HP Service Center 6.x to uCMDB 6.x

Integration Deployment Guide

Version 2.1, 20 July 2007

Introduction

Document Scope

This document describes technically how to deploy the integration between the HP Universal CMDB (uCMDB) and HP Service Center (SC).

The purpose of the integration is:

1. To enable the exchange of CIs between the uCMDB and SC CMDBs.
2. To establish a framework that will simplify customization.

Integration Components and How they Work Together

uCMDB uses discovery patterns to read and write XML files.

Service Center's API is called by **Connect-IT** to read and write CIs.

Connect-IT also reads and writes the XML files used by **uCMDB**. **Connect-IT** mapping connectors map selected CI types and attributes between the models.

Versions and Prerequisites

This documentation is version specific and covers the following product versions:

- The uCMDB 6.5 and 6.6
- Connect-IT 3.7.1
- Service Center 6.1 and 6.2

The reader/deployer should be familiar with at least the basics of these technologies and their configuration and operation. You must refer to the product documentation for installation and deployment details. All products should be installed, licensed, and operational prior to beginning integration installation. The uCMDB section has a introductory section explaining the basic terms and architecture.

For information on enhancing / extending this integration, refer to the uCMDB – SC Integration Enhancement Guide (provided separately with this document).

For a more general, less technical overview of this integration, refer to the uCMDB – SC Integration Overview document (provided separately with this document).

Integration Physical Packaging and Contents

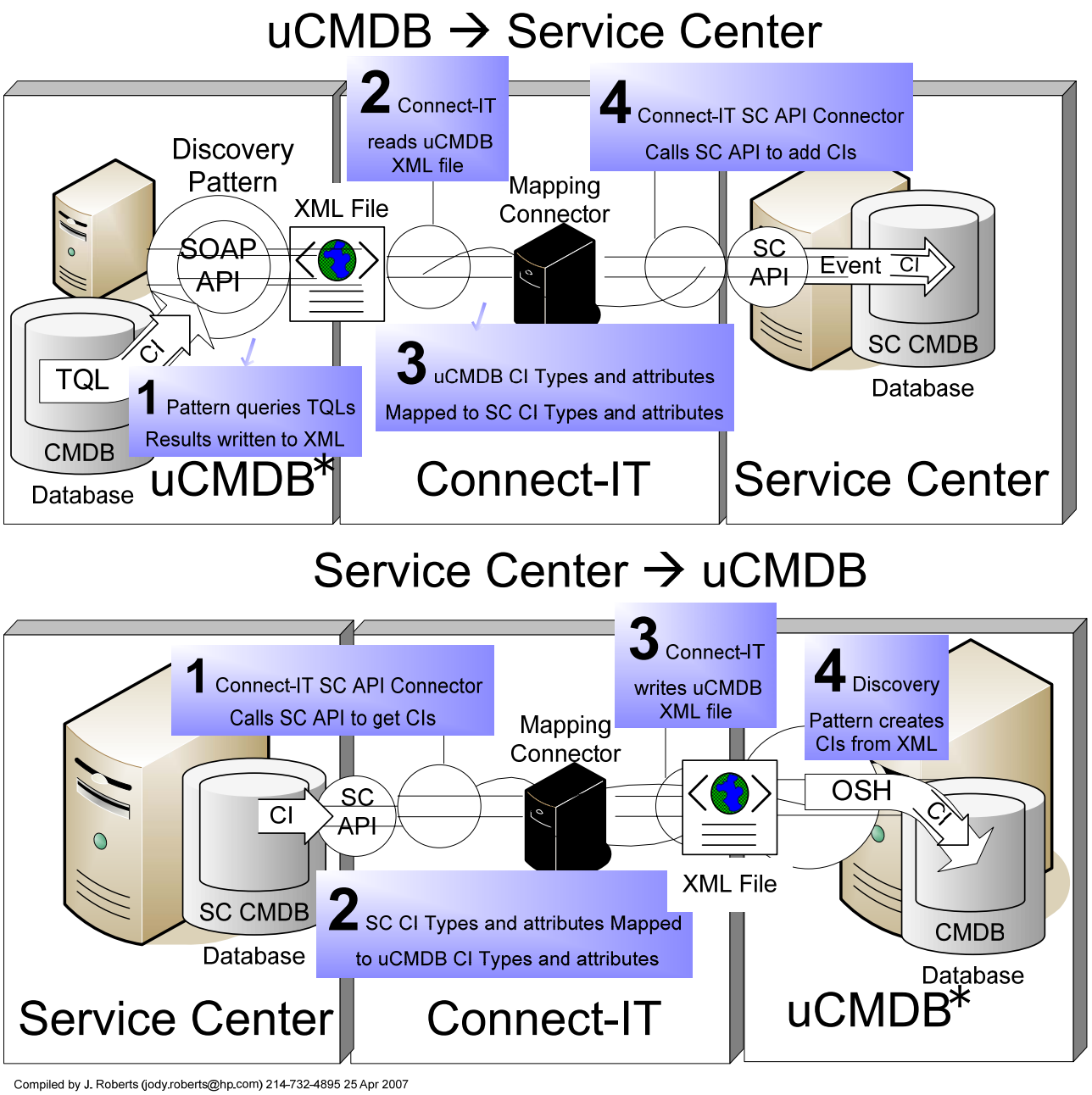
The integration is packaged as a zip file, whose structure and contents are described in the Integration Overview (provided separately with this document). There are three folders: Connect-IT, uCMDB, and Service Center. Each folder contains the files required to deploy the integration for that component. Caution: the uCMDB folder contains a file named "SC6-UCMDB6-Integration.zip".

Caution! Do not confuse this file with the file name of the entire integration package, which may be similarly named (it was originally named something like "uCMDB6-SC6-Integration-v2-050107.zip").

The integration deployment process is organized into three general sections:

	uCMDB to SC	**SC to uCMDB**
uCMDB	1. deploy integration package 2. Share&map data directory 3. configure probe 4. configure pattern parameters 5. schedule and run pattern	1. Same as uCMDB to SC, different pattern
SC	1. Import Integration unload file	No configuration required – data is pulled by Connect-IT.
ConnectIT	1. Deploy integration scenario 2. configure uCMDB connector 3. configure SC connector 4. deploy via service console	1. Same as uCMDB to SC, different scenario, same configuration

Integration Architecture



Compiled by J. Roberts (jody.roberts@hp.com) 214-732-4895 25 Apr 2007

The integration provides a bidirectional flow of CIs between uCMDB and SC. Selected CI Types and attributes are reconciled between the two repositories.

The integration consists of a uCMDB discovery package, Connect-IT scenarios, and SC unload files. The uCMDB packages contain discovery patterns and TQLs using XML. The outbound portion of the integration uses the CMDBs SOAP API. The Connect-IT scenarios consist of connectors to map, read, or write data sources.

* the single uCMDB component shown here is expanded in detail in the uCMDB section.

CI Types Transferred between uCMDB to Service Center

These are the uCMDB and SC CI Types which are transferred from the CMDB to the SC at the current stage.

uCMDB CI Class	**uCMDB CI Type**	**SC CI Type (Subtype)**	**SC CI Subtype**
Business Service Management	BusinessService	BizService	n/a
Host	Host	networkcomponents	n/a
Host	Nt (Windows)	computer	Server
Host	Unix	computer	Server
Network Devices	Network	networkcomponents	n/a
Network Devices	Printer	Officeelectronics (Printer)	Printer
Network Devices	Switch	networkcomponents (Switch)	Switch
Network Devices	Router	networkcomponents (Router)	Router
Network Devices	Concentrator	networkcomponents (Hub)	Hub
Host Resources	Memory	Computer components	Memory
Host Resources	Disk	Computer components	Disk
Host Resources	CPU	Computer components	CPU
Network Resource	IP	Networkcomponents	IPaddress
Network Resource	Interface	Networkcomponents	Interface
Relationships	Container	Logical	n/a
Relationships	Parent	Logical	n/a
Relationships	Dependency	Logical	n/a
Relationships	Application	Logical	n/a

Deployment Planning Checklist

Operational Environment

- ☐ HP MAM or uCMDB 6.5 or 6.6 is installed and operational, and permission to deploy the package
- ☐ At least one MAM or uCMDB Discovery probe installed.
- ☐ Connect-IT 3.7.1 is installed and operational, and permission to configure and deploy the scenarios
- ☐ HP Service Center 6.1 or 6.2 is installed and operational, and permission to import the unload files (enables the uCMDB-to-SC part of the integration).
- ☒ BP4SC Incompatibility: The current version of this integration is incompatible with the current version of BP4SC (Best Practices for Service Center), a package of modifications to Service Center. An update is being developed and will be released when available that provides compability for BP4SC.

Integration files

- ☐ Documentation
- ☐ Connect-IT scenario ZIP files
- ☐ Service Center unload files
- ☐ uCMDB package ZIP file

Deployment Task Planning

- ☐ Initiate processes share a directory on the uCMDB server*
- ☐ Allow the SC server to map to the directory share on the uCMDB server.
- ☐ Ensure CIs and relationships are present in the source CMDB of the selected types.

* This directory is created during the deployment. Refer to the following sections for the exact directory that should be mapped and shared.

uCMDB Configuration

MAM/uCMDB Basics

If you are unfamiliar with MAM or the uCMDB, then in order to deploy and use the integration, you will need to know at a minimum the following:

uCMDB, UCMDB, Universal CMDB, MAM, or MAM CMDB: All names for the same thing for the purposes of this integration document. All these terms can be used interchangeably.

<uCMDB Home> or <MAM Home>: - Typically, *d*:\Mercury or *d*:\hp\uCMDB\ depending on the exact version of MAM or uCMDB installed.

Package: a special meaning in UCMDB for a ZIP file with a specific structure that is used to package, store and transport uCMDB functionality. This integration is partially composed of one of these packages.

Discovery Pattern: a MAM component that performs integration or discovery. A discovery pattern is *activated* and then a *discovery task* is *dispatched* to and *runs* on a discovery probe. The italicized terms are also referred to in this specific context later in this section.

Probe or Discovery Probe: A software component of uCMDB that performs discovery and integration tasks. This integration is partially composed of a discovery pattern that runs on a probe.

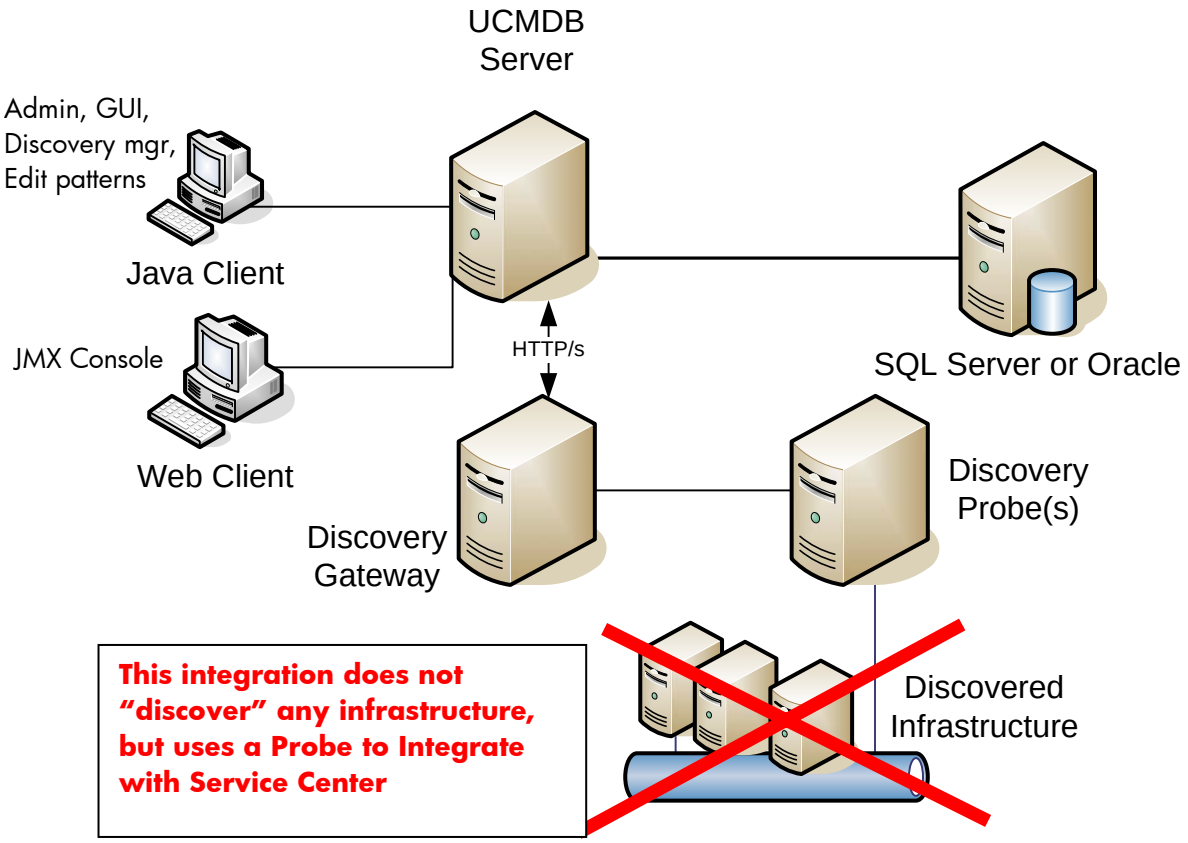
You will need to know where the probes are and be able to get to them. In small deployments and POCs you may see a single probe on the MAM server itself. In large deployments, you will have multiple probes on separate servers.

You need a probe to run the uCMDB Integration patterns. You must modify this probe's CLASPATH (described in section 6) before you can run the integration. If necessary, ask the local MAM administrator what probe they want you to run the integration on. The probe can be any probe as long as it can share a mapped drive to the SC server. Suggest they use their "main" or "default" probe. The probe will still function normally in all other respects.

Gateway or Discovery Gateway: An intermediate component used for communication between the MAM server and the MAM probe. Can be run as a "union" with the discovery probe for simple deployments.

MAM GUI or UI: Required to deploy the integration. Must have MAM userid and password – obtain from MAM administrator. If you don't know who the MAM administrator is, talk to your contact and they can help you find the MAM administrator. The MAM administrator should also be able to tell you where the probes are, provide access to the probes (or take this doc and do the procedures themselves), and provide the location of the MAM server.

uCMDB Basic Architecture:



This diagram is an expansion of the single icon in the "Integration Architecture" section at the beginning of this document.

Deployment Overview

The following steps are required to deploy the uCMDB parts of the integration:

1. Deploy the uCMDB integration file.
2. Open the UI and the Discovery Manager.
3. Configure path and file names in the UCMDBtoSC pattern.

Possible shortcut hint for advanced users:

If, after reading the below, you are confused, disregard and proceed to step 4.

If you have already deployed the UCMDBtoAC pattern for Asset Center-uCMDB Integration on the same uCMDB server, you may not need to deploy this pattern. Out of the box, UCMDBtoSC and UCMDBtoAC produce exactly the same results and put the data in the same file. Obvious exceptions include:

- If you have customized or plan to customize only one of these patterns, or plan to customize each one differently, so that their results differ.

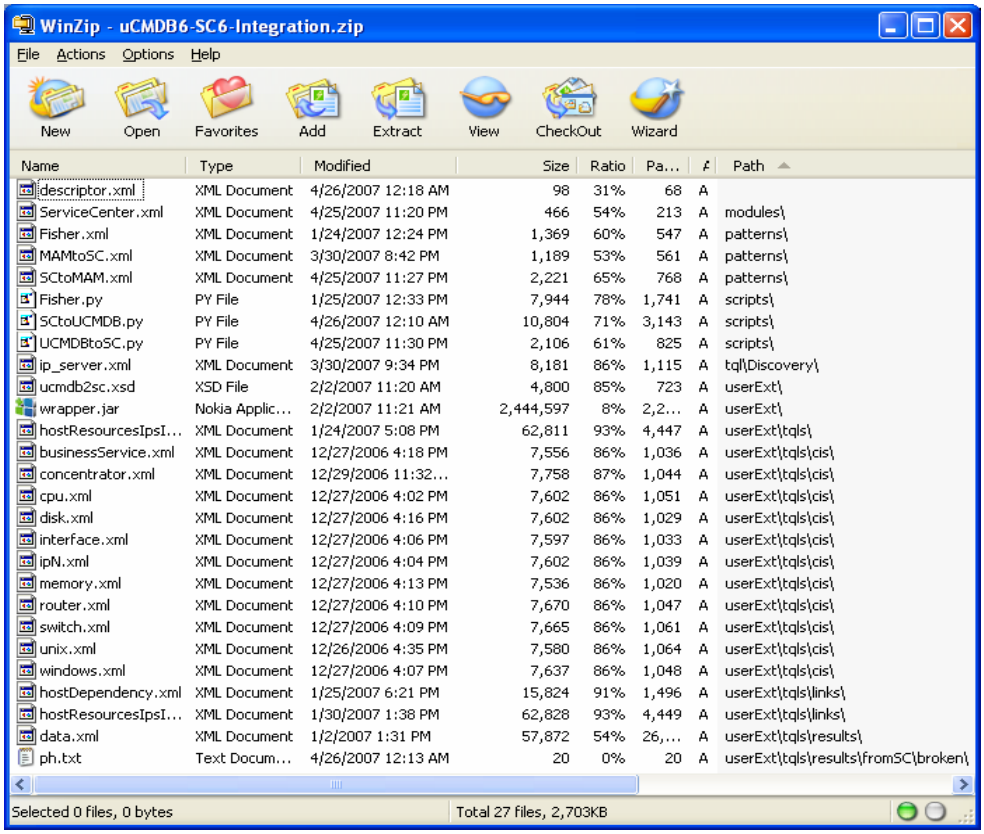
If scheduling requirements differ between SC and AC, you can simply make a copy of the UCMDBtoAC and schedule it appropriately. If you need the SC to UCMDB direction of the integration to work, you will still need to deploy the SC/uCMDB integration package. The advantage to the copy approach is you will also copy your configured pattern parameters, allowing you to skip this step.The advantage to the copy approach is you will also copy your configured pattern parameters, allowing you to skip this step.
Important! Do not do the same thing with the ACtoUCMDB pattern! This and the SCtoUCMDB patterns are **not** the same and produce different results.

4. Configure path name in the SCtoUCMDB pattern.
5. Choose a discovery probe to run the integration.
6. Add the package-supplied wrapper.jar to the probe's class path.
7. Schedule the patterns to run at an appropriate interval.
8. Edit the UCMDBtoSC pattern's TQL trigger to point to the CMDB server.
9. share the uCMDB results directory
10. Activate the integration patterns.

1. Deploy the SC Integration package

1.1 Locate (and extract if necessary) the MAM Integration package file UCMDB6-SC6-Integration.zip, located in the uCMDB folder of the integration package zip file. Place this file in a location where it can stay (not temporary).

