

How to utilize Administration and Monitoring Console (AMC) in your TDI solution

An overview of the basic functions of Tivoli Directory Integrator's Administration and Monitoring Console and how it can be used in conjunction with any AssemblyLine.

**Written by
Jonathan Elwood**

**Edited and reviewed by
Eddie Hartman**

February 2009

Introduction

It was only when I went to try and use AMC as part of my TDI solution that I discovered the distinct lack of documentation available. While there are many “How-to” guides available along with Eddie Hartman's detailed video tutorials, the Administration and Monitoring Console has been largely ignored. AMC should not be considered as an after-thought; it is a simple yet powerful tool provided with TDI.

This tutorial seeks to fill in the knowledge gaps that exist in the TDI documentation and provide detailed guides on the following topics:

- Installation and running AMC
- Enabling your solution to function with AMC
- Running an AL that can be accessed via AMC
- Controlling an AL via the web interface
- Adding a Health AL to your solution.

N.B. This tutorial uses TDI v6.1.1, installed on Windows XP. The Assembly Line used is purely for illustration purposes and the content thereof does not affect the AMC in any way.

Installation and running AMC

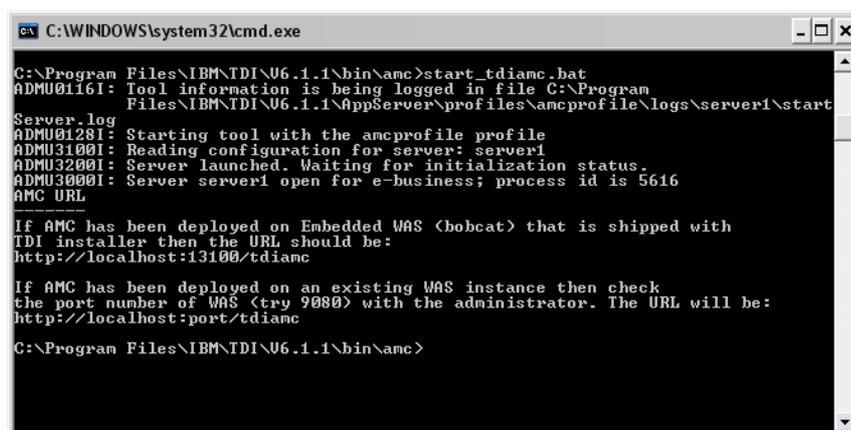
AMC is a web-based interface that uses various remote APIs to communicate with one or more TDI servers. The TDI installer installs WebSphere Express upon which AMC can be run. However, you can install AMC on any installation of WebSphere Application Server (WAS). There are detailed instructions in the TDI user guide on how to install AMC.

This can be found here: http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/index.jsp?topic=/com.ibm.IBMDI.doc_6.1/adminguide11.htm

If you have already installed AMC, or have successfully completed the installation steps in the manual, you are ready to start your AMC server.

To start AMC using bundled WebSphere Express:

1. Navigate to your TDI installation directory. e.g. C:\Program Files\IBM\tdi\V6.1.1
2. If your AMC installation was successful you should have the directories \AppServer and \amc which contain WebSphere Express and the AMC war file respectively.
3. Navigate to <installation directory>\bin\amc. You will see a selection of batch files.
4. To start Websphere Express and start AMC, run the batch file: start_tdiadc.bat
5. The server may take a while to start but once it does you will see a screen similar to the one below:



```

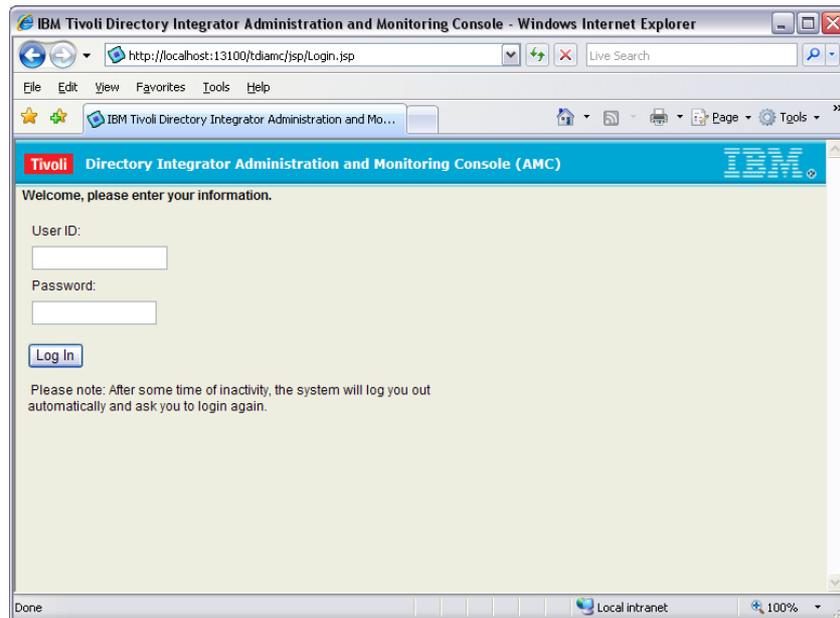
C:\WINDOWS\system32\cmd.exe
C:\Program Files\IBM\TDI\06.1.1\bin\amc>start_tdiadc.bat
ADMU0116I: Tool information is being logged in file C:\Program
Files\IBM\TDI\06.1.1\AppServer\profiles\ancprofile\logs\server1\start
Server.log
ADMU0128I: Starting tool with the ancprofile profile
ADMU3100I: Reading configuration for server: server1
ADMU3200I: Server launched. Waiting for initialization status.
ADMU3000I: Server server1 open for e-business; process id is 5616
AMC URL
-----
If AMC has been deployed on Embedded WAS (bobcat) that is shipped with
TDI installer then the URL should be:
http://localhost:13100/tdiadc
-----
If AMC has been deployed on an existing WAS instance then check
the port number of WAS (try 9080) with the administrator. The URL will be:
http://localhost:port/tdiadc
C:\Program Files\IBM\TDI\06.1.1\bin\amc>

```

N.B. You can copy and paste the link for the AMC home page from here if required.

6. To stop the server, simply run the stop_tdiadc.bat batch file in the same directory.
7. Open an internet browser and enter the URL to AMC. e.g. <http://myserver.ibm.com:13100/tdiadc>. When this

page loads you should be able to see the login window below:



The default user name is “superadmin” and the default password is “secret.” It is recommended that you change these. Detailed instructions on user management, connection to an LDAP server and other security settings can be found in the TDI documentation.

8. Once you have logged in you can then proceed to use AMC! However, before you continue, you must perform some simple configuration to your assembly line.

