

# DeskNow

## Administration and Configuration

Version : 2.1  
Date : 4 October 2003



Ventia Pty Limited A.C.N. 090 873 662  
Web : <http://www.ventia.com>  
Email : [info@ventia.com](mailto:info@ventia.com)

# Overview

This manual describes the procedures involved in administering and configuring a DeskNow server.

DeskNow is a computing platform that provides a rich collaborative environment, accessible from everywhere and from different devices. The DeskNow platform is open to third-party developers and to integration with legacy systems.

A more detailed overview of DeskNow can be found at <http://www.desknow.com>.

In this document it is assumed that the Reader is already familiar with DeskNow features and concepts.

The Reader of this manual that wants to perform custom setup and advanced administration should also have a good knowledge of web server products, RDBMS systems and system administration (Windows/Unix) in general.

# Legal

DeskNow is a registered trademark of Ventia Pty Limited.

Every other trademark present in this document belongs to its registered owner.

Several patent applications have been lodged by Ventia Pty Ltd to cover ideas and technologies that are illustrated in this document.

The concepts, names, or functionalities illustrated in this document may change without prior advice.

We wish to thank the following people for their translations of the DeskNow interface:

Carlos Espinosa (Spanish)  
Knut Torgersen (Norwegian)  
Rob Landaal (Dutch)  
Luca Sormani (Italian)

# Table of contents

Overview.....	2
Legal.....	3
Table of contents.....	4
1 Introduction.....	8
2 Basic administration.....	9
2.1 Starting and Stopping DeskNow.....	9
2.1.1 Simple installation.....	9
2.1.2 Advanced installation.....	9
3 User administration.....	10
3.1.1 Domains.....	10
3.1.2 Communities.....	10
3.1.3 Users.....	11
3.1.4 'Guest' users.....	11
3.1.5 'Restricted' users.....	11
3.1.6 Planning disk quotas.....	11
3.1.7 Planning the virtual file system.....	12
3.1.8 Creating users, communities and domains.....	12
3.1.9 Bulk creation of user accounts.....	12
3.1.10 Domains and login.....	13
3.1.11 Domains and POP3 and SMTP authentication.....	14
3.1.12 Su-like login.....	14
3.1.13 Su-like login for community managers.....	14
4 Mail.....	15
4.1.1 DeskNow as MTA (Mail Transfer Agent).....	15
4.1.2 DeskNow as SMTP server.....	15
4.1.3 SMTP Forwarding.....	15
4.1.4 DeskNow as POP3 server.....	16
4.1.5 More configuration options.....	16
5 Antivirus scanning.....	17
5.1 Antivirus scanning for Linux.....	18
6 Common configuration tasks.....	19
6.1 Using a different database.....	19
6.1.1 Creation of the database schema.....	19
6.1.2 Installation of the JDBC drivers.....	20
6.1.3 Configuration of DeskNow.....	20
6.1.4 Restart DeskNow.....	21
6.2 Using your own SSL certificate.....	21
6.2.1 To stop browser alerts.....	21
6.2.2 To install a certificate signed by a Certificate Authority.....	21
6.2.3 To install a self-signed certificate.....	22
7 Configuration reference.....	23
7.1 Configuration files.....	23
7.1.1 Simple installation.....	23
7.1.2 Advanced installation.....	23
7.1.3 Format of the configuration files.....	23
7.2 Main.cfg.....	24
7.2.1 ServerTimeZone.....	24
7.3 DeskNowBase.cfg.....	25
7.3.1 %SET%DESKNOWDATA.....	25
7.4 Log.cfg.....	26
7.4.1 BaseFileName.....	26
7.4.2 FileANDConsole.....	26

7.4.3 Debug	26
7.4.4 InternalError	26
7.4.5 Security	26
7.4.6 Mail	26
7.4.7 SMTP	26
7.4.8 Login	26
7.4.9 Admin	27
7.4.10 Database	27
7.4.11 EmbeddedMailServer	27
7.4.12 Messenger	27
7.4.13 CleanupDays	27
7.5 Database.cfg	28
7.5.1 JdbcDriver	28
7.5.2 Url	28
7.5.3 Username	28
7.5.4 Password	28
7.5.5 NumConnections	28
7.5.6 ConnectionPoolTimeout	28
7.5.7 IDFactory.DBAllocatedPoolSize	28
7.5.8 IDFactory.ChunckedPoolSize	28
7.5.9 IDFactory.NumProducers	29
7.6 Zroots.cfg	30
7.7 Vroots.cfg	31
7.8 Registration.cfg	32
7.8.1 MinCommunityNameLength	32
7.8.2 MinUserNameLength	32
7.8.3 MinPasswordLength	32
7.8.4 InitialFiles section	32
7.8.5 WelcomeEmail section	32
7.8.6 ReservedNames section	32
7.9 VisibleFolders.cfg	33
7.10 Security.cfg	34
7.10.1 AllowSuLoginFromCommunityManager	34
7.10.2 MaxFileUploadSize	34
7.10.3 ShowServerFiles	34
7.11 Mail.cfg	35
7.11.1 Domain	35
7.11.2 Administrator	35
7.11.3 AddressSeparators	35
7.11.4 ExternalPOP3	35
7.11.5 ExternalPOP3.delay	35
7.11.6 ExternalPOP3.fetchThreads	36
7.11.7 ExternalPOP3.parseThreads	36
7.11.8 ExternalPOP3.rootPath	36
7.11.9 ExternalPOP3.retryDelay	36
7.11.10 ExternalPOP3.retryMaxCycles	36
7.11.11 ExternalPOP3.CopyFetchedMailToPOP3Inbox	36
7.11.12 DirectIn	36
7.11.13 DirectIn.use	36
7.11.14 DirectIn.delay	36
7.11.15 DirectIn.parseThreads	37
7.11.16 DirectIn.rootPath	37
7.11.17 DirectIn.retryDelay	37
7.11.18 DirectIn.retryMaxCycles	37
7.11.19 DirectIn.RecipientPrefix	37
7.11.20 SMTP.use	37

7.11.21 ForceSMTPSend.....	37
7.11.22 SMTP.host and SMTP.port .....	37
7.11.23 SMTP.delay.....	38
7.11.24 SMTP.rootPath.....	38
7.11.25 SMTP.retryDelay.....	38
7.11.26 SMTP.retryMaxCycles.....	38
7.11.27 SMTP.CheckInternalRecipientExistence .....	38
7.11.28 EmbeddedServer .....	38
7.11.29 EmbeddedServer.use.....	38
7.11.30 EmbeddedServer.SMTPO daemonPort.....	38
7.11.31 EmbeddedServer.routingThreads.....	38
7.11.32 EmbeddedServer.remoteDeliveryThreads.....	39
7.11.33 EmbeddedServer.remoteDeliveryRetryDelay .....	39
7.11.34 EmbeddedServer.remoteDeliveryTimeout.....	39
7.11.35 EmbeddedServer.remoteDeliveryRetryMaxCycles.....	39
7.11.36 EmbeddedServer.SMTPThreads.....	39
7.11.37 EmbeddedServer.rootPath.....	39
7.11.38 EmbeddedServer.usePOP3.....	39
7.11.39 EmbeddedServer.POP3Threads .....	39
7.11.40 EmbeddedServer.authorizeRelay .....	39
7.11.41 EmbeddedServer.acceptSMTPAuthentication.....	40
7.11.42 EmbeddedServer.verifySMTPIdentity .....	40
7.11.43 EmbeddedServer.localDomain .....	40
7.11.44 EmbeddedServer.SMTPForwardHost.....	41
7.11.45 EmbeddedServer.SMTPForwardPort .....	41
7.11.46 EmbeddedServer.SMTPForwardUsername .....	41
7.11.47 EmbeddedServer.SMTPForwardPassword.....	41
7.11.48 EmbeddedServer.SynchronizeWebmailWithPOP3Read.....	41
7.11.49 EmbeddedServer.SynchronizeWebmailWithPOP3Delete .....	41
7.11.50 EmbeddedServer.DetectTimeLimitedEmailAddresses .....	41
7.11.51 EmbeddedServer.DNSServer .....	42
7.11.52 OutputCharset.....	42
7.12 Messenger.cfg .....	44
7.12.1 use.....	44
7.12.2 PlainConnector.use.....	44
7.12.3 PlainConnector.port.....	44
7.12.4 PlainConnector.NIO .....	44
7.12.5 SSLConnector.use .....	44
7.12.6 SSLConnector.port.....	44
7.12.7 WebConnector.use.....	44
7.12.8 RosterMode.....	45
7.13 Web.cfg.....	46
7.13.1 Layouts section .....	46
7.13.2 Custom section.....	46
7.13.3 Custom.BaseURL.....	46
7.13.4 Custom.original.DefaultLongDatePattern.....	46
7.13.5 Custom.original.DefaultShortDatePattern .....	46
7.14 International.cfg .....	48
7.14.1 DefaultLanguage .....	48
7.15 HTTP port.....	48
8 Language and timezone .....	49
8.1 Default language and user language.....	49
8.2 Adding a language to DeskNow .....	49
8.2.1 Translation tips .....	50
8.2.2 Testing your progress.....	50
8.2.3 Share your work .....	51

8.2.4 What happens when a new version of DeskNow is released .....	51
8.3 Time zone .....	51
9 Administration API .....	52
9.1 Overview .....	52
9.1.1 Invocation result .....	52
9.2 Create user .....	52
9.3 Create community .....	53
9.4 Create domain .....	53
9.5 Authenticate user .....	53
9.6 Post to message board .....	54
9.7 Send IM alert.....	54
9.8 Send IM broadcast.....	54
10 Licensing DeskNow .....	56
11 Support .....	57

# 1 Introduction

This manual assumes that you have already successfully installed DeskNow, either using the self-installing executable in Windows, or performing a custom setup. The procedures are detailed respectively in the manuals “DeskNow – Simple setup” and “DeskNow – Advanced setup”.



