PC-Pages User Guide

© **Computing and Media Services, Brunel University**

Version 1.1. April 28, 1994.

Contents

1. Introduction 4

1.1. About This Document....4

2. Installation and Setup 5

2.1. Requirements....5

2.2. Installation....5

2.3. Starting PC-Pages....6

3. Looking Up Directory Entries 7

3.1. Lookup by Form Filling....7

3.2. Lookup by Directory Name....9

4. Data Display Windows 10

4.1. Entry Display Window....10

4.2. Entry List Window....11

4.3. Entry Collection Window....12

5. Data Editing Windows 13

5.1. Entry Edit Window....13

5.1. Postal Address Editor....13

5.2. Fax Editor....14

5.3. Text Attribute Editor....14

Appendix 1. Configuring With RBW.INI 16

Appendix 2. Configuring With DUA.INI 18

Appendix 3: Attributes and Syntaxes. 25

Appendix 4. Object Classes 30

Appendix 5. Glossary of Terms 35

Index 38

1. Introduction

PC-Pages can be thought of as the electronic equivalent of a phone book. It is better than a paper phone book in a number of ways.

- It performs all lookups for you.
- The information contained is much richer.
- Many more people can be found.
- Information stored is usually more up-to-date.

PC-Pages can be a powerful aid to personal communication. It provides the ability to determine contact points such as phone numbers or electronic mail addresses quickly and efficiently. This helps to free effort required for the often laborious task of looking up information in paper directories.

Data retrieved by PC-Pages can be passed into other applications, generally electronic mail or address book programs.

1.1. About This Document

This document is aimed at two groups of people: those who install and configure the software and those who use it on a day to day basis.

The first chapter provides step-by-step instructions on installation and start up. Other important details such as hardware and software requirements are listed.

Chapters 2 through to 5 are intended for those who wish to learn about general usage. They attempt to guide users by providing simple instructions and back this up by demonstrating a few basic examples. Appendix 5 is a glossary of some of the terminology used in this document.

Appendices 1 to 4 assume some knowledge of the underlying directory and network system. These sections are concerned with tailoring and setup, and as such should only be read by systems administrators and the like.

2. Installation and Setup

This chapter describes the PC-Pages installation procedure. The basic setup options necessary for an initial local configuration are also given.

2.1. Requirements

To run PC-Pages you will require an IBM AT compatible with an 80386 or greater with at least 2Mb of memory. In addition you'll need the following software:

- Microsoft Windows 3.0 or greater, running in Standard or Enhanced mode.
- WhiteStack version 1.104 or greater.

WhiteStack will not be required on the user's machine if you are using an alternative stack (e.g. NetBIOS and a WhiteGate gateway).

2.2. Installation

This section describes the procedure you must follow to install PC-Pages. Before attempting an installation note that:

- You must have write permission for your RBW.INI (if one is present and can be found).
- You must have write permission to the chosen installation directory.

Step by step instructions follow:

1. PC-Pages is usually provided as a uuencoded or boo encoded ZIP archive file. Copy the release file into an empty temporary directory and unpack it. This directory will be referred to as the 'source' directory.

   The install program assumes that Windows and WhiteStack are already installed on your system.

2. Type SETUP from the source directory. Setup will attempt to start up Windows and will then undergo an initialization process. This may take a few moments.

3. You will be asked to enter the directory in which you want to install PC-Pages. PC-Pages will attempt create this directory if not already present.

   If you wish PC-Pages to run in conjunction with a mail application (e.g. WhiteMail) you will have to install PC-Pages into a directory that is present in the DOS PATH environment variable. This is also true of the mail application you're using.

4. If you don't have an old PC-Pages installation on your system you will be prompted for the type of transport stack installed with WhiteStack. If you have an old installation you should skip this step by clicking on OK.

The choices offered are TCP/IP (if you're using PC-NFS, Beame & Whiteside NFS, etc.) or CONS (if you're using Pinkbook).

5. You will now be asked to enter the presentation address of your local directory server. If you already have a server configured into RBW.INI this will be displayed. If not you will be shown an address template based on your selection in step 4.

   An example TCP/IP address is:

   ```
   TCPIP+134.83.128.41+PORTNUM+17003
   ```

   An example CONS address is:

   ```
   CONS+38826110000411000012824800+SNPA+00800200f19ae+LSAP+7e
   ```

6. Next you will be asked if you have a WhiteMail installation present on you computer. If you select 'YES', the setup program will add the relevant options to RBW.INI.

7. Finally all files will be copied to the installation directory and updates made to RBW.INI.

2.3. Starting PC-Pages

Before PC-Pages is started, the WhiteStack support application must be running. If it is running you should be able to see its icon at the bottom of your screen. If not, start it up by double clicking on the WhiteStack icon in Program Manager (if you are running PC-Pages over an alternative stack then this won't be necessary).

Once this is done double click on the PC-Pages icon in Program Manager to start your session. You will now have to go through the following steps before starting the session proper:

1. A start up dialogue will appear first giving you some brief guidelines on how to use PC-Pages. Click on "OK" to bypass this. Clicking on "Help" will give you an introductory help page on PC-Pages.

2. "Login" dialogue. Enter your full directory distinguished name (DN) and password. If you wish to login anonymously, just click on OK.

   Your DN and password should be available from your local network administrator.

PC-Pages will now attempt to connect to the local directory service.

3. Looking Up Directory Entries

The directory database consists of a number of *directory entries*. Each entry corresponds to some real world object, e.g. a person or an organization, and contains information on that object. A person's entry may contain various phone numbers, a fax number, e-mail address, etc. Directory entries can be searched for in either of two ways: using the form based search dialogue or by entering an approximate directory name. This chapter describes two methods of looking up directory entries: using a form based dialogue and using informal directory names.

3.1. Lookup by Form Filling

This search method relies on the user supplying attributes associated with the *target entry* (the entry being searched for). As an example, if searching for a person, you could supply the department that the person belongs to.

To search for an entry select an entry type from the Search menu, e.g. Person or Department.

Once an entry type has been selected a form dialogue will appear. An example dialogue for "Person" type searches is shown in Figure 1. The uppermost field is referred to as the target field (again, the target is the entry being searched for). The remaining fields are referred to as the context fields. These are used to help PC-Pages locate the target entry.

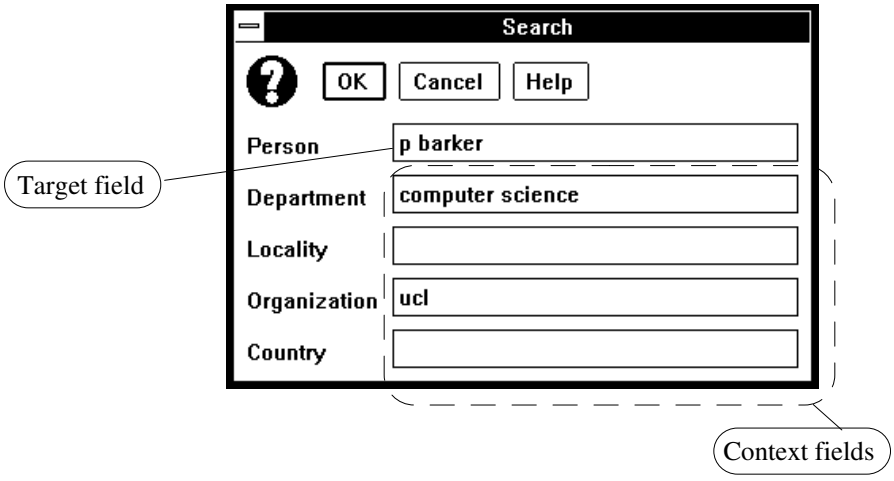

Figure 1. Person Search Dialogue

To start a search fill out as many of the fields as you can. You must always fill in the target field (the uppermost) as this is the name of the entry you're searching for (e.g. a person's name when searching for a person).

