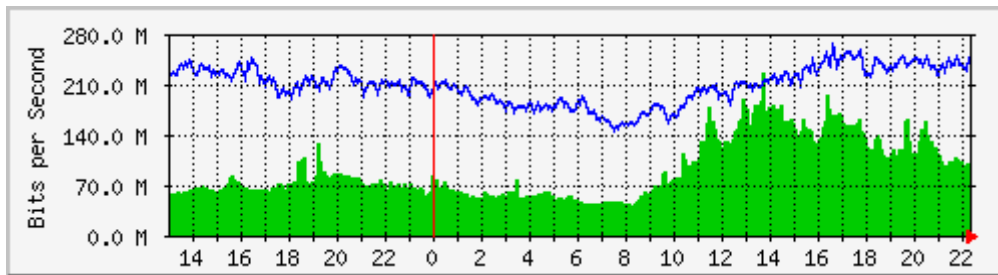




## ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ



## ΔΙΑΓΡΑΜΜΑΤΑ ΚΙΝΗΣΗΣ ΔΙΚΤΥΟΥ

**ΣΠΟΥΔΑΣΤΕΣ : ΓΕΩΡΓΙΑΔΗΣ ΘΕΟΔΩΡΟΣ  
ΚΑΛΑΣ ΦΙΛΙΠΠΟΣ**

ΑΡΤΑ 2005

# ΔΙΑΓΡΑΜΜΑΤΑ ΚΙΝΗΣΗΣ ΔΙΚΤΥΟΥ

## MRTG

### ΙΣΤΟΡΙΑ

(Από τον δημιουργό)

Το 1994 εργαζόμουν σε ένα site όπου είχαμε μια γραμμή 64kbit. Προφανώς όλοι ενδιαφέρονταν να μάθουν πως συμπεριφερόταν το link . Έτσι έγραψα μια γρήγορη εφαρμογή που δημιούργησε μια συνεχώς ενημερωμένη γραφική παράσταση στον Ιστό που παρουσίαζε την κίνηση στο link. Αυτό εξελίχθηκε τελικά σε ένα μάλλον διαμορφώσιμο χειρόγραφο Perl αποκαλούμενο MRTG - 1,0 που απελευθέρωσα την άνοιξη του 1995. Μετά από μερικές αναπροσαρμογές, άφησα την εργασία μου στο DMU για να αρχίσω την εργασία στο ελβετικό ομοσπονδιακό ίδρυμα τεχνολογίας. Λόγω έλλειψης χρόνου έπρεπε να βάλω το MRTG κατά μέρος. Μια ημέρα τον Ιανουάριο του 1996, έλαβα e-mail από τον Dave Rand ρωτώντας εάν είχα ιδέα γιατί το MRTG ήταν τόσο αργό. Πραγματικά είχα . Ο προγραμματισμός του δεν ήταν πολύ αποδοτικός και γράφτηκε εξ ολοκλήρου σε Perl. Μετά από μια εβδομάδα ή και περισσότερο , ο Dave μου έγραψε ότι είχε δοκιμάσει αυτό που του είχα προτείνει για τη βελτίωση της ταχύτητας του MRTG. Δεδομένου ότι οι αλλαγές δεν βοήθησαν πολύ, αποφάσισε να ξαναγράψει το script του MRTG σε C. Ο κώδικας συνδέθηκε με το ηλεκτρονικό ταχυδρομείο του. Το εργαλείο του αύξησε την ταχύτητα του MRTG στο 40πλάσιο! Αυτό με πήρε από "την άγνοια του MRTG μου" και άρχισα να ξοδεύω χρόνο για να αναπτυχθεί το MRTG-2.

Σύντομα αφότου είχε αρχίσει η ανάπτυξη του MRTG-2 άρχισα να δίνω τα δοκιμαστικά αντίγραφα στα ενδιαφερόμενα συμβαλλόμενα μέρη. Σε αντάλλαγμα πήρα πολλά μπαλώματα χαρακτηριστικών γνωρισμάτων, πολλές ανατροφοδότησεις χρηστών και bugs. Το προϊόν παίρνετε τώρα δεν θα ήταν σε αυτή τη κατάσταση εάν δεν ήταν οι μεγάλες συνεισφορές και η υποστήριξη που έλαβα από πολλούς ανθρώπους .

Το Multi Router Traffic Grapher (MRTG) είναι ένα εργαλείο για παρακολούθηση του φόρτου της κίνησης πάνω σε συνδέσεις δικτύων. Το MRTG παράγει HTML σελίδες που

περιέχουν εικόνες GIF, οι οποίες παρέχουν μία ζωντανή εικονική αναπαράσταση της κίνησης. Ένα παράδειγμα μπορεί να δει κανένας στο <http://netmon.grnet.gr>. Τα MRTG

βασίζονται στις γλώσσες Perl και C και δουλεύουν κάτω από Unix και Windows NT. Η λειτουργία του MRTG στηρίζεται σε ένα Perl αρχείο σεναρίου που χρησιμοποιεί SNMP

κλήσεις για να διαβάσει τους μετρητές κίνησης των δρομολογητών και ένα γρήγορο C πρόγραμμα, για να καταγράψει τα δεδομένα της κίνησης και να δημιουργήσει διαγράμματα αναπαριστώντας την κίνηση της υπό παρακολούθησης δικτυακής σύνδεσης. Αυτά τα διαγράμματα ενσωματώνονται σε σελίδες, οι οποίες μπορούν να ειδωθούν από οποιονδήποτε web browser.

Επιπρόσθετα το MRTG δημιουργεί επίσης εικονικές αναπαραστάσεις της κίνησης που εθεάθη κατά τη διάρκεια των τελευταίων επτά ημερών, των τελευταίων τεσσάρων

εβδομάδων και των τελευταίων δώδεκα μηνών. Αυτό είναι δυνατό, γιατί το MRTG κρατάει ένα αρχείο με όλα τα δεδομένα που κατέγραψε από τον δρομολογητή. Αυτό το αρχείο είναι αυτόματα σταθεροποιημένο, έτσι ώστε να μην μεγαλώνει πέρα από το χρόνο, αλλά να εξακολουθεί να περιέχει όλα τα σχετικά δεδομένα για όλες τις κινήσεις που κατέγραψε τα τελευταία δύο χρόνια.

Το MRTG δεν είναι περιορισμένο μόνο στην παρακολούθηση δικτυακής κίνησης, αλλά είναι δυνατό να παρακολουθήσει οποιαδήποτε SNMP μεταβλητή επιλεχθεί.

Επιπρόσθετα είναι δυνατό να χρησιμοποιηθούν εξωτερικά προγράμματα τα οποία θα συλλέγουν την

πληροφορία για λογαριασμό του MRTG

### **Κύρια Σημεία**

Τα χαρακτηριστικά του MRTG είναι:

- Τρέχει σε κάθε πλατφόρμα Unix και Windows NT
- Χρησιμοποιεί την Perl για εύκολη διαμόρφωση
- Έχει μεγάλη διαθέσιμη SNMP υλοποίηση γραμμένη ολοκληρωτικά σε Perl. Δεν χρειάζεται να εγκατασταθεί κανένα εξωτερικό SNMP πακέτο
- Τα αρχεία καταγραφής του MRTG δεν μεγαλώνουν.
- Το MRTG έρχεται μαζί με ένα ημι-αυτόματο εργαλείο διαμόρφωσης
- Η μηχανή ερωτήσεων του MRTG, ελέγχει για αλλαγές στη διαμόρφωση των θυρών του δρομολογητή και προειδοποιεί τον χρήστη όταν αυτές συμβούν.
- Ρουτίνες κρίσιμες στον χρόνο έχουν γραφτεί σε C
- Τα γραφικά παράγονται κατευθείαν σε GIF μορφοποίηση, χρησιμοποιώντας την GD βιβλιοθήκη του Thomas Boutell.
- Αυτό το πακέτο παράγει PNG ή GIF μορφοποίηση, που εξαρτάται από την έκδοση της GD LIB που χρησιμοποιείται για να χτιστεί το MRTG. Οι GD LIB κάτω από την έκδοση 1.3, παράγουν LZW συμπιεσμένα GIFS, οι εκδόσεις 1.3-1.5 χρησιμοποιούν run-length συμπίεση και η έκδοση 1.6 και αργότερες χρησιμοποιούν PNG μορφοποίηση
- Η μορφή των ιστοσελίδων που παράγονται με το MRTG είναι πάρα πολύ διαμορφώσιμες
- Το MRTG είναι διαθέσιμο κάτω από τη GNU General Public License

## **ΕΓΚΑΤΑΣΤΑΣΗ**

Υποθέτουμε ότι τα ακόλουθα γίνονται στο μηχάνημα που θα τρέξει το MRTG, το οποίο πρέπει να είναι και web server. Όλα τα παραδείγματα είναι για τοπική μηχανή. **Πρώτα**

Unzip MRTG to C:\mrtg-2.13.0rc4 στο μηχάνημα της επιλογής σας .

### **Next**

Εγκαταστήστε το Perl στην ίδια μηχανή .Μπορεί να θέλετε να δείτε αν το Perl μπήκα στη σωστή θέση .

```
C:\Perl\bin;%SystemRoot%\system32;%SystemRoot%;...
```

Μπορείτε με το χέρι να ελέγξετε αυτό με τη μετάβαση [Control Panel]->[System]->[Environment]

Για να δείτε εάν όλα εγκαθίστανται κατάλληλα μπορείτε να πατε σε command και να δώσετε `c:\mrtg-2.13.0rc4\bin`. Γράψτε :

```
perl mrtg
```

Αυτό πρέπει να σας δώσει ένα φιλικό μήνυμα λάθους παραπονούμενος για το ελλείπον mrtg αρχείο διαμόρφωσης. Τώρα, έχετε εγκαταστήσει επιτυχώς mrtg και perl

## Συμβουλές Διαμόρφωσης

### CONFIGURING MRTG

Τώρα είναι ώρα να δημιουργηθεί μια διαμόρφωση για το mrtg. Αλλά προτού να αρχίσουμε πρέπει να ξέρετε μερικά πράγματα.

Την IP ή το hostname και το snmp port number της συσκευής που θέλετε να αποικονίσετε

- 

Εάν θέλετε να ελέγξετε κάτι εκτός από τα bytes, πρέπει επίσης να ξέρετε το SNMP OID αυτού που θέλετε να ελέγξετε.

- 

Τελικά πρέπει να ξέρετε τη read-only SNMP κοινότητα για τη συσκευή σας. Εάν δεν το ξέρετε, προσπαθήστε **public**, αυτή είναι η προεπιλογή.

Για το υπόλοιπο του παρόντος εγγράφου θα χρησιμοποιούμε τη συσκευή **10.10.10.1** με την κοινοτική σειρά **public**. Ενδιαφερόμαστε για τον έλεγχο της κυκλοφορίας, και το φορτίο της CPU. Ας ξεκινήσουμε :

Το πρώτο πράγμα που κάνουμε στη σύσταση mrtg είναι ένα αρχείο προεπιλογής config. Μπαίνουμε σε cmd περιβάλλον και πάμε στο *c:\mrtg-2.13.0rc4\bin*. Δακτυλογραφήστε την ακόλουθη εντολή:

```
perl cfgmaker public@10.10.10.1 --global "WorkDir: c:\www\mrtg" --output mrtg.cfg
```

Αυτό δημιουργεί ένα αρχικό config αρχείο MRTG για σας. Σημειώστε ότι σε αυτό το αρχείο όλες οι διεπαφές του δρομολογητή σας θα αποθηκευτούν σε αριθμό. Δυστυχώς, αυτοί οι αριθμοί είναι πιθανό να αλλάξουν όποτε μετατρέπετε το δρομολογητή σας. Προκειμένου να εργαστείτε γύρω από αυτό μπορείτε να πάρετε *cfgmaker* να παραγάγετε μια διαμόρφωση που είναι βασισμένη στους αριθμούς IP, ή ακόμα και τις περιγραφές διεπαφών.

Εάν λαμβάνετε ένα μήνυμα λάθους παραπονούμενο για **κανένα τέτοιο όνομα** ή **καμία απάντηση**, το κοινοτικό όνομά σας είναι πιθανόν λάθος .

Τώρα, ρίξτε μια ματιά στο αρχείο mrtg.cfg που δημιουργήθηκε.

Στο Perl, υπάρχει ένα σχόλιο # , συνώνυμο με το REM στο DOS.

Προσθέστε το ακόλουθο στην κορυφή του αρχείου mrtg.cfg:

WorkDir: D:\InetPub\wwwroot\MRTG

```
#####  
#  
# Description: LCP SUWGB  
# Contact: Administrator  
# System Name: LC-Bridge  
# Location: Here
```

```
#.....  
.
```

TargetDevice's IP Address: Interface Number: Community: IP Address

```
Target[10.10.10.1.1]: 1:public@10.10.10.1
```

Αυτό είναι η ταχύτητα διαπαφών (Default is 10 megabits; for 100Mbit devices use 12500000 and so on...)

```
MaxBytes[10.10.10.1.1]: 1250000  
Title[10.10.10.1.1]: LC-Bridge (sample.device): ether0
```

Αυτό το τμήμα καθορίζει πώς θα εμφανίζονται οι επιγραφές ιστοσελίδας

```
PageTop[10.10.10.1.1]: <H1>Traffic Analysis for ether0</H1>  
<TABLE>  
<TR><TD>System:</TD><TD>LC-Bridge inAndover</TD></TR>  
<TR><TD>Maintainer:</TD><TD>Administrator</TD></TR>  
<TR><TD>Interface:</TD><TD>ether0(1)</TD></TR>  
<TR><TD>IP:</TD><TD>sample.device(10.10.10.1)</TD></TR>  
<TR><TD>Max Speed:</TD>  
<TD>1250.0 kBytes/s (ethernetCsmacd)</TD></TR>  
</TABLE>
```

```
Target[10.10.10.1.2]: 2:public@10.10.10.1
```

```
MaxBytes[10.10.10.1.2]: 1250000
```

```
Title[10.10.10.1.2]: LC-Bridge (): ulink0
```

```
PageTop[10.10.10.1.2]: <H1>Traffic Analysis for ulink0</H1>
```

```
<TABLE>  
<TR><TD>System:</TD><TD>LC-Bridge inAndover</TD></TR>  
<TR><TD>Maintainer:</TD><TD>Administrator</TD></TR>  
<TR><TD>Interface:</TD><TD>ulink0(2)</TD></TR>  
<TR><TD>IP:</TD><TD>()</TD></TR>  
<TR><TD>Max Speed:</TD>  
<TD>1250.0 kBytes/s (ethernetCsmacd)</TD></TR>  
</TABLE>
```

```
#-----
```

Αυτό είναι ένα βασικό conf αρχείο . Μπορείτε να τρέξετε αυτό και να δείτε τα αποτελέσματά σας με τη μετάβαση στο *c:\mrtg-2.13.Orc4\bin* και γράψετε :

```
perl mrtg mrtg.cfg
```

Είναι κανονικό να αποκτηθούν τα λάθη για τους πρώτους δύο χρόνους που τρέχετε αυτήν την εντολή. Τα λάθη θα σας προειδοποιήσουν για το γεγονός ότι δεν έχουν υπάρξει οποιαδήποτε αρχεία ημερολογίου πριν.

Εάν ρίχνετε μια ματιά σε αυτές τις ιστοσελίδας δεν είναι και πολύ συναρπαστικές (ακόμα). Πρέπει να τρέχει το mrtg συνέχεια για να παραγάγετε τα επιθυμητά αποτελέσματα. Τρέξτε το πάλι μετά από λίγα λεπτά. Πρέπει τώρα να είστε σε θέση να δείτε τις πρώτες γραμμές στις γραφικές παραστάσεις σας.

Σε αυτή την παράγραφο παραθέτονται μερικές συμβουλές για την διαμόρφωση.

- Αν παρακολουθούνται πολλές συνδέσεις, ενδεχομένως να χρειάζεται η δημιουργία μιας σελίδας αναφοράς. Για τον σκοπό αυτό μπορεί να χρησιμοποιηθεί το script `indexmaker` ως εξής:

`indexmaker <mrtg.cfg> <regexp for router-names>`

- Αν παρακολουθούνται πολλοί στόχοι, μπορούν να χρησιμοποιηθούν οι ειδικούς χαρακτήρες για στόχους ``^`` και ``$`` για να προηγούνται ή να ακολουθούν κείμενο σε οποιαδήποτε από τις λέξεις κλειδιά στα παρακάτω μέρη. Ας σημειωθεί ότι οι ορισμοί των στόχων ``^`` και ``$`` είναι ανεξάρτητοι από την τοποθεσία αλλά πάντα επηρεάζουν τις γραμμές που έπονται στο αρχείο διαμόρφωσης. Ο ειδικός στόχος ``_`` για να ορίζει εξ ορισμού τιμές, μπορεί να αποδειχθεί πολύ εύχρηστος επίσης.

