Tivoli Endpoint Manager for Configuration Management

Checklists Guide

for Windows and UNIX

Note: Before using this information and the product it supports, read the information in Notices.

Contents

Part One

Configuring Windows checklists

The Configuration Management checklists for Windows systems are delivered as a set of Fixlets and tasks that can help you find the information you need to manage your deployment.

Understanding Windows-based Configuration Management

The Configuration Management checklists for Windows-based platforms are distributed by Tivoli® Endpoint Manager in externally-provided Fixlet sites. Each site represents a single platform or standard combination, such as DISA STIG on Windows XP or FDCC on Windows Vista.

Each Fixlet corresponds to a specific configuration setting and uses the standard Tivoli Endpoint Manager Relevance language to define how that particular setting is evaluated on the Windows-based endpoints. Standard Fixlet fields include the following categories:

- **Name** – A descriptive title for the Fixlet
- **Description** – A plain-text explanation of the source of the problem and various remedies
- **Source ID** – An identifier based on the standard addressed by the particular Fixlet site
- **Category** – Fixlets are grouped into categories that allow you to sort, group, and find them by function
- **Source** – An indicator of the originating standard and version from which the configuration setting was drawn

Each check is assigned a category, such as File Permissions or Password Guidelines, which can be used for sorting or reporting. Checks have associated actions and tasks that can provide one or more of the following features:

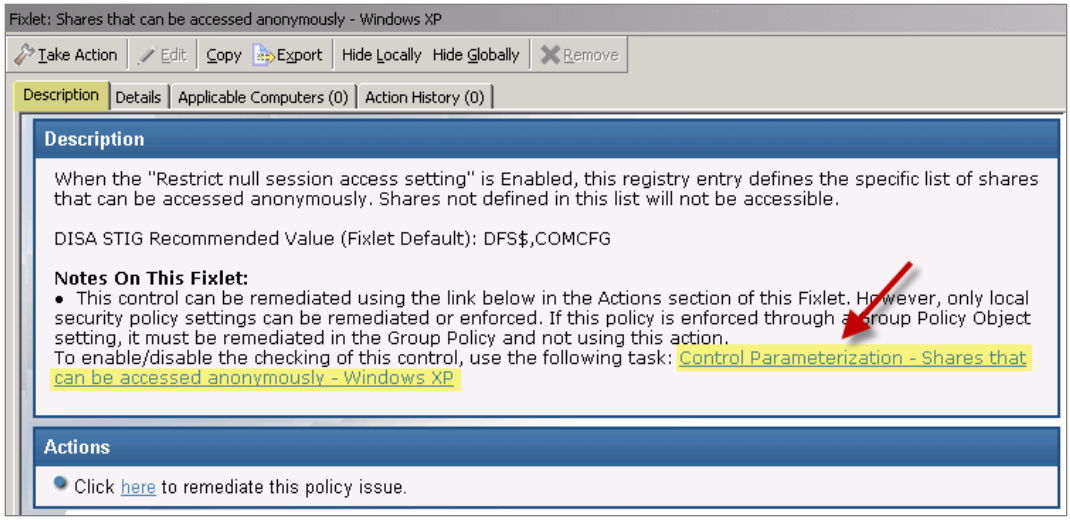
- **Enable/Disable Fixlet evaluation** – You can exclude the given Fixlet from evaluation on one or more endpoints. This is a toggle that you can turn back on to include the Fixlet again.
- **Parameterize Fixlet** – You can change the parameter value of a Fixlet on one or more endpoints.
- **Remediate Issue** – You can enforce and reset the actual value of the configuration setting on one or more endpoints.

Disabling Windows checks

You might want to stop the Relevance evaluation of a Fixlet for a certain segment of your endpoints. You can do this by creating a custom site or by disabling the Fixlet for specific computers.

To disable a Fixlet for a given set of computers, follow the steps below:

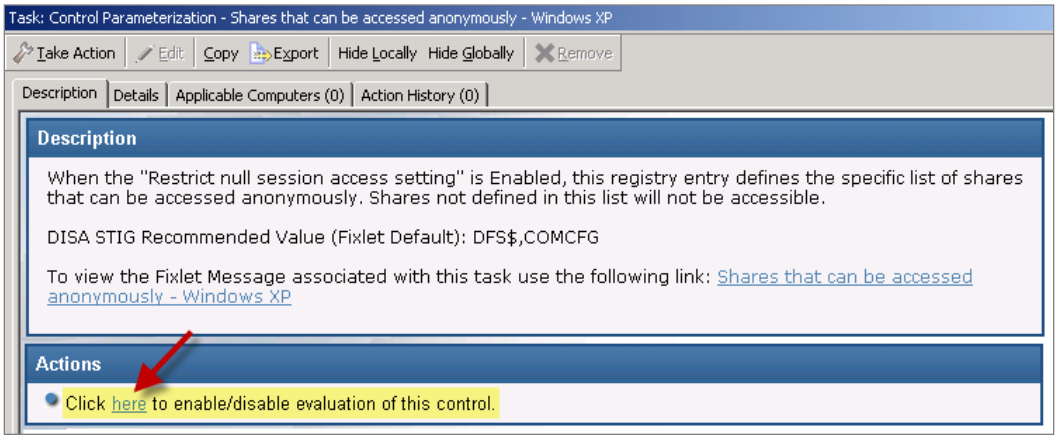
1. After opening a Fixlet, click the *Description* tab to see the message associated with this particular check.



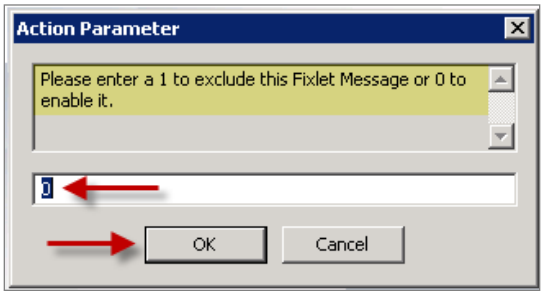
2. If the selected Fixlet can be disabled, click the *Check Parameterization* link at the bottom of the description to access the related settings task.

3. The associated task displays in the work window, typically with a title starting with “Check Parameterization”. Select the *Description* tab.

4. At the bottom of the description, click the link in the Actions box to enable or disable the evaluation of the check.



5. An Action Parameter dialog is displayed. Enter a “1” to disable the Fixlet check.

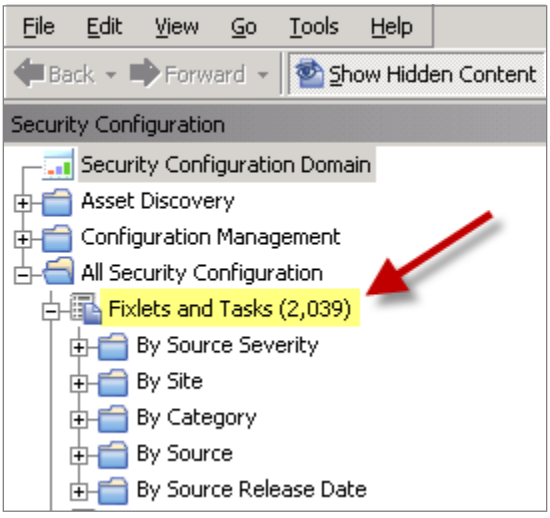


6. The Take Action dialog is displayed, where you can target the set of machines on which you want to disable the check. To deploy the action, click *OK* and enter your password. If you disable the check on all applicable computers, this Fixlet is no longer visible in the list of relevant Fixlets.

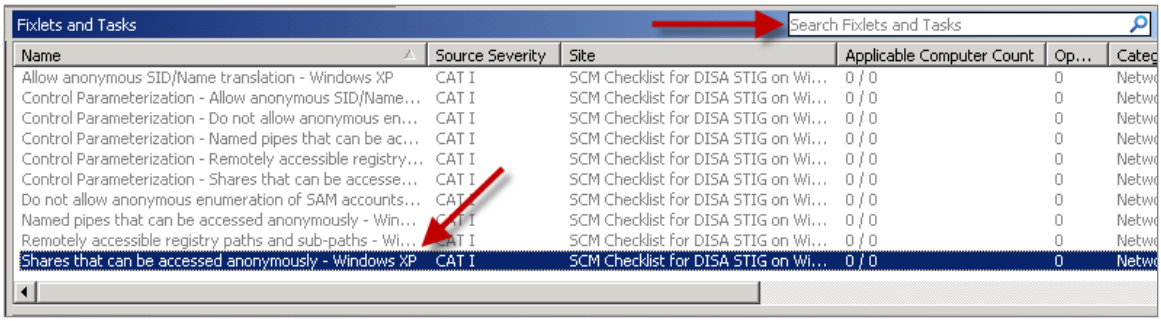
Enabling Windows checks

You can enable a Fixlet that has been disabled by the previous procedure by entering a “0”. However, if the Fixlet has been disabled on all endpoints, it no longer displays in the relevant Fixlet list. Because it is still stored in the Fixlet site, you can re-enable the Fixlet at any time.

