VIRTUAL ACCESS

SMART NETWORK SOLUTIONS
FOR SERVICE PROVIDERS

watchdogRecovery: Dynamically initiating watchdog recovery procedure.

Issue: 1.1

Date: 31 October 2014

© Virtual Access 2014

1 Introduction

The watchdogRecovery script is designed to start the standard watchdog recovery procedure. It is intended to be started and stopped by custom scripts where a standard watchdog script does not suit the environment. It does no checking but simply runs through the recovery procedure.

The watchdogRecovery script is commonly used in a scenario shown in the figure below. The WAN connection is in the example is via a bridged ADSL connection with a mobile backup connection. The pingFailover script generally is used for failover from bridged ADSL to mobile backup in this environment. Due to the ADSL connection being bridged there is no standard watchdog script that can be used at boot for watchdog recovery of the primary interface. In this case the watchdogRecovery script can be dynamically started by a custom script that is waiting for the failover from ADSL to GSM. If the ADSL link comes back, either on its own, or by one of the recovery methods, then the watchdogRecovery script should be stopped.

An example of a custom script is shown in section 2.3.2.

Note: this script is designed for a single WAN interface. It will only reset one physical interface. It can be used in a multi-WAN scenario, such as ADSL with mobile backup, but if you do this the script must be stopped and started dynamically on routing switchover.

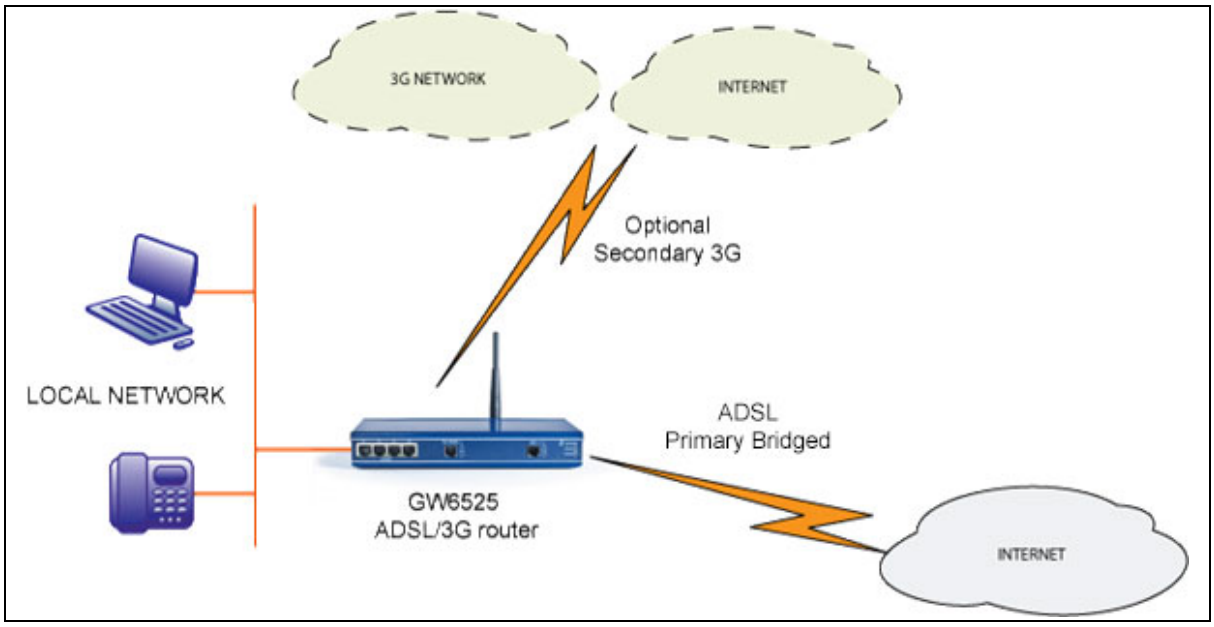

Figure 1: Possible network architecture

2 Configuring the watchdogRecovery script

2.1 Script overview

The script can be run multiple times. It is designed to be started and stopped dynamically from a custom script. On initiation, the script does the following:

- Waits a configurable duration before attempted recovery of the desired interface. For adsl-x and modem-x interfaces a reset is executed on the interface. For eth-x interfaces the router will release and renew the IP address.