- Αφού το MRTG ενημερώνει τα διαγράμματα κάθε 5 λεπτά, μπορεί να υπάρξουν προβλήματα με τη λανθάνουσα μνήμη (cache memory) των proxy και την τοπική λανθάνουσα των browser. Μερικές φορές επιστρέφονται τα παλιά αποθηκευμένα γραφικά, αντί για τα πραγματικά και τις ανανεωμένες εκδόσεις του διακομιστή ιστοσελίδων. Αν χρησιμοποιείται ο διακομιστής ιστοσελίδων apache, μπορεί να χρησιμοποιηθεί η λέξη κλειδί `WriteExpire` στο `mrtg.cfg` αρχείο. Έτσι, τα mrtg θα δημιουργήσουν \*.meta αρχεία για κάθε gif και html σελίδα. Αυτά τα αρχεία θα περιέχουν επικεφαλίδες 'Expire' τις οποίες οι διακομιστές apache μπορούν να στείλουν μαζί με τα gif και τις ιστοσελίδες (Είναι απαραίτητη η χρησιμοποίηση της λέξης κλειδί `MetaDir` στο αρχείο διαμόρφωσης του apache για να ενεργοποιηθεί). Με τις πληροφορίες για τις επικεφαλίδες expire, όλοι οι browsers και οι λανθάνουσες μνήμες των proxy, θα ξέρουν πότε θα πρέπει να ζητήσουν νέα έκδοση του αρχείου από το website και πότε μπορούν να χρησιμοποιούν την αποθηκευμένη έκδοση τους.

- Αν και ο πρωταρχικός σκοπός του MRTG είναι η παρακολούθηση της κίνησης, μπορεί να παρακολουθείται οποιαδήποτε SNMP μεταβλητή.

## Τροπος Διαμορφωσης

Σε αυτό το κεφάλαιο θα περιγραφούν όλες οι παράμετροι διαμόρφωσης του MRTG. Θα πρέπει να δοθούν μερικές διευκρινιστικές υποδείξεις:

- Οι λέξεις κλειδιά θα πρέπει να ξεκινούν από την αρχή της πρότασης.
- Οι γραμμές που ακολουθούν μία λέξη κλειδί και ξεκινούν με κενό, επισυνάπτονται στη γραμμή με τη λέξη κλειδί.
- Οι κενές γραμμές αγνοούνται.
- Γραμμές που ξεκινούν με το ``#`` λαμβάνονται ως σχόλια.

## Καθολικη Διαμορφωση

Το `Workdir` καθορίζει το μέρος όπου τα αρχεία καταγραφής και οι ιστοσελίδες θα πρέπει να δημιουργηθούν.

Παράδειγμα:

`WorkDir: /usr/local/mrtg/html`

## Προαιρετικες Καθολικες Παραμετροι

Refresh

Σε πόσα δευτερόλεπτα θα πρέπει να πληροφορηθεί ο browser (Netscape) να φορτώσει

ξανά τη σελίδα. Αν αυτό δεν έχει οριστεί, η εξ' ορισμού τιμή είναι 300 δευτερόλεπτα (5 λεπτά).

Παράδειγμα:

**Refresh: 600**

Interval

Πόσο συχνά καλείται το MRTG; Η εξ' ορισμού τιμή είναι 5 λεπτά. Αν καλείται λιγότερο

συχνά, θα πρέπει να καθοριστεί εδώ. Αυτό κάνει δύο πράγματα:

- Οι παραγόμενες σελίδες HTML περιέχουν τη σωστή πληροφορία για το χρονικό διάστημα της κλήσης.
- Μία META επικεφαλίδα στη παραγόμενη σελίδα HTML θα πληροφορήσει για το χρόνο που θα ζηήσει αυτή η σελίδα στη cache. Σε αυτό το παράδειγμα, το MRTG θα καλείται κάθε 10 λεπτά. Αν το mrtg καλείται κάθε 5 λεπτά, η γραμμή μπορεί να αφαιρεθεί χωρίς σχόλιο.

Παράδειγμα:

**Interval: 10**

WriteExpires

Με αυτή την επιλογή το MRTG θα δημιουργήσει .meta αρχεία για CERN και Apache διακομιστές που θα περιέχουν ετικέτες λήξης για τις ιστοσελίδες και τα αρχεία GIF.

Τα

\*.meta αρχεία θα δημιουργηθούν στο ίδιο κατάλογο όπως όλα τα αρχεία, έτσι μπορεί να

χρησιαστεί να τεθεί το "MetaDir ." στο srm.conf αρχείο για να δουλέψει αυτό.

ΣΗΜΕΙΩΣΗ: Αν χρησιμοποιείται η έκδοση Apache-1.2 μπορεί να χρησιμοποιηθεί το mod\_expire για να επιτευχθεί το ίδιο αποτέλεσμα (βλ. htaccess.txt).

Παράδειγμα:

**WriteExpires: Yes**

IconDir

Αν τα εικονίδια του MRTG θα βρίσκονται σε άλλο μέρος, εκτός του καταλόγου εργασίας,

μπορεί να χρησιμοποιηθεί η μεταβλητή IconDir για να καθοριστεί το URL τους.

Παράδειγμα:

**IconDir: /mrtgicons/**

LoadMIBs

Φορτώνει το αρχείο MIB που καθορίζεται και κάνει τα OID's του διαθέσιμα σαν συμβολικά ονόματα. Μια cache των MIBs διατηρείται μέσα στο WorkDir.

Παράδειγμα:

**LoadMIBs:/usr/local/lib/mibdir/demo,**

**/usr/local/lib/mibdir/cisco.def**

Language

Αλλάζει τη μορφοποίηση της εξόδου στην επιλεγμένη γλώσσα (οι τιμές danish, french,

english, spanish, greek, russian, italian, brasilian και dutch υποστηρίζονται).

Παράδειγμα:

**Language: greek**

UseRRDTool

Η ρύθμιση του UseRRDTool στο Yes ενεργοποιεί την μέθοδο του rrdtool. Στην rrdtool μέθοδο, το MRTG βασίζεται στο rrdtool για να κάνει το logging. Τα διαγράμματα και

οι

ιστοσελίδες θα δημιουργηθούν εκείνη την ώρα με το 14all.cgi που μπορεί να βρεθεί στο

contrib τμήμα μαζί με ένα σύντομο readme.

Παράδειγμα:

**UseRRDTool: Yes**

### 4.3. Διαμορφωση Για Τον Καθε Στοχο

Οι λέξεις κλειδιά Target θα πρέπει να ακολουθούνται από ένα μοναδικό όνομα. Αυτό θα

είναι επίσης το όνομα που θα χρησιμοποιείται για τις ιστοσελίδες, τα αρχεία καταγραφής

και τα GIFs που παράγονται για αυτό τον στόχο. Θα πρέπει να σημειωθεί ότι τα τμήματα

του Target μπορούν να παραχθούν αυτόματα με τη βοήθεια του εργαλείου cfmaker.

Target Με τη λέξη κλειδί Target μεταδίδεται η πληροφορία στο MRTG σχετικά με το τι θα πρέπει να παρακολουθεί. Η λέξη κλειδί Target παίρνει ορίσματα διαφόρων μορφών.

- Η πιο βασική μορφή είναι "port:community@router". Αυτό θα παράγει ένα διάγραμμα κίνησης για τη διεπαφή 'port' του δρομολογητή 'router' (DNS όνομα ή IP διεύθυνση) και θα χρησιμοποιήσει τη κοινότητα 'community' (SNMP password) για την SNMP ερώτηση.

Παράδειγμα:

**Target[mainrouter]: 1:public@athens.grnet.gr**

- Προσθέτοντας το σημάδι '-' μπροστά από την περιγραφή του Target, το MRTG αντιστρέφει τον εισερχόμενο και εξερχόμενο ρυθμό κίνησης.

Παράδειγμα:

**Target[mainrouter]: -1:public@athens.grnet.gr**

- Μπορεί επίσης να οριστεί απευθείας το OID που θα χρησιμοποιηθεί ακολουθώντας την σύνταξη 'OID\_1&OID\_2:community@router'. Το παρακάτω παράδειγμα θα ανακτήσει τον αριθμό των λαθών για είσοδο και έξοδο στη διεπαφή 1. Το MRTG χρειάζεται να ζωγραφίσει δύο μεταβλητές, έτσι θα πρέπει να καθοριστούν δύο OIDs όπως είναι η θερμοκρασία και η υγρασία ή λάθη εισόδου και λάθη εξόδου.

Παράδειγμα:

**Target[mainrouter]:1.3.6.1.2.1.2.2.1.14.1&1.3.6.1.2.1.2.2.1.2**

**0.1:public@athens.grnet.gr**

- Το MRTG έχει προκαθορισμένο ένα σύνολο από συμβολικά ονόματα μεταβλητών της MIB-II. Ένα παράδειγμα είναι το ifInErrors και το ifOutErrors. Αυτό σημαίνει ότι μπορείτε να το καθορίσετε όπως παρακάτω:

Παράδειγμα:

**Target[mainrouter]:**

**ifInErrors.1&ifOutErrors.1:public@athens.grnet.gr**

- Μερικές φορές μία διεπαφή μπορεί να αλλάξει, π.χ. όταν καινούργιες διεπαφές προστίθενται ή αφαιρούνται. Αυτό μπορεί να προκαλέσει όλες τις εισόδους των στόχων στο αρχείο διαμόρφωσης να γίνουν λάθος από μία μετατόπιση, προκαλώντας στα διαγράμματα του MRTG λάθος στιγμιότυπα κλπ. Το MRTG υποστηρίζει IP διευθύνσεις εκτός από το ifindex στον ορισμό του Target. Έτσι το MRTG θα ρωτήσει τη SNMP συσκευή και θα προσπαθήσει να αντιστοιχίσει την IP διεύθυνση στον τρέχοντα ifindex. Μπορεί να χρησιμοποιηθεί η IP διεύθυνση σε κάθε τύπο ορισμού του στόχου, προσθέτοντας την IP διεύθυνση της αριθμημένης διεπαφής μετά από το OID και το διαχωριστικό χαρακτήρα '/'. Πρέπει να είναι εξασφαλισμένο ότι η δοθείσα IP διεύθυνση χρησιμοποιείται στον ίδιο δρομολογητή στόχου, ειδικά όταν καταγράφονται δύο διαφορετικά OID και οι διεπαφές χωρίζονται με το διαχωριστικό '&'.

Παράδειγμα:

**Target[mainrouter]: /194.177.209.129:public@athens.grnet.gr**



**Target[mainrouter]: -/194.177.209.129:public@athens.grnet.gr**

**Target[mainrouter]:**

**1.3.6.1.2.1.2.2.1.14/194.177.209.129&/194.177.209.129:public@athens.grnet.gr**

**Target[mainrouter]:**

**ifInErrors/194.177.209.129&ifOutErrors/194.177.209.129:public@athens.grnet.gr**

Σε όλες τις θέσεις όπου το "community@router" είναι αποδεκτό, μπορεί να προστεθούν επιπλέον παράμετροι για την SNMP επικοινωνία χρησιμοποιώντας δίστιγμα επιθέματα. Η πλήρης σύνταξη είναι όπως ακολουθεί:

`community@router[:port[:timeout[:retries[:backoff]]]]`

όπου η σημασία της κάθε παραμέτρου είναι όπως παρακάτω:

port

ο αριθμός της πόρτας UDP με την οποία θα επικοινωνήσει με τον SNMP agent (εξ' ορισμού: 161)

timeout

αρχικοποιεί το timeout για τις SNMP ερωτήσεις, σε δευτερόλεπτα (εξ' ορισμού: 2.0)

retries

ο αριθμός των προσπαθειών όπου ένα timeout αίτημα θα επαναληφθεί (εξ' ορισμού: 5)

backoff

παράγοντας με τον οποίον το timeout πολλαπλασιάζεται σε κάθε επανάληψη

Μία τιμή που είναι ίδια με την εξ' ορισμού τιμή μπορεί να παραλειφθεί. Συνεχόμενα δίστιγμα μπορούν να παραλειφθούν επίσης.

Παράδειγμα:

**Target[mainrouter]: 1:public@athens.grnet.gr:9161::4**

Αυτό θα επιστρέψει τον μετρητή οκτάδων εισόδου/ εξόδου για τη διεπαφή με ifIndex 1

στον athens.grnet.gr, αφού ρωτηθεί ο SNMP agent που ακούει στη UDP πόρτα 9161.

Το

σπάνταρ αρχικό timeout (2.0 δευτερόλεπτα) χρησιμοποιείται, αλλά ο αριθμός των επαναλήψεων έχει τεθεί στο 4. Η backoff τιμή είναι η εξ' ορισμού.

Αν υπάρχει η ανάγκη να παρακολουθείται κάτι που δεν παρέχει δεδομένα μέσω του SNMP,

μπορεί να χρησιμοποιηθεί ένα εξωτερικό πρόγραμμα για να κάνει τη συλλογή δεδομένων. Το εξωτερικό πρόγραμμα θα πρέπει να έχει 4 γραμμές για έξοδο:

Γραμμή 1

τρέχουσα τιμή της πρώτης μεταβλητής, συνήθως 'εισερχόμενη μέτρηση bytes'

Γραμμή 2

τρέχουσα τιμή της δεύτερης μεταβλητής, συνήθως 'εξερχόμενη μέτρηση bytes'

Γραμμή 3

Ο χρόνος uptime για τον συγκεκριμένο στόχο

Γραμμή 4

Το όνομα του στόχου

Ανάλογα με τον τύπο δεδομένων που το script επιστρέφει μπορεί να θέλετε να χρησιμοποιηθούν τα ορίσματα 'gauge' ή 'absolute' στη λέξη κλειδί Options.

Παράδειγμα:

**Target[mainrouter]: `usr/local/bin/df2mrtg /dev/dsk/c0t2d0s0`**

Προσοχή χρειάζεται στην χρήση της ανάποδης αποστρόφου ( ` ) γύρω από την εντολή.

Μπορούν να χρησιμοποιηθούν πολλά ορίσματα σε μία μαθηματική έκφραση. Έτσι μπορούν να προστεθούν οι τιμές από και τα δύο κανάλια σε μία ISDN σύνδεση ή πολλαπλά T1 ενοποιημένα σε ένα μονό κανάλι για μεγαλύτερο εύρος ζώνης. Προσοχή χρειάζεται στο κενό ανάμεσα στους στόχους.

Παράδειγμα:

**Target[mainrouter]: 2:public@wellfleetA + 1:public@wellfleetA \*  
4:public@ciscoF**

RouterUptime

Σε περιπτώσεις που υπολογίζεται το χρησιμοποιημένο εύρος ζώνης από πολλές διεπαφές,

συνήθως δεν χρησιμοποιείται το uptime του δρομολογητή.

Αν κάτι τέτοιο είναι εν τούτοις επιθυμητό, μπορεί να ενεργοποιηθεί ως εξής:

**Target[mainrouter]: 1:public@194.177.210.41 +**

**2:public@194.177.210.41**

**RouterUptime[mainrouter]: public@194.177.210.41**

MaxBytes

Η μέγιστη τιμή και των δύο μεταβλητών που παρακολουθούνται. Για παρακολούθηση δρομολογητών, αυτό συνήθως καθορίζεται σε bytes που μπορεί η πόρτα να διακινήσει

στην μονάδα του χρόνου. Αν ένας αριθμός μεγαλύτερος από το MaxBytes

επιστραφεί, θα αγνοηθεί. Το MaxBytes επίσης χρησιμοποιείται για να υπολογίσει την Υ ακτίνα για τα χωρίς διαβάθμιση διαγράμματα (βλέπε Unscaled).

Αφού οι περισσότερες συνδέσεις έχουν ρυθμό σε bits ανά δευτερόλεπτο, θα πρέπει να

διαιρεθεί το μέγιστο εύρος ζώνης (σε bits) με το οχτώ (8) για να ληφθούν τα bytes ανά

δευτερόλεπτο. Αυτό είναι πολύ σημαντικό ώστε οι χωρίς διαβάθμιση γράφοι να

εμφανίζουν περισσότερο ρεαλιστικές πληροφορίες. Αν χρειάζεται δύο διαφορετικά

MaxBytes τιμές για τις δύο παρακολουθούμενες μεταβλητές, μπορούν να

χρησιμοποιηθούν οι μεταβλητές MaxBytes1 και MaxBytes2 αντί για MaxBytes.

Παράδειγμα:

**MaxBytes[mainrouter]: 1250000**

MaxBytes1

Το ίδιο με το MaxBytes, για την μεταβλητή 1.

MaxBytes2

Το ίδιο με το MaxBytes, για την μεταβλητή 2.

Title

Ο τίτλος για την ιστοσελίδα που παράγεται για το διάγραμμα.

Παράδειγμα:

**Title[mainrouter]: athens.grnet.gr: Backbone Athens-Thessaloniki  
[ATM 16 Mbps]**

PageTop

Επικεφαλίδα που θα προστεθεί στην κορυφή της κάθε παραγόμενης ιστοσελίδας. Θα πρέπει να σημειωθεί ότι οι συνεχόμενες γραμμές θα καταλήξουν σε μία γραμμή στην ιστοσελίδα. Μπορεί εν τούτοις να χρησιμοποιηθεί η ακολουθία «\n» για να προκαλέσει αλλαγή γραμμής.

