Global Knowledge®

Expert Reference Series of White Papers

Getting the Most Out of Your Tivoli Endpoint Manager Deployment

Getting the Most Out of Your Tivoli Endpoint Manager Deployment

David Biondi, Global Knowledge Instructor and IBM Advanced Deployment Professional

Introduction

IBM Tivoli Endpoint Manager (ITEM) is a product designed to let enterprises automatically manage computers, allowing thousands of machines to be managed by just a few support staff. With ITEM, tasks such as patch application, software distribution, and security policy enforcement can be performed on all of an organization's machines with minimal supervision.

Ideally, ITEM allows the entire lifecycle of a workstation to run its course with little to no individual attention from IT staff. From the moment the machine is first placed on the network to the moment it is last removed from the network, ITEM can take care of all of the management tasks needed to maintain the software and security of that machine.

To achieve this level of automation, you will probably have to tweak your ITEM installation a bit from its out-of-the-box settings. There are several ways to get the most out of your Tivoli Endpoint Manager Deployment including:

1. Creating custom settings for automatic groups
2. Using bandwidth throttling
3. Designating relay affiliation and custom relay settings
4. Pre-caching data for more efficient data transfers
5. Applying policy or persistent baselines

This paper will cover each of these options along with screenshots and step-by-step instructions on how to increase the level of ITEM performance in almost any setting.

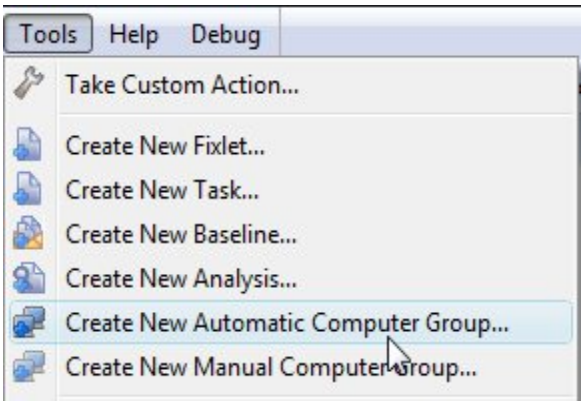
Make the Most of ITEM by Using Custom Settings for Automatic Groups

When using ITEM, machines can be grouped using various criteria, including factors such as which operating system the machines are running or to which department the machine belongs. When designating groups, ITEM allows for two types of grouping methods: manual and automatic. Manual grouping requires a human operator to make a static assignment of machines to a particular group. Once machines are assigned manually to a group, they will stay in that group unless they are manually removed from it.

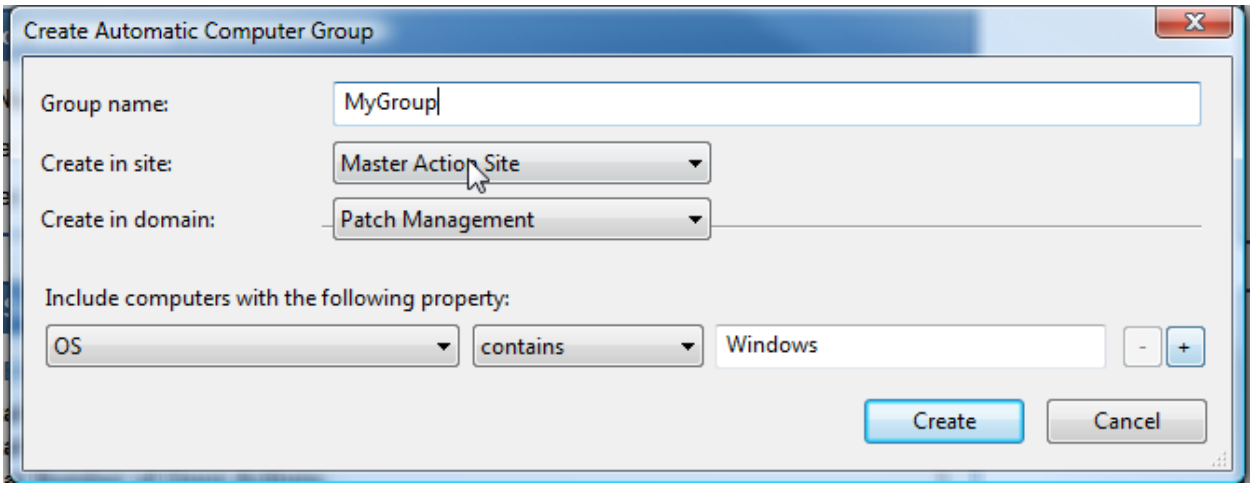
Automatic grouping allows ITEM to build a list of the computers in an organization and assign them to dynamic groups based on their attributes. Many organizations already use this ITEM automatic group function by allowing ITEM to group machines based on settings retrieved from the computers in the group. Right out of the box, ITEM is configured to easily group machines based on characteristics such as their operating system version or subnet. That's a great way to start using the power of automatic grouping, but it's only the tip of the iceberg.

If you aren't using these automated groups, here's how you can start:

From the Tools menu, select Create Automatic Computer Group.



Next, name your group and select the Site and Domain where it should be created. Finally, select the property and property value (hit the + button for multiple properties) and press the Create button.



Beyond the basic automatic groups, groups can also be built using the full range of relevance language. For example, you can build a group based upon the existence of a particular file on a computer, the value of a registry entry, or which user is logged into the machine.

While relevance queries allow us to query computers for things that they know about themselves, custom settings provide a way to teach individual machines things that are not intrinsic to the machine itself. For example, we can inform the computer about which department it belongs to, or whether it is part of a test environment (as opposed to a production environment). We can then use custom settings to create automatic groups based on these external factors.