# 2 Basic administration

## 2.1 Starting and Stopping DeskNow

The way you start and stop DeskNow server depends on the installation that you performed.

### 2.1.1 Simple installation

From the Windows Start menu, select “Programs/DeskNow/Start DeskNow” and “Stop DeskNow” respectively. If you are running DeskNow on Windows NT, 2000 or XP, you have probably also installed DeskNow as a service. In this case, DeskNow will start automatically when the system starts.

### 2.1.2 Advanced installation

Modern application servers/servlet engines such as Tomcat version 4 can stop the web application from the administration page, using the “manager” web application. *Please note that in some Tomcat configurations, especially when it is connected to a web server, it is necessary to restart the web server as well.*

# 3 User administration

The Administrator ('admin') is the only user account that DeskNow creates automatically the first time it is run. You will need to create accounts for all the other users of the system.

## 3.1.1 Domains

DeskNow users are organized in **domains**. A domain generally represents a separated organization. It usually has its own associated internet domain name (ex. company1.com). Domains enable ISPs to offer hosted DeskNow services to multiple customers using a single DeskNow server.

Each domain is completely separated from the others. It is as if each domain was running on its own DeskNow server. Users of a domain cannot use collaboration features (apart from some exceptions, like public sharing, and, of course, email) with users of another domain. This ensures security and privacy for each domain.

Each domain has its own domain administrator, communities (see below) and users. User accounts with the same name can exist in different domains without conflict. For example, there can be joe.smith in the domain company1.com and joe.smith in the domain company2.com . They are completely different users.

When run for the first time, DeskNow contains only one domain, called the *default* domain. If you are not an ISP, you can completely ignore the use of domains and create users within your default domain. Otherwise, you can create as many domains as you want. Additional domains are also known as **virtual domains**.

Each domain, including the default one, has a special user account: **admin**. The admin user is the domain administrator, and has total control over the domain. He/she can create and manage communities (see below), user accounts and assign disk quotas to them. He/she can create mail aliases for the domain, and also create public folders that are accessible to every user of the domain.

The admin user of the default domain (very likely you, the reader of this document) is called **super administrator**. This user is the most powerful user of DeskNow. He/she can create and manage other domains, and can create or delete users in any domain.

Important: DeskNow uses the name of domains to determine what to do with incoming mail messages (e.g. to decide which accounts they must be delivered to). It is therefore essential that the name of a DeskNow domain is identical to the internet domain name for which DeskNow is receiving emails (ex. company.com).

## 3.1.2 Communities

DeskNow offers you the possibility of organizing user accounts of a domain into *communities*. A community is a group of users within a domain that is administered by a *community manager*.

Community managers can administer (create, change, delete) accounts within their community, becoming a sort of 'admin' user for the specific community. In this way, the domain Administrator can offload part of his tasks and responsibilities, with better response times for everyone.

For small organizations, you probably don't need to group your users in communities. All the users can belong to the default community, managed by the Administrator.

In large organizations however, planning the use of communities is a good way of decentralizing the administration of accounts.

If your organization has two branches, for instance, you can create two different communities and appoint a person within each branch as manager of the respective community. In this way, the managers can create accounts for employees of their branch without having to contact the Administrator.

The domain Administrator keeps full control over all users of the domain, anyway.

### 3.1.3 Users

A user belongs to a specific domain and community. The disk usage of a user can be limited by the user's disk quota, the community disk quota, and the domain disk quota (if they are not set to unlimited).

There can not be two users with the same user name within a domain, but users with the same username can exist in different domains.

### 3.1.4 'Guest' users

Users can be marked as *guests*. Guest users have limited privileges in DeskNow:

- they cannot create personal file folders, calendars, message boards, etc. They can only access objects shared with them by other non-guest users
- they cannot use email functionalities

### 3.1.5 'Restricted' users

Users can be marked as *restricted*. Restricted users have almost all the privileges of normal users except that they cannot share personal objects (calendars, folders, etc.) with other users. They can access objects shared by normal users.

### 3.1.6 Planning disk quotas

A disk quota is a limit to the disk space that a user, a community, or a domain can use. Setting disk quotas prevents users to abuse the system, and to ensure that the system will not run out of disk space unexpectedly.

DeskNow provides two different disk quotas:

- the soft disk quota is the amount of disk space (expressed in bytes) that a user can normally use. If a user exceeds his soft disk quota, he will receive a warning message in his account status, which appears at every login. However, he will still be able to write files to his account.
- the hard disk quota is the maximum amount of disk space that an user account can use. When an user account reaches its hard disk quota, no new files can be created in his file system, and existing files cannot be extended. Obviously, the hard disk quota should never be less than the soft disk quota (DeskNow does not enforce this, anyway). Usually DeskNow sets the hard disk quota to be 120% of the soft disk quota.

Disk quotas can be applied to single users, communities and domains. The disk usage of a community is the sum of the disk usages of its users. The disk usage of a domain is the sum of the disk usages of its communities.

Note that all the mail messages and attachments contribute to the disk occupation of an account. This means that the disk quota can be reached by just receiving mail messages. This also means that if the hard disk quota is reached, DeskNow will NOT download new mail for the account. Depending on your configuration, *this could cause the loss of emails*. In a standard configuration, however, DeskNow keeps trying storing emails to a user account for some time, at repeated intervals, before giving up and discarding the message. Whenever it encounters a disk quota error, it reports the problem in its log files.

In addition, files, messages or attachments uploaded by a user into the shared space of another user will concur to the disk usage of the user sharing the space, and not of the uploading user.

Users should always check their disk usage and remove unnecessary files, mails, messages from their account, or contact the administrator to increase their disk quota.

Anyway, you can decide to NOT apply disk quotas to users, communities or domains, by setting their disk quota to "Unlimited".

It is also possible to exclude some sections of the file system from disk quota calculation. To do so, simply create a file named "x\_DNFSToQuota" (can be empty) in the directory that you want to exclude from disk quota calculation. Any operation (file delete, create, write) in this directory or a subdirectory will not affect the diskquota of users.

### 3.1.7 Planning the virtual file system

Before creating users, it is important to plan how the file system that they will be able to access is composed. In other words, what default branches will be displayed under the "Files" node on the client. Consult section 7.7 for more details.

### 3.1.8 Creating users, communities and domains

The Administrator can create users and communities, manage quotas and passwords, by clicking on the "Administration" node on DeskNow. With the same link, the super Administrator can also manage domains.

**Important:** *a community must have a manager, and the manager must be a user of that community, or the domain administrator.*

Administrators of a community can create, manage, add users in their community as well, by clicking on the node corresponding to their community.

*DeskNow has the possibility to create user accounts in many other ways, including online user registration for publicly accessible sites, or with automated procedures. These services are available to licensed customers, along with many more backend integration and automation facilities. Please contact us for more details.*

### 3.1.9 Bulk creation of user accounts

DeskNow lets you create many user accounts in a single, fast operation, by importing the accounts from a spreadsheet. This is useful when setting up a new domain which must have many users.

All you need to do is to create a spreadsheet: each row represents an account, and the spreadsheet must have 5 columns:

- community name
- username
- password
- disk quota (in Mb, use -1 for unlimited)

- user type: must be one of the three values: normal, guest, restricted

Example:

	A	B	C	D	E
1	tech	richard	mypassword	100	restricted
2	tech	malcom	password1	-1	normal
3	sales	joe	aa1234	-1	normal
4	sales	frances	abcde12	50	guest
5	tech	sarah	password2	-1	normal

Once the spreadsheet is created, save it *using the CSV (comma separated values) format*. You can upload the file into DeskNow following the apposite link in the domain page (under Administration).

When you upload the file DeskNow will create all the accounts listed. Communities that do not exist will be automatically created. 'admin' will be the default manager, and they will have unlimited disk quota. User accounts that already exist will NOT be modified in any way.

### 3.1.10 Domains and login

How does DeskNow distinguish between users of different domains when they are logging in?

Users of the default domain can login by simply using their username and password.

Users of other domains have different options:

1. they can login by typing *username@domainname* in the username field of the login page (ex. joe@company1.com)
2. if (as it should be if you want email to work properly for a domain) the login page can be reached by using the domain name in the internet address, users can simply use their username and DeskNow will automatically match the domain from the URL.

Example: your DeskNow login page is accessible at [www.isp.com](http://www.isp.com) **and** [www.company1.com](http://www.company1.com) (or [abc.company1.com](http://abc.company1.com), it doesn't matter) because of DNS settings. If user 'joe' accesses the login page at [www.isp.com](http://www.isp.com) and types 'joe' in the username field, DeskNow will verify his credentials against the user account 'joe' in the domain [www.isp.com](http://www.isp.com). If user 'joe' accesses the login page at [www.company1.com](http://www.company1.com) (the page is the same, on the same DeskNow server: only the URL is different!) and types 'joe' in the username field, DeskNow will match his credentials against the user account 'joe' in the domain [www.company1.com](http://www.company1.com).

Note that DeskNow always finds the best match between the URL and the registered domains, i.e. it can distinguish between [xxx.abc.org](http://xxx.abc.org), [yyy.abc.org](http://yyy.abc.org) and [abc.org](http://abc.org).

### 3.1.11 Domains and POP3 and SMTP authentication

Unfortunately the POP3 and SMTP protocols do not provide a standard way to handle per-domain authentication.

Users of the default domain can authenticate themselves simply using their username and password.

Users of other domains must authenticate themselves by using *username—domainname* (that's 2 dash "-" characters) as their POP3/SMTP username.

Ex.: joe—company1.com

### 3.1.12 Su-like login

The su-like login is a mean for the Administrator to login as any user, without knowing the appropriate password. This is equivalent to the 'su' command in Unix systems, hence the name.

This feature is very useful to perform particular tasks on an account, without involving the user directly and without changing passwords.