It isn't always necessary to fill in all or even any of the context fields. If you are searching for someone within your own organization it shouldn't be necessary to enter the name of your organization and your country. Similarly, when searching for an

organization in your own country it shouldn't be necessary for you to type in your country's name. This default behaviour is dependent on your local configuration (see Appendix 2 for instructions on how to do this).

The following sections give a few sample exercises for you to perform whilst getting used to PC-Pages.

Example 1.

As a first example you could try searching for yourself:

1. Select "Person" from the "Search" menu.
2. Enter your own name into the field labelled "Person". The preferred format here is initial and then surname, although this may vary according to local data formats.
3. Click on OK or press return to start the search.

PC-Pages should now be attempting to search within your local organization. If you are found in the directory database your entry will be displayed. If more than one entry matches the name supplied (i.e. if other people have a name similar to your own) a list of matching entries will be displayed.

Notice that you didn't have to type anything into the remaining fields in order to find your own entry.

During the course of this search you may have seen a few new windows. These are described in detail in later chapters.

Example 2.

This example deals with people in other organizations in your country. This can be demonstrated by searching for your own entry again, but this time in a slightly different manner.

1. Select "Person" from the "Search" menu.
2. Type your name into the "Person" field.
3. Type the name of your organization into the "Organization" field.
4. Click on OK or press return.

PC-Pages will begin by searching for your organization. Once this has been found it will attempt to search for your entry. You should get the same results that you got by trying example 1.

You can now go further by trying to search for people in other organizations in your own country. To search within organizations in other countries try filling in the "Country" field as well.

Example 3.

This example describes how to search for organization entries. Try looking up your own organization first:

1. Select "Organization" from the "Search" menu.
2. Enter the name of your organization into the uppermost field.
3. Click on OK or press return.

3.2. Lookup by Directory Name

A directory name a single line of text used to locate a directory entry. It consists of a series of comma separated values, for example:

```
Fred Smith, English Language, Muddle College, GB.
```

The directory name for a 'Person' entry usually consists of a sequence of the person's name, their department name, organization name and country.

PC-Pages can use approximate directory names to find entries. The approximate directory name below would probably find the entry named above.

```
F Smith, Muddle, Great Britain
```

It is possible to search on a substring basis (e.g. "F Smith" from "Fred Smith") and by leaving out name components (in the above example the department name was omitted).

Assumptions are made about the location of data being searched for. If only a person's name is supplied then PC-Pages will 'guess' that the person being searched for belongs to your organization, and will form a search request accordingly. Note though that this default behaviour is dependent on local configuration.

To search for an entry by it's name select "By name" from the "Search" menu.

The following examples should help you get used to searching by directory name.

Example 1

Follow the instructions below to find your own entry:

1. Select "By name" from the "Search" menu.
2. Enter your name (initial followed by surname) into the "Name" field.
3. Click on OK or press enter to start the search.

PC-Pages will now attempt to search for the supplied name in the default organization (usually your own). If the search is successful your entry will be read and displayed. If more than one matching entry is found a list of directory names will be shown instead.

Try searching for yourself again, but this time with a fuller version of your directory name (adding your organization name for example). Something like;

```
A Fellow, South Central University
```

or

```
A Fellow, Research Division, South Central, USA.
```

Supplying this additional information enables you to search other organizations and countries.

4. Data Display Windows

This chapter describes each of the data windows found in PC-Pages. A data window contains information retrieved from the directory database. This is typically in the form of a list of directory names or the contents of an entry.

4.1. Entry Display Window

This window displays a formatted directory entry. Various *attributes* of the entry are shown in a set of text boxes. The attributes shown will depend on the type of entry being viewed, for example a "Person" entry may have a "Job title" attribute whereas an "Organization" entry won't.

In addition two rows of buttons are present. The top row of buttons allow the entry (or related entries) to be manipulated in some way. The bottom row offers access to other entry attributes, e.g. a photo image.

A typical entry window is shown in Figure **Error! Bookmark not defined.**. More important areas are highlighted and explained.

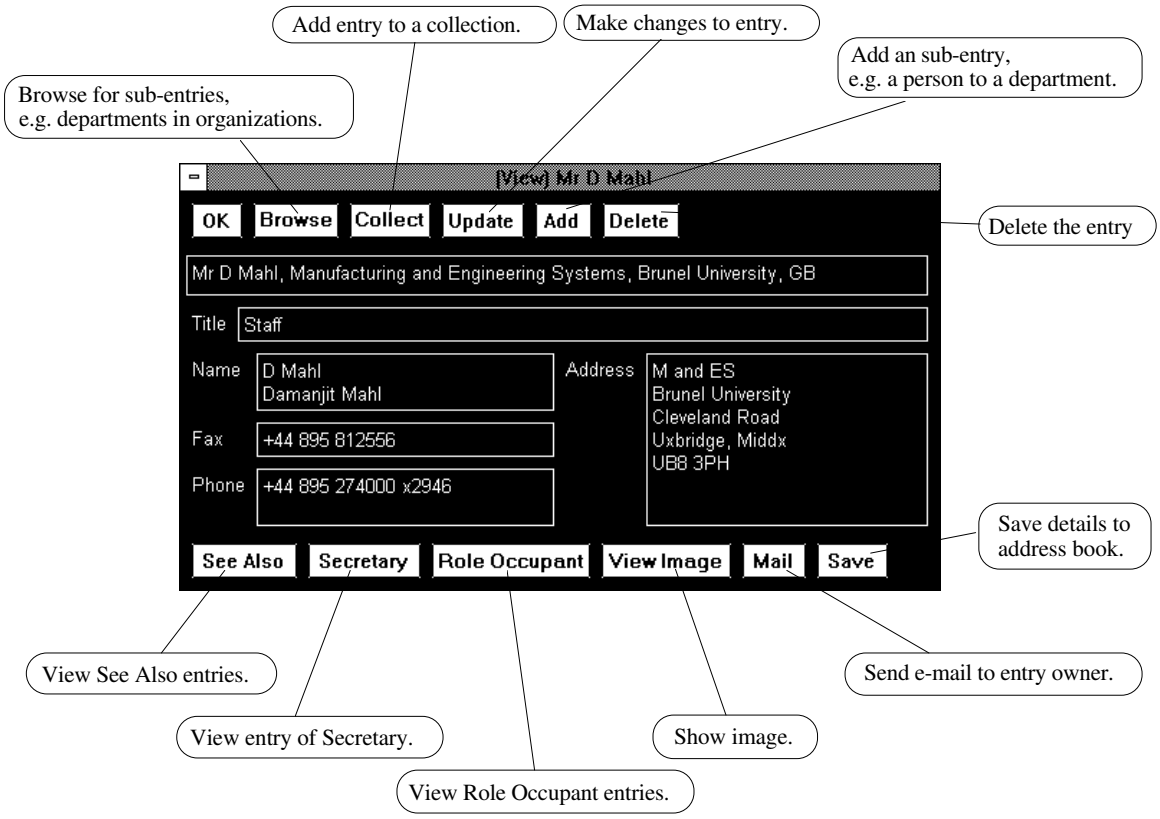

Figure 2. Entry Display Window

The buttons labelled in Figure **Error! Bookmark not defined.** are described in more detail in the following sections.

See also

A See also entry is an entry related to the one displayed. Clicking on this button causes the related entry to be retrieved and displayed.

Secretary

This button retrieves the entry for the secretary of the person whose entry is displayed.

Role occupant

Role occupant refers to the person who fulfils the role described in the displayed entry. A role might be "Head of Department", "Managing Director", etc.

View image

Some entries have graphical data associated with them. Clicking on the View image button executes a viewing application which displays the image.