Enabling your solution to function with AMC

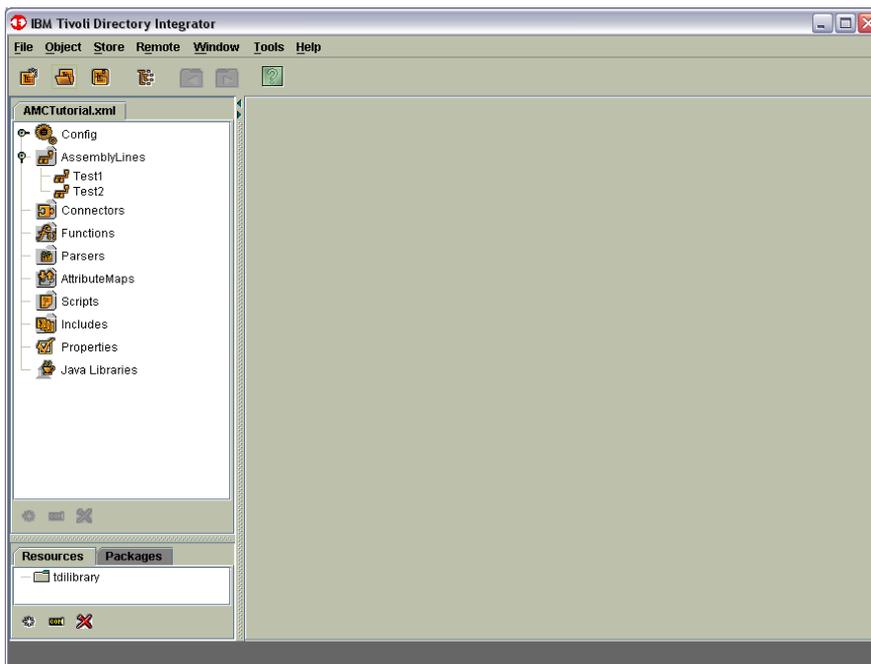
Suppose we wish to use AMC for the following tasks:

- Start/Stop Assembly lines running on a server
- View/change properties
- View assembly line logs

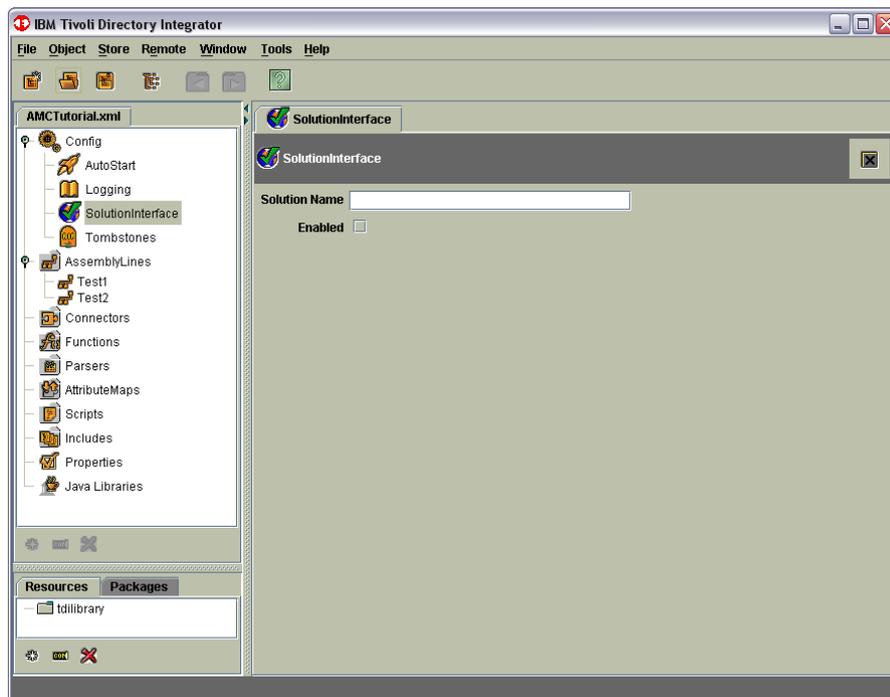
Each of these three features requires some configuration. The solution is bundled in AMCTutorial.zip if you wish to test it before trying it on your own TDI solution.

The “Solution Interface” settings that are configured in the following steps are there to let a developer decide what gets published in AMC, instead of leaving it to whoever sets up the config monitoring on the AMC console. (See *Controlling an AL via the web interface* steps later in tutorial). It is not 100% necessary for allowing AMC to function correctly but it does simplify the process of setting up a config view/monitor using the web interface. You can also give your config a simple mnemonic label. If you don't do this, TDI/AMC will generate a name using the path to the config. It will replace all special characters (: & and \ for example) with underscores. This makes it very hard to work with certain features of AMC, such as the Action Manager (AM). (The AM allows you to generate a set of rules so that assembly lines run based on certain events. e.g. an AL may start as soon as another has finished).

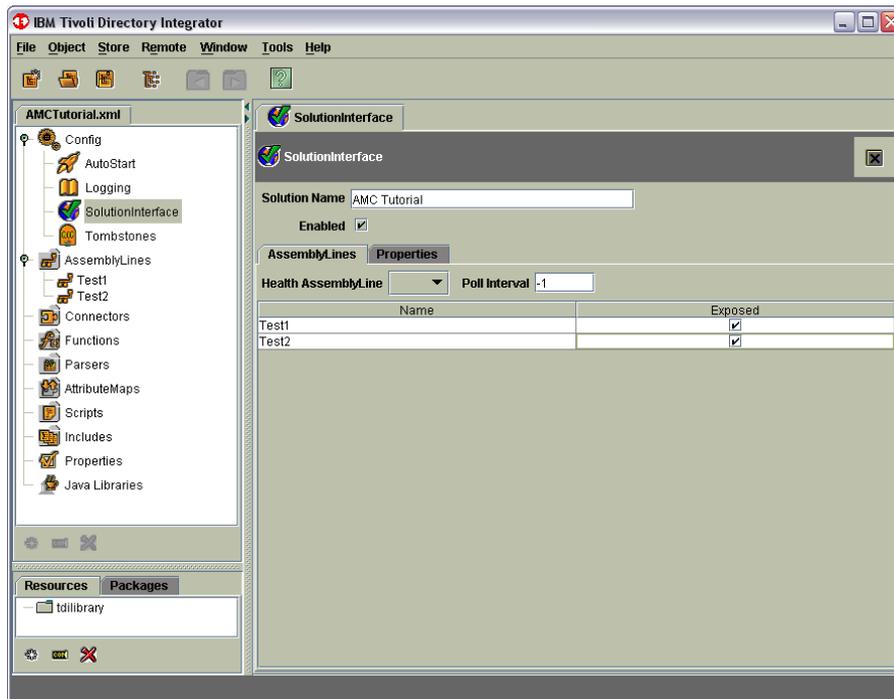
1. The first step in allowing a TDI solution to be accessed via AMC is to allow AMC to connect to it. Open TDITutorial.xml (or your solution). You will see the following screen:



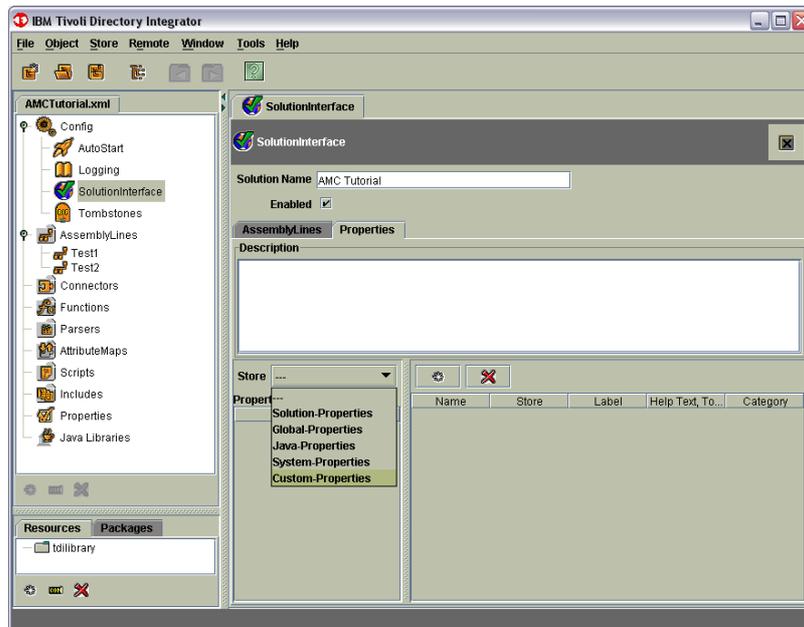
2. Expand "Config" and select "SolutionInterface" from the left hand window. You will see the following window.