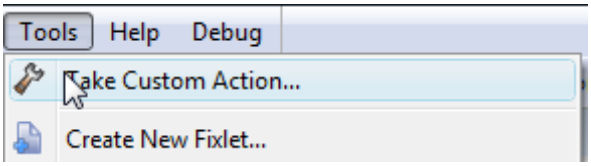
For instance, if we apply a custom setting about which environment the machine is in (e.g., test, QA, or production), machines that are part of a test environment can then be grouped separately from machines in a production environment.

Once the machines are in appropriate groups, these groups can be used by ITEM to control operator permissions. For example, custom settings can be used to ensure that an organization's development team can only deploy to servers defined to be in the development environment.

If we add a setting to define which department in an organization owns a particular device, we can use custom group settings to control or define which machines will be the targets of a particular distribution (e.g., making sure that machines in an organization's accounting department should receive a particular software distribution, and machines in the HR or sales departments should not).

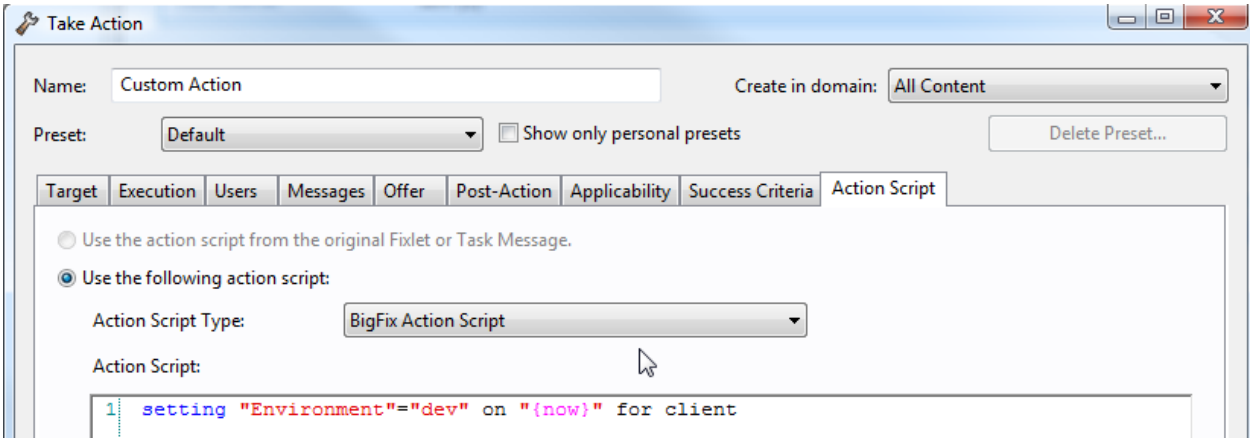
You can create a custom setting on your target machines by deploying a new action. Here's how to do it:

First, select Take Custom Action from the Tools menu.

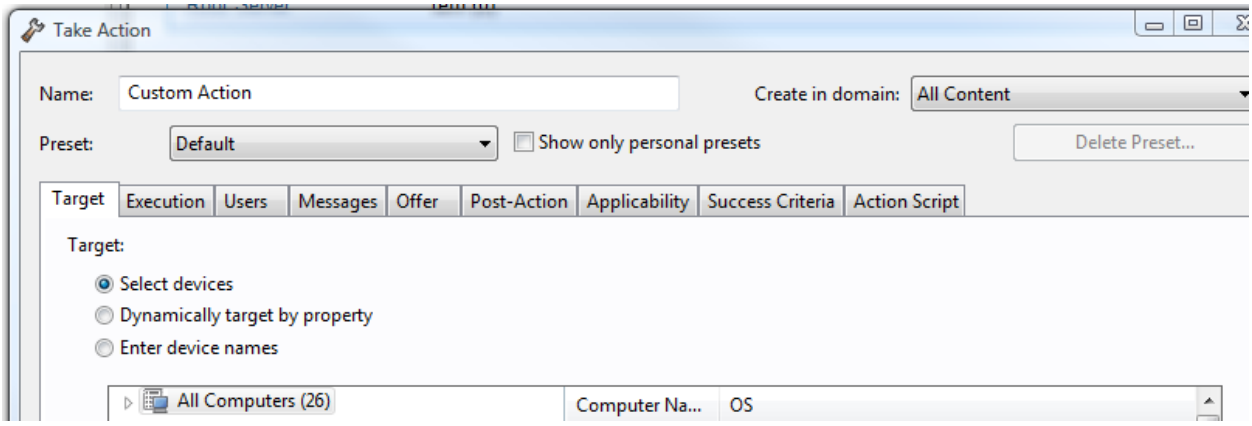


Then, select the Action Script tab and in the Action Script, enter:

