••••••

Owner

Service prerequisite



Manage Services > Change a Service > Commands

To change the service information, make the required changes, and then cli
click Test Connection.

*Add request command
CLIx/clix-add.bat

*Modify request command
CLIx/clix-mod.bat

*Delete request command
CLIx/clix-del.bat

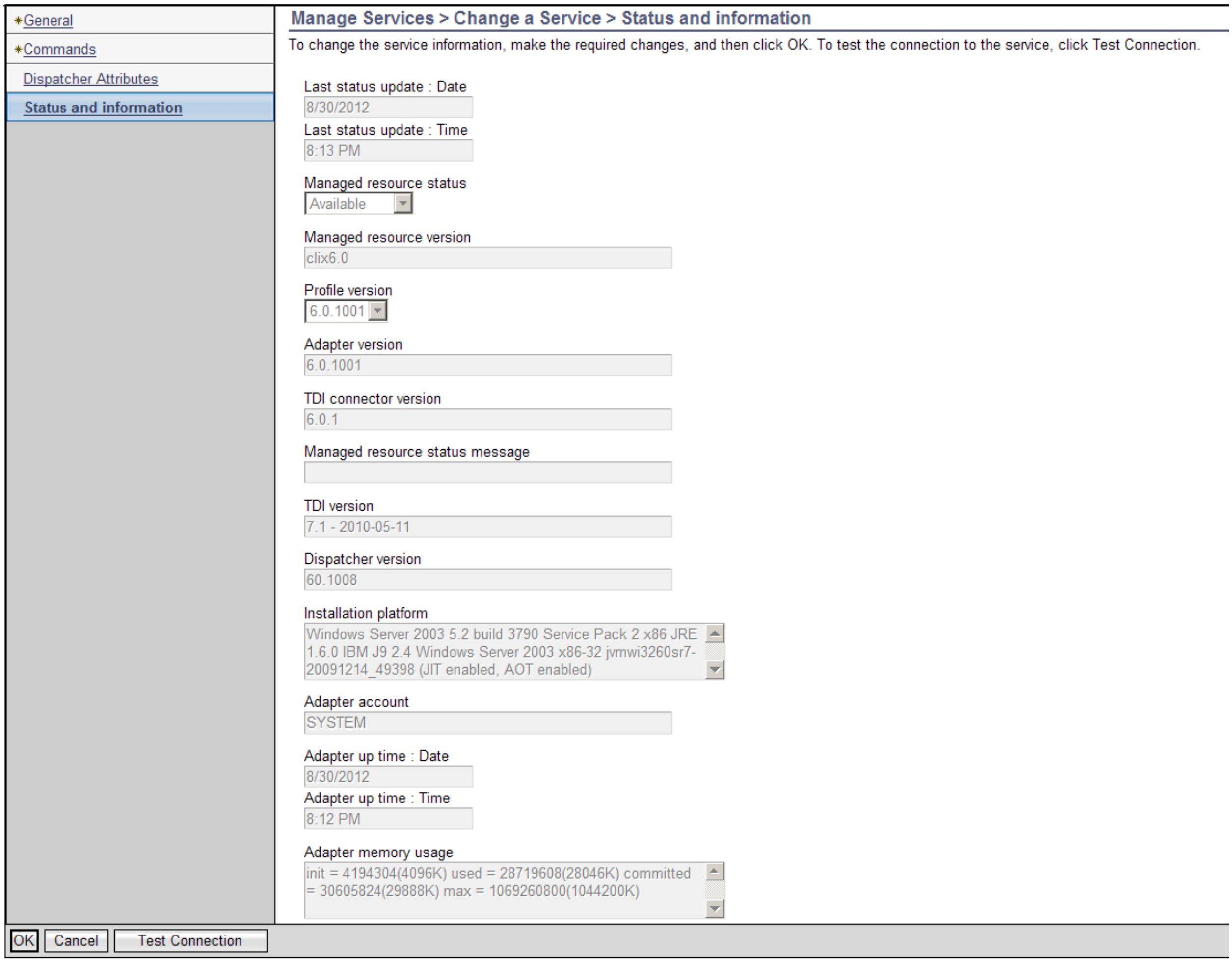
*Suspend request command
CLIx/clix-suspend.bat

*Restore request command
CLIx/clix-restore.bat

*Test request command
CLIx/clix-test.bat

*Reconciliation request command
CLIx/clix-recon.bat

*Supporting data only request command
CLIx/clix-data.bat



Manage Services > Change a Service > Status and information

To change the service information, make the required changes, and then click OK. To test the connection to the service, click Test Connection.

Last status update : Date: 8/30/2012
Last status update : Time: 8:13 PM
Managed resource status: Available
Managed resource version: clix6.0
Profile version: 6.0.1001
Adapter version: 6.0.1001
TDI connector version: 6.0.1
Managed resource status message:
TDI version: 7.1 - 2010-05-11
Dispatcher version: 60.1008
Installation platform: Windows Server 2003 5.2 build 3790 Service Pack 2 x86 JRE 1.6.0 IBM J9 2.4 Windows Server 2003 x86-32 jvmwi3260sr7-20091214_49398 (JIT enabled, AOT enabled)
Adapter account: SYSTEM
Adapter up time : Date: 8/30/2012
Adapter up time : Time: 8:12 PM
Adapter memory usage: init = 4194304(4096K) used = 28719608(28046K) committed = 30605824(29888K) max = 1069260800(1044200K)

Deploy Your CLI adapter

An adapter profile is a service type on the Identity Manager server. A service type describes the managed resource and therefore each type of resource that you need to manage must have a separate adapter profile. If you want to manage multiple instances of the same resource, then only one profile is needed.

