

Dynamic Network Configuration with

LCFG

Revision 0.99.63

06/01/05 14:10

1 Introduction

LCFG is primarily intended to control the configuration of a host according a single specification, held in a central repository. This is not appropriate in certain situations, such as when a laptop is moved from one network to another – several different configurations may be required, the parameters may need to be obtained from the user, or determined automatically from the network, and there may be no connection available to the central configuration repository at the time when a configuration change is required.

This document describes the facilities provided by LCFG for managing such dynamic network configurations:

- ❑ Section 2 provides “Quickstart” instructions for end-users wishing to configure LCFG-managed laptops to operate on remote dialup (2.1), wired Ethernet (2.2), or wireless (2.3) networks.
- ❑ Section 3 describes the facilities for managing *schemes* so that groups of different configuration settings can be associated with different networks.
- ❑ Section 4 addresses some common questions.
- ❑ The appendices include reference sections for all the supported configuration parameters, and corresponding resources.

2 Quickstart

The default LCFG configuration will normally provide automatic detection and configuration of local networks at the home site, as well as simple external networks which are designed for access by “foreign” laptop users. In most cases, the following quickstart process is usually sufficient to obtain a working connection.

If this is insufficient, then it is probably necessary to create a personal *scheme* supplying explicit configuration information (see 3). *Note that the ability to create personal schemes for a machine must be preconfigured* (see 3.1). It is essential that this is enabled before travelling with a laptop.

2.1 Dialup

Before attempting to make a dialup connection you need:

- A modem card which is compatible with a standard Linux installation. Unfortunately, many modems are now *winmodems* which work only under Windows.
- A dialup account. Edinburgh University users may use the Edinburgh University dialup service¹.

To make a dialup connection, use the *dial* command:

```
→ dial -v
phone number: 01314730000
username: fred
password: ****
[INFO] dialup: ...
...
[OK] dialup: run
```

The `-v` option is used to get a report of the dialup progress which most people tend to like, since the connection negotiation can be rather slow. The dialup parameters for multiple connections can be stored, to avoid entering them each time, using the scheme editor `nse` (see 3.4).

To close the connection, use the `hangup` command:²

```
→ hangup
[OK] dialup: run (hangup)
```

If the system is unable to find a modem (perhaps you have a winmodem?), there will be a long delay while it tries different modem possibilities. Eventually, the command will fail, but most people will get bored and interrupt it before then.

¹<http://www.ucs.ed.ac.uk/dialup>

²Note that simply logging out will not close the connection.

```
→ dial -v
[INFO] dialup: user scheme file ...
[INFO] dialup: autodetecting modem ...
[FAIL] dialup: modem autodetection failed (see logfile)
```

Other error messages should be obvious. In case of problems, the `-D` option can be used to generate very verbose messages, and the man pages for `wvdial` and/or `pppd` may be helpful (the `lcfg-dialup` component uses these to make the dialup connection). The files `/var/lcfg/log/dialup` and `/var/lcfg/log/divine` contain a log of the connection attempt.

2.2 Ethernet

A (wired) Ethernet connection to a remote network requires:

- ❑ A compatible Ethernet card (most cards are likely to work).
- ❑ The default LCFG configuration must be set up to provide a suitable scheme for the network. By default, there will almost certainly be a scheme which is suitable for a generic LAN with a DHCP service (usually called `def_lan`). There may also be site-specific defaults for various local networks. The available schemes can be inspected using the network scheme editor `nse` (see 3.4).
- ❑ If there is no suitable default scheme, then a personal scheme must be created (see 3.5).

If the network is available when the machine boots, or wakes up from sleep mode, then the LCFG components will attempt to autodetect the scheme, and in many cases, the system will simply start up with the network automatically available. If this is not the case, a network scan can be initiated manually with the scheme configuration program `sconf`. In case of difficulties, rescanning the network is usually safe and will normally provide some indication of the problem:

```
→ sconf -v
[INFO] divine: reason for probe: run
[INFO] divine: reading user schemes ...
[INFO] divine: probing eth0 (link up)
[INFO] divine: no response
[INFO] divine: trying default scheme : lan ...
[INFO] divine: trying DHCP (pump) on eth0
[INFO] divine: Device eth0
[INFO] divine: IP: 192.168.93.100
...
[INFO] divine: detected scheme : def_lan (Default LAN)
[INFO] divine: setting context: scheme=def_lan net=remote
[INFO] divine: disabling interface (not used) : eth1
[OK] divine: run (def_lan)
```

This example shows the autodetection process trying to probe for specific local network schemes and failing (because the laptop is connected to a foreign network). It then falls back to the default scheme for a generic LAN connection and succeeds in configuring the network with DHCP. The disabled interface is the wireless interface which has been disabled because it is not required when a LAN connection is available.

Again, the `-v` option is used to provide a verbose display of the progress, and the file `/var/lcfg/log/divine` will contain a log of the connection attempt.

If the autodetection fails to detect a network, it will disable all network interfaces. This reconfiguration may also cause some useful application reconfiguration to occur - for example, some network daemons will be automatically disabled:

```
...
[INFO] divine: setting context: scheme= net=none
[OK] divine: run (none)
```

If there is a network available, but it is not detected, then this probably indicates that the network requires manual configuration, and a personal scheme should be created. In rare cases, if the DHCP service is very slow, it may be necessary to lengthen the timeout, or to increase the number of retries. This can be achieved by passing appropriate values to the LCFG `divine` component (see appendix C) via `sconf` as follows:

```
→ sconf -v -- -e 4 -t 5
...
```

This increases the number of retries to 4, and the timeout to 5 seconds³.

