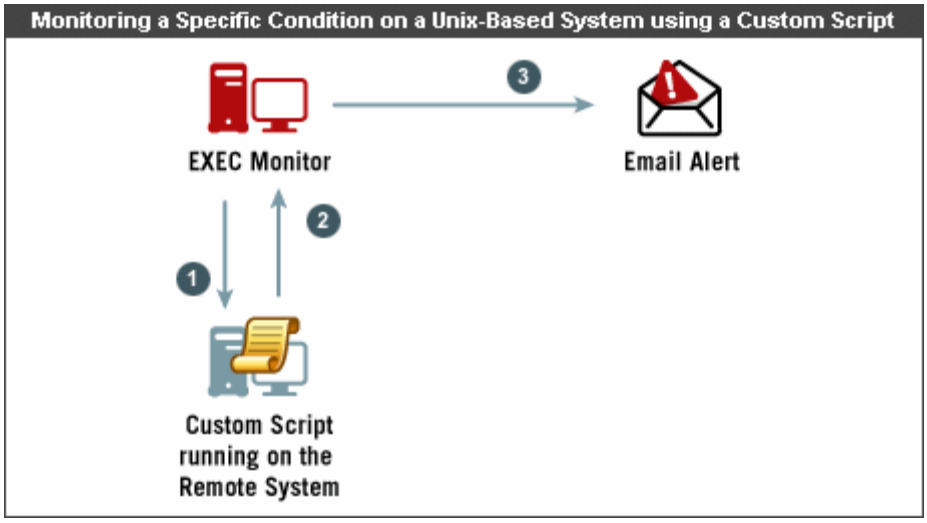
HOW TO MONITOR A SPECIFIC CONDITION ON A UNIX-BASED SYSTEM USING A CUSTOM SCRIPT AND SSH

Many Unix-based systems utilize custom shell scripts in order to monitor a specific condition on the server. The majority of these scripts have been configured by a Unix system administrator to run through a Cron job or a manual process. Using ipMonitor's EXEC Monitor and a utility called "plink", it is possible to have ipMonitor establish a Secure Shell (SSH) session with the remote server and run a script. An ipMonitor Alert can then be triggered if the script does not complete or exit properly.

This setup is useful when:

- A custom script already exists on the remote host but Alerting capabilities are missing or lacking functionality.
- SNMP connectivity to the system is not available.
- Connections to the remote system must be made through a Secure Shell.

How the Process Works



The above diagram shows that:

1. The EXEC Monitor uses the **plink.exe** utility to establish an SSH connection to the remote host and run a script.
2. The custom script returns an exit code to ipMonitor.
3. Based on the script completion or the exit code returned by the script, an Email Alert is sent (if configured).

Before you begin

Step 1 – Download the "plink" utility:

Plink (PuTTY Link) is a command-line utility that provides a Unix SSH connection:

1. Log into the ipMonitor host machine using either:
   A. The windows Account assigned to the ipMonitor Service.
   B. The Account that the EXEC Monitor will impersonate via a Credential.

Note: This process is necessary to ensure that the communication is tested using the same user account that will be used by ipMonitor and to allow the rsa2 key to be stored in cache and be available for use by the account used by ipMonitor.

2. Download the Plink utility from the following location:

   http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html

3. Save the plink.exe utility to the .\system32 folder on the ipMonitor host system.

Step 2 – Test SSH connectivity to the remote system:

1. Open a command prompt and enter the following command:

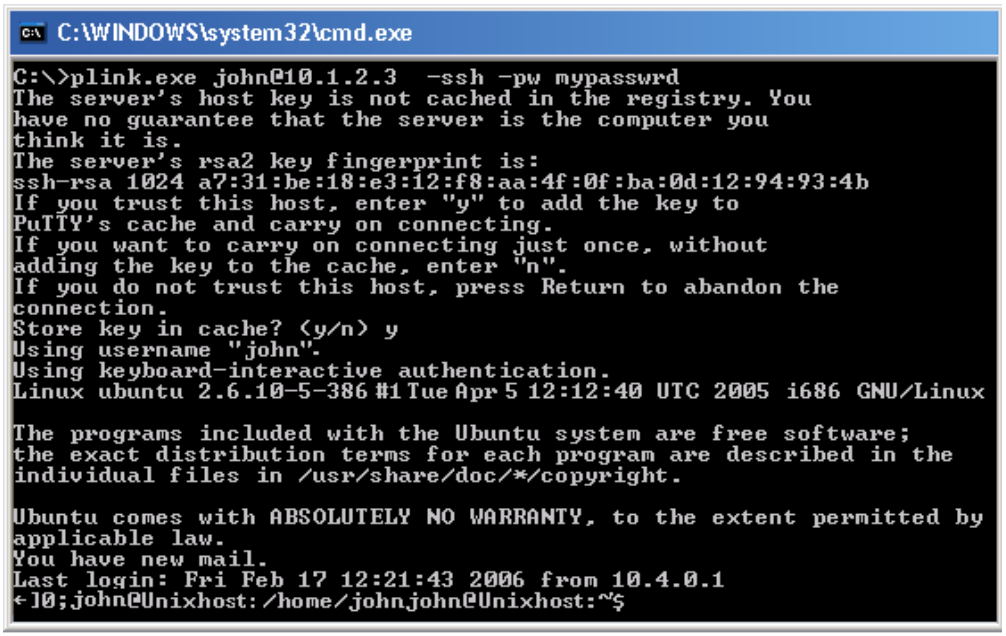
   Plink.exe *username@ip address* -ssh –pw *password*

   Note:

   Username: the username of an account with SSH access on the remote system.
   ip address: the IP address of the remote system.
   password: the password for the account used.

   For example: plink.exe john@10.1.2.3 -ssh –pw mypasswrd

   The remote system will present you with an rsa2 key fingerprint. Enter "y " when prompted to store the key in cache.



If the connection is successful, you will receive a response resembling the following:

   ←]0;john@Unixhost: /home/johnjohn@Unixhost:~$

2. Keep the Plink connection open, and proceed to Step 3.

Step 3 - Test the script:

1. Using the Plink connection established in Step 2, enter the path to the custom script and attempt to run it.

> **TIP**
>
> To ensure the script ran as intended, we suggest adding a custom line inside the script instructing it to dynamically write content to a file.

2. If the script ran as intended, you can now test the entire process from a command line. Close the Plink session and run the following command from the command prompt:

   Plink.exe *username@ip address* -ssh –pw *password script*

   Script: the path and name of the script.

   For example: plink.exe john@10.1.2.3 -ssh –pw mypasswrd ./script/drive_space.sh

If this test is also successful, proceed to the Configuring the EXEC Monitor section of this tutorial.

Top

AVAILABLE RESOURCES

Click the XML icon to download a preconfigured Monitor to be used with this tutorial.

Sample Script

This script monitors the percentage of a partition; it is set to fail (causing the EXEC Monitor to fail) if the partition is more than 70% used.

Bash script:

```
#!/bin/bash
#
# Getting the percentage of use of hda1, stripping out the "%" symbol and inserting the result into a variable named VDpercentage.

VDpercentage=`df -k | grep /dev/hda1 | awk '{print $5}' | tr -d "%"`

# Setting the threshold.

Thresh=70

# Comparison of the value retrieved with a threshold of 70% utilization and setting the exit code appropriately.

if [ "$VDpercentage" -gt "$Thresh" ];
   then
    exit 0
else
    exit 1
fi
```

Top

SAMPLE MONITOR SETTINGS

Monitor Name	**EXEC :: Server Name :: Script name**
Monitor Type	EXEC (Third Party) Monitor
Executable Name	plink.exe
Directory	c:\windows\system32
Command Line Parameters	username@ip address -ssh -pw password ./script/drive_space.sh
Startup Directory	c:\
Credential for Monitoring	Credential name
Returns	Returns the exit code generated by custom script
Monitoring Condition	**Expected Return Value: 0** Monitor failure indicates that the drive space is more then the set threshold or that the script failed to complete.

Note: The imported Monitor is initially disabled. This allows you to make changes to the default settings before the Monitor is enabled to go live in a production environment. Once the Monitor is imported, the following settings will need to be verified (and if necessary, modified) for the Monitor:

- Directory
- Command Line Parameters
- Startup Directory
- Credential for Monitoring

To learn more about modifying the above parameters, please refer to the Configuring the EXEC Monitor section of this tutorial.

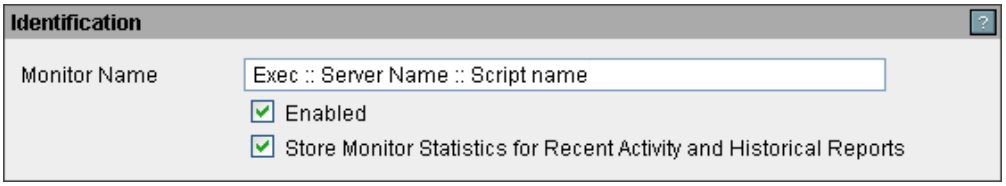
> **TIP**
>
> For detailed instructions regarding importing XML files into your ipMonitor installation, please refer to the tutorial entitled "How to Import and Export Configuration Settings".