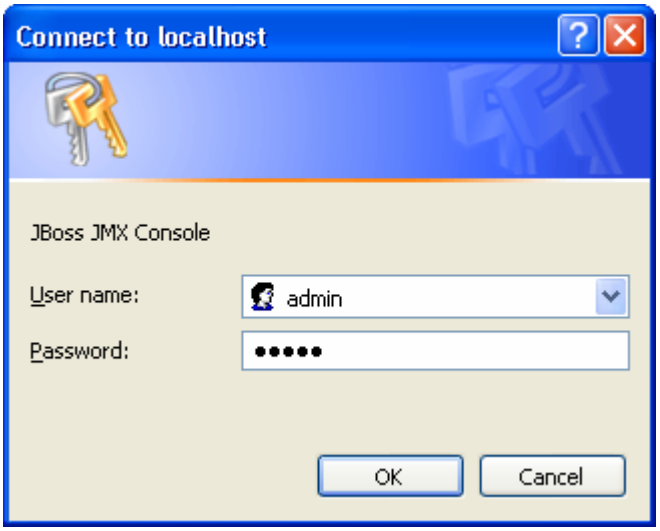
The package's contents should look similar to the following:



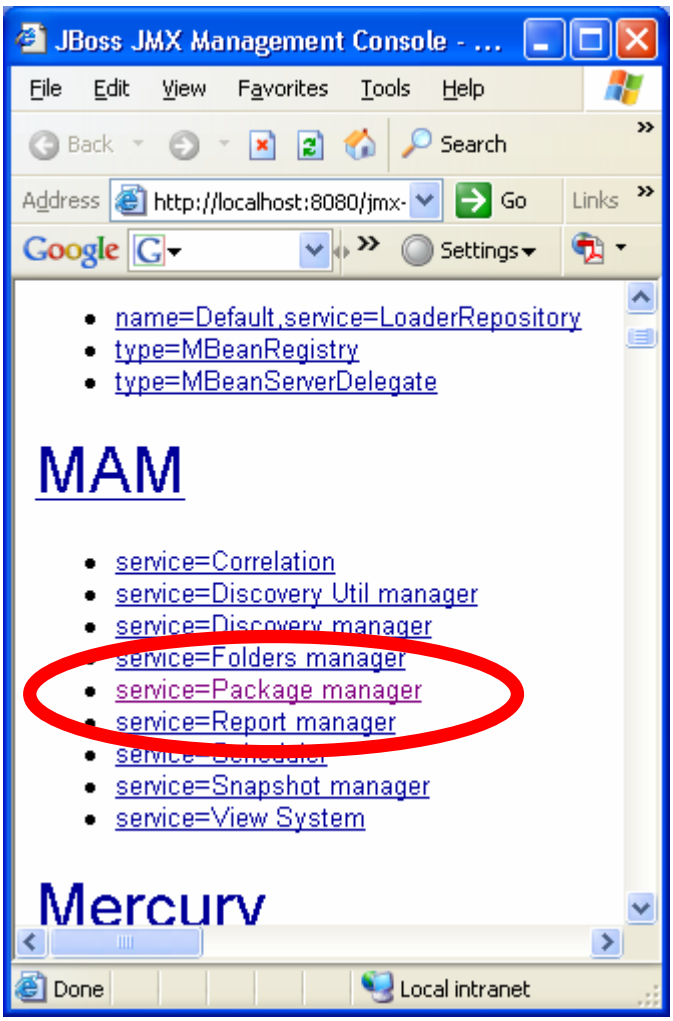
SC6-UCMDB6-Integration.zip package contents

Best practice: copy the file directly to ***<uCMDB Home>*\MAM-V6.*x*\MAMServer\root\lib\packages** along with all the other packages. This is the default value, so if you do this, you can leave the directory field blank when you deploy the package in the JMX Console.

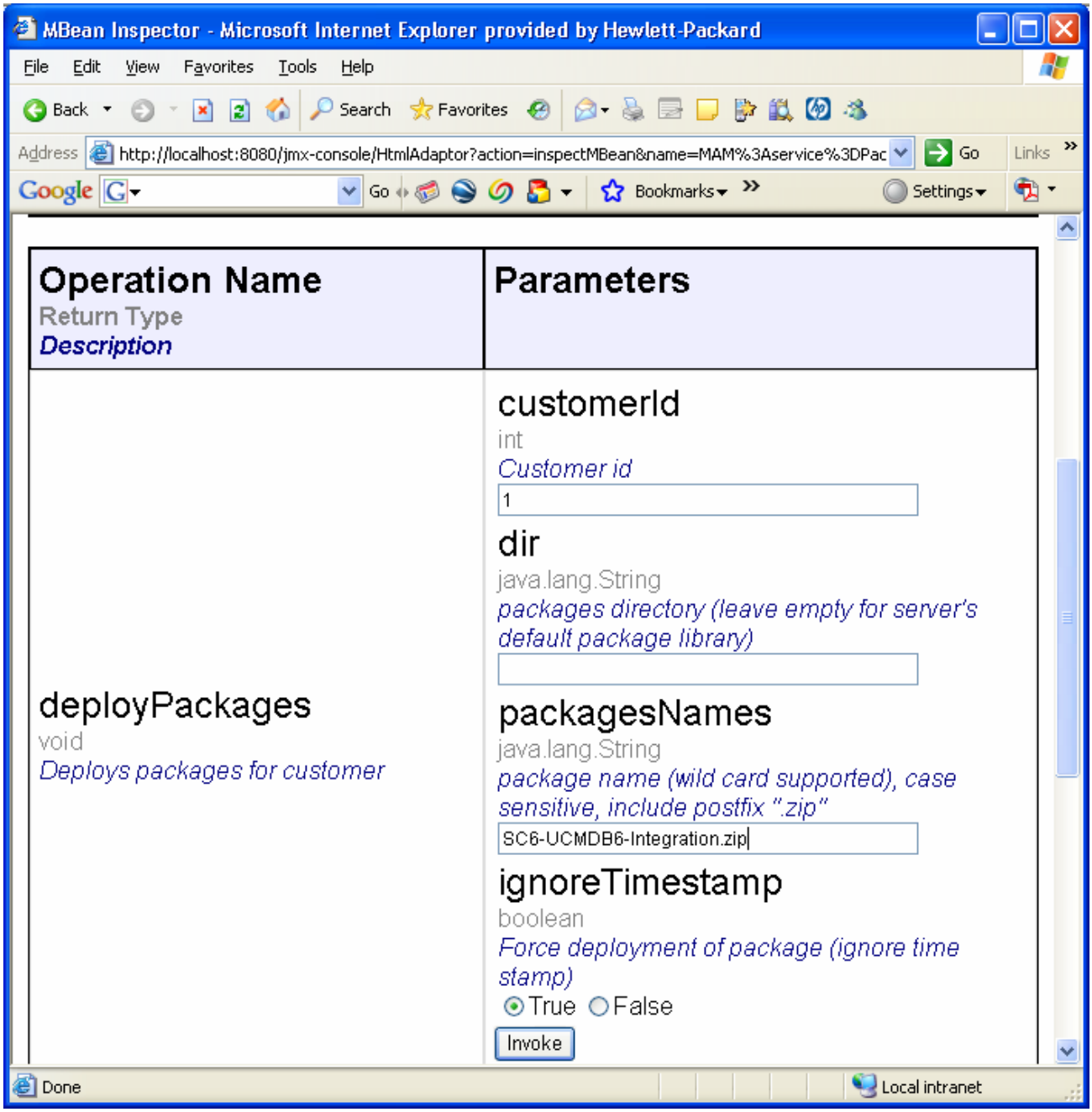
1.2 Start the MAM JMX Console at http://mamserver:8080/jmx-console.



1.3 Login first. Get the id and password from the MAM administrator.



1.4 Select Service=Package Manager



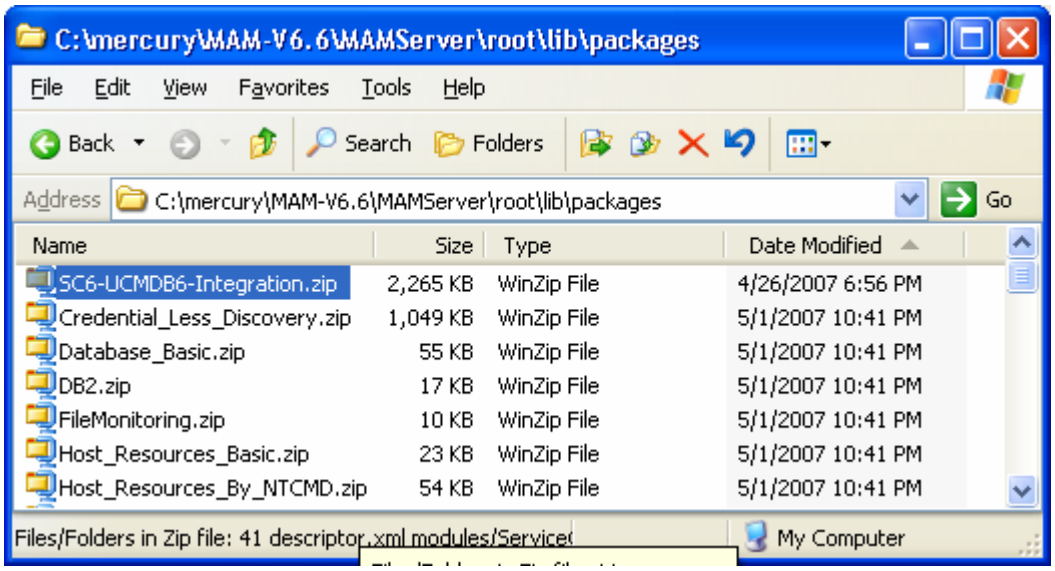
1.5 Complete the "deployPackages" section as shown, then click "Invoke". If you followed the best practice in step 1.1, you may leave the dir field blank. Otherwise, supply the directory where the uCMDB package is located. Again do not confuse the entire integration package's zip file with the uCMDB package zip file, which is what is being deployed here.

The package will deploy.

1.6. To verify the package was deployed correctly, you can check **<*uCMDB Home*>\MAM-V6.*x*\MAMServer\j2f\log\packageVerify.log**. Look for messages similar to the following:

Package [uCMDB6-SC6-Integration.zip] was installed successfully

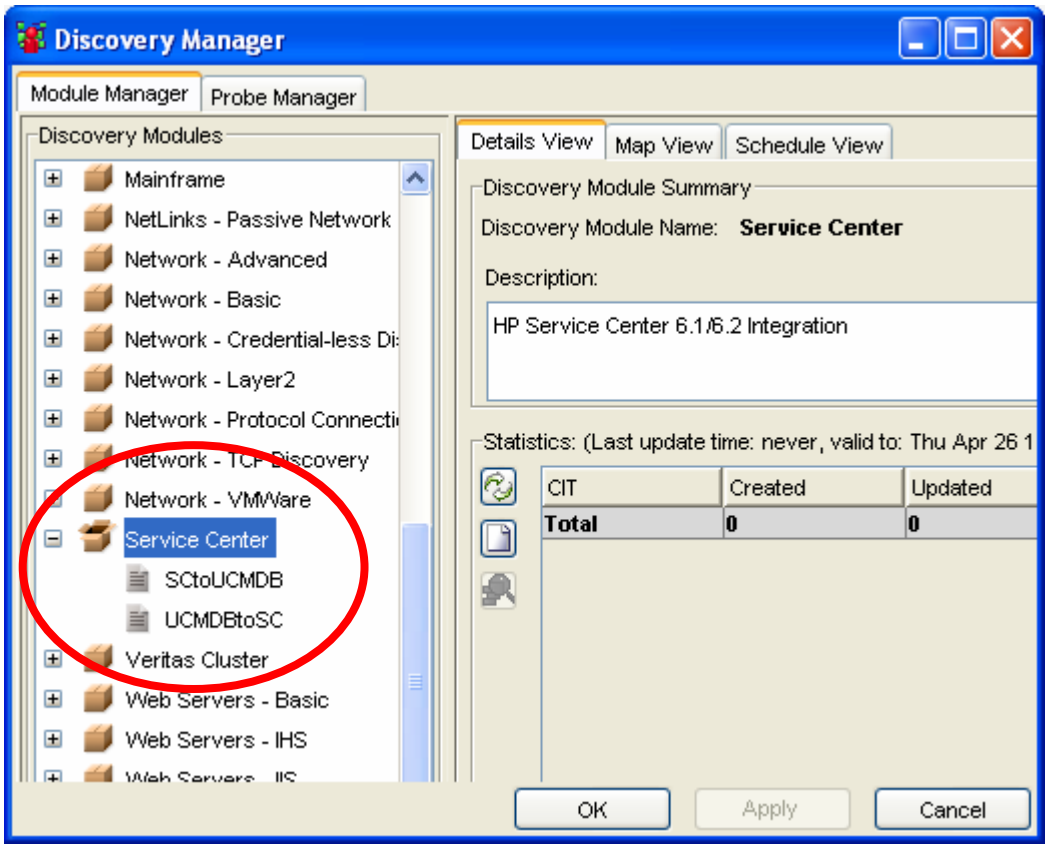
The package contents will look similar to the following:



SC6-UCMDB6-Integration.zip package copied to the uCMDB Server's packages directory.

2. Start the MAM/UCMDB Discovery Manager.

2.1 In the MAM GUI, Select the "Administration" menu item, then "Discovery Manager".
In the 'Discovery Manager' in MAM you should see a new module called Service Center. This module contains two discovery patterns: **SCtoUCMDB** and **UCMDBtoSC**.

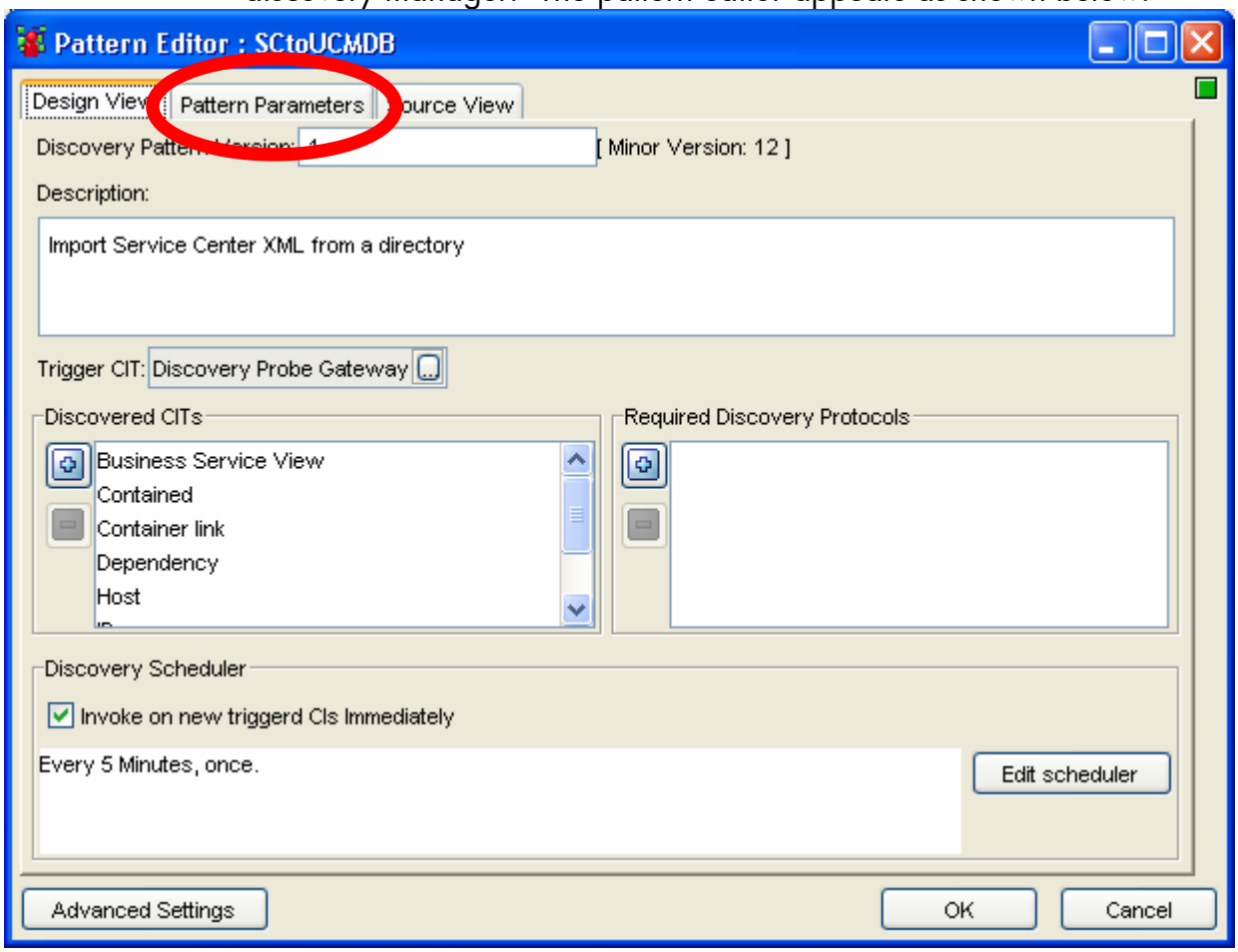


Successful Package deployment. Discovery Manager now shows a new module two two discovery patterns.

3. Edit the 'SCtoUCMDB' pattern and edit the parameters.

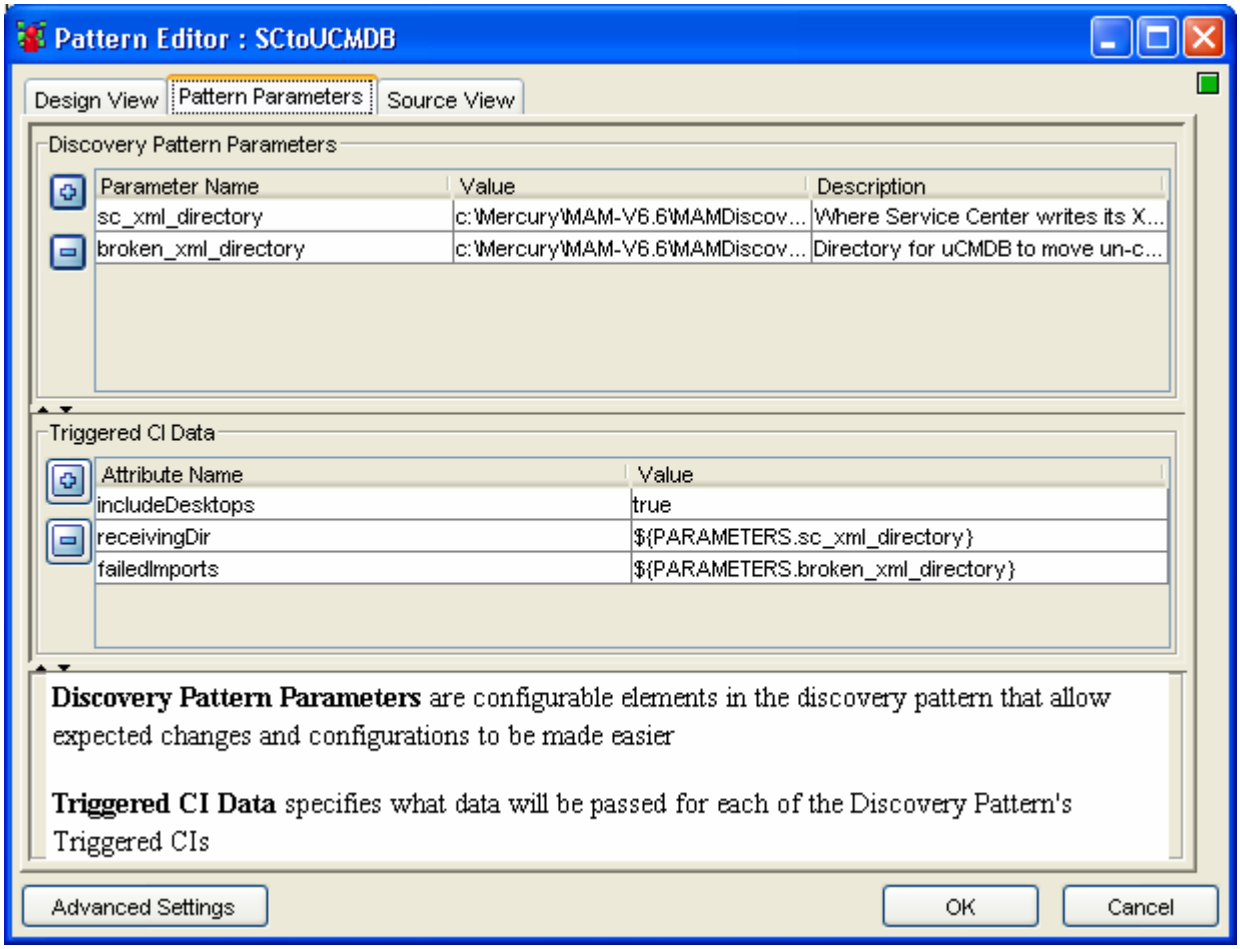
Caution!: This section speaks of editing patterns. Do not edit the XML of the patterns directly unless you know what you're doing! Use the MAM GUI to edit the patterns and their parameters. Do not be confused by this.

