CHAPTER 2

Understanding Configuration Manager components

Components form the basis of the architecture of System Center 2012 Configuration Manager and they work together to implement different functionality. You can install all the components on the site server or, alternatively, you can separate different components to other servers to offload some of the work from the site server to improve the performance.

This book doesn't cover all the components but focuses on the following ones which are heavily used by many administrators:

- Content distribution
- Pull distribution points
- Software update points
- Troubleshooting rotating management points and failover software update points
- Application deployment troubleshooting

A thorough understanding of how the various Configuration Manager components work together is essential for successful troubleshooting when problems arise. The goal of this chapter is to help build such an understanding.

Content distribution

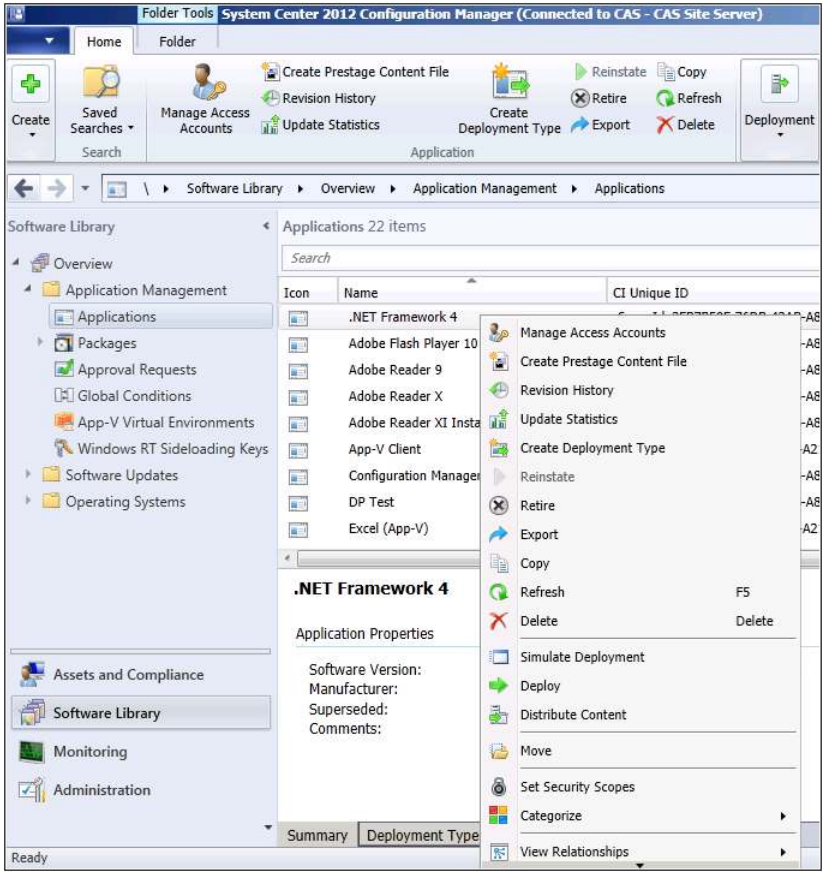
When you install the Distribution Manager role on a site server, the Site Component Manager (SMS_SITE_COMPONENT_MANAGER) triggers the installation of the role and invokes the related component for installation. This section examines the Distribution Manager and other components used when distributing content to distribution points.

Sending packages/applications to distribution points

When deploying any applications or packages, packages must be sent to a distribution point. Configuration Manager clients then download the package from the distribution point. If packages/applications are not distributed to distribution points, the clients will be unable to find the package and they won't be able to deploy that application.

The process for sending a package/application to a distribution point is as follows:

1. Open the Configuration Manager Console and click Software Library, and then Application Management.
2. Click Applications or Packages to see the list of created applications or packages.
3. Right-click one of the applications or packages and then select Distribute Content.



4. Follow the wizard to add the required distribution points.

Examining the log files

Understanding Configuration Manager components helps you troubleshoot issues when they arise. A good way to learn how these components work together is by reviewing the various log files that Configuration Manager uses. Verbose logging can also be configured to provide further information concerning components.

Let's look at what actually happens when you distribute content to distribution points. When you add a package or application to a distribution point, the SMS_DATABASE_NOTIFICATION component updates the database with the information and you can review the details in smsdbmon.log as shown in Figure 2-1.

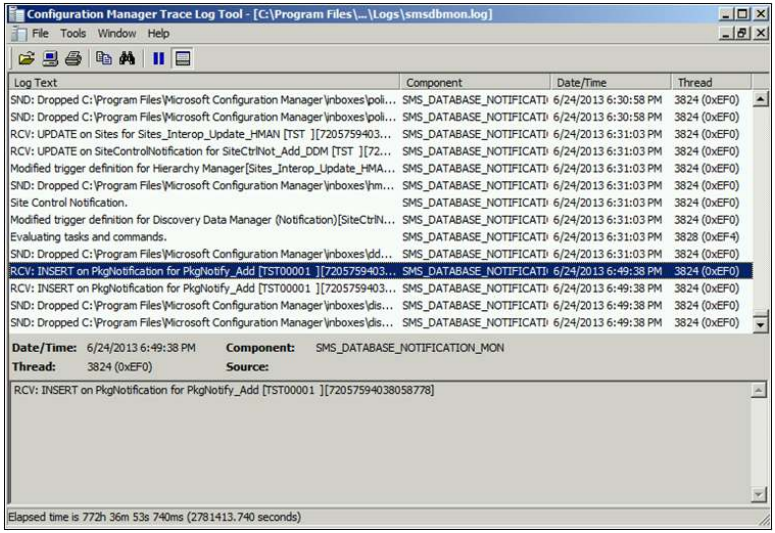


FIGURE 2-1 A package notification is inserted in the smsdbmon.log.

The Distribution Manager (SMS_DISTRIBUTION_MANAGER) component then starts the process of adding the package to the distribution point. This information is logged in the distmgr.log file as shown in Figure 2-2.

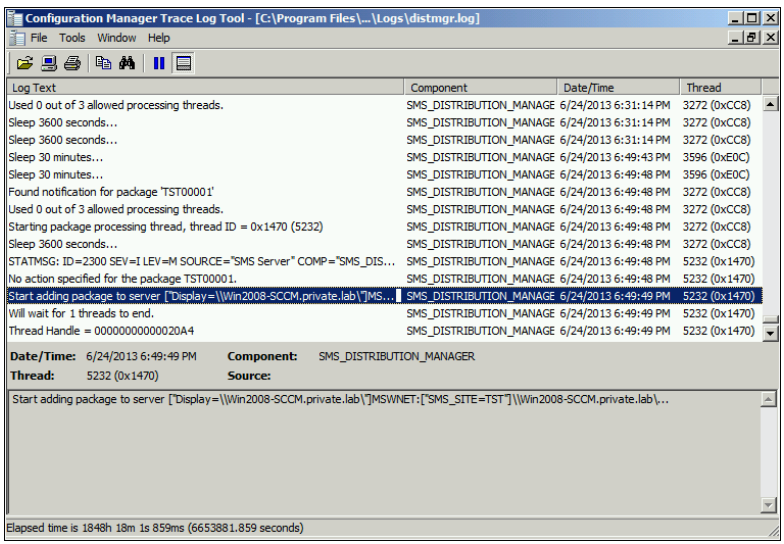


FIGURE 2-2 A package is being added to distribution point.

If for any reason the Distribution Manager fails to send the package to the distribution point, it will log the resulting errors in the distmgr.log. We'll look at the distmgr.log again later in this chapter.

Package Transfer Manager

What if you have second distribution point that is remote from your primary site server? System Center 2012 Configuration Manager introduces a new component called Package Transfer Manager that is used to distribute packages to a remote distribution point.

The process of troubleshooting deployment of applications and packages to remote distribution points is similar to what was described previously except that Package Transfer Manager (not Distribution Manager) is used to transfer the application or package to the remote distribution point.

Monitoring distribution of content to remote distribution points

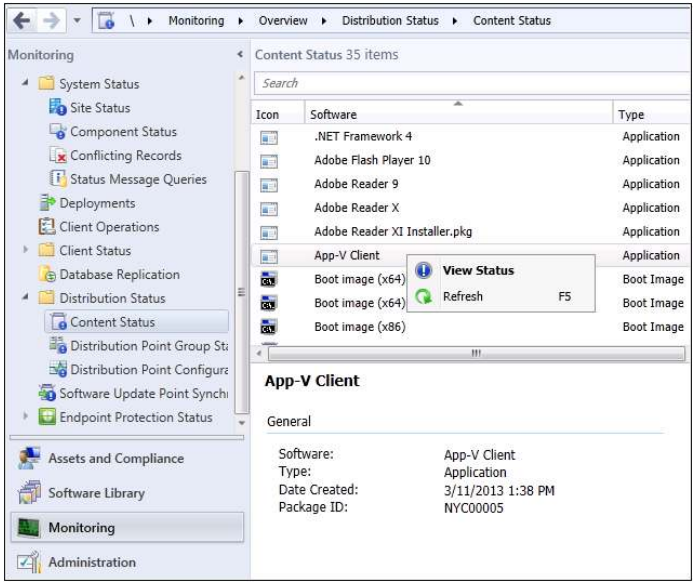
When you distribute content to a remote distribution point, there are two ways to monitor progress:

- Using the Monitoring workspace in the console
- Using the Package Transfer Manager log (PkgXferMgr.log)

Using the Monitoring workspace

To monitor progress in distributing content to remote distribution points using the Configuration Manager console, follow these steps:

1. Connect to the Console and then select Monitoring | Distribution Status | Content Status. Highlight the application you want to monitor and review the Completion Statistics in the lower half of the window. Click View Status for additional details.



2. Click the various tabs such as Success, In Progress, Error, and Unknown and review the details. For example, click the Error tab to review errors on why distribution of content is failing.
3. Under Asset Details, review the data and click More Details to view the description of the errors.