To login as user 'joe' for instance, the Administrator can simply use the following login:

username: %admin%joe

password: the Administrator's password

Administrators of virtual domains can control their own users by logging in with  
*%admin%joe@company1.com*

The super administrator can login as any user, of any domain, using

*%admin%joe@company1.com*

### 3.1.13 Su-like login for community managers

It is possible to enable the su-like login also for community managers, so that they can login as users of their community (they cannot login as a user of a different community, however).

To enable this feature, set the parameter AllowSuLoginFromCommunityManager to TRUE in \$(DESKNOW\_WEBAPP)/WEB-INF/cfg/Security.cfg.

If 'manager1' is manager of 'community1' and 'joe' is a member of 'community1', manager1 can login as joe by using the following login:

username: %manager1%joe

password: password for 'manager1'

# 4 Mail

DeskNow is a full SMTP and POP3 mail server.

DeskNow can receive and send emails on behalf of every domain registered with it.

Users have two ways of accessing and sending emails:

- using the web-based interface (webmail)
- using a traditional mail client (Outlook, Eudora, etc.)

DeskNow can also automatically retrieve emails on behalf of users from other POP3 servers.

DeskNow can be used as standalone mail server, or in conjunction with another existing mail server, leaving to it the SMTP and POP3 functionalities and adding webmail capabilities.

DeskNow can also be used as a closed system, i.e. for internal mail only, preventing users to send mail outside DeskNow.

## 4.1.1 DeskNow as MTA (Mail Transfer Agent)

In order for DeskNow to work as a full mail server that can receive and send mail from/to the Internet, the “embedded mail server” in DeskNow must be enabled (see the Configuration wizard or section 7.11.29).

You will also need to setup the MX records for every Internet domain that corresponds to a domain in DeskNow to point to your DeskNow server. Obtain help from your Internet Registrar on how to do this.

Finally, the TCP port 25 must be open between your DeskNow server and the Internet, in both directions. Check your firewalls to make sure this is the case.

## 4.1.2 DeskNow as SMTP server

DeskNow supports the standard SMTP Internet protocol to transfer emails. By default, DeskNow does not allow anyone (except from applications running on the computer where DeskNow is running, and users sending emails via the webmail interface) to send emails to the external world (“mail relay”). This is to prevent abuse from spammers.

You can authorize SMTP clients to send emails outside in two ways (they can co-exist):

- by setting a pool of “trusted” IP addresses (ex. your office LAN): every computer connecting to the DeskNow SMTP server from a trusted address will be able to send emails anywhere. This can be done using the Configuration Wizard, or by manually editing the server properties (see section 7.11.40)
- by using SMTP authentication: users must configure their mail client to send their DeskNow username and password before trying to send emails outside (see sections 7.11.41 and 7.11.42)

Note: users of virtual domains (i.e. not the default domain) must use *username--domainname* (2 dash characters) as their SMTP username. Ex.: joe—company1.com

## 4.1.3 SMTP Forwarding

DeskNow can be configured to forward all SMTP traffic to another SMTP server (ex. the server of your ISP) without attempting to deliver messages directly. This is useful in many

circumstances (ex. the DeskNow server is behind a firewall, you want to use additional functionalities provided by the external SMTP server such as filtering/logging, etc.).

#### 4.1.4 DeskNow as POP3 server

Besides webmail, users can retrieve emails from their DeskNow inbox using a traditional POP3 client.

DeskNow supports **POP3 synchronization with webmail**, in other words you can configure DeskNow so that when users download messages using POP3, the messages are automatically marked as “read” in the webmail interface (this is the default for new installations). See section 7.11.48.

You can additionally configure DeskNow to delete messages from the webmail interface when they are removed from the POP3 mailbox (see section 7.11.49).

Note: users of virtual domains (i.e. not the default domain) must use *username--domainname* (2 dash characters) as their POP3 username. Ex.: joe—company1.com

#### 4.1.5 More configuration options

DeskNow is a very flexible mail server with many configuration parameters. See section 7.11 for more information.



# 5 Antivirus scanning

**Antivirus scanning is available for the default DeskNow Lite license only for a trial period of time (30 days from the date of installation). After this period, it will not be available. See chapter 8 for information on how to purchase a DeskNow license.**

DeskNow can integrate with most professional antivirus software to scan all incoming / outgoing emails for dangerous attachments. DeskNow can also request the virus scanning of all the files uploaded to the system, either via the web interface or via WebFolders.

To configure antivirus scanning, you need an antivirus that supports command line operation, i.e. it must provide a command that can be invoked to scan a single file only. Most commercial antivirus software allows this.

DeskNow has been tested with the following antivirus software:

- Norton Antivirus 2003
- F-Secure Antivirus 2003
- McAfee Antivirus Command Line edition

Many other antivirus applications can be used without problems, as long as they support command line invocation.

To configure antivirus scanning:

1. login as admin user
2. click on Administration and then on Antivirus
3. enable antivirus scanning by checking the appropriate check box
4. select one of the predefined antivirus commands, and adjust the path of the command depending on where you installed the antivirus
5. if your antivirus software is not listed in the predefined commands, find out if it can be invoked by command line (most are), and what is the required syntax.

Typically the command line will be like "C:\Program Files\MyAntivirus\scan.exe" followed by some options and the full path of the file to scan. Use %FILE% to specify that parameter.

For example, the command line for Norton Antivirus 2003 is the following:  
"C:\Program Files\Norton AntiVirus\navw32.exe %FILE% /NORESULTS"

Every antivirus return a result code to indicate if the file was infected. The result code for a clean file is typically 0, but it is a good idea to double check with the documentation. You need to specify the result code for success in the apposite field.

Please inform us of the suitable command line for other antivirus products, so that we can add it to future releases to help other users.

6. When you press OK, DeskNow will try to invoke the antivirus to scan a clean file, and will expect to receive the success result code. It will display a message if an error occurred. In this case, check that the command line specified corresponds to your antivirus (it may have been installed in a different location) and retry.

**Important:** antivirus integration is available in DeskNow Lite only for a trial period. After this period is expired, files and attachments will not be scanned, and messages in both the Security and InternalError logs will be created, to warn you of the insecurity. You can

then decide to purchase a DeskNow license, or disable antivirus scanning to avoid the warnings.

## 5.1 Antivirus scanning for Linux

Popular command line antivirus scanners exist for Linux, and they can scan for viruses that are targeting Windows computers. This is useful when your users run Windows computers but you prefer to use Linux for your mail server.

We recommend Panda Antivirus for Linux, (<http://www.pandasoftware.com>), which is free.

The antivirus has a semi-interactive interface, so you need to use a small script in order to integrate it into DeskNow. Here's how to:

1. install the antivirus
2. run it once, to review the license agreement
3. create a file made of 10 "space" characters and save it in `/usr/local/bin/spaceinput`
4. create a file called `/usr/local/bin/myscan.sh`, with the following content:

```
#!/bin/sh
/usr/bin/pavcl $1 -NOR -NSO < /usr/local/bin/spaceinput > /dev/null
```
5. make `myscan.sh` executable:

```
chmod a+x /usr/local/bin/myscan.sh
```

Now you can configure DeskNow to use the following command for virus scanning:

```
/usr/local/bin/myscan.sh %FILE%
```

The antivirus success code is 0.

# 6 Common configuration tasks

This chapter describes common configuration tasks that can be used to customize DeskNow.

## 6.1 Using a different database

By default DeskNow uses its own embedded SQL database to store user information. The embedded database used is McKoi DB (HypersonicSQL for earlier releases), a popular open source database.

McKoi DB is a mature product, used in many commercial applications.

However we recommend its use only for sites with a small number of users (20-30 maximum). For larger organizations, we suggest using a standalone database.

DeskNow currently supports two popular RDBMS:

- Microsoft SQL Server 2000
- PostgreSQL (see <http://www.postgresql.org> for more details) – free and open source

MySQL is currently not supported, since many of its implementations still don't have a stable support for some important SQL features (transactions and referential integrity).

If you want to use one of the supported databases, you need to complete three steps:

- creation of the database schema
- installation of the JDBC (Java Database Connectivity) drivers
- configuration of DeskNow

The three steps are detailed in the following sections.

### 6.1.1 Creation of the database schema

*In our example we will assume that you decide to replace the embedded database with Microsoft SQL Server 2000, running on localhost, and accepting connections on the default port (1433).*

According to the documentation provided with your database system, create a database named 'desknow'. Create an user account 'desknow\_server' with read, write and DDL permissions over the database created. It is important that the account have DDL permissions (i.e. that it is able to create, alter or delete tables and indexes), otherwise server upgrades will not run properly.

*In our example, we will assume that the password for the account 'desknow\_server' is 'password'. We recommend using a stronger password in a real system. Also note that you can choose a different name for the database and for the account. desknow\_server must have db\_owner privileges on the database.*

*IMPORTANT: the account must be an SQL account, and not a Windows logon account*

Depending on your DBMS, pick from the directory 'sql' in the installation package the sql script that is appropriate. If you used the Windows installer, you will find this directory in a location like C:\Program Files\DeskNow\sql .

*In our example, the correct script file is 'desknow\_sqlserver2000.sql'.*

According to the documentation provided with your DBMS, execute the SQL script within the created database. The script will create all the tables and indexes necessary.

*In our example, start the 'Query Analyzer' tool of SQL Server and connect as desknow\_server to the desknow database. Open the file 'desknow\_sqlserver2000.sql' located on the CD and execute it.*

## 6.1.2 Installation of the JDBC drivers

DeskNow needs appropriate JDBC drivers to connect to your database. Every major database system has JDBC drivers that can be used, and often there are many versions provided by different vendors.

*To connect to SQL Server, you can download and install the free Microsoft JDBC driver (available from the downloads in the SQL Server section at <http://www.microsoft.com>). Other drivers provided by different vendors are available.*

JDBC drivers usually come as one or more .jar files. To install your driver into DeskNow, you just need to copy the .jar files into the "lib" directory of the DeskNow webapplication. If you used the Windows installer to install DeskNow, this directory is usually in a location like C:\Program Files\DeskNow\webapps\desknow\WEB-INF\lib. If you used the manual installation, this directory is in \$(DESKNOW\_WEBAPP)/WEB-INF/lib .