3.1 To edit the pattern, double-click on the SCtoUCMDB pattern in the discovery manager. The pattern editor appears as shown below:



Pattern editor

3.2 Click on the "Pattern Parameters" tab. You will see the pattern parameters tab appear, as shown below:



Pattern Parameters tab

3.3 Change the "sc_xml_directory" and "broken_xml_directory" file paths to point to the following directory structure created by the package, respectively. The default path for sc_xml_directory is

C:\Mercury\MAM-V6.6\MAMDiscoveryProbe\root\lib\collectors\probeManager\userExt\tqls\results\fromSC

The default path for the Broken_xml_directory is

C:\Mercury\MAM-V6.6\MAMDiscoveryProbe\root\lib\collectors\probeManager\userExt\tqls\results\fromSC\broken

Important! Verify the drive letter, uCMDB or MAM, and the version are correct.

Important! Ensure there is a slash on the end of the file path.

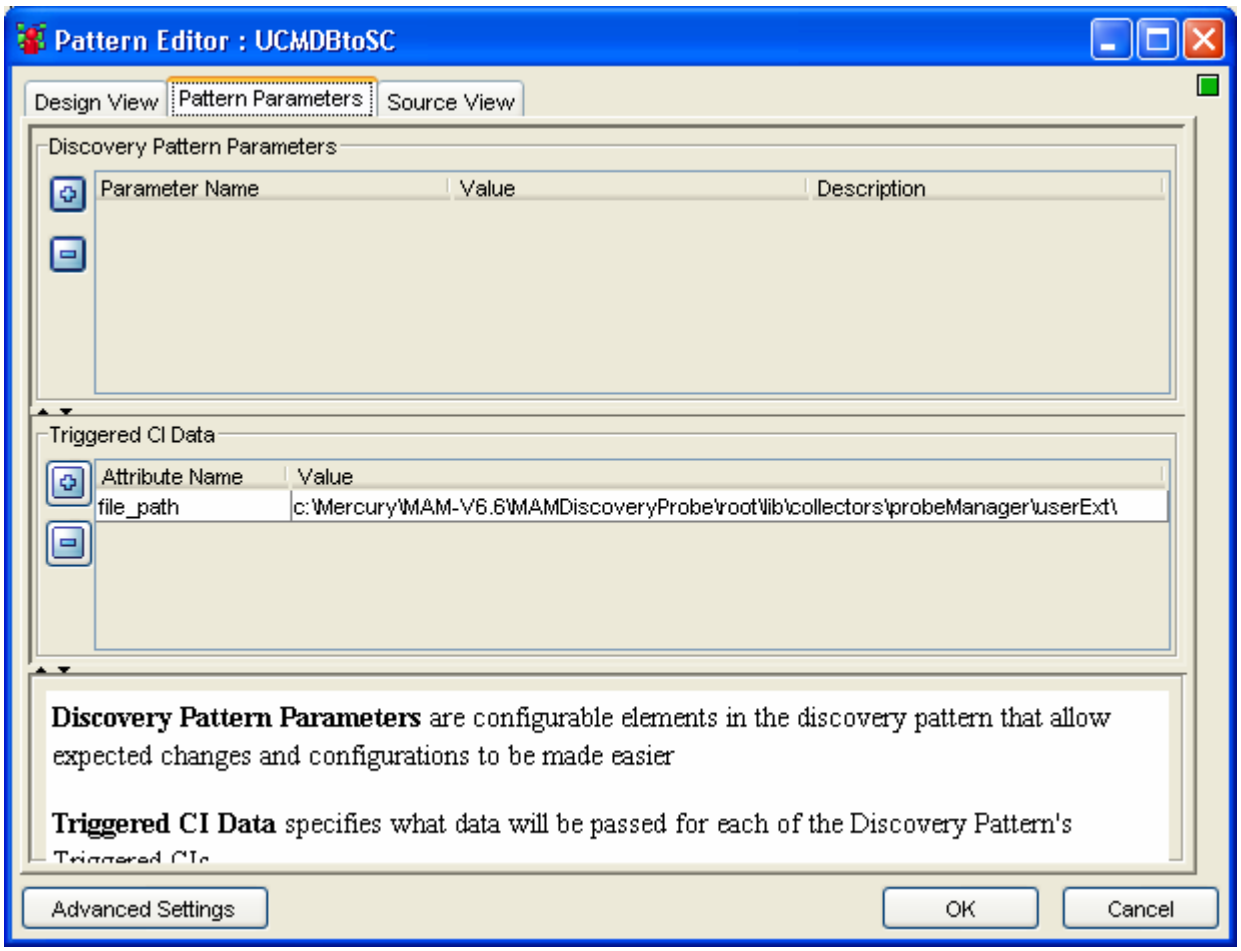
For example, if your environment is uCMDB 6 6 and you installed on the E drive, you would change the path of the sc_xml_directory to the following (changes bolded):
E:\Mercury**UCMDB**-V6.**6**\MAMDiscoveryProbe\root\lib...

4. Repeat Step 2 for the 'UCMDBtoSC' pattern

4.1 Edit the uCMDBtoSC pattern as described in Step 2 above, except this time double-click on the uCMDBtoSC pattern rather than the SCtoUCMDB pattern. Go to the pattern

parameters tab. For UCMDBtoSC, edit the 'file_path' value to match your settings. The default is:

C:\Mercury\MAM-V6.6\MAMDiscoveryProbe\root\lib\collectors\probeManager\userExt\



Editing pattern parameters to specify environment-specific directories.

5. Choose a discovery probe that will run the integration pattern. If you don't know what a probe is, refer to the "primer" section at the beginning of the uCMDB section of this document. If you only have one probe installed, simply use this probe. Do nothing to the probe at this time, simple choose it - you will perform a procedure later on this probe. If you have multiple probes installed, choose one probe and point the integration patterns to run on this probe only.

6. Add wrapper.jar to the probe's classpath.

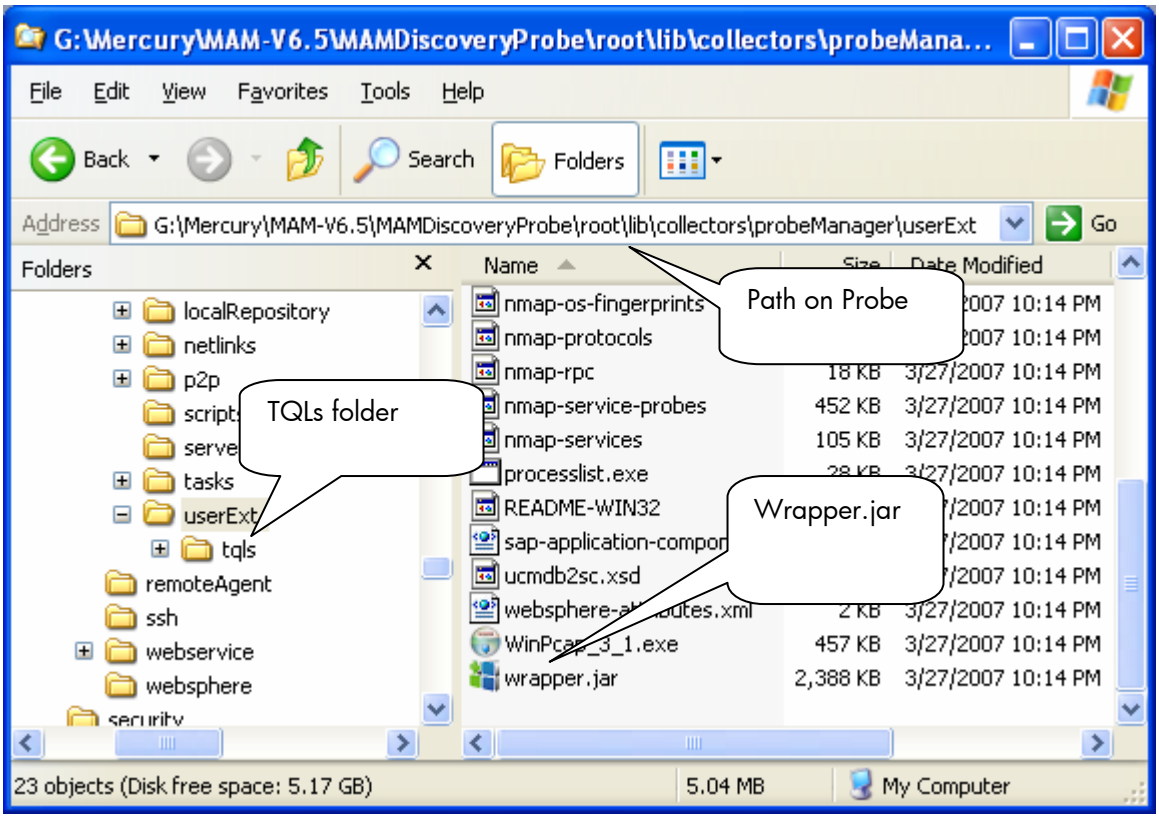
6.1 Log on to the probe, or access the drive where the probe is installed.

On the probe selected in step 5, the package will have deployed a directory structure and files onto the uCMB probe server. Verify this by browsing to

`<MAM-Home>\MAM-V6.x\MAMDiscoveryProbe\root\lib\collectors\probeManager\userExt.`

You should see a tqls/results subdirectory structure under the userext directory, as shown below.

6.2 Locate the `wrapper.jar file` and the `tqls` sub directory inside the `userExt` directory, as shown below:



Folder view on the Discovery probe, the TQLs folder in the userEXT directory has been created.

6.3 On the discovery probe server, edit **`<MAM Home>\MAM-V6.x\MAMDiscoveryProbe\scripts\install\conf\WrapperEnv.conf.`**

6.4 Find the COMMON_CLASSPATH setting.

6.5 Insert the following to the beginning of the probe's COMMON_CLASSPATH entry, as follows:

```
<MAM Home>MAM-
V6.x/MAMDiscoveryProbe/root/lib/collectors/probeManager/userExt/wrapper.jar;
```

Note the forward, rather than backward slashes. Your edit should look similar to the following:

```
set.COMMON_CLASSPATH=G:/Mercury/MAM-
V6.5/MAMDiscoveryProbe/root/lib/collectors/probeManager/userExt/wrapper.jar; %lib%;…rest of original
line follows here
```

6.6 Restart the discovery probe.

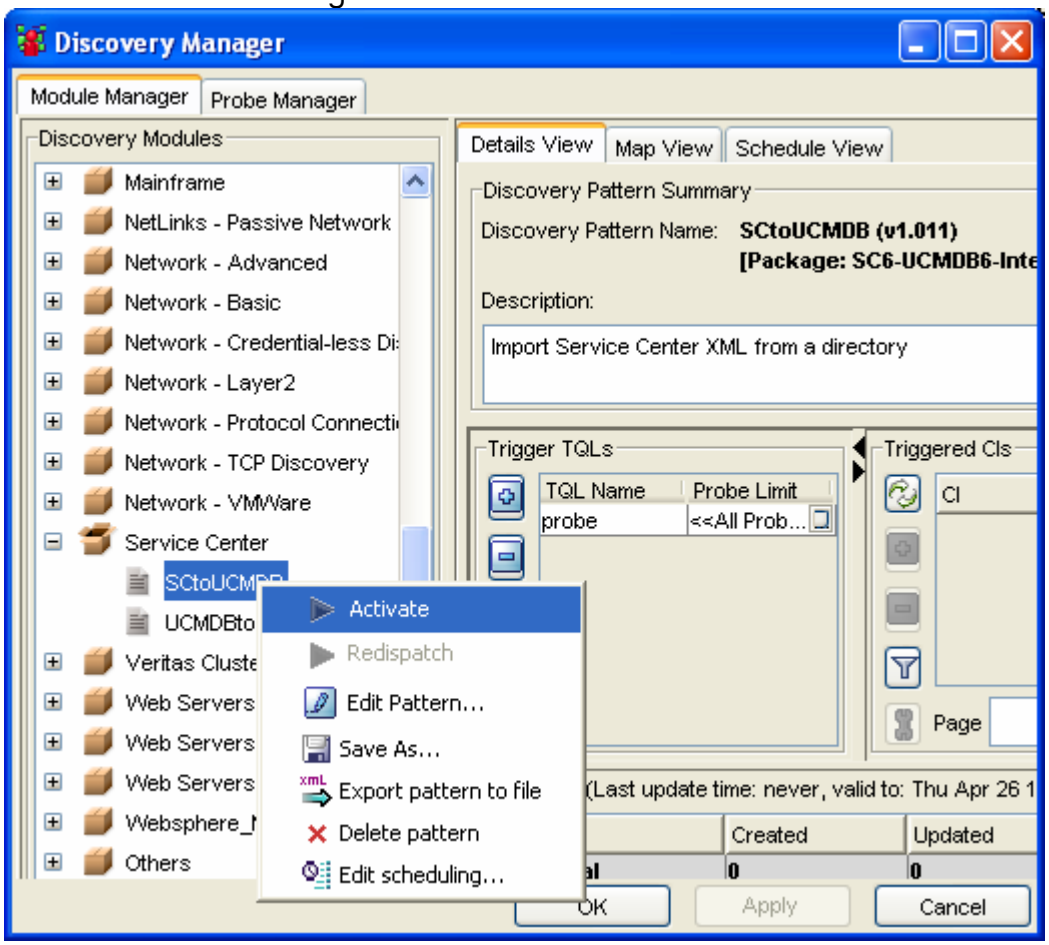
7 Schedule the patterns to run at a suitable interval.

Both patterns include a scheduler that allows it to automatically execute the integrations. As uCMDB is the reference for the CIs, the uCMDBtoSC pattern could be executed every 30 minutes to maintain an updated view of the CIs in ServiceCenter, during the period 08:00 AM to 06:00 PM. In the case a CI has been created in ServiceCenter, SCtoUCMDB will run nightly at 03:00 AM to propagate the information.

These schedule must be updated if they do not match your needs. For instance, if users may work tardily with the applications, you may want to extend the period of execution for the uCMDB to ServiceCenter scenario. If you prefer to only use ServiceCenter to perform modifications on the CIs via the Service Center application, you may consider inverting the schedulers: the ServiceCenter to uCMDB scenario could be run every 30 minutes during work hours, and the uCMDB to ServiceCenter scenario could be run only during the night at 03:00 AM.

To modify a scheduler, please refer to the Connect-It documentation, section "Schedulers and Scheduling".

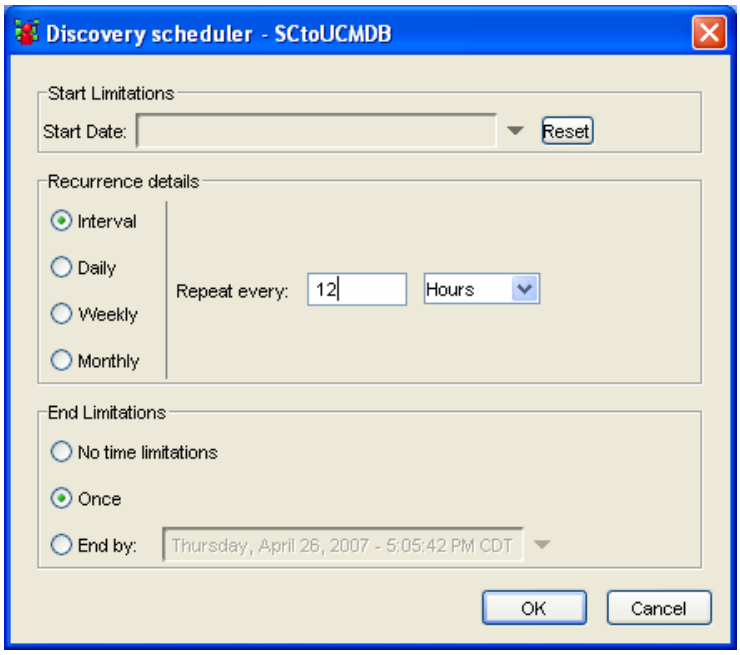
7.3 In the Discovery Manager, right-click each pattern and select "Edit scheduling…" as shown below:



Preparing to edit the patterns' schedules

Best Practice: Choose an interval that provides as much currency as required, but no more than the time it takes to complete a single run. This will vary depending on your CMDB and

SC contents. For example, the pattern requires 1 hour but you only need 12 hour-currency, so select a 12-hour interval, as shown below:



Here, the interval has been set to run every 12 hours.

8. Change the uCMDBtoSC pattern's trigger

You must make a change to the pattern's triggers.
You do not understand what a trigger is to do this part of the integration, but it will help you verify the results and also help you understand what you are doing (which helps in a lot of little ways). You must do this step correctly or the patterns will not run. Therefore, to briefly explain:

1. **Patterns need a trigger:** When a pattern runs based on 1) when you want it to run, e.g., when you schedule it to run or invoke it in the GUI, and 2) when at least one "trigger", or input CI is available.

   During normal discovery, uCMDB patterns operate on lists of input CIs, or "triggers", essentially discovery targets. The pattern executes once for each trigger. For example, if there were 50 IP CIs being input to a host discovery pattern, that pattern would run 50 times and try to discover each of the IPs' hosts.

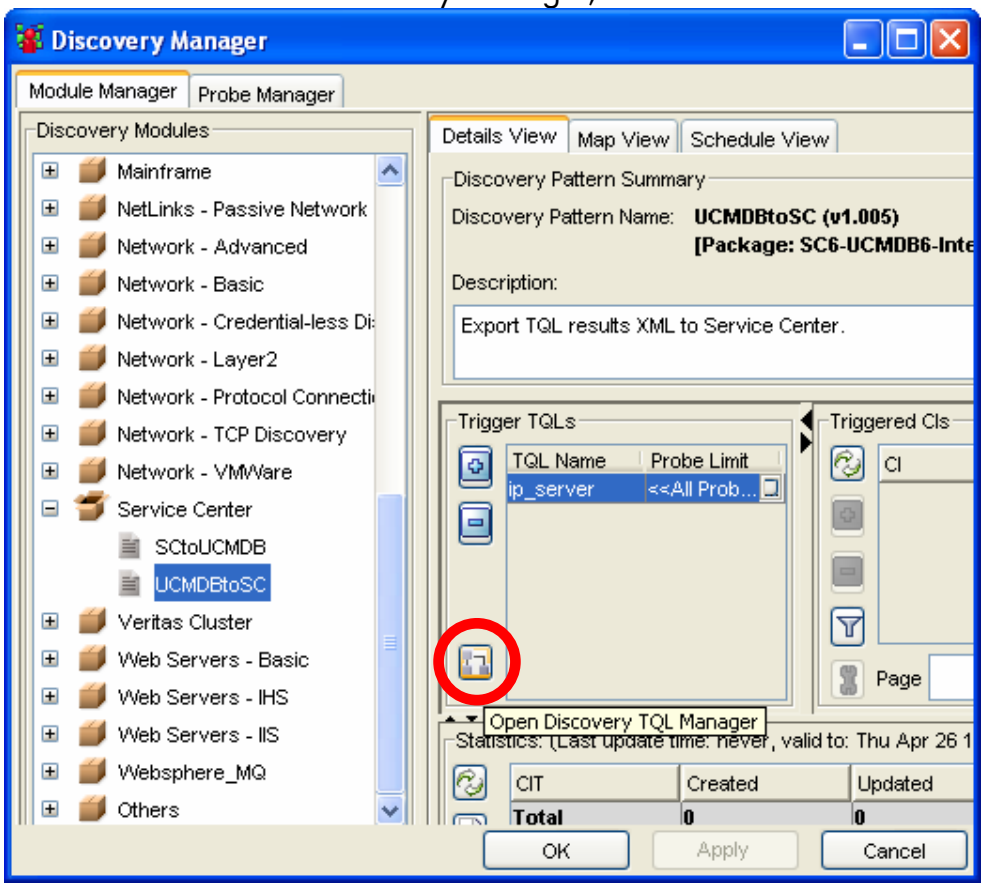
   The patterns create new CIs and link them (usually) to the input CIs. For example, a host would be an input CI for a pattern that discovered disk resources. The host CI would be input to the pattern, and the pattern would output a disk CI and a container link back to it's host CI.