Παράδειγμα:

**PageTop[mainrouter]: <H1>Traffic Analysis for ATM</H1>**

**Our Campus Backbone runs over an ATM line\n  
with maximum transfer rate of 34 megabits per  
second.**

## 5. Προαιρετικές Επιλογές Διαμορφώσης Στοιχού

AddHead

Έχει την ίδια λειτουργία με το PageTop, αλλά τα περιεχόμενά θα προστεθούν μεταξύ του

</TITLE> και </HEAD>.

Παράδειγμα:

**AddHead[mainrouter]:** <link rev="made" href="mailto:grnet@noc.ntua.gr";>

AbsMax

Αν παρακολουθείται μία γραμμή που μπορεί να χειριστεί περισσότερη κίνηση από τη MaxBytes τιμή (για παράδειγμα κάποια που χρησιμοποιεί συμπίεση ή κάποια frame relay

σύνδεση), μπορεί να χρησιμοποιηθεί η λέξη κλειδί AbsMax, για να οριστεί η απόλυτα μέγιστη τιμή. Αν δεν υπάρχει το AbsMax, το rateup θα αγνοήσει τιμές μεγαλύτερες από το MaxBytes.

Παράδειγμα:

**AbsMax[mainrouter]:** 250000

Unscaled

Εξ' ορισμού κάθε γράφος κατασκευάζεται με κινητή κάθετη κλίμακα για να γίνονται τα

πραγματικά δεδομένα ορατά, ακόμα και όταν είναι πολύ μικρότερα από το MaxBytes.

Με

τη Unscaled μεταβλητή μπορεί να απενεργοποιηθεί αυτή η λειτουργία. Το όρισμά του είναι μία συμβολοσειρά που περιέχει ένα γράμμα για κάθε διάγραμμα που δεν είναι επιθυμητό να έχει κινητή κλίμακα: d=day, w=week, m=month, y=year. Σε αυτό το παράδειγμα η κινητή κλίμακα για το ετήσιο και το μηνιαίο διάγραμμα αποτρέπεται.

Παράδειγμα:

**Unscaled[mainrouter]:** ym

WithPeak

Εξ' ορισμού τα διαγράμματα περιέχουν μόνο τις μέσες και τις παρακολουθούμενες τιμές,

συνήθως δηλαδή τους ρυθμούς μετάδοσης για την εισερχόμενη και εξερχόμενη κίνηση. Η παρακάτω επιλογή ρυθμίζει το MRTG να εμφανίζει την πεντάλεπτη τιμή της κορυφής στο [w]eekly, [m]onthly και [y]early διάγραμμα. Στο παράδειγμα ορίζεται το μηνιαίο και ετήσιο διάγραμμα να περιέχει κορυφές, καθώς επίσης και τις μέσες τιμές.

Παράδειγμα:

**WithPeak[mainrouter]:** ym

Suppress

Εξ' ορισμού το MRTG παράγει 4 διαγράμματα. Με αυτή την επιλογή μπορεί να σταματήσει η παραγωγή ενός συγκεκριμένου διαγράμματος. Η σύνταξη της επιλογής αυτής είναι ανάλογη με τις δύο προηγούμενες επιλογές. Σε αυτό το παράδειγμα αποτρέπεται η δημιουργία του ετήσιου διαγράμματος.

Παράδειγμα:

**Suppress[mainrouter]:** y

Directory

Εξ' ορισμού το MRTG τοποθετεί όλα τα αρχεία που δημιουργεί για τον κάθε στόχο (τα

GIFs, τις ιστοσελίδες, τα αρχεία καταγραφής κλπ) στο WorkDir.

Αν η παράμετρος Directory είναι ορισμένη, τα αρχεία αποθηκεύονται κάτω σε ένα κατάλογο κάτω από το WorkDir. Ο κατάλογος αυτός θα πρέπει να υπάρχει, γιατί το MRTG δεν πρόκειται να τον δημιουργήσει.

Παράδειγμα:

**WorkDir:** /usr/local/mrtg/html

**Directory:** grnet

**Xsize και Ysize**

Εξ' ορισμού τα διαγράμματα του MRTG είναι 100 επί 400 pixels πλατιά (συν μερικά για τις ετικέτες). Στο παράδειγμα παίρνουμε σχεδόν τετράγωνους γράφους.

Σημείωση: Το Xsize θα πρέπει να είναι μεταξύ 20 και 600. Το Ysize θα πρέπει να είναι

μεγαλύτερο από 20.

Παράδειγμα:

**XSize[mainrouter]: 300**

**Ysize[mainrouter]: 300**

#### **XZoom και YZoom**

Χρησιμοποιούνται για να μεγεθύνουν τους γράφους.

Παράδειγμα:

**XZoom[mainrouter]: 2.0**

**YZoom[mainrouter]: 2.0**

#### **XScale και YScale**

Χρησιμοποιούνται για τον ορισμό της κλίμακας των γράφων.

Παράδειγμα:

**XScale[mainrouter]: 1.5**

**YScale[mainrouter]: 1.5**

#### **YTics και YTicsFactor**

Αν είναι επιθυμητό να φαίνονται περισσότερες από 4 γραμμές για κάθε διάγραμμα, μπορεί να χρησιμοποιηθεί η παράμετρος YTics. Αν είναι επιθυμητό να αλλάξει η κλίμακα στη χρησιμοποιούμενη γραμμή του YLegend για αυτές τις τιμές, να χρησιμοποιηθεί η

παράμετρος YTicsFactor. Η εξ' ορισμού τιμή του YTics είναι 4 και για το YTicsFactor είναι

1.0.

Παράδειγμα:

Αν υποθεθεί ότι χρειάζονται τιμές από το 0 έως το 700

**YTics[mainrouter]: 7**

**YTicsFactor[mainrouter]: 0.01**

#### **Step**

Αλλάζει το εξ' ορισμού βήμα από 5\*60 δευτερόλεπτα, σε κάποια άλλη τιμή:

Παράδειγμα:

**Step[mainrouter]: 60**

#### **Options**

Η λέξη κλειδί Options επιτρέπει να τεθούν ορισμένες λογικές παράμετροι:

growright

Το διάγραμμα μεγαλώνει από την αριστερή μεριά. Αυτή η επιλογή αντιστρέφει τη κατεύθυνση, προκαλώντας την τρέχουσα ώρα να είναι στην δεξιά άκρη του διαγράμματος και τις παλιές τιμές στα αριστερά.

bits

Όλες οι παρακολουθούμενες τιμές πολλαπλασιάζονται με το 8 (για να δείχνονται σε bits

αντί για bytes). Επίσης επηρεάζονται οι ετικέτες και οι μονάδες για τον συγκεκριμένο στόχο.

perminute

Όλες οι παρακολουθούμενες μεταβλητές πολλαπλασιάζονται με το 60 (δηλαδή φαίνονται

σε μονάδες ανά λεπτό αντί για μονάδες ανά δευτερόλεπτο). Σε περίπτωση μικρών τιμών,

δημιουργούνται διαγράμματα με περισσότερη ακρίβεια. Επηρεάζει επίσης τις ετικέτες και

τις μονάδες που μπαίνουν εξ' ορισμού.

perhour

Όλες οι παρακολουθούμενες τιμές πολλαπλασιάζονται με το 3600 (δηλαδή φαίνονται σε

μονάδες ανά ώρα αντί για μονάδες ανά δευτερόλεπτο). Σε περίπτωση μικρών τιμών, δημιουργούνται περισσότερο ακριβή διαγράμματα. Επίσης επηρεάζει τις ετικέτες και τις

μονάδες που μπαίνουν εξ' ορισμού.

noinfo

Αποτρέπει την πληροφορία για το χρόνο λειτουργίας και το όνομα της συσκευής στις παραγόμενες ιστοσελίδες.

nopercent

Δεν εκτυπώνει τα ποσοστά.

transparent

Κάνει το φόντο των διαγραμμάτων διαφανές.

integer

Εκτυπώνει της γραμμές περίληψης κάτω από τα διαγράμματα σαν ακέραιους χωρίς κόμμα.

dorepercent

Το σχετικό ποσοστό της εισερχόμενης σε εξερχόμενη, υπολογίζεται και εκτυπώνεται στο

διάγραμμα σαν επιπλέον γραμμή. Ας σημειωθεί ότι μόνο μία σταθερή κλίμακα είναι διαθέσιμη (από 0 έως 100%). Έτσι για εισερχόμενη κίνηση μεγαλύτερη από την εξερχόμενη εμφανίζεται επίσης το 100%. Αν υπάρχει η υποψία ότι η εισερχόμενη κίνηση

δεν είναι πάντα μικρότερη ή ίση από την εξερχόμενη, καλό θα είναι να μην χρησιμοποιείται η επιλογή αυτή.

gauge

Μεταχειρίζεται τις τιμές που συλλέγονται από τον στόχο σαν απόλυτες και όχι σαν αύξουσες μετρήσεις. Αυτό είναι χρήσιμο για παρακολούθηση πραγμάτων όπως ο χώρος

στο δίσκο, ο φόρτος του επεξεργαστή, η θερμοκρασία κλπ.

Απουσία της gauge και της absolute επιλογής, το MRTG μεταχειρίζεται της μεταβλητές σαν μετρητές, υπολογίζει τη διαφορά της τρέχουσας και της προηγούμενης τιμής και διαιρεί με τον χρόνο που πέρασε μεταξύ των δύο τελευταίων διαβασμάτων, για να πάρει τη τιμή η οποία θα μπει στο διάγραμμα.

absolute

Αυτό είναι για πηγές δεδομένων που αρχικοποιούν ξανά τη τιμή τους όταν διαβαστούν.

Αυτό σημαίνει ότι το rateup δεν χρειάζεται να φτιάξει τη διαφορά μεταξύ αυτής και της

προηγούμενης τιμής που διάβασε από την πηγή δεδομένων. Η τιμή που επιστρέφεται εξακολουθεί να διαιρείται με το χρόνο που πέρασε μεταξύ των δύο μετρήσεων, η οποία

κάνει τη διαφορά από την επιλογή gauge. Είναι χρήσιμη για εξωτερικούς συλλέκτες δεδομένων.

unknazero

Αν καταγραφούν άγνωστα δεδομένα μηδενίζει την επιστρεφόμενη τιμή. Καλό είναι να χρησιμοποιείται με προσοχή, γιατί μία ευθεία γραμμή στο διάγραμμα είναι πολύ περισσότερο προφανής από μία γραμμή στο 0.

Παράδειγμα:

**Options[mainrouter]: growright, bits**

kilo

Η τιμή αυτή μεταβάλλει τον πολλαπλασιαστή για την δημιουργία προθεμάτων. Η εξ' ορισμού τιμή είναι το 1000. Αυτή η επιλογή είναι για την ειδική περίπτωση που το 1Kb=1024b, 1MB=1024Kb κλπ.

Παράδειγμα:

**kilo[mainrouter]:1024**

kMG

Αλλάζει τα εξ' ορισμού προθέματα (,k,M,G,T,P). Στην επιλογή ShortLegend ορίζονται μόνο οι βασικές μονάδες. Η μορφοποίηση της επιλογής είναι με μία λίστα από προθέματα, διαχωριζόμενα με κόμμα. Δύο διαδοχικά κόμματα στην αρχή ή στο τέλος της γραμμής, δεν δίνουν κανένα πρόθεμα για αυτό το αντικείμενο. Για να μην υπάρχει πρόθεμα, η επιλογή αυτή μπορεί να τοποθετηθεί κενή.

Παράδειγμα: ταχύτητα σε nm/s που εμφανίζεται σε nm/h

**ShortLegend[mainrouter]: m/min**

**kMG[mainrouter]: n,u,m,,k,M,G,T,P**

**Options[mainrouter]: perhour**

Colours

Η επιλογή Colours επιτρέπει την αλλαγή του χρωματικού σχήματος. Όλα τα 4 απαιτούμενα χρώματα θα πρέπει να δηλωθούν εδώ. Το όνομα του χρώματος είναι το όνομα της λεζάντας που εμφανίζεται, καθώς η RGB τιμή είναι το πραγματικό χρώμα για τον σχεδιασμό και για τον γράφο και για το HTML έγγραφο.

Η μορφοποίηση είναι: Colour1#RRGGBB, Colour2#RRGGBB, Colour3#RRGGBB, Colour4#RRGGBB

Colour1

Πρώτη μεταβλητή (συνήθως είσοδος) στο εξ' ορισμού διάγραμμα.

Colour2

Δεύτερη μεταβλητή (συνήθως έξοδος) στο εξ' ορισμού διάγραμμα.

Colour3

Μέγιστο για την πρώτη μεταβλητή (είσοδος)

Colour4

Μέγιστο για την δεύτερη μεταβλητή (έξοδος)

RRGGBB

Δύο δεκαεξαδικά ψηφία για το Κόκκινο, το πράσινο και το μπλε.

Παράδειγμα:

**Colours[mainrouter]: GREEN#00EB0C, BLUE#1000FF, DARK GREEN#006600, VIOLET#FF00FF**

Background

Με την Background επιλογή, μπορεί να οριστεί το χρώμα του φόντου των παραγόμενων ιστοσελίδων.

Παράδειγμα:

**Background[mainrouter]: #A0A0A0**

**YLegend, ShortLegend, Legend[1234IO]**

Οι παρακάτω λέξεις κλειδιά επιτρέπουν να γίνει υπέρβαση στο κείμενο που εμφανίζεται για τις διάφορες ετικέτες των διαγραμμάτων στο HTML έγγραφο.

YLegend

Η ετικέτα του άξονα Y στο διάγραμμα. Κείμενο που είναι πολύ μεγάλο για να χωρέσει στο γράφο, θα αγνοηθεί σιωπηλά.

ShortLegend

Η συμβολοσειρά της μονάδας (εξ' ορισμού b/s) που χρησιμοποιείται για τη μέγιστη, μέση

και τρέχουσα μέτρηση.

Legend[1234IO]

Οι συμβολοσειρές για τις χρωματιστές ετικέτες.

Παράδειγμα:

**YLegend[mainrouter]: Bits per Second**

**ShortLegend[mainrouter]:** b/s

**Legend1[mainrouter]:** Incoming Traffic in Bits per Second

**Legend2[mainrouter]:** Outgoing Traffic in Bits per Second

**Legend3[mainrouter]:** Maximal 5 Minute Incoming Traffic

**Legend4[mainrouter]:** Maximal 5 Minute Outgoing Traffic

**LegendI[mainrouter]:** &nbsp; In:

**LegendO[mainrouter]:** &nbsp; Out:

Αν το LegendI ή το LegendO έχουν τεθεί σε κενή συμβολοσειρά με

**LegendO[mainrouter]:**

Η αντίστοιχη γραμμή δεν θα τυπωθεί καθόλου κάτω από το διάγραμμα.

**Timezone**

Επιτρέπει την παραγωγή διαγραμμάτων σε διαφορετικές χρονικές ζώνες. Αυτό ρυθμίζεται

στη μεταβλητή TZ. Σε μερικά λειτουργικά συστήματα, όπως το Solaris, αυτό θα προκαλέσει την κλήση της localtime να δώσει την ώρα στην επιλεγμένη χρονική ζώνη.

Παράδειγμα:

**Timezone[mainrouter]:** Japan

**Weekformat**

Εξ' ορισμού το MRTG (κανονικά το rateup) χρησιμοποιεί τη strftime (3) '%W' επιλογή, για να μορφοποιήσει τους αριθμούς των εβδομάδων στα μηνιαία διαγράμματα. Η ακριβής

σύνταξη για αυτή τη μορφοποίηση, διαφέρει από σύστημα σε σύστημα. Αν παρατηρηθεί

ότι οι αριθμοί των εβδομάδων είναι λάθος και η ρουτίνα του συστήματος strftime(3) το

υποστηρίζει, μπορεί να δοκιμαστεί μία άλλη επιλογή μορφοποίησης. Η POSIX επιλογή '%V' φαίνεται να ανταποκρίνεται είναι η καλύτερη επιλογή στις περισσότερες περιπτώσεις. Η μορφοποίηση της εβδομάδας θα πρέπει να καθορισθεί με ένα από γράμμα, είτε W, είτε V ή U.

Παράδειγμα:

**Weekformat[mainrouter]:** V

## 6. Έλεγχος Κατωφλίου

Με τη λειτουργικότητα του έλεγχου κατωφλίου, το MRTG είναι σε θέση να διακρίνει προβλήματα κατωφλίου για διάφορους στόχους και μπορεί να καλέσει εξωτερικά scripts για να χειριστούν αυτά τα προβλήματα. Ο έλεγχος κατωφλίου πραγματοποιείται με βάση τις παρακάτω παραμέτρους.

**ThreshDir**

Για την ανίχνευση περιπτώσεων στις οποίες μία παράμετρος είναι εκτός των προβλεπομένων τιμών θα πρέπει να οριστεί αυτός ο κατάλογος. Εκεί θα αποθηκευτούν

προσωρινά αρχεία μεταξύ των κύκλων για να δηλώσουν ποια παράμετρος έχει προβλήματα κατωφλίου στον προηγούμενο κύκλο.