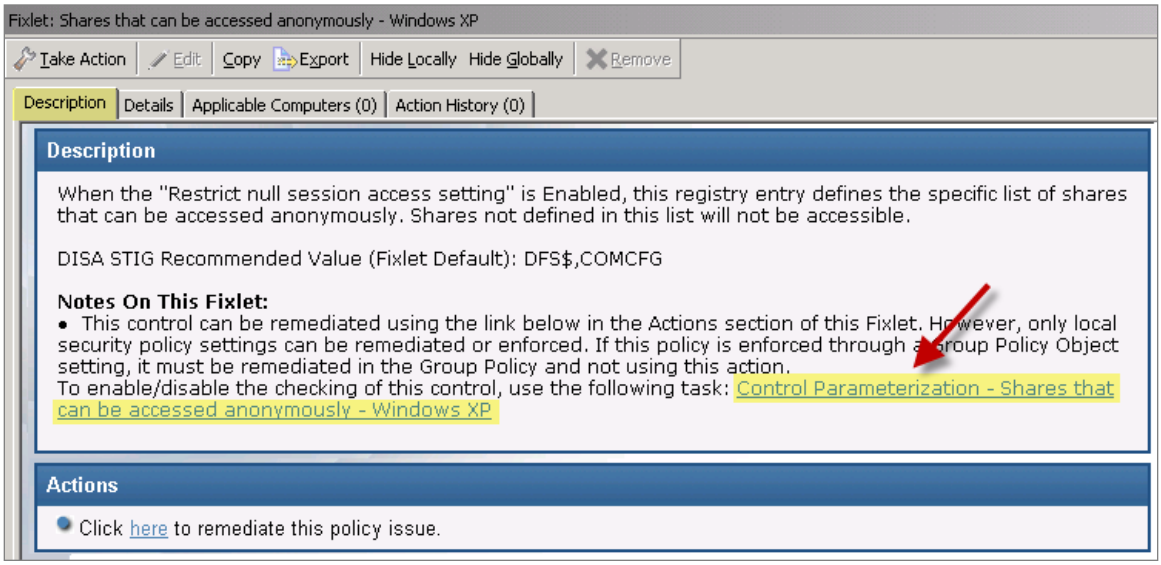
1. To locate the disabled Fixlet, click *All Security Configuration* node and expand the *Fixlets and Tasks* sub-node. You can view all Fixlets related to the entire Security Configuration domain, regardless of their relevance to a particular Configuration Management site.



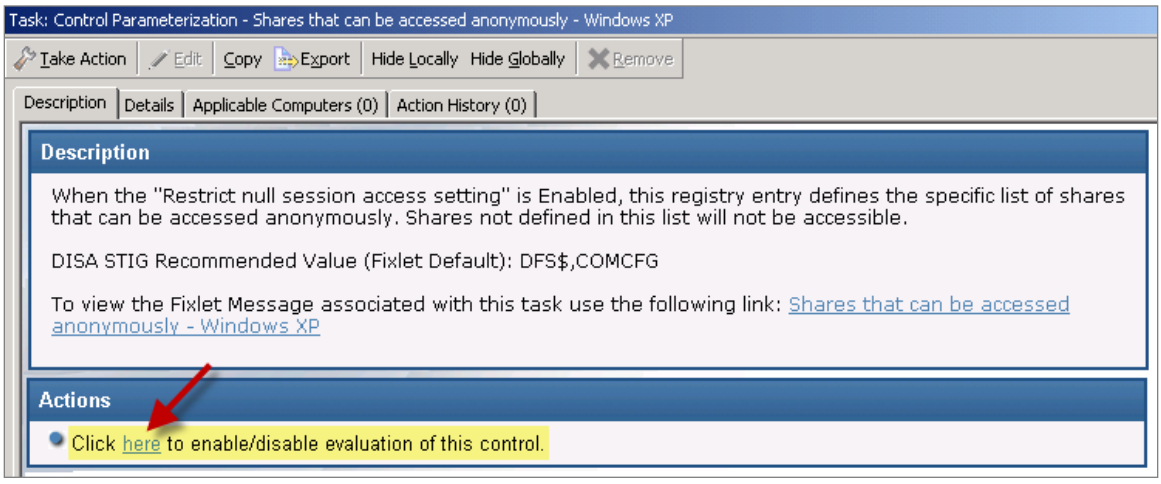
2. Search for the Fixlet you want by clicking the subfolders (Source Severity, Site). Double-click the Fixlet to view it in the work panel, or enter the Fixlet name in the Search box in the upper right of the console.



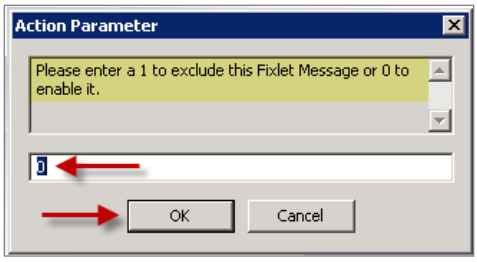
3. In the Fixlet window, click the *Description* tab and scroll down to see the *Check Parameterization* link.



4. To see the related settings task, click the *Control Parameterization* link.



5. To enable the Fixlet, click the enable/disable link and enter a “0” (zero) in the Action Parameter dialog.



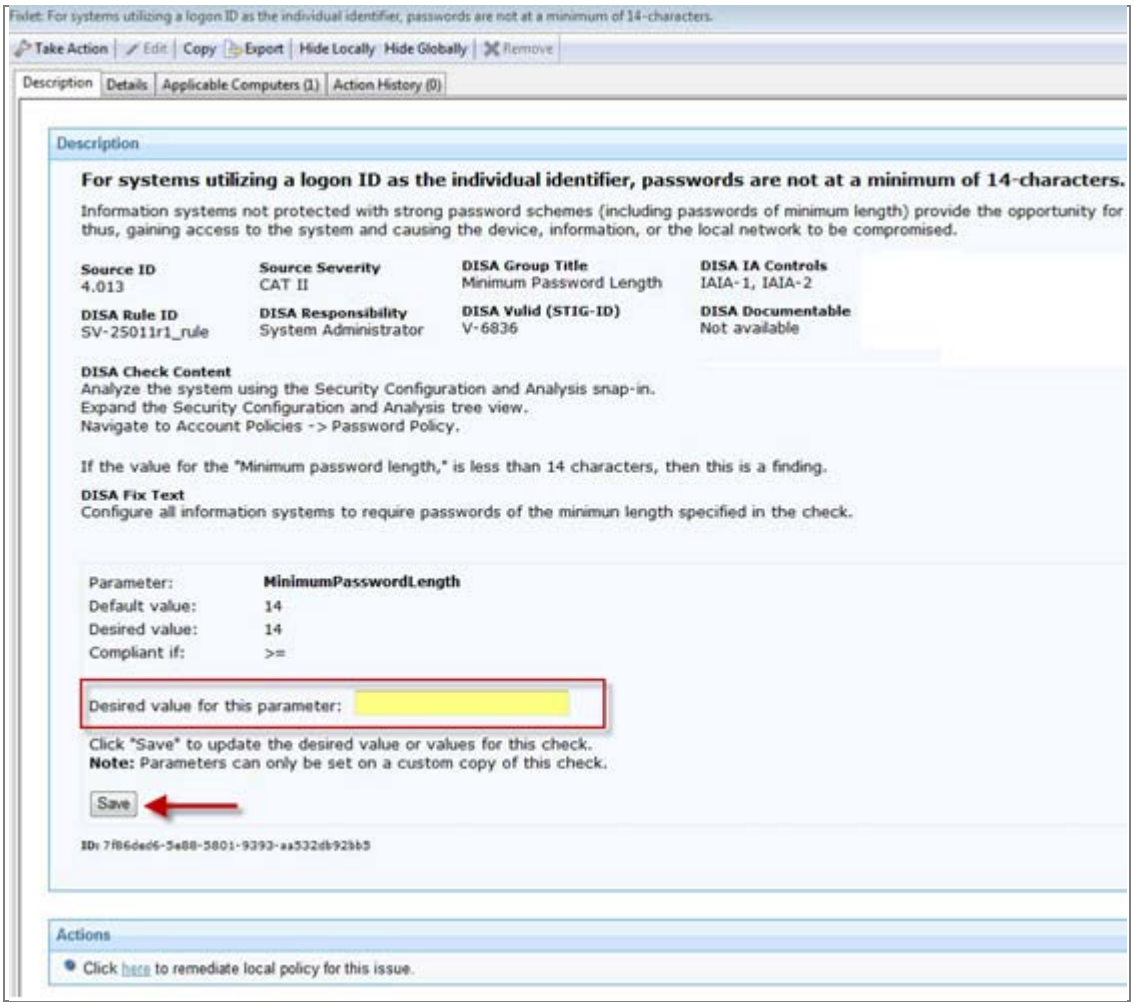
6. The Take Action dialog opens. As before, target the computers, click *OK,* and enter your Private Key Password to deploy the action. If there are any computers out of compliance with this issue, the check is displayed again in the Fixlet list after several minutes.

By using this method for enabling and disabling Windows checks, the Fixlet always displays as Not Relevant (Compliant). This means that the check always shows as compliant in the dashboard and reports. This feature can be applied to any set of computers by using the Take Action dialog.

Modifying Windows check parameters

In some cases, you can modify the parameters used in determining the compliance of checks. For example, you can set the minimum password length on an endpoint to be 14 characters. You can customize the password-length parameter to your specific policy.