   Triggers are fundamental to the architecture of discovery patterns, meaning, all patterns must have a trigger.

2. **When there is no trigger:** However, with integration patterns, there is no input CI. The pattern reads or writes XML files and does not accept input as a list of targets, rather, a directory path and file specified in it's parameters. This creates the small problem of "then what is the trigger?"
3. **The answer:** The common practice to bypass the trigger requirement is to supply an integration pattern with a single CI, the one representing the IP address of the MAM server. This ensures the pattern will always run exactly one time when invoked. when Because this CI is guaranteed to exist, even if no other discovery has occurred, thus guaranteeing that the pattern will have an input CI and run.

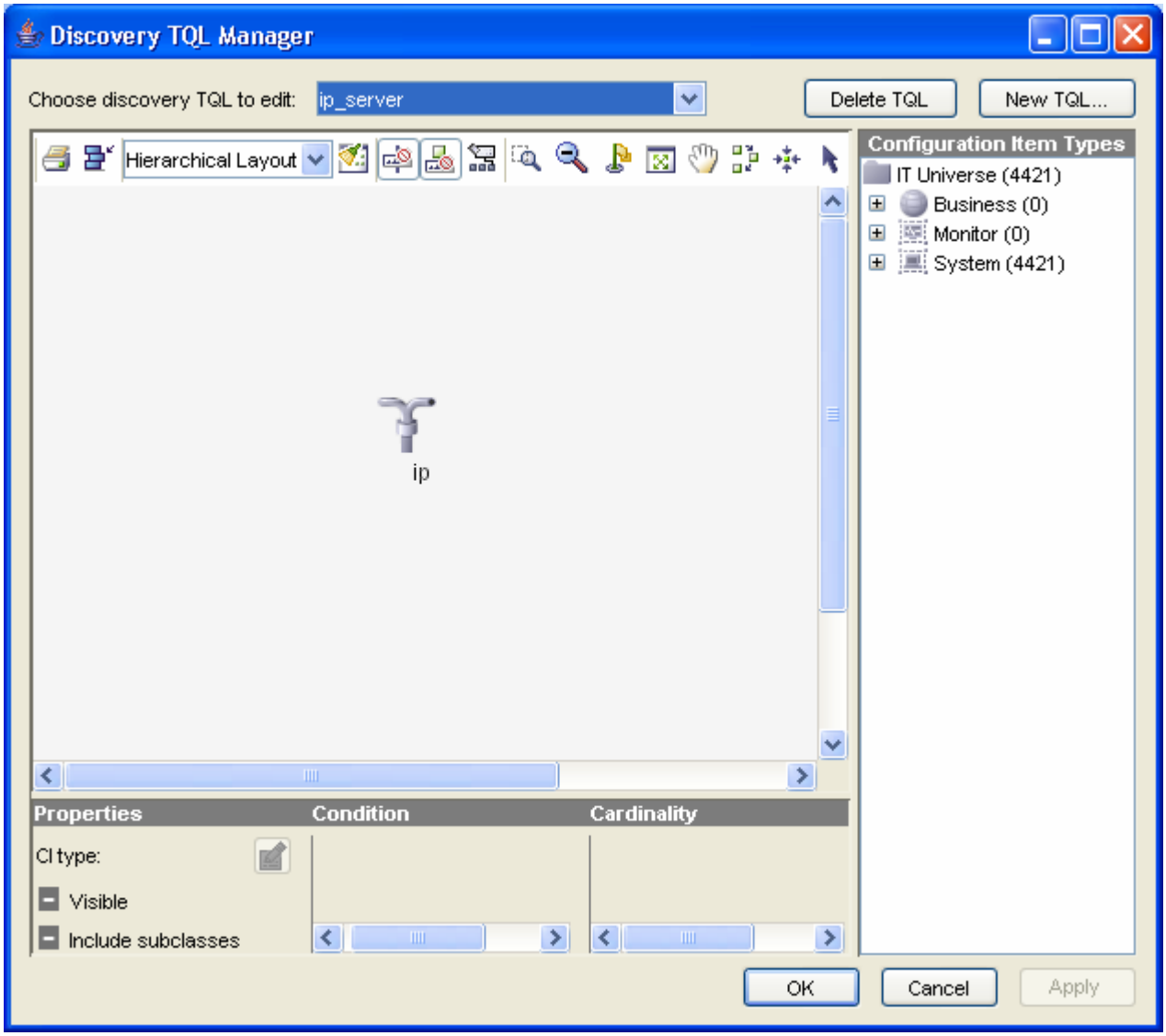
With this explanation, you can now understand what is happening in the next step. You may need this information in order to verify that the steps worked. If any of this is confusing, simply follow the steps and if the patterns run, you did it correctly.

8.1 From discovery manager, click the TQL icon as shown here:



How to edit the UCMDBtoSC pattern's trigger

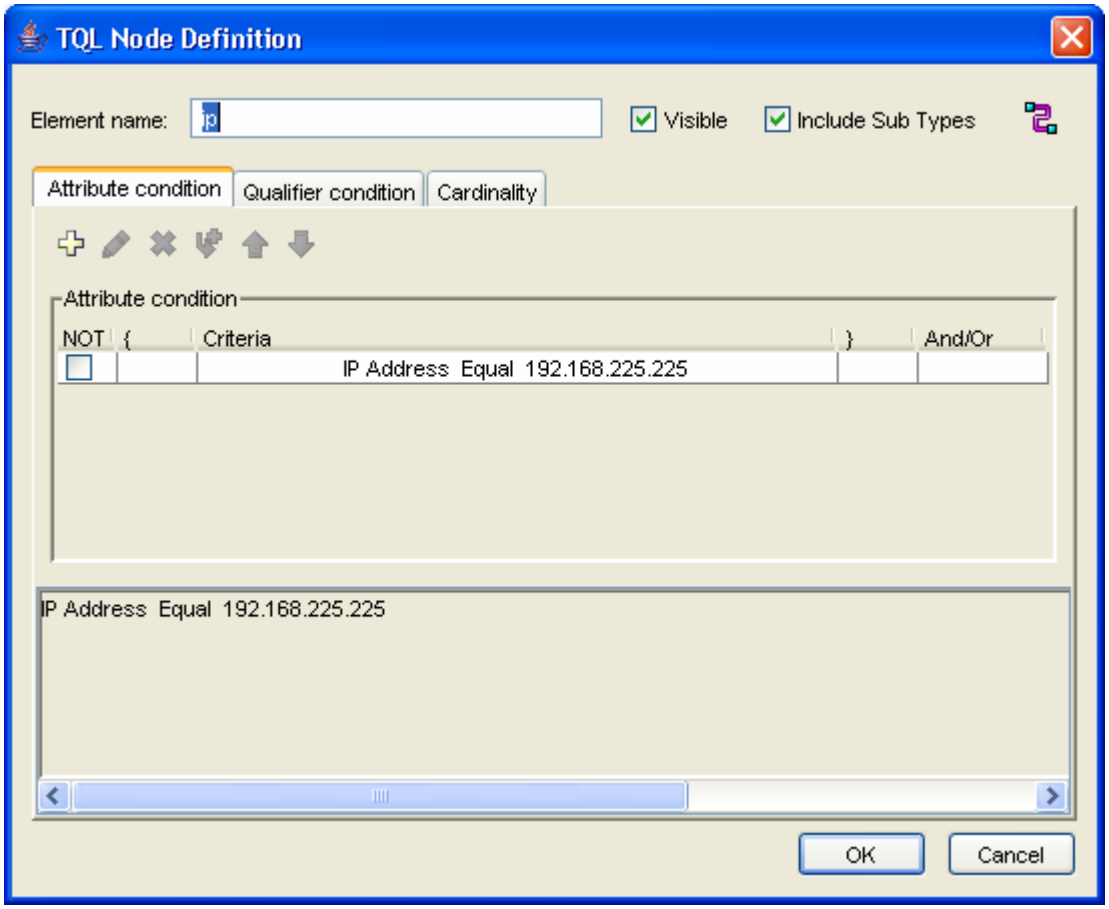
The Discovery TQL Manager will appear:



Discovery TQL Manager

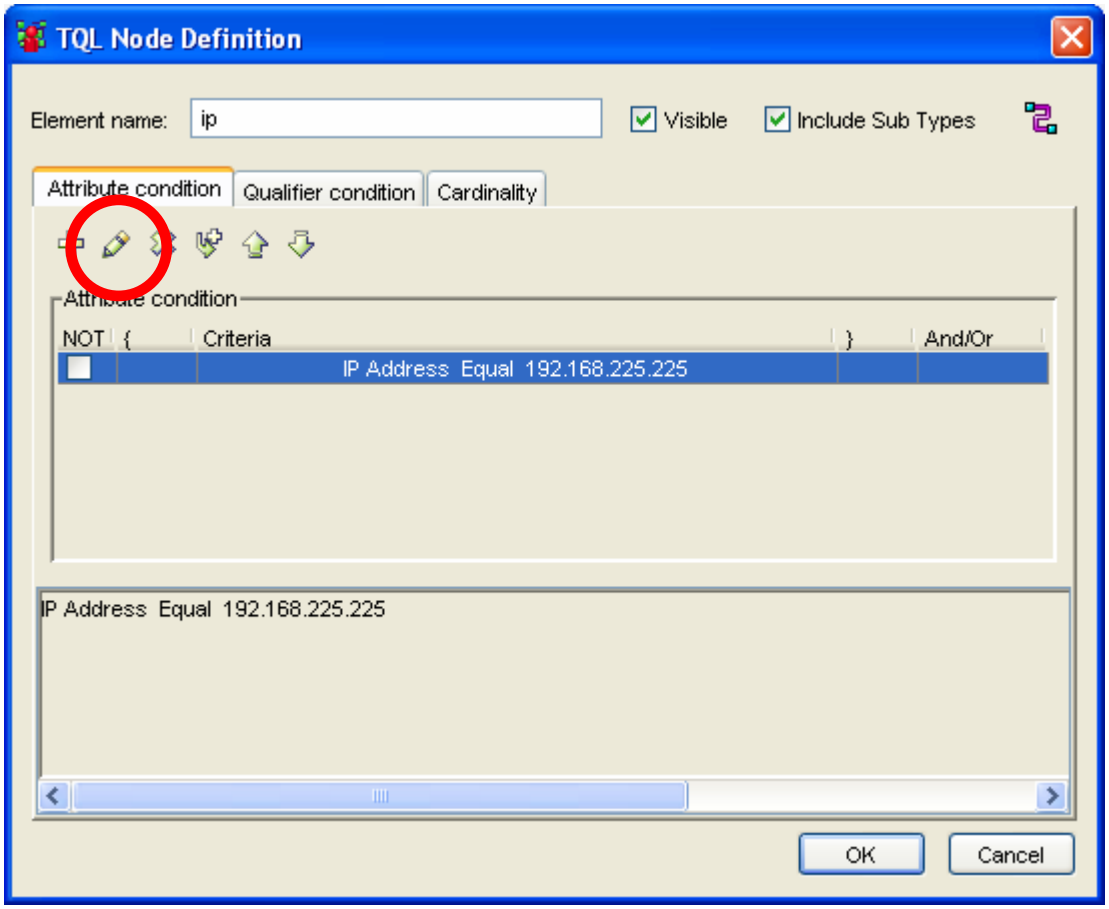
Note the small icon labeled "ip" near the center of the pane.

8.2 Right-click on the IP icon.
8.3 Select "TQL Node Definition". The following window appears:

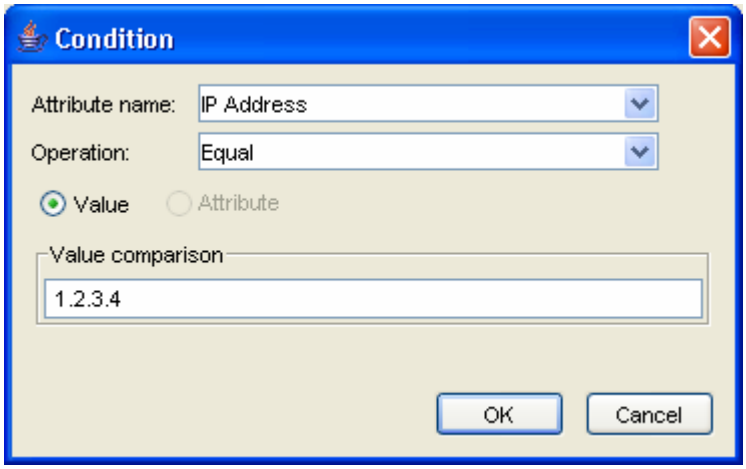


TQL Node Definition window.

8.4 Select the Attribute Condition (the line containing "IP Address Equal…")
8.5 The selection will turn blue:

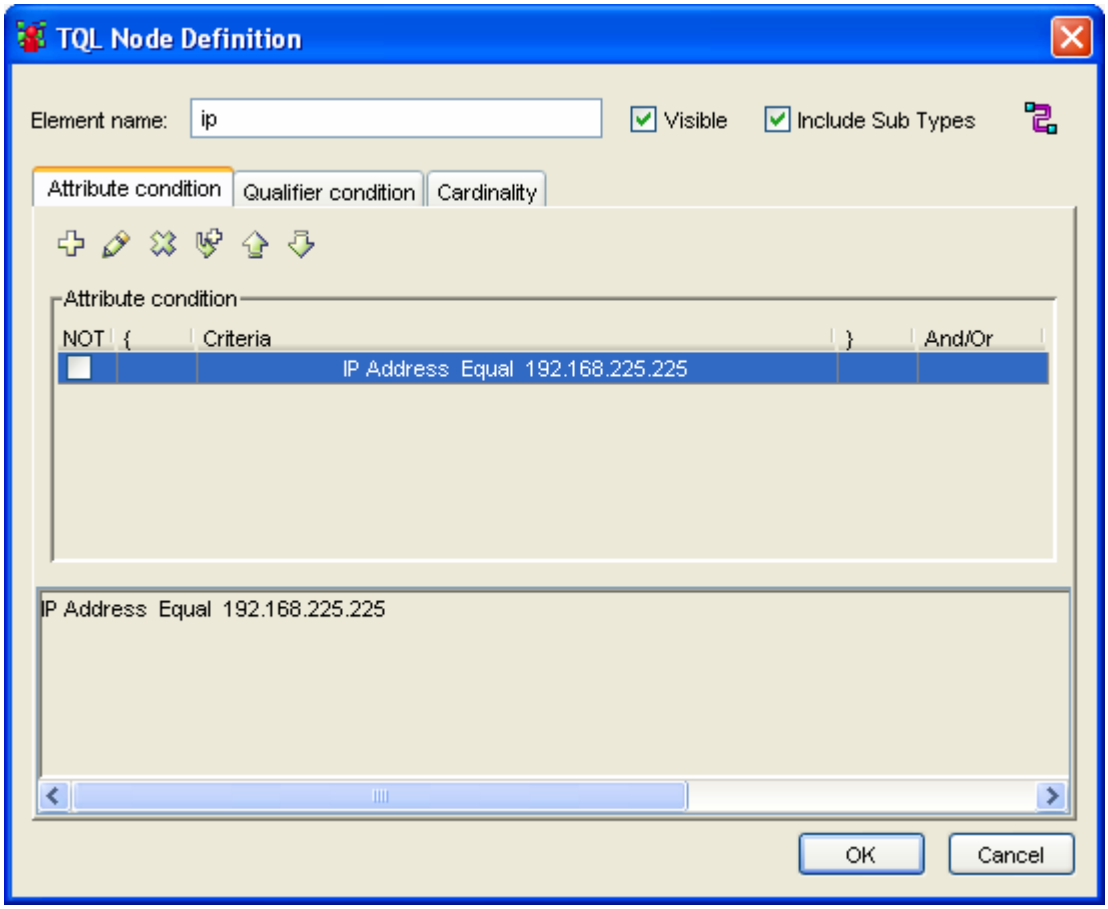


8.6 Click the Pencil Icon to edit the condition. The Condition window appears as shown below:



TQL Attribute Condition window

8.7 Change the IP address to match the IP address of the uCMDB server. This may be obtained from the MAM administrator or by typing "ipconfig" on a command prompt on the MAM server. In this example, the IP address of the MAM server is 1.2.3.4.



The TQL node condition of the discovery TQL "ip_server" has been altered to point to the uCMDB server.

Note the `192.168.225.225` IP address shown should now match the IP address of your CMDB server.

9. Share the results directory.

On the CMDB Server, share the `results` directory if the CMDB server and SC server are separated. Name the share "`SCshare`" or other suitable name. Allow a user on the SC server to read and write to that share.

10 Activate the patterns.

10.1 To Run a Discovery Pattern:

10.1.1 Open the Discovery Manager in the MAM GUI.
10.1.2 Open the Service Center Module.
10.1.3 Right-click on the pattern to activate.
10.1.4. Select "Activate".

