PowerSHAPE 2015 R2

Reference Help

Managing your PowerSHAPE system

Delcam Advanced Manufacturing Solutions

PowerSHAPE

Delcam Ltd has no control over the use made of the software described in this manual and cannot accept responsibility for any loss or damage howsoever caused as a result of using the software. Users are advised that all the results from the software should be checked by a competent person, in accordance with good quality control procedures.

The functionality and user interface in this manual is subject to change without notice in future revisions of the software.

Acknowledgements

This documentation references a number of registered trademarks and these are the property of their respective owners. For example, Microsoft and Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States.

Patent Information

Emboss functionality is subject to patent number GB 2389764 and patent applications US 10/174524 and GB 2410351.

Morphing functionality is subject to patent application GB 2401213.

PowerSHAPE 2015 R2. Published on 09 March 2015

Contents

Managing your product 2

Managing your product

To manage PowerSHAPE, you need to be able to:

- install PowerSHAPE.
- configure your system (see page 3).
- set up users (see page 4).
- manage your data (see page 21).
- solve common problems (see page 40).
- use commands and scripts to manage PowerSHAPE (see page 55).

Understanding the powershape.con file

The *powershape.con* file contains information about the configuration of your software. Optionally, you can create another file that contains customisation details.

The following assumes that C:\Program files\Delcam\ is the default installation directory of the software.

The default.*powershape.con* file is read from:

C:\Program files\Delcam\powershapeXXXX\sys\misc\

You can make changes to the default *powershape.con* file, but these changes are overwritten when you install a new version of the software. To ensure that you retain any changes when future versions of the program are installed, you can create a customised version of the *powershape.con* file. This file is created in C:\Program files\Delcam\config and custom changes are added to this file.

Any changes that are stored in this file overwrite the defaults in C:\Program files\Delcam\powershapeXXXX\sys\misc\, because anything new that is added to the program is included before the customised settings are added.

If the default value is changed in the newly installed version the user changes overwrite this. It is recommended that you check the powershape.con file in the new release to see if any changes have been made.

Creating a customised powershape.con file

The following assumes that C:\Program files\Delcam\ is the default installation directory of PowerSHAPE.

Assuming a default installation directory, navigate to C:\Program files\Delcam\.

1 If the *config* directory doesn't exist, create a directory at the same level as the PowerSHAPE directory called **config**. For example,

   C:\Program files\Delcam\config

2 In this directory you need to create a new text file called *powershape.con*.

3 In the *powershape.con* that is located in C:\Program files\Delcam\powershapeXXXX\sys\misc\, identify the string that you want to change for every version of PowerSHAPE.

For example, if you add the following to the custom *powershape.con*, the inside colour will be blue instead of red.

```
inside_colour: 0.0, 0.0, 1.0
```

Monitor sizes

1:1 zoom is supported on widescreen monitors, but you can use powershape.con settings to specify an unusual size.

Model sizes

PowerSHAPE supports model sizes over 2GB by default. This is controlled by the `large_sector_size` command in the powershape.con file:

```
large_sector_size: true
```

This resource takes effect on new models. Alternatively, use **Save As** to save older models with a new name. Models with larger sector sizes can be opened by with older versions of PowerSHAPE.

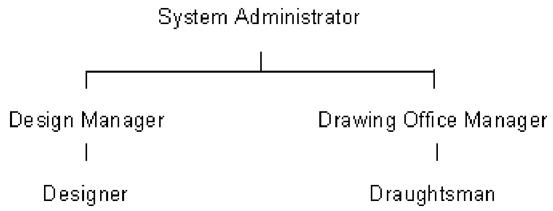
Users

The following sections contain information on users.

What is a user?

A user is a person who is set up to use PowerSHAPE. The software keeps a list of users.

Each user is allocated a security level that gives them privileges associated with one of the roles below.



PowerSHAPE restricts access to certain functionality to users of a particular security level, for example, only a System Administrator or Drawing Office Manager can delete a database area.

The System Administrator is responsible for controlling the security and access rights of users of PowerSHAPE. This could be the same person as the Design Manager or Drawing Office Manager.

Each user is defined by the following information:

- Short login name (often their initials or first name).
- User's full name.
- Password (optional).
- Security level.
- Pathname of a macro that will run whenever the user logs on. This macro can be used to set personal preferences such as those available using the Options dialogs

You can also tell PowerSHAPE to:

- restrict a user from accessing certain models.
- store the models created by a user in a particular area on the network.