2.3 Wireless

Wireless connections are handled in exactly the same way as the LAN connections described above (2.2). However:

- ❑ A smaller number of wireless cards are likely to be supported. In particular, only those which are compatible with the Linux Wavelan driver.
- ❑ In addition to the LAN network parameters, wireless networks may also require an SSID (network name) and a WEP key (for authentication). The default wireless scheme (normally called `def_wireless`) is only suitable for networks which allow access with a default SSID (`ANY`), and do not use WEP. Individual sites will usually preconfigure additional schemes for the local wireless networks. In other cases, a personal scheme must be created (see 3.6) to supply these parameters.

The following example shows autodetection of a wireless network:

³Increasing these values unnecessarily will increase the time taken for autodetection.

```
reading user schemes from: ...
disabling eepro100 interface (link down) : eth0
wireless network not found on eth1 : informatics ...
wireless network not found on eth1 : central ...
found wireless network on eth1 : ANY=purplepatch ...
access point: 00:01:24:C0:39:FE (responded in 0.1 secs)
trying default scheme : def_wireless (Default Wireless)
trying DHCP (pump) for wireless on eth1 : ANY=purplepatch
pump event: up eth1 10.30.242.167
Device eth1
  IP: 10.30.242.167
...
detected scheme : def_wireless (Default Wireless)
setting context: scheme=def_wireless net=remote
```

Two explicit local schemes have been attempted before falling back on the default wireless scheme which has detected an open “hotspot” network⁴. Note that the wired interface (eth0) was attempted first, but has been disabled because no network connection was found.

⁴The Edinburgh cafe “BeanScene”, in this case

3 Managing Schemes

A *scheme* is a collection of configuration parameters which are grouped under a scheme name, and associated with a particular network. In general, a scheme may contain four types of information:

- ❑ Parameters specifying how to establish a dialup connection, such as the telephone number and account information.
- ❑ Parameters which are used by the autodetection process in an attempt to identify a wireless or Ethernet network automatically.
- ❑ Parameters which are used to configure the network settings for correct operation of the network, once it has been identified.
- ❑ Parameters which are used to reconfigure applications depending on the particular network. For example, some applications (such as SMTP mail) may need to be configured differently for use outside of the local environment, perhaps for authentication reasons.

The parameters provided by a scheme override any global parameters specified in the central LCFG configuration for the machine.

The parameters for a particular schemes may come from different sources:

- ❑ The LCFG configuration for a machine will usually include a set of default schemes using the resources for the `divine` and `dialup` components. The resource details are described in the manual pages (see [C](#) and [D](#)). Such schemes will normally provide standard, generic defaults, as well as schemes suitable for the local networks of the particular site. Users are not able to change these schemes, although they may override them by supplying personal schemes with the same name.
- ❑ Users may create their own personal schemes using the “network scheme editor” `nse` (see [3.4](#)).

3.1 Enabling Personal Schemes

The central LCFG specification for the machine must specify which user’s personal schemes to consider when reconfiguring –

⇒ To enable the use of personal schemes, the LCFG resource `divine.userfile` should be set to the filename in which the personal schemes are stored. Normally, this would be set to the pathname of the `.schemes` file in the user’s home directory. Note that this gives the named user complete control over the network configuration of the machine, so this resource should only be set for machines where this is acceptable. The variable `%HOME` can be used in the resource value to refer to the home directory of the user at the machine console. This provides a way for different people to share the configuration of machine, but the final behaviour of the machine can be difficult to predict and this is not recommended.

3.2 Autodetection

The configuration corresponding to a particular scheme can explicitly be enabled by specifying the scheme name to `sconf`⁵:

```
→ sconf -- -s myscheme
```

For dialup connections, the `dial` command automatically calls `sconf` after establishing the dialup connection, and any scheme name that is specified as an argument of the `dialup` command is passed to `sconf`. If no scheme name is given to the `dial` command, then it uses the default dialup scheme (`def_dialup`).

If no scheme name is given to `sconf`, it will examine all the available schemes in an attempt to automatically find one which is suitable. The search sequence is rather complicated, but the following general rules apply:

1. Wired interfaces are attempted before wireless interfaces.
2. Wireless schemes with explicit SSIDs are only attempted if a network with a matching SSID can be found.
3. Wireless schemes with explicit SSIDs are attempted before schemes with an SSID of ANY.
4. Wireless schemes with explicit Access Point addresses are only attempted if the AP address matches.
5. Schemes specifying “probe” addresses are attempted before those without. The network is probed (using the ARP protocol) for the existence of the specified machines, and the scheme is ignored if none of them respond.
6. Otherwise, schemes are attempted in the order that they appear in the `nse` scheme list (see 3.4).

Machines are normally configured (via the `boot` component) to initiate autodetection on wakeup from sleep, as well as at component startup time (during the boot process).

3.3 Scheme Application

Once a scheme has been identified, the LCFG `divine` component will use the scheme parameters to configure the network. In general, parameters which are not specified will be determined automatically where possible. For example, schemes which do not specify an IP address will have the IP address assigned automatically via DHCP or PPP.

Once the network is established, the following actions are initiated:

⁵`sconf` simply calls the `run` method of the `lcfg-divine` component and the manual page (see appendix D) provides full details of the capabilities.