Mail

When an entry has an associated electronic mail address, it is possible to mail to it directly by selecting this button. This executes an electronic mail package which will pop up ready to send to the specified address.

Save

This button causes personal entry details (nickname, e-mail address and telephone number) to a local address book database.

Browse

Many entries have sub-entries held below them, usually reflecting some real world relationship. So a departmental hierarchy could be represented by departmental entries being subordinate to an organizational entry. The Browse operation displays a selected list of entries below the one displayed.

Note that Browse won't always find all sub-entries, but often just a small selection. If this is the case then you will be notified with an error message.

Collect

Store the name of the displayed entry in a Collect window, if one is currently in existence.

Update

Raise an Edit Entry dialogue window. This allows you to modify part of the entry.

Add

Create and add a new entry below the one displayed (e.g. add a new person under a department).

Delete

Remove the displayed entry from the directory database.

4.2. Entry List Window

The List window is used to display a collection of directory entries. Entries listed can be displayed, browsed under or added to the Collect window. To perform one of these actions select an entry from the list box, then click on one of the Show, Browse or Collect buttons as appropriate.

4.3. Entry Collection Window

The Collect window is used to build and maintain a list of directory entries. A list of entries built up in this way can be used to print a personal address book. In future versions such collections will be used to build and manage mailing lists, and provide the ability to mail to multiple electronic mail addresses.

To print an entry collection, choose the Print option from the File menu.

5. Data Editing Windows

This chapter describes the windows used in entry modification and add operations.

5.1. Entry Edit Window

This dialogue window divides each entry into two sets of attributes: must attributes and may attributes. Attributes that the entry must contain, and those that it can contain on an optional basis.

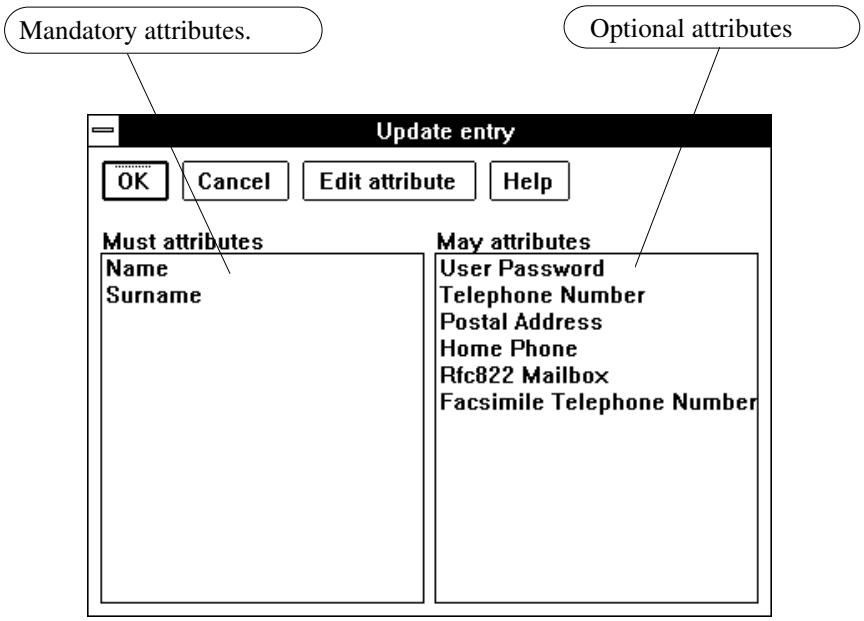

Figure 3. Edit Entry Dialogue Window

To edit an attribute:

1. Select the desired attribute from either list box.

2. Click on the Edit Attribute button at the top of the dialogue.

This will raise an editor appropriate to the type of attribute being edited. The various editor dialogues are described in the following sections.

Once all changes have been made click on OK to commit the update. Click on Cancel if you decide not to save about the changes you've made.

5.1. Postal Address Editor

Postal addresses are limited to six lines of text, with each line of text containing no more than 32 characters.

The editor dialogue for postal addresses consists of two areas:

1. An upper area containing two buttons: OK and Cancel.

2. A row of six edit fields representing the postal address.

5.2. Fax Editor

Fax numbers consist of a simple phone number (a line of text) together with a set of flags which describe the capabilities of your fax machine (these are usually ignored!).

The fax number edit dialogue has three parts:

1. A row of buttons: OK and Cancel.

2. An edit field for the fax number.

3. A group of check boxes which describe the capabilities of your fax.

5.3. Text Attribute Editor

Most attributes consist of a single line of text. In addition each attribute can have multiple values. The "Name" attribute of a person's directory entry may represent several names by which that person is known, e.g. "David Smith", "Mr D Smith" and "Dave Smith".

Figure 4 depicts a Text Attribute Editor Dialogue.

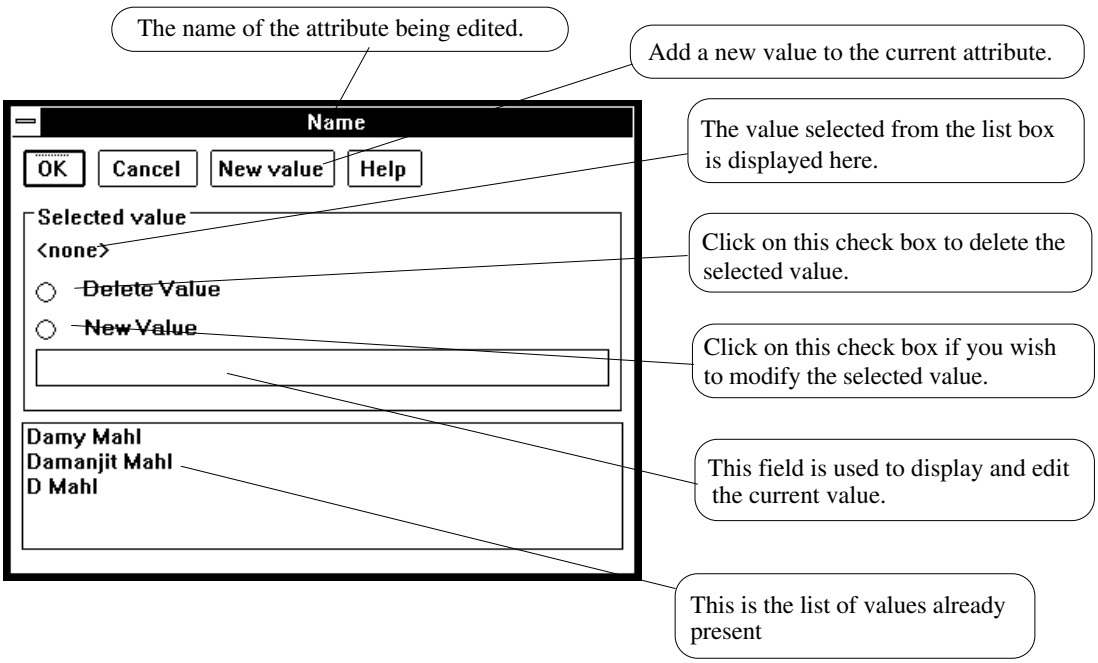

Figure 4. Text Attribute Editor Dialogue.

To update an already existing attribute value:

1. Select a value from the list box at the bottom of the edit dialogue.

2. Click on the Update/Add radio button or text field, then enter a new value in the text field. If you simply want to delete the selected value then click on the Delete radio button.

3. Click on OK to get back to the main Update dialogue. Click on Cancel if you want PC-Pages to ignore any changes you've made.

To add a new value:

1. Click on the New Value button at the top of the dialogue.
2. Follow the steps for editing values outlined above.

Appendix 1. Configuring With RBW.INI

The RBW.INI tailor file is (potentially) shared across a number of co-operating applications. PC-Pages uses it to store the following information:

- Network connection information.
- Type of network stack that PC-Pages should use.
- Details on co-operating mail user agent.
- Details on co-operating address book application.