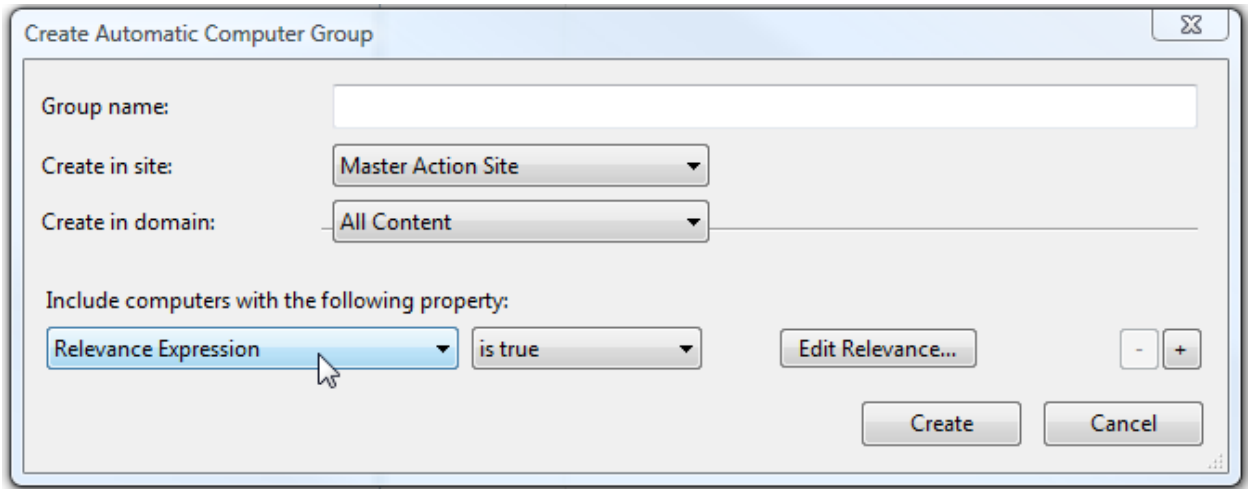
setting "SETTINGNAME"="SETTINGVALUE" on "{now}" for client



Then just select your targets and press OK.

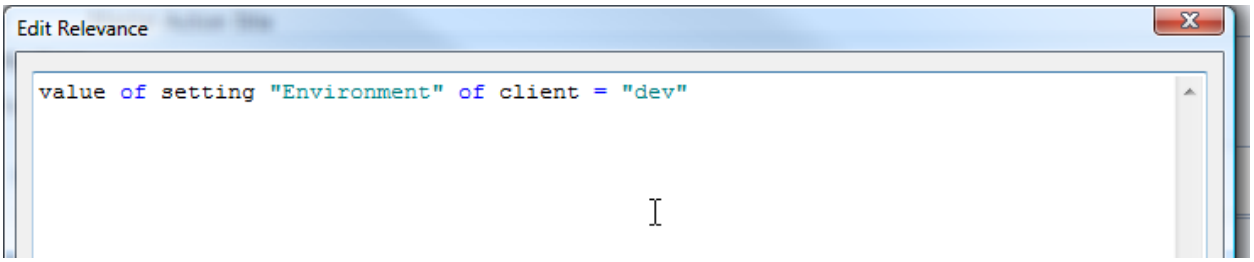


Once you have deployed your custom settings, you can create groups based on that setting. Just follow the instructions for creating an automatic group and select Relevance Expression from the property list. Then press the Edit Relevance button.



Then press the Edit Relevance button and enter the following Relevance query:

value of setting "SETTINGNAME" of client ="SETTINGVALUE"



You'll now have a custom automatic group with all of the computers with that custom client setting!

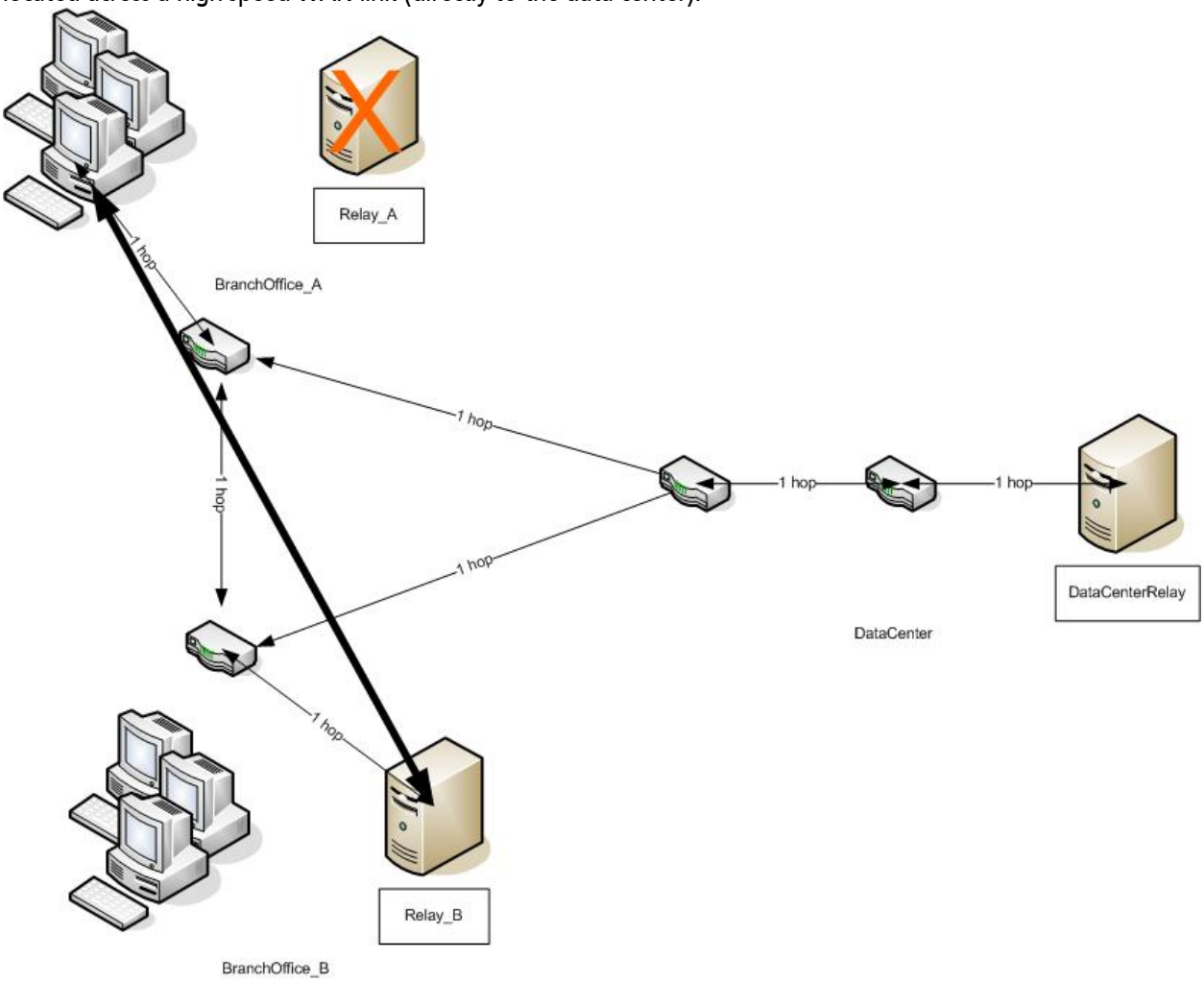
Using Relay Affiliation and Automatic Relay Selection Options