- ❑ LCFG resources for some significant applications are automatically set (using a local context) to override the default values. This enables configuration of DNS, routing, SMTP etc. via the normal components, rather than having hardwired code.
- ❑ LCFG context variables are set for the name of the active scheme (`scheme=`), and for any additional contexts defined by the scheme itself. This allows any application to specify reconfiguration depending on the connected network scheme.
- ❑ The scheme may optionally supply a script to execute on scheme activation. This could be used, for example, to reconfigure some user environment setting, or to initiate some automatic authentication process.
- ❑ A sound can optionally be played to indicate the success or otherwise of the network establishment. This can be useful since network reconfiguration may occur when the user is not logged-in, and there is no obvious way of reporting the status.

3.4 Creating Personal Schemes

The network scheme editor `nse` provides a GUI interface for editing personal scheme parameters (these are normally stored in `~/.schemes`).

Figure 1 shows the main `nse` window – a list of available schemes is shown in the panel on the left, and schemes can be created, deleted, and copied using the **Edit** menu or associated keys. Since the schemes are attempted in the listed order during autodetection, the **Edit** menu also provides options to move schemes up or down in the list.

Schemes whose names appear in brackets are defined in the central LCFG configuration and cannot be edited. They can however be copied, and creating a scheme with the same name as a predefined one will allow the parameters to be overridden. Some caution is required here because the parameters of the predefined schemes may be changed from time to time, to reflect changes in the local network infrastructure. If these are overridden with personal copies, then these changes will not take effect and the network connection may cease to work, or behave unpredictably. Schemes can be temporarily disabled by appending a “*” to the end of the scheme name.

The parameters for each scheme are grouped in pages which can be accessed via the “tabs” in the right hand panel. Clicking on the label for a parameter field will display a short description of the field in the message area at the bottom, and sensible default values are usually supplied for parameter fields which are left blank – the **Name** and **Description** are the only parameters which should normally always be present.

The **General** page contains parameters which are mostly relevant for all types of network connection, and the **Wireless** page contains additional parameters for wireless networks. All of these parameters are described in detail in appendix A, and the corresponding LCFG resources are described in the `lcfg-divine` manual page (appendix D). Additional parameters for dialup connections appear on the **Dialup** page, and are described in appendix B. The associated resources are described in the `lcfg-dialup` manual page (appendix C).

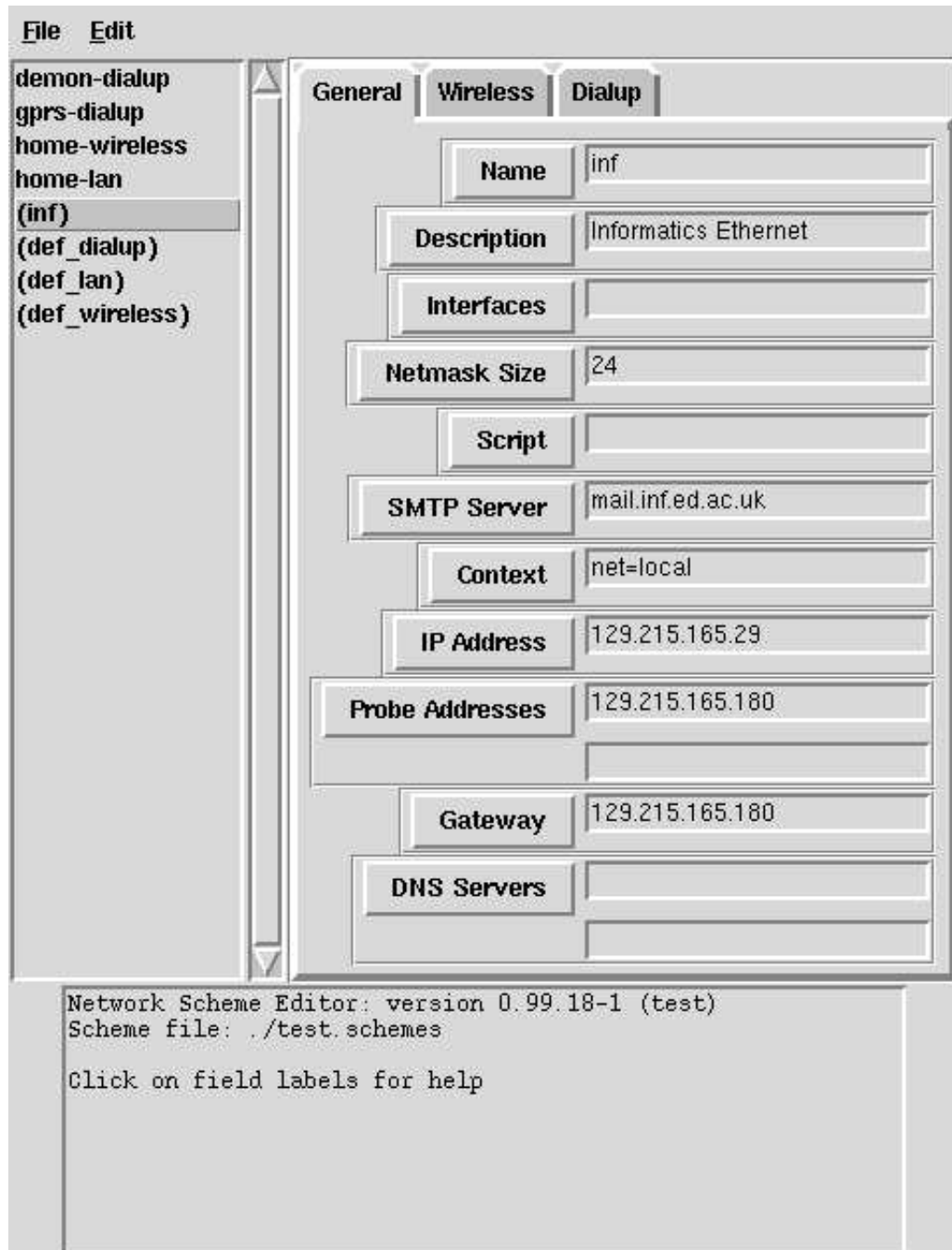


Figure 1: General Parameters

3.5 Creating a LAN scheme

A basic LAN scheme can be created simply by editing a new scheme and leaving all the configuration parameters blank (except for the **Name**). Normally, such a configuration will already be provided by default (with the name `def_lan`), and the purpose of creating the scheme will be to specify one or more of the following parameters:

IP Address

An explicit IP address.