Using PkgXferMgr.log

Sometimes the Monitoring workspace might not provide you with enough information to troubleshoot an issue relating to the distribution of content to a remote distribution point. In such cases, your next step should be to examine the Package Transfer Manager log (PkgXferMgr.log) for further details concerning the process.

For example, if the Content Status indicates that the server's computer account does not have access to the package source or the distribution point doesn't have enough disk space, what should you do? First, review your environment to make sure that the computer account has proper access and that there is enough disk space on the remote distribution point.

If the problem persists, review the PkgXFerMgr.log on the primary site server. The following log entry is a potential error for the application:

```
ExecStaticMethod failed (80041001) SMS_DistributionPoint, AddFile
        SMS_PACKAGE_TRANSFER_MANAGER        7/26/2013 2:07:43 PM       5152 (0x1420)
CSendFileAction::AddFile failed; 0x80041001     SMS_PACKAGE_TRANSFER_MANAGER    7/26/2013
2:07:43 PM        5152 (0x1420)
```

```
CSendFileAction::SendFiles failed; 0x80041001  SMS_PACKAGE_TRANSFER_MANAGER   7/26/2013
2:07:44 PM       5152 (0x1420)
CSendFileAction::SendFiles failed; 0x80041001  SMS_PACKAGE_TRANSFER_MANAGER   7/26/2013
2:07:44 PM       5152 (0x1420)
Notifying pkgXferJobMgr  SMS_PACKAGE_TRANSFER_MANAGER                        7/26/2013
2:07:44 PM       5152 (0x1420)
Sending failed. Failure count = 7, Restart time = 7/26/2013 2:37:44 PM Eastern Daylight
Time    SMS_PACKAGE_TRANSFER_MANAGER       7/26/2013 2:07:44 PM      5152 (0x1420)
Sent status to the distribution manager for pkg LA100005, version 2, status 4 and
distribution point

["Display=\\Cm12PRINA.Contoso.com\"]MSWNET:["SMS_SITE=LA1"]\\Cm12PRINA.Contoso.com\
         SMS_PACKAGE_TRANSFER_MANAGER      7/26/2013 2:07:44 PM     5152 (0x1420)
```

What does this log tell you? It has an error code 0x80041001 which means "Generic Failure – Source: WMI." It is not giving you any information other than that it is a generic failure.

Next, review the smsdbprov.log on the remote distribution point. The following log excerpt shows that an error is being thrown:

```
Error Code 0x80040154 means "Class not registered"
Remote DP – smsdpprov.log: (located on remote DP under C:\SMS_DP$\sms\logs folder):
[1608][Fri 07/26/2013 15:46:18]:Failed to add file 'ccmsetup.cab' to content library.
Error code: 0X80040154
[1920][Fri 07/26/2013 15:52:46]:CFileLibrary::AddFile failed; 0x80040154
[1920][Fri 07/26/2013 15:52:46]:CFileLibrary::AddFile failed; 0x80040154
[1920][Fri 07/26/2013 15:52:46]:CContentDefinition::AddFile failed; 0x80040154
[1920][Fri 07/26/2013 15:52:46]:Failed to add file 'ccmsetup.exe' to content library.
Error code: 0X80040154
Remote DP – smsdpprov.log:
[10DC][Fri 07/26/2013 16:10:42]:Content 'Content_e89f02f4-6fa0-41d8-b9da-2cdaadf6b82f.1'
for package 'LA100005' has been added to content library successfully
[E64][Fri 07/26/2013 16:18:38]:Content 'CAS00001.3' for package 'CAS00001' has been
added to content library successfully
[564][Fri 07/26/2013 16:23:04]:Content 'CAS00002.3' for package 'CAS00002' has been
added to content library successfully
```

The error code 0x80040154, which is explained as "Class not registered," indicates that there might be some class or component missing on the remote distribution point. Your next step would be to review the prerequisites for distribution points as listed on Microsoft TechNet at *http://technet.microsoft.com/en-us/library/gg682077.aspx* to ensure all the prerequisites have been met. First on the list of prerequisites is the Remote Differential Compression (RDC) component which you discover is missing on a remote distribution point running Windows Server 2008 R2. In this case, you go ahead and install the RDC component on your remote distribution point. After the RDC component has been installed, the content distribution process finishes and the application is successfully installed on the remote distribution point.

As you can see in this example, one of the error codes (0x80041001) was not useful but the second one (0x80040154) at least provided you with a hint. So the lesson learned here is to always check all of the appropriate logs before spending too much time looking for other possible causes of your problem.

Pull distribution points

Microsoft System Center 2012 Configuration Manager SP1 introduces a new type of distribution point called a *pull distribution point*. The task of distributing content to a large number of distribution points puts a huge load on a site server, especially the Distribution Manager (distmgr) and Package Transfer Manager (pkgxfermgr) components of the site server. Basically, the Distribution Manager becomes a bottleneck, and this is why the previous recommendation in the RTM release of System Center 2012 Configuration Manager was to have not more than 250 distribution points per site.

You can examine this problem in more detail with the help of some diagrams. In Figure 2-3 you can see a primary site connected to three distribution points. Two of them are connected with 100 Mbps links and one is connected with a 2 Mbps link. All of these distribution points are under same distribution group.

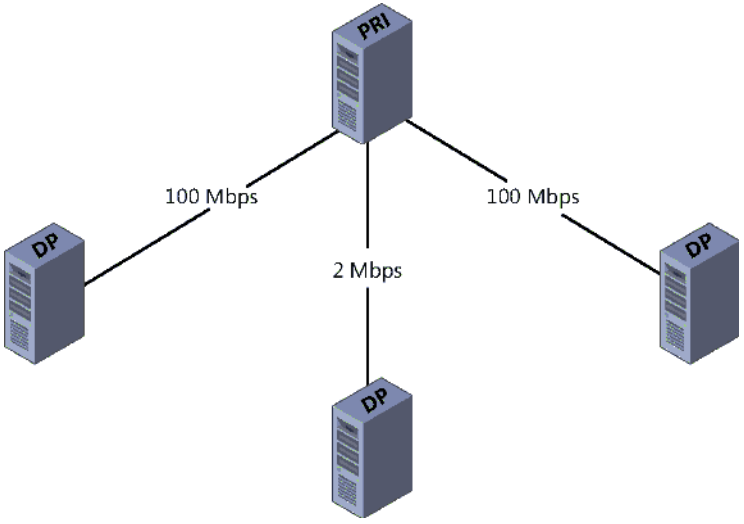


FIGURE 2-3 Three distribution points complete this content distribution scenario.

Once you start distributing content from the primary site, the content will route to all the distribution points via Distribution Manager. However, since the originating source is the same in all the distribution points, the Distribution Manager and Package Transfer Manager components are under heavy load.

Figure 2-4 shows the new pull distribution scenario supported by System Center 2012 Configuration Manager SP1. Instead of having to get the content from the primary site, a distribution point can pull the content from the nearest distribution point. Pull distribution points still allow you to specify where each distribution point resides in the hierarchy but also gives you the flexibility of defining the source distribution point. The result also allows you to overcome the previous limitation of a maximum of 250 distribution points and helps reduce the load of content distribution on primary sites.

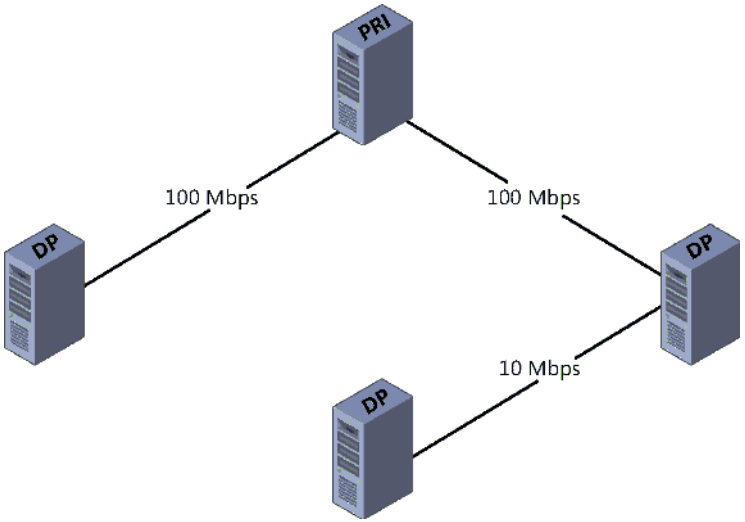


FIGURE 2-4 An example of a pull distribution scenario.

