

# Пример сценария проверки доступности веб-страниц для Webstring

## Содержание

[Введение](#)

[Предварительные условия](#)

[Требования](#)

[Используемые компоненты](#)

[Образец сценария](#)

[Дополнительные сведения](#)

## **Введение**

Этот сценарий проверяет веб-страницу для веб-строки. Если строка отсутствует, отметьте сервис как вниз. Этот сценарий используется с любым видом страницы, особенно динамические, которые генерируются с помощью сценариев, ColdFusion, и так далее. Этот документ также обращается к реализации заданных сценарием средств поддержки активности. Этот метод сценариев является самым тесно связанным с функциональностью, которая присутствует в клиентах удаленного доступа сервера удаленного доступа (RAS), программах терминала и общих утилитах сценариев. Эта функция использует богатый язык сценария WebNS.

Вместе с простым разъемом (соединяется/разъединяет/передает/получает) Прикладной программный интерфейс (API), заданное сценарием средство поддержки активности даст пользователю способность адаптировать их собственный протокол или записать их собственную последовательность шагов для обеспечения надежного `ALIVE` или состояния `DOWN` сервиса. Без функциональности заданного сценарием средства поддержки активности вы в настоящее время ограничиваетесь FTP, HTTP, ICMP и TCP. С заданными сценарием средствами поддержки активности, однако, можно остаться поверх текущих протоколов путем записи собственных сценариев. Например, можно разработать сценарий, в частности настроенный для соединения с сервером POP3, не требуя, чтобы WebNS создал POP3 типа проверки активности. Эта функция позволяет клиентам создавать свои собственные пакеты Keeralive для удовлетворения их определенным требованиям. Несмотря на то, что это - компонент Коммутатора контент-сервисов (CSS), настраиваемые сценарии не поддерживаются Центром технической поддержки Cisco (Центр технической поддержки Cisco).

Заданные сценарием средства поддержки активности ниже официально не поддерживаются TAC, но были протестированы и доступны для использования по вашему собственному усмотрению.

## **Предварительные условия**

## Требования

Знакомство с WebNS богатый язык сценария.

## Используемые компоненты

Сведения, содержащиеся в этом документе, касаются следующих версий программного обеспечения и оборудования:

- WebNS: версии 3.x и выше
- CSS 11x00 серия

Сведения, представленные в этом документе, были получены от устройств, работающих в специальной лабораторной среде. Все устройства, описанные в этом документе, были запущены с чистой (стандартной) конфигурацией. В рабочей сети необходимо изучить потенциальное воздействие всех команд до их использования.

## Образец сценария

Сценарий ниже может использоваться для проверки веб-страницы для 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
```

## Дополнительные сведения

- [Аппаратная поддержка коммутаторов контент-сервисов CSS 11000](#)
- [Аппаратная поддержка коммутаторов контент-сервисов CSS 11500](#)
- [Загрузка программного обеспечения для CSS11500 только для зарегистрированных пользователей\)](#)
- [Cisco Systems – техническая поддержка и документация](#)