The basic syntax of RBW.INI parameters is:

```
parameter = value
```

If the parameter value is a text string (e.g. a presentation address or a file name) then it must enclosed in single quotes (').

PC-Pages makes use of two sections in RBW.INI: `[DUA]` and `[PCPAGES]`.

Network Connection

Two pieces of information are required by PC-Pages. One is the presentation address of the directory server to contact (indicated by the '`called`' parameter), the second is an address used to identify the user's machine (the '`calling`' parameter).

The format of the address depends on the type of transport layer you're using.

These options should be placed in the `[DUA]` section of RBW.INI.

Network Protocol Stack

The '`stack`' parameter is used to indicate the network protocol stack in use. Table 1 shows the list of currently accepted stacks along with the appropriate parameter value.

Protocol Stack	Value of 'stack'
WhiteStack	1
NetBIOS via WhiteGate	2
WhiteLite	3

Table 1. Protocol Stack Specification

This parameter should be placed in the `[PCPAGES]` section of RBW.INI.

Mail User Agent

The following options tell PC-Pages about the mailer you are using. These are found in the `[PCPAGES]` section.

MUAFILENAME

The `'muafilename'` parameter must be the file name of the mail user agent executable. If the file is not in the DOS PATH environment variable, the full path name must be supplied.

The following line would work with the WhiteMail mail user agent:

```
muafilename = 'WHITEML.EXE'
```

MUAWINNAME

PC-Pages must know the name of your mail agent's main window in order to be able to communicate with it. The following line would work for WhiteMail version 1.102:

```
muawinname = "WHITEMAIL"
```

MUARFC822ONLY

Some mail user agents will only accept a electronic mail address of the RFC822 format. If this is the case then the `'rfc822only'` parameter must be set to 1.

MUACBONLY

If communication with your mail user agent is to take place via the Windows clipboard then the `'muacbonly'` option must be set to 1.

Address Book Agent

The following parameters define the address book agent used by PC-Pages. Note that the `'rfc822only'` and `'usecbonly'` options described above also apply to the address book.

These parameters are contained in the `[PCPAGES]` section.

ABFILENAME

The `'abfilename'` parameter must be the file name of the address book executable. If the file is not in the DOS PATH environment variable, the full path name must be supplied.

The following line would work with the WhiteMail user agent:

```
abfilename = 'WHITEML.EXE'
```

ABWINNAME

PC-Pages must know the name of your address book agent's main window in order to be able to communicate with it. The following line would work for WhiteMail version 1.102:

```
abwinname = "WHITEMAIL"
```

Appendix 2. Configuring With DUA.INI

The DUA.INI configuration file covers a number of configuration areas:

- Local context for searching. Searches are targeted towards parts of the directory database that are most likely to contain required information.
- Friendly naming for attributes. Each attribute can, for the purposes of add and modify, be given a user friendly name, e.g. in the default configuration the attribute 'ou' is given the label 'Department'.
- Entry presentation. The layout and content (i.e. which attributes are displayed) of entry windows is configurable.
- Entry add/modify attributes. The attribute presented to the user can be selected. This has the benefit of 'hiding' system attributes from non-technical users.
- General preferences. This covers display colours, service parameters, etc.

This appendix provides a section by section description of the options available. Note that in this section basic knowledge of the X.500 directory is assumed. In addition be aware that that PC-Pages will read in lines of up to 1024 characters in length. Anything longer will be concatenated!

The [BIND] Section

AUTHFILE

The 'authfile' option permits you to supply authentication information (X.500 distinguished name and password) in a file. This bypasses the need to type in particulars at the start of your session.

The syntax of the 'authfile' option is:

```
authfile : filename
```

The syntax of the authentication file is:

```
username : your_dn
password : your_password
```

Note that your username and password should be treated as strictly confidential information. It is only safe to use this option if the file name given refers to a secure network (or otherwise) file.

ANONYMOUS

If no authentication file has been defined with the 'authfile' option, perform initial DSA bind anonymously. This has the additional effect of preventing the authentication dialogue from appearing at startup.

To enable this option set the 'anonymous' flag to '1'.

The [SERVICE] Section

CONNECTTIME

This option defines how long PC-Pages should maintain a network connection if the application is left idle. The value supplied should be a number in minutes.

TIMELIMIT

The time limit of any single directory request (e.g. a read or search operation). The time limit is measured in seconds. If this value is left at 0 then requests will take the maximum possible time.

SIZELIMIT

This parameter defines the maximum number of matches returned by an individual search or browse operation. A value of 0 forces PC-Pages to request the maximum possible number of matches.

The [PREFERENCES] Section

STARTUPHELP

Display a startup dialogue window which gives some brief guidelines on how to use PC-Pages. Set parameter value to 1 if you want the dialogue box to appear.

MAILFORMAT

If required PC-Pages will display Internet mail addresses in Greybook format. To enable set the `'mailformat'` variable to 1.

JPEGCOMMAND

If you wish to view JPEG photo images found in the directory a suitable viewing application must be present on the user's system. Any Windows based JPEG viewer that accepts a file name as an argument should work.

The `'jpegcommand'` option must point to the viewer's executable file name. Syntax:

```
jpegcommand : filename
```

The value must be the full path of the file name if it is not contained in the DOS PATH variable.

The [ENTRYDB] Section

Alias names for X.500 DNs. These aliases also contain the `objectClass` of the given entry. Syntax:

```
entrydb-key : x500dn : objectclass
```

The `entrydb-key` can be any simple string (i.e. containing no spaces). For example:

```
organization1 : "c=GB@o=Brunel University" : 2.5.6.4
```

Note that the supplied distinguished name must be double quoted. The object class must be supplied as a *table name* or *object identifier* (see Appendix 3). The object class given in the above example is that for `'organizationalUnit.'`

The default `[ENTRYDB]` is:

```
country : "c=GB" : "country"
org : "c=GB@o=Brunel University" : "organizationalUnit"
```

The [UFNPATHS] Section

A *search path* is a sequence of directory entries used to direct a search towards a part of the directory database that is most likely to contain the data being looked for. For example, it may be assumed that someone performing a search is more likely to look for people within his own organization or department than elsewhere.

This section defines the search paths used by PC-Pages for name based directory lookups. The syntax of this option is:

```
path : entrydb-key-list : component-num
```

For example:

```
path : organization1, organization2 : 1
```

In this case if a single component directory name is supplied than PC-Pages will first search the entry `'organization1'` (specified in the `[ENTRYDB]` section), and then `'organization2'` in case the first search fails.

The [ATTRIBUTENAMES] Section

This section defines attribute type to name mappings used in the Edit Entry dialogue windows. The section consists of a list of mappings with the following syntax:

```
attribute : name
```

The 'attribute' can be defined by it's table name or object identifier (see Appendix 3). Any name value containing spaces must be double quoted.

The [SEARCHCLASSTYPES] Section

A great deal of PC-Pages ability to handle various types of entry (e.g. a person's entry as opposed to the entry for a department) correctly depends on a set of abstract *search class* definitions. A search class identifies a type of entry by looking at it's `objectClass`. The syntax for search class definitions is:

```
name : objectclasslist
```

The 'Person' search class is defined by:

```
Person : person, organizationalRole
```

This means that a 'Person' entry is defined to be any entry that has `person` or `organizationalRole` in it's `objectClass` attribute. Entry classes defined in this section are also referenced in sections dealing with entry modification (`[UPDATECLASSES]`), addition (`[ADDCLASSES]`) and display (`[SHOWTEMPLATES]`).