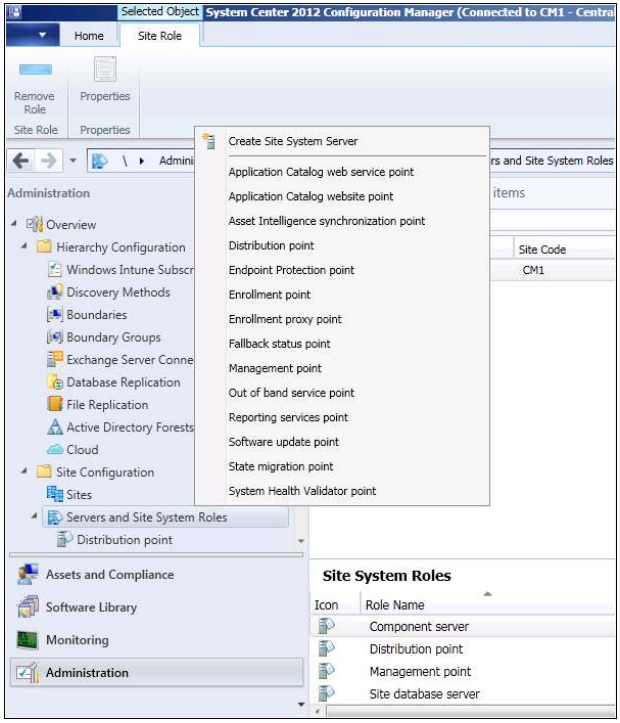
IMPORTANT **Background Intelligent Transfer Service (BITS) is used for transferring content to pull distribution points. This means you can configure BITS throttling using Group Policy to throttle downloads.**

Installing a pull distribution point

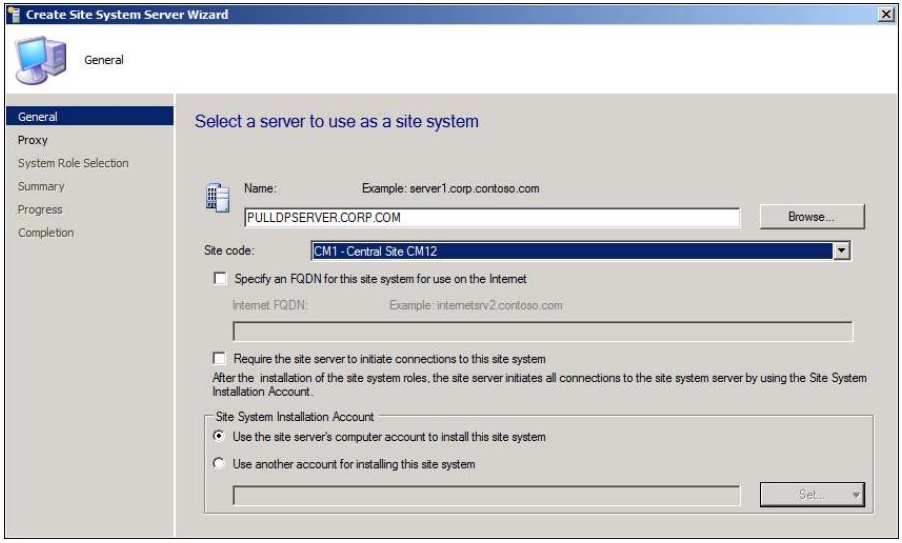
This section describes how to install a pull distribution point. It also shows how to verify installation with the help of the relevant log files.

Follow these steps to install a pull distribution point:

1. In the Configuration Manager console, select the Administration workspace, Site Configuration, right-click Servers And Site System Roles, and then select Create Site System Server:

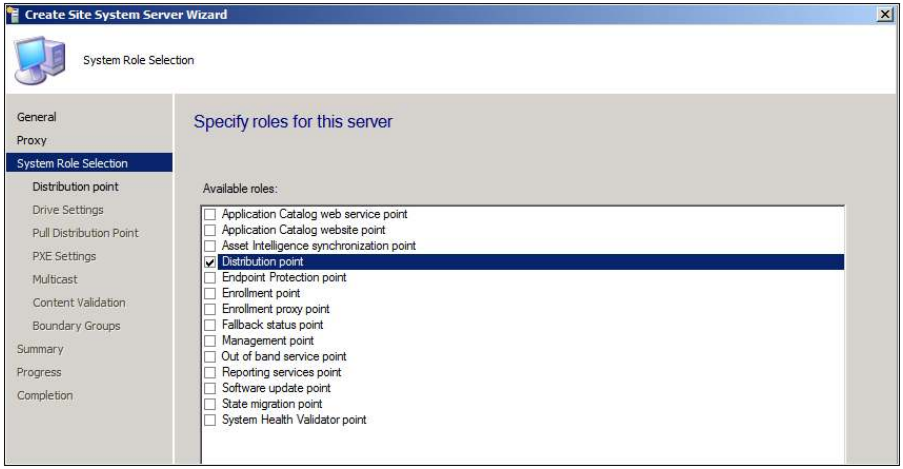


2. On the General page of the Create Site System Server Wizard, specify the name of the server you want to designate as a pull distribution point:

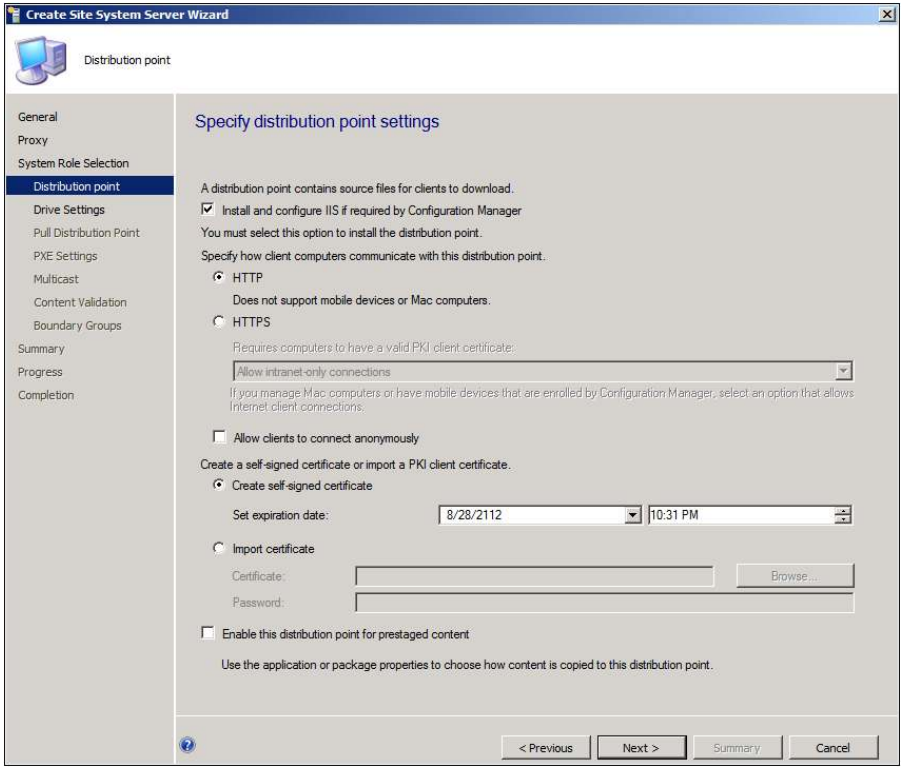


3. Click Next and then Next again on the Proxy page.

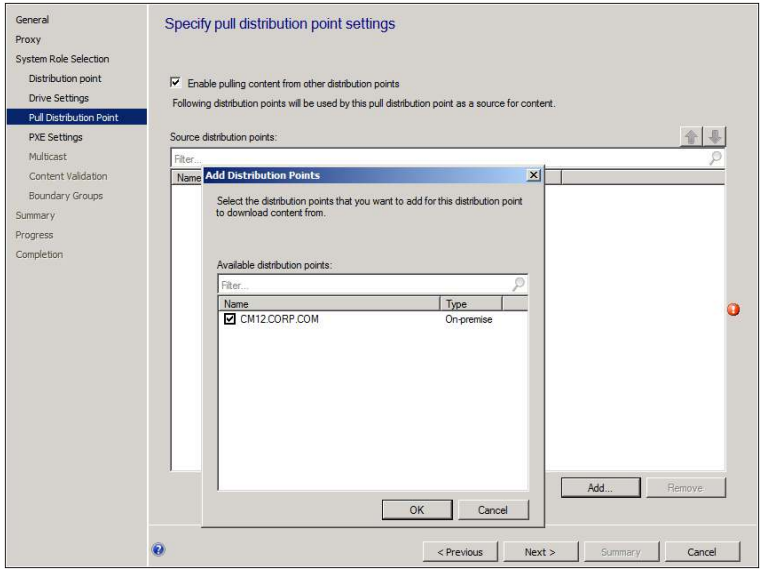
4. On the System Role Selection page, select Distribution Point as the role and then click Next:



5. On the Distribution Point page, select the Install And Configure IIS If Required By Configuration Manager check box and then click Next:

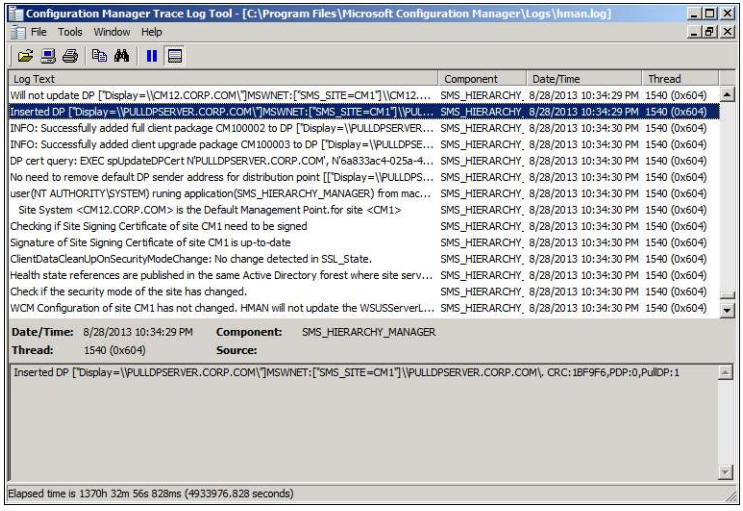


6. On the Pull Distribution Point page, select the Enable Pulling Content From Other Distribution Points check box. Then, under Source Distribution Points, click Add and use the Add Distribution Points dialog box to add the distribution point you want to act as the source distribution point:

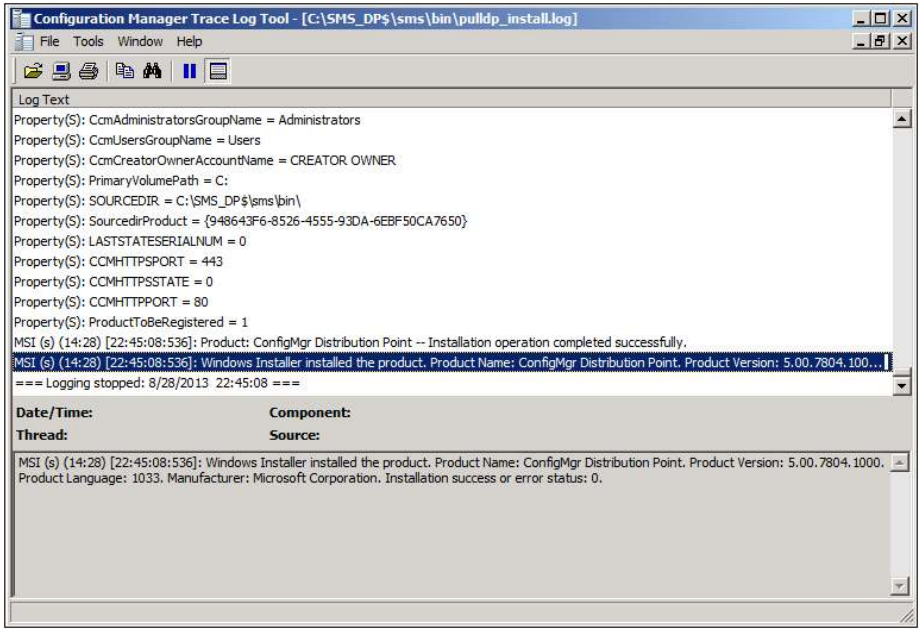


7. Click OK and then click Next and complete the remaining wizard pages.

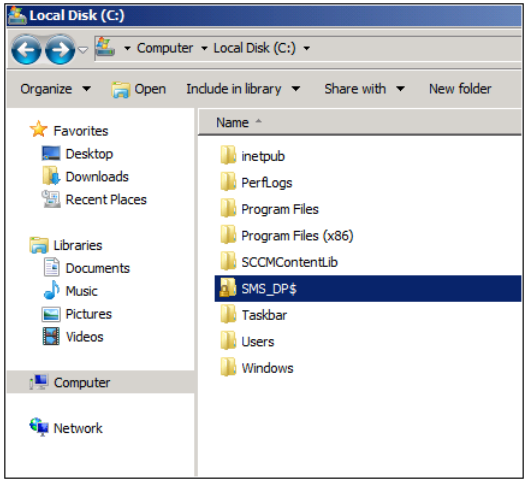
8. After installation of the pull distribution point is finished, you will see the following entry in the hman.log :



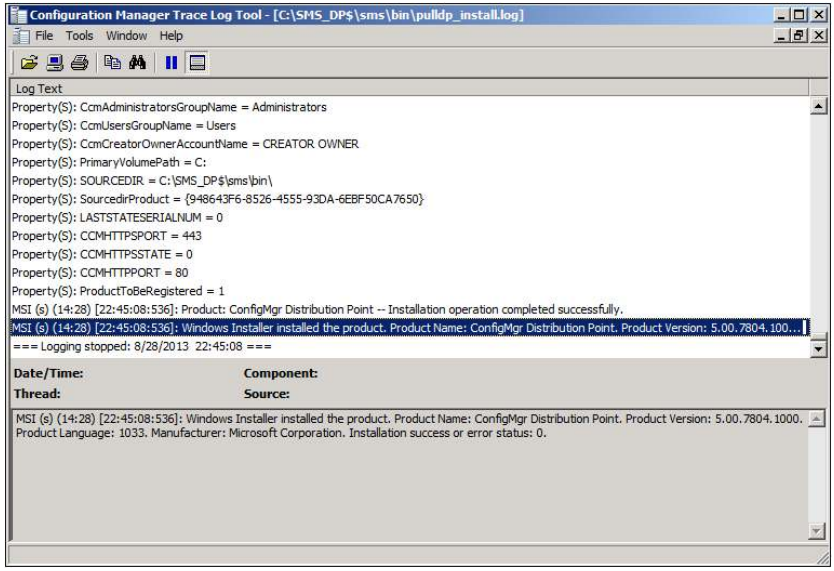
The distmgr.log on the primary site server will look like this:



9. If you now open Windows Explorer on the server where the pull distribution point has been created, you will see that the following folder structure has been created.



10. To further verify the installation of the pull distribution point, review the log files under \SMS_DP$\SMS\BIN\pulldp_Install.log on the server where the pull distribution point resides:



Troubleshooting pull distribution point installation

You can use the log files to troubleshoot issues involving installation of pull distribution points. As an example of how to do this, let's say that while installing a pull distribution point in a lab environment you encounter the error shown in Figure 2-5.

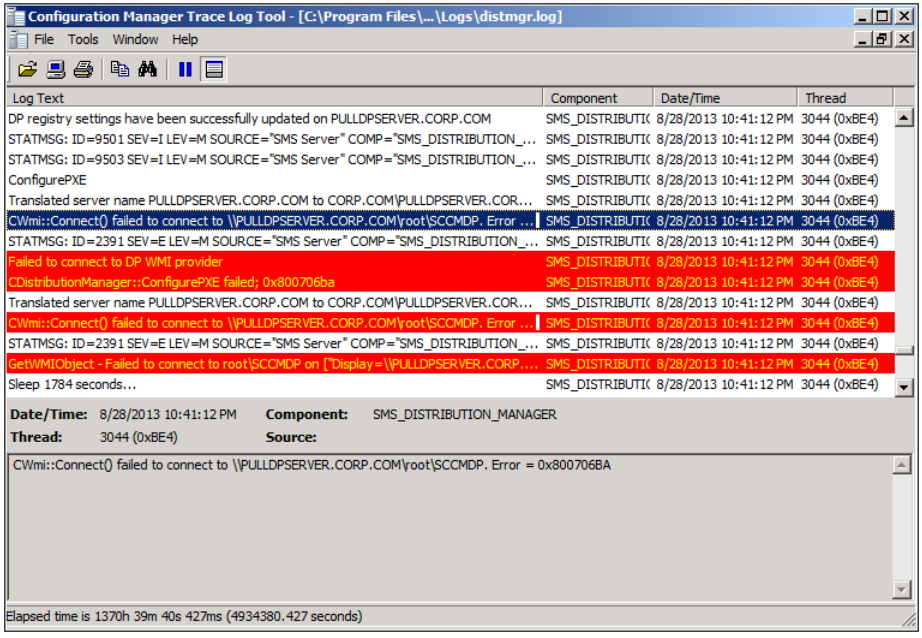


FIGURE 2-5 The distmgr.log displays a pull distribution point error.

In addition, during your troubleshooting of this issue you checked the pull distribution point server and discovered that none of the expected folders were created on it.

The distmgr.log excerpt shown in Figure 2-5 shows that Distribution Manager has failed to connect with the Windows Management Instrumentation (WMI) provider on the pull distribution point. When you get a WMI error like this, you should perform the following steps to troubleshoot:

1. Check Windows Firewall on the pull distribution point server to see if the connection to the remote WMI provider is being blocked.
2. Check to see if an anti-virus program might be blocking the communication.
3. Verify that the site server's computer account (for example, PrimaryServer$ if PrimaryServer is the name of the server) is part of the local Administrator group on the pull distribution point server.

For example, you might discover that the site server's computer account is not a member of the local Administrator group on the pull distribution point server. In this case, your problem will be solved as soon as you add the site server computer account to the pull distribution point local Administrator group.

Software update points

Software update point (SUP) in Configuration Manager is a required component for software updates on primary sites and an optional component for software updates on secondary sites. It is installed as a site system role using the Configuration Manager console.

The SUP site system role must be created on a server that has Windows Server Update Services (WSUS) 3.0 SP2 installed. The SUP interacts with the WSUS services to configure update settings and to request synchronization to the upstream update source. It also interacts with the central site to synchronize software updates from the WSUS database to the site server database.

Beginning with System Center 2012 Configuration Manager SP1, you can have multiple software update points in your Configuration Manager environment to support clients in an untrusted forest. In addition, if you configure multiple SUPs at a site and one fails or becomes unavailable, clients will switch to another SUP. This behavior is called software update point switching or failover. We will discuss more about SUP switching and troubleshooting process related to switching later in the section.

Troubleshooting installation of software update points

When you add a SUP as a site system role, the sitecomp.log file shows that the SMS_WSUS_CONTROL_MANAGER has been flagged for installation (see Figure 2-6). It also shows the installation process for the SUP on the server. If the installation of the role fails for any reason, you'll find detailed information in the sitecomp.log.

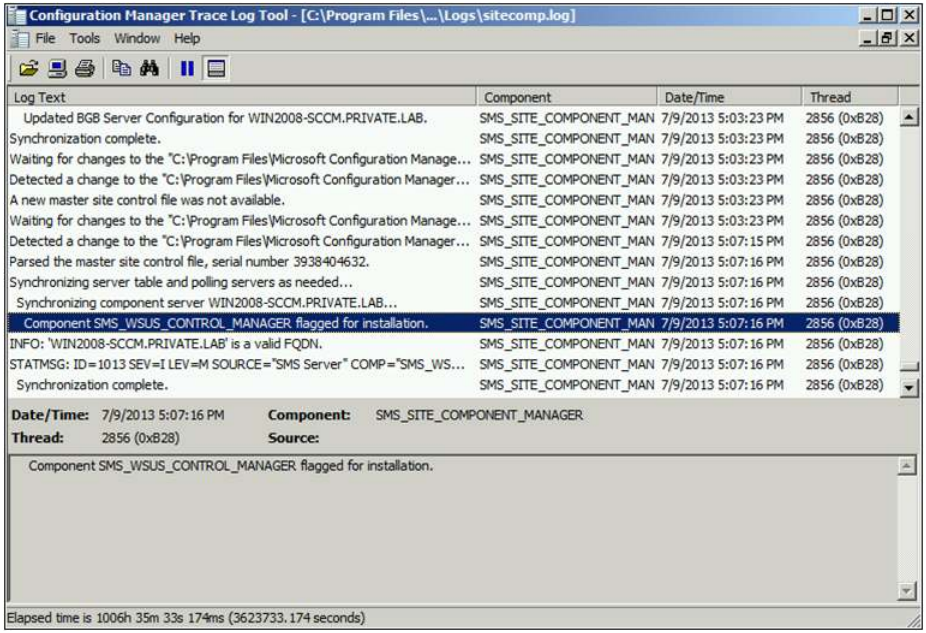


FIGURE 2-6 Software update point installation can be seen in sitecomp.log

Certain prerequisites must be met before installing a SUP. For example, you need to install Windows Server Update Services (WSUS) 3.0 SP2 with KB2734608 on Windows Server 2008 in System Center 2012 Configuration Manager SP1. When you go through the WSUS installation and reach the configuration part of the WSUS, you want to cancel and exit at that point. You should not configure WSUS because the software update point will take over WSUS after it is installed. Once WSUS 3.0 SP2 plus KB 2734608 are installed, you can start installation of the software update point and configure it the way you want with categories and products. You can also review the SUPsetup.log, which provides additional details on the software update point installation process.