Optimizing relay affiliation allows for faster and more dependable communication over a network. ITEM gives client computers the option of having relays selected manually or enabling automatic relay selection. Manual relay selection is acceptable in a small environment that has very few relays, but automatic relay selection is preferable in larger environments.

Generally, we want a client computer to select the relay that is closest within the network. Automatic relay selection allows the client computers to determine the distance to the available releases and choose the closest one.

This concept can be taken one step further by using ITEM's relay affiliation feature, which enhances auto relay selection to ensure that the ITEM clients are communicating with the correct relays. For example, this feature would come into play if there were a number of clients connected to their local relay, and the relay failed. Those clients would need to find an appropriate backup relay. If those relays existed within the network, it would be a simple process for an ITEM endpoint to locate the closest relay.

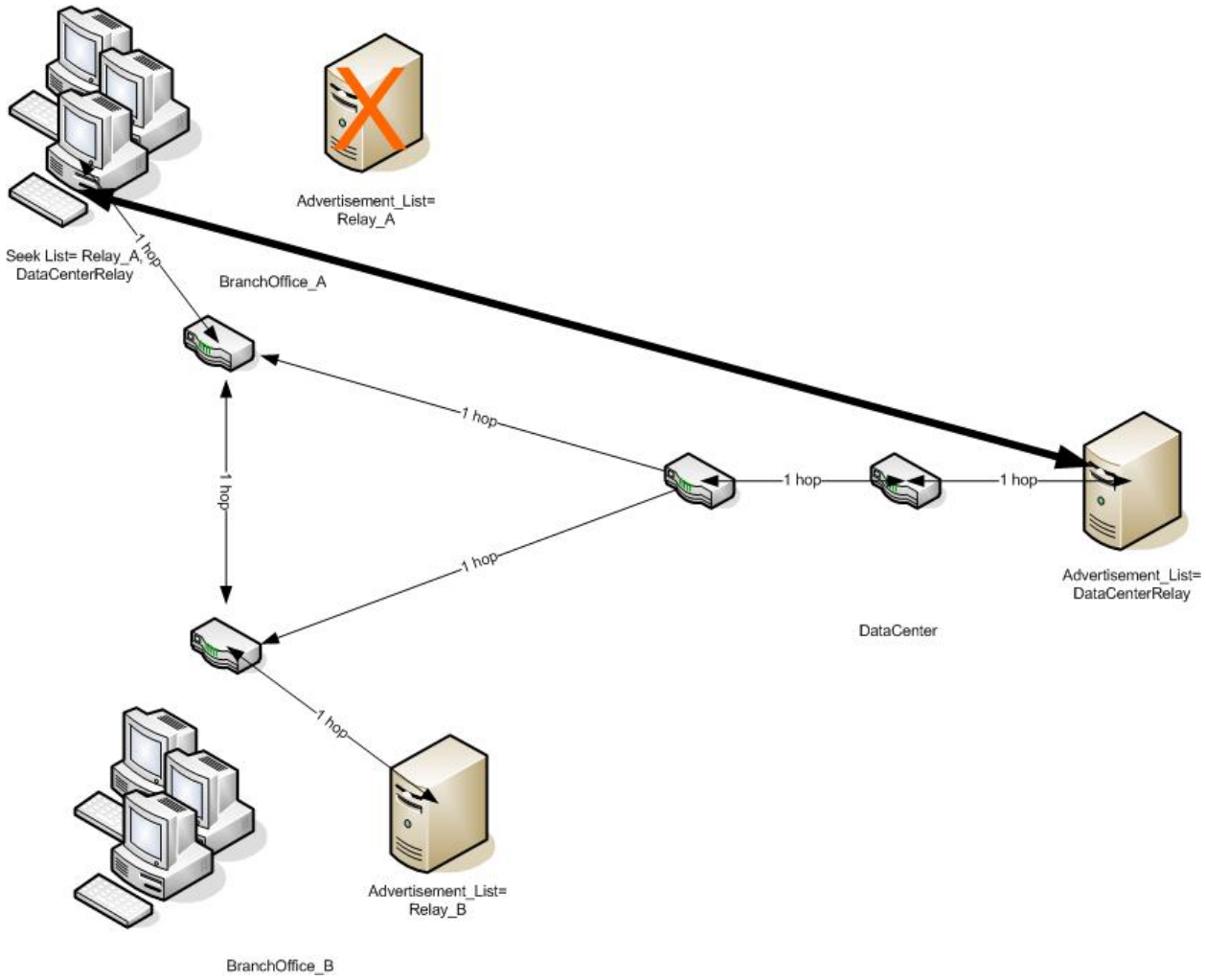
However, if there were a lack of local relays, clients would begin searching across WAN connections to find an available relay. Depending on the network topology, it is possible that the next closest relay would be across a slow connection (e.g., two interconnected branch offices) while a relay only one step further away would be located across a high-speed WAN link (directly to the data center).