10.2 UCMDBtoSC: After the UCMDBtoSC pattern runs:

You should see output in the discovery probe that appears similar to the following. If you do not have direct access to the discovery probe, the same information appears in the log file on the Probe

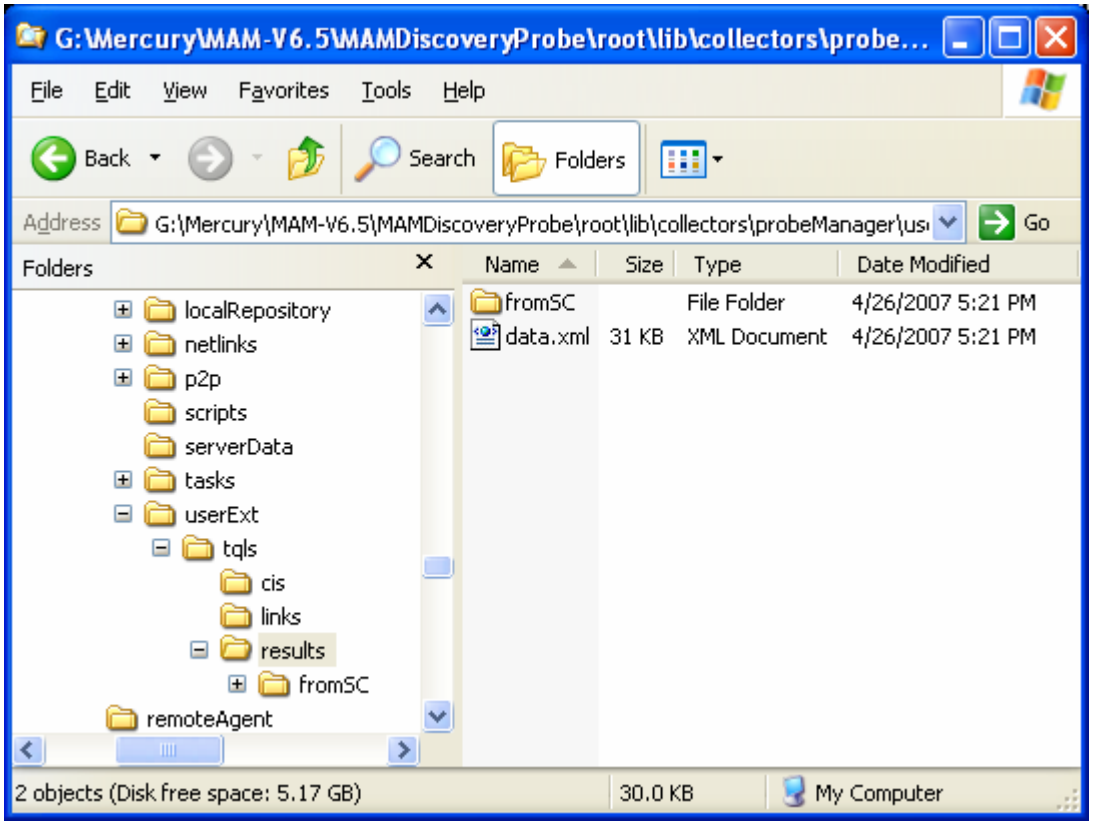
<uCMDB Home>\MAM-V6.6\MAMDiscoveryProbe\root\logs\WrapperProbe.log

```
jvm 1    | <2007-04-26 17:21:44,768> 55112 [INFO ] [Worker: UCMDBtoSC] (BaseServ
```

```
ice.java121) - The task UCMDBtoSC started invocation (on 1 destinations)
jvm 1    | *sys-package-mgr*: processing modified jar, 'G:\Mercury\MAM-V6.5\MAMD
iscoveryProbe\root\lib\collectors\probeManager\userExt\wrapper.jar'
jvm 1    | <2007-04-26 17:21:46,011> 56355 [INFO ] [serviceWorker-29:UCMDBtoSC]
(??) - Start uCMDB to SC integration.py
jvm 1    | working on hostDependency.xml
jvm 1    | working on hostResourcesIpsInterface.xml
jvm 1    | <2007-04-26 17:21:53,174> 63518 [INFO ] [serviceWorker-29:UCMDBtoSC]
(??) - End export
jvm 1    | <2007-04-26 17:21:53,424> 63768 [INFO ] [serviceWorker-29:UCMDBtoSC]
(BaseService.java315) - Result Received -  taskId=UCMDBtoSC, subject=2b8dbeafe74
f4241ca813fe10515aeef, Passed 0 out of 0 objects for 1 destinations (state 0 ->
0)
```

Note the "Passed 0 out of 0 objects". This is normal because UCMDBtoSC is not creating any inbound CIs, only outbound CIs.

10.2.1 You should see a file named `data.xml` that resides in the `tqls\results` directory as shown below:



If you see data.xml get updated here, the pattern ran successfully.

10.3 **SCtoUCMDB:** If After the other parts of the integration are not yet deployed, nothing will happen when you activate the pattern. Once a file appears in the `fromSC` directory uner the results directory described earlier, the pattern should see the file and create CIs based on its contents.

10.3.1 **To verify the pattern ran,** you should see something similar in the discovery probe's output:

```
jvm 1    | added link  SBLAPP1  to  Customer Service
jvm 1    | added link  SBLGW  to  Customer Service_app
```

jvm 1 | added link Email A to Email
jvm 1 | added link Exchange A to Email A
jvm 1 | added link CALDERONE to Manufacturing
jvm 1 | added link s-ldap01.server.com to Online Banking
jvm 1 | added link so-j2ee01.merc-int.com to Web Portal
jvm 1 | <2007-04-26 18:58:19,357> 4038529 [INFO] [serviceWorker-29:SCtoUCMDB] (??) - **SCtoMAM.py : END**
jvm 1 | <2007-04-26 18:58:19,357> 4038529 [INFO] [serviceWorker-29:SCtoUCMDB] (BaseService.java315) - Result Received - taskId=SCtoUCMDB, subject=61e52a1503940b993164138d9d831871, **Passed 700 out of 700 objects for 1 destinations** (state 0 -> 700)
jvm 1 | <2007-04-26 18:58:19,367> 4038539 [INFO] [Worker: SCtoUCMDB] (LocalTaskResultsDistributer.java242) - (ProbeMgr) **Sent a total of grouped 700 update objects for pattern: SCtoUCMDB**

SCtoUCMDB *should* return greater than zero objects because it is creating CMDB CIs from the XML input.

The integration deployment for uCMDB is now complete.

ServiceCenter Configuration

For Service Center to uCMDB, no configuration is necessary. All API calls and mapping are configured from Connect-IT.

For uCMDB to Service Center, the only task is to import an unload file.

1. Deploy the Integration kit's unload file.

Choose the appropriate unload file for your version of ServiceCenter:

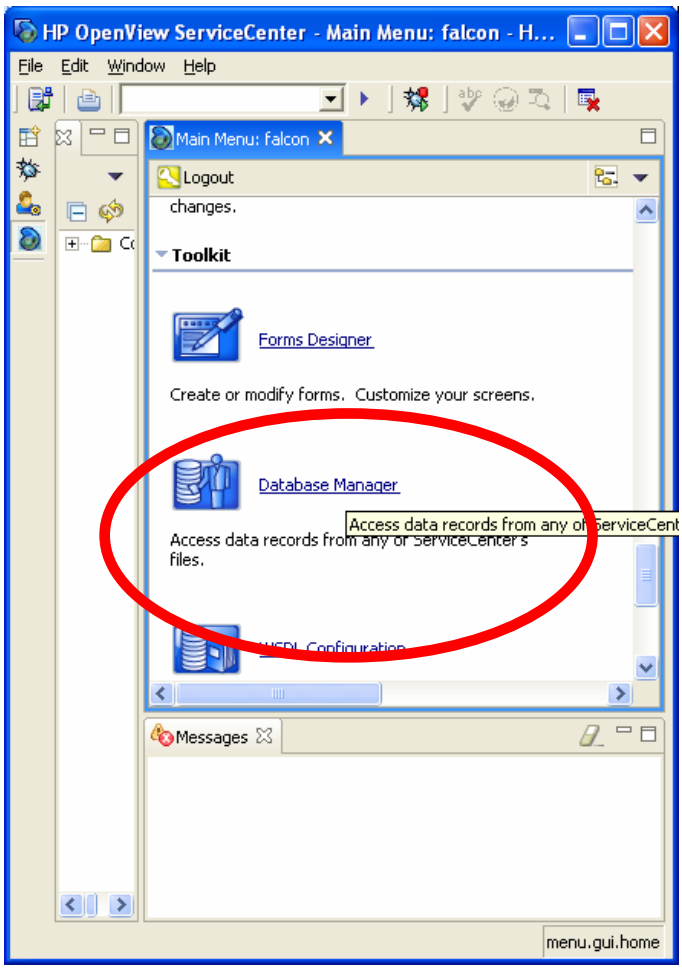
- ServiceCenter 6.2 – ucmdbSC62.unl
- ServiceCenter 6.1 – ucmdbSC61.unl

These files are supplied in the Integration kit accompanying this document.

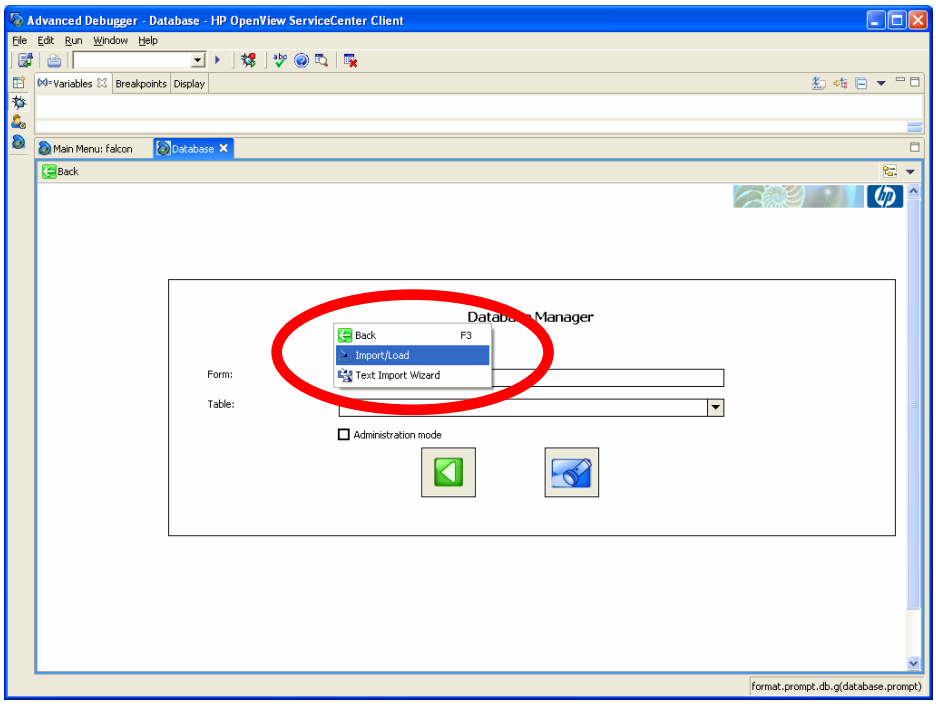
Use the following steps to load one of these unload files:

In the SC Windows Client:

1.1 Click **Toolkit > Database Manager**.



1.2 Right-click anywhere on the form and select **Import/Load**.



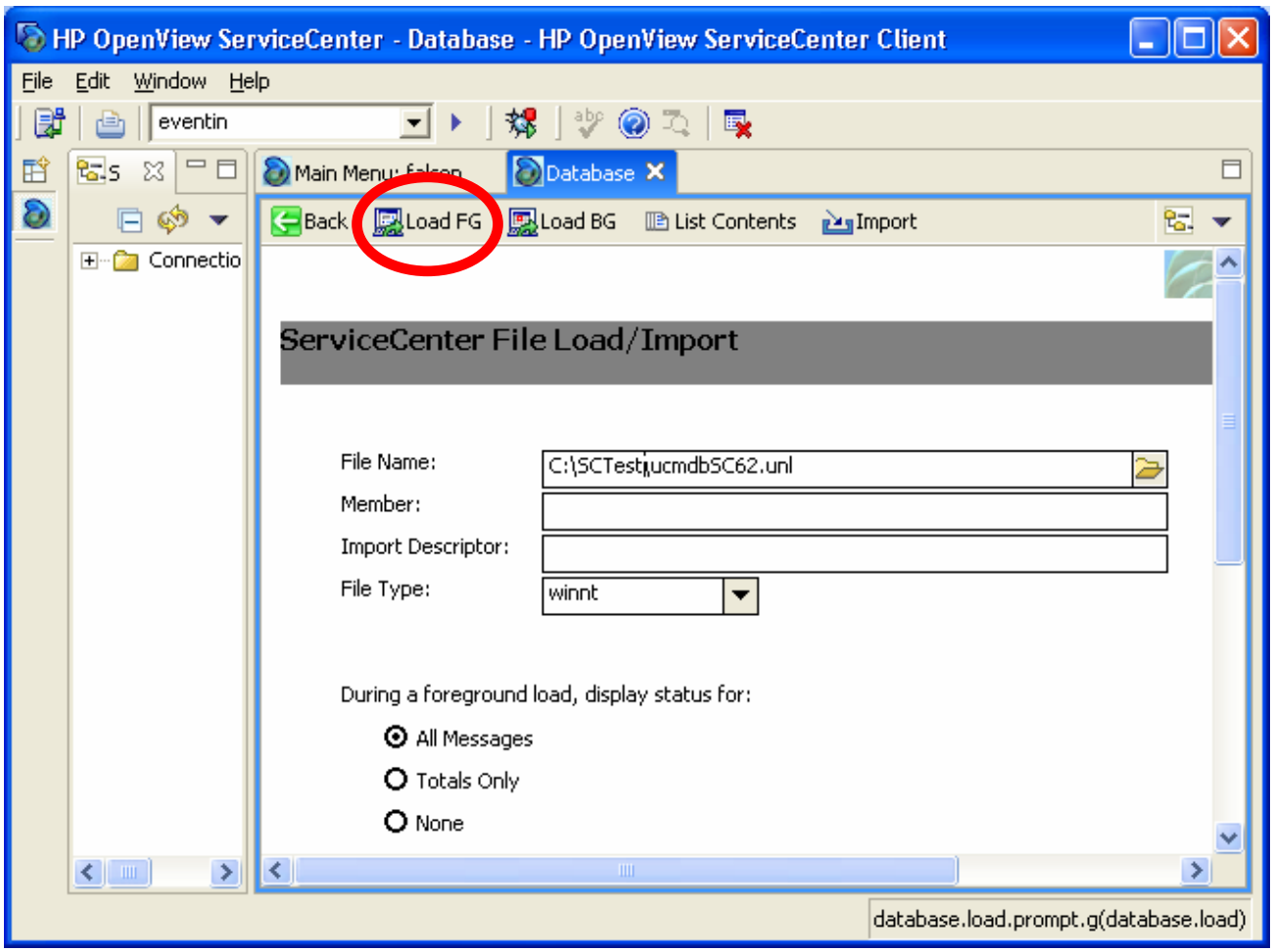
2 Type or select the following information.

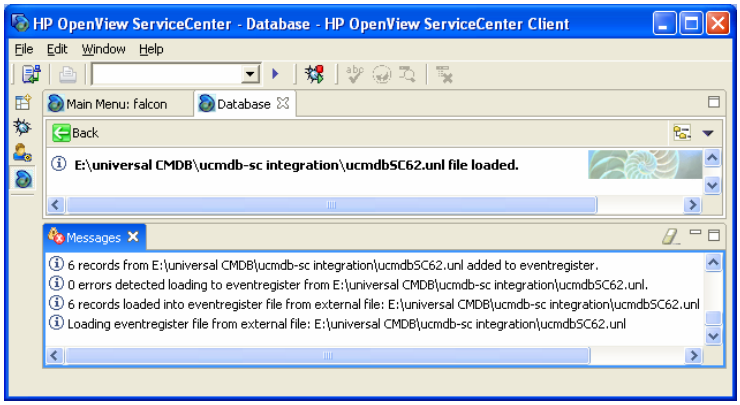
Field	Description
File Name	Type the name and path of the file to load.
Member	Leave blank
Import Descriptor	Type the name of the optional Import Descriptor record used to import records. Since unloads files do not require an Import Descriptor record, leave this field blank.
File Type	Select the source operating system of the unload file.
Messages Option — All Messages	Select this option to see all messages that HP OpenView ServiceCenter generates loading the file.
Messages Option — Totals Only	Select this option to see only the total number of files HP OpenView ServiceCenter loads.
Messages Option — None	Select this option to hide all messages that HP OpenView ServiceCenter generates loading the file.

Best Practice: Show All Messages the first time the integration is operated. Then, change to Totals only for regular operation.

Note: You can view the contents of an unload file before importing it by clicking List Contents.

1.3 Click **Load FG**.





A success message should appear in the main window.

Specific Messages will appear in the Messages window.

The integration deployment for Service Center is now complete.

Connect-IT Configuration

Note: Connect-IT 3.7.1 was used for this integration.

The steps for deployment of both directions of the integration are essentially the same and are logical inverses of each other. The Connect-IT deployment tasks are summarized as follows:

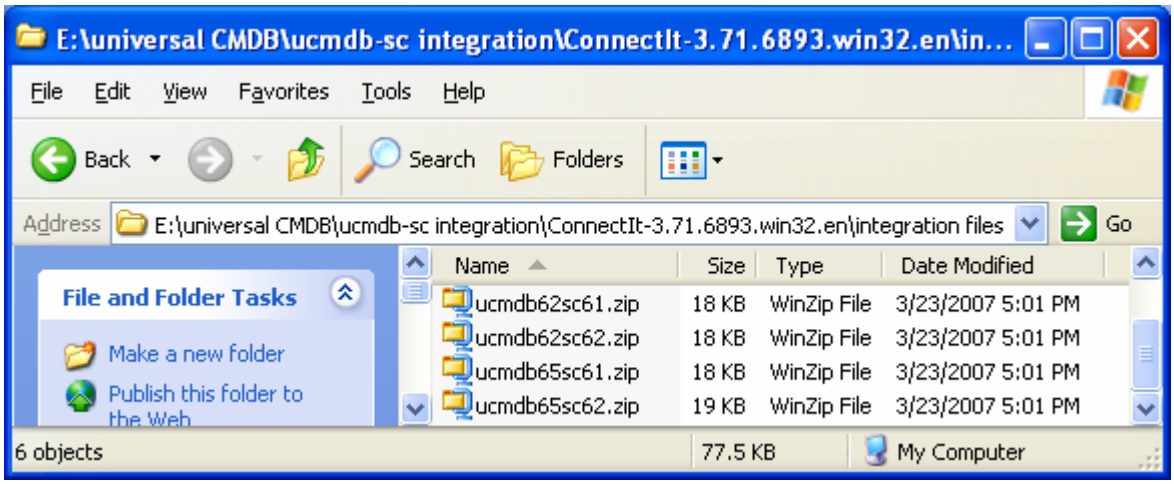
1. Load the scenario into the Connect-IT Scenario Builder.
2. Configure the Service Center Connector, essentially, login information.
3. Configure the MAM/CMDB Connector, essentially, the XML file location.
4. Test the scenarios by opening the connectors and observing the results.
5. Deploy the "live" integration using the Connect-IT service console.

1. Load the scenario into the Connect-IT Scenario Builder.

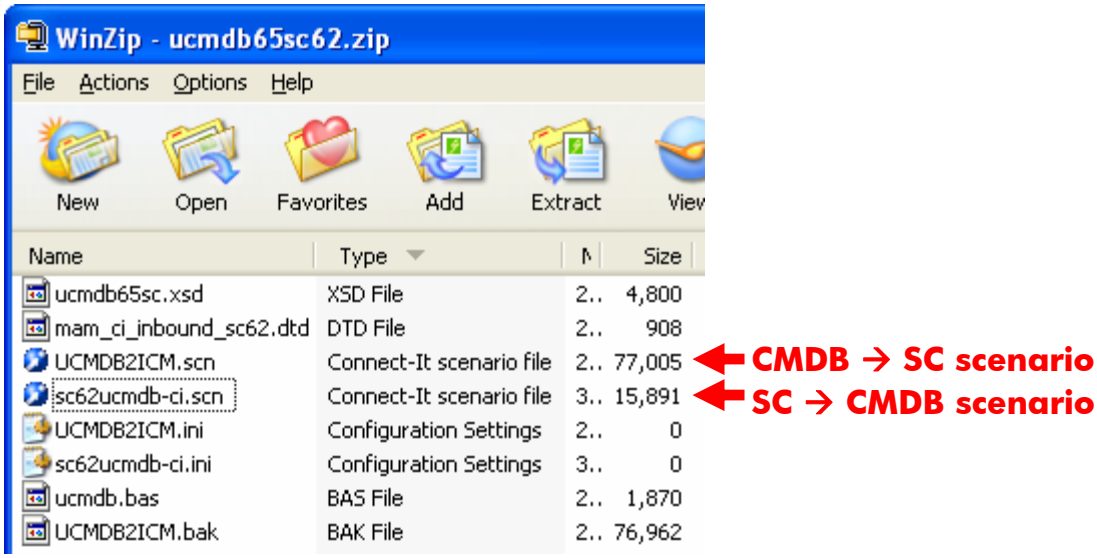
1.1. Depending on the version of uCMDB and SC, choose the appropriate scenario to load. Each scenario contains scenarios for both directions of the integration.