The [SEARCHCLASSFILTERS] Section

Form based search depends on a *search filter* being defined for a particular search class. A search filter consists of a list of attributes that are used to match on that entry. The syntax for this section is:

```
classname : attributelist
```

'Classname' refers to an entry search class defined in the `[SEARCHCLASSTYPES]` section. The 'attributelist' is a comma separated list of table names or object identifiers (these can be found Appendix 3). The filter for the 'Person' search class is specified as follows:

```
Person : cn
```

Thus when searching for 'Person' entries PC-Pages will filter on the `cn` (`commonName`) attribute.

The [SEARCHCLASSHIERARCHY] Section

In order to search for an entry PC-Pages must know about the various relationships between entries in the directory database. For example, by default, it is assumed that 'Person' entries are owned by 'Department' or 'Organization' entries. Such assumptions may not apply to local databases, and may require changing.

The syntax for establishing such a relationship is:

```
owned : objectclasslist
```

Where 'owned' is a search class name and 'objectclasslist' is a list of comma separated object classes. The default relationship for 'Person' entries is:

```
Person : organization, organizationalUnit
```

Note that the search class name of the 'owned' entry must be defined in the `[SEARCHCLASSTYPES]` section.

The [SEARCHCLASSCONTEXTS] Section

When performing form based searches PC-Pages must know which search classes you wish to use in order to generate the form dialogue. The syntax of this section is:

```
name : contextlist
```

Where 'name' is the search class name of the entry being searched for (the target entry), and 'contextlist' is a list of comma separated search class names which may be used to search for the target.

The list of context entries must come from the set of possible entries that can be ancestors of target entry. PC-Pages assumes a directory database with the hierarchy depicted in Figure 5 (this hierarchy is defined in the `[SEARCHCLASSHIERARCHY]` section).

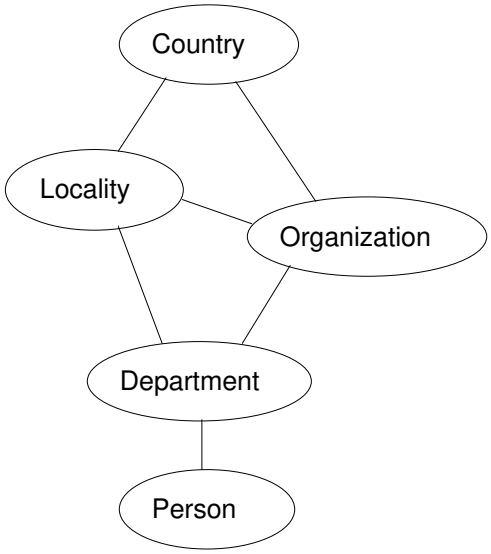

Figure 5. Default Directory Hierarchy

From this, the default context for 'Person' searches is:

```
Person: Country, Organization, Locality, Department
```

This results in a form search dialogue looking like the one shown in Figure 1 on page 7.

It could make sense to exclude some ancestors from the context list of a search class in order, say, to limit searches to your organization. This would also require changes to the search path associated with a search class (see the following description of the `[SEARCHCLASSPATHS]` section).

The [SEARCHCLASSPATHS] Section

This section defines the search paths for form based searches. The path used for a given search depends on the search class of the target entry and the values provided in the context fields. The syntax for this section is as follows:

```
targetname: contextlist : pathlist
```

The 'targetname' is the search class of the entry being searched for. The 'contextlist' is the set of context search classes that have search values provided (i.e. the fields of the form dialogue that have been filled in). An empty context list specifies the case when only the target field has a user supplied value. The 'pathlist' is a sequence of directory entry names or references to entries specified in the `[ENTRYDB]` section.

The default search paths for 'Person' searches are:

```
Person : Department, Organization, Locality: country
Person : Locality, Organization : country
Person : Organization : country
Person : Department : org
Person : : org
```

The default [ENTRYDB] section is listed on page 20.

The [SEARCHMENU] Section

This section specifies the contents of the 'Search' menu on the PC-Pages menu bar. The syntax is as follows:

```
classname : 1
```

This means that the named search class will have an entry present on the 'Search' menu.

The [UPDATECLASSES] Section

PC-Pages presents a subset of all possible attributes to the user when editing an entry. This subset is specified as follows:

classname : attributelist

Where 'classname' is a search class name and 'attributelist' is a comma separated list of attribute names.

The [ADDCLASSES] Section

The following information needs to be provided when an entry is added to the directory database:

- The attribute type of the RDN of the new entry.
- A complete list of `objectClass` attribute values that define the entry.

This information is provided as default values corresponding to an already defined search class. The syntax for this is:

```
classname : rdntype : objectclasslist
```

The 'rdntype' is an attribute type and 'objectclasslist' a list of comma separated `objectClass` values.

The [SHOWTEMPLATES] Section

The Entry Display windows information area is conceptually divided into a set of lines (each one text character high), with each line spanning two column widths. This area can then be filled with a set of individual text boxes containing the retrieved values of one or more attributes.

Each text box is specified with the following pieces of information:

- A label for the text box.
- A set of attributes whose values are to be displayed in the box.
- The start line (uppermost) of the text box.
- The height of the box (in text lines).
- Column span of the text box, i.e. left column, right column or span both.

This information is supplied using the following syntax:

```
textlabel : attributelist : startline : height : span
```

The 'span' is specified with one of `Left`, `Right` or `Both` as appropriate.

A different display template is used as appropriate to the type of entry being displayed. The syntax used to define a template for a specific entry type (as defined by it's search class) is:

```
%
class : classname
textboxlist
```

Where 'textboxlist' is a list of lines specifying text boxes (described above).

Appendix 3: Attributes and Syntaxes.

Each directory entry attribute consists of a name, an object identifier and a syntax. The syntax of defines the nature of the information contained in an attribute, e.g. a fax image, a line of text, etc.

Syntaxes may be supported in the following ways.

- Can be displayed.
- Editable (i.e. when adding or modifying entries).

Table 4 lists the attribute syntaxes understood by PC-Pages and the extent to which it supports them.

Syntax	Displayable	Editable
CaseExactString	Yes	Yes
TelephoneNumber	Yes	Yes
PrintableString	Yes	Yes
IA5String	Yes	Yes
DestinationString	Yes	Yes
CaseIgnoreString	Yes	Yes
CaseIgnoreIA5String	Yes	Yes
NumericString	Yes	Yes
OctetString	Yes	No
Password	Yes	Yes
TelexNumber	Yes	No
CaseIgnoreList	Yes	No
TeletexTerminalIdentifier	Yes	No
PostalAddress	Yes	Yes
Guide	No	No
FacsimileTelephoneNumber	Yes	Yes
PreferredDeliveryMethod	Yes	No
OID	Yes	No
Integer	Yes	No
Boolean	Yes	No
BitString	Yes	No
UTCTime	Yes	No
DN	Yes	No
ACL	No	No
EDBInfo	No	No

ProtectedPassword	Yes	No
InheritedAttribute	No	No
Mailbox	Yes	No
DataQuality	No	No
DSAQuality	No	No
DocumentStore	Yes	No
ORName	No	No
ORAddress	No	No
ORDescriptor	No	No
JPEG	Yes	No

Table 4. Attribute Syntaxes Supported by PC-Pages

Table 5 lists the attributes that PC-Pages understands along with their unique object identifiers and syntaxes. The attribute object identifier or table name (not informal name) can be used to key attributes in DUA.INI.

Some of the syntaxes listed below are currently ignored by PC-Pages. Support for these will be added in future versions.