3. Enter a name for your solution in the "Solution Name" text area and ensure the "Enabled" tickbox is checked. This is the name that will appear on AMC with reference to your server.
4. Once you select "Enabled" you will be presented with a table of Assembly Lines that exist in your solution. You can then choose which you wish to make "public" to AMC. In our case we want to share both.

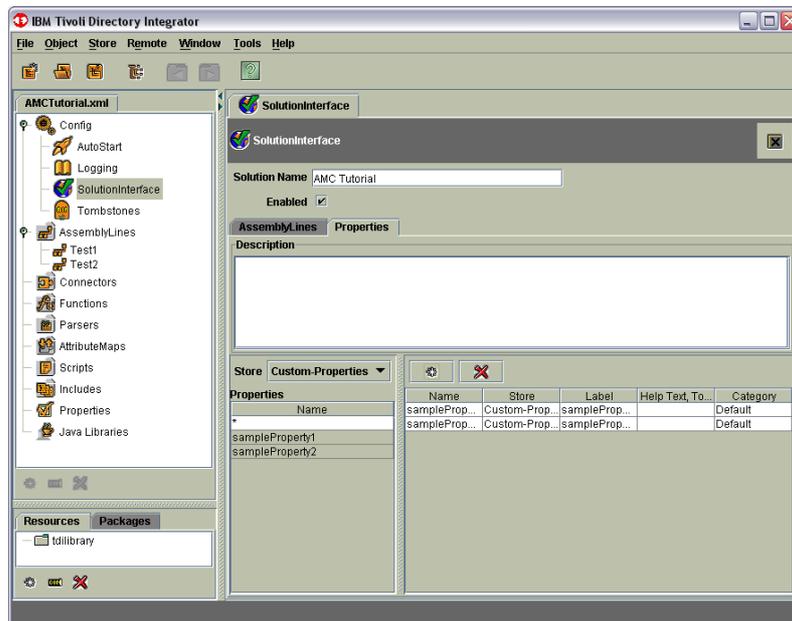


5. Ignore the “Health AssemblyLine” options (This will be covered in a separate section at the end of this tutorial) and save your work.
6. Next we will configure our properties. In the example solution, a basic properties file has been created. You can view this by clicking on “Properties” in the solution explorer on the left hand side. Select the “Properties” tab. At the bottom of this tab there are two frames. In the left you can choose a property store. Using the pull-down box. In our case we will select “Custom-Properties”
N.b. If you have problems accessing the properties file, select “Properties” from the left hand menu. Choose Custom-Properties and select the “Connector Tab” - Check the URL to the properties file. All paths are relative to the solution directory.



7. This will populate the left hand frame with all the properties that exist in that property store. We can then select which properties we want to make publicly exposed to AMC. We can select the “*” wild card if we want to make all the properties in that store available on AMC. Or we

can select single or multiple properties. Click the  button to add the selected property to the right hand pane.

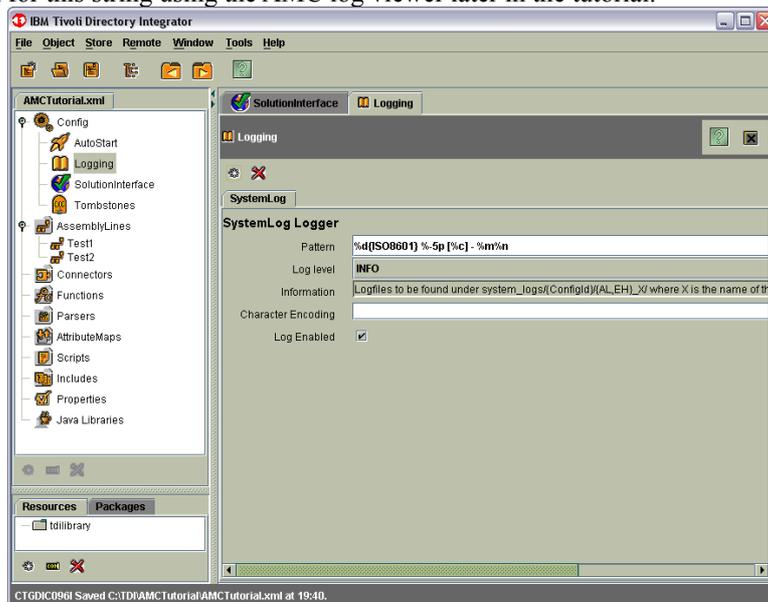


8. To expose logs to AMC, select “Logging” from the left hand solution explorer.
9. Select the  and choose “SystemLog”. You are free to use other loggers as well, but only logs created by the SystemLog can be viewed via the AMC. Feel free to change the format and settings of this log, but the defaults are usually adequate. You can add as many loggers as you need for your solution.

Any `task.logmsg("string")` line will be logged to the SystemLog combined with other TDI server logging. In our example we have two simple strings printed out using script components:

```
task.logmsg("This is a sample assembly line called: " + task.getName());
```

We will search for this string using the AMC log viewer later in the tutorial.



Finally, in your solution properties find the property: “api.remote.on”. If it doesn't exist, create it. And set it

to: "api.remote.on=true". This will allow AMC to access the remote APIs.

There are also two settings that relate to SSL:

```
api.remote.ssl.on=false
api.remote.ssl.client.auth.on=false
```

By setting these two settings to "false" we can disable SSL. This can often cause problems when getting AMC to work so disabling this will eliminate a potential problem (for example, if other Java security libraries are currently running on a users PC causing a conflict). This can of course be enabled again once the solution is working correctly. There are other port related settings in solution.properties. If, for example, you attempt to start your ibmdisrv.bat process and get a port conflict for the remote API, you can change it using the "api.remote.naming.port" property.

Upon completion of these steps your solution should be capable of being able to start/stop an assembly line, view/edit properties and view log files via AMC.

Next we want to test our config on our running instance of AMC by starting a TDI server in daemon mode.

Running an AL using daemon mode

Usually we would execute the following command to start an AL:

```
"c:\Program Files\IBM\TDI\V6.1.1\ibmdisrv.bat" -c AMCTutorial\AMCTutorial.xml
```

However, if we want to be able to control when the assembly line starts and stops we should include the *-d* option which starts the server in daemon mode (unless ALs are configured to AutoStart):

```
"c:\Program Files\IBM\TDI\V6.1.1\ibmdisrv.bat" -c AMCTutorial\AMCTutorial.xml -d
```

The path to the configuration file is relative to the solution directory. If your solution is stored in a folder within your solution directory you will need to include this in the relative path to the config file.

When you run this command you should see the following output:



```
Shortcut to cmd.exe - "c:\Program Files\IBM\TDI\V6.1.1\ibmdisrv.bat" -c AMCTutorial\AMC...
G:\TDI>"c:\Program Files\IBM\TDI\06.1.1\ibmdisrv.bat" -c AMCTutorial\AMCTutorial.xml -d
CTGDKR024I Remote API successfully started on port:1099, bound to:'SessionFactor'. SSL and Client Authentication are enabled.
```

If you don't get this output, check your settings again, in particular your solution.properties file (check for "api.remote.on=true").