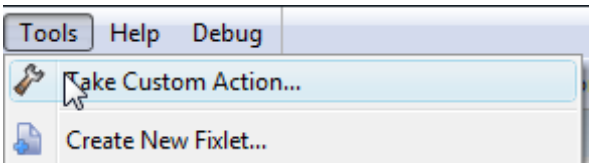
In the above example, when Relay_ A fails, the Relay_B at the second branch office is only two hops away from our endpoints whereas the DatacenterRelay is three hops away. Thus, even though we would prefer that they use the DataCenterRelay as their backup, they actually begin using Relay_B instead.

Relay affiliation can alleviate this problem by preventing clients from connecting to particular relays. In the above example, we could provide relay affiliation that would allow clients to connect to a relay at their own location or to the datacenter, but not to a relay located at a different branch office.

In the above example, each endpoint is provided a "Seek List" of relays they are allowed to communicate with and each relay presents an advertisement list that matches the seek list of the endpoints. Thus, when Relay_A fails, the endpoints seek out the DataCenterRelay as their backup instead of using Relay_B.



Relay Advertisement and Seek lists are set by configuring a setting on your managed computer. To set these, first select Take Custom Action from your Tools menu.

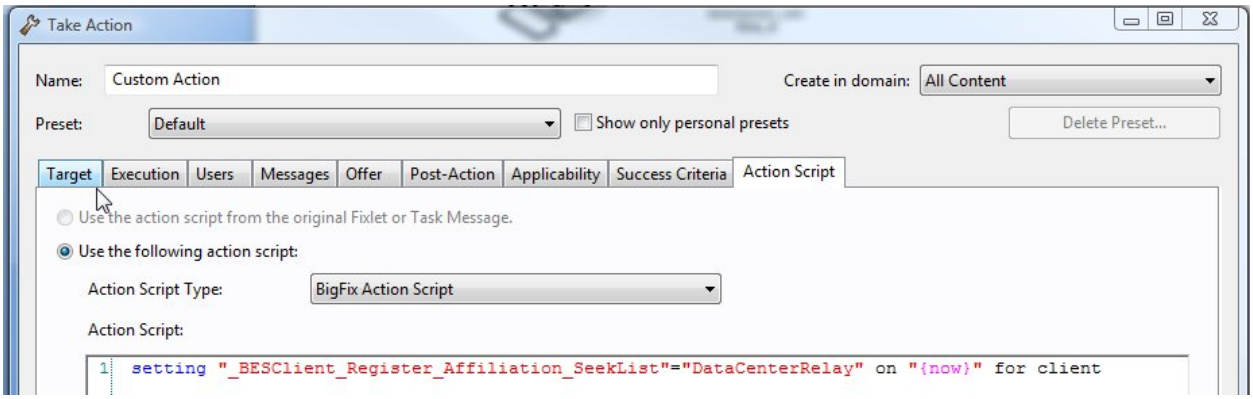


Then, select the Action Script tab. To set your Client Seek List, in the Action Script, enter:

setting "_BESClient_Register_Affiliation_SeekList"="SEEKLIST" on "{now}" for client

To set your Relay Advertisement List, in the Action Script, enter:

setting "_BESRelay_Register_Affiliation_AdvertisementList"="ADVERTISEMENT" on "{now}" for client



Another way that relay affiliation helps optimize a deployment is by limiting which machines an ITEM client can communicate with. For example, if a software distribution is performed to a group of clients, that distribution will be cached on the relays along the path of the distribution. If that distribution contains sensitive data, it would be best to ensure that all of the clients in that distribution use a select set of relays with increased security settings.

Relay affiliation can ensure that clients receiving this distribution are only able to connect to a subset of relays that are permitted to cache this sensitive data.

Take the Guesswork Out of Bandwidth Throttling

When a distribution to a client is in progress, you don't want the end user to experience poor or decreased performance on their workstation or computer. If a distribution is going to be performed when end users are likely to be using their workstations, the distribution should be designed to occupy only a fraction of their available bandwidth.

In a corporate setting where all the endpoints are connected directly to a high-speed LAN, determining how much bandwidth to occupy with distributions is usually relatively easy: generally the quantity of available bandwidth is known, so a maximum amount of bandwidth that ITEM is reasonably allowed to use can be hard coded.

Even in a situation where the amount of bandwidth is known, the percent of available bandwidth that a distribution should be allowed to take varies. It can be limited to a small amount of bandwidth, say 5% of the network, during peak times and can be allowed to use up to 100% of the available bandwidth during dedicated maintenance windows. Large distributions may occupy most of a client's available bandwidth for a significant amount of time. This should be considered when planning a distribution (this is where pre-caching can become very useful, so keep on reading!).

Figuring out how much bandwidth is available for distributions gets more complicated if the target devices are remote. For instance, remote laptops often connect across a variety of lower-speed networks with unknown or variable connectivity, yet still need to receive large distributions.

To distribute data to a device in that situation, we will need to determine how much bandwidth that device has on a moment-to-moment basis and scale the distribution based on how much of the network the distribution will occupy at the current bandwidth. Luckily, ITEM provides tools that enable us to make these determinations. ITEM's dynamic bandwidth throttling capabilities allow us to set a cap on what percentage of the network we will allow the distribution to occupy. Once our maximum is set, ITEM will automatically monitor the current bandwidth available to the target machine and will adjust the data transfer based on that number.

