

11. NIS - Network Information Service
oder Yellow Pages unter UNIX
=====

Problem:

Ein Rechner in einem lokalen Netzwerk benutzt viele Konfigurationsfiles, deren Inhalt sich bei der Erweiterung oder Veränderung des Netzwerkes aber ändert. Diese Konfigurationsfiles sind auf den meisten Rechner eines Netzwerkes nahezu identisch. Z.B.:

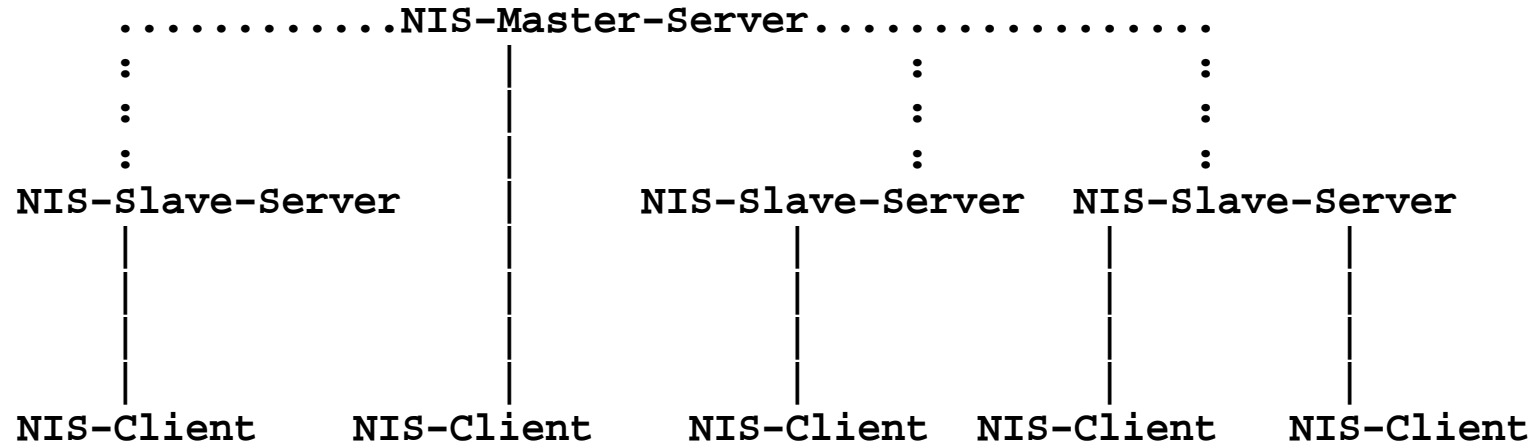
```
/etc/hosts      /etc/netmasks  /etc/networks
/etc/services   /etc/ethers     /etc/protocols
/etc/rpc        /etc/passwd     /etc/group
```

Es wäre also zweckmäßig diese Files zentral zu verwalten und sie dann zu verteilen. Dadurch wird erstens der Verwaltungsaufwand reduziert und zweitens sichergestellt, daß die Rechner eines lokalen Netzwerkes immer über der selben Datenbasis arbeiten.

Dies wird mit Hilfe von Maps durch das NIS bezüglich einer Domain realisiert.

NIS ursprünglich als Yellow Pages (YP) bezeichnet. Aus lizenzrechtlichen Gründen in Network Information Service umbenannt.

Struktur des NIS



Verteilung der Maps:

Beantwortung von Anfragen: _____

NIS-Master-Server: erzeugt aus Quellen Maps, verteilt Maps an
NIS-Slave-Server, beantwortet Fragen

NIS-Slave-Server: bantwortet Fragen nach Inhalten von Maps

NIS-Client: Stellt Fragen nach Inhalten von Maps

Jeder NIS-Server kann auch gleichzeitig NIS-Client sein!!!

NIS-Maps

NIS-Maps sind DBM-Files, die mittels des Kommandos "makedbm" aus den Quelltextfiles gebildet werden. Die Informationen der Maps ersetzen oder ergänzen die Informationen der entsprechenden Files der NIS-Clients.

NIS-Maps können verschieden eingebunden werden:

1. ersetzt das lokale File (rep.)
2. ergänzt das lokale File (app.)

Dies ist von der Map beziehungsweise von der Implementation abhängig. Bei modernen Systemen wird die Einbindung über /etc/nsswitch.conf gesteuert.

Private Maps sind möglich.

Jede Map besteht aus einem .dir- und einem .pag-File.

Umwandeln eines Files in eine Map:

```
/usr/sbin/makedbm host >host.byname
```

Umwandeln einer Map in ein File

```
/usr/sbin/makedbm -u host.byname > hosts
```