If you run into problems, make sure the following items have been implemented correctly:

- The port settings configured for the active SUP must be the same as the port settings configured for the WSUS website in Internet Information Services (IIS) (that is, port 8530).
- The computer and local Administrator accounts must be able to access virtual directories under the WSUS website in IIS from the site server.

To sum up, you should review the following two logs when troubleshooting SUP installation:

- Sitecomp.log
- SUPsetup.log

Synchronizing software update points with Microsoft Update

In a Configuration Manager environment, the first step in deploying software updates to systems is to configure the SUP. From the central administration site, there are three ways to sync with a SUP (see Figure 2-7):

- **Synchronize From Microsoft Update** This option synchronizes updates directly from the Microsoft Update.
- **Synchronize From An Upstream Data Source Location (URL)** This is a new feature in Configuration Manager.
- **Do Not Synchronize From Microsoft Update Or Upstream Data Source** This option can be used to synchronize manually when the central administration site does not have access to the Internet.

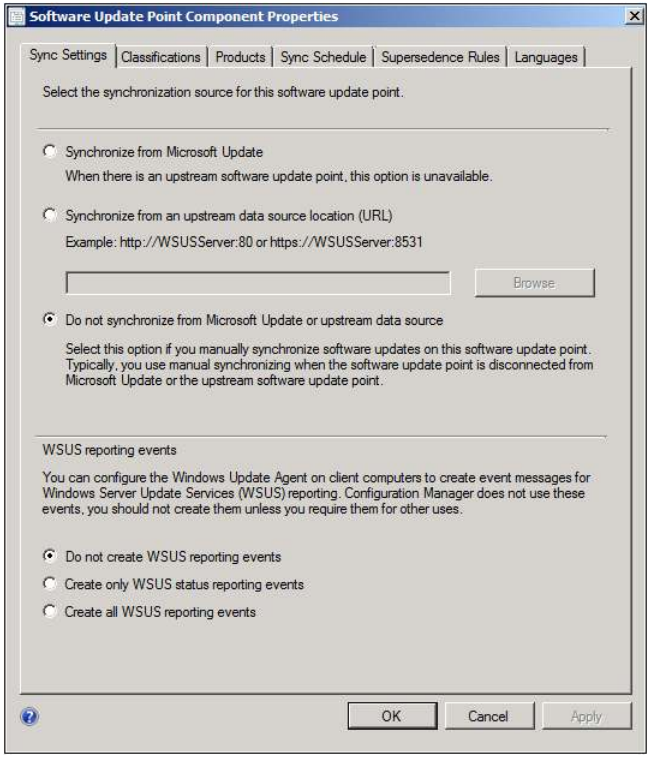


FIGURE 2-7 There are a few synchronization options for a software update point.

Troubleshooting synchronization with Microsoft Update

When you configure your software update point to synchronize with Microsoft Update, you can monitor or troubleshoot any issues by using the following logs:

- **WsynMgr.log** The wsyncmgr.log is located on the site server in the <ConfigMgrInstallationPath>\Logs folder. When there are any issues with synchronizations, it will be logged here.
- **WCM.log** The WCM.log file is located on the site server in the <ConfigMgrInstallationPath>\Logs folder. WSUS Configuration Manager connects to WSUS running on the active SUP once every hour. If there are any issues with ports or connectivity, it will log the errors.
- **WSUSCtrl.log** The WSUSCtrl.log file is located on the site server in the <ConfigMgrInstallationPath>\Logs folder. Where there are configuration or database connectivity issues, they will be logged in this log file.

For example, you might see the errors shown in Figure 2-8 if the minimum requirement of WSUS are not detected (that is, WSUS 3.0 SP2 with KB2734608) or when the port configuration is incorrect (that is, port 80 compared to 8530).

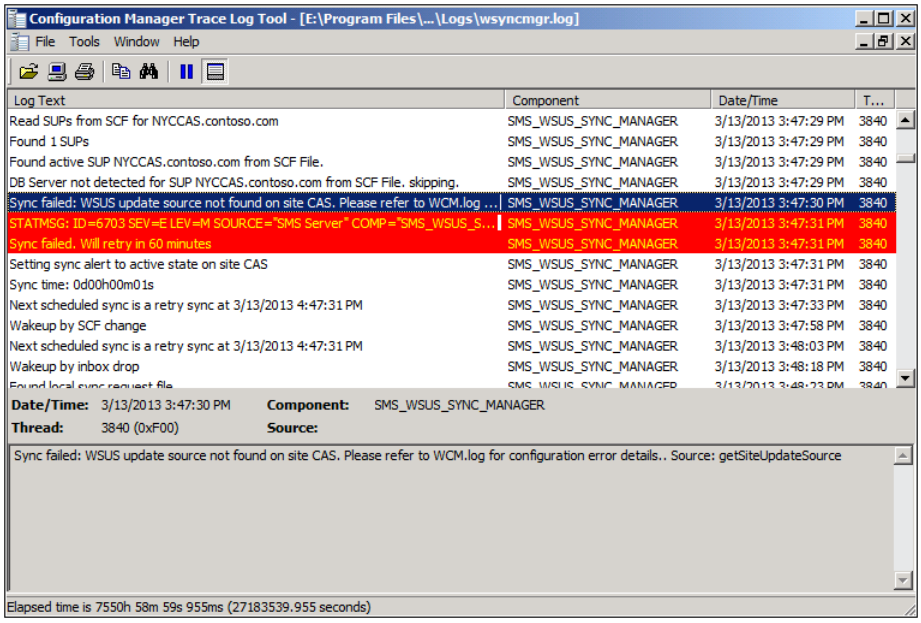


FIGURE 2-8 A synchronization failed error is displayed in wsyncmgr.log.

In the case of port misconfiguration, the wsyncmgr.log will report a "WSUS server not configured" message as shown in Figure 2-9. In this case, the question arises: How do you find out which port WSUS is trying to use? To find out, you should review the WCM.log file for an entry that says "Attempting connection to WSUS Server: <SiteServerName, port: <portnumber>, useSSL:<True or False>." As an example of this, if *portnumber* is listed as 80 and you configured WSUS to use the custom port 8530, then you would run into this issue.

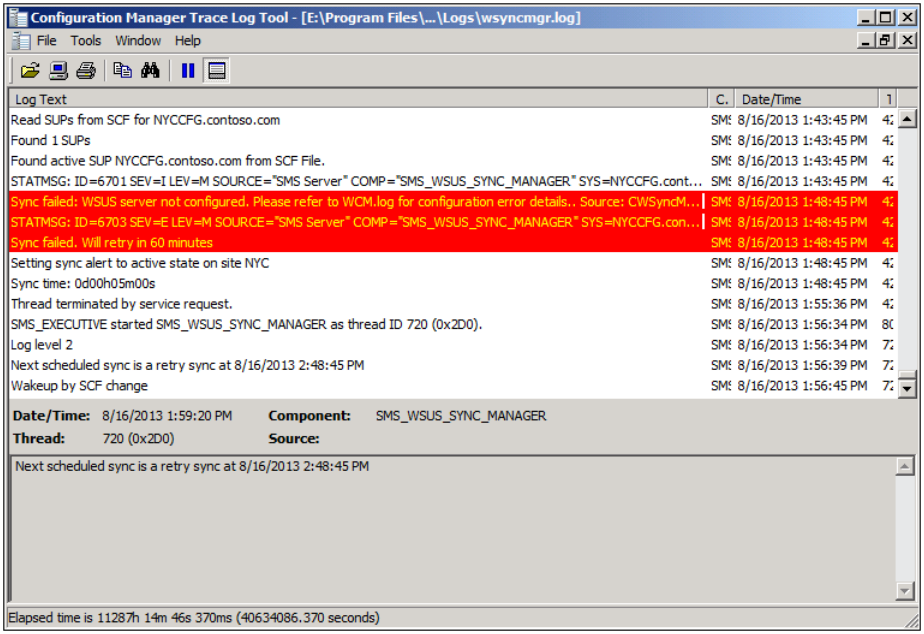


FIGURE 2-9 A synchronization failed error is displayed in wsyncmgr.log.

Now let's say you check the IIS configuration for WSUS and you determine that it is using port 8530 as shown in Figure 2-10. If this is the case, you would also want to check the SUP properties to make sure WSUS is configured to use port 8530 and not port 80.

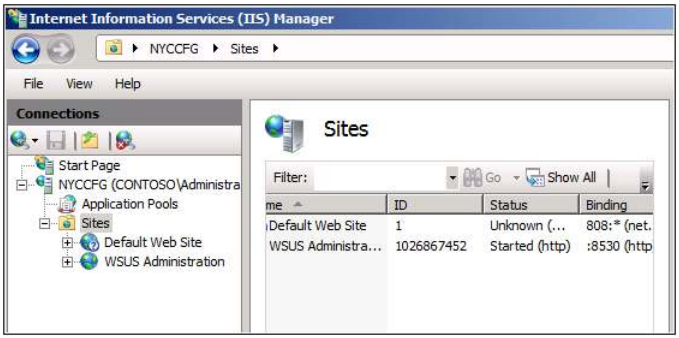


FIGURE 2-10 An example of WSUS port configuration in IIS.