Table name (informal name)	Object Identifier	Syntax
objectClass	attributeType.0	OID
aliasedObjectName	attributeType.1	DN
knowledgeInformation	attributeType.2	CaseIgnoreString
cn (commonName)	attributeType.3	CaseIgnoreString
sn (surname)	attributeType.4	CaseIgnoreString
serialNumber	attributeType.5	PrintableString
c (countryName)	attributeType.6	CountryString
l (locality)	attributeType.7	CaseIgnoreString
st (stateOrProvinceName)	attributeType.8	CaseIgnoreString
streetAddress	attributeType.9	CaseIgnoreString
o (organizationName)	attributeType.10	CaseIgnoreString
ou (organizationalUnitName)	attributeType.11	CaseIgnoreString
title	attributeType.12	CaseIgnoreString
description	attributeType.13	CaseIgnoreString
searchGuide	attributeType.14	Guide
businessCategory	attributeType.15	CaseIgnoreString
postalAddress	attributeType.16	PostalAddress
postalCode	attributeType.17	CaseIgnoreString
postOfficeBox	attributeType.18	CaseIgnoreString
physicalDeliveryOfficeName	attributeType.19	CaseIgnoreString
telephoneNumber	attributeType.20	TelephoneNumber

telexNumber	attributeType.21	TelexNumber
teletexTerminalIdentifier	attributeType.22	TeletexTerminalIdentifier
facsimileTelephoneNumber	attributeType.23	FacsimileTelephoneNumber
x121Address	attributeType.24	NumericString
internationaliSDNNumber	attributeType.25	NumericString
registeredAddress	attributeType.26	PostalAddress
destinationIndicator	attributeType.27	DestinationString
preferredDeliveryMethod	attributeType.28	DeliveryMethod
presentationAddress	attributeType.29	PresentationAddress
supportedApplicationContext	attributeType.30	OID
member	attributeType.31	DN
owner	attributeType.32	DN
roleOccupant	attributeType.33	DN
seeAlso	attributeType.34	DN
userPassword	attributeType.35	Password
userCertificate	attributeType.36	Certificate
cACertificate	attributeType.37	Certificate
authorityRevocationList	attributeType.38	CertificateList
certificateRevocationList	attributeType.39	CertificateList
crossCertificatePair	attributeType.40	CertificatePair
userid	pilotAttributeType.1	CaseIgnoreString
textEncodedORaddress	pilotAttributeType.2	CaseIgnoreString
rfc822Mailbox	pilotAttributeType.3	CaseIgnoreIA5String
info	pilotAttributeType.4	CaseIgnoreString
drink	pilotAttributeType.5	CaseIgnoreString
roomNumber	pilotAttributeType.6	CaseIgnoreString
photo	pilotAttributeType.7	Photofile
userClass	pilotAttributeType.8	CaseIgnoreString
host	pilotAttributeType.9	CaseIgnoreString
manager	pilotAttributeType.10	DN
documentIdentifier	pilotAttributeType.11	CaseIgnoreString
documentTitle	pilotAttributeType.12	CaseIgnoreString
documentVersion	pilotAttributeType.13	CaseIgnoreString
documentAuthor	pilotAttributeType.14	DN
documentLocation	pilotAttributeType.15	CaseIgnoreString
homePhone	pilotAttributeType.20	TelephoneNumber
secretary	pilotAttributeType.21	DN
otherMailbox	pilotAttributeType.22	Mailbox
lastModifiedTime	pilotAttributeType.23	UTCTime
lastModifiedBy	pilotAttributeType.24	DN
domainComponent	pilotAttributeType.25	CaseIgnoreIA5String
aRecord	pilotAttributeType.26	IA5String
mDRecord	pilotAttributeType.27	IA5String

mXRecord	pilotAttributeType.28	IA5String
nSRecord	pilotAttributeType.29	IA5String
sOARecord	pilotAttributeType.30	IA5String
cNAMERecord	pilotAttributeType.31	IA5String
associatedDomain	pilotAttributeType.37	CaseIgnoreIA5String
associatedName	pilotAttributeType.38	DN
homePostalAddress	pilotAttributeType.39	PostalAddress
personalTitle	pilotAttributeType.40	CaseIgnoreString
mobileTelephoneNumber	pilotAttributeType.41	TelephoneNumber
pagerTelephoneNumber	pilotAttributeType.42	TelephoneNumber
friendlyCountryName	pilotAttributeType.43	CaseIgnoreString
uniqueIdentifier	pilotAttributeType.44	CaseIgnoreString
organizationalStatus	pilotAttributeType.45	CaseIgnoreString
janetMailbox	pilotAttributeType.46	CaseIgnoreIA5String
mailPreferenceOption	pilotAttributeType.47	MailPreference
buildingName	pilotAttributeType.48	CaseIgnoreString
dSAQuality	pilotAttributeType.49	DSAQualitySyntax
singleLevelQuality	pilotAttributeType.50	DataQualitySyntax
subtreeMinimumQuality	pilotAttributeType.51	DataQualitySyntax
subtreeMaximumQuality	pilotAttributeType.52	DataQualitySyntax
dITRedirect	pilotAttributeType.54	DN
documentPublisher	pilotAttributeType.56	CaseIgnoreString
jpegPhoto	pilotAttributeType.60	jpeg
timeStamp	thornSystemAttributeType.2	UTCTime
ruleDescription	thornSystemAttributeType.3	RDType
objectDescription	thornSystemAttributeType.4	ODType
attributeDescription	thornSystemAttributeType.5	ADType
thornRule	thornSystemAttributeType.6	OID
knowledgeReference	thornSystemAttributeType.7	KnowledgeReference
treeStructure	quipuAttributeType.1	Schema
acl (AccessControlList)	quipuAttributeType.2	Acl
eDBinfo	quipuAttributeType.3	Edbinfo
masterDSA	quipuAttributeType.4	DN
slaveDSA	quipuAttributeType.5	DN
control	quipuAttributeType.15	IA5String
quipuVersion	quipuAttributeType.16	IA5String
protectedPassword	quipuAttributeType.17	ProtectedPassword
entrySecurityPolicy	quipuAttributeType.18	SecurityPolicy
dsaDefaultSecurityPolicy	quipuAttributeType.19	SecurityPolicy
dsaPermittedSecurityPolicy	quipuAttributeType.20	SecurityPolicy
inheritedAttribute	quipuAttributeType.21	InheritedAttribute
execVector	quipuAttributeType.22	PrintableString
relayDSA	quipuAttributeType.23	DN

subordinateReference	quipuAttributeType.25	AccessPoint
crossReference	quipuAttributeType.26	AccessPoint
nonSpecificSubordinateReference	quipuAttributeType.27	AccesPoint
listenAddress	quipuAttributeType.28	PresentationAddress
cachedEDB	quipuAttributeType.29	DN
mhsDeliverableContentLength	mhsAttributeType.0	Integer
mhsDeliverableContentTypes	mhsAttributeType.1	OID
mhsDeliverableEits	mhsAttributeType.2	OID
mhsDLMembers	mhsAttributeType.3	ORName
mhsDLSubmitPermissions	mhsAttributeType.4	DLSubmitPermission
mhsMessageStoreName	mhsAttributeType.5	DN
mhsORAddresses	mhsAttributeType.6	ORAddress
mhsPreferredDeliveryMethods	mhsAttributeType.7	DeliveryMethod
mhsSupportedAutomaticActions	mhsAttributeType.8	OID
mhsSupportedContentTypes	mhsAttributeType.9	OID
mhsSupportedOptionalAttributes	mhsAttributeType.10	OID
dl-policy	ppAttributeType.3	DlPolicy
obsoletesDocument	oldPSIattributeType.3	DN
obsoletedByDocument	oldPSIattributeType.4	DN
updatesDocument	oldPSIattributeType.5	DN
updatedByDocument	oldPSIattributeType.6	DN
keywords	oldPSIattributeType.7	CaseIgnoreString
subject	oldPSIattributeType.8	CaseIgnoreString
abstract	oldPSIattributeType.9	CaseIgnoreString
documentStore	oldPSIattributeType.10	DocumentStoreSyntax
documentAuthorCommonName	oldPSIattributeType.11	CaseIgnoreString
documentAuthorSurName	oldPSIattributeType.12	CaseIgnoreString
whoisIdent	oldPSIattributeType.13	CaseIgnoreString
fipsStateNumericCode	nadfAttributeType.1	NumericString
fipsStateAlphaCode	nadfAttributeType.2	PrintableString
fipsCountyNumericCode	nadfAttributeType.3	NumericString
fipsPlaceNumericCode	nadfAttributeType.4	NumericString
ansiOrgNumericCode	nadfAttributeType.5	Integer
addmdName	nadfAttributeType.6	CaseIgnoreString
nadfSearchGuide	nadfAttributeType.7	NadfGuide
supplementaryInformation	nadfAttributeType.8	CaseIgnoreString
namingLink	nadfAttributeType.9	DN
reciprocalNamingLink	nadfAttributeType.10	DN

Table 5. Attributes Supported by PC-Pages

Appendix 4. Object Classes

This appendix lists the properties of each objectClass value that is understood by PC-Pages. The properties listed in Table 6 are:

- Object class value name.
- Object identifier.
- Super classes (if any).
- Attributes made mandatory (must) by the object class.
- Attributes that are optional (may) to the object class.