Map-Name	Nickname	Suchbegriff	Quelle	Bemerk.
auto.home		Mount-Point	/etc/auto.home	rep.
auto.master		Directory	/etc/auto.master	rep.
bootparams		Hostname	/etc/bootparams	app.
ethers.byaddr	ethers	Hostname	/etc/ethers	repl.
ethers.byname		MAC-Addr.	/etc/ethers	repl.
group.bygid	group	Gruppenname	/etc/group	app.
group.byname		Gruppen-ID	/etc/group	app.
hosts.byaddr	hosts	Hostname	/etc/hosts	rep.
hosts.byname		IP-Addr.	/etc/hosts	rep.
mail.aliases	aliases	Alias-Name	/etc/aliases	app.
mail.byaddr		exp. Alias	/etc/aliases	app.
netgroup.byhost		Hostname	/etc/netgroup	rep.
netgroup.byuser		Username	/etc/netgroup	rep.
netid.byname		Username	UID+GID	new.
netmasks.byaddr		IP-Adresse	/etc/netmasks	rep.
networks.byaddr		IP-Adresse	/etc/networks	rep.
networks.byname		Network-Name	/etc/networks	rep.
passwd.byname	passwd	Username	/etc/passwd	app.
passwd.byuid		UID	/etc/passwd	app.
protocols.byname		Protocol-Nr.	/etc/protocols	rep.
protocols.bynumber	protocols	Port Nummer	/etc/protocols	rep.
publickey.byname		User	/etc/publickey	app.
rpc.bynumber		RPC-Nummer	/etc/rpc	rep.
services.byname	services	Service-Name	/etc/services	rep.
timezone.byname		Zonen-Name	/etc/timezone	rep.
ypservers		Servername	---	--
ypslaves		Slaveservername	---	--

Sicherheit und NIS

NIS-Server sind geschwätzig. Im Normalfall geben sie einem Fremden alle Informationen, wenn dieser nur den NIS-Domainnamen kennt.

Sicherheitseinrichtungen:

1. Spezieller Portmapper mit TCPD-Funktionalität

Sicherheit wird über die Files

/etc/hosts.allow

```
rpcbind: 141.20.20.0/255.255.255.0
```

```
rpcbind: 141.20.20.1/255.255.255.0
```

```
ALL: LOCAL .informatik.hu-berlin.de
```

/etc/hosts.deny

```
rpcbind: ALL :( /usr/sbi/safe_finger -l @%h | /usr/ucb/mail -s \  
    "`hostname`-%d-%h-%a" bell@informatik.hu-berlin.de ) &
```

```
ALL: ALL :( /usr/sbi/safe_finger -l @%h | /usr/ucb/mail -s \  
    "`hostname`-%d-%h-%a" bell@informatik.hu-berlin.de ) &
```

gegeben.

2. NIS-Server mit TCPD-Funktionalität

Sicherheit wird wie bei 1. hergestellt

3. NIS-Server mit "securenets" Unterstützung

Durch das File:

/var/yp/securenets

werden die zugriffsberechtigten Netzwerke/Hosts bestimmt

Eintrag:

```
<Netzmaske> <Netzadresse>
```

```
255.255.255.0 141.20.20.0
```

```
255.255.255.0 141.20.21.0
```

NIS-Installieren

1. Bestimmung der Topologie:

- Pro Netzsegment mindestens 1 Server
- Nicht mehr als 40 Clienten pro Server
- Masterserver sollte sicher sein

2. Festlegen des Domainnamen

- ist nicht notwendig identisch mit DNS-Domainname
- sollte nicht leicht erratbar sein

3. NIS-Masterserver einrichten

- Kommando "domainname" von Hand abarbeiten

Syntax: domainname [name-of-domain]

z.B.: domainname informatik

- Kommando: "ypinit" von Hand abarbeiten

Syntax: ypinit -m

z.B.: ypinit -m

Bei einigen Systemen wird jetzt im Dialog nach den NIS-Slave-Servern gefragt.

- Kommando "ypserv" starten

Syntax:

ypserv [-dv] [-i | -I] [-r | -R]

-d - DNS benutzen für mehr Hostinformationen

iIrR - für Verbindung zum Ldap bei Solaris

Logging: /var/yp/ypserv.log wenn File existiert

- Kommando "ypbind" starten

Syntax:

```
ypbind [-s] [-ypset|-ypsetme|-S domainname,host1,host2,...]
```

```
-s          - benutze reservierten Port
```

```
-ypset      - Kommando "ypset" für Remote Hosts erlaubt
```

```
-ypsetme    - Kommando "ypset" nur lokal erlaubt
```

z.B.:

```
/usr/etc/ypbind
```

```
/usr/etc/ypbind -ypsetme
```

```
/usr/sbin/ypbind -S informatik,delta,epsilon1,master
```

- Modifizieren der Startup-Scripte

Kommando "domainname" einfügen

Kommando "ypserv" einfügen

Kommando "ypbind" einfügen

SunOS:

File "/etc/defaultdomain" anlegen, alles andere dann automatisch. In rc.local steht:

```
#
# set NIS domainname if locally configured.
if [ -f /etc/defaultdomain ]; then
    domainname `cat /etc/defaultdomain`
fi
```

....

```
dname=`domainname`
if [ "$dname" -a -d /var/yp ]; then
    echo "NIS domainname is $dname"
    echo -n "starting NIS services:"
    if [ -f /usr/etc/ypserv -a -d /var/yp/$dname ]; then
        ypserv;          echo -n ' ypserv'
#       Master NIS server runs the XFR daemon
        ypxfrd;         echo -n ' ypxfrd'
    fi
    if [ -f /etc/security/passwd.adjunct ]; then
        ypbind -s;      echo -n ' ypbind'
    else
        ypbind -ypsetme; echo -n ' ypbind -ypsetme'
        /usr/etc/yp/ypset bellus; echo -n ' ypset bellus'
    fi
    if [ -f /usr/etc/rpc.yupdated -a -d /var/yp/$dname ]
    then
        rpc.yupdated;  echo -n ' ypupdated'
    fi
    echo '.'
fi
```