If you have determined that both configurations are set to use port 8530 (or only the SUP is configured to use 8530 but not WSUS in IIS), you might want to run the following command to make sure WSUS is actually using port 8530.

```
C:\Program Files\Update Services\Tools\wsusutil.exe usecustomwebsite true
```

IMPORTANT **Sometimes, even though WSUS is shown as using 8530 in IIS Manager, it is not really using 8530 because the initial default setting (that is, port 80) is still being used.**

For the sync process to work properly, the SUP and the WSUS server must be able to communicate using the correct port (that is, 80 or 8530). A frequent experience of Microsoft Support is that even after fixing this issue (that is, port configuration, connectivity, or WSUS installation), synchronization fails at the first retry but succeeds at subsequent reties, so you might want to wait until the next retry before spending more time in troubleshooting the problem. Note also that the initial synchronization process normally takes longer than any subsequent synchronization.

NOTE **The client side workflow for software updates is covered in Chapter 3.**

Troubleshooting rotating management point and SUP failover

System Center 2012 Configuration Manager SP1 introduces several new features to support clients in an untrusted forest:

- Allowing multiple management points (MPs) so that a management point in an untrusted forest can support clients in an untrusted forest.
- Allowing multiple software update points so that a software update point in an untrusted forest can support clients in an untrusted forest.

This section examines some of the issues you need be aware of when deploying an MP or SUP in a remote forest.

Management point rotating behavior

If you have multiple MPs assigned to your primary site, clients can pick either one. However, there are number of factors involve in this process:

1. If one of the MPs is set up as HTTPS with PKI and a client has the proper PKI certificate, MP with HTTPS will be the first preference and will use that over other HTTP MPs.
2. The next preference is forest affinity for domain-joined clients where the MPs have published their information to Active Directory. If clients are in the same forest as the MP, the client would prefer that MP over other HTTP MPs assigned to the same primary site.
3. Now assume the real world scenario where you have the clients in different forests (that is, Forest C) where there is no MP but there are MPs in other forests (that is, Forest A and B). In this case, you have to be very careful when setting up remote MPs as if there is a firewall between the forests and clients in the forest that are not allowed

to communicate with any other MP but one particular forest (that is, Forest B). There is no way to guarantee that during rotating behavior a client will pick the MP from the forest (that is, Forest B) that the client is allowed to communicate with. In this scenario, you either want to have MP in the Forest C or set up one of the MPs from other Forests as HTTPS with proper PKI certificates. If this is not an option, an unsupported method that might work is using a host file to point to the specific MP.

Software update point switching/failover behavior

Multiple SUPs provide fault tolerance through failover. The way SUP failover works is that a list of SUPs is given to the client and the client chooses one from that list randomly. If the SUP it chooses cannot be reached, the client retries a minimum of four times at 30 minute intervals and after the fourth failure, it waits an additional two minutes and then tries to connect to the next SUP on the list. However, this failover behavior depends on the retry error codes received by the client when the scan fails. The WSUS component has a list of retry error codes and if the client gets the error code which is not part of this list, it will not failover to different SUP. You can find additional details around SUP switching behavior on Microsoft TechNet at *http://technet.microsoft.com/en-us/library/bcf8ed65-3bea-4bec-8bc5-22d9e54f5a6d*.

The list of error codes which would trigger a retry can be found using the following SQL query:

```
Select ID, SiteServerName, Name, Value2 as WSUSErrorCodes
from SC_Component_Property SCPROP
Join SC_SiteDefinition SCSITEDEF on SCSITEDEF.SiteNumber = SCProp.SiteNumber
where SCProp.Name = 'WSUS Scan Retry Error Codes' and SCSITEDEF.SiteCode = '<sitecode>'
'replace <sitecode> with your Primary site's site code
```

Review the list of error codes under the 'WSUSErrorCodes' column.

Application deployment troubleshooting

Application Management in Configuration Manager provides administrators with tools to manage applications in the enterprise. A new feature of Configuration Manager allows administrators to specify dependencies, supersedence, and other criteria within Application Management instead of creating different collections to deploy applications. This new functionality is much more robust than the old method of deploying legacy packages used in Configuration Manager 2007. This section examines the workflow and troubleshooting process for application deployment using Configuration Manager.

Enabling verbose logging

Before you begin troubleshooting for application deployment, always start by enabling verbose logging on the Configuration Manager client. Without verbose logging, many of the relevant log entries might not be recorded in the logs.

Client-side logging

By default, the client-side logging level is set to the value 1 in the registry. This means that Configuration Manager logs only Information, Warning, and Error messages. To enable verbose logging for those Configuration Manager logs that support it, do the following:

1. Open Registry Editor and find:

   HKLM\Software\Microsoft\CCM\Logging\@GLOBAL\LogLevel

2. Change the value of LogLevel from 1 to 0 (note that you will need to change the permissions for Administrators to have Full Control in order to change this value).
3. Restart the SMS Agent Host service.

Client-side debug logging

For even greater detail, you can enable debug logging. To enable debug logging for Configuration Manager logs, do the following:

1. Open Registry Editor and find:

   HKLM\Software\Microsoft\CC\Logging

2. Create a new key called DebugLogging.
3. Create a new value of type REG_SZ under this key and name it Enabled.
4. Set the Enabled value to True.
5. Restart the SMS Agent Host service.

Troubleshooting application deployment

Once you enable verbose logging on the client computer, you can use the additional information the log files provide to help you troubleshoot application deployment problems. The following walkthrough demonstrates how you might do this with an example of a problem deploying Microsoft RichCopy 4.0 using Configuration Manager. Here are the steps you might follow in troubleshooting this issue:

1. Open the Configuration Manager console, select the Software Library workspace, select Applications, right-click the Name column, and add the column CI Unique ID.
2. Write down the entry shown for ScopeID_*xxxxx*/Application_*xxxx* for the problem application as shown in Figure 2-11.

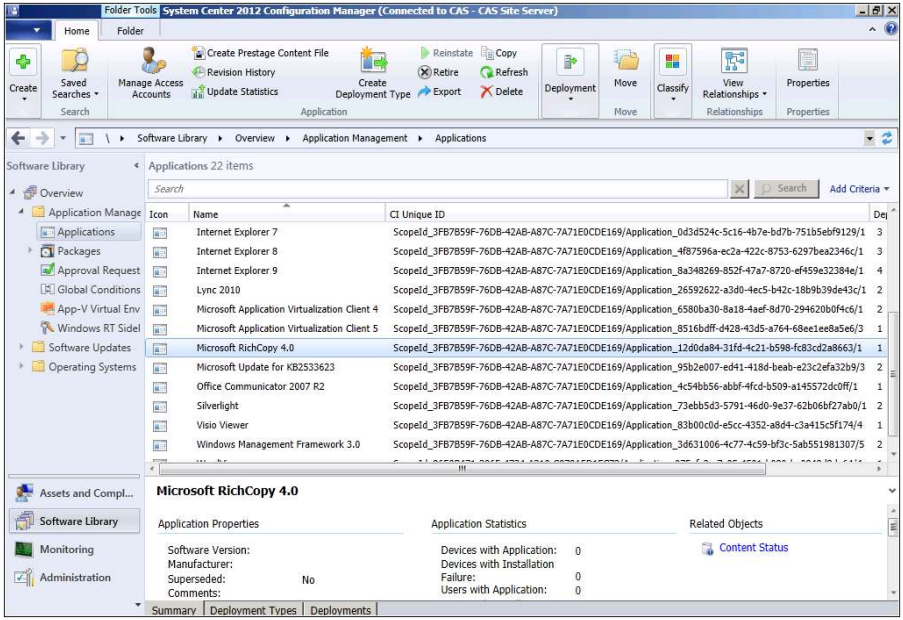


FIGURE 2-11 Determining the application CI unique ID from the console.

3. Use either Microsoft SQL Server Management Studio or the Configuration Manager console to get the Deployment ID. For example, using SQL Server Management Studio, you would connect to the database and run the following query to retrieve Assignment_UniqueID:

```
Select * from dbo.v_CIAssignment where AssignmentName like '%<name of the application>%'
```

In the example here, this would be:

```
Select * from dbo.v_CIAssignment where AssignmentName like '%RichCopy%'
```

Figure 2-12 shows the results of running the above query:

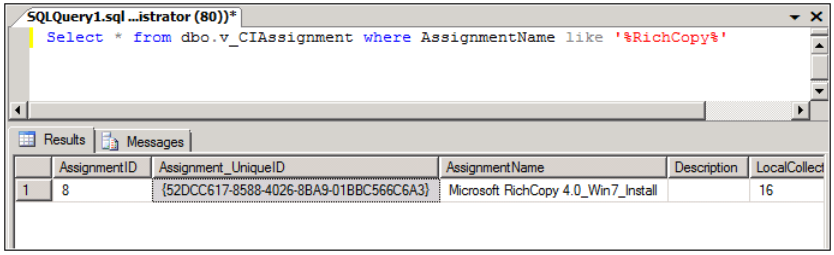


FIGURE 2-12 Determining the PolicyID using SQL Server Management Studio.

4. Write down the Assignment_UniqueID, which in the example here is:

```
{52DCC617-8588-4026-8BA9-01BBC566C6A3}
```

5. With Applications still selected in the Configuration Manager console, add the column Deployment ID. Then highlight the application (Microsoft RichCopy 4.0) and in the lower portion of the window, click Deployments and add a Deployment ID column to get the Deployment ID for the application as shown in Figure 2-13.