In addition to being able to set dynamic parameters for bandwidth use, ITEM also allows users to set static minimums and maximums. This ensures that the determinations ITEM makes are appropriate for a particular environment. For example, if a large amount of data needs to be sent over a network, setting a static minimum would allow the distribution to get done in a reasonable amount of time, even if the distribution will take up a large share of the available bandwidth.

To set Dynamic Throttling, you must first enable it using the BES Client Setting: Enable/Disable Dynamic Throttling Fixlet. You can find this fixlet in the BigFix Management domain with the Fixlet ID of 605.

BES Component Management

Name	ID	Source Severity	Site
BES Client Setting: Enable/Disable Dynamic Throttling	605	<Unspecified>	BES Support

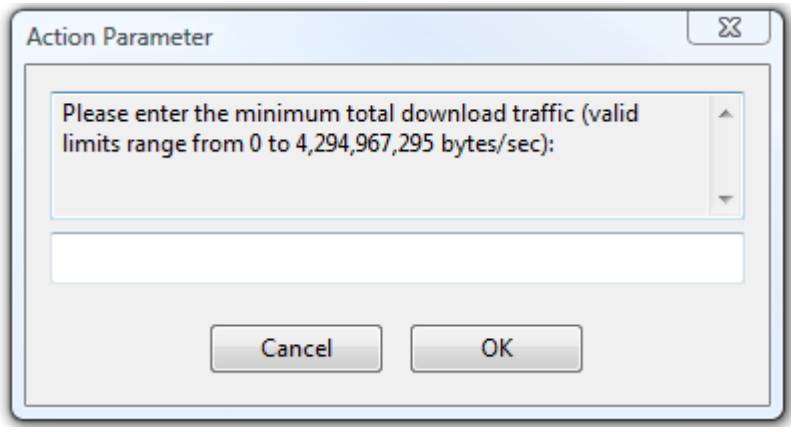
After Enabling Dynamic Throttling, you can set the specific Throttling parameters using the BES Client Setting:

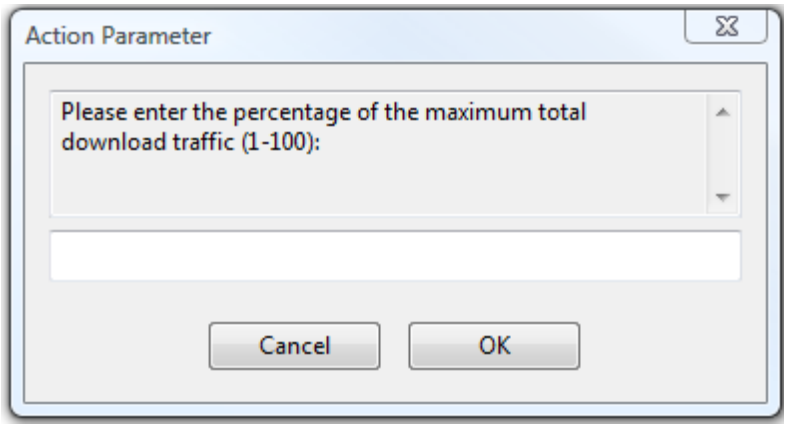
Dynamic Download Throttling Fixlet (ID 457).

BES Component Management

Name	ID	Source Severity	Site
BES Client Setting: Dynamic Download Throttling	457	<Unspecified>	BES Support

When you run this fixlet, you will be prompted for the Minimum and Maximum values as well as the percentage the network you wish to occupy.





Use the Magic of Pre-Caching

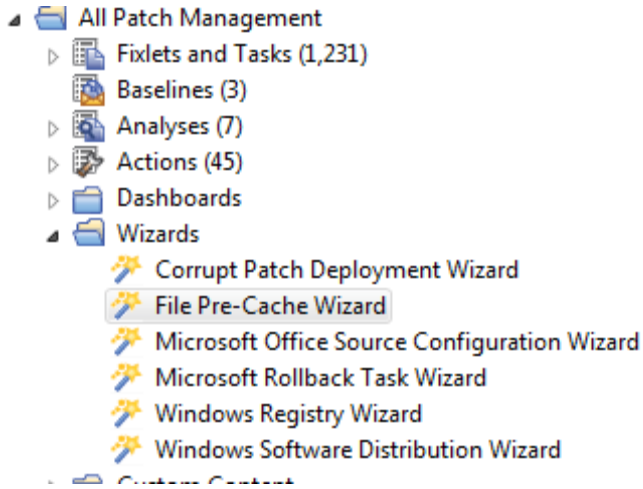
The time allotted for maintenance windows often seems short compared to the amount of work that needs to be performed during that window. One of the reasons for this is that the data transfer of large distributions may take hours to complete, often eating up much of the maintenance time. In this scenario, there may not be enough time for operators to complete the work that they were supposed to during their window. Or, if the operators do manage to complete the upgrades despite spending much of their allotted maintenance window waiting for a distribution to download, there still may not be sufficient time at the end of the window for any necessary troubleshooting.

Using ITEM's pre-caching function allows the data required to perform a distribution to be transferred separately from the execution of that same distribution. Thus, the files required are already available to the client when it is time to perform the update. This makes the best use of allotted maintenance time by ensuring that the maintenance window is used for performing the necessary upgrades instead of waiting for data to transfer.

There are two ways that data can be pre-cached using ITEM: relay pre-caching and endpoint pre-caching.

