Table of Contents

Sample Keepalive Script to Check a List of Interfaces That Users Pass on the Command Line

Document ID: 47385

Introduction

This script is designed to check a list of interfaces that the user passes on the command line. If any one of these links fails, the service will be declared down. This can be used as a critical service with VIP/interface redundancy to provide physical link monitoring capability. This document also addresses implementation of scripted keepalives. This method of scripting is most closely related to functionality, which is present in Remote Access Server (RAS) dialup clients, terminal programs, and general scripting utilities. This feature utilizes WebNS's rich scripting language.

Complete with a simple socket Application Program Interface (API) (connect/disconnect/send/receive), a scripted keepalive will give the user the ability to tailor their own protocol, or write their own sequence of steps to provide a reliable `ALIVE` or `DOWN` state of a service. Without the scripted keepalive functionality, you are currently limited to FTP, HTTP, ICMP, and TCP. With scripted keepalives, however, you can remain on top of the current protocols by writing your own scripts. For example, you can develop a script specifically toned to connect to a POP3 server without requiring WebNS to build a keepalive type POP3. This feature allows customers to create their own custom keepalives to suit their specific requirements. Although this is a component of the Content Services Switch (CSS), custom scripts are not supported by the Cisco Technical Assistance Center (Cisco TAC).

The scripted keepalives below are not officially supported by TAC, but have been tested, and are available for use at your own discretion.

Prerequisites

Requirements

Familiarity with WebNS rich scripting language.

Components Used

The information in this document is based on the software and hardware versions:

- WebNS versions 3.x and higher
- CSS 11x00 Series

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, make sure

that you understand the potential impact of any command.

Sample Script

The script below can be used to check a list of interfaces that users pass on the command line.

```
!--- No echo.
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!--- Filename: ap-kal-phy-check
!--- Parameters: Phy1, Phy2, Phy3, and so on. For CSS 11x50, use e1, e2, e3,
!--- and so on for the syntax. For CSS 11800, use slot/port 1/1, 1/2, 1/3,
!--- and so on for the syntax.

!--- Description:
!--- This script is designed to check a list of interfaces that the user
!--- passes on the command line. If any one of these links fails, the
!--- service will be declared down. This can be used as a critical service
!--- with VIP/interface redundancy to provide physical link monitoring capability.
!
!--- Failure Upon:
!--- 1. Any interface in the down state.
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!


if ${ARGS}[#] "LT" "1"
echo "Usage: ap-kal-phy-check \?phy1 phy2 phy3 ...\?"
  exit script 1
endbranch

while ${ARGS}[#] "GT" "0"
  set Host "${ARGS}[1]"
  var-shift ARGS
  function Phycheck call "${Host}"
endbranch

no set EXIT_MSG
exit script 0

function Phycheck begin

!--- Check the first physical.

show phy ${ARGS}[1] | grep -u Down
if STATUS "NEQ" "0"
exit script 1

endbranch

function Phycheck end
```

Related Information

- **CSS 11000 Series Content Services Switches Hardware Support**
- **CSS 11500 Series Content Services Switches Hardware Support**
- **Software Download for CSS11500**

Updated: Jan 20, 2006 Document ID: 47385