As described earlier in this document, all you need is to supply a profile and a set of command line scripts to deploy your adapter.

Create the command line scripts

Create your command line scripts files as described is section *Command Line Scripts*. Always refer to the CLI Sample adapter command line scripts for reference.

Create the profile

In this section we will go through the steps to create your profile based on the CLI Sample adapter.

1. Create a directory (i.e. cli-profiles) and extract the content of the SampleCmdProfile.jar file into it.
    - mkdir c:\cli-profiles
    - copy SampleCmdProfile.jar c:\cli-profiles
    - cd c:\cli-profiles
    - jar xf SampleCmdProfile.jar

2. Now the c:\cli-profiles directory will contain a sub-directory called "SampleCmdProfile". Rename the "SampleCmdProfile" directory under c:\cli-profiles to the same name as your profile name (this is case sensitive). For example if you profile name called Db2AppProfile, then:
    - cd c:\cli-profile
    - rename SampleCmdProfile Db2AppProfile

3. Delete the account and service form files from the Db2AppProfile directory. These are forms specific to the CLI sample adapter. You can use them as a reference, but they must be deleted from the Db2AppProfile directory.

    - cd c:\cli-profile\ Db2AppProfile
    - delete file erSampleCmdAccount.xml
    - delete file erSampleCmdService.xml

4. Update the schema.dsml file in the Db2AppProfile directory.
    - Define your specific attributes. Refer to sections *Schema.dsml file* and *CLI Sample schema (schema.dsml file)* for more details.
        - ➢You should not have any attributes with erSample prefix after you are done.
        - ➢Note that the <object-identifier> of each attribute is the attribute name with the –OID suffix. The same method can be used, or numeric values may be supplied.

    - You must use a unique account object class name (i.e. erDb2AppAccount).
        - ➢Change object class name erSampleCmdAccount to erDb2AppAccount.
        - ➢Note that the <object-identifier> of each object class is the attribute name with the –OID suffix.
        - ➢Include all your attributes to the erDb2AppAccount object class.

    - You must have a unique service object class name (i.e. erDb2AppService).
        - ➢Change object class name erSampleCmdService to yours erDb2AppService.
        - ➢Note that the <object-identifier> of each object class is the attribute name with the –OID suffix.
        - ➢Include all your service attributes to the erDb2AppService object class.

- The CLI Sample schema file contains two object classes for supporting data (erSampleCmdGroups and erSampleCmdRoles). If there are any supporting data object classes needed for you application, make the changes as you did to the account and service object classes otherwise remove erSampleCmdGroups and erSampleCmdRoles from the schema.dsml file.

5. Update the CustomLabels.properties file.
   As described in section *CLI Sample labels (CustomLabels.properties file)* this file simply contain labels for form and other Identity Manager Menu sections items.
   - Make the proper changes so that all your defined attributes have entries in the file.
   - Make sure the profile name and object class names have entries as well. Refer to section *CLI Sample labels (CustomLabels.properties file)* for more details.

6. Assembly lines file cmd.xml
   The cmd.xml file contains the TDI assembly lines needed by your CLI Adapter. You do not need to modify the content of the assembly line, but it's highly recommended to change the names of the assembly lines.
   - There are two assembly lines in the cmd.xml file: CLIManageUserAL and CLISearchUserAL.
   - CLISearchUserAL is used for reconciliation and CLIManageUserAL is used for all other requests.
   - To modify the names of the assembly lines, simply edit the cmd.xml file and replace all occurrences of CLIManageUserAL and CLISearchUserAL with unique names (i.e MyCLIUserMgr and MyCLISearchMgr).

7. Modify the service.def file
   The service.def requires the most changes. To make this easier, sections of the service.def file of the CLI Sample adapter that requires changes have been tagged with the following:

   ```
   <!-- UPDATE -->
   ```

   Search the service.def file for the above tag and perform the changes as described. Here is an example:

   ```
   <!-- UPDATE -->
   <!-- name="erSampleCmdService" - the service object class name     -->
   <!-- profile="SampleCmdProfile" - the name of your adapter profile -->
   <!-- location="erSampleCmdService.xml" - name of the xml file containing the service form -->
   <!-- ......................... -->
   <!-- service profile definition -->
   <!-- ......................... -->
   <type category="service" location="schema.dsml"
           name="erSampleCmdService" profile="SampleCmdProfile">
           <key>
   ```

```
                <field>erservicename</field>
                </key>
                <form location="erSampleCmdService.xml"/>
        </type>
```

8. Create your profile jar file
   Assuming all the above steps were done under the “cli-profiles” directory as described in step 1 above. To create your profile file, do the following:

   - cd c:\cli-profiles
   - jar cvf Db2AppProfile.jar Db2AppProfile
   - Form the “manage service type” option on the Identity Manager, import Db2AppProfile.jar profile.