DEC-UNIX:

1. 1.NIS konfigurieren mit /usr/sbin/nissetup
2. NIS mit /usr/lib/yp/ypinit konfigurieren und Werte in /etc/rc.config eintragen/modifizieren:
NIS_CONF="YES"
export NIS_CONF
NIS_TYPE="SLAVE"
export NIS_TYPE
NIS_DOMAIN="informatik"
export NIS_DOMAIN
NIS_ARGS="-S informatik,delta,epsilon1,bellus"
export NIS_ARGS
2. /etc/svc.conf aktualisieren

HP-UX:

1. NIS mit /usr/lib/yp/ypinit konfigurieren und /etc/netnfsrc modifizieren:

NIS_MASTER_SERVER=0
NIS_SLAVE_SERVER=1
NIS_CLIENT=1

NISDOMAIN=informatik
##

Solaris bis 9:

NIS konfigurieren mit:

`/usr/sbin/ypinit [-c] [-m] [-s master_server]`

Programme:

`/usr/lib/netsvc/yp/ypserv``/usr/lib/netsvc/yp/ypbind`

Datenfiles:

`/var/yp/binding/<domain>` existiert (notwendig)

enthält Namen der NIS-Server sonst wir broadcast gemacht

`/var/yp/<domain>/` - enthaelt NIS-Maps`/etc/defaultdomain` - enthält NIS-Domainname (notwendig)`/etc/nsswitch.conf`

Scripte beim Start:

`/usr/lib/netsvc/yp/ypstart``/etc/rc2.d/S69inet`**Solaris 10:**

Konfiguration, Programme und Datenfiles wie bei Solaris 9

Aktivieren:

`svcadm enable svc:/network/rpc/nis/server:default``svcadm enable svc:/network/rpc/nis/passwd:default``svcadm enable svc:/network/rpc/nis/update:default``svcadm enable svc:/network/rpc/nis/xfr:default`

Linux:

NIS Konfigurieren mit:

`/usr/lib/yp/ypinit [-m] [-s master]`

Programme:

`/usr/sbin/ypserv [-b | --dns] [-d] [-p port] [-v]``/usr/sbin/ypbind [-v] [-d] [-ypset] [-ypsetme]`

Datenfiles:

`/etc/nsswitch.conf``/etc/ypserv.conf``dns [yes|no]``xfr_check_port [yes|no]``host <IP-Adresse>``/etc/yp.conf``/etc/sysconfig/ypserv``YPPWD_SRCDIR=/etc``YPPWD_CHFN="no"``YPPWD_CHSH="no"``/etc/sysconfig/ypbind``/etc/sysconfig/network/config``NIS-Sektion`

```
/var/yp/securenets
# securenets
#
#       This file defines the access rights to your NIS server
#       for NIS clients. This file contains netmask/network
#       pairs. A clients IP address needs to match with at least
#       one of those.
#       One can use the word "host" instead of a netmask of
#       255.255.255.255. Only IP addresses are allowed in this
#       file, not hostnames.
# Always allow access for localhost
255.0.0.0          127.0.0.0
# This line gives access to everybody. PLEASE ADJUST!
0.0.0.0           0.0.0.0
```

```
/etc/yp.conf
# /etc/yp.conf - ypbind configuration file
# Valid entries are
#domain NISDOMAIN server HOSTNAME
# Use server HOSTNAME for the domain NISDOMAIN.
#domain NISDOMAIN broadcast
# Use broadcast on the local net for domain NISDOMAIN
#ypserver HOSTNAME
# Use server HOSTNAME for the local domain. The
# IP-address of server must be listed in /etc/hosts.
```

Scripte beim Start:

```
/etc/rc.d/init.d/ypserv
/etc/rc.d/init.d/ypbind
```

4. NIS-Slaveserver einrichten

- Kommando "domainname" von Hand abarbeiten

Syntax:

```
domainname [ name-of-domain ]
```

z.B.:

```
domainname informatik
```

- starten von "ypbind", wenn Slaveserver nicht im gleichen Segment wie Masterserver/anderer Slaveserver

```
ypbind -ypset
```

```
/usr/etc/yp/ypset -d informatik bellus
```

Syntax:

```
ypset [-V1|-V2] [-d domain] [-h host] server
```

- Kommando: "ypinit" von Hand abarbeiten

Syntax:

```
ypinit -s Masterserver
```

z.B.:

```
ypinit -s bellus
```

Bei einigen Systemen wird jetzt im Dialog nach Überschreibungsmöglichkeiten gefragt. Maps werden geholt und nach /var/yp/domainname kopiert

- Kommando "ypserv" starten

z.B.:

```
/usr/lib/netsvc/yp/ypserv
```

- Kommando "ypbind" starten falls Masterserver/Slaveserver im gleichen Segment

z.B.:

```
/usr/lib/netsvc/yp/ypbind
```

```
/usr/lib/netsvc/yp/ypbind -ypsetme
```

```
/usr/lib/netsvc/yp/ypbind -S informatik,delta,master
```