To set a parameter value for a given check, find the text input titled “Desired value for this parameter” on the Description tab. Some checks have more than one parameter. Type the value into this input field. In many cases, the Description tab also contains a table of example values that are valid for the parameter. Click *Save*. If successful, you see the “Desired value:” text above the input check change to the value you just entered.



Although parameters can be modified on both Windows and UNIX content, there are differences in how these parameters are implemented. UNIX content is aimed at users who want maximum command-line control.

Not all checks can be parameterized. For information about parameterization for UNIX platforms, see the AIX, Linux, and Solaris parameterization guides available on the Tivoli Endpoint Manager support site.

Only copies of checks located in custom sites can be parameterized.

Remediation of Windows configuration settings

You can audit, assess, and remediate configuration settings using Tivoli Endpoint Manager Configuration Management. For Fixlet checks that can be automatically remediated, you receive an action displayed in the relevant Fixlet. To remediate a configuration setting, perform the following steps:

1. Double-click a Fixlet in the Console list.
2. Click the *Description* tab and scroll down to the Actions box.
3. Click in the Actions box link to remediate the specified policy issue.
4. Set your parameters in the Take Action dialog and click *OK.*
5. Enter your password, and the remediation action deploys across your network to the specified computers. The action changes the value of a setting in a file in the Windows in registry. That setting can be the value supplied by the default Fixlet check or the value you supplied if you customized the parameter.

> ***Note:*** *Not all Fixlets have a remediation action. For more information, see the Knowledge Base on the Tivoli Endpoint Manager Support website.*

Analyses

The Configuration Management DISA for Windows checklists include analyses. Each check Fixlet in the DISA Windows content has an associated analysis. Check Fixlets display the compliance state, and analyses display the actual state of each configuration item.

These analyses are provided to enable the display of "Measured Values" in Tivoli Endpoint Manager Security and Compliance Analytics. If you are using only a subset of the available check Fixlets for your implementation, activate only the analyses that are associated with the check Fixlets you are using.

Part Two

Configuring UNIX checklists

Overview

Configuration Management checklists for UNIX systems are provided as a set of Fixlets and a single task used to scan a UNIX system *on demand* or *periodically* via scheduling. Each Fixlet includes key attributes to help you to manage information about your deployment. These attributes remain attached to the Fixlet even when you copy them to a custom site. Fixlets organize information through the following categories:

Name	A descriptive title for the Fixlet
Description	A plain-text explanation of the source of the problem and various remedies
Source ID	An identifier based on the standard addressed by the particular Fixlet site
Category	Fixlets are grouped into categories that allow you to sort, group, and locate them by function
Source	Indicates the originating standard and version from which the configuration setting was taken
Source Severity (DISA sites only)	The DISA-defined severity for each Fixlet or check

The UNIX content executes a task that runs each of the defined Configuration Management UNIX checks in a batch, as distinct from the real-time assessment employed by the Windows site. When the batch file runs, the results are evaluated on the chosen endpoints, and these results are logged and made available to the corresponding Fixlet checks for evaluation. Fixlets then use the Tivoli Endpoint Manager Relevance language to examine the log and determine relevance. The results are shown in the Tivoli Endpoint Manager console, where compliance can be determined.

Setup and configuration

Setting up your Configuration Management checklists for UNIX involves three basic steps:

- **Create your checklist**
  Use the *Create Custom Checklist* wizard located in the Checklist Tools folder in the navigation tree. For more information about using this wizard, see the Configuration Management User's Guide.

- **Configure your checklist**
  Select checks via task, and then parameterize checks (console and system).

- **Run your checklist**
  Modify run behavior, check global filescan, and then schedule a run task.

> ***Note:*** *Configuration Management measured values only work if analyses are activated. For each Fixlet you want to deploy, activate the appropriate analysis. For more information, see the Analyses section in this document.*

Configuring checklists

Configuring a checklist is an optional step where you configure the task to be used to run the content itself.

Select checks via task

The default behavior for UNIX Configuration Management deployment is to run the scripts as a single batch. However, you can also run any subset of the checks on your own defined schedule. Each time you do this, the batch that you deploy overwrites any previous batch commands. The **`runme.sh`** master script provides a '-F' option, which takes a file name as its argument. It has the following form:

```
./runme.sh -F <FILE>
```

This command causes runme.sh to perform *only* the set of checks specified in <FILE>. This is a 7-bit ASCII file with UNIX newlines containing a list of the specific checks you want to run, of the form:

```
GEN000020
GEN000480
GEN000560
```

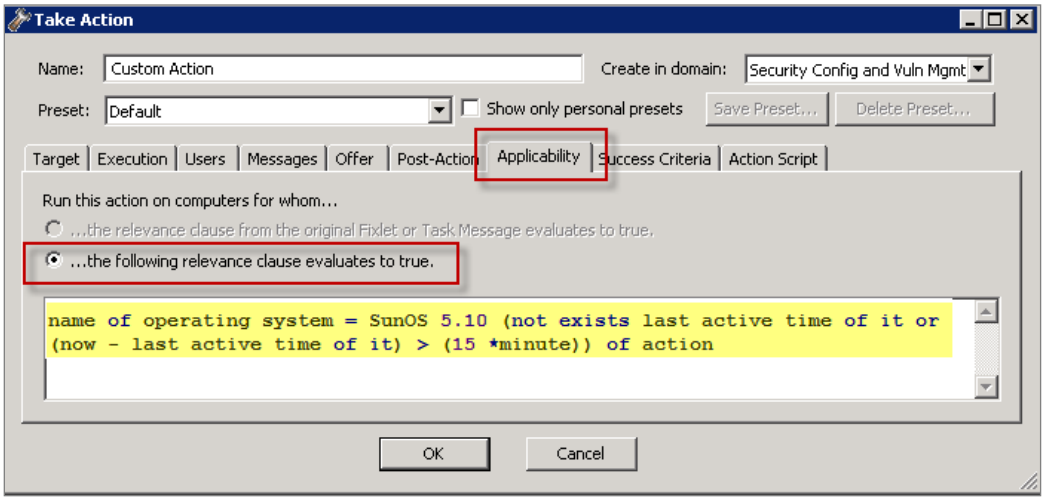
This function allows you to run only the scripts you need when you need them. To enable this function, create a custom action. This action creates the file containing the list of checks and then deploys it to your chosen Tivoli Endpoint Manager clients. This action is similar to creating a custom parameter file.

To create your own custom set of checks, perform the following steps:

1. From the Tools menu in the console, select *Take Custom Action.* This opens the Take Action dialog.

2. On the *Target* tab of the Take Action dialog, select the endpoints on which you want to create checks.

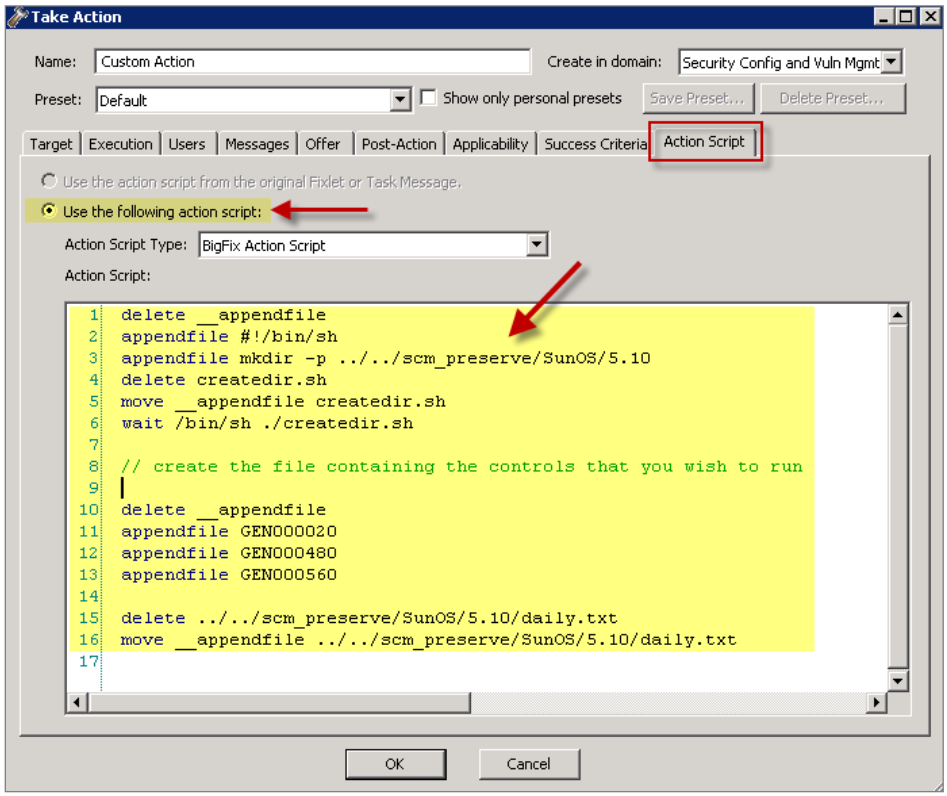
3. On the *Applicability tab,* click the second button to run this action on computers with a custom relevance clause.



In the text box, enter a relevance clause to identify a subset of computers you want to target. For example, to restrict the action to Solaris 10 systems, enter the following expression:

```
name of operating system = "SunOS 5.10" (not exists last
active time of it or (now - last active time of it) > (15
*minute)) of action
```

4. Click the *Action Script* tab to create a script that copies your file onto the target computers. Click the second button and then enter a script such as the one shown below.