Top

CONFIGURING THE EXEC MONITOR

For the purposes of this tutorial, we'll be referring to the preconfigured Monitor included in the XML resource download. Create a new Monitor by clicking the **Monitors** menu option and then selecting **Add a Monitor**. Choose the **EXEC Monitor** from the list provided.

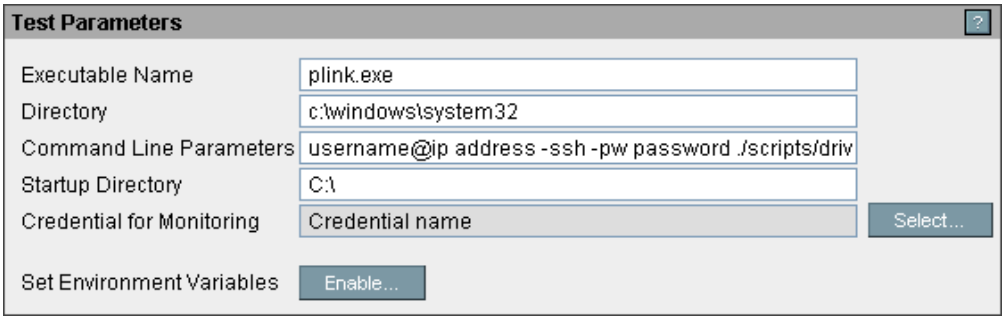
Identification



1. Enter a unique, descriptive name for the Monitor in the **Name** field.
2. Check the **Enabled** checkbox.

3. Choose whether you want the Monitor to **Store Monitor Statistics for Recent Activity and Historical Reports**.

Test Parameters



4. Enter the **Executable Name**, in this case, **plink.exe**.

5 Enter the path to the location of plink.exe in the **Directory** field.

> **TIP**
>
> - For Windows XP and 2003 systems, use: **c:\windows\system32**
> - For a Windows 2000 system, use **c:\winnt\system32**

6. Enter the **Command Line Parameters** required.

Example:

username@ip address -ssh -pw password ./scripts/ drive_space.sh

Note:

Username: the username of an account with SSH access on the remote system.
ip address: the IP address of the remote system.
password: the password for the account used.
Script: the path and name of the script.

In this example, the drive_space.sh script resides in the users directory, under **/script/**.

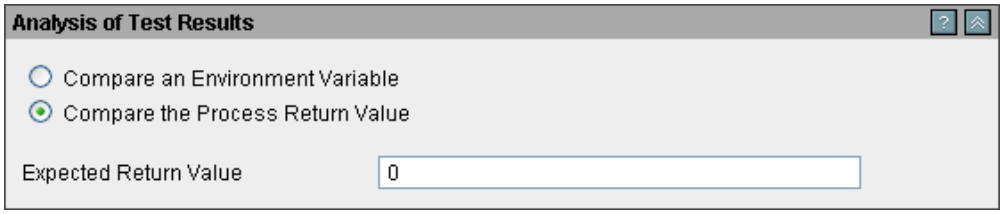
plink.exe john@10.1.2.3 -ssh –pw mypasswrd ./script/ drive_space.sh

7. Enter the **Startup Directory**. In this case, **C:**\.

8. Assign or create a **Credential** to use with the EXEC Monitor.

Note: The EXEC Monitor must use an account with Administrator privileges on the ipMonitor host system in order to be able to run the plink.exe utility.

Analysis of Test Results



1. Configure the EXEC Monitor to **Compare the Process Return Value**.
2. The expected result is determined to be **0**, indicating that there were no errors encountered when running the script, or that the result returned is within the expected value.

> **TIP**
>
> By default, the Timing intervals for each parameter are set to 300 seconds. If needed, adjust these defaults to meet your specific monitoring environment's requirements until the optimum settings are reached.

Top

ADDITIONAL INFORMATION

For more information regarding the EXEC Monitor, refer to the Monitors section of ipMonitor's Context-Sensitive Help system, than select **Monitor Types**, followed by **EXEC**. The Context-Sensitive Help can be accessed by clicking the **Help** link located in the top-right corner of ipMonitor's Administration web interface.