Relay Pre-Caching

Prior to the maintenance window, data is sent to and stored on the relays that are local to any target clients, but not on the target client itself. This will reduce the time it takes to deliver the necessary distribution to the target client during the maintenance window. Support for relay pre-caching is built directly into ITEM. You access relay pre-caching via the File Pre-Cache Wizard in the Patch Management Domain.



Across a slow link, the improvement gained from relay pre-caching can be dramatic. However, it is worth noting that if the relay is supporting a large number of clients, the LAN that supports the clients may still become congested during the delivery of a large pre-cached download.

Endpoint Pre-Caching

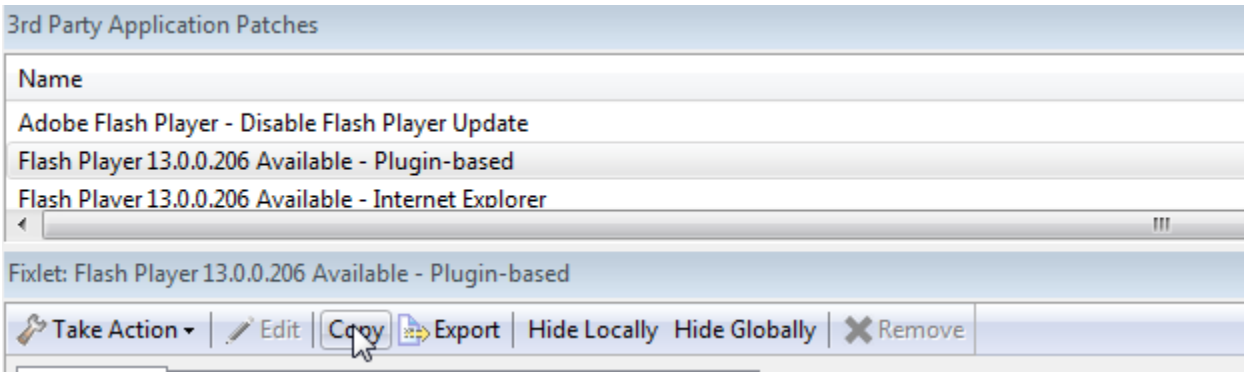
There is no out-of-the-box ITEM tool for pre-caching at a target client, so custom content is required. However, pre-caching at the target client allows for the quickest possible data transfer during a maintenance window, so it is worth investigating in situations where time is a factor.

Client pre-caching is accomplished using a two-stage distribution:

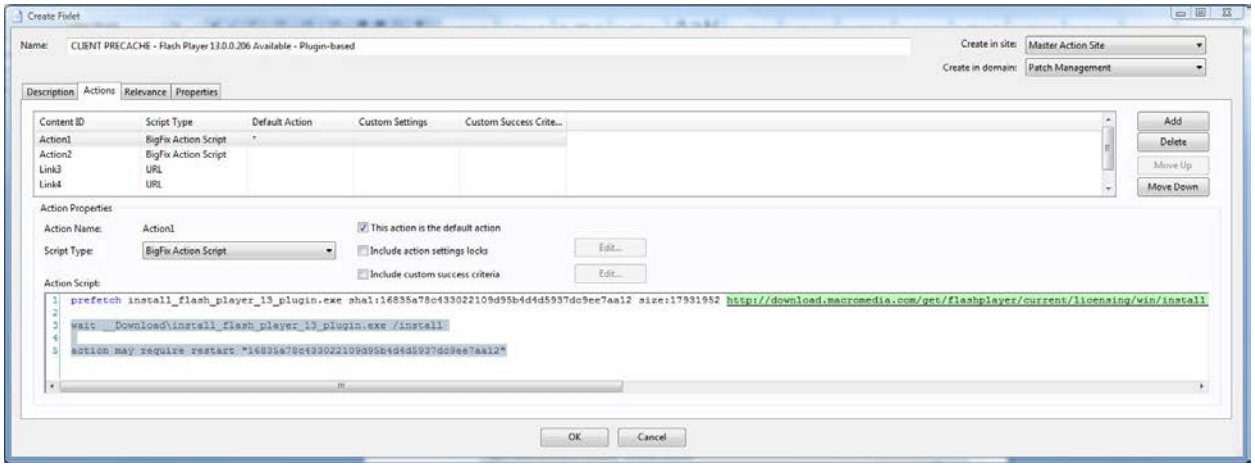
1. The first stage consists of a distribution that contains the files necessary for the update without any action script that would implement the update. This distribution can be performed well in advance of the maintenance window. Also, using the bandwidth throttling techniques discussed earlier can minimize impact to the network.

2. The second stage consists of a distribution that contains the entire update package. This distribution is typically sent at the start of the maintenance window. This distribution normally still contains the required files, but since they were distributed previously, it only needs to verify that the files exist in the client's cache not download them again.

The first step to performing client pre-caching is to select the patch that you wish to apply. Then, select "Copy" to create a custom copy of the fixlet.



Change the name of the fixlet (I prefix the name with CLIENT PRECACHE). Then, select the Action tab and remove everything after the "prefetch" command. Press OK to create a fixlet that will only distribute the files for this patch. Then, deploy the fixlet as you would any other and your data will be awaiting you on distribution day!



Harness the Power of Persistent Baselines

Persistent baselines, also known as policy baselines, allow ordered distributions of fixlets and tasks to be implemented any time a computer inside of ITEM requires those updates. Here are two examples of scenarios in which ITEM's persistent baseline function can be put to good use:

1. When updating gold images
2. When reapplying policy updates that have been undone by end users

Let's take a look at each of these scenarios.