Gateway

An explicit gateway address.

DNS Servers

A list of explicit DNS servers. If this field is left blank, and DHCP is not being used, then the home site DNS servers will be used. This should work, but is not advisable – if no local DNS servers are known, then the value `*` can be specified which will cause the DNS service on the local host to query the root servers directly for unknown addresses.

Probe Addresses

A list of addresses of machines on this network segment. When several LAN schemes are defined, these machines will be probed (using the ARP protocol) when autodetecting, and the responding machine will be used to identify the corresponding network, and hence the appropriate scheme. The gateway address is often a suitable value.

SMTP Server

The address of a mail relay to which outgoing SMTP mail should be sent. If this is set to `localhost`, then mail will be sent directly from the local `sendmail` process to the destination site. Specifying a relay is useful, since mail will be queued on the relay when the remote site is down – this prevents mail being queued on the laptop where it may be delayed when the laptop is disconnected. *Note:* If this field is left blank, the default site mail relay will be used – this is unlikely to work outside of the local site, and the value `localhost` should probably be used on networks where no local SMTP relay is available.

See appendix [A](#) for further details of these parameters, and others on the **General** page which may also be relevant to LAN connections.

Note that parameters which specify a host address may be given as DNS names (rather than IP numbers) providing that the machine is connected to a DNS service at the time when the names are entered⁶

3.6 Creating a Wireless Scheme

All of the above parameters for LAN schemes are applicable to wireless schemes. The additional parameters on the *Wireless* page (figure 2) are also relevant:

Wireless Net

A non-empty value indicates that this scheme is a wireless scheme. The value is the network SSID. The value *ANY* can be used to bind to any available network, and should be used if the SSID is unknown.

WEP Key

If the network uses WEP encryption, then this field must contain the key. Click on the label, or see appendix A for details of the format.

AP Address

If this field specifies an access point MAC address, then the scheme will only be selected for matching access points. This allows different schemes to distinguish between different access points (and hence physical locations) on the same network. This can be useful, for example, to set different default printers (using the `script` parameter).

⁶The host addresses are resolved at the time of entry, and cached. Care is therefore required if the DNS addresses of these hosts are subsequently changed.