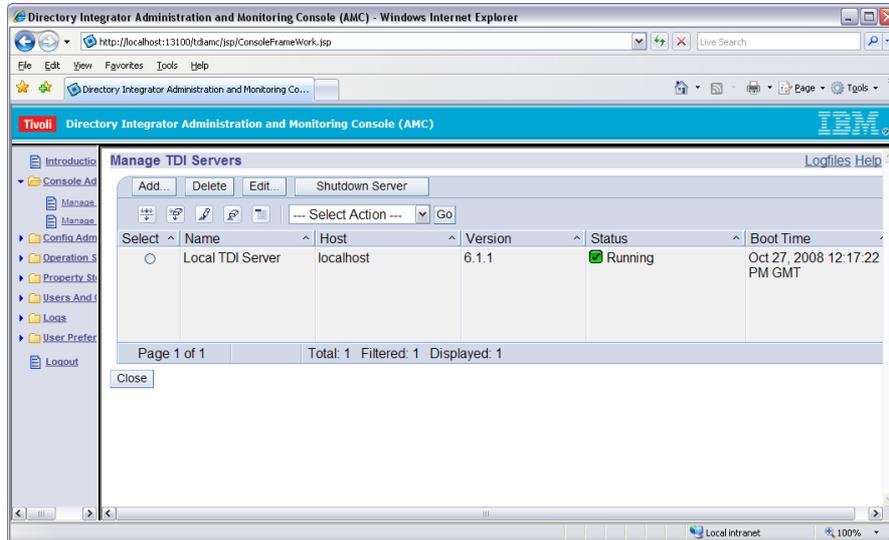
Controlling an AL via the AMC web interface

In this final section of the tutorial we will provide an overview of some of the basic functions of AMC. In particular we will look at:

- Adding a server to AMC
- configuring AL properties

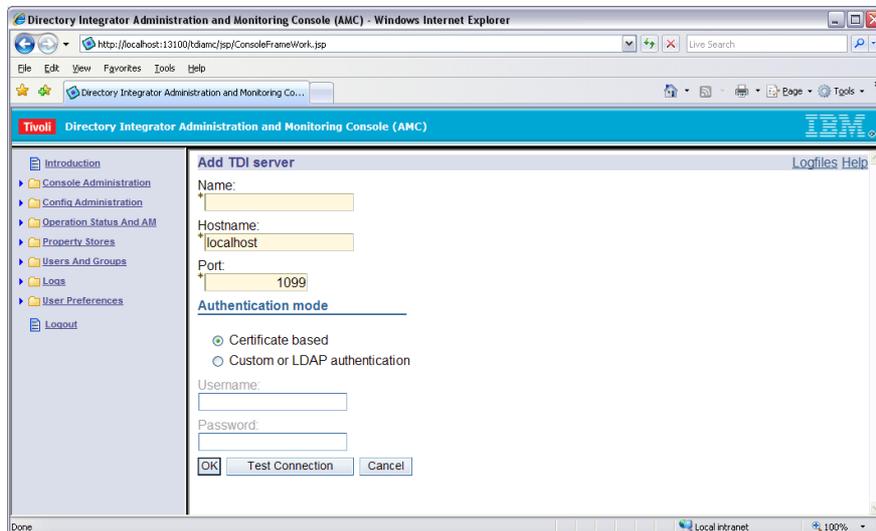
- starting and stopping an AL
- logging functions.

When you login to AMC, select “Console Administration” on the left hand menu. Then select “Manage TDI servers.” This will bring up the following screen:



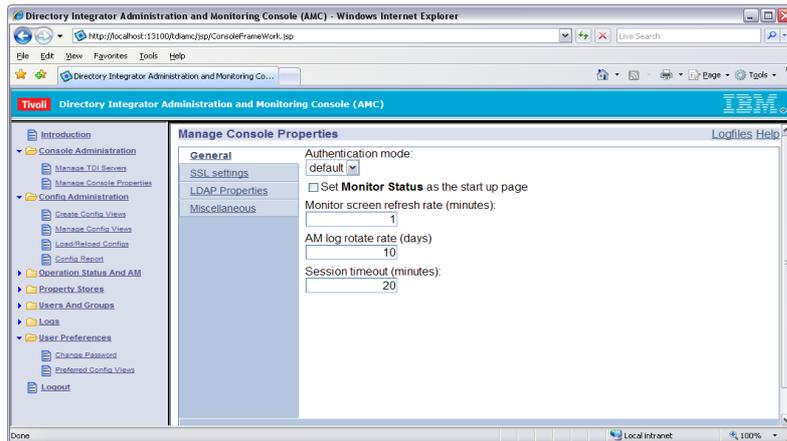
If your AMC console is installed on the same machine as your running TDI server, you may not need to do any configuration. If this is the case you will probably have a “Local TDI Server” on “localhost” listed.

To add a new server, click on the “Add..” button at the top of the table. Complete the form with the relevant details. In most cases this may simply be a case of entering a server name.

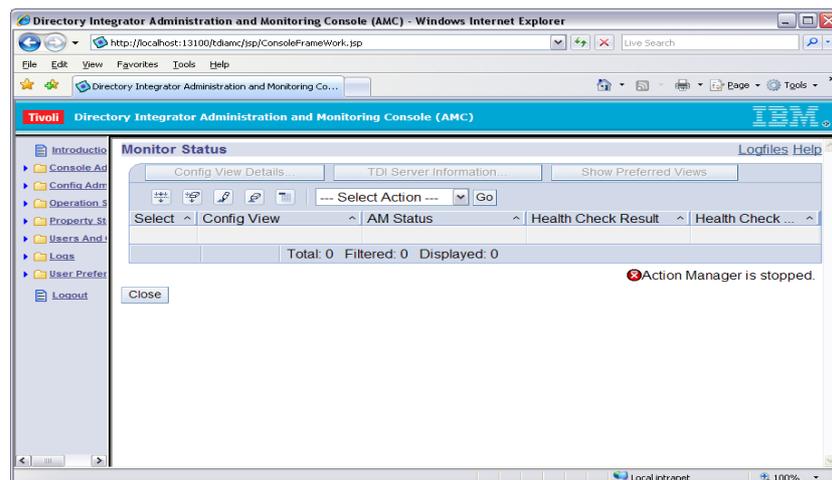


The “Test Connection” button is very useful in ensuring the details you have entered are correct! You can add as many servers as you require. Each will be listed in the table with the status reflecting the state of each server. Usually this will be “Started” or “Stopped” but it can also indicate if there are any errors.

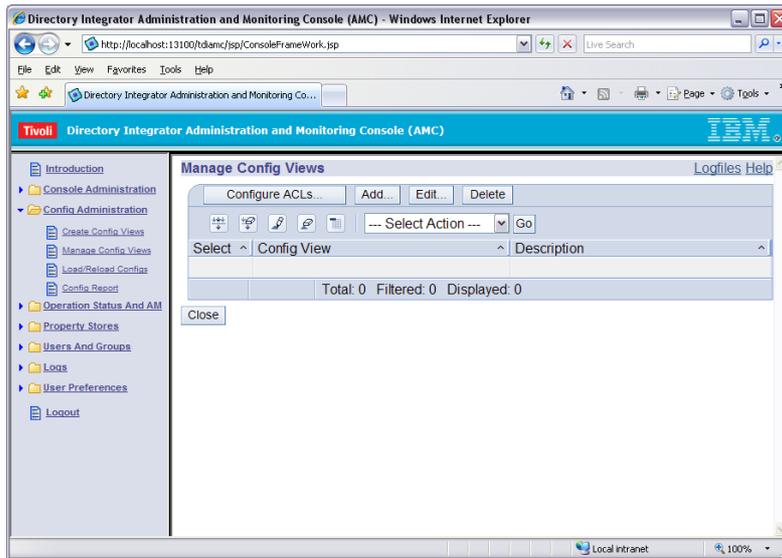
There are other options available relating to configuration of AMC but for the minute we only want to look at one more. Click on “Console Administration” in the left hand menu. Then select “Manage Console Properties” There are a number of options available here but for most instances the defaults are sufficient. I would recommend that you change some of the settings on the “General” tab.