- Skripte modifizieren, analog Masterserver

5. NIS-Client einrichten

- Kommando "domainname" von Hand abarbeiten
z.B.:
 domainname informatik
- ypinit -c
- Kommando "ypbind" von Hand abarbeiten
- Skripte modifizieren, analog Masterserver (nur domainname und ypbind)

6. Besondere Maps

ypservers - enthält alle NIS-Server, wird benutzt für
Verteilung der Maps

anzeige mit:

```
ypcat -k ypserver
enterprise enterprise
maxwell maxwell
epsilon epsilon
euklid euklid
bellus bellus
vogon vogon
sigma sigma
master master
delta delta
curie curie
adler adler
```

extrahieren als Datenbasis:

```
# /usr/sbin/makedbm -u ypservers > nisservers
```

oder

```
# /usr/sbin/makedbm -u ypservers
YP_LAST_MODIFIED 0867400159
YP_MASTER_NAME bellus
enterprise enterprise
maxwell maxwell
epsilon epsilon
euklid euklid
master master
vogon vogon
sigma sigma
hp832 hp832
delta delta
curie curie
adler adler
```

neue Map erzeugen aus File nisservers:

```
# /usr/sbin/makedbm nisservers ypservers
```


`ypslaves.hostname` - enthält alle NIS-Slaveserver, wird benutzt für die Verteilung der Maps

Anzeigen mit:

```
# ypcat -k ypslaves.master
enterprise enterprise
maxwell maxwell
epsilon epsilon
euklid euklid
vogon vogon
sigma sigma
hp832 hp832
delta delta
curie curie
adler adler
```

Datenbasis modifizieren wie bei "ypservers":

```
/usr/sbin/makedbm -u ypslaves.master > nisslaves
```

oder

```
/usr/sbin/makedbm -u ypslaves.master > nisslaves
```

neue Map erzeugen aus File nisslaves:

```
/usr/sbin/makedbm nisslaves ypslaves.master
```

Integration von NIS-Maps in Files

1. /etc/passwd

":::0:0:::" am Ende des passwd-Files: alle Eintragungen der passwd-Map werden übernommen und Nutzer können sich einloggen
+:NOLOGIN:0:0:::/bin/false am Ende des passwd-Files: alle Eintragungen der passwd-Map werden übernommen und Nutzer können sich nicht einloggen
+@rbg: Nutzer der Nutzergruppe rbg werden aus der passwd-Map übernommen
+bell: Nutzer bell aus der Map - ja
-bell: Nutzer bell aus der Map - nicht
+@umx500: Nutzer der Nutzergruppe umx500 werden aus der passwd-Map übernommen
-: am Ende des passwd-Files: keine weiteren Eintragungen aus der passwd-Map werden übernommen.

Reihenfolge ist wichtig!!!

2. /etc/group

+ am Ende: alle Eintragungen aus der group-Map werden übernommen

3. /etc/bootparams

analog /etc/group

4. /etc/aliases

analog /etc/group

5. /etc/netgroup

File zur Bildung von Nutzergruppen (/etc/passwd) und Hostgruppen (/etc/hosts).
Jeder Zeile hat eines der folgenden Formate:

```
<gruppenname>    <gruppenname1>    <gruppenname2> ...
<gruppenname>    <member1>      <member2>      <member3> ...
<gruppenname>    <gruppenname1> ... <member1>    <member2> ....
<member>         (<hostname>,<username>,<domainname>)
```

Fehlt ein Eintrag, wird dies wie ein "Wildcard" gewertet,
d.h., es sind alle entsprechenden Eintragungen zulässig.
z.B. (,bell,) - bell von allen Hosts und allen Domainen

Beispiel /etc/netgroup

```
# computer-goups
#
alpha (beta,,) (gamma,,) (delta,,) (epsilon,,) (delta1,,) \
      (epsilon1,,) (beta1,,) (gamma1,,) (omega,,) (omega1,,) \
      (sigma,,)
sun   (hahn,,) (amalie,,) (harvey,,) (hamy1,,) (nirwana,,)\
      (data,,) (bellus,,) (marita,,) (wolf,,) (henrik,,)
...
allfbinf mx5 dec5 hp8 hp7 mx3 sun pc apple czs czs28 ibm6000 \
          nex hyundai alpha lind1 lind4 lind5 hp715 \
          hp715rok sune pce etlinks slips pcoverw \
          instbib sunpool mast
```

```
#
# computer-user-groups
#
uhep700  rbg unixsoft compsoft simulant
usun1    rbg unixsoft rbg-tutor
umx500   rbg rbg-tutor compsoft mi87 mi88 mi89 \
         unixsoft mi90 mi902 mi911 mi91 simulant petrinet \
         wissen schalter guests guest96 guest97 effalg \
         verwalt matheneb statist projekte mi92 inf-ass \
         signal mi93 mi93-ws parallel rokrbg \
         entwurf rok nebenf nebenf95 nebenf96 neben961 nebenf97 \
         datbank mi94 mi941 mi95 mi96 mi961 mi97\
         bildung schueler schuel96 schuel97 lehrer92 lehrer93\
         lehrer95 lehrer96 lehrer97
uhp715   umx500
uhp715r  rbg rok rokrbg
umast    rok rokrbg
#
# usergroups
#
rbg      (,rbg0,) (,bell,) (,gandre,) (,kaempfer,) (,wozobule,)\
         (,baerwolf,) (,accmgr,)
unixsoft (,burkhard,) (,polze,) (,redlich,) (,x400,)\
         (,hgsystem,) (,wischnow,)
rbg-tutor \
         (,pletat,) (,materna,) (,fhesse,) (,ftpsoft,) (,news,)\
         (,texadm,)
.....
```