*In our example, install the downloaded Microsoft JDBC driver, then copy all the .jar files in the "lib" directory of the JDBC driver in the "lib" directory of DeskNow (ex. C:\Program Files\DeskNow\webapps\desknow\WEB-INF\lib)*

## 6.1.3 Configuration of DeskNow

You now need to tell DeskNow to use your new driver, and what parameters to use to connect to your database.

All you need to do is to set four parameters in the Database.cfg file. See section 7.5 for a detailed reference. In particular, these are the parameters that you need to change:

- JdbcDriver
- Url
- Username
- Password

*Following our example, edit Database.cfg and comment out (by typing a # character at the beginning of the line) the section for the embedded database. Then uncomment the section for SQL Server 2000. The default parameters are already set, but you may need to change Username and Password:*

```
JdbcDriver    com.microsoft.jdbc.sqlserver.SQLServerDriver
Url           jdbc:microsoft:sqlserver://localhost:1433;
              SelectMethod=cursor;DatabaseName=desknow

Username     desknow_server
Pasword      password
```

### 6.1.4 Restart DeskNow

Once the three steps are completed, you need to stop and restart DeskNow for the changes to take effect.

If you used the Windows installer to install DeskNow, you may want to start DeskNow in console (see the Advanced Startup submenu in the DeskNow program group) at least the first time, so that you can immediately check for error messages should they appear. Error messages are logged to the normal log files, anyway.

## 6.2 Using your own SSL certificate

**Important: the following instructions apply only if you installed DeskNow using the automatic installer for Windows. If you installed DeskNow manually, please refer to the "DeskNow – Advanced Setup" manual for more information.**

**SSL is available for the default DeskNow Lite license only for a trial period of time (30 days from the date of installation). After this period, it will not be available. See chapter 8 for information on how to purchase a DeskNow license.**

When DeskNow is installed using the automatic installer for Windows, it comes with a built-in test SSL certificate. This certificate is not signed by a Certification Authority normally trusted by web browsers (ex. Verisign, Thawte, etc.). This usually means that although SSL is enabled and communications using the HTTPS protocol are encrypted, your browser may display alert messages to inform you that the certificate is not fully trusted.

### 6.2.1 To stop browser alerts

These instructions apply to Microsoft Internet Explorer. The procedure for other browsers is usually very similar.

When you open DeskNow using HTTPS, the browser will popup an alert message. Click on "View certificate", then click on "Install Certificate".

### 6.2.2 To install a certificate signed by a Certificate Authority

To install your own SSL certificate, signed by a trusted Certification Authority, follow these steps:

1. open a Command Prompt window and type the following commands in **bold**
2. change the current directory to c:\Program Files\DeskNow\conf (adjust if you installed DeskNow in a different location): **cd "c:\Program Files\DeskNow\conf"**
3. backup the current keystore (a file where all tomcat certificates are kept): **move keystore keystore.bak**
4. create your own untrusted certificate: **..\java\bin\keytool -genkey -validity 730 -keystore keystore -alias tomcat -storepass changeit** (note: if you want to set a password different from 'changeit', you will have to edit server.xml)
5. **IMPORTANT:** when prompted for your first and last name, *type the address of your server, for example: www.myhost.com*
6. answer all the following questions
7. when requested for a key password, press ENTER
8. generate a certificate signing request: **..\java\bin\keytool -certreq -alias tomcat -file mycert.csr -keystore keystore -storepass changeit**

9. using the generated mycert.csr file, request a trusted certificate from a Certification Authority (es [Verisign](#), [Thawte](#)). Follow their instructions on how to submit the csr.
10. the Certification Authority will return you a trusted certificate, in the form of a certificate.crt file (the file name could change, but the extension should be .crt). This process could take a few days.
11. import the trusted certificate in your keystore: **..java\bin\keytool -import -alias tomcat -file certificate.crt -keystore keystore -storepass changeit**
12. stop and restart DeskNow.

### 6.2.3 To install a self-signed certificate

If the default self-signed certificate is not suitable for you, you can create a new self-signed certificate.

1. open a Command Prompt window and type the following commands in **bold**
2. change the current directory to c:\Program Files\DeskNow\conf (adjust if you installed DeskNow in a different location): **cd "c:\Program Files\DeskNow\conf"**
3. backup the current keystore (a file where all tomcat certificates are kept): **move keystore keystore.bak**

type the following command: **..java\bin\keytool -selfcert -keystore keystore -alias tomcat -storepass changeit -dname "cn=www.mydomain.com, ou=Administration, o=My Company, c=UK"** (use appropriate values for domain, organizational unit, company name and country code)

# 7 Configuration reference

This chapter provides a reference for all the configuration parameters of DeskNow server.

## 7.1 Configuration files

The configuration parameters are contained in several files, all with extension “.cfg”, that are located in \$(DESKNOWDATA)/cfg. \$(DESKNOWDATA) is the directory where all DeskNow data is stored, for example “C:\desknowdata” or “/var/desknowdata”.

***If you make any change to any configuration file, you must restart DeskNow for the change to take effect.***

### 7.1.1 Simple installation

To edit the configuration files, from the Windows Start menu select “Programs/DeskNow/Configuration/” and then one of the “Edit xyz.cfg” items.

### 7.1.2 Advanced installation

To edit the configuration files, open them with a normal text editor.

### 7.1.3 Format of the configuration files

Configuration files are normal text files. Parameters are expressed by a parameter name followed by a parameter value, like:

*Username*                      *desknow\_server*

If the parameter value needs to include spaces, you can enclose it in double quotes:

*Description.1*                      *"Your public web pages"*

The character # marks the beginning of a **comment**. Everything on a line after the character # is ignored by DeskNow.

Configuration files are **case sensitive**.

## 7.2 Main.cfg

This is the configuration file for the main servlet, which is in charge of handling all the requests coming from the web client.

This file mainly includes other configuration files, which are described in the following sections.

### 7.2.1 ServerTimeZone

See section 8.3 for an overview of how time zones are used in DeskNow.

You can use this property to change the default time zone. See the file `timezones.txt` in the `/docs` folder for possible values to use for this property.

It is also possible (and easier) to use the Configuration Wizard in graphical mode to change the time zone.



## 7.3 DeskNowBase.cfg

Defines the base path for the DeskNow directories and files.

### 7.3.1 %SET%DESKNOWDATA

This is a variable setting, used by many configuration files. The value set for the variable should be the full path of the “desknowdata” folder that you copied from the CD.

*Advanced configurations may not store all the files under the same base, and therefore may not need this variable.*

*Example:*

```
%SET%DESKNOWDATA      "c:\desknowdata"
```

## 7.4 Log.cfg

Defines what is logged, and where.

### 7.4.1 BaseFileName

There is a different log file for every enabled logging flag (see below). Every day a new file is opened, automatically. The flag and date are automatically appended to what you specify here.

*Example: if you specify "c:\desknowdata\log\log", DeskNow will produce log files like "c:\desknowdata\log\log.SMTP.2001-07-19.txt"*

### 7.4.2 FileANDConsole

Specify whether log messages must be sent to the console in addition to the log files. Can be either TRUE or FALSE.

### 7.4.3 Debug

Enable / disable the logging of debug messages. Can be either TRUE (enable) or FALSE (disable).

IMPORTANT: debug logging can generate very big log files.

Recommended: FALSE

### 7.4.4 InternalError

Enable / disable the logging of internal server errors. Internal server errors can be due to incorrect configuration (i.e. physical file folders missing, incorrect configuration files, etc) or to problems in the server's code. If an internal server error occurs, please examine carefully the log file. If the information displayed does not point to a configuration problem, please contact the support service, providing all relevant information (i.e. what operation caused the problem, etc.).

Can be either TRUE (enable) or FALSE (disable).

Recommended: TRUE.

### 7.4.5 Security

Enable / disable the logging of security related events (failed logins, suspect activity and so on). Can be either TRUE (enable) or FALSE (disable).

Recommended: TRUE.

### 7.4.6 Mail

Enable / disable the logging of incoming mail activity. Can be either TRUE (enable) or FALSE (disable).

IMPORTANT: mail logging can generate big log files. At the beginning, however, it is very useful to track problems.

### 7.4.7 SMTP

Enable / disable the logging of SMTP activity. Can be either TRUE (enable) or FALSE (disable).

### 7.4.8 Login

Enable / disable the logging of logins. Can be either TRUE (enable) or FALSE (disable).

Failed login attempts are logged under the "Security" flag.

### **7.4.9 Admin**

Enable / disable the logging of messages useful to the Administrator. Can be either TRUE (enable) or FALSE (disable).  
Recommended: TRUE

### **7.4.10 Database**

Enable / disable the logging of database activity. Can be either TRUE (enable) or FALSE (disable).  
IMPORTANT: database logging can generate very big log files and slow down server operations, and should be used only for performance tuning.  
Recommended: FALSE

### **7.4.11 EmbeddedMailServer**

Enable / disable the logging of activity of the DeskNow mailserver (SMTP and POP3).  
Can be either TRUE (enable) or FALSE (disable).  
Recommended: TRUE

### **7.4.12 Messenger**

Enable / disable the logging of activity of the DeskNow Instant Messaging server. Can be either TRUE (enable) or FALSE (disable).  
Recommended: TRUE

### **7.4.13 CleanupDays**

Sets the number of days after which old log files are automatically deleted.

## 7.5 Database.cfg

Defines how DeskNow connects to the database. Additionally, defines how IDs are allocated (advanced feature).

### 7.5.1 JdbcDriver

Java class implementing the JDBC driver. Consult the driver documentation for more details. The class must be on the CLASSPATH of the Java Virtual Machine running DeskNow.

*Example:* `com.microsoft.jdbc.sqlserver.SQLServerDriver`

### 7.5.2 Url

URL addressing the database. Consult the JDBC driver documentation for more details.