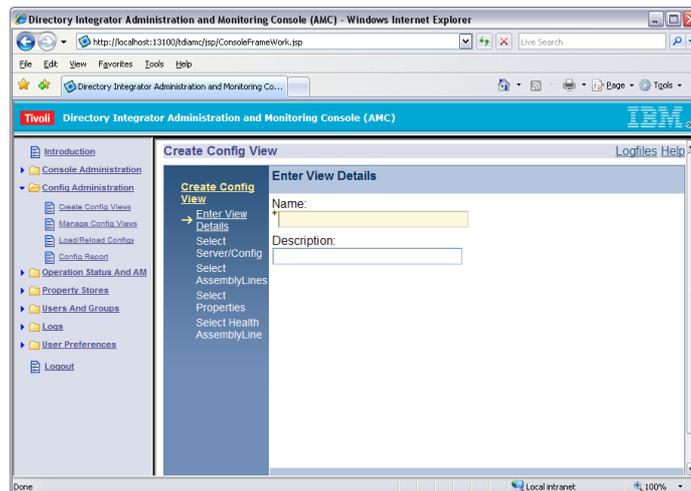
Change the value of “Monitor Screen Refresh rate” to “1” or “2”. This means the screen will automatically refresh, updating the status more regularly. If your assembly line only takes 5 minutes to complete, you would not benefit from an refreshed screen every 15 minutes! Optionally, you can choose to select the “Set Monitor Status as the start up page”, The “Monitor Status” page is where we will do most of our work, as we will see later in the tutorial. It is useful to have this as the start-up page as we can instantly monitor the status of all our servers on login. This is purely a convenience function. Ensure you click “OK” after making any changes to save them. Now, lets take a look at that Monitor Status. Select “Operation Status and AM” from the menu then choose the “Monitor Status” option. You will see the following screen:



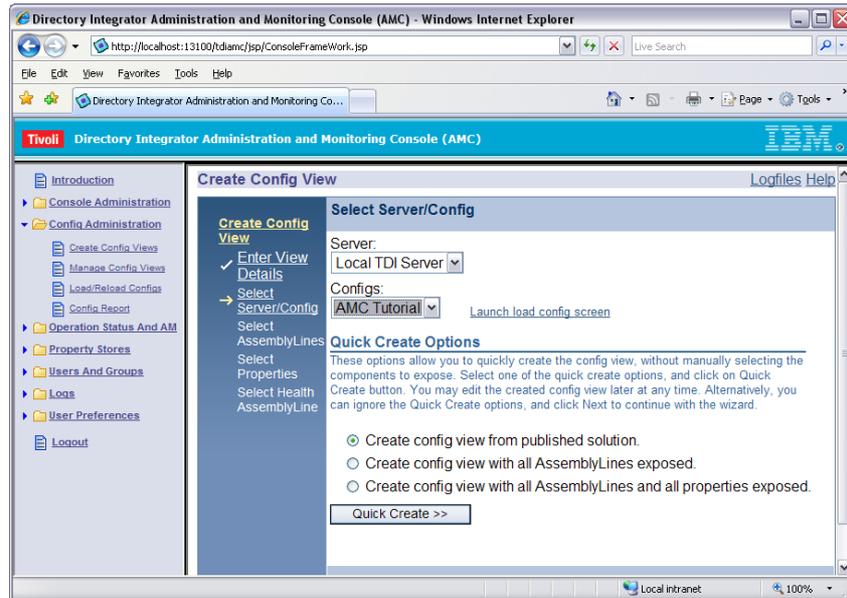
You will notice that the table is empty. We need to create and customize a “config view” before we can perform any further AMC operations. A config view is a customized view of your solution. It will give the user various controls depending on what properties/assembly lines you have in your solution. To do this: Select “Config Administration” from the menu and “Manage Config Views”. You will see a table similar to the one below:



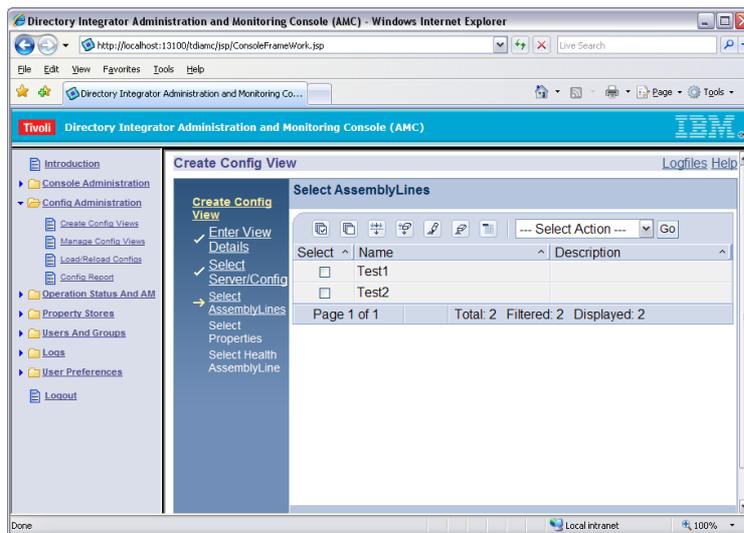
Click “Add...” to create a new config view. Enter a name for your config view. In this example, we will use “AMC Tutorial”



Click “Next” and check your server and config details.

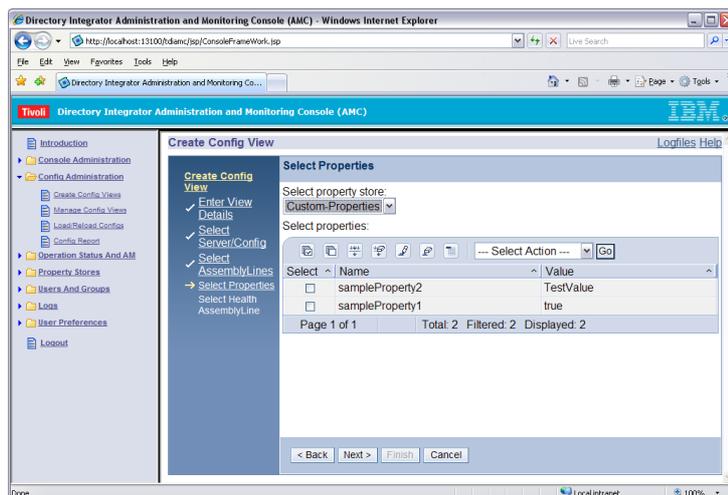
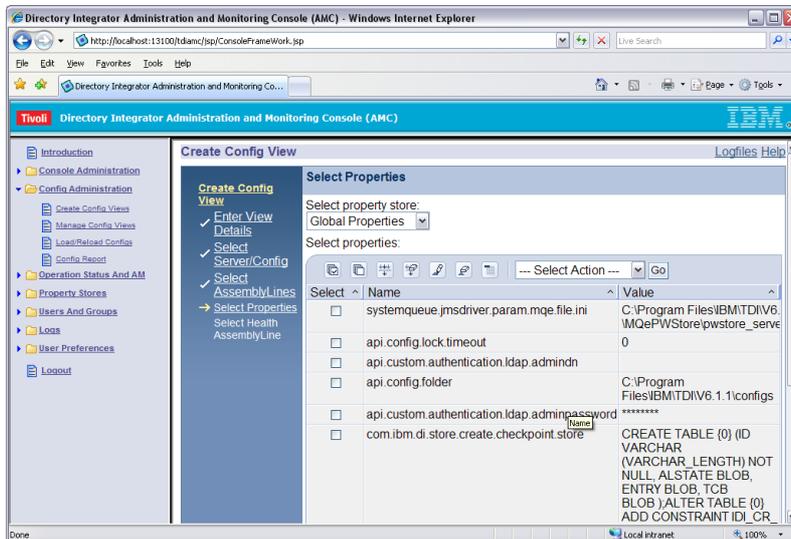


The “Quick Create” option is useful but takes a little bit of control away from the user. If you have followed the steps to configure the solution interface, then the configuration you did there, will be used to create the config view when you click “Quick Create.” This would usually be the best option, but for the sake of this tutorial we will step through the wizard manually and examine each configuration option. Click “Next” to continue.