- uCMDB 6.2 → ServiceCenter 6.1 – ucmdb62sc61.zip
- uCMDB 6.2 → ServiceCenter 6.2 – ucmdb62sc62.zip
- uCMDB 6.5 → ServiceCenter 6.1 – ucmdb65sc61.zip
- uCMDB 6.5 → ServiceCenter 6.2 – ucmdb65sc62.zip

1.1.1. Obtain the correct file corresponding to the versions of uCMDB and SC you have.

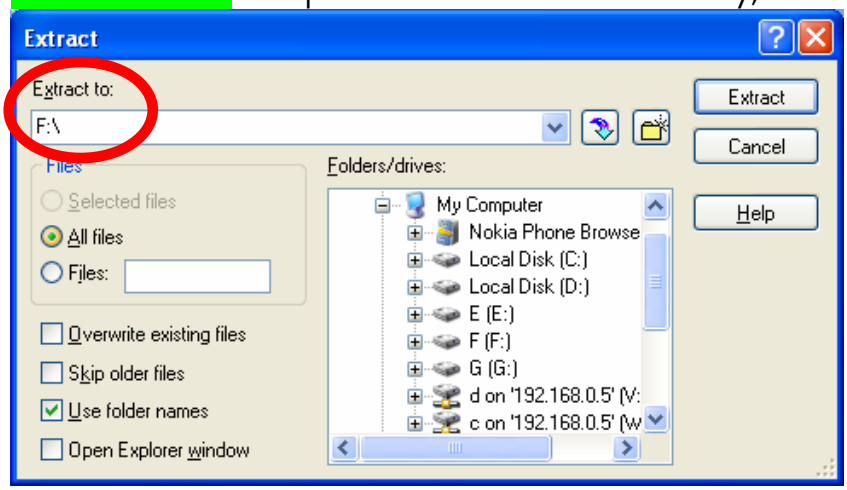


Unzip the files into the Connect-IT directory structure as shown here.



Contents of the scenario file.

Best Practice: Unzip the files to the Drive letter only, like this:

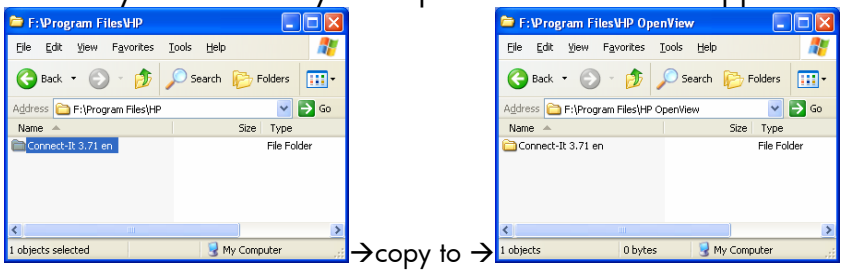


WinZIP extract window showing only drive letter where Connect-IT is installed.

The entire directory structure is contained in the ZIP file. For example, if you installed Connect-IT to the "F:" drive, have uCMDB 6.5, and SC 6.2, you would use the above ZIP file and extract its contents to F:\

If the version of Connect-IT you are using has a directory structure containing "/hp Openview/" instead of "/hp/", you may copy the contents of Program Files\HP\ to Program Files\HP Openview. If you do so, the scenario files will be copied to the existing Connect-IT directory where other out-of-the-box scenarios are stored.

Answer yes to all if any file replacement windows appear.



Copying from the HP folder into the HP Openview folder, if necessary.

2 Open the scenarios and configure the endpoint connectors.

2.1 Load the scenario in Connect-IT scenario builder.

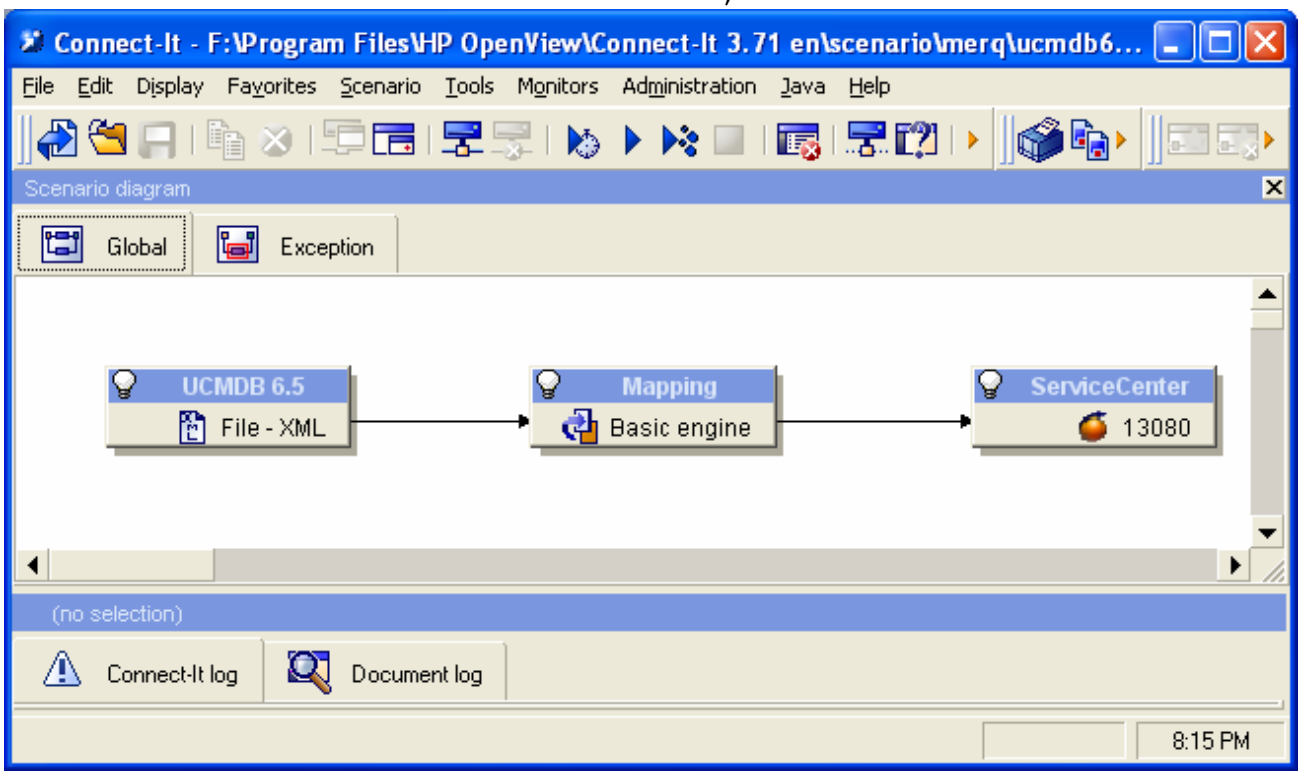
2.1.1 Start Connect-IT by opening the Connect-IT Scenario Builder.

2.1.2 Select **File > Open** and select the appropriate scenario for the versions of the uCMDB and ServiceCenter product that will be used for this integration, as described in sections 2.1.4 and 2.1.5.

2.1.3 Right Click on the UCMDB6.x connector and select **Configure Connector...** A wizard window will now be displayed.

2.1.4 CMDB to SC

2.1.4.1 Use the UCMDB2ICM scenario, which will look like this:



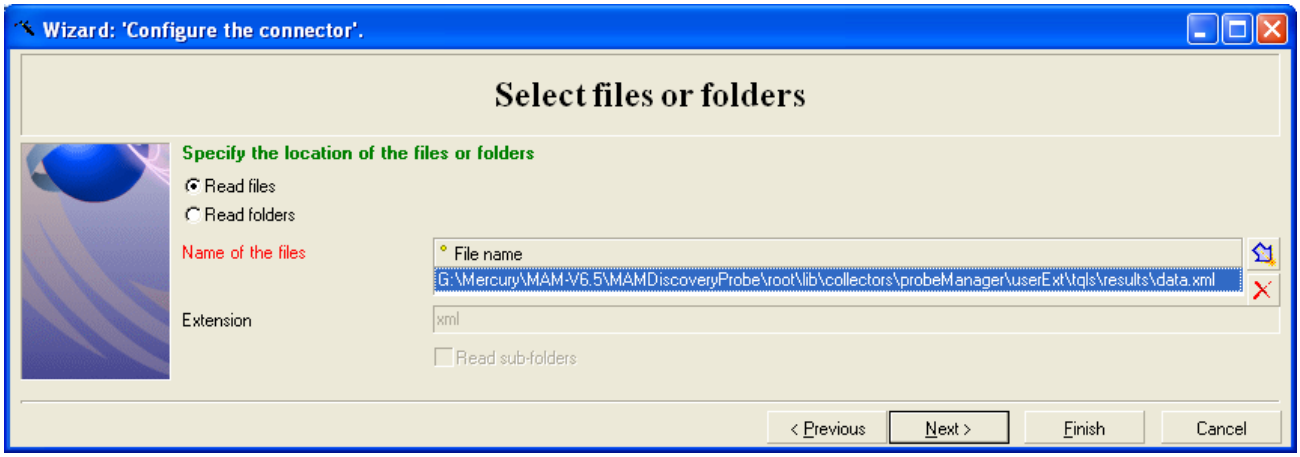
UCMDB2ICM scenario connectors: UCMDB6.5, Mapping, and ServiceCenter.

2.1.4.2 UCMDB6.5 Connector.

2.1.4.2.1 Right-click the UCMDB6.5 Connector.

2.1.4.2.2 Keep clicking on the **Next** button until the "Select files or folders" wizard page is displayed.

2.1.4.2.3 Click on the current value in the File Name box and then click on the directory icon that becomes visible.



Select the XML data file from SC.

2.1.4.2.4 Click on the current value in the File Name box and then click on the directory icon that becomes visible.

This file should be in the tqls/results directory created and discussed in Step 6 in the uCMDB section.

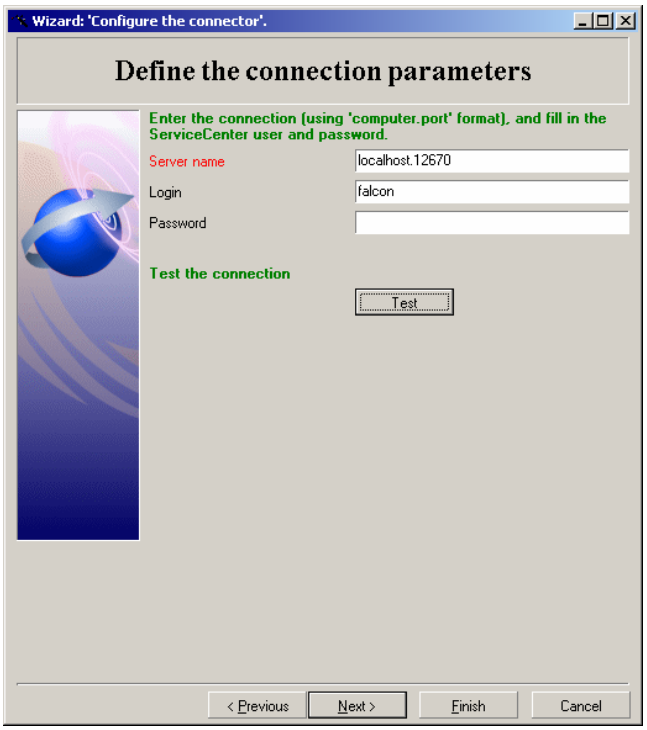
2.1.4.2.5 Click on the current value in the File Name box and then click on the directory icon that becomes visible.

2.1.4.2.6 Click next until the **Finish** button appears, then click **Finish**.

2.1.4.3 Service Center Connector

2.1.4.3.1 Right Click on the ServiceCenter connector and select Configure Connector… A wizard window will now be displayed.

2.1.4.3.2 Click **Next** to display the "Define the connection parameters" wizard page.



SC Connector Configuration

2.1.4.3.3 Populate the Server Name (format is server.port i.e. localhost.12670), Login and Password fields with the appropriate ServiceCenter instance information and click the **Test** button to ensure the information is correct.

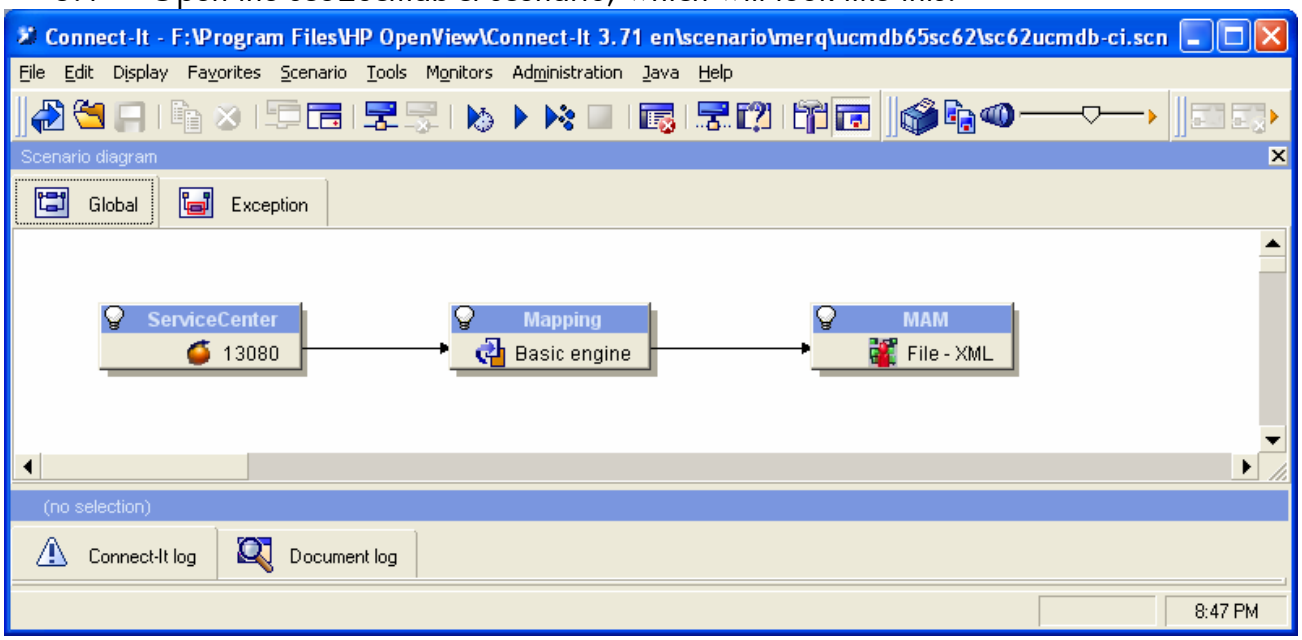
2.1.4.3.4 Click **Close** on the test window.

2.1.4.3.5 If the test fails, verify the connection information is entered properly or validate the configuration of the ServiceCenter instance.

2.1.4.3.6 Click on the **Finish** button.

3. SC to UCMDB

3.1 Open the sc62ucmdb-ci scenario, which will look like this:



SC to uCMDB scenario loaded.

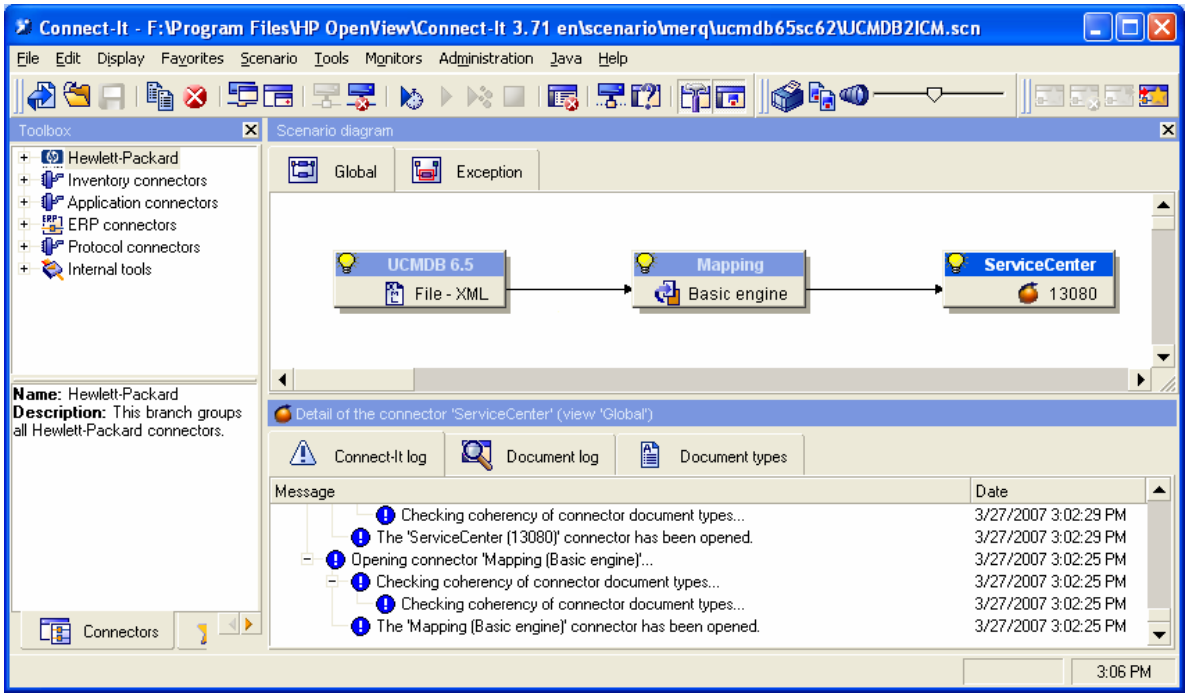
3.2 You will now configure the two endpoint connectors.

3.2.1 **Service Center** - Follow the instructions for step 2.1.4.3 above.

3.2.2 **UCMDB** – Follow the instructions for step 2.1.4.2 above. The references for "read" are generally replaced with "write" for this connector. The file specification should point to the `fromSC` directory a specified in step 9 of the uCMDB section.

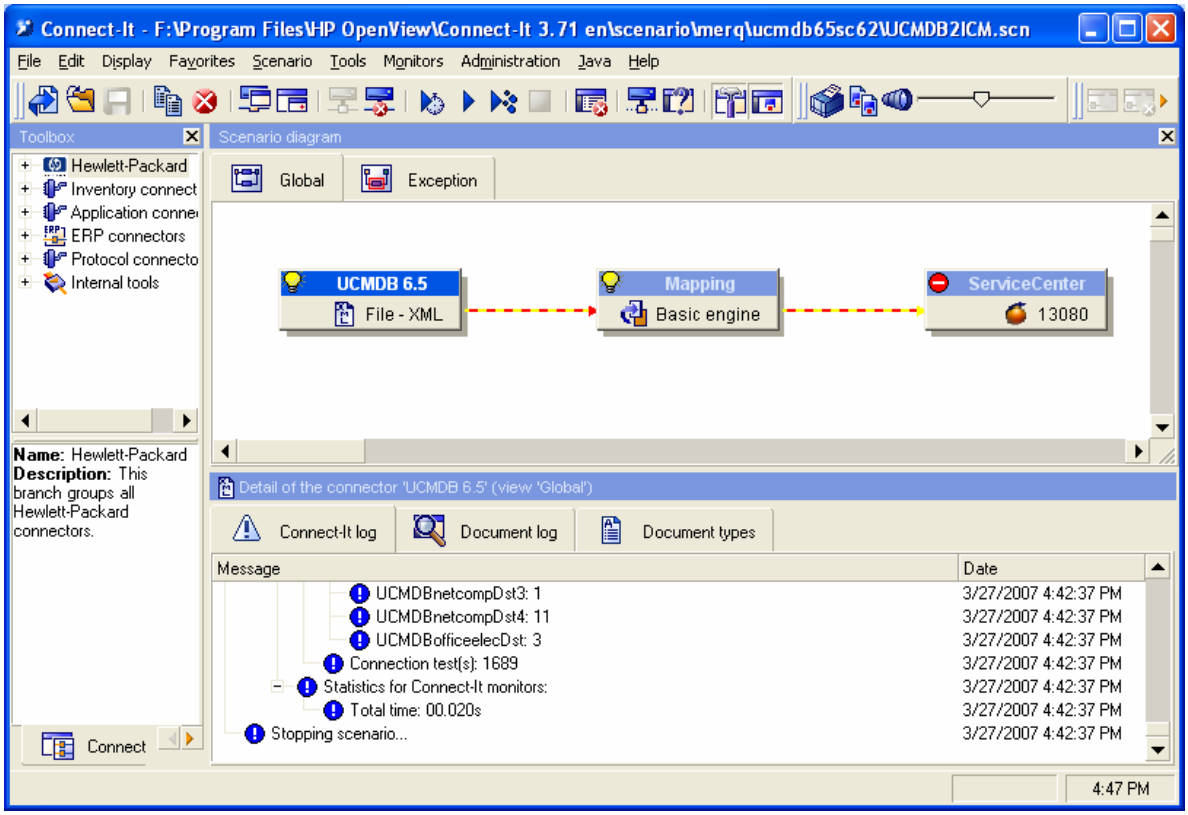
4 Opening the Connectors and testing the Integration

4.1 Select **Scenario > Open all connectors**. The light bulbs, in the upper left corner of each of the connectors, should turn yellow. This indicates the connectors have successfully connected. There should not be any error messages in the Connect.It log window.