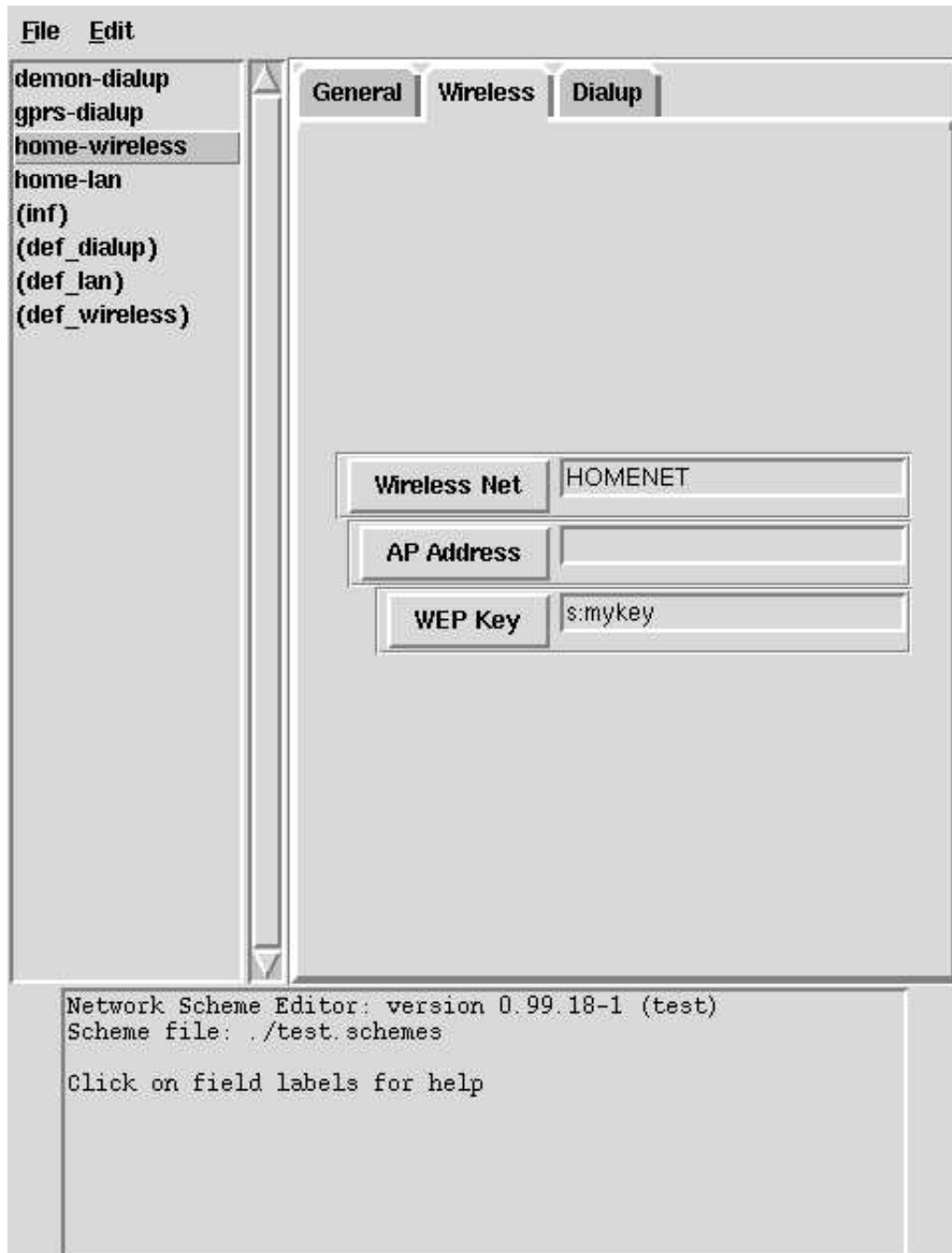


Figure 2: Wireless Parameters

3.7 Creating a Dialup Scheme

Most of the parameters on the **General** page are appropriate for dialup schemes. When parameters such as the IP address are not specified explicitly, the values provided by the PPP protocol are used. In addition, the parameters on the **Dialup** page (figure 3) are used to establish the dialup connection. These are listed in appendix B and are largely self-explanatory. In most cases, the dialup parameters are passed directly to the standard `wvdial` program, and more details can be found in the `wvdial` manual page. The following points are worth noting:

- ❑ If any of the the phone number, user name, or password are left blank, they will be prompted for when the `dial` command is run.
- ❑ All values are currently stored in plaintext in the user scheme file, so important passwords, or other sensitive information should not be specified in the scheme.
- ❑ Blank parameters will inherit values from the default `def_dialup` scheme. See the use of the special values `<blank>` and `<default>` in the `lcfg-dialup` manual page (appendix C) for details of how to explicitly enter a blank value.
- ❑ If the `modem` field is left blank, then it will inherit the special value `<auto>` from the default scheme, and the `dial` command will use `wvdialconf` in an attempt to locate the modem.
- ❑ In addition to the `wvdial` parameters which are provided explicitly, arbitrary parameters can be passed to `wvdial` using the **Wvdial Parameters** fields.

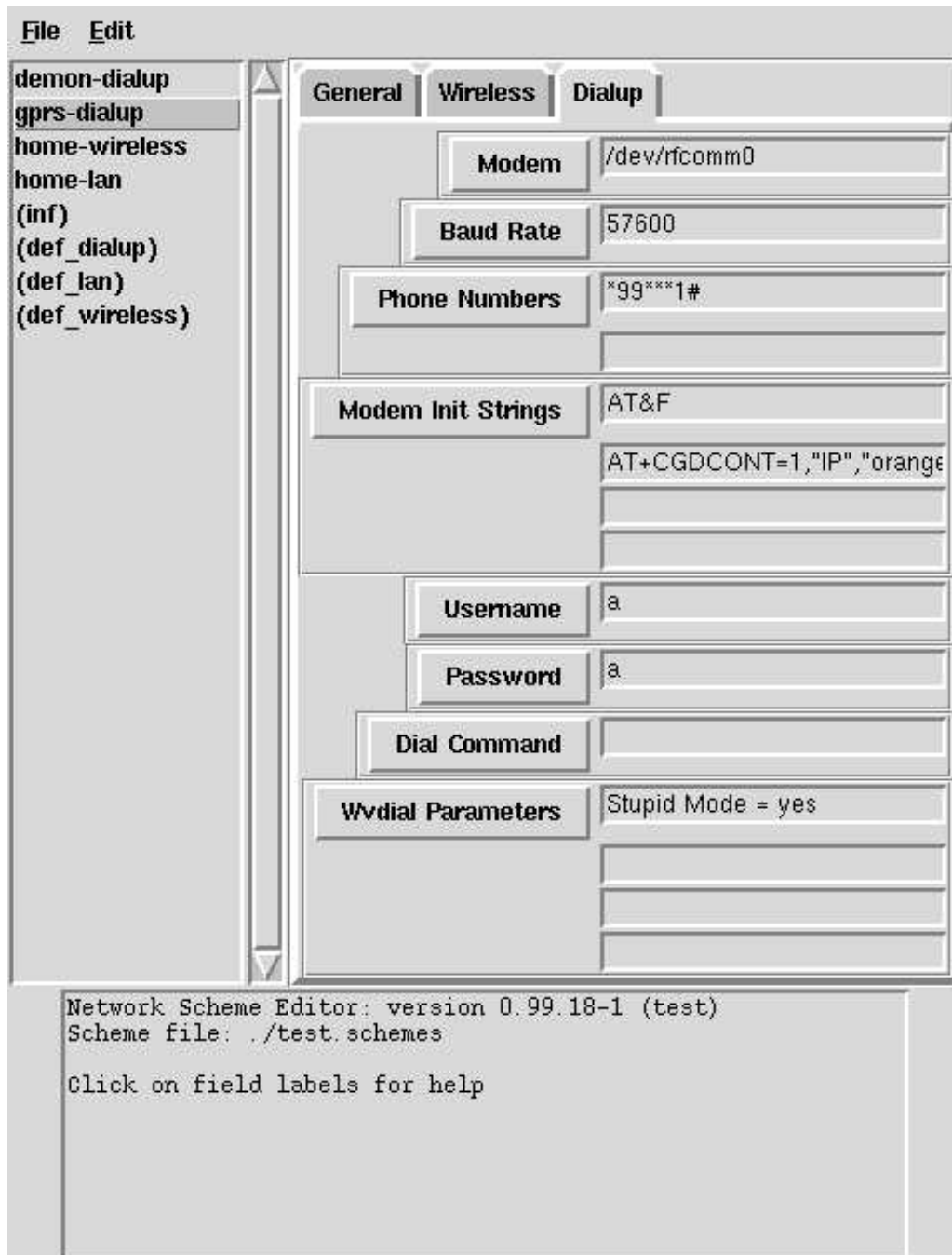


Figure 3: Dialup Parameters

4 Common Question


- Can I use my Bluetooth phone to make a dialup connection?*

Yes. However ...

