

# Script do Keepalive da amostra para verificar o página da web para ver se há o Webstring

## Índice

[Introdução](#)

[Pré-requisitos](#)

[Requisitos](#)

[Componentes Utilizados](#)

[Exemplo de script](#)

[Informações Relacionadas](#)

## Introdução

Este script verifica o página da web para ver se há a corda da Web. Se a corda falta, marque o serviço como para baixo. Este script é usado com meio a página, as especialmente dinâmicas que são geradas através dos scripts, ColdFusion, e assim por diante. Este documento igualmente endereça a aplicação dos keepalive de script. Este método do script é o mais estreitamente relacionado à funcionalidade, que esta presente em clientes dialup do servidor de acesso remoto (RAS), em programas terminal, e em utilidades gerais do script. Esta característica utiliza o linguagem de script rico de WebNS.

Termine com um Application Program Interface do soquete simples (API) (conecte/disconexão/enviam/recebem), um keepalive de script dará ao usuário a capacidade para costurar seu próprio protocolo, ou escreva sua própria sequência das etapas para fornecer um `VIVO` ou um `estado inativo` seguro de um serviço. Sem a funcionalidade do keepalive de script, você é limitado atualmente ao FTP, ao HTTP, ao ICMP, e ao TCP. Com keepalive de script, contudo, você pode permanecer sobre os protocolos atual escrevendo seus próprios scripts. Por exemplo, você pode desenvolver um script tonificado especificamente para conectar a um server POP3 sem exigir WebNS construir um tipo keepalive POP3. Esta característica permite que os clientes criem seu próprio Keepalives feito sob encomenda para serir suas exigências específicas. Embora este seja um componente do Content Services Switch (CSS), os scripts personalizados não são apoiados pelo centro de assistência técnica da Cisco (tac Cisco).

Os keepalive de script abaixo não são apoiados oficialmente pelo TAC, mas foram testados, e estão disponíveis para o uso em sua própria discreção.

## Pré-requisitos

### Requisitos

Familiaridade com o linguagem de script dos ricos de WebNS.

## Componentes Utilizados

As informações neste documento são baseadas nas versões de software e hardware:

- WebNS versões 3.x e superiores
- 11x00 Series CSS

As informações neste documento foram criadas a partir de dispositivos em um ambiente de laboratório específico. Todos os dispositivos utilizados neste documento foram iniciados com uma configuração (padrão) inicial. Se a sua rede estiver ativa, certifique-se de que entende o impacto potencial de qualquer comando.

## Exemplo de script

O script abaixo pode ser usado para verificar o página da web para ver se há webstring.

```
!--- No echo. !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! !--- Filename:
ap-kal-httpstring !--- Parameters: WebsiteIP WebPage WebString [Port] !--- Requirements:
WebNS4.x or higher ! !--- Uses: !--- Checks the Web page for the Web string. If the string is
missing, !--- mark the service as down. Used with any sort of page, especially dynamic !--- ones
that are generated via scripts, ColdFusion, and so on. ! !--- Logic: !--- The script connects to
a Web server on port 80 by default. !--- It performs a GET on the specified page. !--- If the
Web string is returned, the service stays up. !--- If anything fails, the service is marked
down. ! !--- Notes: !--- The Web string is case-sensitive. !--- Only the first 10Kb of the
response is inspected. ! ! !--- Tested: 04/12/01-KGS !
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! no set CONTINUE_ON_ERROR no
set EXIT_MSG !--- Make sure the user has the proper number of arguments. if ${ARGS}[#] "LT" "3"
echo "Usage: ap-kal-httpcheck \'WebsiteIP WebPage WebString [Port]\'" exit script 1 endbranch !-
-- Set variables corresponding to the args. set WebSite "${ARGS}[1]" set WebPage "${ARGS}[2]"
set WebString "${ARGS}[3]" set WebPort "80" if ${ARGS}[#] "GT" "3" set WebPort "${ARGS}[4]"
endbranch echo "Requesting ${WebPage} from ${WebSite} on port ${WebPort}." !--- Connect to the
remote server. set EXIT_MSG "Connect: Failed. Could not connect to ${WebSite} on port
${WebPort}" set CONTINUE_ON_ERROR "1" socket connect host ${WebSite} port ${WebPort} tcp if
${STATUS} "NEQ" "0" exit script 1 endbranch no set CONTINUE_ON_ERROR !--- Request the desired
Web page. set EXIT_MSG "Send: Failed. Could not send to ${WebSite}:${WebPort}" socket send
${SOCKET} "GET ${WebPage} HTTP/1.0\n\nHost: ${WebSite}:${WebPort}\n" !--- Look for the Web
string. set EXIT_MSG "Waitfor: Failed. Did not find [${WebString}]" set CONTINUE_ON_ERROR "1"
socket waitfor ${SOCKET} "${WebString}" case-sensitive if ${STATUS} "NEQ" "0" exit script 1
endbranch no set CONTINUE_ON_ERROR !--- Disconnect from the server. no set EXIT_MSG socket
disconnect ${SOCKET} graceful exit script 0
```

## Informações Relacionadas

- [Suporte a hardware dos CSS 11000 Series Content Services Switch](#)
- [Suporte a hardware dos CSS 11500 Series Content Services Switch](#)
- [Download do software para CSS11500 \(clientes registrados somente\)](#)
- [Suporte Técnico e Documentação - Cisco Systems](#)