Using Persistent Baselines for Gold Images

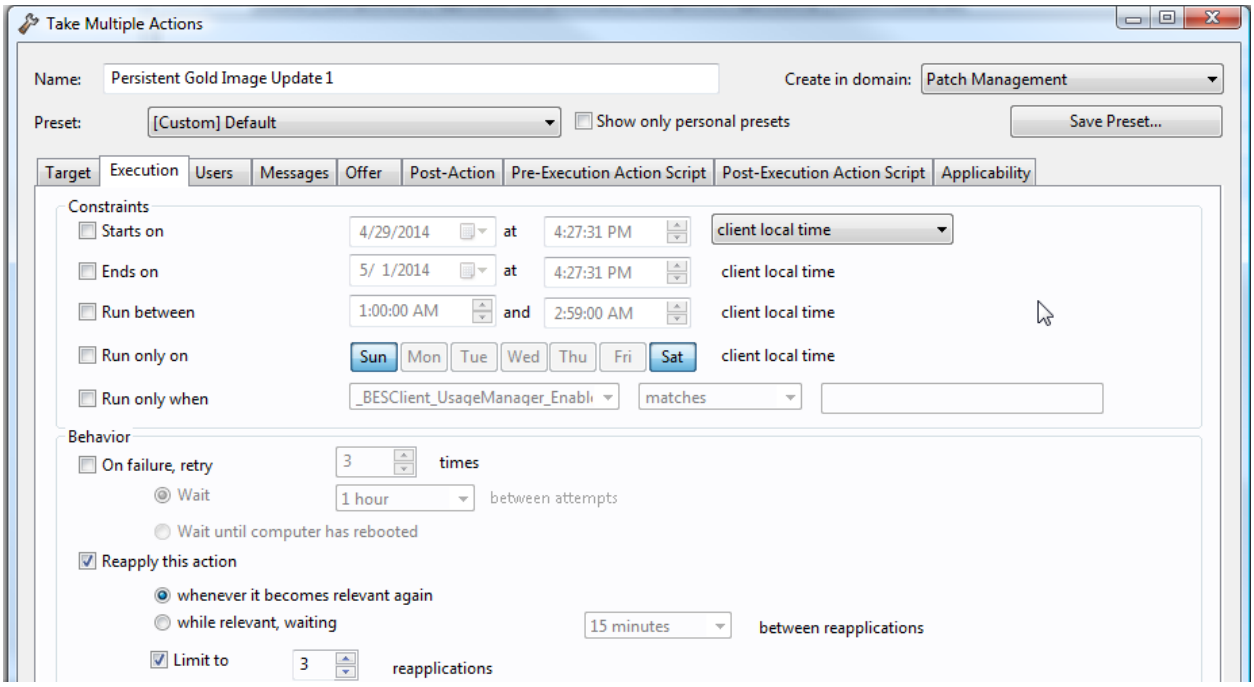
Gold images (a snapshot of a system that is used as an ideal or "gold" image for other machines to be modeled after) should be monitored to ensure that machines in an organization are all running at the current release level. Building new gold images can be time consuming for operators. Conversely, if attempting to save labor by purchasing machines with an organization's gold image pre-installed, deploying a new version of a gold image can be very expensive.

ITEM's persistent baseline function allows operators to select which patches are required to bring a new machine on a network up to the current release level. This ensures that when a new machine with the current gold image is deployed, the system is already upgraded to the current release levels. Using ITEM in this way saves labor by reducing the number of new gold images that an enterprise's release team has to build. It can also help organization's bottom line by reducing the amount of "set-up fees" paid to vendors to get them to change the preinstalled gold images on new machines.

Using Persistent Baselines to Reapply Policy Updates

ITEM's persistent baseline features can be used to reapply a policy update to a computer even if an end user has undone a previous distribution. For example, imagine a scenario where some end users are dissatisfied with the impact of a security patch on their workstations. Some of those users may uninstall that patch, compromising the security of their workstation. Persistent baselines can automatically reinforce the policy that the patch must be installed and would reinstall it, thus returning the computer to its protected state.

To create a persistent baseline, simply select the baseline or fixlet that you wish to deploy. However, when presented the Take Action dialog, select the Execution tab. Deselect the "Ends on" option and select the "Reapply this action" option. Then, press OK to begin using your new persistent baseline.



Conclusion

Among other topics, this paper has covered some ways that ITEM can help reduce operator workloads, optimize the use of available time during maintenance windows and keep end users happily working during distributions. These are just some of the ways that you can put ITEM to work for you. ITEM has many customization options that can be configured to meet your organization's needs. Hopefully this information has answered some of your questions about ITEM and provided some insight into how to get the most out of this useful product.

Learn More

Learn more about how you can improve productivity, enhance efficiency, and sharpen your competitive edge through training.

- IBM Endpoint Manager 9 Platform Fundamentals (TP403G)
- IBM Tivoli Endpoint Manager 8.2 Content Development and Customization (TP432G)
- IBM Tivoli Endpoint Manager 8.2 Platform Fundamentals (TP402G)

Visit **www.globalknowledge.com** or call **1-800-COURSES (1-800-268-7737)** to speak with a Global Knowledge training advisor.

About the Author

David Biondi is the founder and CEO of ESM Solutions, which is an IT consulting company that is recognized as an IBM Advanced Business Partner that's accredited across multiple IBM product groups. In addition to working as an IT consultant, David also takes pride in his work as an instructor and course director for Global Knowledge.

David has a strong commitment to broadening his knowledge base and keeping his skills cutting edge. He is ITIL® Foundations certified and maintains current certifications on many IBM products including Tivoli Endpoint Manager. David is also pursuing a master's degree in computer science from Georgia Tech where he is focusing on artificial intelligence and machine learning.