*Example:* `jdbc:microsoft:sqlserver://localhost:1433;DatabaseName=desknow`

### 7.5.3 Username

The username to access the database. The account must have read, write and DDL privileges.

*Example:* `desknow_server`

### 7.5.4 Password

The password to access the database.

*Example:* `password`

### 7.5.5 NumConnections

Number of simultaneous connections to the database. If the number of concurrent requests at a given time is higher than this value, some requests will have to wait (and possibly be aborted if they wait too long: see `ConnectionPoolTimeout`). The server reserves one connection for special internal uses. This connection will not be available to fulfil normal client requests.

Generally, the number of database connections should be equal to the number of QMail, ExternalPOP3 and SMTP threads specified in Mail.cfg (see 7.11) PLUS the number of concurrent client requests that the server is expected to serve.

### 7.5.6 ConnectionPoolTimeout

Timeout before an operation is interrupted waiting for a connection to be available (expressed in seconds).

### 7.5.7 IDFactory.DBAllocatedPoolSize

(documentation incomplete, leave default values for now)

### 7.5.8 IDFactory.ChunckedPoolSize

(documentation incomplete, leave default values for now)

### **7.5.9 IDFactory.NumProducers**

(documentation incomplete, leave default values for now)

## 7.6 Zroots.cfg

Defines multiple root bases for the native filesystem.

A root is a base path onto which a vroot is mapped (see Vroots.cfg).

A root usually points to a directory in the native filesystem.

Under this directory, for every user account there is a directory named with the username of the account.

*Example:*

*The root "webfiles" is mapped to the folder  
"c:\desknowdata\userfolders\webfiles".*

*The vroot WebFiles is mapped to the root "webfiles" , and is visible to the user  
(as defined in Vroots.cfg).*

*As a result, when user "joe.smith" logs in, he will see "WebFiles" in the DeskNow  
tree, and the files contained in it are the files actually contained in  
"c:\desknow\userfolders\webfiles\joe.smith"*

Thanks to DeskNow double-layered virtual filesystem, it is very easy to manage user's data.

For instance, if a single disk is not large enough to store all the files, it is possible to store all the "WebFiles" files in a second disk, just by changing the native path of the associated root.

The default configuration, however, maps all the roots under {DESKNOWBASE}/userfolders.

Syntax:

<zroot> <native directory path>

as a convention, roots are all in lowercase

It is strongly discouraged to have roots that differ only by their case.

*Example:*

```
%SET%ZROOT_BASE $(DESKNOWDATA)/userfolders
```

```
webfiles      $(ZROOT_BASE)/web
```

## 7.7 Vroots.cfg

Defines multiple roots in the virtual file system of a user.  
A root is a "virtual root" for the virtual file system of a user.

Not all the roots are visible to the users: some of them are used for internal services only (such as storing email attachments, and so on).  
Every vroot is mapped to a zroot (see file Zroots.cfg), which in turn is mapped to a physical directory on the native filesystem.

The roots specified here are created for every new account.  
Once the account is created, the mappings for that account are saved in the database, and not read from this file.

For the reason above, it is very easy to distribute the load among multiple disks, as the system grows:

for instance if you already have 1000 users and you want to use a second disk for the mail attachments of new users, it is sufficient to create a new zroot "inmail2" (in file Zroots.cfg) and then change the vroot "InMail" in this file to point to "inmail2" instead of "inmail".

While old users will keep using the old disk, new accounts will use the new disk.

Some roots are "system" roots, and are needed by the server to work properly. You can chose how to map them to zroots, but you cannot remove them. The following are system roots: InMail, Mbm, Cts, Temp, ServerIcons, Wombats.

Syntax:

<vroot> <zroot>

As a convention, zroots are all in lowercase, whereas vroots are not.  
It is strongly discouraged to have vroots that differ only by their case.

## 7.8 Registration.cfg

### 7.8.1 MinCommunityNameLength

Minimum length for a community name. Creation of communities with a shorter name will be refused.

### 7.8.2 MinUserNameLength

Minimum length for a username. Creation of user accounts with a shorter username will be refused.

### 7.8.3 MinPasswordLength

Minimum length for a user password. Attempts to set or change a user password shorter than this value will fail.

### 7.8.4 InitialFiles section

This section instructs the server to copy to the user file system some predefined files when the account is created.

Syntax:

```
InitialFiles.<vroot>      <source path>
(copy all the files in <source path> to the user's vroot)
```

*Example: InitialFiles.WebFiles \$(DESKNOWDATA)/initialwebpages*

### 7.8.5 WelcomeEmail section

This section enables the delivery of a welcome email message in the inbox of newly created accounts. The section can be commented out. In this case no welcome email will be sent.

*Example:*

```
WelcomeEmail.From      admin
WelcomeEmail.Subject    "Welcome !"
WelcomeEmail.Body       $(DESKNOWDATA)/WelcomeEmailBody.txt
```

### 7.8.6 ReservedNames section

Defines names that cannot be registered as usernames or community names.

*Example:*

```
ReservedNames.root      1
ReservedNames.superman  1
```

Note that it is impossible anyway to create users or communities with a name that is already used by another user or community in the same domain. Different domains can have user accounts and communities with the same name.



## 7.9 VisibleFolders.cfg

Defines which parts of the user's file system are shown in the client.

*Example:*

*The following section states that the vroot "WebFiles" is to be shown on the client, under the "Files" node, with the name "Web Pages", and the description "Your public web pages".*

*The folder is published on the web with the url*

*"http://public.desknow.com/<username>"*

*NB It is the administrator's responsibility to set up the web server so that the folder is actually published!*

```
Vroot.1      WebFiles
PrettyName.1 "Web pages"
Description.1 "Your public web pages"
URL.1       http://public.desknow.com/
```

*The following section states that the vroot "MyDocs" is to be shown on the client, under the "Files" node, with the name "Personal Documents", and the description "Your personal documents".*

*The folder is NOT published on the web.*

```
Vroot.2      MyDocs
PrettyName.2 "Personal Documents"
Description.2 "Your personal documents"
```

***IMPORTANT: in many circumstances, the default DeskNow DirectFiles feature is more than enough to let users publish their documents on the web. The configuration presented here, however, illustrates an example of more complex uses of DeskNow.***

## 7.10 Security.cfg

Grants / denies particular access to features.

### 7.10.1 AllowSuLoginFromCommunityManager

Enables / disables su-like login for community managers (see section 3.1.13).  
Possible values are TRUE and FALSE.

### 7.10.2 MaxFileUploadSize

Determines the maximum allowed size for an upload operation. This is the sum of the sizes of all files uploaded in a single operation. This includes files uploaded as attachments.

The value is expressed in Mb.

### 7.10.3 ShowServerFiles

Enables / disables access to all the files on the native filesystem of the server.  
Possible values are TRUE and FALSE.

If TRUE, the Administrator account will have a **[All server files]** folder in the Files section. Under this folder are visible all the files on the server's file system, and all the normal operations can be performed on them. As Administrator you can also share parts of the native file system to users (ex. drives that contains documents).

## 7.11 Mail.cfg

Specifies the parameters for Mail management.

DeskNow can integrate with SMTP servers, POP3 servers and directly with Qmail servers, or can run as stand-alone mail server.

DeskNow can also disable external email, preventing users to send emails to the Internet, and functioning as internal messaging system.

**NOTE: the most important mail settings of DeskNow can be configured by the Configuration Wizard. If you have used the Windows installer, you can run the wizard from the DeskNow program group in the Start Menu.**

**If you have used the manual installer, the wizard can be run by changing the current directory to the bin directory contained in the package, and executing:**

**`./config.sh <path of desknowdata> (Unix/Linux)`**

**`config.bat <path of desknowdata> (Windows)`**

*Example:*

`./config.sh /var/desknowdata`

`config.bat c:\desknowdata`

### 7.11.1 Domain

This parameter is now obsolete, and replaced by virtual domain management. It has no relevance.

### 7.11.2 Administrator

All mail problems (malformed messages, etc.) will be notified to this email address.

*Example:*

`administratorEmail myname@myisp.com`

### 7.11.3 AddressSeparators

Specify the characters (each single one of them) are treated as address separators for the To, Cc, Bcc fields typed in by the user.

*Example:*

`AddressSeparators ;,`

### 7.11.4 ExternalPOP3

This section refers to the ExternalPOP3 Daemon that is inbuilt in DeskNow. This daemon is responsible of periodically fetching emails on behalf of the users from their **External accounts**.

### 7.11.5 ExternalPOP3.delay

When DeskNow starts, the daemon checks all the external accounts for all the users, and retrieves new emails. Then it waits for the amount of time specified by this parameter (expressed in seconds), and then repeats the cycle.

*Example:*

`ExternalPOP3.delay 600`

### 7.11.6 ExternalPOP3.fetchThreads

Sets the maximum number of concurrent threads that are used to access external POP3 servers and download new emails. Under light load, DeskNow will automatically reduce the number of threads in use as necessary.

### 7.11.7 ExternalPOP3.parseThreads

Sets the maximum number of concurrent threads that are used to decode and process new emails retrieved by the fetchThreads. Since mail decoding is usually less network dependant, the number of parseThreads should generally be lower than that of fetchThreads.

### 7.11.8 ExternalPOP3.rootPath

DeskNow uses this folder to process all the emails retrieved before they are assigned to the proper account.

### 7.11.9 ExternalPOP3.retryDelay

Sets the delay in seconds between two attempts to manage those emails that DeskNow was not able to decode or assign to an user (ex. because the user's disk usage has reached its limit).

### 7.11.10 ExternalPOP3.retryMaxCycles

Sets the number of times that an email will be processed before it will be considered undeliverable. Error emails are saved in the **error** folder under the rootPath. Since the ExternalPOP3Daemon is effectively a POP3 client, it will NOT automatically send an error message back to the sender.

### 7.11.11 ExternalPOP3.CopyFetchedMailToPOP3Inbox

If this flag is TRUE and the DeskNow POP3 server is enabled, DeskNow will store a copy of the mail retrieved from external POP3 accounts into your DeskNow POP3 inbox.