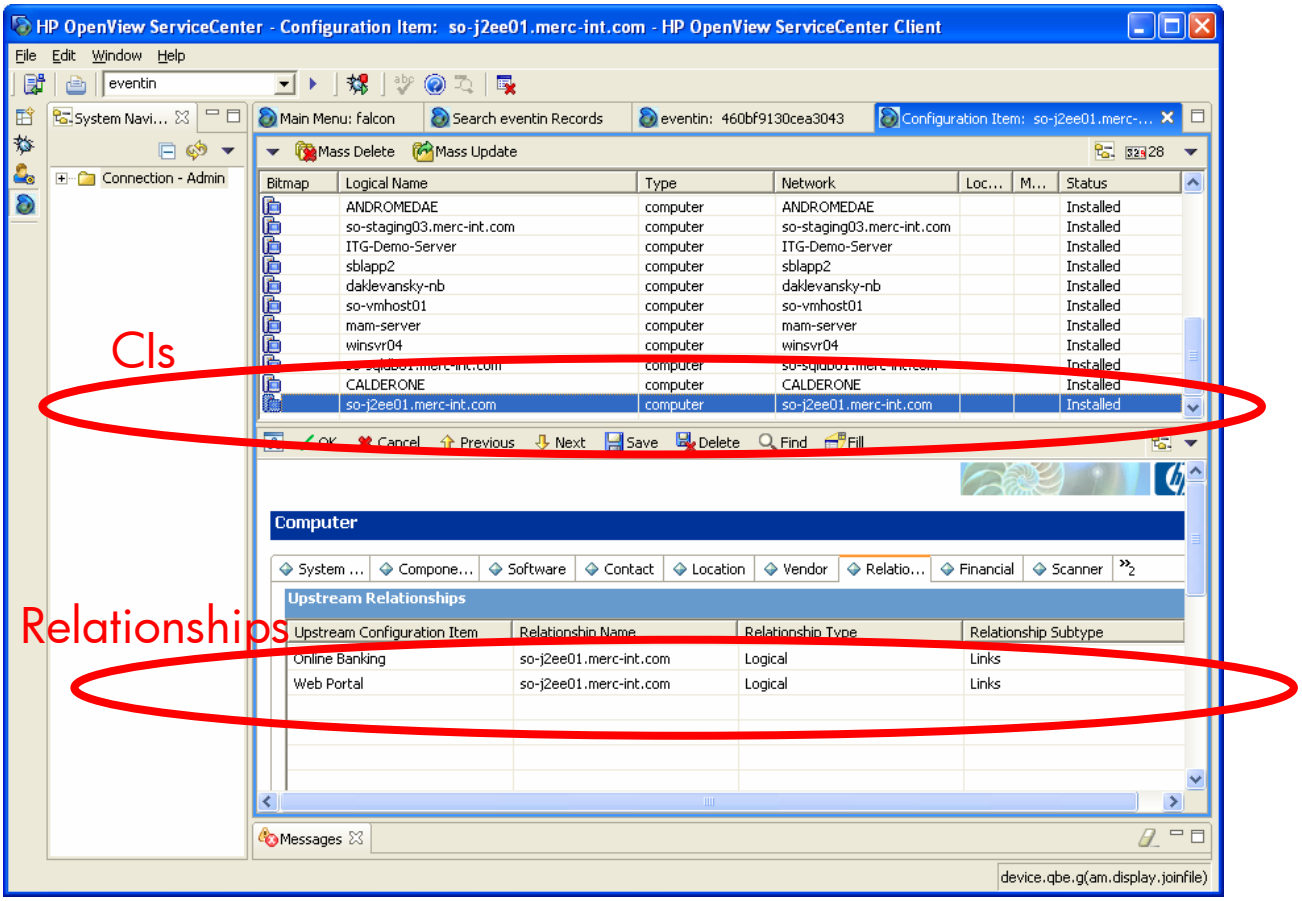


uCMDB Connect-IT scenario with all connectors successfully opened.

1. Right click on the UCMDB6.x connector and select **Produce now** or select **Tools > Produce now**.

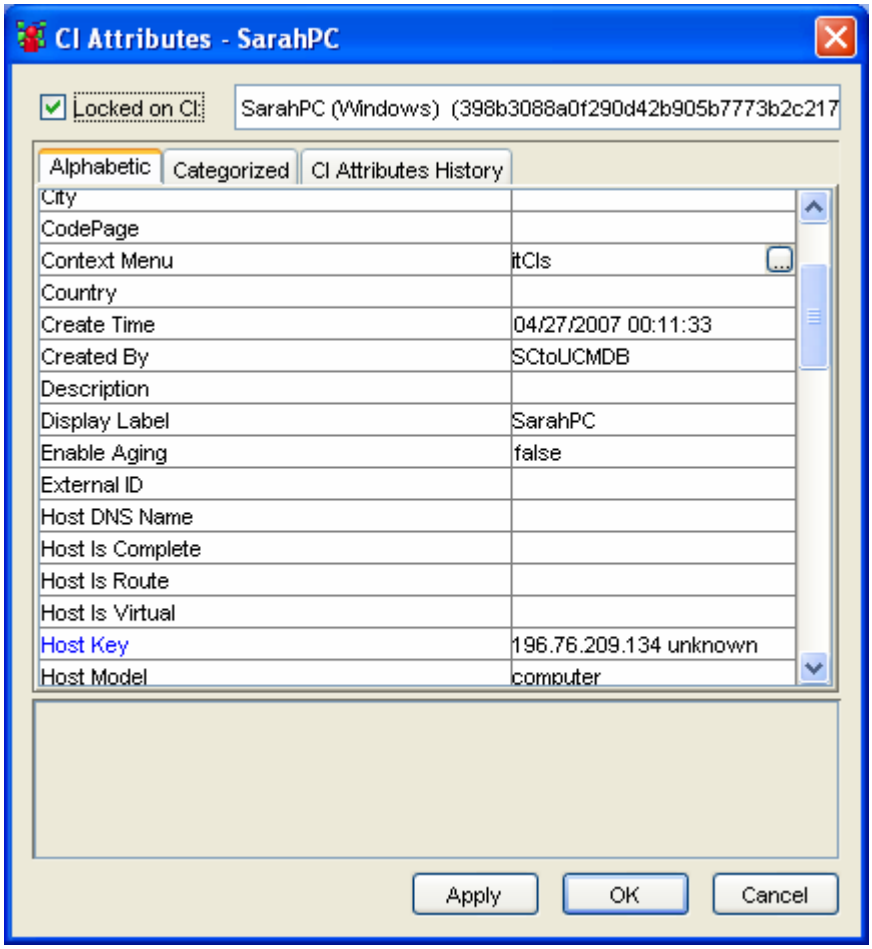


Example showing running (Produced) scenario. The stop light indicates warnings (see Known Issues section)



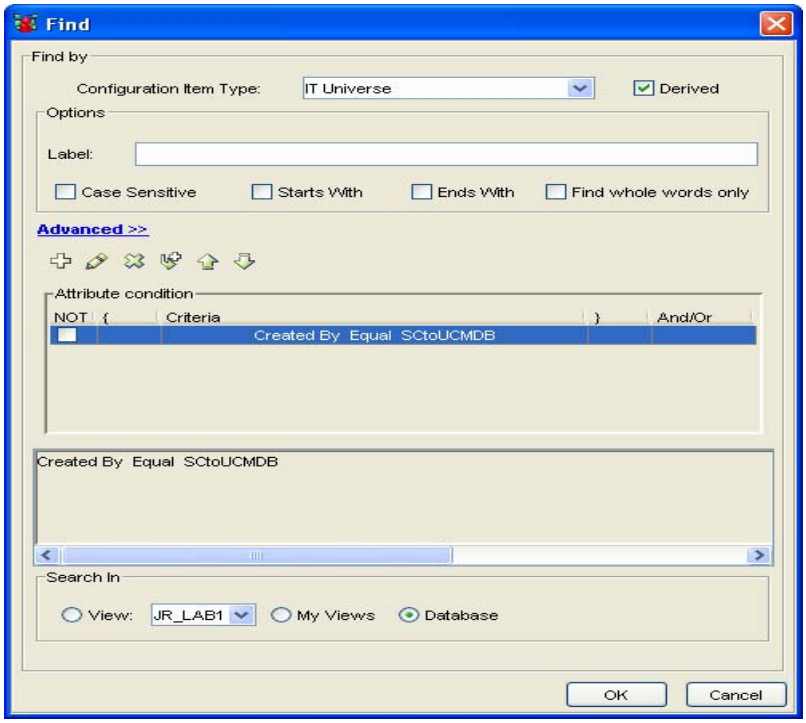
Service Center CI Manager showing CIs and relationships from uCMDB.

You can see CIs in uCMDB that came from Service Center by using a number of tools, whichever is more convenient. If the following CI is imported from SC:



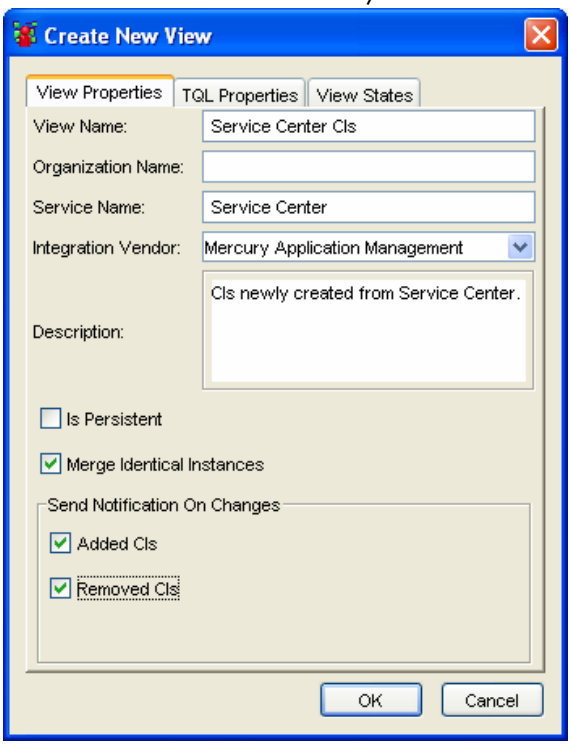
You can use the following tools to see it:

1. The Find tool:

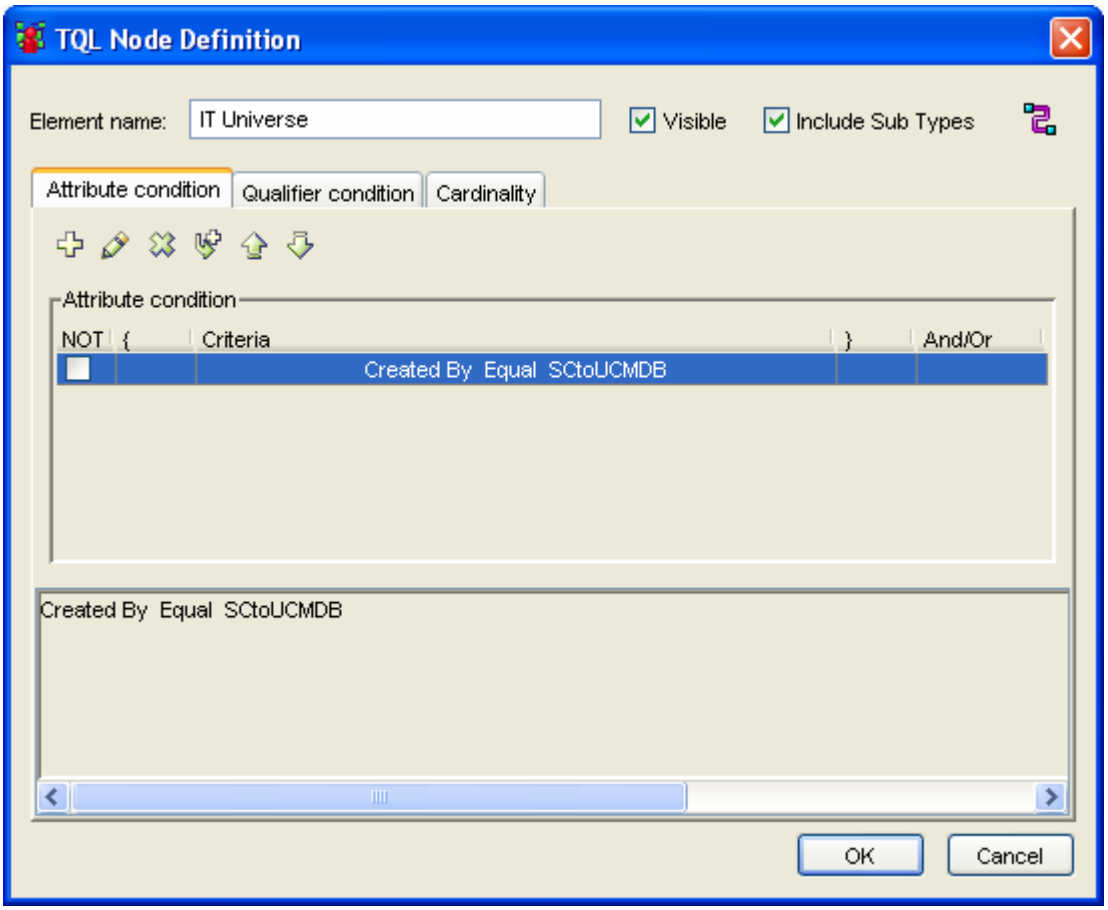


Find tool to find CIs with the created by attribute of the name of the integration pattern. This will find CIs newly created by the integration until another pattern updates the CI.

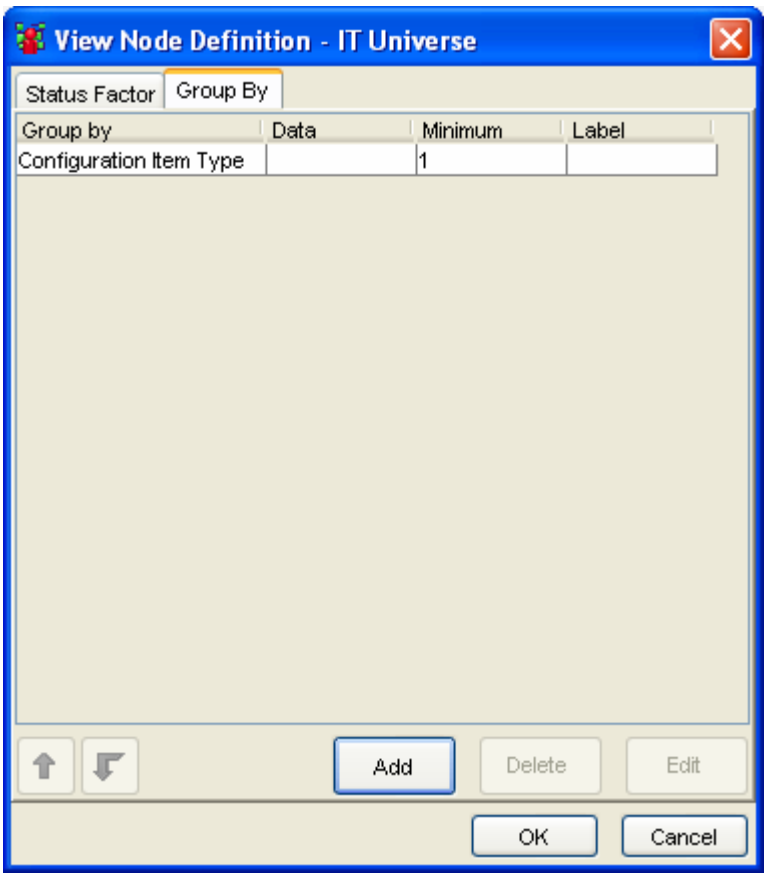
2. Service View/TQL:



Create a Service View. Notify when CIs come and go if needed.

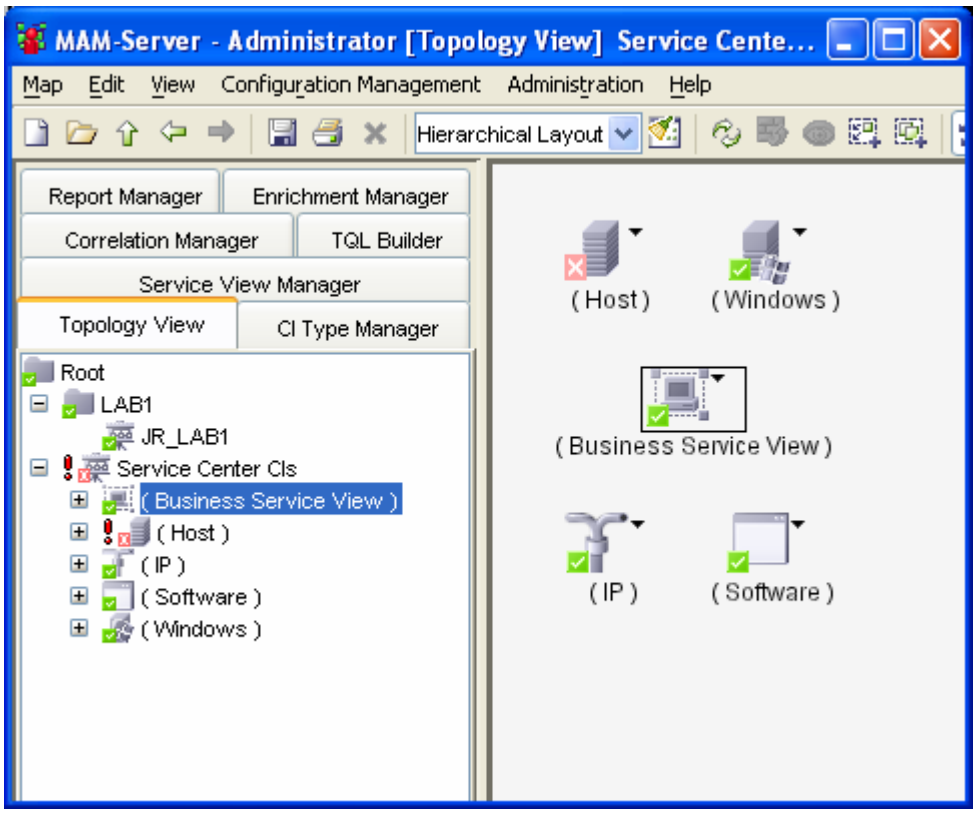


Add a TQL node condition where the "Created By" attribute is equal to "SCtoUCMDB".