Bildung von Maps, wenn die Quellen geändert wurden

```
cd /var/yp
make
```

Beispiel für Makefile (Linux)

```
#
# Makefile for the NIS databases
#
# This Makefile should only be run on the NIS master server of a domain.
# All updated maps will be pushed to all NIS slave servers listed in the
# /var/yp/ypservers file. Please make sure that the hostnames of all
# NIS servers in your domain are listed in /var/yp/ypservers.
#
# This Makefile can be modified to support more NIS maps if desired.
#
# Set the following variable to "-b" to have NIS servers use the domain
# name resolver for hosts not in the current domain. This is only needed,
# if you have SunOS slave YP server, which gets here maps from this
# server. The NYS YP server will ignore the YP_INTERDOMAIN key.
#B=-b
B=

# If we have only one server, we don't have to push the maps to the
# slave servers (NOPUSH=true). If you have slave servers, change this
# to "NOPUSH=false" and put all hostnames of your slave servers in the file
# /var/yp/ypservers.
NOPUSH=true
```

```
# We do not put password entries with lower UIDs (the root and system entries)
# in the NIS password database, for security. MINUID is the lowest uid that
# will be included in the password maps. If you create shadow maps, the
# UserID for a shadow entry is taken from the passwd file. If no entry is
# found, this shadow entry is ignored.
# MINGID is the lowest gid that will be included in the group maps.
MINUID=100
MINGID=100
# Should we merge the passwd file with the shadow file ?
# MERGE_PASSWD=true|false
MERGE_PASSWD=true
# Should we merge the group file with the gshadow file ?
# MERGE_GROUP=true|false
MERGE_GROUP=true
# These are commands which this Makefile needs to properly rebuild the
# NIS databases. Don't change these unless you have a good reason.
AWK = /usr/bin/gawk
MAKE = /usr/bin/gmake
UMASK = umask 066
# These are the source directories for the NIS files; normally
# that is /etc but you may want to move the source for the password
# and group files to (for example) /var/yp/ypfiles. The directory
# for passwd, group and shadow is defined by YPPWDDIR, the rest is
# taken from YPSRCDIR.
YPSRCDIR = /etc
YPPWDDIR = $(shell . /etc/sysconfig/ypserv; echo $$YPPWD_SRCDIR)
YPBINDIR = /usr/lib/yp
YPSBINDIR = /usr/sbin
YPDIR = /var/yp
YPMAPDIR = $(YPDIR)/$(DOMAIN)
```

```
# These are the files from which the NIS databases are built. You may edit
# these to taste in the event that you wish to keep your NIS source files
# separate from your NIS server's actual configuration files.
GROUP          = $(YPPWDDIR)/group
PASSWD         = $(YPPWDDIR)/passwd
SHADOW        = $(YPPWDDIR)/shadow
GSHADOW       = $(YPPWDDIR)/gshadow
ADJUNCT       = $(YPPWDDIR)/passwd.adjunct
#ALIASES      = $(YPSRCDIR)/aliases # aliases could be in /etc or /etc/mail
ALIASES       = /etc/mail/aliases
ETHERS        = $(YPSRCDIR)/ethers # ethernet addresses (for rarpd)
BOOTPARAMS   = $(YPSRCDIR)/bootparams # for booting Sun boxes (bootparamd)
HOSTS        = $(YPSRCDIR)/hosts
NETWORKS     = $(YPSRCDIR)/networks
PRINTCAP     = $(YPSRCDIR)/printcap
PROTOCOLS    = $(YPSRCDIR)/protocols
PUBLICKEYS   = $(YPSRCDIR)/publickey
RPC          = $(YPSRCDIR)/rpc
SERVICES     = $(YPSRCDIR)/services
NETGROUP     = $(YPSRCDIR)/netgroup
NETID        = $(YPSRCDIR)/netid
AMD_HOME     = $(YPSRCDIR)/amd.home
AUTO_MASTER  = $(YPSRCDIR)/auto.master
AUTO_HOME    = $(YPSRCDIR)/auto.home
AUTO_LOCAL   = $(YPSRCDIR)/auto.local
TIMEZONE     = $(YPSRCDIR)/timezone
LOCALE      = $(YPSRCDIR)/locale
NETMASKS    = $(YPSRCDIR)/netmasks
YPSERVERS   = $(YPDIR)/ypservers # List of all NIS servers for a domain
```

```
target: Makefile
    @test ! -d $(LOCALDOMAIN) && mkdir $(LOCALDOMAIN) ; \
    cd $(LOCALDOMAIN) ; \
    $(NOPUSH) || $(MAKE) -f ../Makefile ypservers; \
    $(MAKE) -f ../Makefile all

# If you don't want some of these maps built, feel free to comment
# them out from this list.

#all: passwd group hosts rpc services netid protocols netgrp mail \
#      shadow publickey # networks ethers bootparams printcap \
#      # amd.home auto.master auto.home auto.local passwd.adjunct \
#      # timezone locale netmasks
all: passwd group rpc services netid

#####
#
# DON'T EDIT ANYTHING BELOW IF YOU DON'T KNOW WHAT YOU ARE DOING !!!
#
#####

DBLOAD = $(YPBINDIR)/makedbm -c -m `$(YPBINDIR)/yphelper --hostname`
MKNETID = $(YPBINDIR)/mknetid
YPPUSH = $(YPSBINDIR)/yppush
MERGER = $(YPBINDIR)/yphelper
DOMAIN = `basename `pwd``
LOCALDOMAIN = `/bin/domainname`
REVNETGROUP = $(YPBINDIR)/revnetgroup
CREATE_PRINTCAP = $(YPBINDIR)/create_printcap
```



```

ethers:      ethers.byname ethers.byaddr
hosts:      hosts.byname hosts.byaddr
networks:   networks.byaddr networks.byname
protocols:  protocols.bynumber protocols.byname
rpc:        rpc.byname rpc.bynumber
services:   services.byname services.byservicename
passwd:     passwd.byname passwd.byuid
group:      group.byname group.bygid
shadow:     shadow.byname
passwd.adjunct: passwd.adjunct.byname
netid:      netid.byname
netgrp:     netgroup netgroup.byhost netgroup.byuser
publickey:  publickey.byname
mail:       mail.aliases
timezone:   timezone.byname
locale:     locale.byname
netmasks:  netmasks.byaddr

ypservers:  $(YPSERVERS) $(YPDIR)/Makefile
            @echo "Updating $@..."
            @$ (AWK) '{ if ($$1 != "" && $$1 !~ "#") print $$0"\t"$$0 }' \
                $(YPSERVERS) | $(DBLOAD) -i $(YPSERVERS) -o $(YPMAPDIR)/$@ - $@
            -@$ (NOPUSH) || $(YPPUSH) -d $(DOMAIN) $@

$(YPSERVERS):
            @echo -n "Generating $*..."
            @uname -n > $(YPSERVERS)