1. Bluetooth support is rather experimental with the current version of LCFG and Linux. The Bluetooth component needs to be enabled, and the address and PIN for the phone need to be defined as bluetooth resources in the central LCFG configuration⁷.
2. The modem device must be specified explicitly (usually as `/dev/rfcomm0`).
3. Other dialup parameters may depend on the handset, and the provider. Especially when using GPRS, these may be rather cryptic. Figure 3 shows an (incomplete) example of GPRS dialup settings for a Panasonic X70 on the Orange network.

- Do I need a VPN?*

It is often not necessary to use a VPN, as long as the the SMTP mail relay is correctly configured (see 3.5)⁸. However, some remote networks (Edinburgh University Wireless, for example) do not permit the use of unencrypted protocols such as SMTP. It may also be necessary to use a VPN for access to other restricted resources on the home site (such as address-restricted web pages).

 DICE laptops are preconfigured with commands to enable the DICE VPN – use the `renc` command to authenticate, followed by `infvpn` to start up the VPN.

⁷Of course, compatible bluetooth hardware is also necessary

⁸Note that mail, for example, will then be sent across the remote network in plaintext which may not be acceptable.

A Divine Parameters

Name

This is a short identifier for the scheme. If more than one scheme has the same identifier, then the first occurrence takes precedence. Schemes shown with brackets around the name are system defaults that cannot be changed. The scheme named "ppp" will be used as a default for dialup connections. Schemes can be disabled by postfixing the name with *.

Description

An optional description of the scheme.

Interfaces

A (space-separated) list of interfaces for which this scheme is appropriate. If this field is blank, the scheme is a candidate for all interfaces. Use "ppp0" for schemes which are intended only for dialup.

Wireless Net

The name of a wireless network to which the scheme should be applied. If this is blank, the scheme applies only to wired interfaces. The name may be prefixed with /ad-hoc for an ad-hoc network (default is "/managed").

AP Address

A (space-separated) list of access point MAC addresses for which this scheme is valid. By default, schemes are valid for any AP. Mac Addresses must have the form XX:XX:XX:XX:XX:XX.

WEP Key

The WEP encryption key for this wireless network as a hex string, or in the form "s:PASSWORD". If this is blank, no encryption is performed. This field is ignored for wired networks.

Netmask Size

The size (in bits) of the netmask for this network (default 24). If DHCP is being used (IP field is blank) or the scheme is being used for PPP, then this is determined automatically.

Script

The name of script to execute when this scheme is detected. The name of the scheme is passed as an argument, and the script is run with the uid of the user owning the scheme file. The script is executed in the background with input and output to /dev/null.

SMTP Server

The name of a host to use as the mail relay. Local mail will always be delivered via the local sendmail program which will forward it to this relay. If this is blank,

then the LCFG-defined default mail relay will be used. Setting this to "localhost" will cause the local sendmail to deliver mail directly; this is usually necessary when connected to a foreign network.

Context

A context to be passed to the LCFG profile component. This can be used to tune the system configuration towards the type of connection. The default is set by the defcontext resource (normally "net=remote"). The additional context "scheme=ID" is also selected.

IP Address

The IP address (or hostname) to which the host should be configured if this scheme is detected. If this is blank, then DHCP will be used in an attempt to determine the host address. If a name is specified in the LCFG (rather than an IP number), then the name must be available in the local DNS server. This field is ignored for dialup schemes and the IP address is determined via PPP.

Probe Addresses

The IP address (or hostname) of one or more well-known hosts (eg. the gateway) to be probed to identify this scheme. If this is blank, the scheme will be applied as a default if all others fail. Note that more than one default scheme is not meaningful, unless they are on different wireless networks. If a name is specified in the LCFG (rather than an IP number), then the name must be available in the local DNS server. This field is not used for dialup connections.

Gateway

The IP address (or hostname) of the gateway corresponding to this scheme. This value is set automatically if DHCP is being used (IP field is blank), or when using a dialup scheme. If a name is specified in the LCFG (rather than an IP number), then the name must be available in the local DNS server. The special value ;route; indicates that the existing default values for the routing component will be used to set up the routing.

DNS Servers

The IP addresses (or hostnames) of DNS servers to use with this scheme. These hosts will be used as forwarders by the local DNS server which is always queried first. If this is blank, then the default forwarders will be used. If it is *, then no forwarders will be set. If a name is specified in the LCFG (rather than an IP number), then the name must be available in the local DNS server. If left blank, DHCP and PPP will supply these values automatically.