### 7.11.12 DirectIn

This section refers to the daemon that processes messages delivered by a mail server to a specific folder, using the qmail message format. This includes, of course, the **qmail** mail server itself, or the DeskNow integrated mail server.

### 7.11.13 DirectIn.use

Enables / disables the DirectIn Daemon. you can disable it if you are not using DeskNow as mail server, or are not using **qmail** as backend mail server. Since this is also used for internal mails, *this flag should always be set to TRUE*, unless you have very good reasons to do otherwise.

Possible values are TRUE (enable) or FALSE (disable).

### 7.11.14 DirectIn.delay

When DeskNow starts, the daemon checks the specific mail folder (see below for the rootPath setting) for new messages, and parses and assigns all the messages it has found. Then it waits for the amount of time specified by this parameter (expressed in seconds), and then repeats the cycle.

### 7.11.15 DirectIn.parseThreads

Sets the number of concurrent threads that are used to decode and process new emails retrieved by the daemon. Since mail decoding is usually very CPU and disk intensive, the number of parseThreads should generally be low, since very little performance would be gained by increasing the parallelism.

### 7.11.16 DirectIn.rootPath

DeskNow uses this folder to process all the emails retrieved before they are assigned to the proper account. In particular, DeskNow looks for new messages in the **base** folder located under the directory specified here. The embedded DeskNow mail server, or a gmail mail server, should be configured to deliver all the email for the domain into this folder.

### 7.11.17 DirectIn.retryDelay

Sets the delay in seconds between two attempts to manage those emails that DeskNow was not able to decode or assign to an user (ex. because the user's disk usage has reached its limit).

### 7.11.18 DirectIn.retryMaxCycles

Sets the number of times that an email will be processed before it will be considered undeliverable. Error emails are saved in the **error** folder under the rootPath. In case the email was addressed to an account that does not exist in DeskNow, an error email will be sent back to the sender.

### 7.11.19 DirectIn.RecipientPrefix

Specifies the default prefix that identify the mail recipient in a mail message in gmail format. The default value should not be changed unless there is a very good reason to do so.

### 7.11.20 SMTP.use

Enables/disables sending emails to the outside world from the web interface. Use FALSE to implement a closed system for internal communication only. Possible values are TRUE (enable) or FALSE (disable).

### 7.11.21 ForceSMTPSend

Sometimes DeskNow delivers emails sent from one user to another directly, without going through SMTP, to achieve better performance. This flag forces DeskNow to use the SMTP server to deliver all the emails. Make sure that SMTP.use is TRUE if you set this parameter to TRUE.

*Example:*  
`ForceSMTPSend        TRUE`

### 7.11.22 SMTP.host and SMTP.port

These settings define how DeskNow sends emails outside. They can point to the embedded DeskNow SMTP server (which is the default setting in most cases) on localhost:25, or to an external SMTP server (ex. your ISP's).

*Example:*  
`SMTP.host                localhost`  
`SMTP.port                25`

### 7.11.23 SMTP.delay

When DeskNow starts, the daemon checks a specific mail folder (see below for the rootPath setting) for new messages to be sent, and connect to the SMTP server to send all the messages. When a user instructs DeskNow to send an email, the daemon is waken up and it immediately sends the message. The daemon is however forced to wakeup after the amount of time specified here (in seconds), to check for astray messages that were not previously sent.

### 7.11.24 SMTP.rootPath

DeskNow uses this folder to process all the emails that are to be sent to the SMTP server.

### 7.11.25 SMTP.retryDelay

Sets the delay in seconds between two attempts to manage those emails that DeskNow was not able to send (ex. because the SMTP server could not be contacted).

### 7.11.26 SMTP.retryMaxCycles

Sets the number of times that an email will be processed before it will be considered undeliverable. Error emails are saved in the **error** folder under the rootPath. In case the email was not delivered, an error message is sent back to the DeskNow user that sent the email.

### 7.11.27 SMTP.CheckInternalRecipientExistence

Enables/disables checking for the existence of the recipient when using webmail. When sending emails via webmail, if the recipient is in a local domain, desknow can check if the recipient user exists in that domain, and warn immediately if it doesn't. This helps correcting immediately typos. In some configurations, especially when using an external mail server, you may want to disable this check (setting this to FALSE). Possible values: TRUE (check user existence) or FALSE (don't check: if the recipient mail server complains about the recipient, a mail delivery error message will be delivered back to the sender). The default is TRUE.

### 7.11.28 EmbeddedServer

DeskNow is a full-featured SMTP and POP3 server. This section is dedicated to its configuration.

### 7.11.29 EmbeddedServer.use

Enables / disables the use of the embedded mail server. Possible values are TRUE (enable) and FALSE (disable). If you enable the mail server, make sure that the DirectIn daemon (see above) is enabled as well.

### 7.11.30 EmbeddedServer. SMTPDaemonPort

Defines the port on which the SMTP server will accept connections. The internet standard is 25, so DO NOT CHANGE THIS unless you perfectly know what you're doing. If this value is different from 25, the DeskNow SMTP server will not be able to receive emails from the Internet, unless there is some SMTP gateway in front.

### 7.11.31 EmbeddedServer.routingThreads

Indicates the number of threads dedicating to routing (ie deciding whether a message is local to the domain, or needs to be delivered across the Internet).

Since this operation is not affected by network delays, the number of routingThreads should generally be low, since very little performance would be gained by increasing the parallelism.

### **7.11.32 EmbeddedServer.remoteDeliveryThreads**

Indicates the maximum number of threads dedicated to delivering emails to the Internet, by connecting to the recipient domain's SMTP server and using the SMTP protocol to transfer the message. This operation is very dependant on network delays, so you should set this number not too low to benefit from parallelism. Under light load DeskNow will automatically reduce the number of threads in use as necessary.

### **7.11.33 EmbeddedServer.remoteDeliveryRetryDelay**

When a message cannot be delivered it is placed in a retry queue. After the amount of time specified by this parameter (in seconds), a new delivery attempt is made.

### **7.11.34 EmbeddedServer.remoteDeliveryTimeout**

Number of seconds after which DeskNow waits for a reply from an SMTP server before closing the connection. This parameter is important to avoid that a broken SMTP server locks up the delivery threads of DeskNow indefinitely.

### **7.11.35 EmbeddedServer.remoteDeliveryRetryMaxCycles**

Number of attempts that DeskNow makes to deliver a message. If after all the attempts the message was still undeliverable, DeskNow will send back an error message to the sender.

### **7.11.36 EmbeddedServer.SMTPThreads**

Indicates the maximum number of threads dedicated to receive emails through SMTP connections.

This number should be proportioned to the expected number of concurrent SMTP connection attempts received by DeskNow.

### **7.11.37 EmbeddedServer.rootPath**

DeskNow uses this folder as a workspace for all the emails processed by the mail server. In particular, POP3 inboxes are kept under the /pop3 folder under the root.

### **7.11.38 EmbeddedServer.usePOP3**

Enables / disables the POP3 service. Unless you really need to keep using traditional email clients, we suggest disabling POP3. In this way users will benefit of a more integrated work environment (mail, files, calendar, collaboration) without the duplication of an external email client. The web mail provided by DeskNow has all the features (and sometimes more) of a traditional mail client, and is not subject to mail viruses, etc.

### **7.11.39 EmbeddedServer.POP3Threads**

Maximum number of threads dedicated to serve concurrent POP3 connections.

### **7.11.40 EmbeddedServer.authorizeRelay**

By default, the DeskNow mail server blocks any attempt of mail relay (i.e. the SMTP server does not accept mail directed to the internet from IP addresses different from 127.0.0.1). This is very important to prevent abuse by mail spammers.

If you want mail clients or other applications to be able to send mails to the internet through the DeskNow SMTP server, you can do two things (**you can do one of them, or both**):

1) explicitly authorize them by adding their IP address or subnet to the configuration file. You can add multiple IPs or subnets.

*Example:*

```
EmbeddedServer.authorizeRelay.1    65.14.13.122
EmbeddedServer.authorizeRelay.2    65.14.13.123
EmbeddedServer.authorizeRelay.3    65.14.14.
```

*(Note the final dot in the third line)*

*This will enable computers with IP 65.14.13.122, 65.14.13.123 and any computer in the subnet 65.14.14.\**

2) enable SMTP authentication: see the following section 7.11.41

### 7.11.41 EmbeddedServer. acceptSMTPAuthentication

If this flag is set to TRUE, DeskNow will accept SMTP authentication to authorize mail relay even from IP addresses not specifically authorized (see 7.11.40).

In other words, if this flag is TRUE users from anywhere will be able to send emails to any address on the internet, by setting their own DeskNow username and password in the email client connection properties. For example, in Microsoft Outlook these settings are under the "Outgoing Mail Server: My server requires authentication" section of the mail account.

User of virtual domains (i.e. domains other than the default) must authenticate using username—domainname (2 dash characters) as their SMTP username (ex.: joe—company1.com)

### 7.11.42 EmbeddedServer. verifySMTPIdentity

If this flag is set to TRUE *and* EmbeddedServer.acceptSMTPAuthentication is set to TRUE, DeskNow will check the identity of the sender before allowing him/her to send emails.

In other words, if a user authenticates as 'joe' and then tries to send an email where the From field is 'mark@xyz.com', then DeskNow will reject the message and will write a warning message in the security log file.

### 7.11.43 EmbeddedServer.localDomain

***NB do not confuse this with virtual domains. This is an option to create 'domain aliases' for the default domain. You can create completely separated mail domains by using the Virtual Domains feature of DeskNow.***

Specifies "domain aliases" so that when DeskNow receives an email for a domain listed in this section, it will deliver the message to the default domain.