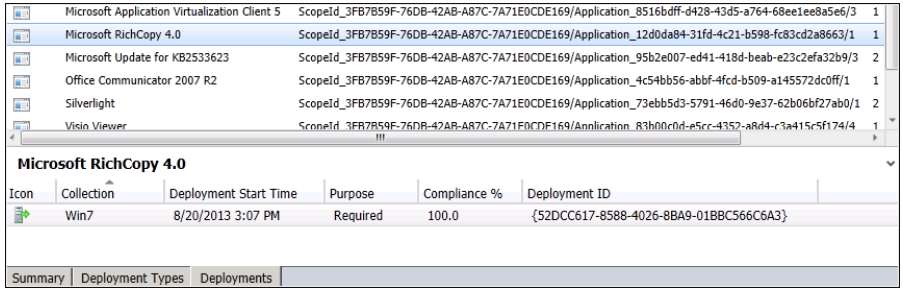


FIGURE 2-13 The Console displays the Deployment ID.

The Assignment_UniqueID and Deployment ID are the same as PolicyID for this particular application (RichCopy). This means that you can use either of these IDs to track the policy on the system using PolicyAgent and other components as shown in later steps in this example. You can also use PolicySpy tool to get the PolicyID as shown in Figure 2-14.

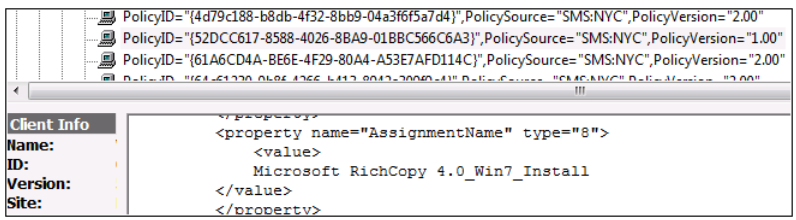


FIGURE 2-14 Determining the PolicyID using PolicySpy.

6. Open the PolicyAgent.log in the CM Trace Tool and select the policy ID determined in Figure 2-13. Look for the following entries in the PolicyAgent.log:

```
Compiling Policy '{<policyID>}…
Initializing download of policy…
```

See Figure 2-15 for an example of what these entries might look like.

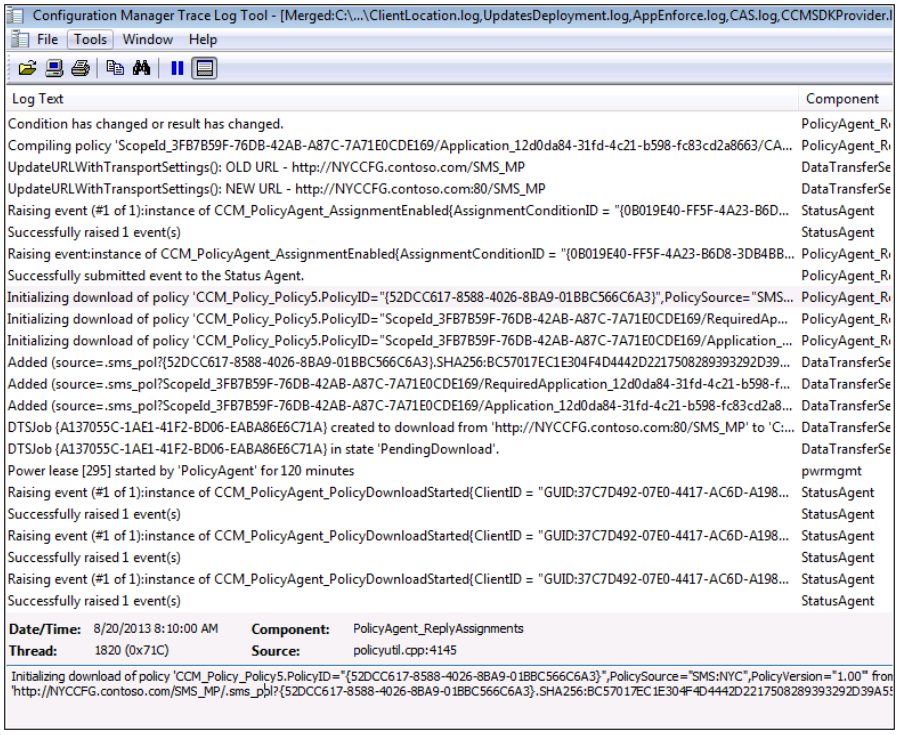


FIGURE 2-15 The initializing download of the policy is displayed.

7. During the application deployment process, PolicyAgent will hand over the task to the DataTransferService component and it will create DTSJob to download the policy. So your next task is to determine the DTS Job ID. In this example, you would search for the phrase "Download of policy CCM_Policy" in the PolicyAgent.log. Once you've found this log entry, you can determine the DTS Job ID as shown in Figure 2-16.

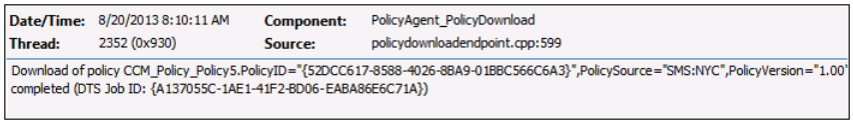


FIGURE 2-16 You can monitor the DTSJob download process.

8. Now you can trace the DTS Job (that is, A137055C-41F2-BD06-EABA86E6C71A) in the DataTransferService.log and you can also find out which MP it is downloading from (this is important if you have more than one MP per site). If there are any issues with that particular MP, you might want to first fix that MP because all of the clients using that MP might also fail to download any policies (see Figure 2-17).

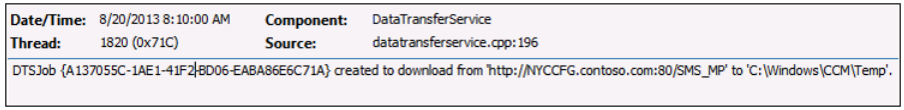


FIGURE 2-17 You can track the DTSJob ID.

Once download of the policy has successfully completed, PolicyAgent will hand over the task of updating and applying the policy to the system to the Policy Evaluator component as shown in Figure 2-18.

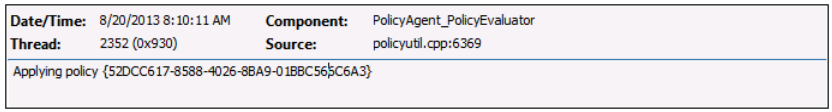


FIGURE 2-18 The policy is being applied.

The Policy Evaluator will also compile and apply the deployment associated with the policy as shown in Figure 2-19.

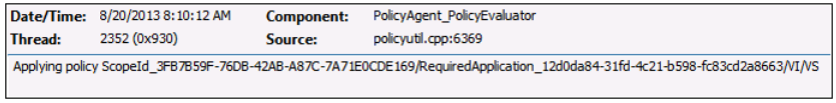


FIGURE 2-19 You can apply policy trace for deployment in PolicyEvaluator log.

9. The PolicyEvaluator component will then update WMI (root\ccm\policy\machine\actualconfig) and you can then either use PolicySpy (under the Actual tab) or directly connect to the WMI namespace and review CCM_ApplicationCIAssignment and CCM_CIVersionInfo using the appropriate AssignmentID and ScopeId. The Schedule component will then take over and initialize the trigger for the deployment as shown in Figure 2-20. There are also a few other components, such as CIStore, CIStateStore, CIDownloader, CITaskMgr, and DCMAgent, that work together throughout this process, but we won't go into details for those components because we are focusing on the ones that are useful when troubleshooting.

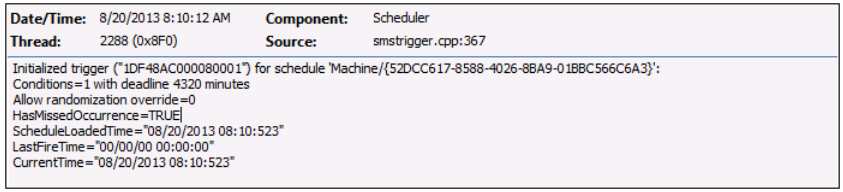


FIGURE 2-20 An example of tracking a deadline trigger with Scheduler.

You can also see in the CIStateStore logs that the CIStateStore component is querying something using an SQL query as shown in Figure 2-21.

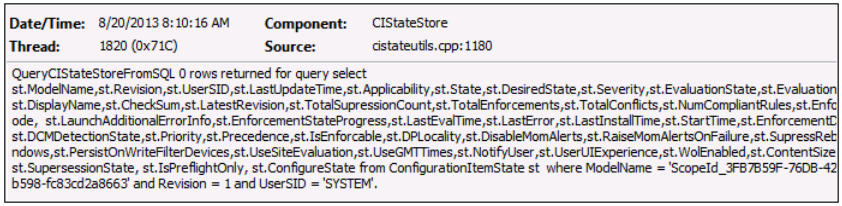


FIGURE 2-21 A view of CIStateStore log and the SQL query.

As you can see, a SQL query is being run that uses the following:

```
ModelName = 'ScopeId_3FB7xxxxx/Application_12d0da84xxxx'.
```

The ScopeID indicates that this is the same application that we are tracking (RichCopy). Since the Configuration Manager Client doesn't access the primary site database directly, the question is: Which SQL database does the CIStateStore component issue its query against? The answer is that the Configuration Manager Client has the Microsoft SQL Compact edition file located under the CCM folder (that is, CIStateDB, CIStoreDB, etc.) and it is running the query against that file, trying to determine if it has a configuration item (CI) related to the application downloaded locally. Notice in Figure 2-21 that it returned 0 rows initially.

Next, the CIAgent will go through all of the dependencies of the policy CI and will work with CIAgent, CIStateStore, and CIDownloader to download them and run the SQL query again. It will then hand over the task to the AppDiscovery component to find out if this application exists (see Figure 2-22).

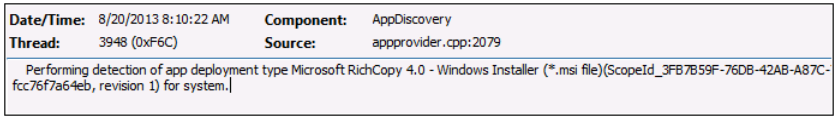


FIGURE 2-22 The AppDiscovery determines if the application exists.