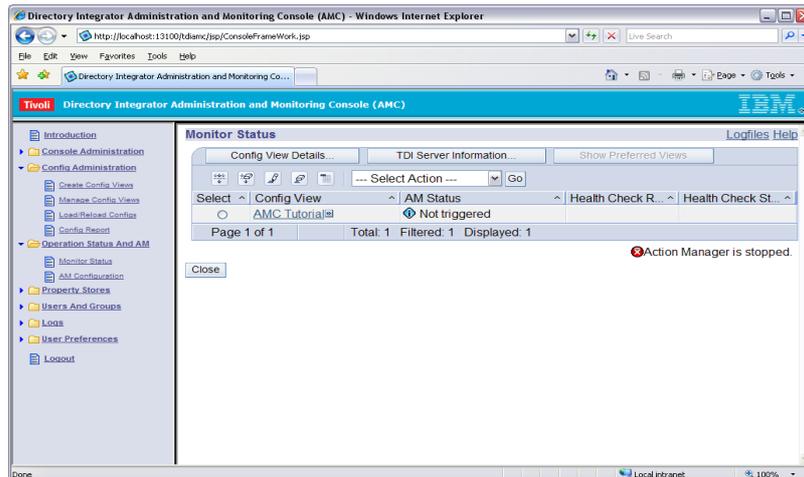
In this screen we can choose what assembly lines we want to make available in the config view. Even though we have made all of the assembly lines available (through the earlier steps in the config editor) we can still limit what certain users have access to by creating a config view with only a subset of assembly lines. In this example we will select both of the assembly lines. Click “Next” to continue.

Now we get to choose which properties will be available for viewing/editing in this particular config view. The default property store seems to be “Global Properties” but we only wish to make the “Custom-Properties” available. Select this from the pull down list.



We can now see the two dummy properties that were created via the config editor at the start of the tutorial. To make both of these available in the config view, select them both. Then click “Next” to continue.

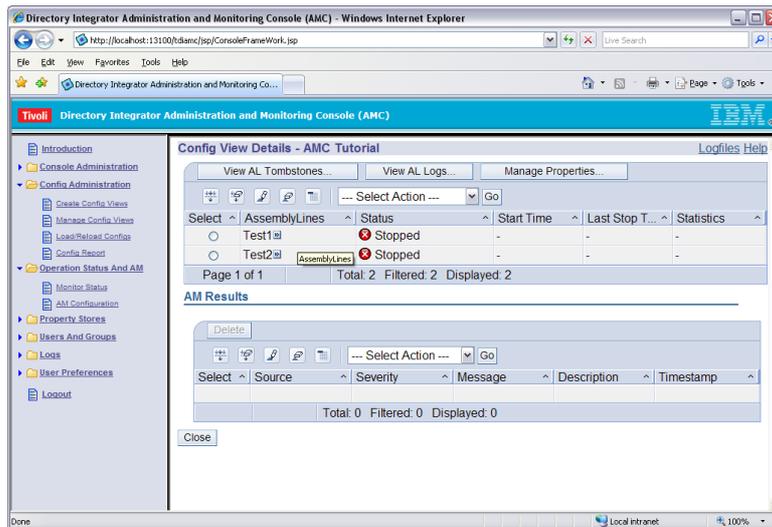
The final window in the wizard relates to Health AssemblyLines which is covered separately at the end of this tutorial. So click “None” on this window and “Finish” to save and create the Config View. This time, when you go to “Operation Status and AM” and click on “Monitor Status” you will see the config view that you have just created:



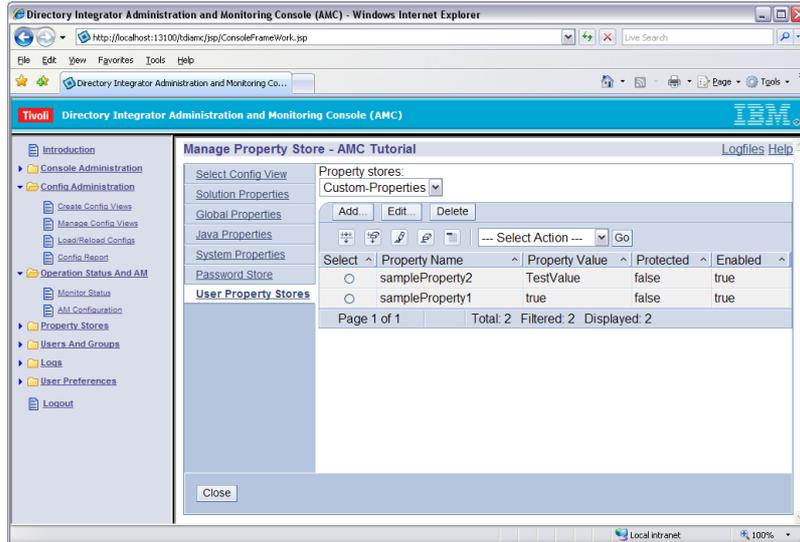
Finally, we are in a position where we can run our assembly line, edit properties and look at logs.

Managing Assembly Line Properties

Suppose we wanted to modify a property, prior to running an assembly line. Following our earlier configuration this is now very simple. In the “Monitor Status” window click on the config view you wish to configure/run. In our case this is “AMC Tutorial”. You will then see a table with a list of assembly lines similar to the screenshot below:



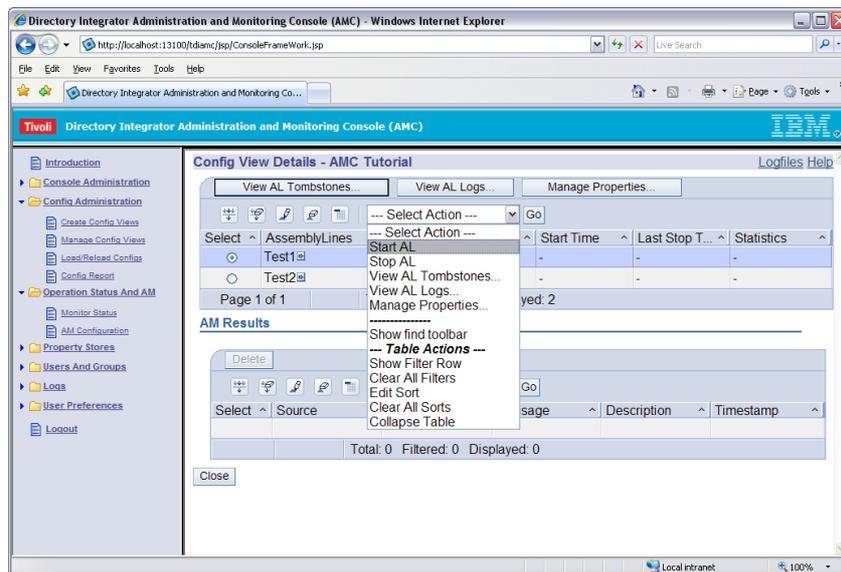
To manage the properties, simply click the manage properties button at the top of the table. You will then be able to edit the properties. In the following window, the properties that were selected while configuring the config view will be listed under “User Property Stores”. You can select any property and edit it or even add/remove properties.