```

```
bootparams: $(BOOTPARAMS) $(YPDIR)/Makefile
@echo "Updating $@..."
@$(AWK) '{ if ($$1 != "" && $$1 !~ "#" && $$1 != "+") \
    print $$0 }' $(BOOTPARAMS) | $(DBLOAD) -r -i $(BOOTPARAMS) \
    -o $(YPMAPDIR)/$@ - $@
-@$(NOPUSH) || $(YPPUSH) -d $(DOMAIN) $@

ethers.byname: $(ETHERS) $(YPDIR)/Makefile
@echo "Updating $@..."
@$(AWK) '{ if ($$1 != "" && $$1 !~ "#" && $$1 != "+") \
    print $$2"\t"$$0 }' $(ETHERS) | $(DBLOAD) -r -i $(ETHERS) \
    -o $(YPMAPDIR)/$@ - $@
-@$(NOPUSH) || $(YPPUSH) -d $(DOMAIN) $@

ethers.byaddr: $(ETHERS) $(YPDIR)/Makefile
@echo "Updating $@..."
@$(AWK) '{ if ($$1 != "" && $$1 !~ "#" && $$1 != "+") \
    print $$1"\t"$$0 }' $(ETHERS) | $(DBLOAD) -r -i $(ETHERS) \
    -o $(YPMAPDIR)/$@ - $@
-@$(NOPUSH) || $(YPPUSH) -d $(DOMAIN) $@
```

.....

```
hosts.byname: $(HOSTS) $(YPDIR)/Makefile
@echo "Updating $@..."
@$(AWK) '/^[0-9]/ { for (n=2; n<=NF && $$n !~ "#"; n++) \
    print $$n"\t"$$0 }' $(HOSTS) | $(DBLOAD) -r $(B) -l \
    -i $(HOSTS) -o $(YPMAPDIR)/$@ - $@
-@$(NOPUSH) || $(YPPUSH) -d $(DOMAIN) $@

hosts.byaddr: $(HOSTS) $(YPDIR)/Makefile
@echo "Updating $@..."
@$(AWK) '{ if ($$1 !~ "#" && $$1 != "") print $$1"\t"$$0 }' \
    $(HOSTS) | $(DBLOAD) -r $(B) -i $(HOSTS) -o $(YPMAPDIR)/$@ - $@
-@$(NOPUSH) || $(YPPUSH) -d $(DOMAIN) $@
```

.....

```

ifeq (x$(MERGE_PASSWD),xtrue)

passwd.byname: $(PASSWD) $(SHADOW) $(YPDIR)/Makefile
    @echo "Updating $@..."
    @$ (UMASK); \
    $(MERGER) -p $(PASSWD) $(SHADOW) | \
        $(AWK) -F: '!/^[+-#]/ { if ($$1 != "" && $$3 >= $(MINUID) ) \
            print $$1"\t"$$0 }' | $(DBLOAD) -i $(PASSWD) \
            -o $(YPMAPDIR)/$@ - $@
    -@$ (NOPUSH) || $(YPPUSH) -d $(DOMAIN) $@

passwd.byuid: $(PASSWD) $(SHADOW) $(YPDIR)/Makefile
    @echo "Updating $@..."
    @$ (UMASK); \
    $(MERGER) -p $(PASSWD) $(SHADOW) | \
        $(AWK) -F: '!/^[+-#]/ { if ($$1 != "" && $$3 >= $(MINUID) ) \
            print $$3"\t"$$0 }' | $(DBLOAD) -i $(PASSWD) \
            -o $(YPMAPDIR)/$@ - $@
    -@$ (NOPUSH) || $(YPPUSH) -d $(DOMAIN) $@

# Don't build a shadow map !

shadow.byname:
    @echo "Updating $@... Ignored -> merged with passwd"