The script creates the target directory with the file containing the checks to run and then moves the file into the appropriate directory.

Below is a sample script that you can copy and paste, which specifies three checks: GEN000020, GEN000480, and GEN000560.

```
// create a script that will create the necessary directory
delete __appendfile
appendfile #!/bin/sh
appendfile mkdir –p ../../scm_preserve/SunOS/5.10
delete createdir.sh
move __appendfile createdir.sh
wait /bin/sh ./createdir.sh

// create the file containing the checks that you wish to run
delete __appendfile
appendfile GEN000020
appendfile GEN000480
appendfile GEN000560
delete ../../scm_preserve/SunOS/5.10/daily.txt
move __appendfile ../../scm_preserve/SunOS/5.10/daily.txt
```

Parameterize checks

Many factors can influence the need to customize security policies. Part of this customizing process includes changing the values for defined configuration settings to meet specific corporate policies. Tivoli Endpoint Manager allows you to customize the content in the default Fixlet site by special targeting, customizing parameters, and disabling checks. Custom sites offer even greater flexibility.

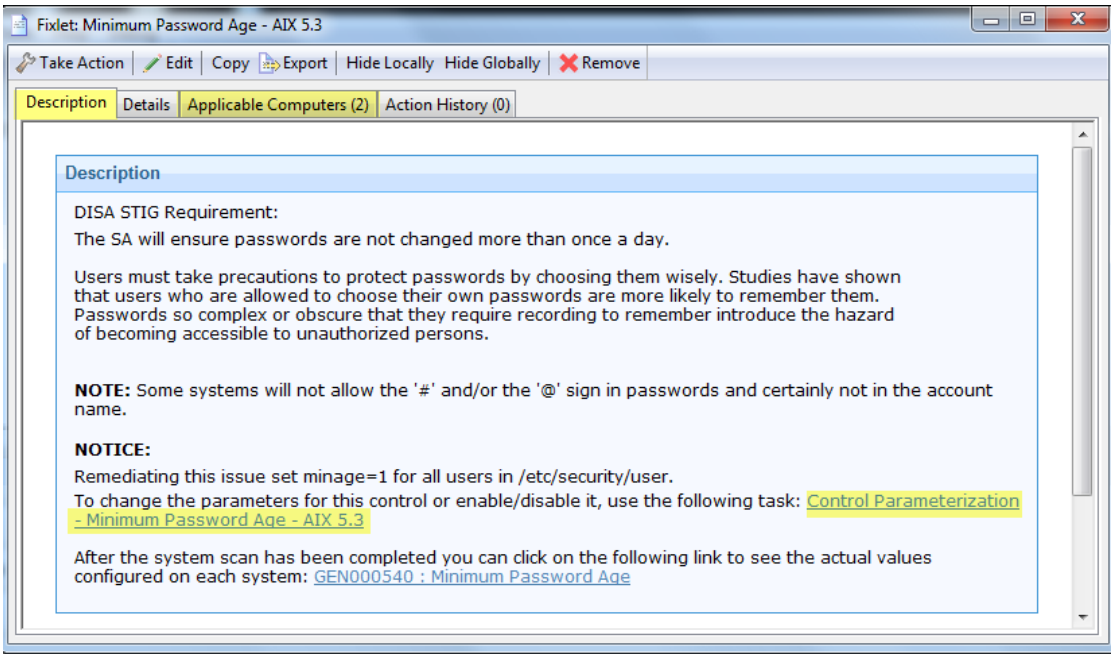
Fixlet checks can be parameterized to suit each individual situation. Because parameters are stored as site settings, you can parameterize the same check differently for each site containing a copy of the check.

You can parameterize UNIX Configuration Management checks in the following ways:

- Some checks can be parameters from within the console.
- Other checks can be set using a customer parameter file stored on the UNIX system. See the descriptions below for an explanation of each option.

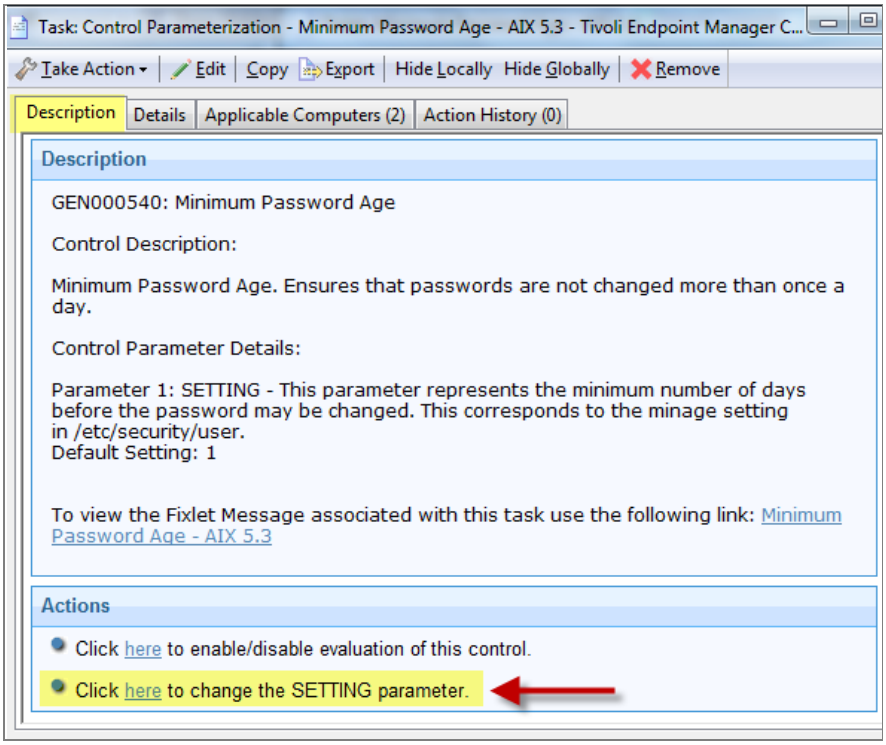
Console option

You can modify parameters for Windows content by using the task associated with the particular Fixlet. From the Fixlet site named *Configuration Management Checklist for DISA STIG on Windows 2003*, select a Fixlet. The Fixlet opens in the work area. Select the Description tab.



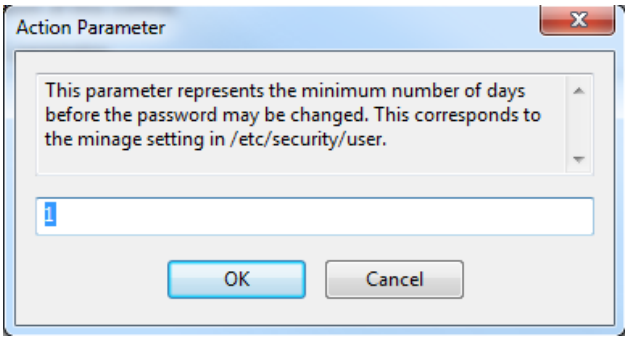
The bottom of the Description box contains a Check Parameterization link. To analyze the relevance clause attached to this Fixlet, click the Details tab. To view affected computers, click the Applicable Computers tab.

1. To open a task, click the *Check Parameterization* link under the Description tab.



You see two actions associated with this task located in the Actions box. The first action lets you toggle the evaluation and the second action lets you modify the parameter associated with the check.

2. To configure the parameter for this check, click the second link.



The recommended parameter is the default value (in this case 1), or the last value entered if you have previously customized the parameter. Enter a new value or click *OK* to accept the existing value.

3. Select the parameters of your action in the Take Action dialog, click *OK,* and enter your password to send the action.

You have now set a parameter for the specified Fixlet, which propagates to the targeted computers to align them with your corporate policy.

System level option

In some cases, the UNIX Configuration Management content might not have a parameter task in the console. The content can still be parameterized at the system level, where you modify the customer_params file by using a task.

Tivoli Endpoint Manager UNIX Configuration Management checklist sites have pre-configured default values for various operating system settings according to a designated standard. However, it is possible to customize your deployment to meet the specific settings required by your organization. This is done by modifying the parameters passed to the UNIX checks. A list of the UNIX parameters is contained in the *SCM Parameter* documents. This section describes how to adjust them.

> ***Note:*** *Before running the Deploy and Run Security Checklist task included in the respective site, perform the steps below.*

To customize the parameters of a check, create and maintain a text file, on each machine, that contains one line for each check you want to override. The line must contain the name of the check, the parameters to customize, and the new value, as follows:

```
CHECK_ID:PARM_NAME=PARM_VALUE
```

To specify a minimum length of 6 and one alphabetic character in each password, customize two controls. The file must have two lines, one per control:

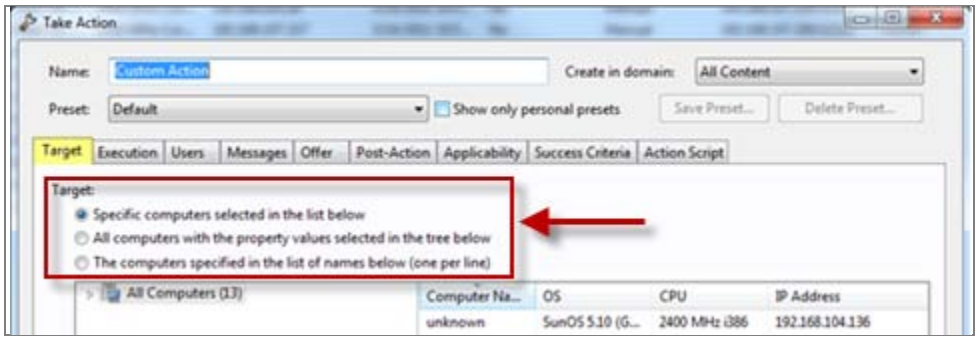
```
GEN000580:VALUE=6
GEN000600a:VALUE=1
```

In this example, the name of the parameter is VALUE. The *Configuration Management Parameter Guides describe* the individual checks, their parameter names, and the default values of each. Consult those documents to see which checks can be parameterized and their default values.