**ThreshMinI**

Αυτή είναι η μικρότερη αποδεκτή τιμή για την πρώτη παράμετρο. Αν αυτή η παράμετρος

πέσει κάτω από αυτή τη τιμή, θα τρέξει το πρόγραμμα που έχει δηλωθεί στο ThreshProgI.

**ThreshMaxI**

Αυτή είναι η μέγιστη επιτρεπτή τιμή για την πρώτη παράμετρο. Αν η παράμετρος πάει πάνω από αυτή την τιμή, θα τρέξει το πρόγραμμα που έχει δηλωθεί στο ThreshProgI.

**ThreshProgI**

Ορίζει ένα πρόγραμμα που θα τρέξει αν το ThreshMinI ή το ThreshMaxI ξεπεραστεί. Δέχεται τρία ορίσματα: τη μεταβλητή \$router, την τιμή του κατωφλίου που έσπασε και τη τρέχουσα τιμή της παραμέτρου.

ThreshProgOKI

Ορίζει το πρόγραμμα που θα πρέπει να τρέξει, αν η παράμετρος είναι επί του παρόντος

εντάξει (βασισμένη το ThreshMinI και στο ThreshMaxI), αλλά δεν ήταν εντάξει στον προηγούμενο κύκλο, με βάση τα αρχεία που βρήκε στο ThreshDir.

**ThreshMinO, ThreshMaxO, ThreshProgO και ThreshProgOKO**

Δουλεύουν όπως τα \*I μέρη, αλλά για την δεύτερη παραμέτρο.

## 7. Ειδικά Ονοματα Στόχου

Ο στόχος σε ένα cfg αρχείο μπορεί να οριστεί και με το όνομα '^'. Το κείμενο για κάθε

λέξη κλειδί που ορίζεται για αυτόν τον στόχο, θα προηγηθεί των αντίστοιχων λέξεων κλειδί όλων των υπόλοιπων στόχων που ορίζονται κάτω από αυτή τη γραμμή.

Ένα κενό εισέρχεται μεταξύ του εισερχόμενου στην αρχή κειμένου και την τιμή της λέξης κλειδί, καθώς επίσης και της τιμής της λέξης κλειδί και του επισυναπτόμενου κειμένου. Αυτό δουλεύει καλά για λέξεις κλειδιά με τιμές συμβολοσειρές, αλλά δεν είναι πολύ χρήσιμη για άλλες λέξεις κλειδιά.

Παράδειγμα:

**PageTop[^]: <H1>Somewhere Traffic Statistics</H1><HR>**

**PageTop[\$]: Contact grnet@noc.ntua.gr if you have any questions**

**<HR>**

Για να διαγραφούν οι τιμές που προηγούνται ή έπονται, μπορεί να οριστεί μια κενή γραμμή:

**PageTop[^]:**

**PageTop[\$]:**

Ο στόχος '\_' ορίζει μία εξ' ορισμού τιμή για αυτή τη λέξη κλειδί.

Παράδειγμα:

**YSize[\_]: 150**

**Options[\_]: growright, bits, nopercnt**

**WithPeak[\_]: ymw**

**Suppress[\_]: y**

**MaxBytes[\_]: 1250000**

Για να διαγραφεί η εξ' ορισμού τιμή αρκεί και πάλι μία κενή γραμμή:

**YLegend[\_]:**



# APACHE SERVER

Για να εγκαταστήσετε τον Apache στη θέση προεπιλογής /usr/local/etc/httpd, πρέπει να είστε ο superuser. Θα πρέπει να ξέρετε τον κωδικό πρόσβασης ρίζας της μηχανής στην οποία εγκαθιστάτε τον Apache. Βάλτε το αρχείο διανομής στον κατάλογο /tmp. Μια ακολουθία εντολών για να κάνει την εγκατάσταση από το /tmp μοιάζει με αυτή:

```
mkdirs /usr/local/etc (your system may use 'mkdir -p /usr/local/etc' instead)
```

```
cd /usr/local/etc
```

```
mv /tmp/apache_1.1.1.tar.gz /usr/local/etc
```

```
gzcat apache_1.1.1.tar.gz | tar -xf -
```

```
mv apache_1.1.1 httpd
```

Εάν το σύστημά σας δεν υποστηρίζει gzcat, μπορεί να πρέπει να κάνετε ένα gunzip – αντί γιαυτό.

Σημειώστε ότι ανάλογα με την έκδοση που εγκαθιστάτε, θα αφεθείτε με έναν κατάλογο αποκαλούμενο apache\_x.x.x. Τώρα πρέπει να έχετε ένα δέντρο κάτω από το httpd που περιέχει την πλήρη διανομή Apache. Αυτό το δέντρο καταλόγου περιέχει τα πρότυπα διαμόρφωσης και άλλους πόρους που oserver χρειάζεται για τη λειτουργία του. Να μην μετακινήσετε ή μετονομάσετε οποιοδήποτε από τα αρχεία και τους καταλόγους, εκτός αν ξέρετε τι κάνετε.

## Installing a Precompiled Server

Ο Apache έχει συνταχθεί για πολλές πλατφόρμες, και οι πιθανότητες είναι ότι ένα αντίγραφο prebuilt του server να υπάρχει για το setup σας.

Η εγκατάσταση ενός precompiled server είναι παρόμοια με την εγκατάσταση μιας source-only διανομής. Η μόνη διαφορά είναι ότι ένα httpd υπάρχει ήδη στον κατάλογο του server src. Η εγκαταστημένη βιβλιοθήκη του Apache server καλείται httpd. Οι διανομές αυτές ονομάζονται συνήθως httpd - η *αρχιτεκτονική*, όπου η *αρχιτεκτονική* είναι το όνομα του OS που εσείς έχετε εγκαταστήσει. Για το Linux, η httpd διανομή ονομάζεται httpd-linux.

Αν και είναι καλύτερα να κρατηθεί η διανομή στον κατάλογο src, θέλουμε να εγκαταστήσουμε τη διανομή στην κορυφή του δέντρου διανομής έτσι ακολουθούμε αυτό το παράδειγμα /usr/local/etc/httpd/httpd. Μερικές φορές, η διανομή δεν έχει αλαχθεί. Η αλλαγή αφαιρεί τις πρόσθετες πληροφορίες συμβόλων που ο συνδετικός εκδότης άφησε από τη διαδικασία σύνδεσης σύνταξης. Η αφαίρεση αυτών των πρόσθετων πληροφοριών κάνει συνήθως τη βιβλιοθήκη για 30 έως 40 τοις εκατό μικρότερη. Θέλουμε επίσης να προστατεύσουμε το εκτελέσιμο από οποιοδήποτε χρήστη που προσπαθεί να το αλλάξει. Ένας εύκολος τρόπος να γίνει αυτό είναι με την αλλαγή του ιδιοκτήτη και της ομάδας του εκτελέσιμου στη ρίζα.

Κατόπιν αφαιρούμε όλα τα προνόμια στη βιβλιοθήκη από τον καθένα εκτός του ιδιοκτήτη. Στο `chown` και `chmod` αρχείο (*change owner and change mode*) στη ρίζα, πρέπει να το κάνετε ως `superuser`. Η ακόλουθη εγκατάσταση μπορεί να γίνει κάπως έτσι :

1. `cd /usr/local/etc/httpd`
2. `cp src/httpd-linux httpd`
3. `strip httpd`
4. `chown root.wheel httpd`
5. `chmod 755 httpd`

## Installing from the Source

Η εγκατάσταση από την πηγή είναι λίγο πιο χρονοβόρα, αλλά θα σας βοηθήσει να εξοικειωθεί με τη διανομή πηγής `Apache server`. Θα σας δώσει επίσης την ευκαιρία να προσαρμόσετε τις ενότητες που χτίζονται σε `Apache`. Πολλές από αυτές τις ενότητες παρέχουν τη λειτουργία που θα καταστήσει τον κεντρικό υπολογιστή σας ισχυρότερο. Όλα τα αρχεία που θα πρέπει να εκδώσετε για να χτίσετε το δυαδικό είναι στον κατάλογο `src`. Η διαδικασία αυτή περιλαμβάνει 3 βήματα :

1. Edit the one-time configuration file, `src/Configuration`.
2. Run the `Configure` script.
3. Run `make`.

## Editing the Configuration File

Προτού να μπορέσετε να συντάξετε το λογισμικό, πρέπει να εκδώσετε το αρχείο και το σύνολο διαμόρφωσης μερικές συντάσσω-χρονικές επιλογές που εξετάζουν τις ιδιαιτερότητες του λειτουργικού συστήματός σας (και, σε μερικές περιπτώσεις, του υλικού στο οποίο πρόκειται να τρέξετε τον `server`). Ο `Apache` υποστηρίζει ένα σύνολο ενοτήτων που επεκτείνουν και ενισχύουν τη λειτουργία του `server`. Εντούτοις, για να μπορούν να δουλέψουν κάποιες από αυτές τις ενότητες θα πρέπει να ενεργοποιηθούν στο αρχείο διαμόρφωσης. Το αρχείο διαμόρφωσης μπορεί να βρεθεί μέσα στο `/usr/local/etc/httpd/src`. Για να κάνετε ένα αντίγραφο του αρχείου διαμόρφωσης γράψτε :

```
cd /usr/local/etc/httpd/src
```

```
cp Configuration.tmpl Configuration
```

Κράτησαμε όλες τις τοποθετήσεις προεπιλογής με εξαίρεση την αλλαγή των επιλογών μεταγλωττιστών και σύνταξης για να ταιριάξουμε με το υλικό και το λειτουργικό σύστημά μου. Αυτό εκτελείται με την αφαίρεση των σχολίων (the `# signs`) από το `AUX_CFLAGS` και το `AUX_LIBS` που υπάρχουν στο αρχείο

διαμόρφωσης. Θυμηθείτε να βάλετε στα σχόλια το a# μπροστά από άλλες τοποθετήσεις που μπορούν να μην ισχύσουν για τη διαμόρφωση συστημάτων σας.

## Running the Configure Script

Αφού εκδίδετε το αρχείο διαμόρφωσης, πρέπει να τρέξετε το Configure script. Το Configure script είναι ένα Bourne shell script που χρησιμοποιεί το configuration file που φτιάξατε (src/Configuration) για να δημιουργήσει ένα νέο Makefile και modules.c που να ταιριάζει στις επιλογές του configuration file που δώσαμε κατά την δημιουργία του. Για να κάνετε αυτό, γράψτε τα ακόλουθα στο τελικό πρόγραμμά σας:

```
cd /usr/local/etc/httpd/src
./Configure
```

Η διαμόρφωση σας επιτρέπει να τροποποιήσετε ένα αρχείο διαμόρφωσης στη γραμμή εντολής με τη χρησιμοποίηση -file flag:

```
Configure -file NEXTSTEP.configuration
```

```
Using alternate config file NEXTSTEP.configuration
```

Αυτό το χαρακτηριστικό γνώρισμα είναι χρήσιμο εάν πρέπει να διατηρήσετε τις διάφορες διαφορετικές διαμορφώσεις για αρκετές διαφορετικές μηχανές.

## Running make

Αν η διαδικασία διαμόρφωσης προχωράει κανονικά, πρέπει να είστε σε θέση να δημιουργήσετε την βιβλιοθήκη του server σε αυτό το σημείο. Για να γίνει αυτό γράψτε τα παρακάτω στο τερματικό σας :

```
make
cp httpd ../httpd
cd ..
strip httpd
chown root.wheel httpd
chmod 755 httpd
```

## Runtime Server Configuration Settings

Το Apache διαβάζει τις τοποθετήσεις διαμόρφωσής του από τρία αρχεία: access.conf, httpd.conf, και srm.conf. Πρώτιστα, αυτό έχει γίνει για να διατηρήσει την οπίσθια συμβατότητα με τον κεντρικό υπολογιστή NCSA, αλλά ο συλλογισμός πίσω από αυτήν την οργάνωση έχει μια λογική. Τα αρχεία διαμόρφωσης κατοικούν στο

subdirectory διασκέψεων της διανομής του server. Τα αρχεία διαμόρφωσης δειγμάτων που συμπεριλαμβάνονται στη διανομή λογισμικού ονομάζονται access.conf-dist, httpd.conf-dist, και srm.conf-dist, respectively. Θα πρέπει να δημιουργήσετε τα αντίγραφα αυτών των αρχείων χωρίς - dist μερίδα :

```
cd /usr/local/etc/httpd/conf
cp httpd.conf-dist httpd.conf
cp access.conf-dist access.conf
cp srm.conf-dist srm.conf
```

Αφού έχετε κάνει τα τοπικά αντίγραφα σας, είστε έτοιμοι να διαμορφώσετε έναν βασικό server.

Η διαμόρφωση χρόνου εκτέλεσης του κεντρικού υπολογιστή σας γίνεται μέσω οδηγίες διαμόρφωσης. Οι οδηγίες είναι εντολές που θέτουν κάποια επιλογή τους , χρησιμοποιήστε τις για να πείτε τον κεντρικό υπολογιστή για τις διάφορες επιλογές που θέλετε να επιτρέψετε, όπως η θέση των αρχείων σημαντικών στη διαμόρφωση και τη λειτουργία κεντρικών υπολογιστών. Οι οδηγίες διαμόρφωσης ακολουθούν αυτήν την σύνταξη:

*Directive option option...*

Οι οδηγίες είναι διευκρινισμένη μια ανά γραμμή. Μερικές οδηγίες θέτουν μόνο μια αξία όπως ένα όνομα αρχείου , άλλες σας αφήνουν να διευκρινίσετε τις διάφορες επιλογές. Υπάρχουν ειδικές οδηγίες, αποκαλούμενες *τμήματα* , τα οποία μοιάζουν με τις ετικέτες HTML. Οι οδηγίες τμημάτων περιβάλλονται από τα υποστηρίγματα γωνίας, όπως < οδηγία >. Τα τμήματα εσωκλείουν συνήθως μια ομάδα οδηγιών που ισχύουν μόνο για τον κατάλογο που διευκρινίζεται στο τμήμα:

```
<Directory somedir/in/your/tree>
    Directive option option
    Directive option option
</Directive>
```

Όλα τα τμήματα είναι κλειστά με μια ετικέτα τμημάτων ταιριάσματος που μοιάζει με < οδηγία >. Θα δείτε μερικά από αυτά τα κατασκευάσματα στα conf/access.conf και στα conf/ httpd.conf αρχεία σας. Σημειώστε ότι οι ετικέτες τμημάτων, όπως οποιαδήποτε οδηγία, είναι διευκρινισμένη μια ανά γραμμή.

## **Editing httpd.conf**

httpd.conf περιέχει τις οδηγίες διαμόρφωσης που ελέγχουν πώς ο κεντρικός υπολογιστής τρέχει, όπου τα αρχεία ημερολογίου του βρίσκονται, η ταυτότητα χρηστών (UID) αυτό τρέχει κάτω, το port που αναφέρετε, και τα λοιπά.. Κρατάμε τις περισσότερες από τις προεπιλογές που βρέθηκαν σε httpd.conf , με εξαίρεση την ακόλουθη:

ServerAdmin	The ServerAdmin directive should be set to the address of the webmaster managing the server. It should be a valid e-mail address or alias, such as <i>webmaster@your.domain</i> . Setting this to a valid address is important because this address will be returned to a visitor when there's a problem.
User and Group	The User and Group directives set the UID and group ID (GID) that the server will use to process requests. I kept these to the defaults: nobody and nogroup. Please verify that the names nobody and nogroup exist in your <i>/etc/passwd</i> and <i>/etc/group</i> files, respectively. If you want to use a different UID or GID, go ahead; however, be aware that the server will run with the permissions you define here. The permissions for the specified UID and GID should be very low because, in case of a security hole, whether on the server or (more likely) on your own CGI programs, those programs will run with the assigned UID. If the server runs as root or some other privileged user, someone may exploit the security holes and do nasty things to your site. Instead of specifying the User and Group directives using a name, you can specify them by using the UID and GID numbers. If you use numbers, be sure that the numbers you specify correspond to the user and group you want, and that they are preceded by the pound (#) symbol.