Note: Do not run this script directly from boot.

2.2 Script parameters

The script name is **watchdogRecovery** and it takes in three required parameters:

watchdogRecovery [phy port] [reload] [wait]

These parameters are described in the example and table below.

watchdogRecovery adsl-0, 1, 300

Parameter	Type	Description
adsl-0	Required	The physical interface to reset when pings fail. Available interfaces are adsl-x; modem-x, eth-x.
1	Required	Whether to reload after reset of the physical interface (default: 1; 0 for no reload)
300	Required	The wait before resetting the physical interface and also the wait after resetting the physical interface before reload of the router.

Table 1: watchdogRecovery parameter descriptions

2.3 Configuring the script

This script was introduced into firmware in versions 9.09.29 and 10.00.25. To use the script on older firmware versions first paste the script from Section 5 ‘watchdogRecovery script’ into the script editor and then use the scheduler to run the script at boot up.

From the start page, click **Advanced** to open the Expert View menu.

2.3.1 Pasting the script into the script editor

If you are using 9.09.xx firmware, in the Expert View menu, click **system > scripts->script editor**. The Script Editor page appears.

© Virtual Access 2014

If you are using 10.00.xx firmware, in the Expert View menu, click **system > management > scripts > script editor.** The Script Editor page appears.

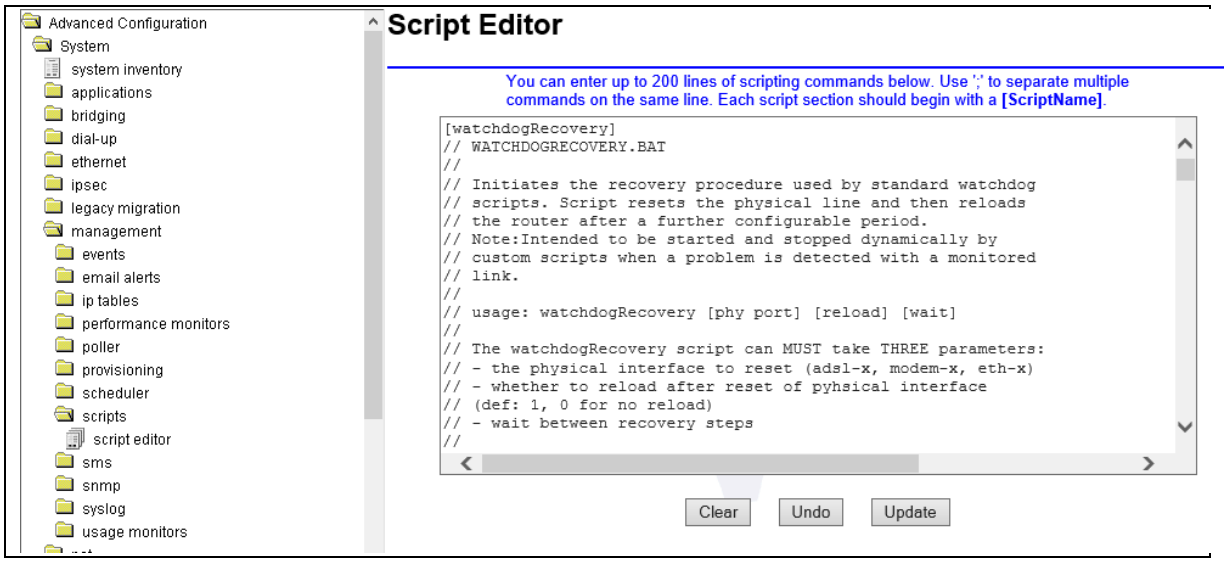

Figure 2: The script editor page in version 9.09.xx

Paste in the script from Section5 'watchdogRecovery script' from this document. The first line of the script should begin with the script name in square brackets, [watchdogRecovery]. This name will be used to call the script using the scheduler.

If the number of script lines needs to be reduced, you can omit any line beginning with '//'as this denotes a comment tag. Also, you can enter multiple script lines onto the same script editor line separated by ';' (semi colon).

When you have completed the script, click **Update**.