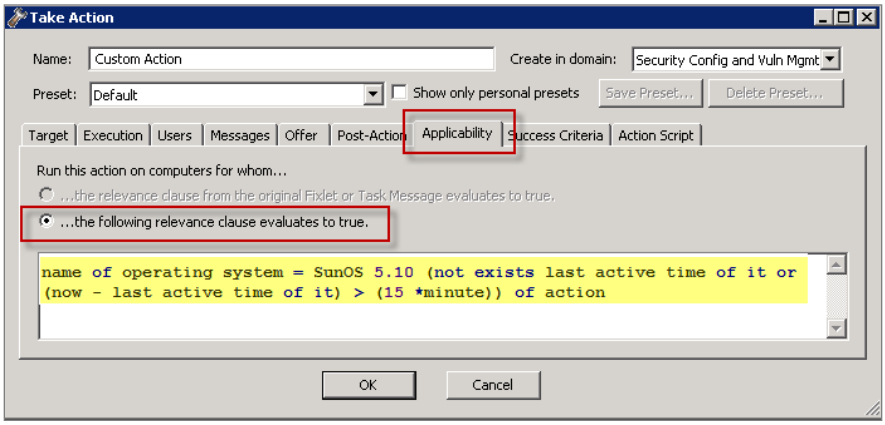
The basic steps for parameterization are as follows:

1. Create a custom action to deploy the override file to the appropriate endpoints. To do this, click Tools and *Take Custom Action*. The Take Action Dialog opens.

2. Under the *Target* tab of the Take Action Dialog, select the computers you want to customize from the list.



3. Click the *Applicability* tab and select the second button to run the action on computers with a custom relevance clause.



In the text box, enter the following relevance expression:

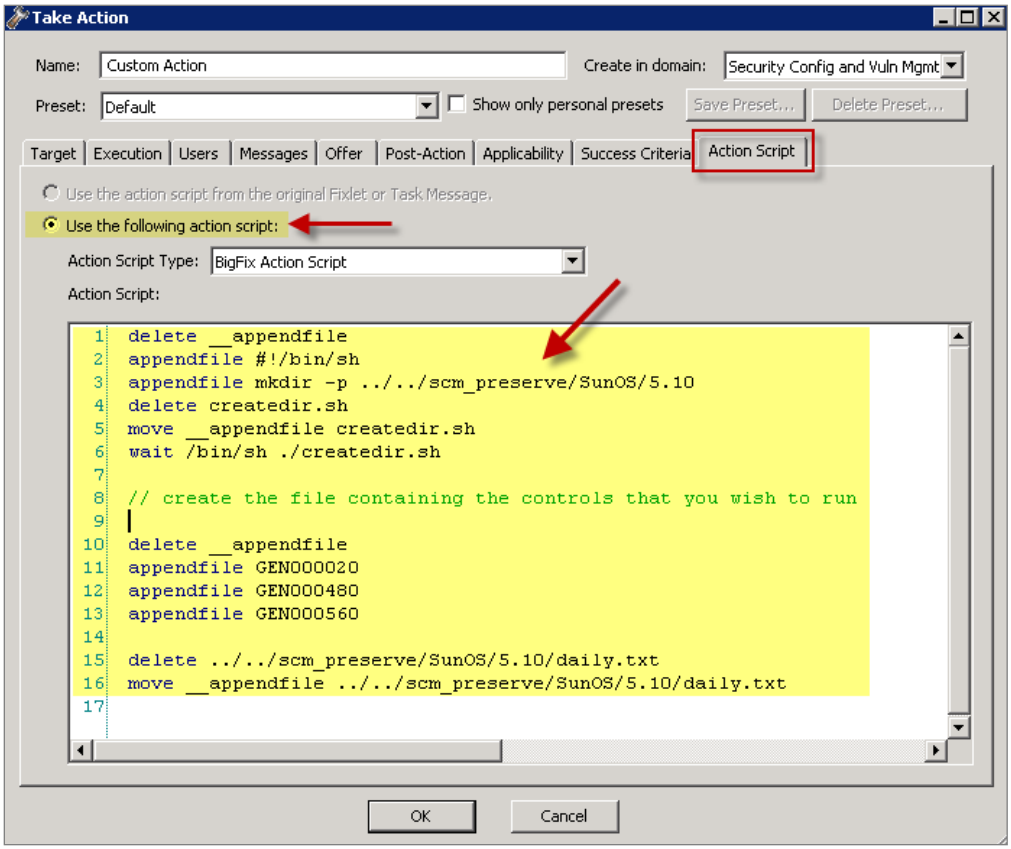
```
name of operating system = “SunOS 5.10” AND (not exists
last active time of it or (now - last active time of it) >
(15 *minute)) of action
```

This expression restricts the action to Solaris 10 systems and ensures that the task reapplies successfully if reapplication behavior is specified on the *Execution* tab.

For a list of the various operating system strings that can be used, see the table below:

Operating System	String
Windows 7	Win7
Windows 2008	Win2008
Windows XP	WinXP
Windows Vista	WinVista
Windows 2003	Win2003
SUN Solaris 10	SunOS 5.10
SUN Solaris 9	SunOS 5.9
SUN Solaris 8	SunOS 5.8
IBM AIX 5.1	AIX 5.1
IBM AIX 5.2	AIX 5.2
IBM AIX 5.3	AIX 5.3
HP-UX 11.0	HP-UX B.11.00
HP-UX 11.11	HP-UX B.11.11
HP-UX 11.23	HP-UX B.11.23
Red Hat Enterprise Linux 3	Linux Red Hat Enterprise AS 3
	Linux Red Hat Enterprise ES 3
	Linux Red Hat Enterprise WS 3
Red Hat Enterprise Linux 4	Linux Red Hat Enterprise AS 4
	Linux Red Hat Enterprise ES 4
	Linux Red Hat Enterprise WS 4
Red Hat Enterprise Linux 5	Linux Red Hat Enterprise AS 5
	Linux Red Hat Enterprise ES 5
	Linux Red Hat Enterprise WS 5

4. To create a script that copies the file onto the target computers, click the *Action Script* tab. Click the second button to enter a script.



Insert a script in the text box to create the target directory with the file containing your custom parameters. The script must then move the file into the appropriate directory.

Below is a sample script that customizes password parameters:

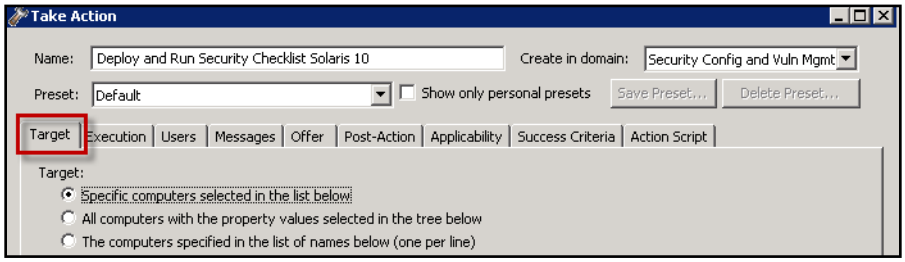
```
// create a script that will make the necessary directory
delete __appendfile
appendfile #!/bin/sh
appendfile mkdir -p ../../scm_preserve/SunOS/5.10
delete createdir.sh
move __appendfile createdir.sh
wait /sbin/sh ./createdir.sh

// create the customer_params file and move it to the correct place
delete __appendfile
appendfile GEN000580:VALUE=6
appendfile GEN000600a:VALUE=1

delete ../../scm_preserve/SunOS/5.10/customer_params
move __appendfile ../../scm_preserve/SunOS/5.10/customer_params
```