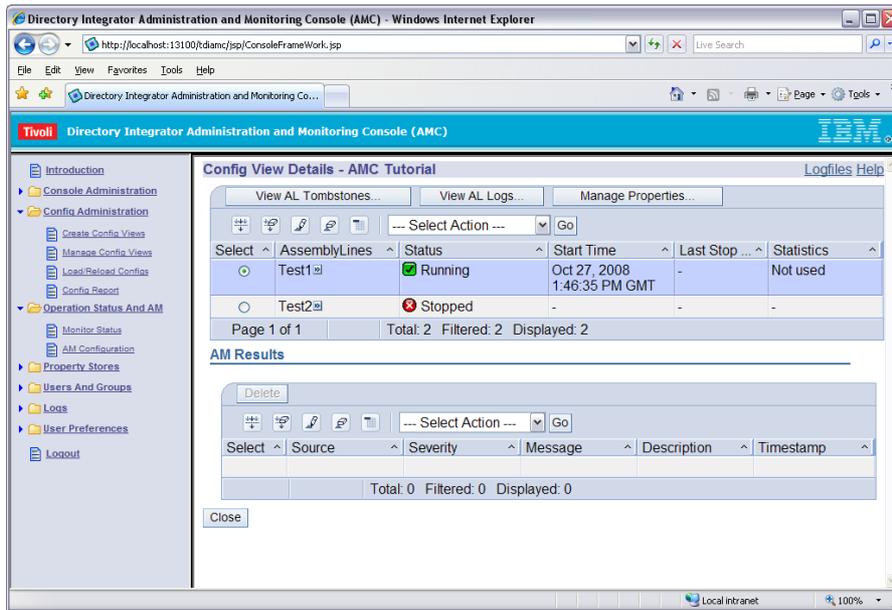
When you are happy with any changes, click “Close” to return to the list of assembly lines.

Starting / Stopping an Assembly Line

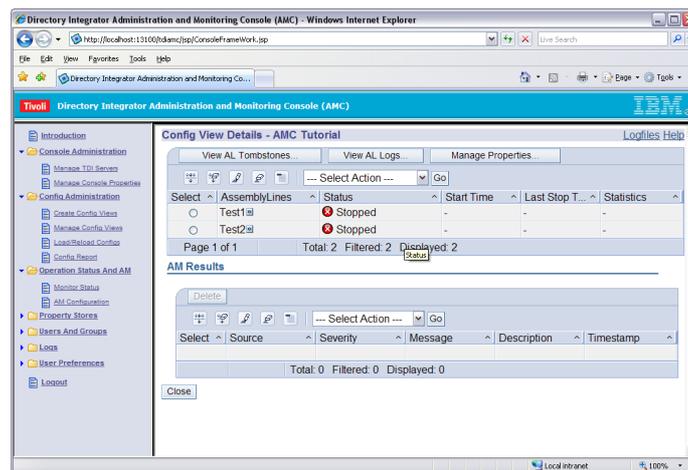
Suppose we wish to start the assembly line called “Test1”. Firstly we must select our assembly line, then we can choose “Start AL” from the pull down menu and click “Go”. This is shown in the screenshot below. There are also options in this pull down menu for viewing logs, managing properties and stopping an AL. To stop an AL running, we simply use the “Stop AL” option. “Manage properties” has already been covered and we will look at the logging later in the tutorial.



Once the AL is running, the status will reflect this as shown in the screenshot below:



However, as we configured earlier, the page will refresh every minute to update the status. As the assembly line should complete in a matter of seconds (it is only printing a line to the logs!), after the minute it should show a stopped status:

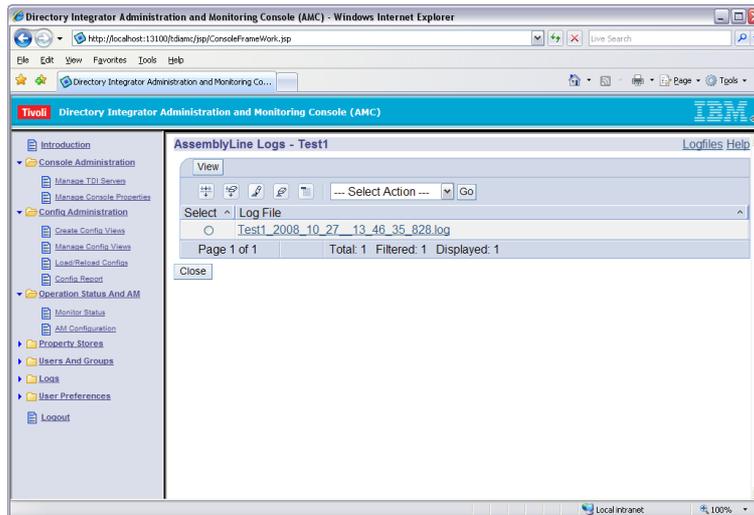


We can do the same for “Test2” if required!

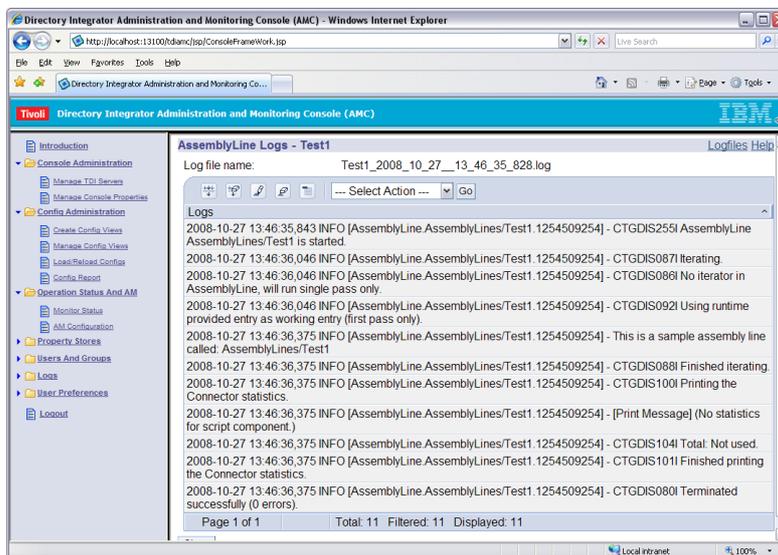
Viewing Logs

After running our assembly line log files should have been created. In particular we should have a log with our custom message in it: *This is a sample assembly line called...*

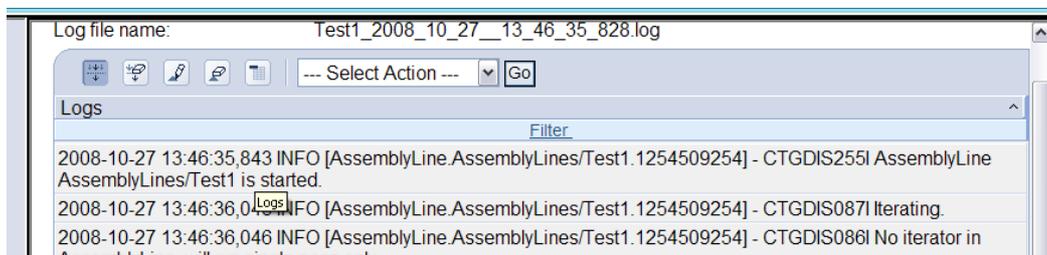
To view the logs for a particular assembly line, simply select the assembly line, and click on “View AL Logs...”. You will see a screen similar to the one below, where all the log files are listed. The date and time that the AL was run will be included in the log file name.