2.3.2 Scheduling the script to run from a custom script

2.3.2.1 Example custom script overview

The watchdogRecovery script is designed to be started and stopped dynamically from a custom script. An example environment where this can be used is a bridged ADSL primary link with a mobile backup link. In this scenario failover is generally monitored and actioned by the pingFailover script. The custom script waits for failover to the backup interface, dynamically initiates the watchdog recovery procedure and then stops the watchdog recovery on fall back to the primary interface. A decision that needs to be made when there is a backup interface is whether to reload the router if it does not recover to the primary interface. On a genuine primary interface outage this could mean that the router is continually being reset. The watchdogRecovery script can optionally reload to accommodate.

© Virtual Access 2014

2.3.2.2 Example custom script

The example custom script name is pingFailoverRecovery and it takes in five required parameters:

```
pingFailoverRecovery   [watchdog script name] [watchdog phy port] [watchdog logical if]
                       [watchdog wait] [watchdog reload]
```

These parameters are described in the example and table below.

pingFailoverRecovery watchdogRecovery, adsl-0, ppp-1, 300, 1

Parameter	Type	Description
watchdogRecovery	Required	The watchdog script to be called dynamically
adsl-0	Required	The physical interface to reset when pings fail. Available interfaces are adsl-x; modem-x, eth-x.
bridge-1	Required	The logical primary interface
300	Required	The wait before resetting the physical interface and also the wait after resetting the physical interface before reload of the router.
1	Required	Whether to reload after reset of the physical interface (default: 1; 0 for no reload)

Table 2: pingFailoverRecovery parameter descriptions

© Virtual Access 2014

```
   *** EXAMPLE SCRIPT TO DYNAMICALLY START AND STOP WATCHDOGRECOVERY ***
[pingFailoverRecovery]
//
// Waits for primary route down as detected by pingFailover script.
// Once the primary route is detected as down a watchdog script is enabled
// to attempt recovery of primary interface. When primary route is detected
// as available again the watchdog script is stopped.
//
// usage: pingFailoverRecovery [watchdog name] [watchdog phy port]
//                             [watchdog logical if] [watchdog wait]
//                             [watchdog reload]
//
// The pingFailoverReovery script MUST take FOUR parameters:
// - the watchdog script name
// - the primary interface for the watchdog to reset (0 for no reset)
// - the watchdog logical interface (eg. ppp-1)
// - the wait between watchdog recovery steps
// - whether to reload as a last resort if all recovery steps fail
//   (0 for no reload)
//
// CONFIGURATION
// -------------
// This script MUST be used in conjunction with pingFailover.bat
// that is embedded in the firmware.
// Both pingFailover and pingFailoverRecovery MUST be run on boot.
//
// EXAMPLES
// --------
// pingFailoverRecovery watchdogRecovery, adsl-0, ppp-1, 300, 1
// (On primary route down enable watchdog recovery script. The
//  recovery script will wait 300 seconds before resetting modem-1.
//  If recovery script not stopped then it will reload the router
//  after a further 300 seconds).

!echo off
!arg recoveryScript, recoveryPhyIf, recoveryLogicalIf, recoveryWait,
recoveryReload
```

© Virtual Access 2014

```
//logging
!if $recoveryReload > 0
  $evt_reload = yes
!else
  $evt_reload = no
!endif

!log pingFailoverRecovery running script $recoveryScript on $recoveryPhyIf
reload $evt_reload wait $recoveryWait

!while 1

  !waitevent script.40:pingFailover_primary_default_route_down
  !endevent

  !label PRIMARY_DOWN
    !log pingFailoverRecovery starting $recoveryScript
    $z = `$recoveryScript $recoveryPhyIf, $recoveryReload, $recoveryWait`

    !waitevent script.40:pingFailover_primary_default_route_up
    !endevent

  !label PRIMARY_UP
    !log pingFailoverRecovery stopping $recoveryScript
      $z = `kill $recoveryScript`

!endwhile
```

© Virtual Access 2014

3 Debugging commands

Useful debug commands via the command line are described in the table below.