Running checklists

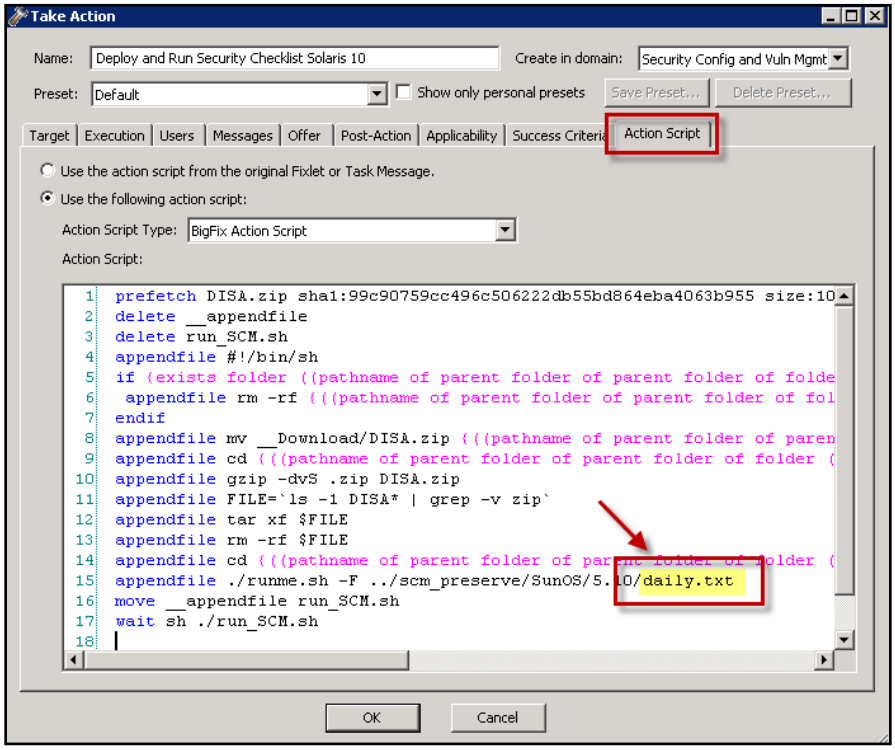
Modify run behavior

The Master Run script (runme.sh) runs the individual check scripts located on the UNIX system when the *Deploy and Run Security Checklist* task is run. By default, the master script runs all Tivoli Endpoint Manager check scripts, but this behavior can be modified by using the –F option.

Make a custom copy of the *Deploy and Run Security Checklist* task (for the given operating system) that comes with the content. Find this task, double-click it, and select the endpoints you want in the Take Action dialog.



Click the *Action Script* tab. Modify the Action Script to make runme.sh use the –F option and point to the file that contains the checklist. In the example below, the file is named daily.txt (file names are arbitrary).



Below is a sample script that you can copy, paste, and modify:

```
prefetch DISA.zip sha1:99c90759cc496c506222db55bd864eba4063b955 size:108089
http://software.bigfix.com/download/SCM/SunOS-20080417.zip
delete __appendfile
delete run_SCM.sh
appendfile #!/bin/sh
if {exists folder ((pathname of parent folder of parent folder of folder
(pathname of client folder of current site)) & "/SCM")}
 appendfile rm -rf {((pathname of parent folder of parent folder of folder
(pathname of client folder of current site)) & "/SCM")}
endif
appendfile mv __Download/DISA.zip {((pathname of parent folder of parent
folder of folder (pathname of client folder of current site)))}
appendfile cd {((pathname of parent folder of parent folder of folder
(pathname of client folder of current site)))}
appendfile gzip -dvS .zip DISA.zip
appendfile FILE=`ls -1 DISA* | grep -v zip`
appendfile tar xf $FILE
appendfile rm -rf $FILE
appendfile cd {((pathname of parent folder of parent folder of folder
(pathname of client folder of current site)) & "/SCM")}
appendfile ./runme.sh -F ../scm_preserve/SunOS/5.10/daily.txt
move __appendfile run_SCM.sh
wait sh ./run_SCM.sh
```

In addition to the –F option, there are several other options that you can use on the master run script to change the behavior:

Options	*Behavior*
-g	This is the default option. Run *globalfind* and run all scripts.
-t	Turn on tracing. The master script creates a trace file of the commands executed by the OS-specific scripts. This option is used for debugging only.
–f <source id>	Run a single check, where <check> is the name of the check to be run.
–F <FILE>	Runs all scripts listed in <FILE>. This allows you to run any subset of scripts by listing them in a file. When specifying the –F option, the file format must be a 7-bit ASCII text file with UNIX-style newline characters.

Note: *When using the –F option, the contents of the <FILE> include a list similar to the following:*

```
GEN000020
GEN000400
GEN000440
```

Click *OK* and enter your Private Key Password to run the action.

Note: *Several checks use the globalfind utility and require a fresh find.out file to work correctly. If you are running one or more of the following checks, you must supply the '-g' option to runme.sh.*

The following checks require the global option and can be included with the '-F' option only if the '–g' option is also supplied.

GEN001160	GEN001180	GEN001200	GEN001220
GEN001240	GEN001260	GEN001280	GEN001300
GEN001360	GEN001540	GEN001560	GEN002160
GEN002180	GEN002200	GEN002220	GEN002240
GEN002280	GEN002480	GEN002500	GEN002520
GEN002540	GEN005360c	GEN006340	GEN006360
SOL00360	SOL00380		

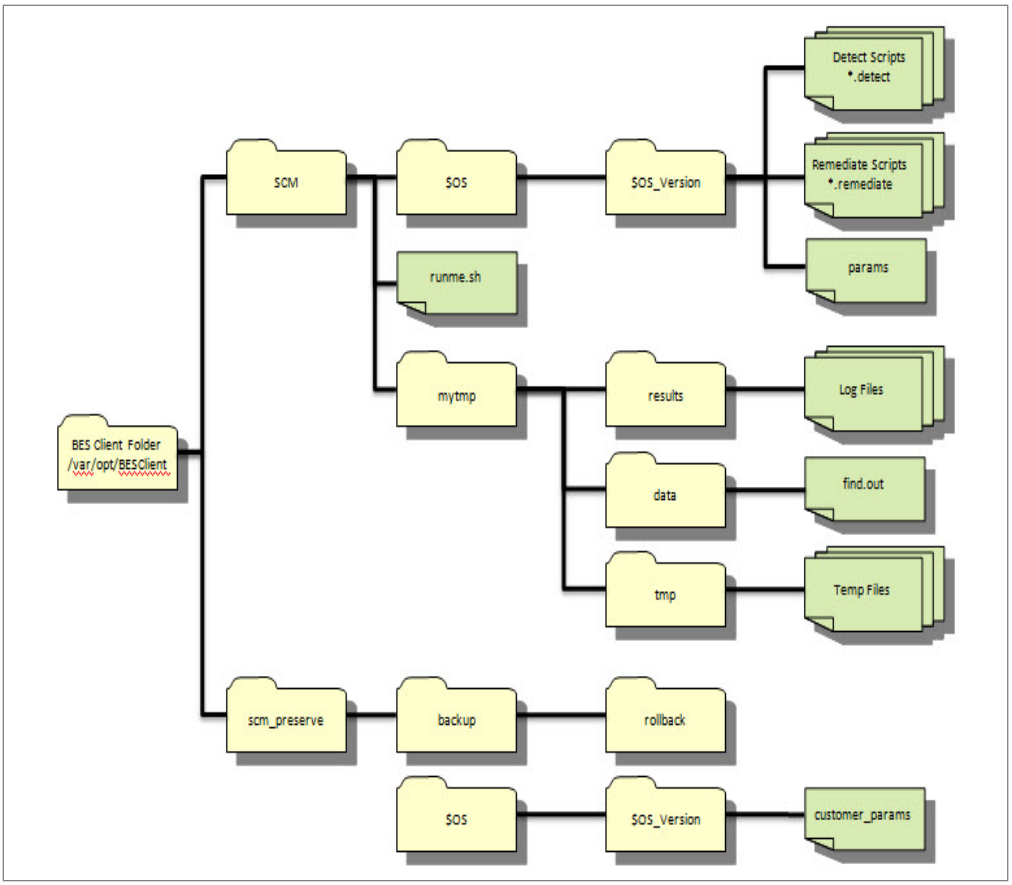
Understanding the output

With most Tivoli Endpoint Manager content, Fixlets constantly evaluate conditions on each endpoint and results are displayed in the console when the relevance clause of the Fixlet evaluates to true.

With UNIX content, a task initiates a scan of the endpoints, which can be run on an ad-hoc basis each time a scan is required, or can be run as a recurring policy from the console.

The endpoint scan is accomplished by a series of UNIX Bourne Shell scripts. While each script runs, it detects a setting or condition and then writes the information to an output file that is made available to the corresponding Fixlet check for evaluation. When the log files have been written to disk, the Fixlets read each log file and display the results in the console. Although the end result is similar, this method of detection provides greater accessibility to UNIX system administrators.