```

```

else

passwd.byname: $(PASSWD) $(YPDIR)/Makefile
    @echo "Updating $@..."
    @$ (UMASK); \
    $(AWK) -F: '!/^[-+#]/ { if ($$1 != "" && $$3 >= $(MINUID) ) \
        print $$1"\t"$$0 }' $(PASSWD) | $(DBLOAD) -i $(PASSWD) \
        -o $(YPMAPDIR)/$@ - $@
    -@$ (NOPUSH) || $(YPPUSH) -d $(DOMAIN) $@
passwd.byuid: $(PASSWD) $(YPDIR)/Makefile @echo "Updating $@..." @$ (UMASK); \
    $(AWK) -F: '!/^[-+#]/ { if ($$1 != "" && $$3 >= $(MINUID) ) \
        print $$3"\t"$$0 }' $(PASSWD) | $(DBLOAD) -i $(PASSWD) \
        -o $(YPMAPDIR)/$@ - $@
    -@$ (NOPUSH) || $(YPPUSH) -d $(DOMAIN) $@
shadow.byname: $(SHADOW) $(YPDIR)/Makefile
    @echo "Updating $@..."
    @$ (UMASK); \
    $(AWK) -F: '{ if (FILENAME ~ /shadow$$/) { \
        if (UID[$$1] >= $(MINUID) ) print $$1"\t"$$0; \
        } else UID[$$1] = $$3; }' $(PASSWD) $(SHADOW) \
        | $(DBLOAD) -s -i $(SHADOW) -o $(YPMAPDIR)/$@ - $@
    -@$ (NOPUSH) || $(YPPUSH) -d $(DOMAIN) $@

endif

```

```
passwd.adjunct.byname: $(ADJUNCT) $(YPDIR)/Makefile
@echo "Updating $@..."
@$(UMASK); \
$(AWK) -F: '!/^[-+#]/ { if ($$1 != "" ) print $$1"\t"$$0 }' \
    $(ADJUNCT) | $(DBLOAD) -s -i $(ADJUNCT) -o $(YPMAPDIR)/$@ - $@
@chmod 700 $(YPDIR)/$(DOMAIN)/$@*
-@$(NOPUSH) || $(YPPUSH) -d $(DOMAIN) $@

$(NETID):
netid.byname: $(GROUP) $(PASSWD) $(HOSTS) $(NETID) $(YPDIR)/Makefile
@echo "Updating $@..."
@$(MKNETID) -q -p $(PASSWD) -g $(GROUP) -h $(HOSTS) -d $(DOMAIN) \
    -n $(NETID) | $(DBLOAD) -o $(YPMAPDIR)/$@ - $@
-@$(NOPUSH) || $(YPPUSH) -d $(DOMAIN) $@
```

Verlagerung der Quellen vom Standard

```
/var/yp/Makefile: SunOS
```

```
DIR =/etc
```

```
DOM = `domainname`
```

```
NOPUSH = ""
```

```
B=-b
```

```
PWFILE = /etc/netpasswd
```

```
ALIASES = /etc/aliases
```

```
YPDIR=/usr/etc/yp
```

```
YPDBDIR=/var/yp
```

```
YPPUSH=$(YPDIR)/yppush -v
```

```
MAKEDBM=$(YPDIR)/makedbm
```

```
REVNETGROUP=$(YPDIR)/revnetgroup
```

```
STDETHERS=$(YPDIR)/stdethers
```

```
STDHOSTS=$(YPDIR)/stdhosts
```

```
MKNETID=$(YPDIR)/mknetid
```

```
.....
```

```
.....
passwd.time: $(PWFILe)/passwd
  @(awk 'BEGIN { FS=":"; OFS="\t"; } /^[a-zA-Z0-9_]/ \
    { print $$1, $$0 }' $(PWFILe)/passwd $(CHKPIPE))|\
    $(MAKEDBM) - $(YPDBDIR)/$(DOM)/passwd.byname;
  @(awk 'BEGIN { FS=":"; OFS="\t"; } /^[a-zA-Z0-9_]/ \
    { printf("%-10d ", $$3); print $$0 }' $(PWFILe)/passwd \
    $(CHKPIPE))| $(MAKEDBM) - $(YPDBDIR)/$(DOM)/passwd.byuid;
@touch passwd.time;
@echo "updated passwd";
@if [ ! $(NOPUSH) ];then $(YPPUSH) -d $(DOM) passwd.byname;fi
@if [ ! $(NOPUSH) ];then $(YPPUSH) -d $(DOM) passwd.byuid;fi
@if [ ! $(NOPUSH) ];then echo "pushed passwd"; fi
```


Hilfskommandos

Kommandos zum Anzeigen der Informationen einer Map

Anzeigen der ganzen NIS-Datenbasis

```
ypcat [ -kt ] [ -d domainname ] mname
```

```
ypcat -x
```