Diagnostic Command	Description
Show tasks	Displays all running tasks.
Show task <tasknum>	Displays running task. Also indicates position task is currently at.
Show task vars <tasknum>	Displays variables and variable values associated with task.
Show ip route	Shows routing table.
Show stats adsl	Shows ADSL stats.
Show modem interface status modem-x	Displays GSM stats.
show modem interface gsm sim status modem-x	Displays GSM SIM status.
Show dhcp client info [eth-x | waneth-x]	Displays DHCP client information
Show events	Displays event log.
Show change log	Displays recent configuration changes.
Dir scripts	Displays all scripts embedded in the firmware.
Show config script ALL	Displays all scripts in the script editor.
Show config script <scriptname>	Displays the <scriptname> script as configured in script editor. Includes line numbers.
Show config script –n <scriptname>	Displays the <scriptname> script as configured in the script editor. Does not include line numbers.

Table 3: Debug command lines and their descriptions

Useful trace commands via the command line are described in the table below.

Trace command	Description
++All 6	Traces all INFO events
++ip	Traces IP traffic
++ip:icmp	Traces ICMP IP traffic
++modem	Traces modem events
++script	Traces script events
--script	Stops script event tracing
--	Stops all event tracing
Trace on <script_name>	Traces each line in a script as it executes
Trace off <script_name>	Turns off tracing for script

Table 4: Trace command lines and their descriptions

© Virtual Access 2014

4 Script events

Severity	Class	Subclass	Text
INFO	49	40	watchdogRecovery error invalid phy port <phy port>
INFO	49	40	watchdogRecovery running phy port <phy port> reload <yes|no> wait <wait in secs>
INFO	49	40	watchdogRecovery rebooting
INFO	49	40	watchdogRecovery resetting <phy port>
DEBUG	49	9999	watchdogRecovery resetting <phy port> (only generated after INFO event above)

Table 5: Script events

© Virtual Access 2014

5 watchdogRecovery script

```
[watchdogRecovery]
// WATCHDOGRECOVERY.BAT
//
// Initiates the recovery procedure used by standard watchdog
// scripts. Script resets the physical line and then reloads
// the router after a further configurable period.
// Note:Intended to be started and stopped dynamically by
// custom scripts when a problem is detected with a monitored
// link.
//
// usage: watchdogRecovery [phy port] [reload] [wait]
//
// The watchdogRecovery script can MUST take THREE parameters:
// - the physical interface to reset (adsl-x, modem-x, eth-x)
// - whether to reload after reset of pyhsical interface
//   (def: 1, 0 for no reload)
// - wait between recovery steps
//
// EXAMPLES
// --------
// watchdogRecovery modem-0, 1, 300
// - waits 300 seconds before resetting modem-0. After a further
//   300 seconds reloads router.

!echo off

!arg phyPort, routerReload, recoveryWait

//checks
!if $phyPort <> "*adsl-"
  !if $phyPort <> "*modem-"
    !if $phyPort <> "*eth-"
      !log watchdogRecovery error invalid phy port $phyPort
      !exit
```

© Virtual Access 2014

```
    !endif
  !endif
!endif

$phy_reset = 0
$restart_checks_event = 0

//logging
!if routerReload > 0
  $evt_reload = yes
!else
  $evt_reload = no
!endif

!log watchdogRecovery running phy port $phyPort reload $evt_reload wait
$recoveryWait

!while 1

  !pause $recoveryWait

  !if $phy_reset <> 0
    !if $routerReload <> 0
      !log watchdogRecovery rebooting
      !pause 2
      reload
    !else
      $phy_reset = 0
    !endif

  !else
      //only generate reset event as INFO once until ping success so do not
fill event log
      !if $restart_checks_event <> 0
        !event script.9999 watchdogRecovery resetting $phyPort
      !else
        !log watchdogRecovery resetting $phyPort
```

© Virtual Access 2014

```
        $restart_checks_event = 1
      !endif

    //ADSL reset
    !if $phyPort = "*adsl-"
       $z = `reset adsl $phyPort `
       !pause 90
    !endif

    //3G modem reset
    !if $phyPort = "*modem-"
        $z = `reset modem $phyPort `
        !pause 30
    !endif

    //DHCP release and renew
    !if $phyPort = "*eth-"
      $z = `dhcp release $phyPort
      !pause 2
      $z = `dhcp renew $phyPort
      !pause 10
    !endif

    $phy_reset = 1
  !endif

!endwhile
```

© Virtual Access 2014

6 Script history

Version	Changes	Firmware version changes introduced
1.0	Document created	9.08.29 10.00.25