After running the Deploy and Run Security Checklist task, the scripts are located in a directory under /var/opt/BESClient/SCM. Below is a graphical representation of the directory structure:



<BES Client Folder> / SCM	This is the base directory for the OS-specific check scripts and the master script (runme.sh). The contents of this directory are overwritten each time the 'Deploy and Run Security Checklist' task is run from the Tivoli Endpoint Manager console.
../SCM/util	A subdirectory of the BES Client Folder / SCM directory, this contains utility scripts that are used by the master script and in the individual detection and remediation scripts. The primary utility found in this directory is the 'globalfind' script.
../SCM/$OS/$OS_version	This directory is specific to the platform on which it runs as specified by $OS and $OS version. For example, the Red Hat Enterprise Linux 4 shows as (../SCM/Linux/4). This directory path contains the specific detection scripts, remediation scripts, and the base parameter file used by the scripts. Each check script is named with the corresponding control ID that is used to describe the check. Each corresponding Fixlet also references the check ID.
../SCM/runme.sh	This is the master script that is called by the 'Deploy and Run Security Checklist' task within the BigFix Console. This script runs the 'globalfind' script and

	the individual check scripts.
../SCM/mytmp/results	This folder is where the OS-specific detection scripts write their log files. These logs are examined by Fixlets and used to determine if a check is compliant or non-compliant. Each log file corresponds to the check ID for the given check.
../SCM/mytmp/data	This folder contains the **find.out** file. This file is generated by the **globalfind** script and contains a directory listing of all local file systems and other information. This file is used by many of the OS-specific scripts and is updated only when the **globalfind** script is run.
<BES Client Folder>/scm_preserve	This is the base directory that is used to retain the rollback scripts, custom checks, parameters, and other information not intended to be overwritten each time the 'Deploy and Run Security Checklist' task is run.
../scm_preserve/backup/rollback	Each time a remediation script is run, a corresponding rollback script is created. This allows the administrator to roll back to the previous setting associated with the specific check.
../scm_preserve/$OS/$OS_version	This directory might contain custom scripts produced by the administrator and not provided by Tivoli Endpoint Manager. Scripts that are located in this directory must conform to the input / output specifications are run in conjunction with out-of- the-box checks when running the 'Deploy and Run Security Checklist' task.
../scm_preserve/$OS/$OS_version /customer_params	This file is used to store any custom parameters that are defined by the administrator. Any parameters defined in this file override the default parameters specified in the **params** file stored in <BES Client Folder>/SCM/$OS/$OS_version/params).

Each operating system-specific script writes two files in **/var/opt/BESClient/mytmp/results**. The filenames correspond to the name of the OS-specific script. For example GEN000020.detect writes two files GEN000020.detect.log and GEN000020.results.

The file with the .log extension contains the STDOUT and STDERR of the operating system-specific script. Under normal conditions, this file is empty. When **runme.sh** is run with the **–t** option, this file contains the trace output of the operating system-specific script.

When created, the files with the **.results** extension are read by a Fixlet and the result becomes available through the Tivoli Endpoint Manager console. The Fixlets examine the [STATUS] section to determine relevance.

The following is an example of a results file:

```
[RUN_DATE]
01 Apr 2008
[RUN_DATE_EOF]
[DESCRIPTION]
The UNIX host is configured to require a password for access to single-user
and maintenance modes
[DESCRIPTION_EOF]
[FIXLET_DESCRIPTION]
This UNIX host is not configured to require a password for access to single-
user and maintenance modes
[FIXLET_DESCRIPTION_EOF]
[CHECK_COVERAGE]
DISA-STIG-GEN000020
[CHECK_COVERAGE_EOF]
[STATUS]
PASS
[STATUS_EOF]
[PARAMETERS]
CONFIG_FILE=/etc/default/sulogin;SETTING=PASSREQ;OP='=';VALUE=NO
[PARAMETERS_EOF]
[TIMETAKEN]
0
[TIMETAKEN_EOF]
[REASON]
The /etc/default/sulogin file does not exist, the system will default to
requiring a password for single-user and maintenance modes
[REASON_EOF]
```

Each of the sections found within the log file output are described in the following table:

Section Name	Description
[RUN_DATE]	Contains the date that the script was run.
[DESCRIPTION] and [FIXLET_DESCRIPTION] Deprecated	No longer used – deprecated file
[CHECK_COVERAGE]	Contains the names of the regulations to which this Fixlet applies. (No longer used – deprecated files)
[STATUS]	Used by the associated Fixlet to determine relevance. It contains one of the following strings: PASS, FAIL, or NA. If this section contains the string FAIL, then the associated Fixlet becomes relevant.
[PARAMETERS]	Contains the parameters associated with the script. Spaces display as a semicolon. On output into this file, spaces are converted to semicolons for display purposes. This is not representative of how the parameters are set.

[TIMETAKEN]	Contains the number of seconds of wall-clock time that the script took to run.
[REASON]	Contains a description of why the script passed or failed. This section provides information needed to construct analysis properties and return specific information to the Tivoli Endpoint Manager Console.

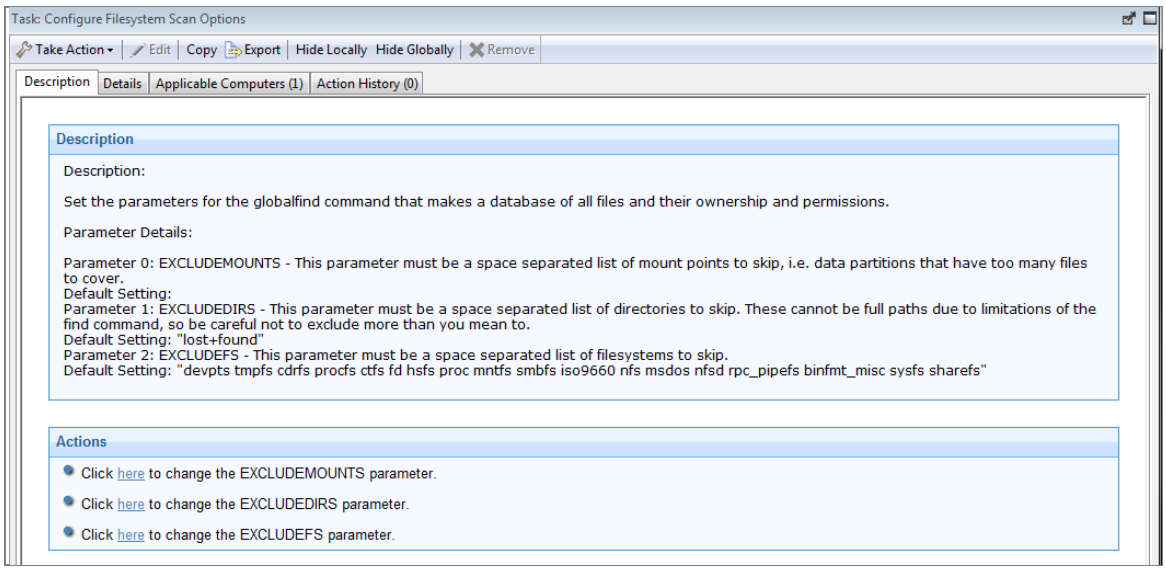
The **runme.sh** script also creates a file containing the overall results of running the various OS-specific scripts.

This file, named **/var/opt/BESClient/SCM/mytmp/results/master.results**,displays as follows:

```
TOTAL_SCRIPTLETS_RUN:69
TOTAL_SCRIPTLETS_PASS:33
TOTAL_SCRIPTLETS_FAIL:36
TOTAL_SCRIPTLETS_NA:0
TOTAL_SCRIPTLETS_ERR:0
TOTAL_TIME_TAKEN:1367
```

Modifying global scan options

UNIX content includes a global scan script that is used to perform a full system scan. The results of this scan are used in a number of scripts. The purpose of this script is to eliminate the need to run a full system scan multiple times when evaluating a set of checks on a single system. This feature allows Tivoli Endpoint Manager to be more efficient and cause less impact on the system during a configuration scan. The global scan script runs by default when using the Tivoli Endpoint Manager-provided *Deploy and Run Security Checklist* task. It is used by the Master Run script using the –g option. The behavior of the global scan script can be controlled through the Configure Filesystems Scan Options task.



EXCLUDEFS	A list of specific file systems to exclude from scanning. This must be a space-separated list of all the file system types to exclude from the search. By default, the global find script excludes the following file system types from its search: ▪ cdrfs ▪ procfs ▪ ctfs ▪ fd ▪ hsfs ▪ proc ▪ mntfs ▪ smbfs ▪ iso9660 ▪ nfs ▪ msdos
EXCLUDEMOUNTS	A list of specific mount points to exclude from scanning. This parameter must be defined as a space-separated list of all the file system mounts to exclude from the search. This prevents the shared file system from being scanned from multiple systems. For example, if several systems mount a shared directory on a Storage Area Network named /san, you might want to exclude them with a parameter such as: EXCLUDEMOUNTS="/san" By default, this parameter is not used and is represented as an empty value.
EXCLUDEDIRS	List of directories to exclude from scanning. Any directory names specified in EXCLUDEDIRS are omitted from the directory listing. By default, this parameter excludes the lost+found directory.