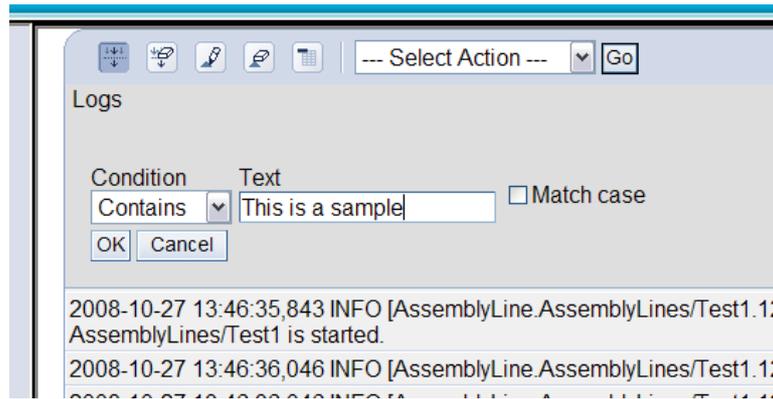
To view a log simply select it and click view.



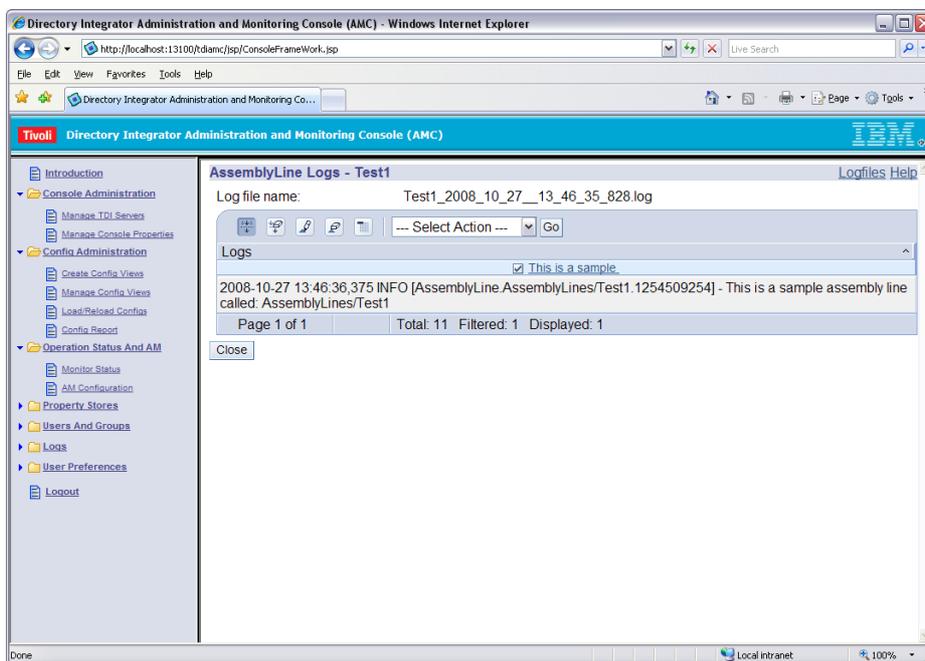
In this case the log is very short. A longer log would have been split into pages so we would have to navigate through the pages. This can be very difficult if there are hundreds of pages/the log is very long. So we can actually filter the logs. To do this, select the filter button (first icon at the top of the table). This will add a filter line to the top of the table. (See screen shot below).



Click the “Filter” link to bring up the search options.



If we search for a portion of our custom string (“*This is a sample assembly line called...*”) and click “OK” we get a list of all the lines in the log that contain that string:



In our case there is only the one line. We can use this as a basic way of searching a log for successes or perhaps “ERROR” messages.

Finally, if we want to clean up our logs, there is a useful self explanatory tool under the “Logs” menu option which allows us to clean up logs before a certain date. Alternatively we can delete all logs but leave the last n logs.

Using a Health AL

A Health Assembly Line is used to monitor the health of a solution. The Action Manager runs the Health AL every so often, like a heart beat. The status is returned and displayed on the AMC. This can be useful to get a quick update on the state of your solution. For example, suppose an assembly line has a global variable “errorCount”, initially 0, which updates every time there is an error. Our example HealthAL will query this variable and display the count on the AMC. Any value other than 0 will be shown as an error. Open “HealthALTutorial.xml” and examine the assembly line. The Health AL stores the status of the “set Error Count” assembly line in two variables:

- heathAL.result
- healthAL.status

These two attributes must exist in the final work entry of the assembly line if the AM is to successfully retrieve the heartbeat/health. If the value of healthAL.result matches the name of an image in

\AppServer\profiles\amcprofile\installedApps\DefaultNode\tdiamc.war.ear\tdiamc.war/resources\amc_images\healthAL

then that image will also be displayed. In our case we use the two default images:

- ✓ OK.gif
- ⊗ NotOK.gif

Running the example

To start the AM – there is a batch file called startAM.bat (in <installation directory>/bin/amc directory). Also, there should be an “Action Manager” directory here. Inside this directory there are a number of property files where logging and other behaviour of the AM can be configured. A useful property is found in am_config.properties:

```
# Controls the frequency of triggering the Health ALs
com.ibm.di.amc.am.healthAL.interval.time=5
```

This property controls how often the HealthAL runs and returns a result to the AMC. If logging is configured this may create an excessive amount of log files. Change the value (in seconds) to suit your individual need. Once the AM has started the AMC should indicate this on the Monitor Status page:



If this message is not displayed, check AM is started and also if there are errors in the AM logs. Create a config view, selecting “Health” as the HealthAL in the wizard:

Edit Config View

- ✓ [Select AssemblyLines](#)
- ✓ [Select Properties](#)
- [Select Health AssemblyLine](#)

Select Health AssemblyLine

Select an health AssemblyLine to monitor heart beats. This is used by the Act

None

Select	Name	Description
<input checked="" type="radio"/>	Health	
<input type="radio"/>	set Error Count	

Page 1 of 1 Total: 2 Filtered: 2 Displayed: 2

Select	AssemblyLines	Status	Start Time	Last Stop Time	Statistics
<input type="radio"/>	Health	⊗ Stopped	-	-	-
<input type="radio"/>	set Error Count	✓ Running	Feb 5, 2009 12:02:14 PM GMT	-	Not used

Page 1 of 1 Total: 2 Filtered: 2 Displayed: 2

Health Details

Health check result	Health check status
✓ OK	No Errors Found

When you run “set Error Count”, the AM will query the HealthAL every so often and return the various results. For no errors:

If the errorCount variable is above 0, the following message appears:

Health Details

Health check result	Health check status
 NotOK	Error Count: 1.0

You can change the images and also the string that is returned in healthAL.status to suit your individual needs. But the general process is the same.

Summary

In reality there is not much work involved in configuring and running a TDI solution using the Administration Console. While the features offered are limited and basic they are a powerful way of implementing a quick interface for prototyping or projects where there are no resources for UI development. The AMC boasts many other features including multiple users with individual Access Controls, an inbuilt Scheduler (Action Manager) and the capabilities of using an external DB or LDAP server.