If the application is not detected, the AppDiscovery component will hand over the task to the AppIntentEval component to find out if there are any dependencies associated with the application deployment (see Figure 2-23). If there are any dependencies, it will download and install them first.

FIGURE 2-23 The AppIntentEval component identifies dependencies.

The ContentAccess component (CAS logs) will then request the content with ID Content_xxxxx together with its size and priority. The ContentTransferManager component then creates a CTM job with an ID for the download of the application to

the local Configuration Manager client cache. When you are reviewing these logs (CAS and ContentTransferManager), they will now indicate the name of the application. This raises the question of how to determine which content is being requesting for download. You'll determine this next.

10. In the Configuration Manager console, select the Software Library workspace | Application Manager | Applications. Select the application (RichCopy) and in the lower portion of the window, click the Deployment Types tab to determine Content ID (that is, Content_e900e6c0-b55c-496d-b210-0d53de88c3e3) as shown in Figure 2-24.

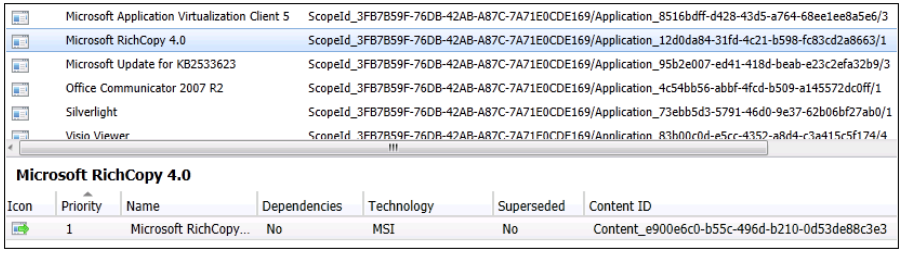


FIGURE 2-24 A display of the Content ID from the console.

11. When you review the ContentTransferManager.log, notice that the CTM job is starting (see Figure 2-25).

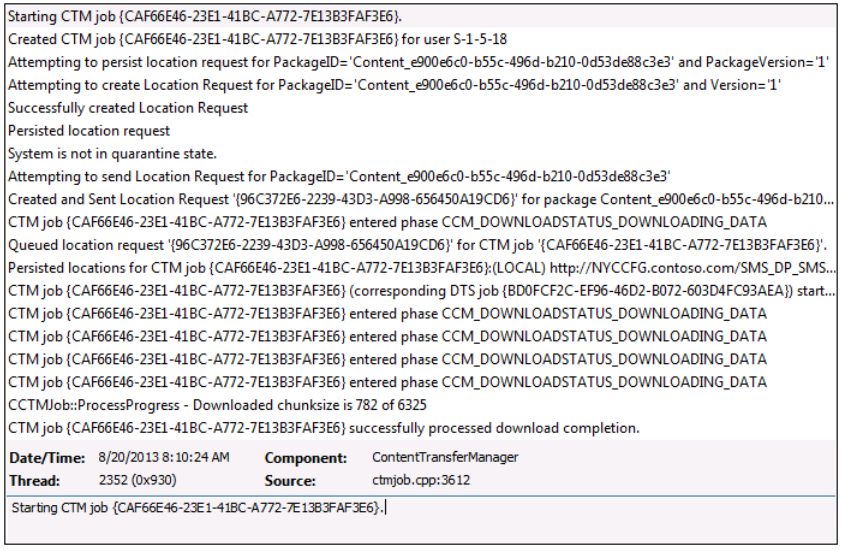


FIGURE 2-25 The ContentTransferManager.log shows that the CTM job is starting.

12. The question now arises: How can you determine which CTM job is being referred to by the log entry shown in Figure 2-25? The answer is that you need to find that CTM job in the CAS.log as shown in Figure 2-26.

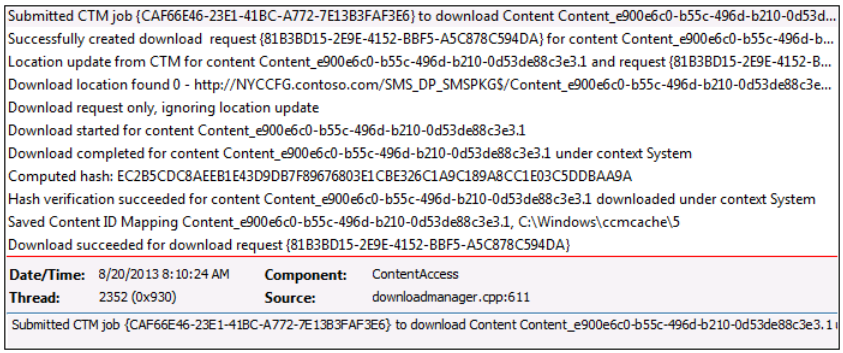


FIGURE 2-26 The Cas.log shows that the CTM job has been submitted.

13. The ContentAccess component will also create a download request with a completely different ID for the same content ID. You can track that download request in the CAS.log to make sure the content was successfully downloaded as shown in Figure 2-27. You can then follow it through ContentTransferManager.log, CAS.log and DataTransferService.log which will provide all the details around downloading the contents, hash verification, and cache location.

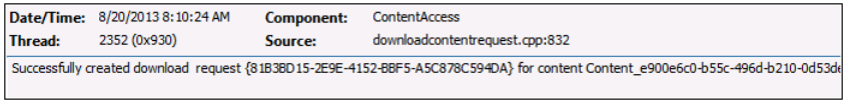


FIGURE 2-27 The download request has been successfully created.

14. You can also monitor the SCClient component by using the. _SCNotify_<username>. log which provides information around displaying notification balloons with other details such as downloading and installing software. Once the download completes, ServiceWindowManager will check to see if there are any maintenance windows specified for the system. If there are no maintenance windows specified for the system, the AppEnforce component will begin installation of the application as shown in Figure 2-28. It will also display the actual command line being executed on the client as well as the exit-code.

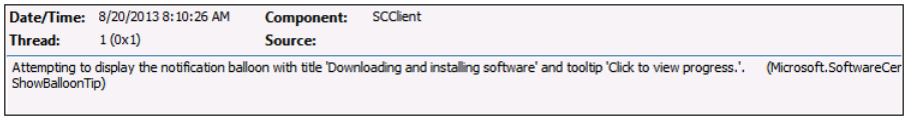


FIGURE 2-28 The client is attempting to display a notification balloon.

IMPORTANT **Please note that the application installation activity is no longer logged in execmgr.log. If you use legacy software distribution process, the installation activity is logged in execmgr.log.**

15. Finally, at the end of the installation, the AppEnforce component will perform the check again to see if application has been detected on the system (see Figure 2-29).

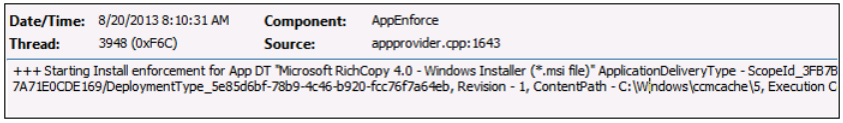


FIGURE 2-29 An example of application installation tracking.

So here you have it: the end-to-end process of troubleshooting application deployment using Configuration Manager. The diagram in Figure 2-30 shows the overall application deployment process and how the various Configuration Manager components work together. The diagram does not include all of the components involved in the process, just the main ones that are useful for troubleshooting application deployment issues.

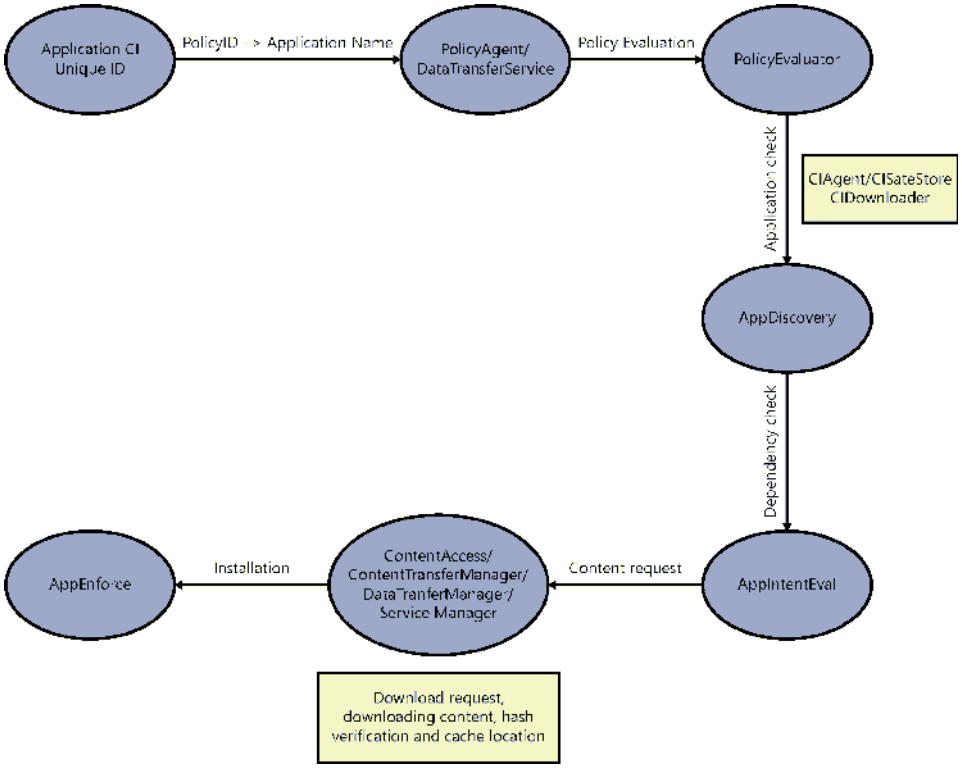


FIGURE 2-30 The application deployment process contains various components.