Εδώ βλέπουμε πως θα φαινόταν αν αλλάζαμε το όνομα :

```
User nobody
```

```
Group nogroup
```

Εδώ η ίδια προδιαγραφή, αλλά από UID και GID:

```
User #-1
```

```
Group #-1
```

ServerName	The ServerName directive sets the hostname the server will return. Set it to a fully qualified domain name (fqdn). If not set, the server will try figure it out by itself and set to its canonical name. However, you may want the server to return a friendlier address such as <i>www.your.domain</i> . Whatever you do, ServerName should be a real Domain Name System (DNS) name for your network. If you are administering your own DNS, remember to add a CNAME alias for your host. If someone else manages the DNS for you, ask her to do this for you. Your ServerName entry should look like this: ServerName <i>www.your.domain</i>
------------	--

ServerRoot	This directive sets the absolute path to your server directory (where your httpd binary is located). This directive tells the server where to find all the resources and configuration files. Many of these resources are specified in the configuration files relative of the ServerRoot directory. If you installed your server using my examples, your ServerRoot directive will read: ServerRoot /usr/local/etc/httpd
------------	---

### Editing srm.conf

Το αρχείο srm.conf είναι το αρχείο διαμόρφωσης των πόρων. Ελέγχει τις τοποθετήσεις σχετικές με τη θέση του δέντρου εγγράφων Ιστού σας, τους καταλόγους προγράμματος της CGI, και άλλα ζητήματα διαμόρφωσης των πόρων που έχουν επιπτώσεις στον ιστοχώρο σας. Κρατήσαμε τις περισσότερες από τις προεπιλογές που βρέθηκαν σε srm.conf. Οι σημαντικότερες οδηγίες για αυτό το αρχείο διαμόρφωσης είναι

DocumentRoot	Set this directive to the absolute path of your document tree. Your <i>document tree</i> is the top directory from which Apache will serve files. By default it is set to /usr/local/etc/httpd/htdocs.
--------------	--

Μπορείτε επίσης να θελήσετε να θέσετε την οδηγία UserDir στα άτομα με ειδικές ανάγκες. Η οδηγία UserDir καθορίζει το συγγενή κατάλογο του εγχώριου καταλόγου ενός τοπικού χρήστη όπου θα βάλουν τα δημόσια έγγραφα HTML. Είναι σχετικό επειδή κάθε χρήστης θα έχει έναν κατάλογο public\_HTML. Αν και αυτό μπορεί να δώσει στους χρήστες επί του τόπου σας την ελευθερία να δημιουργήσει και να διατηρήσει τις αρχικές σελίδες τους, πολλοί χρήστες μπορούν να μην γνωρίζουν αυτό το γεγονός, με συνέπεια τη δυνατότητα της δημιουργίας ενός καταλόγου public\_html που περιέχει τα ιδιωτικά αρχεία. Επιπλέον, ανάλογα με το είδος χρηστών στο σύστημά σας και τη διαμόρφωση κεντρικών υπολογιστών, αυτό μπορεί να δημιουργήσει τα πιθανά προβλήματα ασφάλειας. Θα το καθιστούσε πιθανό για έναν χρήστη να δημοσιεύσει το ολόκληρο filesystem σας με τη δημιουργία μιας συμβολικής σύνδεσης με έναν κατάλογο έξω από ένα δέντρο εγγράφων Ιστού. Με να κάνει αυτό, καθένας θα ήταν σε θέση να διαβάσει και να πλοηγήσει το filesystem σας ελεύθερα.

### Editing access.conf

Κρατήσαμε όλες τις προεπιλογές access.conf στο αρχείο μας. Διαμορφώνει τον τύπο χρηστών πρόσβασης που έχει στην περιοχή σας και τα έγγραφα που παρέχετε, καθώς επίσης και ζητήματα ασφάλειας

καθορίζοντας το βαθμό στον οποίο οι χρήστες μπορούν να αλλάξουν τις τοποθετήσεις ασφάλειας που μπορεί να είχατε καθορίσει. Η διαμόρφωση προεπιλογής παρέχει την απεριόριστη πρόσβαση στα έγγραφα στο DocumentRoot σας.

Εάν θέλετε να παρέχετε μια πιό περιοριστική περιοχή, μπορείτε να ρυθμίσετε όλα τα < Directory path > τμήματα καταλόγου να ταιριάζουν με τους καταλόγους που απαριθμούν στην εγκατάστασή σας. Τα τμήματα καταλόγου διευκρινίζουν ένα σύνολο επιλογών, περιλαμβάνοντας συνήθως τα ζητήματα ασφάλειας, σε μια βάση ανά-καταλόγου. Ειδικότερα, μπορείτε να θελήσετε να αφαιρέσετε την επιλογή δεικτών που ακολουθεί την οδηγία επιλογών για το τμήμα που μοιάζει με αυτό:

```
<Directory /usr/local/etc/httpd/cgi-bin>
```

```
Options Indexes FollowSymLinks
```

```
</Directory>
```

Η επιλογή δεικτών επιτρέπει τις υπολογιστής-παραγμένες λίστες καταλόγου. Πιθανώς δεν θέλετε κάποιον που κρυφοκοιτάζει στο περιεχόμενο των καταλόγων CGI-DOHEJ'WN σας.

Οι επιλογές που εφαρμόζετε στο σφαιρικό ACT σας μπορούν να αγνοηθούν με την χρήση ενός αρχείου .htaccess. Μπορείτε να θέσετε εκτός λειτουργίας όλες τις htaccess αγγελίες με τον καθορισμό της οδηγίας AllowOverride σε None , που είναι εξ ορισμού ρυθμισμένο να επιτρέπει ότι ο άλλος αγνοεί:

```
AllowOverride None
```

## Configuring an inetd Server

Κανονικά, ο Apache τρέχει σε standalone mode ή daemon mode . Πώς οργανώνεται από το σύστημα εξαρτάται από τον τρόπο με τον οποίο διαμορφώνεται από την οδηγία ServerType στο conf/httpd.conf.

Ένας *standalone server* προσφέρει ανώτερη απόδοση πάνω σε inetd-run servers επειδή συνήθως θα υπάρξει μια διαδικασία του server έτοιμη να εξυπηρετήσει ένα αίτημα. Όταν οργανώνεται κάτω από το inetd (the Internet daemon), ένας νέος server αρχίζει κάθε φορά που παραλαμβάνεται ένα αίτημα στο HTTP port. Υπάρχει ένα μη αμελητέο ποσό γενικών εξόδων στην έναρξη μιας νέας διαδικασίας ενός server με κάθε νέο αίτημα.

Η προεπιλογή που θέτει για ServerType είναι αυτόνομη, εκτός αν έχετε μια εξαιρετικά ελαφριά περιοχή κυκλοφορίας, πρέπει να κολλήσετε με αυτήν την ρύθμιση.

Για να τρέξετε έναν κεντρικό υπολογιστή από το inetd, θα πρέπει να τροποποιήσετε το conf/httpd.conf ακόμα μια φορά και να αλλάξετε την οδηγία ServerType από αυτόνομο στο inetd:

```
ServerType inetd
```

Η οδηγία των port δεν έχει καμία επίδραση σε έναν server inetd. Ένας αυτόνομος κεντρικός υπολογιστής χρησιμοποιεί αυτές τις πληροφορίες διαμόρφωσης για να μάθει το port. Επειδή inetd κάνει τη σύνδεση μεταξύ του port και του λογισμικού, αυτή η ρύθμιση δεν έχει καμία επίδραση σε μια διαμόρφωση inetd.

## Configuring inetd

inetd είναι το "internet superserver". Ενεργοποιήτε όταν ξεκινούν οι μηχανές από το /etc/rc. Όταν ενεργοποιηθεί ψάχνει να βρεί τα απαραίτητα ports για το internet.. Όταν βρίσκει μια σύνδεση, θέτει σε εφαρμογή το πρόγραμμα αρμόδιο για τη διαχείριση εκείνου του port. Όταν το αίτημα εξυπηρετείται, το inetd συνεχίζει να αφογκράζεται τα πρόσθετα αιτήματα σε εκείνο το port.

Για να δουλέψετε τον Apache με το inetd θα πρέπει να δημιουργήσετε το /etc/inetd.conf και /etc/services. Η διαμόρφωση ενός server inetd απαιτεί λίγο περισσότερη διαμόρφωση συστημάτων από έναν αυτόνομο κεντρικό υπολογιστή.

Κατ' αρχάς, πρέπει να εκδώσετε το /etc/services αρχείο. Η βάση /etc/services περιλαμβάνει πληροφορίες για όλες τις διαθέσιμες υπηρεσίες του Internet. Κάθε υπηρεσία αντιπροσωπεύεται από την ενιαία γραμμή που απαριθμεί τις ακόλουθες πληροφορίες:

Official service name

Port number

Protocol name

Aliases by which the service is known

Κάθε είσοδος χωρίζεται από μια ετικέτα ή τα διαστήματα. Μια είσοδος που περιγράφει httpd θα μοιάσει με αυτό:

```
http portnumber/tcp httpd httpd
```

Δηλώστε το portnumber στον αριθμό του port που θέλετε να τρέξετε τον server. Χαρακτηριστικά αυτό θα είναι λιμένας 80 για κεντρικούς υπολογιστές τους αυτόνομους servers. Οι inetd servers τρέχουν καλύτερα στο port 8080, έτσι η είσοδος σας θα μοιάσει με αυτό:

```
http 8080/tcp httpd httpd
```



Εάν τρέχετε NetInfo, μπορείτε να δακτυλογραφήσετε αυτήν την γραμμή σε ένα προσωρινό αρχείο, όπως /tmp/services, και τρέξτε το

```
niload services . < /tmp/services
```

Έπειτα, πρέπει να κάνετε το /etc/inetd.conf που διαμορφώνουν inetd για να αφουγκραστείτε τα httpd τα αιτήματα. Κάθε γραμμή σε inetd.conf περιέχει τις ακόλουθες πληροφορίες:

- Service name
- Socket type
- Protocol
- Wait/no wait
- User the server program will run as
- Server program
- Server program arguments

Η ολοκληρωμένη είσοδός μας μοιάζει με αυτό:

```
httpd stream tcp nowait nobody /usr/local/etc/httpd/httpd  
httpd -f /usr/local/etc/httpd/conf/httpd.conf
```

Αφού προσθέσετε το httpd στο /etc/inetd.conf, θα πρέπει να ξαναξεκινήσετε το inetd. Μπορείτε εύκολα να κάνετε αυτό με να ανακαλύψετε τον αριθμό διαδικασίας inetd με τα ps και την αποστολή του ενός σήματος HANGUP:

```
# kill -HUP InetdProcessID
```

Αντικαταστήστε το *InetdProcessID* με τον αριθμό διαδικασίας που απαριθμείται από την εντολή ps. Εάν το PID που απαριθμήθηκε ήταν 86, θα δακτυλογραφούσατε το kill -HUP 86.

inetd θα ξαναξεκινήσει, ξαναδιαβάζοντας το αρχείο διαμόρφωσής του που θα το καθοδηγήσει για να αφουγκραστεί το αίτημα για το port 8080.

## Running the Web Server for the First Time

Προτού να μπορέσετε να τρέξετε τον server για πρώτη φορά, πρέπει να δημιουργήσετε ένα έγγραφο HTML. Η τυποποιημένη διανομή Apache περιλαμβάνει ένα τέτοιο αρχείο, αλλά έχουμε δημιουργήσει άλλο που είναι πιά χρήσιμο και είμαστε βέβαιοι ότι θα χρησιμοποιήσετε επανειλημμένως. Χρησιμοποίηση του αγαπημένου συντάκτη κειμένων σας, δημιουργήστε ένα αρχείο αποκαλούμενο HTML μέσα στον κατάλογο htdocs με αυτό το περιεχόμενο:

```
<HTML>
```

```
<HEAD>
```

```
<TITLE>Apache Server Survival Guide</TITLE>

</HEAD>

<BODY BGCOLOR="#ffffff" LINK="#000080" VLINK="#000080">

<H1><CENTER>Apache Server Survival Guide </CENTER></H1>

<H2><CENTER>Congratulations! Your Apache server was
successfully

installed.</CENTER></H2>

<H3>Here are some interesting sites that host information
about

the Apache server: </H3>

<UL>

<LI>The official homepage for the

<A HREF="http://www.apache.org">Apache Group</A>

<LI>The official homepage for

<A HREF="http://www.us.apache-ssl.com">Community
Connexion</A>

developers of Stronghold: Apache-SSL-US (A Netscape
compatible

SSL server based on Apache)

<LI>The official homepage for

<A HREF="http://www.algroup.co.uk/Apache-SSL">Apache-
SSL</A>

(A Netscape compatible SSL server based on Apache - only
available

to users outside of the United States).

<LI><A
HREF="http://www.zyzyva.com/server/module_registry/">
Apache Module Registry</A>, the place where you can find
information

about 3<SUP>rd</SUP> party Apache modules and other
development stuff.

<LI><A HREF="http://www.apacheweek.com">The Apache Week
Home</A>,

here you will find an essential weekly guide dedicated to
Apache server nformation.
```

```
<LI><A HREF="http://www.ukweb.com">UK Web's Apache  
Support Center</A>
```

```
<LI><A HREF="http://www.fastcgi.com">The FastCGI  
Website</A>
```

```
</UL>
```

```
<P>
```

```
<STRONG>Deja News a very handy USENET news search  
engine:</STRONG>
```

```
<FORM ACTION="http://search.dejanews.com/dnquery.xp"  
METHOD=POST>
```

```
<P>
```

```
<CENTER>
```

```
<STRONG>Quick Search For:</STRONG> <INPUT NAME="query"  
VALUE="Apache" SIZE="37">
```

```
<INPUT TYPE="submit" VALUE="Search!"><INPUT  
NAME="defaultOp" VALUE="AND" TYPE="hidden">
```

```
<INPUT NAME="svcclass" VALUE="dncurrent" TYPE="hidden">
```

```
<INPUT NAME="maxhits" VALUE="20" TYPE="hidden">
```

```
</CENTER>
```

```
</FORM>
```

```
</BODY>
```

```
</HTML>
```

Βάλτε αυτό το αρχείο στον κατάλογο htdocs σας. Σε αυτό το σημείο είστε έτοιμοι να εξετάσετε τον server.

## Starting Up a Standalone Server

Εάν τρέχετε έναν αυτόνομο server , θα πρέπει να αρχίσετε httpd εσείς. Έτσι το κάνετε:

```
# /usr/local/etc/httpd/httpd -f  
/usr/local/etc/httpd/conf/httpd.conf
```

## Starting Up an inetd Server

Όπως υποθέσατε πιθανώς, δεν πρέπει να αρχίσετε έναν server inetd. inetd θα αρχίσει httpd κάθε φορά που παραλαμβάνεται ένα αίτημα στο port που ορίζεται στονserver. Οι inetd servers κάνουν την καλή

πλατφόρμα ανάπτυξης επειδή οι τοποθετήσεις διαμόρφωσης ξαναδιαβάζονται κάθε φορά που στέλνετε ένα αίτημα.

## Starting and Stopping the Server

Ο Apache server, httpd, έχει μερικές επιλογές εντολή-γραμμών που μπορείτε να χρησιμοποιήσετε για να θέσετε μερικές προεπιλογές που διευκρινίζουν όπου httpd θα διαβάσει τις οδηγίες διαμόρφωσής του. Το Apache httpd εκτελέσιμο καταλαβαίνει τις ακόλουθες επιλογές:

```
httpd [-d ServerRoot] [-f ConfigurationFile] [-x] [-v] [-  
?]
```

```
Usage: httpd [-d directory] [-f file] [-v]
```

```
-d directory : specify an alternate initial ServerRoot
```

```
-f file : specify an alternate ServerConfigFile
```

### The start Script

```
cd WhereEverYouEnterThem
```

```
chmod 755 start stop restart
```

Μπορείτε να θελήσετε να αποθηκεύσετε το χειρόγραφο σε ισχύ όπου είναι κατάλληλο, όπως /usr/local/bin. Εδώ η λίστα για το χειρόγραφο έναρξης:

```
#!/bin/sh
```

```
/usr/local/etc/httpd/httpd -f  
/usr/local/etc/httpd/conf/httpd.conf
```

### The stop Script

```
#!/bin/sh
```

```
#Stop Apache Script
```

```
kill 'cat /usr/local/etc/httpd/logs/httpd..pid'
```

### The restart Script

```
#!/bin/sh
```

```
#Restart Apache Script
```

```
kill -HUP 'cat /usr/local/etc/httpd/logs/httpd.pid'
```

## Auto-Starting the Web Server at Boot Time

Μπορείτε να ενδιαφερθείτε για την αυτοματοποίηση της διαδικασίας ξεκινήματος του server όποτε αρχίζετε το μηχανήμά σας. Να κάνει είναι έτσι πολύ εύκολο. Στο δικό σας /etc/rc.local αρχείο, προσθέστε μια περιοχή όπως αυτή :

```
#  
# Run httpd server automatically  
#  
echo 'starting httpd server'  
if [ -f /etc/rc.httpd ]; then  
    sh /etc/rc.httpd  
fi
```

Δημιουργήστε ένα νέο αρχείο, /etc/rc.httpd, με το ακόλουθο περιεχόμενο:

```
#!/bin/sh -u  
#  
cd /usr/local/etc/httpd/  
./httpd -d /usr/local/etc/httpd/ -f conf/httpd.conf
```

## Testing the Newly Installed Server

Εάν τα αρχεία διαμόρφωσής σας εκδόθηκαν κατάλληλα, πρέπει να είστε σε θέση να δείξετε τον browser σας τη μηχανή που τρέχει τον server και να δείτε τη σελίδα που δημιουργήσατε νωρίτερα, επιβεβαιώνοντας ότι ο server σας τρέχει.