Note that mandatory or optional attributes may be referred to by their inclusion in an attribute set. These sets are listed in Table 7.

Object class name	Object identifier	Superclass(es)	Must attributes	May attributes
top	standardObjectClass.0		objectClass	
alias	standardObjectClass.1	top	aliasedObjectName	aliasedObjectName
country	standardObjectClass.2	top	c	description searchGuide
locality				description localityName stateOrProvinceName searchGuide seeAlso streetAddress
organization	standardObjectClass.3	top	O	organizationalAttributeSet
organizationalUnit	standardObjectClass.4	top	OU	organizationalAttributeSet
person	standardObjectClass.5	top	CN surname	description seeAlso telephoneNumber userPassword
organizationalPerson	standardObjectClass.6	top		localeAttributeSet OU postalAttributeSet telecommunicationAttributeSet title
organizationalRole	standardObjectClass.7	person	CN	description localeAttributeSet OU postalAttributeSet preferredDeliveryMethod roleOccupant seeAlso telecommunicationAttributeSet
groupOfNames	standardObjectClass.8	top	CN member	description O OU owner seeAlso businessCategory
residentialPerson	standardObjectClass.9	top	localityName	localeAttributeSet postalAttributeSet preferredDeliveryMethod telecommunicationAttributeSet businessCategory

applicationProcess	standardObjectClass.10	person	CN	description localityName OU seeAlso
applicationEntity	standardObjectClass.11	top	CN presentationAddress	description localityName O OU seeAlso supportedApplicationContext
dSA	standardObjectClass.12	top		knowledgeInformation
device	standardObjectClass.13	applicationEntity	CN	description localityName O OU owner seeAlso serialNumber
strongAuthenticationUser	standardObjectClass.14	top	userCertificate	
certificationAuthority	standardObjectClass.15	top	cACertificate certificateRevocationList authorityRevocationList	crossCertificatePair
thornObject,pilotObject	standardObjectClass.16	top		info photo manager uniqueIdentifier lastModifiedTime lastModifiedBy dITRedirect audio
thornPerson,newPilotPerson	pilotObjectClass.3	top		userid textEncodedORAddress rfc822Mailbox favouriteDrink roomNumber userClass homePhone homePostalAddress secretary personalTitle preferredDeliveryMethod businessCategory janetMailbox otherMailbox mobileTelephoneNumber pagerTelephoneNumber organizationalStatus mailPreferenceOption personalSignature
account	pilotObjectClass.4	person	userid	description seeAlso localityName organizationName organizationalUnitName host

document	pilotObjectClass.5	top	documentIdentifier	commonName description seeAlso localityName organizationName organizationalUnitName documentTitle documentVersion documentAuthor documentLocation obsoletesDocument obsoletedByDocument updatesDocument updatedByDocument keywords subject abstract documentAuthorCommonName documentAuthorSurName documentStore documentPublisher
room	pilotObjectClass.6	top, pilotObject	commonName	roomNumber description seeAlso telephoneNumber
documentSeries	pilotObjectClass.7	top	commonName	description seeAlso telephoneNumber localityName organizationName organizationalUnitName
domain	pilotObjectClass.9	top	domainComponent	associatedName organizationName organizationalAttributeSet
rFC822localPart	pilotObjectClass.13	top		commonName surname description seeAlso telephoneNumber postalAttributeSet telecommunicationAttributeSet
dNSDomain	pilotObjectClass.14	domain		ARecord MDRecord MXRecord NSRecord SOARecord CNAMERecord
domainRelatedObject	pilotObjectClass.15	domain	associatedDomain	
friendlyCountry	pilotObjectClass.17	top	friendlyCountryName	
simpleSecurityObject	quipuObjectClass.3 pilotObjectClass.18	country	userPassword	
pilotOrganization	pilotObjectClass.19	top		buildingName
pilotDSA	pilotObjectClass.20	organization, organizationalUnit	dSAQuality	
qualityLabelledData	pilotObjectClass.21	dsa	singleLevelQuality	subtreeMinimumQuality subtreeMaximumQuality
quipuDSA	pilotObjectClass.23	top	acl edbinfo userPassword manager quipuVersion	description lastModifiedBy lastModifiedTime dsaDefaultSecurityPolicy dsaPermittedSecurityPolicy relayDSA listenAddress info dSAQuality
quipuObject	quipuObjectClass.1	dSA	acl	lastModifiedBy lastModifiedTime entrySecurityPolicy

quipuNonLeafObject	quipuObjectClass.2	top	masterDSA	slaveDSA treeStructure inheritedAttribute
quipuSecurityUser	quipuObjectClass.6	quipuObject	protectedPassword	
iSODEApplicationEntity	quipuObjectClass.7	quipuObject	execVector	
externalNonLeafObject	quipuObjectClass.8	applicationEntity		subordinateReference crossReference nssr
thornSchema	quipuObjectClass.9	quipuObject	timestamp ruleDescription objectDescription attributeDescription commonName	
knowledgeManagement	thornSystemObjectClass.1	top	KnowledgeReference	knowledgeInformation
thornSystemObject	thornSystemObjectClass.2	top	thornRule	thornACL
mhsDistributionList	thornSystemObjectClass.3	top	cn mhsDLSubmitPermissions mhsORAddresses	description o ou owner seeAlso mhsDeliverableContentTypes mhsdeliverableEits mhsDLMembers mhsPreferredDeliveryMethods
mhsMessageStore	mhsObjectClass.0	top		description owner mhsSupportedOptionalAttributes mhsSupportedAutomaticActions mhsSupportedContentTypes
mhsMessageTransferAgent	mhsObjectClass.1	applicationEntity		description owner mhsDeliverableContentLength
mhsOrganizationalUser mhsUser	mhsObjectClass.2	applicationEntity	mhsORAddresses	mhsDeliverableContentLength mhsDeliverableContentTypes mhsDeliverableEits mhsMessageStoreName mhsPreferredDeliveryMethods
mhsUserAgent	mhsObjectClass.3	top		mhsDeliverableContentLength mhsDeliverableContentTypes mhsDeliverableEits mhsORAddresses owner
ppDistributionList	mhsObjectClass.4 mhsObjectClass.5	applicationEntity	mhsDLMembers owner	dl-policy
ppRole	ppObjectClass.1	mhsDistributionList quipuObject	mhsORAddresses	mhsDeliverableContentLength mhsDeliverableContentTypes mhsDeliverableEits mhsMessageStoreName mhsPreferredDeliveryMethods
onlineDocument	ppObjectClass.2	OrganizationalRole	documentStore	
nadfObject	psiObjectClass.1	document quipuObject		lastModifiedTime nadfSearchGuide supplementaryInformation
usStateOrEquivalent	nadfObjectClass.7	top	localityName fipsStateNumericCode fipsStateAlphaCode stateOrProvinceName	
usPlace	nadfObjectClass.1	locality nadfObject	localityName fipsPlaceNumericCode	
usCountyOrEquivalent	nadfObjectClass.2	locality nadfObject	fipsCountyNumericCode	
usOrganization	nadfObjectClass.3	usPlace	ansiOrgNumericCode	

nadfApplicationEntity	nadfObjectClass.4	organization nadfObject	supportedApplicationContext	
nadfADDMD	nadfObjectClass.5	applicationEntity	addmdName	organizationName organizationalAttributeSet
publicObject	nadfObjectClass.8	top	namingLink	
providerObject	nadfObjectClass.9	top	reciprocalNamingLink	
foreignOrganization	nadfObjectClass.10	organization	countryName	
fips55Object	nadfObjectClass.11	top	fipsPlaceNumericCode	stateOrProvinceName

Table 6. Object classes and their properties.