-k - Keys und Map

-t - verbiete die Benutzung von Nick-Namen z.B. passwd

-x - Ausgabe der Nick-Namen

-d domainname - Name der Domain

mname - Mapname oder Nick-Name

Anzeigen von Einträgen aus einer Map

```
ypmatch [ -d domain ] [ -k ] [ -t ] key ... mname
```

```
ypmatch -x
```

-k - Key wird vor dem Eintrag ausgegeben

-t - verbiete die Benutzung von Nick-Namen z.B. passwd

-x - Ausgabe der Nick-Namen

-d domainname - Name der Domain

key - Schlüssel

mname - Mapname oder Nick-Name

Änderungskommandos für /etc/passwd und passwd-Maps auf Masterserver

Voraussetzung:

/usr/lib/netsvc/yp/rpc.yppasswdd ist gestartet.

Syntax:

```
/usr/lib/netsvc/yp/rpc.yppasswdd filename [adjunct_file] [-nogecos]
[-noshell] [-nopw] [-m argument1 argument2 ]
```

adjunct_file - zu änderndes passwd-File
-nogecos - gecoc-Feld nicht änderbar
-noshell - Shell nicht änderbar
-nopw - Passwort nicht änderbar
-m argument1 ... - make sofort nach Änderung des
passwd-Files aufrufen.

Ändern der Passwortes:

```
yppasswd [ <username> ]
```

Ändern des Gecos-Feldes:

```
ypchfn [ <username> ]
```

Ändern der Shell:

```
ypchsh [ <username> ]
```

Wartungskommandos

Bestimmen des aktuellen NIS-Servers

```
ypwhich [-d domain] [-m [ mname ] [-t] | [-V1|-V2 ] hostname ]
ypwhich -x
  -d domain   - Domainname
  -m mname    NIS-Masterserver für eine Map
  -t          - keine Nick-Namen
  -x          - anzeigen der Nick-Namen
hostname     - NIS-Server von hostname anzeigen
```

Setzen des aktuellen NIS-Servers

```
/usr/sbin/ypset [-V1|-V2] [-d domain] [-h host] server
nur wenn ypbind mit -ypset oder -ypsetme gestartet ist
```

Verteilen der Maps

```
/usr/sbin/yppush [ -v ] [ -d domain ] mapname
  -v - ausführliches Protokoll
  -d domain - Domainname
  mapname  - Name der Map
Maps werden an alle NIS-Server verteilt, die in der Map
ypslaves.hostname bzw. ypservers stehen.
```

Holen der Statusinformationen von Maps von einem anderen NIS-Server

```
/usr/sbin/yppoll [ -h host ] [ -d domain ] mapname
  -h host   - Hostname des Servers
  -d domain - Domainname
  mapname   - Name der Map
```

z.B.:

```
$ /usr/sbin/yppoll passwd.byname
Domain informatik is supported.
Map passwd.byname has order number 868284006.
The master server is master.
$
```

Transportieren von NIS-Maps von einem Server zum lokalen Rechner

```
/usr/lib/netsvc/yp/ypxfr [-b] [-c] [-f] [-d domain] [-h host]
  [-s domain] [-C tid prog ipadd port] mapname
  -b - unterdrücke -b-Option
  -c - keine "Clear Map" Anforderung an den ypserv-Prozeß
  -f - unterdrücken des Transportes, wenn keine neue
      Version der Maps vorliegt
  -d domain - Domainname
  -h host   - Hostname
  -s domain - Name der Sourcedomain
  -C ....   - nur für ypserv
  mapname   - Name der Map, kein Nick-Name
```

Protokoll: /var/yp/ypxfr.log

Scripte für die Wartung für "cron"

ypxfr_1perhour- Holen von passwd

```
#!/bin/sh
# ypxfr_1perhour - Do hourly NIS map check/updates, to see, if we
#                 have missed one update.
YPBINDIR=/usr/lib/yp
MAPS_TO_GET=passwd.byname passwd.byuid shadow.byname publickey.byname
for map in $MAPS_TO_GET
do
    $YPBINDIR/ypxfr $map
done
```

ypxfr_2perday - Holen von hosts und netgroup

```
#!/bin/sh
#
# ypxfr_2perday - Do twice-daily NIS map check/updates, to see, if we
#                 have missed one update.
#
YPBINDIR=/usr/lib/yp
MAPS_TO_GET=hosts.byname hosts.byaddr netgroup netgroup.byuser \
            netgroup.byhost
for map in $MAPS_TO_GET
do
    $YPBINDIR/ypxfr $map
done
```

```
ypxfr_1perday - Holen von group, protocols, networks, rpc,  
services und ypservers
```

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

```
#
```

```
# ypxfr_1perday - Do daily NIS map check/updates, to see, if we  
# have missed one update.
```

```
#
```

```
YPBINDIR=/usr/lib/yp
```

```
MAPS_TO_GET=group.byname group.bygid protocols.byname protocols.bynumber \  
networks.byname networks.byaddr rpc.byname rpc.bynumber \  
services.byname ypservers
```

```
for map in $MAPS_TO_GET
```

```
do
```

```
    $YPBINDIR/ypxfr $map
```

```
done
```