Εάν υπήρξε ένα πρόβλημα με τον server ή μια διαμόρφωση, ο Apache μπορεί να σας δώσει μια ένδειξη για το πρόβλημα. Εάν ο Apache δεν μπορούσε να βρεί μερικά από τα αρχεία διαμόρφωσης, πρέπει να σας δώσει ένα μήνυμα όπως αυτό:

```
httpd: could not open document config file  
/usr/local/etc/httpd/conf/httpd.conf
```

```
fopen: No such file or directory
```

Ελένξτε για τον τύπο ή ανακριβείς πληροφορίες για τη γραμμή εντολής σας ή στα αρχεία διαμόρφωσης. Χαρακτηριστικά αυτό το πρόβλημα εμφανίζεται εάν έχετε προσαρμόσει την εγκατάστασή σας σε έναν διαφορετικό κατάλογο

Άλλα προβλήματα μπορούν να έχουν τις τιμές απορρίψεων όπως

```
httpd: could not bind to port
```

```
bind: Address already in use
```

Εάν λαμβάνετε αυτό το μήνυμα το πιθανότερο είναι μια εργασία να χρησιμοποιεί περισσότερα ports από αυτά που εσείς έχετε δηλώσει .

Άλλο πρόβλημα με τα port μπορεί να εμφανιστεί αν διαφοροποιήσετε τον αριθμό των port. Μόνο οι διαδικασίες ρίζας μπορούν να δεσμεύσουν τα διατηρημένα ports. Η λύση είναι να αρχιστεί ο server ως root έτσι ώστε ο server να μπορεί να δεσμεύσει το κατάλληλο port.

Εάν ο server δεν αποκρίνεται, ελέγξτε για να δείτε εάν τρέχει.

Χρησιμοποιήστε το πρόγραμμα ps για να φανεί τι γίνεται :

```
ps -guax | grep httpd (on BSD-type systems)
```

```
ps -ef | grep httpd (on SVR4-type systems)
```

Μην ανησυχήσετε αν δείτε το τρέξιμο διάφορων διαδικασιών httpd.

Το πρότυπο του Apache server τρέχει διάφορες ταυτόχρονες διαδικασίες που συντονίζονται με την κύρια διαδικασία

## Configuration File Listings

Για την ευκολία σας, εδώ μια λίστα των διάφορων αρχείων.

### src/Configuration File

Αφότου διαμορφώνετε και χτίζετε τον server σας, δεν θα είναι απαραίτητο να εκδώσετε αυτό το αρχείο. Τουλάχιστον μέχρι που επιτρέπετε τις πρόσθετες ενότητες. Θα πρέπει έπειτα να τους προσθέσετε εδώ, και να επαναλάβετε τη διαμόρφωση και τη δυαδική διαδικασία οικοδόμησης πάλι.

```
# Config file for the Apache httpd.
```

```
# Configuration.tmpl is the template for Configuration.  
Configuration should
```

```
# be edited to select system type. Configuration.tmpl  
should only be changed
```

```
# when a new system or module is added, or an existing  
one modified.
```

```
# There are three types of lines here:
```

```
# '#' comments, distinguished by having a '#' as the  
first non-blank character
```

```
#
```

```
# Lines which set a Make option --- these are simply
copied into the Makefile

#

# Module selection lines, distinguished by having
'Module' at the front.

# These list the configured modules, in priority order
(highest priority

# first). They're down at the bottom.

# First, ordinary compile-time configuration.

# What to call the compiler: For normal machines with
ANSI compilers

# CC= cc

# For Suns or other non-ANSI platforms. Please make sure
your gcc is

# 2.0 or later, as 1.40 seems to create bad code for the
Sun 4.

CC= gcc

# CFLAGS, compile flags.

# -DMINIMAL_DNS is now obsolete. Use httpd.conf settings
of

# HostnameLookups on

# or

# HostnameLookups off

#

# If you want to have more secure hostname resolution at
the cost of some

# performance, use -DMAXIMUM_DNS.

# If you want setting the xbit of a file to cause it to
be treated as

# server-included HTML (unless it is a CGI script), say -
DXBITHACK. Note

# that this is a run-time option, per-directory, either
way (via the XBITHACK

# command); this option only sets the default.

# If you find that your OS can't cope with mmap (compiles
OKAY but refuses
```

```
# to run and moans "httpd: Could not mmap memory" .. or
similar) try

# disabling use of shared memory for process management
(scoreboard with

# -DNO_MMAP

# Status Instrumentation

# In order for the status module to obtain full
statistics Apache must

# be modified to keep track of various information. This
is not

# turned on by default. In order to enable full status
details add -DSTATUS

# to the end of the CFLAGS line below.

# Using SOCKS

# Apache can be compiled to work over a SOCKS firewall by
# adding the following string to your CFLAGS define:

#

# -Dconnect=Rconnect -Dselect=Rselect -
Dgethostbyname=Rgethostbyname

#

# and by adding the following to the EXTRA_LIBS define:

#

# -L/usr/local/lib -lsocks

#

# making sure that -L points to wherever you've put
libsocks.a.

# [Some other former Apache compile-time options are now
treated differently;

# the virtual host code is always present; DBM auth is an
optional module, and

# may be configured out by changing the module config
below, though it still

# defaults in. Note that this config file does not
include DBM auth by

# default --- configure it in below if you need it].

CFLAGS= -O2
```



```
# Place here any flags you may need upon linking, such as
a flag to

# prevent dynamic linking (if desired)

LFLAGS=

# Place here any extra libraries you may need to link to.

# -lndbm is commonly required for DBM auth, if that is
configured in.

EXTRA_LIBS=

# AUX_CFLAGS are system-specific control flags.

# NOTE: IF YOU DO NOT CHOOSE ONE OF THESE, EDIT httpd.h
AND CHOOSE

# SETTINGS FOR THE SYSTEM FLAGS. IF YOU DON'T, BAD THINGS
WILL HAPPEN.

# For SunOS 4

#AUX_CFLAGS= -DSUNOS4

# For Solaris 2.

#AUX_CFLAGS= -DSOLARIS2

#AUX_LIBS= -lsocket -lnsl

# For SGI IRIX. Use the AUX_LIBS line if you're using NIS
and want

# user-supported directories

#AUX_CFLAGS= -DIRIX

#AUX_LIBS= -lsun

# For HP-UX n.b. if you use the paid-for HP CC compiler,
use flag -Ae

#AUX_CFLAGS= -DHPUX

# For AIX

#AUX_CFLAGS= -DAIX -U__STR__

# For Ultrix

#AUX_CFLAGS= -DULTRIX

# For DEC OSF/1

#AUX_CFLAGS= -DOSF1

# For NeXT

#AUX_CFLAGS= -DNEXT
```

```
# For Sequent

#AUX_CFLAGS= -DSEQUENT

# For Linux -m486 ONLY IF YOU HAVE 486 BINARY SUPPORT IN
KERNEL

#AUX_CFLAGS= -DLINUX

# For A/UX

#AUX_CFLAGS= -DAUX -D_POSIX_SOURCE

#AUX_LIBS= -lposix -lbsd -s

# For SCO ODT 3

# libcrypt_i available from sosco.sco.com, files
/SLS/lng225b.Z and

# /SLS/lng225b.ltr.Z

# the -Oe option causes cc to die compiling mod_imap
(using 3.0.0a of the dev sys)

#CFLAGS= -Oacgiltz

#AUX_CFLAGS= -DSCO

#AUX_LIBS= -lPW -lsocket -lmalloc -lcrypt_i

# For SCO OpenServer Release 5

# -K noline is needed to work around an optimiser bug
which appears in

# http_bprintf.c

#AUX_CFLAGS= -DSCO5

#AUX_LIBS=-lsocket -lmalloc -lprot

#BROKEN_BPRINTF_FLAGS=-K noline

# For SVR4

# Some SVR4 implementations will require SO_LINGER option
to be set in order

# to guarantee buffer flushes. Dell, Esix, and UnixWare
are a few of these.

# Use -DNEED_LINGER in addition to other AUX_CFLAGS for
these.

#AUX_CFLAGS= -DSVR4

#AUX_LIBS= -lsocket -lnsl -lc
```

```
# For UnixWare 2.x, no longer just SVR4 (sigh) - use cc,
not gcc

# AUX_LIBS= -lsocket -lnsl -lcrypt

# For Amdahl UTS 2.1

# -Xa enables ANSI mode, -eft is expanded types
#AUX_CFLAGS= -Xa -eft -DUTS21
#AUX_LIBS= -lsocket -lbsd -la

# For HP/Apollo Domain/OS
#AUX_CFLAGS= -DAPOLLO

# For NetBSD/FreeBSD/BSDI 2.x

# -m486 only if you are running on Intel 486/586
#AUX_CFLAGS= -m486

# BSDI doesn't need -lcrypt
#AUX_LIBS= -lcrypt

# For QNX
#AUX_CFLAGS= -DQNX
#AUX_LFLAGS= -N 0x20000

# For LynxOS
#AUX_CFLAGS= -DLYNXOS
#EXTRA_LIBS=-lbsd -ldes -lc_p

# For DG/UX 5.4
#AUX_CFLAGS= -DDGUX
#AUX_LIBS=

# For EMX OS/2 port
#AUX_CFLAGS= -Zbsd-signals -Zbin-files
#-DNO_KILLPG -DNEED_STRCASECMP -DNO_SETSID

#-g

#AUX_LIBS= -lsocket -llibufc -lgdbm -lbsd

#####
#####

# Module configuration

#
```

```
# Modules are listed in reverse priority order --- the
ones that come

# later can override the behavior of those that come
earlier. This

# can have visible effects; for instance, if UserDir
followed Alias,

# you couldn't alias out a particular user's home
directory.

# The configuration below is what we consider a decent
default

# configuration. If you want the functionality provided
by a particular

# module, remove the "#" sign at the beginning of the
line. But remember,

# the more modules you compile into the server, the
larger the executable

# is and the more memory it will take, so if you are
unlikely to use the

# functionality of a particular module you might wish to
leave it out.

## Basic modules (i.e., generally useful stuff that works
everyplace):

## You should probably not comment out any of these
unless you know what it

## does and you know you won't need it.

Module mime_module mod_mime.o

Module access_module mod_access.o

Module auth_module mod_auth.o

Module negotiation_module mod_negotiation.o

Module includes_module mod_include.o

Module dir_module mod_dir.o

Module cgi_module mod_cgi.o

Module userdir_module mod_userdir.o

Module alias_module mod_alias.o

Module env_module mod_env.o

Module common_log_module mod_log_common.o
```

```
## The asis module implemented ".asis" file types, which
allow the embedding

## of HTTP headers at the beginning of the document.
mod_imap handles internal

## imagemaps (no more cgi-bin/imagemap/!). mod_actions is
used to specify

## CGI scripts which act as "handlers" for particular
files, for example to

## automatically convert every GIF to another file type.

Module asis_module mod_asis.o

Module imap_module mod_imap.o

Module action_module mod_actions.o

## Optional modules for NCSA user-agent/referer logging
compatibility

## We recommend, however, that you migrate to the
configurable logging

## module, below.

# Module agent_log_module mod_log_agent.o

# Module referer_log_module mod_log_referer.o

## This is a *replacement* for mod_log_common which
supports a

## LogFormat directive which allows you to specify what
goes into

## the TransferLog (if you want Referer, etc.) source
code for docs.

##

## If you play with this, remember to drop the standard

## mod_log_common --- a server with both will work, but
you'll get

## very confused trying to figure out what's going on...

# Module config_log_module mod_log_config.o

## cern_meta mimicks the behavior of the CERN web server
with regards to

## metainformation files.

# Module cern_meta_module mod_cern_meta.o

## The status module allows the server to display current
details about
```

```
## how well it is performing and what it is doing.
Consider also enabling

## -DSTATUS (see the CFLAGS section near the start of the
file) to allow

## full status information. Check conf/access.conf on how
to enable this.

# Module status_module mod_status.o

## The Info module displays configuration information for
the server and

## all included modules. It's very useful for debugging.

# Module info_module mod_info.o

## Optional authentication modules.

##

## The anon_auth module allows for anonymous-FTP-style
username/

## password authentication.

# Module anon_auth_module mod_auth_anon.o

## db_auth and dbm_auth work with Berkeley DB files -
make sure there

## is support for DBM files on your system. You may need
to grab the GNU

## "gdbm" package if not.

# Module db_auth_module mod_auth_db.o

# Module dbm_auth_module mod_auth_dbm.o

## msql_auth checks against an MSQL database. You must
have MSQL installed

## and an "msql.h" available for this to even compile.
Additionally,

## you may need to add a couple entries to the CFLAGS
line, like

##

## -lmsql -L/usr/local/lib -L/usr/local/Minerva/lib

##

## This depends on your installation of MSQL.

# Module msql_auth_module mod_auth_msql.o
```

```
## "digest" implements HTTP Digest Authentication rather
than the less

## secure Basic Auth used by the other modules.

# Module digest_module mod_digest.o

## Outright experiments --- mod_dld defines commands
which

## allows other modules to be loaded in at runtime, and
mod_cookies

## uses Netscape cookies to automatically construct and
log accurate

## click-trails from Netscape cookies, for Netscape-using
clients who

## aren't coming in via proxy.

# Module dld_module mod_dld.o

# Module cookies_module mod_cookies.o

## Finally, the proxy module. It's not as complete as it
could

## be yet, so use at your own risk.

# Module proxy_module mod_proxy.o
```

## **The Server Configuration File: conf/httpd.conf**

```
# This is the main server configuration file. See URL
http://www.apache.org/

# for instructions.

# Do NOT simply read the instructions in here without
understanding

# what they do, if you are unsure consult the online
docs. You have been

# warned.

# Originally by Rob McCool

# ServerType is either inetd, or standalone.

ServerType standalone

# If you are running from inetd, go to "ServerAdmin".

# Port: The port the standalone listens to. For ports <
1023, you will
```

```
# need httpd to be run as root initially.

Port 80

# HostnameLookups: Log the names of clients or just their
IP numbers

# e.g. www.apache.org (on) or 204.62.129.132 (off)

HostnameLookups on

# If you wish httpd to run as a different user or group,
you must run

# httpd as root initially and it will switch.

# User/Group: The name (or #number) of the user/group to
run httpd as.

# On SCO (ODT 3) use User nouser and Group nogroup

User nobody

Group #-1

# ServerAdmin: Your address, where problems with the
server should be

# e-mailed.

ServerAdmin you@your.address

# ServerRoot: The directory the server's config, error,
and log files

# are kept in

ServerRoot /usr/local/etc/httpd

# BindAddress: You can support virtual hosts with this
option. This option

# is used to tell the server which IP address to listen
to. It can either

# contain "*", an IP address, or a fully qualified
Internet domain name.

# See also the VirtualHost directive.

#BindAddress *

# ErrorLog: The location of the error log file. If this
does not start

# with /, ServerRoot is prepended to it.

ErrorLog logs/error_log

# TransferLog: The location of the transfer log file. If
this does not
```



```
# start with /, ServerRoot is prepended to it.

TransferLog logs/access_log

# PidFile: The file the server should log its pid to

PidFile logs/httpd.pid

# ScoreBoardFile: File used to store internal server
process information

ScoreBoardFile logs/apache_status

# ServerName allows you to set a host name which is sent
back to clients for

# your server if it's different than the one the program
would get (i.e. use

# "www" instead of the host's real name).

#

# Note: You cannot just invent host names and hope they
work. The name you

# define here must be a valid DNS name for your host. If
you don't understand

# this, ask your network administrator.

#ServerName new.host.name

# CacheNegotiatedDocs: By default, Apache sends Pragma:
no-cache with each

# document that was negotiated on the basis of content.
This asks proxy

# servers not to cache the document. Uncommenting the
following line disables

# this behavior, and proxies will be allowed to cache the
documents.

#CacheNegotiatedDocs

# Timeout: The number of seconds before receives and
sends time out

# n.b. the compiled default is 1200 (20 minutes !)

Timeout 400

# KeepAlive: The number of Keep-Alive persistent requests
to accept

# per connection. Set to 0 to deactivate Keep-Alive
support
```

```
KeepAlive 5

# KeepAliveTimeout: Number of seconds to wait for the
next request

KeepAliveTimeout 15

# Server-pool size regulation. Rather than making you
guess how many

# server processes you need, Apache dynamically adapts to
the load it

# sees --- that is, it tries to maintain enough server
processes to

# handle the current load, plus a few spare servers to
handle transient

# load spikes (e.g., multiple simultaneous requests from
a single

# Netscape browser).

# It does this by periodically checking how many servers
are waiting

# for a request. If there are fewer than MinSpareServers,
it creates

# a new spare. If there are more than MaxSpareServers,
some of the

# spares die off. These values are probably OK for most
sites ---

MinSpareServers 5

MaxSpareServers 10

# Number of servers to start --- should be a reasonable
ballpark figure.

StartServers 5

# Limit on total number of servers running, i.e., limit
on the number

# of clients who can simultaneously connect --- if this
limit is ever

# reached, clients will be LOCKED OUT, so it should NOT
BE SET TOO LOW.

# It is intended mainly as a brake to keep a runaway
server from taking

# Unix with it as it spirals down...

MaxClients 150
```

```
# MaxRequestsPerChild: the number of requests each child
process is

# allowed to process before the child dies.

# The child will exit so as to avoid problems after
prolonged use when

# Apache (and maybe the libraries it uses) leak. On most
systems, this

# isn't really needed, but a few (such as Solaris) do
have notable leaks

# in the libraries.

MaxRequestsPerChild 30

# Proxy Server directives. Uncomment the following line
to

# enable the proxy server:

#ProxyRequests On

# To enable the cache as well, edit and uncomment the
following lines:

#CacheRoot /usr/local/etc/httpd/proxy

#CacheSize 5

#CacheGcInterval 4

#CacheMaxExpire 24

#CacheLastModifiedFactor 0.1

#CacheDefaultExpire 1

#NoCache adomain.com anotherdomain.edu joes.garage.com

# Listen: Allows you to bind Apache to specific IP
addresses and/or

# ports, in addition to the default. See also the
VirtualHost command

#Listen 3000

#Listen 12.34.56.78:80

# VirtualHost: Allows the daemon to respond to requests
for more than one

# server address, if your server machine is configured to
accept IP packets

# for multiple addresses. This can be accomplished with
the ifconfig
```

```
# alias flag, or through kernel patches like VIF.

# Any httpd.conf or srm.conf directive may go into a
VirtualHost command.

# See also the BindAddress entry.

#<VirtualHost host.foo.com>

#ServerAdmin webmaster@host.foo.com

#DocumentRoot /www/docs/host.foo.com

#ServerName host.foo.com

#ErrorLog logs/host.foo.com-error_log

#TransferLog logs/host.foo.com-access_log

#</VirtualHost>
```