Attribute group	**Group contents**
telecommunicationAttributeSet	facsimileTelephoneNumber internationaliSDNNumber telephoneNumber teletexTerminalIdentifier telexNumber preferredDeliveryMethod destinationIndicator registeredAddress x121Address
postalAttributeSet	physicalDeliveryOfficeName postalAddress postalCode postOfficeBox streetAddress
localeAttributeSet	localityName stateOrProvinceName streetAddress
organizationalAttributeSet	description localeAttributeSet postalAttributeSet telecommunicationAttributeSet businessCategory seeAlso searchGuide userPassword

Table 7. Attribute sets.

Appendix 5. Glossary of Terms

This appendix contains a glossary of terms that appear in this manual or occur in PC-Pages itself..

Attribute

An attribute is a component of a directory entry. Some example attributes are `postalAddress`, `telephoneNumber` and `surname`.

An attribute is composed of an *attribute type* and *attribute value*. The type defines the syntax and semantic of the attribute. For example the syntax of the `postalAddress` attribute is a list of 6 lines of text with each line being a maximum of 32 characters in length, it's semantic simply being that of a postal address.

Attribute Syntax

See the glossary entry for **Attribute.**

Attribute Type

See the glossary entry for **Attribute.**

Attribute Value

See the glossary entry for **Attribute.**

Child entry

See the glossary entry for **Directory Database**.

Directory Database

The directory is a structured database consisting of a set of *directory entries* organized in a hierarchical structure. Entries can have parents and children. The parent of a person's entry might be the department or organization that that person works in. Conversely a person is usually the child of an organization or department.

Directory Entry

A directory entry is a collection of information on a real world object, e.g. a person, room, mailing list, etc. A directory entry is built up of a set of *attributes* that describe the object.

Directory Name

A directory name is an informal and approximate version of an entries *distinguished name*. It consists of a series of comma separated components, for example:

```
Fred Smith, English Language, Muddle College, GB.
```

The directory name for a "Person" entry usually consists of a sequence of the person's name, their department name, organization name and country. The directory name format was designed with 'guessability' in mind.

Approximation is permitted: by omitting name components (notably the department name when searching for a person) or supplying incomplete values (e.g. entering 'Smith' instead of 'Fred Smith').

Distinguished Name

A distinguished name, or DN for short, is a machine readable name for directory entries. The syntax for directory names is complex. It consists of a series of name components called RDNs (Relative Distinguished Names). Each RDN component is built of an *attribute type* and *attribute value* pair. The RDN for a room entry might look like:

```
room=Staff Common Room
```

A full distinguished name consists of a series of RDNs separated by the '@' character. A full DN then looks like the following:

```
c=GB@o=Muddle College@ou=English Language@cn=Fred Smith
```

DN

See the glossary entry for **Distinguished Name**.

Object Class

Each entry has an object class attribute (table name `objectClass`). This attribute contains a number of values which define all other attributes that the entry can contain. Additionally attributes defined by an object class value can be included on a mandatory (*'must'*) or optional (*'may'*) basis. For example an entry that has an object class including the `organization` value 'must' contain the name of that organization and 'may' contain attributes such as `postalAddress`, `telephoneNumber`, `telexNumber`, etc.

Appendix 4 lists object class values along with their associated 'must' and 'may' attributes.

Object Identifier

An object identifier is a sequence of numbers used to uniquely identify any *attribute type* or *object class* value. In text form the object identifier consists of a series of numbers separated by the '**.**' character.

Each numeric object identifier also has a unique text label called, for the purposes of this document. it's *table name*. This term is used because the whole set of object identifiers used in any application is referred to as the object identifier table (or OIDTABLE for short).

To illustrate this, the object class attribute has the table name `objectClass` and it's object identifier is 2.5.4.0.

Table names and object identifiers are listed in Appendix 4 for object classes and Appendix 3 for attribute types.

Parent entry

See the glossary entry for **Directory Database**.

RDN

See the glossary entry for **Distinguished Name**.

Search Class

PC-Pages uses the object class attribute to classify directory entries. This is necessary in order, amongst other things, to be able to search for entries efficiently. In the default configuration an 'Organization' entry is defined to be any entry that contains the `organization` value in it's object class attribute. Similar defaults appear for 'Person', 'Department' and 'Country' entries.

Appendix 2 contains a description of the `[SEARCHCLASSTYPES]` section of DUA.INI which is used to specify the search classes used by PC-Pages.

Search Path

A search path is a sequence of directory entries used to direct a search towards a part of the directory database that is most likely to contain the data being looked. For example, it may be assumed that someone performing a search is more likely to look for people within his own organization or department than elsewhere.

Appendix 2 contains a description of the `[SEARCHCLASSPATHS]` and `[UFNPATHS]` section of DUA.INI, which are used to specify paths for form based and directory name based searches respectively.

Table Name

See the glossary entry for **Object Identifier**.

Index

A

Adding entries, 10

configuring, 22

Address book, 10

configuring, 16

Attributes

glossary entry, 34

naming, 20

B

Binding, *see Logging in*

Browse, 10

C

Collect window, 10, 11

Colour preferences, 18

Connection time

idle, 17

CONS, *see Pinkbook*

Context field, 6

D

Deleting entries, 10

Directory database

glossary entry, 34

Directory entry

glossary entry, 34

Directory name, *see Name based search*

Directory server address, 5, 15

Distinguished name

glossary entry, 35

DN, *see Distinguished name*

DSA address, *see Directory server address*

DUA.INI

options in, 17-23

E

Edit window, 10, 12
Editing entries
configuring, 22
Electronic mail, 10
display format, 19
Entry window, 9-10

F

Fax numbers
editing, 13
Form based search, 5-7
configuring, 20-22

I

Image attribute, 10
displaying, 19
Installation, 4-5

L

List window, 11
Logging in, 5

M

Mail user agent
configuring, 16
MUA, *see Mail user agent*

N

Name based search, 8, 34
configuring, 19-20
NetBIOS, *see WhiteGate*

O

Object class
glossary entry, 35
Object identifier
glossary entry, 35

P

Password, *see User password*
Photographs, *see Image attribute*
Pinkbook, 5
Postal addresses

editing, 12
Presentation address, 5

R

RBW.INI, 4, 5
options in, 15-16
syntax, 15
RDN, *see Distinguished name*
Role occupant attribute, 10

S

Save entry, *see Address book*
Search class
glossary entry, 36
Search path
glossary entry, 36
Secretary attribute, 10
See also attribute, 9-10
Size limit
directory requests, 18
Startup help, 5

T

Table name
glossary entry, 36
Target entry, 6
Target field, 6
TCP/IP, 5
Text attributes
editing, 13
Time limit
directory requests, 18
Transport stacks, 4, 15

U

User password, 5
automatic authentication, 17

W

WhiteGate, 4, 15
WhiteLite, 15

WhiteMail, 4, 5
WhiteStack, 4, 5, 15
Windows, 4