Note: *When you exclude a directory, you also exclude all similarly-named directories. For example, if you specify EXCLUDEDIRS="foo/", you also exclude /foo/usr/foo and /usr/local/foo.*

Scheduling specific checks

The default behavior for a UNIX deployment is to run the scripts as a single batch. However, you can also run any subset of the checks on your own defined schedule. Each time you do, the batch that you deploy overwrites any previous batch commands. The runme.sh master script provides a '-F' option, which takes a file name as its argument. It has the following form:

```
./runme.sh -F <FILE>
```

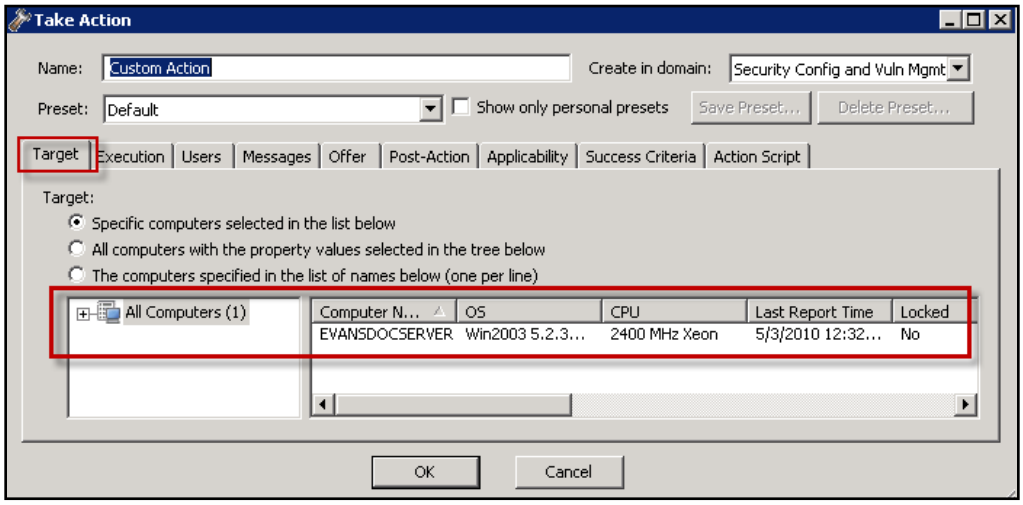
This command causes runme.sh to run *only* the set of checks specified in <FILE>. This is a 7-bit ASCII file with UNIX newlines containing a list of the specific checks you want to run, as follows:

```
GEN000020
GEN000480
GEN000560
```

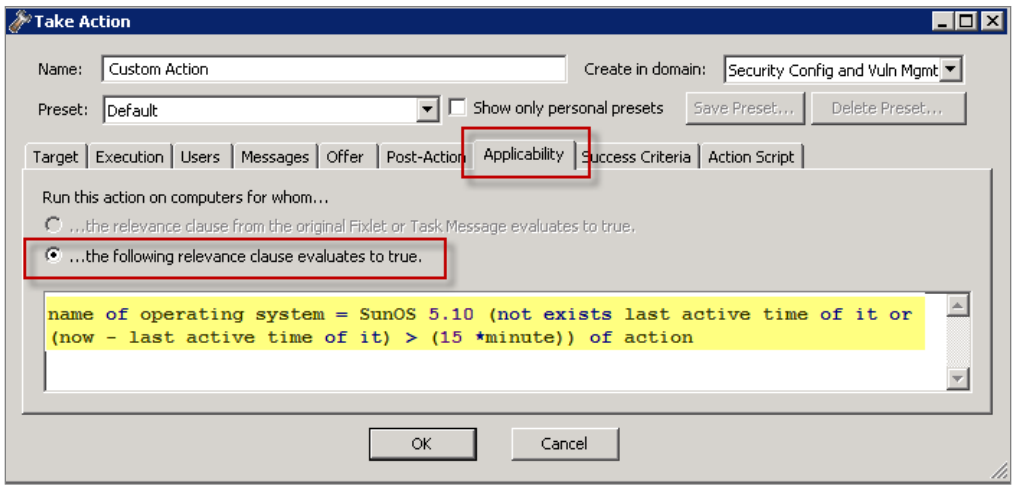
To select a specific script and run schedule, create a custom action. This action creates the file containing the list of checks and deploys it to Tivoli Endpoint Manager clients. This action is similar to the creation of a custom parameter file.

To create a custom action, perform the following steps:

1. To access the Take Action dialog, click *Tools* and select *Take Custom Action* from the console.



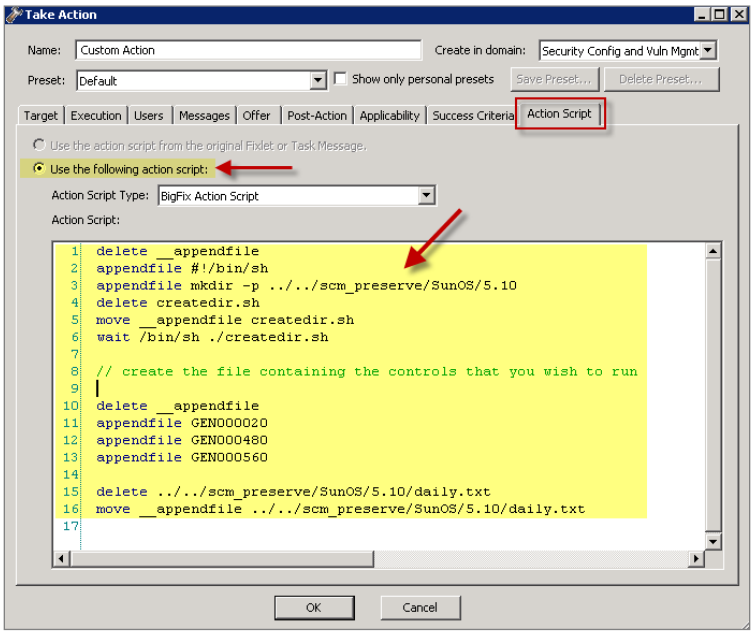
2. To run this action on computers with a custom relevance clause, click the *Applicability* tab and select the second button.



In the text box, enter a relevance clause to identify the subset of computers you want to target. For example, to restrict the action to Solaris 10 systems, enter the following expression:

```
name of operating system = “SunOS 5.10” (not exists
last active time of it or (now - last active time of
it) > (15 *minute)) of action
```

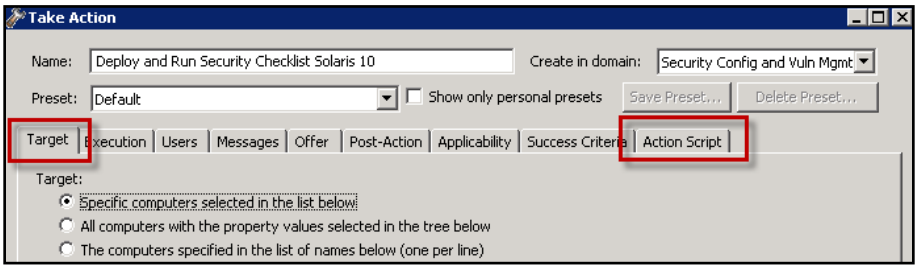
3. Click the *Action Script* tab to create a script that copies your file onto target computers. Click the second button and then enter a script like the one shown below:



This script creates the target directory with the file containing the checks you want to run and moves the file into the appropriate directory. Here is a sample script (that you can copy and paste) that specifies three checks, GEN000020, GEN000480 and GEN000560:

```
// create a script that will create the necessary directory
delete __appendfile
appendfile #!/bin/sh
appendfile mkdir –p ../../scm_preserve/SunOS/5.10
delete createdir.sh
move __appendfile createdir.sh
wait /bin/sh ./createdir.sh

// create the file containing the checks that you wish to run
delete __appendfile
appendfile GEN000020
appendfile GEN000480
appendfile GEN000560
delete ../../scm_preserve/SunOS/5.10/daily.txt
move __appendfile ../../scm_preserve/SunOS/5.10/daily.txt
```

4. Run the runme.sh script with the –F option. To do this, modify the *Deploy and Run Security Checklist Solaris 10* task. Find this task and double-click it, then select endpoints in the Take Action dialog.



5. Under the *Action Script tab,* modify the Action Script to make runme.sh use the –F option and point to the file that contains the check list (which was named daily.txt).

   Below is a sample script that you can copy, paste, and modify:

```
prefetch DISA.zip sha1:99c90759cc496c506222db55bd864eba4063b955 size:108089
http://software.bigfix.com/download/SCM/SunOS-20080417.zip
delete __appendfile
delete run_SCM.sh
appendfile #!/bin/sh
if {exists folder ((pathname of parent folder of parent folder of folder
(pathname of client folder of current site)) & "/SCM")}
 appendfile rm -rf {((pathname of parent folder of parent folder of folder
(pathname of client folder of current site)) & "/SCM")}
endif
appendfile mv __Download/DISA.zip {((pathname of parent folder of parent
folder of folder (pathname of client folder of current site)))}
appendfile cd {((pathname of parent folder of parent folder of folder
(pathname of client folder of current site)))}
appendfile gzip -dvS .zip DISA.zip
appendfile FILE=`ls -1 DISA* | grep -v zip`
appendfile tar xf $FILE
appendfile rm -rf $FILE
appendfile cd {((pathname of parent folder of parent folder of folder
(pathname of client folder of current site)) & "/SCM")}
appendfile ./runme.sh -F ../scm_preserve/SunOS/5.10/daily.txt
move __appendfile run_SCM.sh
wait sh ./run_SCM.sh
```

Analyses

Each check Fixlet in the DISA UNIX content has an associated analysis. Check Fixlets display the compliance state, and analyses display the state of each configuration item.

These analyses enable the display of "Measured Values" in Tivoli Endpoint Manager Security and Compliance Analytics. If you are using only a subset of the available check Fixlets for your implementation, activate only the analyses associated with the check Fixlets you are using. Each UNIX content Fixlet contains a link to the related analysis.

Part Three

Support

Technical support

The Tivoli Endpoint Manager technical support site offers a number of specialized support options to help you learn, understand, and optimize your use of this product:

- Tivoli Endpoint Manager Info Center
- BigFix Support Site
- Documentation
- Knowledge Base
- Forums and Communities

Part Four

Notices

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
1623-14, Shimotsuruma, Yamato-shi
Kanagawa 242-8502 Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
2Z4A/101
11400 Burnet Road
Austin, TX 78758 U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

TRADEMARKS:

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both.

If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at http://www.ibm.com/legal/copytrade.shtml.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.