B Dialup Parameters

Modem

The full pathname of the modem device. The value `;auto;` can be used to autodetect the modem using `wvdialconfig`. (see `wvdial` man page)

Baud Rate

Modem baud rate. The value `;auto;` can be used to autodetect the baud rate using `wvdialconfig`. (see `wvdial` man page)

Phone Numbers

Telephone numbers. (see `wvdial` man page)

Modem Init Strings

Modem initialisation strings. The value `;auto;` can be used to autodetect the init strings using `wvdialconfig`. (see `wvdial` man page)

Username

Dialup username. (see `wvdial` man page)

Password

Dialup password. (see `wvdial` man page)

Dial Command

Modem dial command. (see `wvdial` man page)

Wvdial Parameters

Additional `wvdial` parameters. (see `wvdial` man page)

C dialup

LCFG dialup component

DESCRIPTION

This component uses **wvdial** to establish a dialup PPP connection. The configuration file for wvdial is created from the scheme parameters managed by **lcfg-schemes** and the **lcfg-divine** component is used to configure the network parameters once the connection is established.

The **run** method initiates a connection. An optional argument specifies the scheme to use. If no argument is specified, the scheme **def_dialup** is used.

Two special values can be used for resources which refer to wvdial parameters:

<blank >

This value generates a configuration file line for the resource with an empty value. This is different from omitting the line, because wvdial will use the default value for parameter if the line is omitted.

<default >

This is equivalent to leaving the resource value blank; ie. wvdial will use any default value. However, specifying this value prevents the scheme editor **nse** from closing up blank entries in multi-field values.

RESOURCES

schemes

A (space-separated) list of *scheme tags*. This should include at least one scheme conventionally named `def_dialup` which specifies the default values.

userfile

This resource specifies a (space-separated) list of files containing scheme data that will be read before (and take precedence over) the schemes specified in the resources. This is intended to allow a user to create temporary schemes, eg. while travelling with a portable. These files are normally managed with the scheme editor **nse**. Ability to change this file gives a user the equivalent of root permission. The variable `%HOME` is substituted with the home directory of the user at the console, so typical values might be: `%HOME/.schemes` or `/home/owner/.schemes`.

modem_ <tag >

The full pathname of the modem device. The value `<auto >` can be used to autodetect the modem using `wvdialconfig`. (see `wvdial` man page)

baud_ <tag >

Modem baud rate. The value `<auto >` can be used to autodetect the baud rate using `wvdialconfig`. (see `wvdial` man page)

phone_ <tag >

Telephone numbers. (see wvdial man page)

init_ <tag >

Modem initialisation strings. The value <auto > can be used to autodetect the init strings using wvdialconfig. (see wvdial man page)

username_ <tag >

Dialup username. (see wvdial man page)

password_ <tag >

Dialup password. (see wvdial man page)

dial_ <tag >

Modem dial command. (see wvdial man page)

wvdial_ <tag >

Additional wvdial parameters. (see wvdial man page)

PLATFORMS

Redhat9

AUTHOR

Paul Anderson <dcspaul@inf.ed.ac.uk >

VERSION

0.99.12-1

D **divine**

Network configuration component for LCFG.

DESCRIPTION

This object configures and controls the divine network probe. The `probe` method is called when one of the controlled interface comes, up to probe the network and determine the appropriate "scheme". The interface is set accordingly, and affected LCFG components (such as mail) are reconfigured by using the context mechanism. The `run` method can be used to initiate a probe manually.

If an interface is a wireless interface, then all the specified wireless network names will be probed, in the order that they appear in the scheme list. All schemes corresponding to a particular network will be probed in parallel. If the interface is not a wireless network, then all non-wireless schemes will be probed in parallel. Default schemes will be tried if all probes fail. DHCP is used if no explicit IP address is given.

Most addresses can be specified as DNS names or numbers. However, if the hostnames are not in the local DNS, they must be specified as numbers, otherwise they will not be available when booting on remote networks.

OPTIONS

The following options are supported by the `start`, `run` and `probe` methods:

-a *count*

The number of times to retry an arp request when probing the network. Small values may fail to detect networks with a slow arp response. Large values will increase the time required to probe a series of networks. The default value is set by the **arptries** resource.

-C

Output progress to `/dev/console`.

-D

Debugging.

-e *count*

The number of times to retry a DHCP request. Small values may fail to detect networks with a slow DHCP response. Large values will increase the time required to probe a series of networks. The default value is set by the **dhcptries** resource.

-R *reason*

The given reason for the network probe is noted in any informational messages. This set to the interface name, for example, when a probe occurs because the interface is coming up.

-s scheme

Attempt to set the specified scheme (only). No attempt is made to probe or to determine the applicability of the scheme. The scheme is attempted on all interfaces in the **if** resource.

-t seconds

The timeout on DHCP requests. Small values may fail to detect networks with a slow DHCP response. Large values will increase the time required to probe a series of networks. The default value is set by the **dtimeout** resource.

-v

Verbose. Displays all probe attempts.

-w seconds

The timeout on wireless access point detection. Small values may fail to detect networks with a slow AP response. Large values will increase the time required to probe a series of networks. The default value is set by the **wtimeout** resource.