*Example:*

```
EmbeddedServer.localDomain.1      myoldcompanyname.com
```

*This will instruct DeskNow to deliver all the emails addressed to <somename>@myoldcompanyname.com to the proper local inbox in the default domain.*



#### 7.11.44 EmbeddedServer.SMTPForwardHost

This setting instructs DeskNow to forward all the outgoing emails (i.e. those that are not considered to be local) to another SMTP server, instead of trying to deliver them directly to the recipients' mail servers. Specify the address of the external SMTP server to use. This setting is disabled by default. Also note that DeskNow will still deliver to the local inboxes messages that it considers 'local', i.e. directed to one of its users.

#### 7.11.45 EmbeddedServer.SMTPForwardPort

This setting specifies the port used to connect to the forward SMTP server. It is ignored if EmbeddedServer.SMTPForwardHost is not used. The default is 25.

#### 7.11.46 EmbeddedServer.SMTPForwardUsername

This setting specifies the username used to connect to the forward SMTP server, if it requires SMTP authentication. It is ignored if EmbeddedServer.SMTPForwardHost is not used. The default is to not use SMTP authentication to send emails to the forward server. If you specify this setting, you need to specify the password as well (see below).

#### 7.11.47 EmbeddedServer.SMTPForwardPassword

This setting specifies the password used to connect to the forward SMTP server, if it requires SMTP authentication. It is ignored if EmbeddedServer.SMTPForwardHost is not used, or if EmbeddedServer.SMTPForwardUsername is not used.

#### 7.11.48 EmbeddedServer.SynchronizeWebmailWithPOP3Read

If this flag is TRUE and the DeskNow POP3 server is enabled, DeskNow will mark messages as read in the WebMail interface whenever they are downloaded by a POP3 client. This helps synchronizing the status of messages, so that users know which messages they have already read.

***NB there is no way to know when a message has been actually READ by the user in a POP3 client. If the flag is set to TRUE DeskNow will mark the message as read when the POP3 client DOWNLOADS the message.***

Possible values are TRUE (enable) or FALSE (disable).

The default value for this flag, for new installations created using release 1.3 or above, is TRUE.

#### 7.11.49 EmbeddedServer.SynchronizeWebmailWithPOP3Delete

If this flag is TRUE and the DeskNow POP3 server is enabled, DeskNow will delete messages from the WebMail interface whenever they are deleted from the inbox by a POP3 client. This helps synchronizing the status of messages, especially in situations when POP3 is the main access method, but users need access to webmail when they are out of the office.

Possible values are TRUE (enable) or FALSE (disable).

The default value for this flag is FALSE.

#### 7.11.50 EmbeddedServer.DetectTimeLimitedEmailAddresses

If this flag is TRUE and the DeskNow email server is enabled, DeskNow will detect and filter out time limited email addresses, to prevent spam. See the DeskNow online help for a complete description of time-limited email addresses.

Possible values are TRUE (enable) or FALSE (disable).

The default value for this flag is TRUE.

### 7.11.51 EmbeddedServer.DNSServer

DeskNow usually automatically detects the address of the DNS servers available. In some rare circumstances, however, it does not detect the correct addresses, or the DNS servers detected are just DNS proxies, which do not give full support of mail record (MX) lookups.

You can use this property to manually set the DNS servers to use.

*Example: EmbeddedServer.DNSServer.1      192.168.1.12*

You can indicate multiple DNS servers, for redundancy:

*EmbeddedServer.DNSServer.2 192.168.1.13  
Ets  
...etc*

### 7.11.52 OutputCharset

The default charset to use when encoding mail before sending them via the web interface. Ideally you would want to use UTF-8, which can encode every character of every known language. Unfortunately some old mail readers don't understand UTF-8.

The following charsets can be used with DeskNow:

Big5  
Big5-HKSCS  
EUC-CN  
EUC-JP  
euc-jp-linux  
EUC-KR  
EUC-TW  
GB18030  
GBK  
ISCII91  
ISO-2022-CN-CNS  
ISO-2022-CN-GB  
ISO-2022-KR  
ISO-8859-1  
ISO-8859-13  
ISO-8859-15  
ISO-8859-2  
ISO-8859-3  
ISO-8859-4  
ISO-8859-5  
ISO-8859-6  
ISO-8859-7  
ISO-8859-8  
ISO-8859-9  
JIS0201  
JIS0208  
JIS0212

Johab  
KOI8-R  
Shift\_JIS  
TIS-620  
US-ASCII  
UTF-16  
UTF-16BE  
UTF-16LE  
UTF-8

DeskNow normally attempts to detect the correct charset from your system, but you can override the charset by setting one directly.

## 7.12 Messenger.cfg

Defines properties for the integrated instant messaging (Jabber/XMPP) server.

### 7.12.1 use

Enable or disable the integrated instant messaging server.

TRUE = enable

FALSE = disable

### 7.12.2 PlainConnector.use

Enables or disable plain (non encrypted) XMPP connections.

TRUE = enable

FALSE = disable

### 7.12.3 PlainConnector.port

Defines which port the server should listen to for plain XMPP connections. The IANA registered port number for XMPP is 5222, so it is recommended to use this.

*Example:*

*PlainConnector.port* 5222

### 7.12.4 PlainConnector.NIO

Enable / disable the use of Java NIO (Native Input / Output). NIO is recommended for medium-to large sites (> 100 connections) because it can scale well up to thousands of connections without excessive resource requirements.

Support of Java NIO could still be experimental on some Java versions, or on some OS.

TRUE = enable

FALSE = disable

By default, NIO is disabled.

### 7.12.5 SSLConnector.use

Enables or disable SSL encrypted XMPP connections.

TRUE = enable

FALSE = disable

### 7.12.6 SSLConnector.port

Defines which port the server should listen to for SSL XMPP connections. The IANA registered port number for XMPP is 5223, so it is recommended to use this.

*Example:*

*SSLConnector.port* 5223

### 7.12.7 WebConnector.use

Enable or disable connections through HTTP or HTTPS. These are useful if the client is behind a firewall, and direct connections on the standard ports are not possible.

HTTP/HTTPS connections can pass through normal proxies, but are slower than direct connections. The DeskNow messenger client tries to connect using direct connections first. If it fails, it tries using web connections.

TRUE = enable  
FALSE = disable

### 7.12.8 RosterMode

Specifies what contacts should be automatically made available to a user (i.e. what contacts should be shown to him in the “roster”). This setting can have two possible values:

- **domain** to show all the users of the domain. Users are grouped by community.
- **community** to show only users of the same community

The default value is “domain”.

*Example:*  
*RosterMode community*

## 7.13 Web.cfg

Defines properties for the HTML-based interface.

### 7.13.1 Layouts section

Defines the available layouts, and the default layout to be used

Layouts are defined via a set of JSP files.

Each layout resides in its own subfolder under the '/jsp' folder.

For instance, the files making the layout 'DeskNow Original' can be stored in /jsp/original

*Example*

*Layout.1.Name "DeskNow original"*

*Layout.1.Id original*

*Layout.2.Name "DeskNow noframes"*

*Layout.2.Id noframes*

*Layout.Default original*

### 7.13.2 Custom section

This section contains custom settings that are used by web designers in specific layouts.

As default, the only setting in this section defines the default color theme for the "original" layout.

*Example*

*Custom.original.DefaultTheme orange*

### 7.13.3 Custom.BaseURL

Used in advanced setups (such as SSL, Web server+servlet engine, etc).

Forces the sevlet engine to "include" pages inside other pages by connecting to itself using this url. By default, it uses the same URL that the client is using.

*Example:*

*Custom.BaseURL <http://localhost/desknw/>*

### 7.13.4 Custom.original.DefaultLongDatePattern

Specifies the default format for dates (when using the long format) to be used for new users.

This by default produces dates like "Friday, March 7 2003" (American convention).

You can set a different default date format. The syntax follows the convention of the Java class java.text.SimpleDateFormat (see <http://java.sun.com/j2se/1.4.1/docs/api> ).

A typical default value, if you prefer the European convention with the day before the month, is the following:

*Custom.original.DefaultLongDatePattern "EEEE, d MMMM yyyy"*

NB. users can change this setting in their Preferences.

If you choose a non-standard date format, you should also check that it is

### 7.13.5 Custom.original.DefaultShortDatePattern

Specifies the default format for dates (when using the short format) to be used for new users.

This by default produces dates like "03/07/2003" for the 7th of March 2003 (American convention).

You can set a different default date format. The syntax follows the convention of the Java class `java.text.SimpleDateFormat` (see <http://java.sun.com/j2se/1.4.1/docs/api> ).

A typical default value, if you prefer the European convention with the day before the month, is the following:

*Custom.original.DefaultLongDatePattern*      "dd/MM/yy"

NB. users can change this setting in their Preferences.

## 7.14 International.cfg

This file contains all the necessary setting for the access of DeskNow in multiple languages.

See section 8 for a full overview of language support in DeskNow.

### 7.14.1 DefaultLanguage

Defines the default language assigned to new user accounts when they login. Users can choose a different language from the Preferences page.

The value of this setting must be the language code of one of the languages defined. See section 8 for more details.

## 7.15 HTTP port

If you are using Tomcat as servlet engine (the default for simple installations), you can change the HTTP port used by DeskNow at any time.

In order to do so, edit the file \$TOMCAT/conf/server.xml

***Important: if you are using the self-installing DeskNow for Windows, you can set the HTTP and HTTPS port by running the Configuration Wizard, located in the DeskNow program group in the Start menu.***



# 8 Language and timezone

DeskNow is a multi-language product, which means that the web interface can be accessed in many languages. This section explains how to configure DeskNow for multi-language access, and how to translate DeskNow to your language, if a translation is not already available.

## 8.1 Default language and user language

The languages support of DeskNow is configured in the configuration file `International.cfg` (see section 7.14).

For every language, two lines must be present in the file, defining the language code, name and the location of the translation file.

```
Example:  
Language.it.name      Italian  
Language.it.file     $(LANGUAGE_BASE)/desknow_it.txt
```

In the above example, a new language is defined, where:

- “it” is the language code
- “Italian” is the language name
- the file `desknow_it.txt`, in the specified folder, is the translation file

The translation file is a text file where there is one key translation per line.

```
Example:  
New contact = Nuovo contatto
```

You can configure the default language for DeskNow. This is the language that will be used by default for new users. The setting is described in section 7.14.1.