## **The Resource Configuration File: conf/srm.conf**

```
# With this document, you define the name space that
users see of your http

# server. This file also defines server settings which
affect how requests are

# serviced, and how results should be formatted.

# See the tutorials at http://www.apache.org/ for
# more information.

# Originally by Rob McCool; Adapted for Apache

# DocumentRoot: The directory out of which you will serve
your

# documents. By default, all requests are taken from this
directory, but

# symbolic links and aliases may be used to point to
other locations.

DocumentRoot /usr/local/etc/httpd/htdocs

# UserDir: The name of the directory which is appended
onto a user's home

# directory if a ~user request is recieved.

UserDir public_html
```

```
# DirectoryIndex: Name of the file or files to use as a
pre-written HTML

# directory index. Separate multiple entries with spaces.

DirectoryIndex index.html

# FancyIndexing is whether you want fancy directory
indexing or standard

FancyIndexing on

# AddIcon tells the server which icon to show for
different files or filename

# extensions

AddIconByEncoding (CMP,/icons/compressed.gif) x-compress
x-gzip

AddIconByType (TXT,/icons/text.gif) text/*

AddIconByType (IMG,/icons/image2.gif) image/*

AddIconByType (SND,/icons/sound2.gif) audio/*

AddIconByType (VID,/icons/movie.gif) video/*

AddIcon /icons/binary.gif .bin .exe

AddIcon /icons/binhex.gif .hqx

AddIcon /icons/tar.gif .tar

AddIcon /icons/world2.gif .wrl .wrl.gz .vrml .vrm .iv

AddIcon /icons/compressed.gif .Z .z .tgz .gz .zip

AddIcon /icons/a.gif .ps .ai .eps

AddIcon /icons/layout.gif .html .shtml .htm .pdf

AddIcon /icons/text.gif .txt

AddIcon /icons/c.gif .c

AddIcon /icons/p.gif .pl .py

AddIcon /icons/f.gif .for

AddIcon /icons/dvi.gif .dvi

AddIcon /icons/uuencoded.gif .uu

AddIcon /icons/script.gif .conf .sh .shar .csh .ksh .tcl

AddIcon /icons/tex.gif .tex

AddIcon /icons/bomb.gif core
```

```
AddIcon /icons/back.gif ..

AddIcon /icons/hand.right.gif README

AddIcon /icons/folder.gif ^^DIRECTORY^^

AddIcon /icons/blank.gif ^^BLANKICON^^

# DefaultIcon is which icon to show for files which do
not have an icon

# explicitly set.

DefaultIcon /icons/unknown.gif

# AddDescription allows you to place a short description
after a file in

# server-generated indexes.

# Format: AddDescription "description" filename

# ReadmeName is the name of the README file the server
will look for by

# default. Format: ReadmeName name

#

# The server will first look for name.html, include it if
found, and it will

# then look for name and include it as plaintext if
found.

#

# HeaderName is the name of a file which should be
prepended to

# directory indexes.

ReadmeName README

HeaderName HEADER

# IndexIgnore is a set of filenames which directory
indexing should ignore

# Format: IndexIgnore name1 name2...

IndexIgnore */.*? *~ *# */HEADER* */README* */RCS

# AccessFileName: The name of the file to look for in
each directory

# for access control information.

AccessFileName .htaccess
```

```
# DefaultType is the default MIME type for documents
which the server

# cannot find the type of from filename extensions.

DefaultType text/plain

# AddEncoding allows you to have certain browsers
(Mosaic/X 2.1+) uncompress

# information on the fly. Note: Not all browsers support
this.

AddEncoding x-compress Z

AddEncoding x-gzip gz

# AddLanguage allows you to specify the language of a
document. You can

# then use content negotiation to give a browser a file
in a language

# it can understand. Note that the suffix does not have
to be the same

# as the language keyword --- those with documents in
Polish (whose

# net-standard language code is pl) may wish to use
"AddLanguage pl .po"

# to avoid the ambiguity with the common suffix for perl
scripts.

AddLanguage en .en

AddLanguage fr .fr

AddLanguage de .de

AddLanguage da .da

AddLanguage el .el

AddLanguage it .it

# LanguagePriority allows you to give precedence to some
languages

# in case of a tie during content negotiation.

# Just list the languages in decreasing order of
preference.

LanguagePriority en fr de

# Redirect allows you to tell clients about documents
which used to exist in
```

```
# your server's namespace, but do not anymore. This
allows you to tell the

# clients where to look for the relocated document.

# Format: Redirect fakename url

# Aliases: Add here as many aliases as you need (with no
limit). The format is

# Alias fakename realname

#Alias /icons/ /usr/local/etc/httpd/icons/

# ScriptAlias: This controls which directories contain
server scripts.

# Format: ScriptAlias fakename realname

#ScriptAlias /cgi-bin/ /usr/local/etc/httpd/cgi-bin/

# If you want to use server side includes, or CGI outside

# ScriptAliased directories, uncomment the following
lines.

# AddType allows you to tweak mime.types without actually
editing it, or to

# make certain files to be certain types.

# Format: AddType type/subtype ext1

# AddHandler allows you to map certain file extensions to
"handlers",

# actions unrelated to filetype. These can be either
built into the server

# or added with the Action command (see below)

# Format: AddHandler action-name ext1

# To use CGI scripts:

#AddHandler cgi-script .cgi

# To use server-parsed HTML files

#AddType text/html .shtml

#AddHandler server-parsed .shtml

# Uncomment the following line to enable Apache's send-
asis HTTP file

# feature

#AddHandler send-as-is asis

# If you wish to use server-parsed imagemap files, use
```



```
#AddHandler imap-file map

# To enable type maps, you might want to use

#AddHandler type-map var

# Action lets you define media types that will execute a
script whenever

# a matching file is called. This eliminates the need for
repeated URL

# pathnames for oft-used CGI file processors.

# Format: Action media/type /cgi-script/location

# Format: Action handler-name /cgi-script/location

# For example to add a footer (footer.html in your
document root) to

# files with extension .foot (e.g. foo.html.foot), you
could use:

#AddHandler foot-action foot

#Action foot-action /cgi-bin/footer

# Or to do this for all HTML files, for example, use:

#Action text/html /cgi-bin/footer

# MetaDir: specifies the name of the directory in which
Apache can find

# meta information files. These files contain additional
HTTP headers

# to include when sending the document

#MetaDir .web

# MetaSuffix: specifies the file name suffix for the file
containing the

# meta information.

#MetaSuffix .meta

# Customizable error response (Apache style)

# these come in three flavors

#

# 1) plain text

#ErrorDocument 500 "The server made a boo boo.

# n.b. the (") marks it as text, it does not get output
```

```
#

# 2) local redirects

#ErrorDocument 404 /missing.html

# to redirect to local url /missing.html

#ErrorDocument 404 /cgi-bin/missing_handler.pl

# n.b. can redirect to a script or a document using
server-side-includes.

#

# 3) external redirects

#ErrorDocument 402
http://other.server.com/subscription_info.html

#
```

## **The Global Access Configuration File: conf/access.conf**

```
# access.conf: Global access configuration

# Online docs at http://www.apache.org/

# This file defines server settings which affect which
types of services

# are allowed, and in what circumstances.

# Each directory to which Apache has access, can be
configured with respect

# to which services and features are allowed and/or
disabled in that

# directory (and its subdirectories).

# Originally by Rob McCool

# This should be changed to whatever you set DocumentRoot
to.

<Directory /usr/local/etc/httpd/htdocs>

# This may also be "None", "All", or any combination of
"Indexes",

# "Includes", "FollowSymLinks", "ExecCGI", or
"MultiViews".

# Note that "MultiViews" must be named *explicitly* ---
"Options All"
```

```
# doesn't give it to you (or at least, not yet).

Options Indexes FollowSymLinks

# This controls which options the .htaccess files in
directories can

# override. Can also be "All", or any combination of
"Options", "FileInfo",

# "AuthConfig", and "Limit"

AllowOverride None

# Controls who can get stuff from this server.

order allow,deny

allow from all

</Directory>

# /usr/local/etc/httpd/cgi-bin should be changed to
whatever your ScriptAliased

# CGI directory exists, if you have that configured.

<Directory /usr/local/etc/httpd/cgi-bin>

AllowOverride None

Options None

</Directory>

# Allow server status reports, with the URL of
http://servername/status

# Change the ".nowhere.com" to match your domain to
enable.

#<Location /status>

#SetHandler server-status

#order deny,allow

#deny from all

#allow from .nowhere.com

#</Location>

# You may place any other directories or locations you
wish to have

# access information for after this one.
```

# ACTIVE STATE PERL

Το Perl είναι μια υψηλού επιπέδου γλώσσα προγραμματισμού με μια εκλεκτική κληρονομιά που γράφτηκε από τον Larry Wall. Παράγει από την πανταχού παρούσα γλώσσα προγραμματισμού C και σε μια μικρότερη έκταση από το sed, awk, το shell Unix, και τουλάχιστον δώδεκα άλλα εργαλεία και γλώσσες. Η διαδικασία του Perl, το αρχείο, και οι εγκαταστάσεις χειρισμού κειμένων το καθιστούν ιδιαίτερα καλοταίριασμένο για τους στόχους που περιλαμβάνουν τη γρήγορη διαμόρφωση πρωτοτύπου, τις χρησιμότητες συστημάτων, τα εργαλεία λογισμικού, τους διοικητικούς στόχους συστημάτων, την πρόσβαση βάσεων δεδομένων, το γραφικό προγραμματισμό, τη δικτύωση, και τον προγραμματισμό World Wide Web. Αυτές οι δυνάμεις το καθιστούν ιδιαίτερα δημοφιλές και χρησιμοποιήστε από τους Administrators και τα CGI, αλλά και τους μαθηματικούς, τους γενετιστές, δημοσιογράφους, και managers .

Ο αρχικός πολιτισμός του προ-λαϊκίστικου Διαδικτύου και οι βαθιά-κρατημένες πεποιθήσεις του συντάκτη του Perl, Larry, προκάλεσαν την ελεύθερη και ανοικτή πολιτική διανομής του perl. Το Perl υποστηρίζεται από τους χρήστες του. Ο πυρήνας, η τυποποιημένη βιβλιοθήκη Perl, οι προαιρετικές ενότητες, και η τεκμηρίωση που διαβάζετε τώρα ήταν όλοι που γράφτηκαν από τους εθελοντές. Ειδικότερα, η ομάδα ανάπτυξης πυρήνων (γνωστή ως Perl Porters) είναι μια ομάδα ιδιαίτερα αλτρουιστικών ατόμων δεσμευμένων στην παραγωγή του καλύτερου λογισμικού δωρεάν οπου ο καθένας θα μπορούσε να ελπίζει να αγοράσει για τα χρήματα.

Ενώ το πρόγραμμα GNU περιλαμβάνει Perl στις διανομές του, δεν εμφανίζεται ως "GNU Perl". Το Perl δεν παράγεται ούτε διατηρείται από το ελεύθερο ίδρυμα λογισμικού. Οι όροι χορήγησης αδειών του Perl είναι επίσης ανοικτότεροι από αυτό που το GNU τείνει να γίνει .

Πρέπει σίγουρα να χρησιμοποιήσετε την έκδοση 5. Η έκδοση 4 είναι παλαιά, περιορισμένη, και όχι άλλο διατηρημένη , η τελευταία ενημέρωσή του έγινε το (4.036) 1992, πολύ καιρό πριν. Βέβαια, είναι σταθερό, αλλά όχι και σύγχρονο. Η πιο πρόσφατη έκδοση είναι το 5.005\_03 (αν και το 5.004\_05 υποστηρίζεται ακόμα). Η περισσότερο απελευθερωμένη έκδοση είναι η 5.005\_57. Οι περαιτέρω αναφορές στη γλώσσα Perl στο παρόν έγγραφο αναφέρονται στην απελευθέρωση παραγωγής με την επιφύλαξη τυχόν διαφορετικής ρύθμισης. Όλες οι απελευθερώσεις πριν από 5.004 υπόκειντο σε buffer overruns , ένα σοβαρό ζήτημα ασφάλειας.

Οι απελευθερώσεις παραγωγής, που ενσωματώνουν τις διορθώσεις των κενών ασφαλείας, εξετάζονται ευρέως πριν από την απελευθέρωση. Δεδομένου ότι για το 5.000 , έχουν υπολογίσει κατά μέσο όρο μόνο περίπου μια απελευθέρωση παραγωγής ετησίως.

Ο Larry και η ομάδα ανάπτυξης Perl κάνουν περιστασιακά τις αλλαγές στον εσωτερικό πυρήνα της γλώσσας, αλλά όλες οι πιθανές προσπάθειες καταβάλλονται προς την οπίσθια συμβατότητα. Ενώ όλα τα χειρόγραφα perl4 τρέχουν άψογα κάτω από perl5, μια αναπροσαρμογή στο perl δεν πρέπει σχεδόν ποτέ να ακυρώσει ένα πρόγραμμα που γράφεται για μια προηγούμενη έκδοση του perl.

Το Perl είναι εύκολο να το μάθεις και να συνεχίσεις να το χρησιμοποιήσεις και σε πιο εξηδικευμένες εργασίες . Μοιάζει με τις περισσότερες γλώσσες προγραμματισμού που είναι πιθανό να έχετε εμπειρία, έτσι εάν έχετε γράψει ένα

πρόγραμμα στη C, ένα script awk, ένα shell script, ή ακόμα και ένα πρόγραμμα BASIC, είστε ήδη στο σωστό δρόμο για την εκτέλεση του perl.

Οι περισσότεροι στόχοι απαιτούν μόνο ένα μικρό υποσύνολο της γλώσσας Perl. Ένα από τα καθοδηγώντα ρητά για την ανάπτυξη Perl είναι `` υπάρχουν περισσότεροι από ένας τρόποι να γίνει ``. Η εκμάθηση του Perl είναι επομένως ρηχή (εύκολο στο να το μάθεις) και μακρά (φτάνει κάποιος να έχει όρεξη να ασχοληθεί με κάτι πιο πολύπλοκο).

Τέλος, επειδή το Perl είναι συχνά (αλλά όχι πάντα, και βεβαίως όχι εξ ορισμού) μια ερμηνευμένη γλώσσα, μπορείτε να γράψετε τα προγράμματά σας και να τα εξετάσετε χωρίς ένα ενδιάμεσο βήμα σύνταξης, επιτρέποντας να πειραματιστείτε και να εξετάσετε/διορθώσετε γρήγορα και εύκολα. Αυτή η ευκολία του πειραματισμού κάνει ακόμα πιο εύκολη την εκμάθηση.

## Basic syntax overview

Ένα χειρόγραφο Perl ή ένα πρόγραμμα αποτελείται από μια ή περισσότερες δηλώσεις. Αυτές οι δηλώσεις γράφονται απλά στο χειρόγραφο σε μια απλή φόρμα. Δεν υπάρχει καμία ανάγκη να υπάρξει μια `main()` ή τίποτα εκείνου του είδους.