RESOURCES

arptries

The number of times to retry an arp request when probing the network. Small values may fail to detect networks with a slow arp response. Large values will increase the time required to probe a series of networks. The default value is 6.

dhclient

Use dhclient instead of pump for DHCP. (NOT YET IMPLEMENTED).

dhcptries

The number of times to retry a DHCP request. Small values may fail to detect networks with a slow DHCP response. Large values will increase the time required to probe a series of networks. The default value is 2.

dtimeout

The timeout on DHCP requests. Small values may fail to detect networks with a slow DHCP response. Large values will increase the time required to probe a series of networks. The default value is 2.

interfaces

A (space-separated) list of interfaces to be controlled by divine. Each interface will be tried in the given order and the first interface which matches a particular scheme will be set accordingly. All other interfaces will be disabled.

type *interface* *interface!* *divine resource*

The type of the specified interface. If the interface is supported by the "linkstatus" command, then this can be used to determine whether or not the interface cable is connected. This means that the schemes can be probed faster because there is no need to wait for ARP timeouts on disconnected interfaces. Supported types are "mii" and "eepro100". The default is null (interface type unknown), in which case no test is made for the presence of the cable.

schemes

A (space-separated) list of *scheme tags*. This should include at least one scheme conventionally named `default` which specifies the normal default network parameters.

pidfiles

A (space-separated) list of filenames assumed to contain process IDs. When the scheme changes, each process will be sent a USR2 signal and the name of the new scheme will be in `/var/lcfg/tmp/schemes.scheme`. This can be used by processes to monitor and display scheme changes (eg. the "sleepbutton" provided with `obj-kdm`). The variable `%HOME` is substituted with the home directory of the user at the console, and the process kill is run under the uid of the pidfile owner. The default is `%HOME/.schemes.pid`.

route

If this resource is true, `divine` will add a default for route for the specific gateway. If it is false, it will generate resources for a routing component to handle the routing.

userfile

This resource specifies a (space-separated) list of files containing scheme data that will be read before (and take precedence over) the schemes specified in the resources. This is intended to allow a user to create temporary schemes, eg. while travelling with a portable. These files are normally managed with the scheme editor `nse`. Ability to change this file gives a user the equivalent of root permission. The variable `%HOME` is substituted with the home directory of the user at the console, so typical values might be: `%HOME/.schemes` or `/home/owner/.schemes`.

defcontext

If this resource is non-null, it should be a profile "context" which will be used for schemes which don't specify an explicit context. (This is in addition to the `scheme=name` context).

nocontext

If this resource is non-null, it should be a profile "context" which will be set when no network scheme can be detected. (This is in addition to the `scheme=` context).

dohosts

If this resource is true, then the IP address is registered in `/etc/hosts` as the address of the host.

oksound

The name of a sound file to play when a network scheme is successfully selected.

failsound

The name of a sound file to play when no network scheme can be selected.

cfopts

The options to be applied to divconf when it is called because an interface has come up.

wtimeout

The timeout on wireless access point detection. Small values may fail to detect networks with a slow AP response. Large values will increase the time required to probe a series of networks. The default value is 0.5.

descr_ <tag >

An optional description of the scheme.

if_ <tag >

A (space-separated) list of interfaces for which this scheme is appropriate. If this field is blank, the scheme is a candidate for all interfaces. Use "ppp0" for schemes which are intended only for dialup.

wnet_ <tag >

The name of a wireless network to which the scheme should be applied. If this is blank, the scheme applies only to wired interfaces. The name may be prefixed with /ad-hoc for an ad-hoc network (default is "/managed").

aplist_ <tag >

A (space-separated) list of access point MAC addresses for which this scheme is valid. By default, schemes are valid for any AP. Mac Addresses must have the form XX:XX:XX:XX:XX:XX.

wep_ <tag >

The WEP encryption key for this wireless network as a hex string, or in the form "s:PASSWORD". If this is blank, no encryption is performed. This field is ignored for wired networks.

masksize_ <tag >

The size (in bits) of the netmask for this network (default 24). If DHCP is being used (IP field is blank) or the scheme is being used for PPP, then this is determined automatically.

script_ <tag >

The name of script to execute when this scheme is detected. The name of the scheme is passed as an argument, and the script is run with the uid of the user owning the scheme file. The script is executed in the background with input and output to /dev/null.

smtp_ <tag >

The name of a host to use as the mail relay. Local mail will always be delivered via the local sendmail program which will forward it to this relay. If this is blank, then the LCFG-defined default mail relay will be used. Setting this to "localhost" will cause the local sendmail to deliver mail directly; this is usually necessary when connected to a foreign network.

context_ <tag >

A context to be passed to the LCFG profile component. This can be used to tune the system configuration towards the type of connection. The default is set by the defcontext resource (normally "net=remote"). The additional context "scheme=ID" is also selected.

ip_ <tag >

The IP address (or hostname) to which the host should be configured if this scheme is detected. If this is blank, then DHCP will be used in an attempt to determine the host address. If a name is specified in the LCFG (rather than an IP number), then the name must be available in the local DNS server. This field is ignored for dialup schemes and the IP address is determined via PPP.

probe_ <tag >

The IP address (or hostname) of one or more well-known hosts (eg. the gateway) to be probed to identify this scheme. If this is blank, the scheme will be applied as a default if all others fail. Note that more than one default scheme is not meaningful, unless they are on different wireless networks. If a name is specified in the LCFG (rather than an IP number), then the name must be available in the local DNS server. This field is not used for dialup connections.

gw_ <tag >

The IP address (or hostname) of the gateway corresponding to this scheme. This value is set automatically if DHCP is being used (IP field is blank), or when using a dialup scheme. If a name is specified in the LCFG (rather than an IP number), then the name must be available in the local DNS server. The special value <route > indicates that the existing default values for the routing component will be used to set up the routing.

dns_ <tag >

The IP addresses (or hostnames) of DNS servers to use with this scheme. These hosts will be used as forwarders by the local DNS server which is always queried first. If this is blank, then the default forwarders will be used. If it is *, then

no forwarders will be set. If a name is specified in the LCFG (rather than an IP number), then the name must be available in the local DNS server. If left blank, DHCP and PPP will supply these values automatically.

PLATFORMS

Redhat7, Redhat9

AUTHOR

Paul Anderson <dcspaul@inf.ed.ac.uk >

VERSION

3.5.31-1