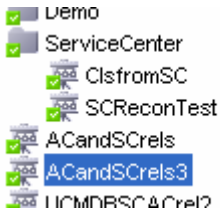
Add a TQL group by condition on CI Type.

This will produce the following topology view:



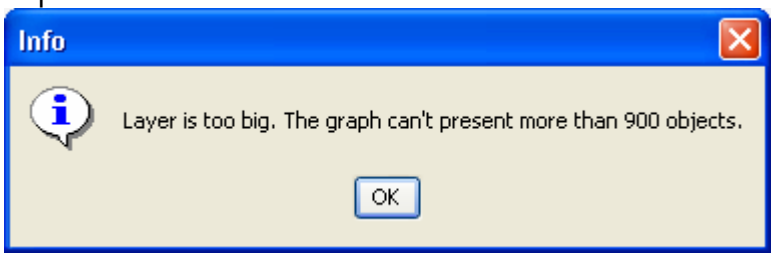
You can also create a system report using the same TQL, and re-use this TQL as needed.

Two sample TQLs are supplied with version 2.1 of this package.

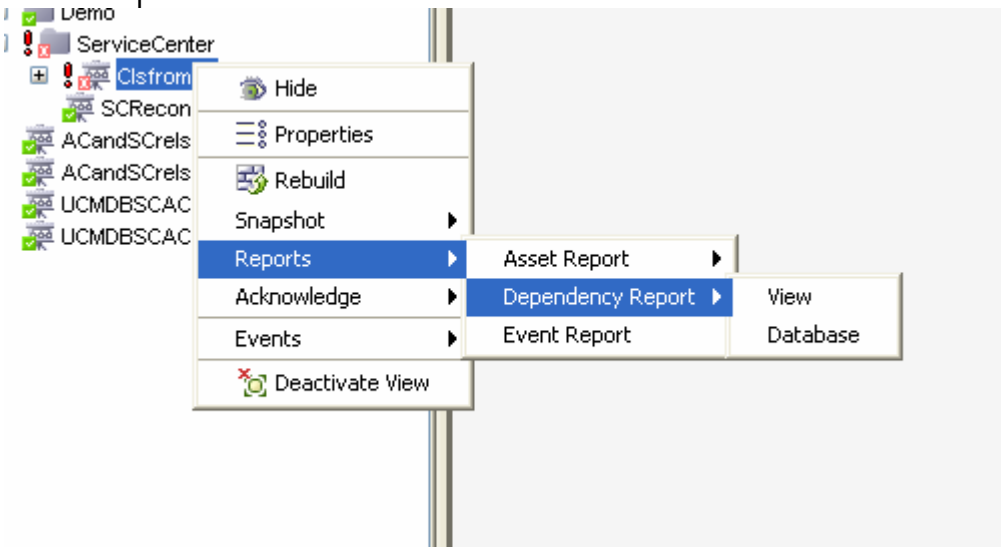


Sample Service views "ClsfromSC" and "SCReconTest"

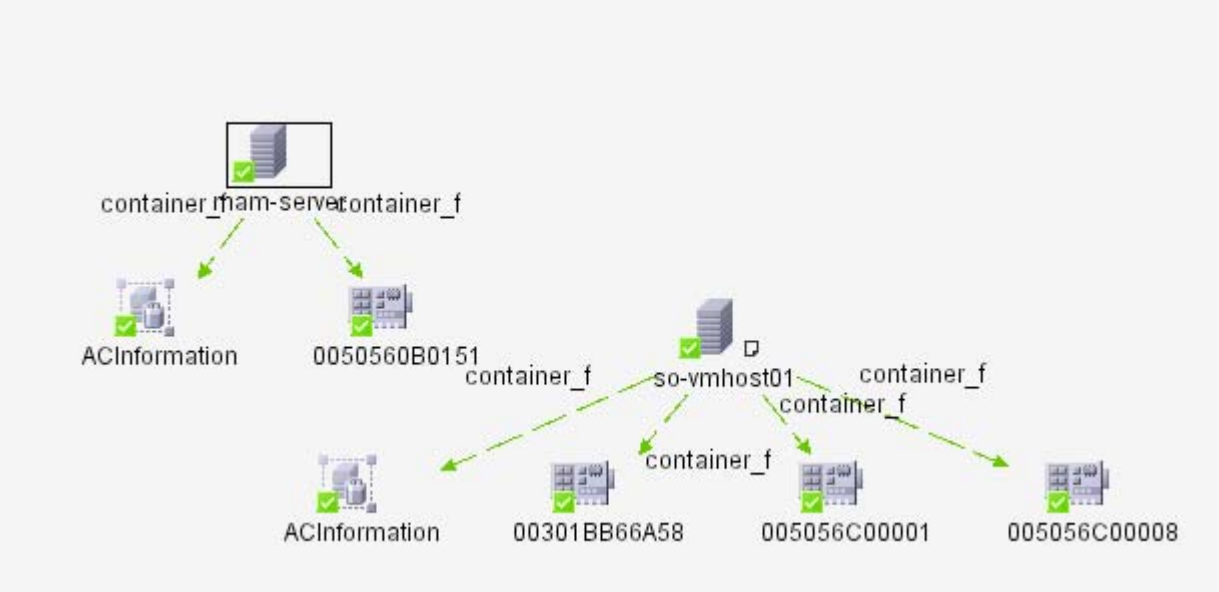
These TQLs show which CIs have been imported from Service Center. If the number of CIs imported exceeds the maximum number of nodes allowed for a topology view:



Then the view can still be used to create reports by right-clicking the view and choosing the correct report:



The SCreconTest view shows CIS which can reconcile with those from Aset Center and within the CMDB:



Sample output of SCReconTest. The upper hosts match in Service Center and Asset Center. The ACInformation labels contain Asset-Center-specific information about the assets referenced in the Service Center CMDB. Note that none of the CIs shown were discovered by the uCMDB.

Activating the Integration

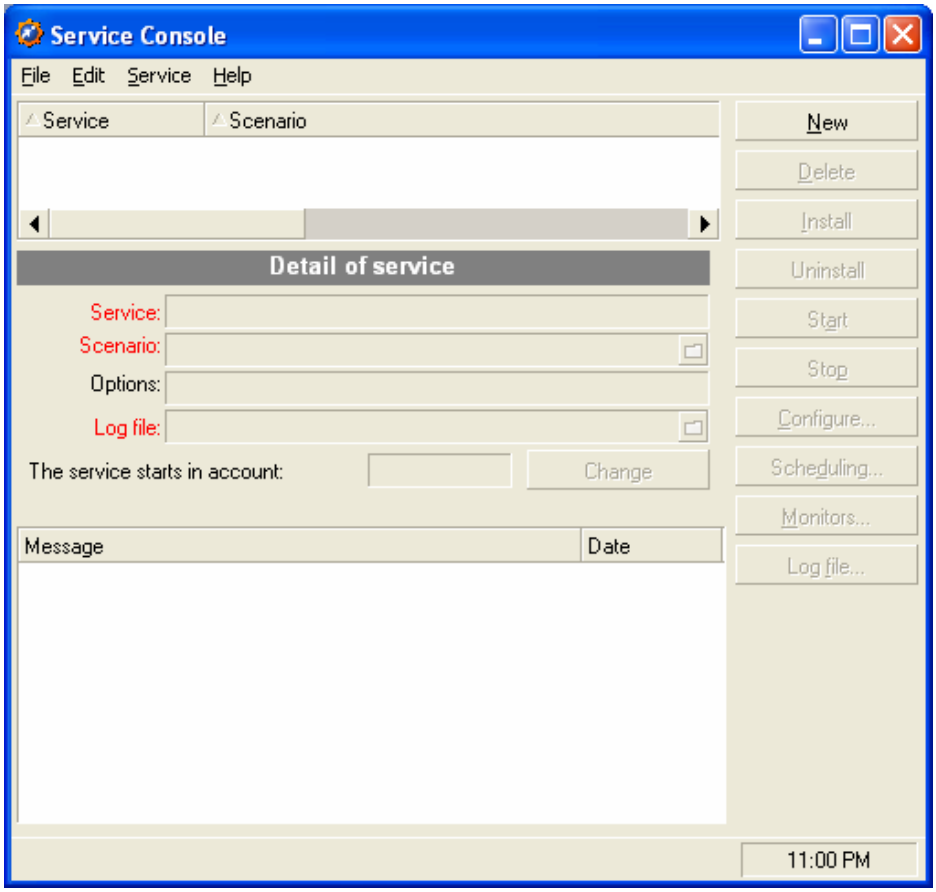
For the uCMDB to SC integration to operate completely:

- The CMDBtoSC pattern should be active in uCMDB
- The CMDB, both uCMDB discovery patterns and both Connect-IT scenarios should be active. The uCMDB section describes how to schedule and activate the uCMDB integration patterns.

Most of this section describes how to deploy a scenario using Connect-IT. The process is done twice, once for each scenario.

1 Deploy the scenarios using the Connect-IT Service Console

1.1 Open the HP Connect-IT Service Console. If no scenarios are deployed, the console will look like this:



Connect-IT Service Console

6.2 Click **New** to deploy a new scenario.

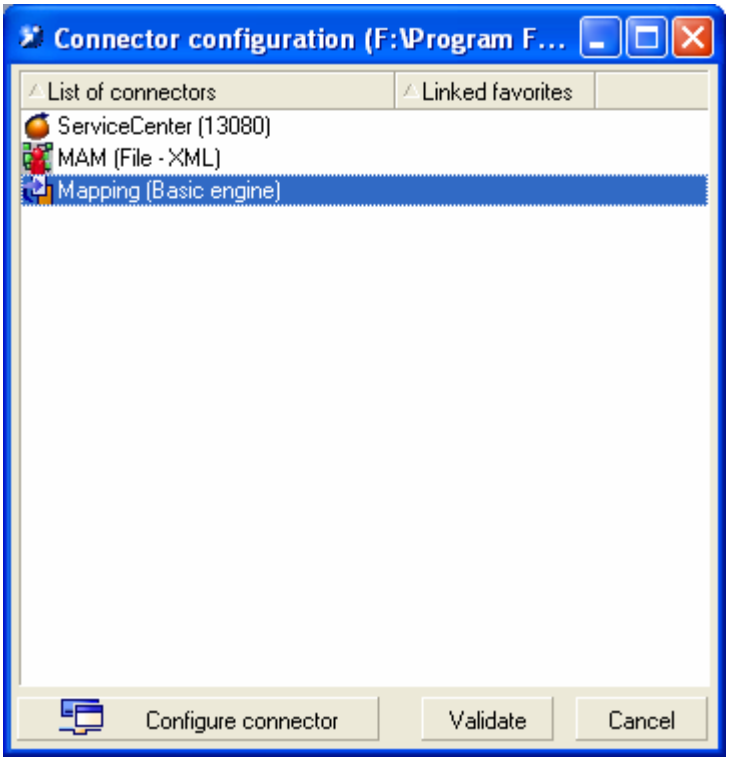
6.3 Complete the details of the Service:

6.3.1 Service: enter uCMDBtoSC or SCtouCMDB.

6.3.2 Scenario: browse to one of the scenarios above.

6.3.3 Leave the other fields blank. A log file will be assigned.

6.3.4 Click **create**, then **Configure**.

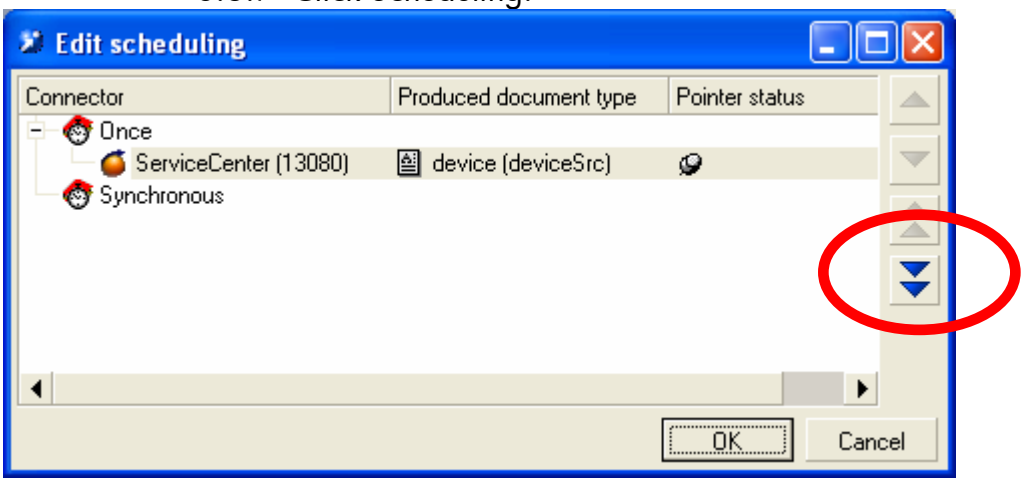


Scenario deployment configuration.

6.3.5 Configure and validate the ServiceCenter and MAM/uCMDB connectors as described in section 4. Follow the same instructions for each scenario's connectors.

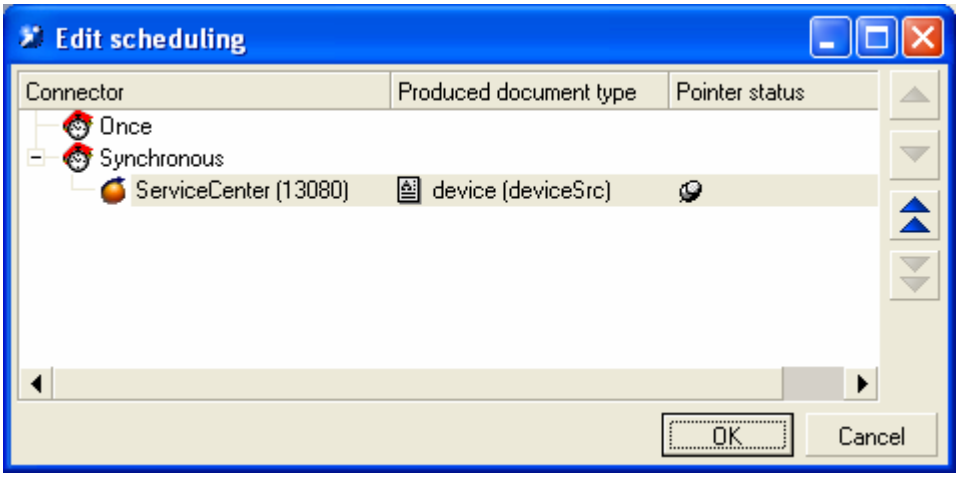
6.3.6 There is no need to configure the Mapping connector unless you wish to extend the integration.

6.3.7 Click Scheduling.



Scheduling Window.

6.3.8 Move the service into the Synchronous schedule by clucking the double-down arrow on the right, then click OK.

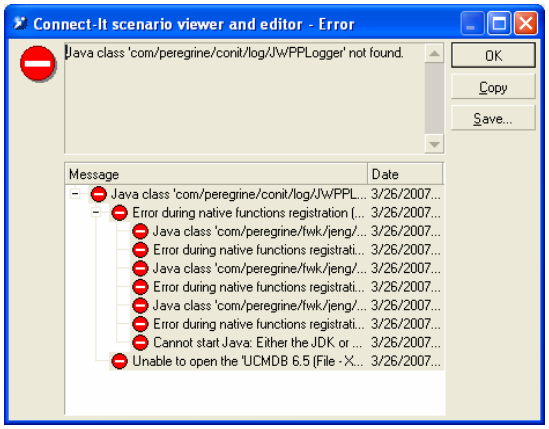


6.3.9 If you need more complex scheduling, click schedulers on the previous window and create a suitable schedule.

6.3.10 Once back on the main console, Click Start to activate the scenario.

Troubleshooting

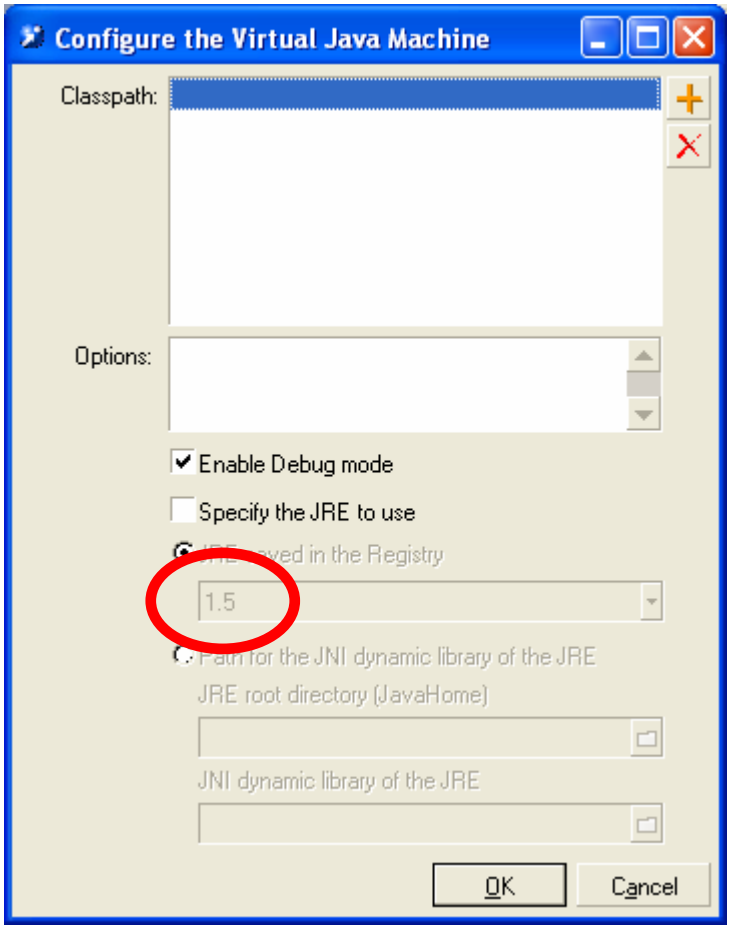
1. The MAM/uCMDB connector will not open, and error messages similar to the following appear:



If you get errors like these when you attempt to open the uCMDB connector, you are using the wrong version of java.

Connect-IT 3.7.1 requires Java 2.0 version 5. You can download this from http://java.sun.com/javase/downloads/index_jdk5.jsp or, go to http://java.sun.com and follow the links.

In the Scenario builder, if you click on **Java→Select JVM...** and you see no java versions this is probably your problem. A correct configuration looks like this:



Java using the 1.5 JRE, required for the MAM/uCMDB Connectors.

a. If a light bulb turns red, check the error message in the log window and rectify the issue. Not all errors are fatal, despite the warnings in the Connect-IT User's Guide on page 214…

Tracking line	Meaning	Message type	Example
	Information about an action	Describes an action performed by Connect-It.	The opening of a connector.
	Details of an action	Gives details concerning an action.	Indication that a particular index is used in the reconciliation process.
	Warning	Warns of any problems that could result in a bad data processing.	The field chosen as a reconciliation key does not have a uniqueness constraint.
	Major problem	Gives the reason why an action failed.	The configuration of a connector is not valid. Unable to communicate with the external application.

The "Major problem" description is not always accurate. This may indicate only a certain attribute was blank, or a CI contained invalid characters.

Known Issues

API characters

The API does not process all characters and sometimes fails on characters discovered previously. If unexpected results occur, investigate the source data, and contact HP Customer Support Organization (CSO) for further assistance. It may be possible to resolve the problem using tools such as uCMDB enrichment or Mass Updates in SC.

Connect-IT connector red lights

The Connect-IT SC connector reports a red stop light, even when CIs were successfully processed. This is normal and does not affect other CIs, as long as the proper setting in the connector configuration is used. Ensure the connector ignores effors and continues processing. This should be the default behaviour; however, this may not be the case for your organization.

4AA0-XXXXENW, May 2006