Οι εντολές του perl τελειώνουν με ερωτηματικό:

```
print "Hello, world";
```

Τα σχόλια αρχίζουν με δίεση και τρέχουν στο τέλος της γραμμής

```
# This is a comment
```

Το κενό είναι άσχετο

```
print
    "Hello, world"
    ;
```

εκτός και αν είναι μέσα σε άγκιστρα :

```
# this would print with a linebreak in the middle
print "Hello
world";
```

Μπορούν να χρησιμοποιηθούν και μονά και διπλά «αυτάκια»

```
print "Hello, world";
```

```
print 'Hello, world';
```

Εντούτοις, μόνο τα διπλά «αυτάκια» "παρεμβάλλουν" τις μεταβλητές και τους ειδικούς χαρακτήρες όπως οι αλλαγές γραμμής (`\n`):

```
print "Hello, $name\n";      # works fine
```

```
print 'Hello, $name\n';    # prints $name\n literally
```

Οι αριθμοί δεν χρειάζονται τα «αυτάκια» γύρω τους

```
print 42;
```

Μπορείτε να χρησιμοποιήσετε τις παρενθέσεις για τα επιχειρήματα των λειτουργιών ή να τις παραλείψετε σύμφωνα με την προσωπική προτίμησή σας. Πρέπει να μπαίνουν μόνο περιστασιακά για να διευκρινίσουν τα ζητήματα της προτεραιότητας.

```
print("Hello, world\n");
```

```
print "Hello, world\n";
```

## Perl variable types

Το Perl έχει τρεις κύριους μεταβλητούς τύπους: scalars, arrays, και hashes.

### Scalars

Το scalars αντιπροσωπεύει μια ενιαία αξία

```
my $animal = "camel";
```

```
my $answer = 42;
```

Οι κλιμακωτές τιμές μπορούν να είναι σειρές, ακέραιοι αριθμοί ή αριθμοί κινητής υποδιαστολής, το Perl θα τις μετατρέψει αυτόματα όπως απαιτείται. Δεν υπάρχει καμία ανάγκη να προ-δηλωθούν οι μεταβλητοί τύποι σας.

Οι κλιμακωτές τιμές μπορούν να χρησιμοποιηθούν με διάφορους τρόπους:

```
print $animal;  
print "The animal is $animal\n";
```

```
print "The square of $answer is ", $answer * $answer, "\n";
```

Υπάρχουν διάφορα "μαγικά" scalars με τα ονόματα που μοιάζουν με το θόρυβο στίξης ή γραμμών. Στην αρχή αυτό που πρέπει να γνωρίζουμε είναι η "μεταβλητή προεπιλογής". Χρησιμοποιείται ως επιχειρήμα προεπιλογής σε διάφορες λειτουργίες του Perl, και τίθεται σιωπηρά από ορισμένα loop.

```
print; # prints contents of $_ by default
```

### Arrays

Μια σειρά αντιπροσωπεύει έναν κατάλογο τιμών:

```
my @animals = ("camel", "llama", "owl");  
my @numbers = (23, 42, 69);
```

```
my @mixed = ("camel", 42, 1.23);
```

Οι σειρές μηδέν-συντάσσονται. Εδώ πώς παίρνετε στα στοιχεία σε μια σειρά

```
print $animals[0];          # prints "camel"
```

```
print $animals[1];        # prints "llama"
```

Το ειδικό μεταβλητό  `$#array` σας λέει το δείκτη του τελευταίου στοιχείου μιας σειράς:

```
print $mixed[$#mixed];    # last element, prints 1.23
```

Μπορείτε να χρησιμοποιήσετε  `$#array + 1` για να σας πει πόσα στοιχεία υπάρχουν σε μια σειρά. Μην ενοχλήστε. Όπως συμβαίνει, χρησιμοποιώντας  `@array` το Perl αναμένει να βρεί ότι μια κλιμακωτή αξία ("στο κλιμακωτό πλαίσιο") θα σας δώσει τον αριθμό στοιχείων στη σειρά:

```
if (@animals < 5) { ... }
```

Τα στοιχεία που παίρνουμε από την έναρξη σειράς με ένα  `$` επειδή παίρνουμε ακριβώς μια ενιαία αξία από τη σειρά -- ζητάτε έναν κλιμακωτό, παίρνετε έναν κλιμακωτό.

Για να πάρει τις πολλαπλάσιες τιμές από μια σειρά:

```
@animals[0,1];           # gives ("camel", "llama");
@animals[0..2];          # gives ("camel", "llama", "owl")
@animals[1..$#animals];  # gives all except the first element
```

Αυτό καλείται "array slice".

Μπορείτε να κάνετε τα διάφορα χρήσιμα πράγματα στους καταλόγους:

```
my @sorted = sort @animals;
my @backwards = reverse @numbers;
```

Υπάρχουν μερικές ειδικές σειρές επίσης, όπως  `@ARGV` (τα επιχειρήματα γραμμών εντολής στο χειρόγραφο σας) και  `@_` (τα επιχειρήματα που περνούν σε μια υπορουτίνα).

## Hashes

Hash αντιπροσωπεύει ένα σύνολο ζευγαριών κλειδιών/αξίας:

```
my %fruit_color = ("apple", "red", "banana", "yellow");
```

Μπορείτε να χρησιμοποιήσετε whitespace και = > για να εμφανιστούν πιά ωραία

```
my %fruit_color = (  
    apple => "red",  
    banana => "yellow",  
  
    );
```

Για να πάρουμε στοιχεία από το hash:

```
$fruit_color{"apple"};      # gives "red"
```

Μπορείτε να πάρετε λίστες από κλειδιά και καταλόγους με keys() και values()

```
my @fruits = keys %fruit_colors;  
my @colors = values %fruit_colors;
```

Hashes δεν έχουν καμία ιδιαίτερη εσωτερική διαταγή, αν και μπορείτε να ταξινομήσετε τα κλειδιά και το βρόχο μέσω τους.

Ακριβώς όπως τα ειδικές scalars και τις σειρές, υπάρχουν επίσης ειδικά hashes. Τα πιο γνωστά είναι %ENV που περιέχει τις μεταβλητές περιβάλλοντος.

Οι πιά σύνθετοι τύποι στοιχείων μπορούν να κατασκευαστούν χρησιμοποιώντας τις αναφορές, οι οποίες επιτρέπουν σε σας να χτίσετε καταλόγους και hashes μέσα στους καταλόγους και hashes.

Μια αναφορά είναι μια κλιμακωτή αξία και μπορεί να αναφερθεί σε οποιοδήποτε άλλοδήποτε τύπο στοιχείων Perl. Έτσι με την αποθήκευση μιας αναφοράς ως αξία μιας σειράς ή hash ενός στοιχείου, μπορείτε εύκολα να δημιουργήσετε τους καταλόγους και hashes μέσα στους καταλόγους και hashes. Το ακόλουθο παράδειγμα παρουσιάζει hash 2 επιπέδων hash της δομής χρησιμοποιώντας τις ανώνυμες hash αναφορές.

```
my $variables = {  
    scalar => {  
        description => "single item",  
        sigil => '$',  
    },  
    array => {  
        description => "ordered list of items",  
        sigil => '@',  
    },  
    hash => {  
        description => "key/value pairs",  
        sigil => '%',  
    },  
};
```

```
print "Scalars begin with a $variables->{'scalar'}->{'sigil'}\n";
```

## [Variable scoping](#)



Σε όλο το προηγούμενο τμήμα όλα τα παραδείγματα έχουν χρησιμοποιήσει τη σύνταξη:

```
my $var = "value";
```

Το `my` πραγματικά δεν απαιτείται θα μπορούσατε ακριβώς να χρησιμοποιήσετε:

```
$var = "value";
```

Εντούτοις, η ανωτέρω χρήση θα δημιουργήσει τις σφαιρικές μεταβλητές σε όλο το πρόγραμμά σας, το οποίο είναι κακή πρακτική προγραμματισμού. Οι μεταβλητές είναι στο φραγμό (δηλ. μια δέσμη των δηλώσεων που περιβάλλεται από τα curly-braces) στον οποίο καθορίζονται.

```
my $a = "foo";
  if ($some_condition) {
    my $b = "bar";
    print $a;          # prints "foo"
    print $b;          # prints "bar"
  }
  print $a;           # prints "foo"

  print $b;          # prints nothing; $b has fallen out of scope
```

Χρησιμοποιώντας το `my` μαζί με `use strict` στην κορυφή των χειρογράφων Perl σας σημαίνει ότι ο διερμηνέας θα πάρει ορισμένα κοινά λάθη προγραμματισμού. Όπως στο παραπάνω παράδειγμα το τελικό `print $b` θα προκαλούσε ένα συντάσσω-χρονικό λάθος και θα σας απέτρεπε από το τρέξιμο του προγράμματος. Η χρησιμοποίηση `strict` συστήνεται ιδιαίτερα.

## [Conditional and looping constructs](#)

Το perl έχει τις περισσότερες από τις ευρέως χρησιμοποιούμενες τακτικές εκτός από `case/switch` (αλλά αν πραγματικά τις θέλετε υπάρχει η δυνατότητα αυτή στο perl 5.8 και στα νεότερα αλλά και στο CPAN).

Οι όροι μπορούν να είναι οποιαδήποτε έκφραση Perl. Δείτε τον κατάλογο χειριστών στο επόμενο τμήμα των πληροφοριών για τη σύγκριση και των `boolean logic operators`, ο οποίος χρησιμοποιείται συνήθως στις υπό όρους `if`

```
if ( condition ) {
    ...
} elsif ( other condition ) {
    ...
} else {
    ...
}

}
```

Υπάρχει επίσης μια αρνημένη έκδοση αυτού :

```
unless ( condition ) {  
    ...  
  
}
```

Αυτό παρέχεται ως πιο αναγνώσιμη έκδοση `if (!condition)`.

```
# the traditional way  
if ($zippy) {  
    print "Yow!";  
}  
  
# the Perlsh post-condition way  
print "Yow!" if $zippy;  
  
print "We have no bananas" unless $bananas;
```

## while

```
while ( condition ) {  
    ...  
  
}
```

Υπάρχει επίσης μια αρνημένη έκδοση, για τον ίδιο λόγο που έχουμε `unless`:

```
until ( condition ) {  
    ...  
  
}
```

Μπορείτε επίσης να χρησιμοποιήσετε το `while` σε post-condition

```
print "LA LA LA\n" while 1;    # loops forever
```

## for

Ακριβώς όπως στη C:

```
for ($i=0; $i <= $max; $i++) {  
    ...  
  
}
```

Το στυλ της C για λoop είναι απαραίτητο στο perlαπο τότε που αυτό δίνει την δυνατότητα για πιο φιλικό σκανάρισμα μες foreach loop .

## foreach

```
foreach (@array) {  
    print "This element is $_\n";  
}
```

```

}

# you don't have to use the default $_ either...
foreach my $key (keys %hash) {
    print "The value of $key is $hash{$key}\n";
}

```

## Builtin operators and functions

Το Perl έρχεται με μια ευρεία επιλογή των λειτουργιών builtin . Μερικές από αυτές τις έχουμε ήδη δει όπως print, sort and reverse.

### Arithmetic

```

+  addition
-  subtraction
*  multiplication
/  division

```

### Numeric comparison

```

==  equality
!=  inequality
<   less than
>   greater than
<=  less than or equal
>=  greater than or equal

```

### String comparison

```

eq  equality
ne  inequality
lt  less than
gt  greater than
le  less than or equal
ge  greater than or equal

```

### Boolean logic

```

&&  and
||  or
!   not

```

### Miscellaneous

```

=   assignment
.   string concatenation
x   string multiplication
..  range operator (creates a list of numbers)

```

Πολλοί χειριστές μπορούν να συνδυαστούν με a= ως εξής:

```
$a += 1;          # same as $a = $a + 1
$a -= 1;          # same as $a = $a - 1
$a .= "\n";      # same as $a = $a . "\n";
```

## Files and I/O

Μπορείτε να ανοίξετε ένα αρχείο για την εισαγωγή ή την παραγωγή χρησιμοποιώντας την `open()` λειτουργία .

```
open(INFILE, "input.txt") or die "Can't open input.txt: $!";
open(OUTFILE, ">output.txt") or die "Can't open output.txt: $!";
open(LOGFILE, ">>my.log") or die "Can't open logfile: $!";
```

Μπορείτε να διαβάσετε από ένα ανοικτό `filehandle` χρησιμοποιώντας το χειριστή `<>`. Στο κλιμακωτό πλαίσιο διαβάζει μια ενιαία γραμμή από το `filehandle`, και στο πλαίσιο καταλόγων διαβάζει ολόκληρο το αρχείο μέσα, ορίζοντας κάθε γραμμή σε ένα στοιχείο του καταλόγου:

```
my $line = <INFILE>;
my @lines = <INFILE>;
```

Η ανάγνωση σε ολόκληρο το αρχείο συγχρόνως καλείται `slurping`. Μπορεί να είναι χρήσιμο αλλά μπορεί να είναι καθι καταχραστής μνήμης .

Ο χειριστής `<>` συχνότερα φαίνεται σε ένα `while loop`:

```
while (<INFILE>) { # assigns each line in turn to $_
    print "Just read in this line: $_";
}
```

Έχουμε δει ήδη πώς να τυπώσουμε στην τυποποιημένη παραγωγή χρησιμοποιώντας το `print()`. Εντούτοις, το `print()` μπορεί επίσης να πάρει ένα προαιρετικό πρώτο επιχείρημα που διευκρινίζει ποιους `filehandle` θα εκτυπώσει :

```
print STDERR "This is your final warning.\n";
print OUTFILE $record;
print LOGFILE $logmessage;
```

When you're done with your `filehandles`, you should `close()` them (though to be honest, Perl will clean up after you if you forget):

Όταν τελειώσετε με τα `filehandles` θα πρέπει να τα κλείσετε αν και το κάνει αυτόματα το πρόγραμμα αν το ξεχάσετε.

```
close INFILE;
```

# Regular expressions

## Simple matching

```
if (/foo/)      { ... } # true if $_ contains "foo"  
if ($a =~ /foo/) { ... } # true if $a contains "foo"
```

## Simple substitution

```
s/foo/bar/;      # replaces foo with bar in $_  
$a =~ s/foo/bar/; # replaces foo with bar in $a  
$a =~ s/foo/bar/g; # replaces ALL INSTANCES of foo with bar in $a
```

## More complex regular expressions

```
.          a single character  
  \s       a whitespace character (space, tab, newline)  
  \S       non-whitespace character  
  \d       a digit (0-9)  
  \D       a non-digit  
  \w       a word character (a-z, A-Z, 0-9, _)  
  \W       a non-word character  
  [aeiou]  matches a single character in the given set  
  [^aeiou] matches a single character outside the given set  
set  
  (foo|bar|baz) matches any of the alternatives specified  
  ^        start of string  
  
  $        end of string
```

Το Quantifiers μπορεί να χρησιμοποιηθεί για να διευκρινίσει πόσους από το προηγούμενο πράγμα θέλετε να ταιριάξετε με επάνω, όπου "το πράγμα" σημαίνει είτε έναν κυριολεκτικό χαρακτήρα, ένα από τα metacharacters που απαριθμούνται ανωτέρω, είτε μια ομάδα χαρακτήρων είτε metacharacters σε παρένθεση.

```
*          zero or more of the previous thing  
+          one or more of the previous thing  
?          zero or one of the previous thing  
{3}       matches exactly 3 of the previous thing  
{3,6}     matches between 3 and 6 of the previous thing  
  
{3,}     matches 3 or more of the previous thing
```

## Μερικά συνοπτικά παραδείγματα

```
/^d+/      string starts with one or more digits  
/^$/      nothing in the string (start and end are  
adjacent)  
/(\d\s){3}/ a three digits, each followed by a whitespace  
            character (eg "3 4 5 ")  
/(a.)+/   matches a string in which every odd-numbered  
letter    is a (eg "abacadaf")
```

```

# This loop reads from STDIN, and prints non-blank lines:
while (<>) {
    next if /^$/;
    print;
}

```

## Parentheses for capturing

Όπως και την ομαδοποίηση, οι παρενθέσεις εξυπηρετούν έναν δεύτερο σκοπό. Μπορούν να χρησιμοποιηθούν για να συλλάβουν τα αποτελέσματα των μερών της αντιστοιχίας regexr για την πιό πρόσφατη χρήση. Τα αποτελέσματα καταλήγουν σε \$1, \$2 και τα λοιπά.

```

# a cheap and nasty way to break an email address up into parts

if ($email =~ /([^\@])+@(.+)/) {
    print "Username is $1\n";
    print "Hostname is $2\n";
}

```

## Writing subroutines

Το γράψιμο των υπορουτινών είναι εύκολο:

```

sub log {
    my $logmessage = shift;
    print LOGFILE $logmessage;
}

```

Τα επιχειρήματα σε μια υπορουτίνα είναι διαθέσιμα σε μας δεδομένου ότι μια ειδική σειρά κάλεσε @\_. Το επιχειρήμα προεπιλογής στη λειτουργία μετατόπισης συμβαίνει ακριβώς να είναι @\_. Άρα το my \$logmessage = shift; μετατοπίζει το πρώτο στοιχείο από τον κατάλογο επιχειρημάτων και το ορίζει \$logmessage .

Μπορούμε να χειριστούμε @\_ με άλλους τρόπους επίσης:

```

my ($logmessage, $priority) = @_;           # common
my $logmessage = $_[0];                    # uncommon, and ugly

```

Οι υπορουτίνες μπορούν επίσης να επιστρέψουν τις τιμές:

```

sub square {
    my $num = shift;
    my $result = $num * $num;
    return $result;
}

```

## **ΠΗΓΕΣ**

<http://mrtg.hdl.com/mrtg.html>

<http://www.activestate.com/>

<http://www.apache.org/>

<http://www.appservnetwork.com/>