Users can choose to use a different language by changing the appropriate setting in the Preferences page. Every user can choose his/her own language.

Note: you may want to check the emails sent are encoded using the appropriate charset. See section 7.11.52 for more information about setting the server’s charset encoding for outgoing email.

## 8.2 Adding a language to DeskNow

If your language is not available in the list of available languages (because it is not configured), visit the support page of DeskNow on the web to look for additional languages (see section 11).

If still no one has produced a translation file for your language, you can create one yourself, and share it with the community of DeskNow users. Or collaborate with other users that are looking for such translation, and share the work. Our public forums have a forum dedicated to the exchange of information regarding language translations for DeskNow.

To create a language for DeskNow, look for the ISO 639 language code of your language. This information can be found on the web, ex. at <http://www.ics.uci.edu/pub/ietf/http/related/iso639.txt>

It is important to use an official language code, because DeskNow will use the code to translate dates accordingly.

Then copy the file `desknow_TEMPLATE.txt` (located in Program Files/DeskNow/webapps/desknow/WEB-INF/international) to the language file that you want to create (ex. `desknow_fr.txt` for French).

Then add the configuration for your language to `International.cfg`.

*Example:*

```
Language.fr.name      French
Language.fr.file     $(LANGUAGE_BASE)/desknow_fr.txt
```

Then open the file using Windows Notepad or any other **UTF-8** editor. It is mandatory to save the file using the UTF-8 encoding so that DeskNow can read any type of character you type, especially non-latin characters. The UTF-8 encoding ensures that DeskNow can display correctly virtually every character of every known language (NB your browser / OS might need to be configured with additional fonts for some languages).

Windows Notepad is the most common editor that can be used for UTF-8, so if you are using Windows we recommend it for editing DeskNow language files.

At this point, all you have to do is translating all the lines provided in the file in your language.

## 8.2.1 Translation tips

Try to be consistent with the terms you choose. For instance, refer to 'Message Boards' using always the same term in your language.

If not sure where a particular term (ex. "xyz" is used, use UltraEdit or the Search function of Windows to look for a file in `/Program Files/DeskNow/webapps/desknow/jsp/original` that contains that text. Some terms are system messages, and are defined in DeskNow itself.

The `{0}`, `{1}`, etc words are placeholders for words that DeskNow will fill in. Do not translate them, only the statement in which they are contained.

*Example:*

```
Invalid parameter: {0}= Parametro non valido: {0}
```

Some terms begin with the \$ character. They are used to translate the names of common objects (ex the email 'in' folder, the 'sent' folder, the 'Announcements' message board, and so on). Translate them as shown in the following example:

```
$mailfolder$drafts= bozze
("bozze" is the Italian translation of "drafts")
```

## 8.2.2 Testing your progress

You can use the DeskNow HTTP API to reload the translation file at any time, without having to restart Desknow. This is useful to test your translation.

In short, just open the following URL in a browser:

[http://www.yourdesknowserver.com/desknow/admin?pwd=password&action=main\\_i18nreload](http://www.yourdesknowserver.com/desknow/admin?pwd=password&action=main_i18nreload)

where “password” is the password of the admin user.

NB this will pick up changes that occurred in the translation files, but not in International.cfg . If you modify International.cfg you need to restart DeskNow.

### 8.2.3 Share your work

We encourage you to share your work with other DeskNow users. Use the DeskNow public forums on the web to communicate your intention of working on a translation, and submit translations so that they can be used by others, and can be included with the main DeskNow distribution. We usually offer discount vouchers for DeskNow licenses for translations in languages that we don't have. Please contact us for more details.

### 8.2.4 What happens when a new version of DeskNow is released

If you have submitted your translation to us, when a new release of DeskNow is ready to be released we will generate an updated translation file that includes your translation and eventual new terms used in the new functionalities. In this way, your translation will not be lost, and only the new terms will need to be translated. This is usually a very easy task, often consisting of no more than 20 lines of new text.

## 8.3 Time zone

DeskNow can use any default time zone, potentially different from the time zone of the computer on which it is running. The default time zone is used for the timestamps in log files, and for new user accounts.

The default time zone can be set using the Configuration Wizard (in graphic mode), or by manually changing the ServerTimeZone parameter (see section 7.2.1).

Every user can choose a different time zone for his / her account. All the dates and times in DeskNow are adjusted according to the time zone of the user viewing the information. This means that an appointment in a shared calendar will appear at different times to two different users that work with different time zones. This is essential to organize conference calls, etc.

# 9 Administration API

The administration API, except for the Internationalization Refresh command, is available for the default DeskNow Lite license only for a trial period of time (30 days from the date of installation). After this period, it will not be available. See chapter 10 for information on how to purchase a DeskNow license.

## 9.1 Overview

DeskNow offers an API (Application Programming Interface) that allows to automate many system operations via an external program/script. For instance, it is possible to automatically create users, verify their username/password, send instant messages, post announcements, etc.

The API is completely based on HTTP, so that it can be used by virtually any scripting or programming language.

An API invocation is simply an HTTP request (GET or POST), like this:  
[http://www.mydomain.com/desknw/admin?pwd=password&action=im\\_sendalert&user=joe&domain=mydomain.com&message=emergency%20evacuation&alertcode=1](http://www.mydomain.com/desknw/admin?pwd=password&action=im_sendalert&user=joe&domain=mydomain.com&message=emergency%20evacuation&alertcode=1)

This invocation sends an instant messaging alert to user [joe@mydomain.com](mailto:joe@mydomain.com). You can test API invocations by typing them directly into a web browser, or creating simple web forms to input the various fields.

**URL Encoding:** when using the GET method (i.e. when all the request parameters are in the URL), the space character needs to be expressed as %20, as per HTTP protocol specifications.

The **pwd** parameter must always be provided, and the value must be the password of the admin user (of the default domain).

All the parameter names are case-sensitive.

It is also possible to use the SSL (https) protocol for the API invocations, for increased security.

### 9.1.1 Invocation result

If the operation is successful, DeskNow will return an HTTP 200 response status, and will also return a more descriptive text in the response body.

If the operation is unsuccessful, the HTTP response status will be different from 200, and a descriptive text will be included. Note that by default Internet Explorer is configured to 'Show friendly HTTP error messages' which will hide the response text in case of an error. You can disable that setting in the Tools/Options/Advanced panel, or use a different browser.

## 9.2 Create user

Parameter name	Description	Required
----------------	-------------	----------

pwd	Password of the admin user	Yes
action	user_createuser	Yes
username		Yes
password		Yes
domain		No. Use default domain if omitted.
communityname	Name of the community for the new user. Must exist.	Yes
diskquota	"Soft" disk quota, in bytes	No. Unlimited disk quota if omitted.
usertype	"Normal", "Restricted" or "Guest"	No. Assume Normal if omitted.
firstname		No
middlename		No
lastname		No

## 9.3 Create community

Parameter name	Description	Required
pwd	Password of the admin user	Yes
action	user_createcommunity	Yes
managerusername		Yes
communityname		Yes
domain		No. Use default domain if omitted.
diskquota	"Soft" disk quota, in bytes	No. Unlimited disk quota if omitted.

## 9.4 Create domain

Parameter name	Description	Required
pwd	Password of the admin user	Yes
action	user_createdomain	Yes
domain		Yes
adminpassword	Password of the admin user for the new domain	Yes
diskquota	"Soft" disk quota, in bytes	No. Unlimited disk quota if omitted.

**Note:** this method will create automatically the admin user for the new domain.

## 9.5 Authenticate user

Parameter name	Description	Required
pwd	Password of the admin user	Yes
action	user_authenticate	Yes
username	Username of the user to be authenticated	Yes
password	Password of the user to be	Yes

	authenticated	
domain	Domain of the user to be authenticated	No. Use default domain if omitted.

## 9.6 Post to message board

Parameter name	Description	Required
pwd	Password of the admin user	Yes
action	mboard_post	Yes
username	Username under whose credentials the message will be posted. The user must have write permission to the messageboard.	Yes
domain	Domain of the posting user	No. Use default domain if omitted.
messageboardname	Name of the messageboard, as seen from the user. If this is a "system" messageboard, the name must be the English version.	Yes
subject	Subject of the message. Use URL encoding where necessary.	Yes
body	Body of the message. Use URL encoding where necessary.	Yes

## 9.7 Send IM alert

Parameter name	Description	Required
pwd	Password of the admin user	Yes
action	im_sendalert	Yes
username	Username of the recipient user	Yes
domain	Domain of the recipient user	No. Use default domain if omitted.
alertcode	The alertcode can be passed to the JavaScript code in the applet page, to trigger events on the browser.	No
message	Body of the message. Use URL encoding where necessary.	Yes

**Note:** the alert will not be delivered if the user is not online.

## 9.8 Send IM broadcast

Parameter name	Description	Required
pwd	Password of the admin user	Yes
action	im_sendbroadcast	Yes
domain	Recipient domain. Omit to	No. Broadcast to all domains

	broadcast to all domains.	if omitted.
message	Body of the message. Use URL encoding where necessary.	Yes

**Note:** the alert will be delivered only to users connected to the IM service.

# 10 Licensing DeskNow

DeskNow is provided by default with the **DeskNow Lite** license.

This license is **FREE**, does not expire and can be used by an unlimited number of users.

Some of its advanced features however, are only available for a trial period of 30 days after its installation. At the end of this period, you can either choose to continue using DeskNow without these features, or to purchase a commercial license.

To purchase a commercial license or to see a full feature comparison list, please visit <http://www.desknow.com/buy.html> .



# 11 Support

DeskNow provides both open-style support using public discussion forums, and fast, premium support for customers that purchased a commercial license or Service Agreement.

Please visit <http://www.desknow.com/support.html> to access DeskNow Support.