For further details, see Managing your data (see page 21).

What are security levels?

Security levels (given below) are assigned to each user and restrict users from carrying out certain administrative tasks.

Level	Description
0	This is used by Delcam system support staff.
1	This allows access to all system administration and design functions. This level is assigned to the system administrator.

2	This allows access to most (but not all) system administration and all design functions. This level is usually assigned to the design office manager.
3	This allows access to design functions, but no access to administration functions. This level is assigned to the designers.

Setting up users

The first tasks your System Administrator must perform are to:

- draw up a list of all the users who need access to PowerSHAPE.
- design a naming-convention for models. For further details, see Managing your data (see page 21).
- establish what access rights individuals and groups of users need for individual projects. For further details, see Managing your data (see page 21).

Setting up a new user

When you run PowerSHAPE for the first time, a new user is automatically created for you using your Windows login name (with security level 3).

Although you can set yourself up as a user by simply running PowerSHAPE, you must have administrative privileges to set up other users.

1. Run PowerSHAPE.
2. Select **View > Window > Command** to display the command window at the bottom of the screen.
3. In the command window, type:

   ```
   admin create user
   ```

4. Type in the login name of the user. This is usually the same as the user’s system login name.

   Type in a password for the user.

   Your Windows and your PowerSHAPE passwords need not be the same, and in fact most users choose not to have a PowerSHAPE password at all. If you do have a PowerSHAPE password, you will always be prompted for it when starting PowerSHAPE.

   If no password is required, type:

none

5. Type in the name of the user in single quotes, for example,

   'Penny Black'

6. Type in the security level of the user.
7. Type in the path of the login macro for the user.

   If no login macro is required, type:

   none

   and the default login macro is used. For further details, see What is the login macro? (see page 7)

What is the login macro?

The login macro is one of the macro files that run when you start up PowerSHAPE. It sets up the system to your personal requirements. For example, you can add the drawing standard to this macro so that when you start up PowerSHAPE, it uses that standard.

In total, three macros are run when PowerSHAPE is started, in the following order:

1. *preconfig.mac*
2. user's login macro or login.mac (the default login macro)
3. *postconfig.mac*

The following path is searched for the macros:

<disk>:\dcam\product\powershape<your version number>\lib\macro

The path you specified when the user was set up is searched for user's login macro.

You can add commands to the default login macro, *login.mac*, that users may want to use in their login macro. When you create a user, if a path is given for the login macro and no file already exists there, then the default login macro file is copied to that location. The user can then modify this file to their requirements.

Use *preconfig.mac* and *postconfig.mac* to store commands to set up PowerSHAPE for everyone at your company. *postconfig.mac* gives you some control over the settings in the user’s login.mac.

For further details, see Writing macros in the Customising PowerSHAPE for your application

Logging in as another user

There is not necessarily a direct one-to-one relationship between a Windows user and a PowerSHAPE user. For example, you can log into Windows as user *ano,* yet log into PowerSHAPE as user *abc.* It is typical to have a PowerSHAPE user set up for administration tasks (often called *admin)* so it is often convenient to swap between being logged in as your normal user, to being logged in as *admin* and back again.

To log in as another user,

1 From the **Tools** menu, select **Login...**.

If models are open and have been modified, a dialog is displayed asking whether you want to lose the changes. If you do not wish to lose the changes, click **No** and this will cancel the login command.

2 Complete the Login dialog.

3 Click **OK**.

The security rights change to those of the new user.

If the user is unknown to PowerSHAPE, you can try again. If after six retries the login attempt has still not succeeded, the command is automatically aborted by PowerSHAPE.

To add a user to the list of known users, see Setting up a new user (see page 6).

Passwords

You can:

Change your password (see page 8)

Change another user's password (see page 9)

Changing your password

1 Select **View > Window > Command** to display the command window at the bottom of the screen.

2 In the command window, type:

admin edit user

3 Type:

password

4 Type in your new password.

Changing another user's password

You may only change the password of users with a lower security level than your own.

1 Select **View > Window > Command** to display the command window at the bottom of the screen.

2 In the command window, type:

admin edit user

3 Type in the user name of the user.

4 Type:

password

5 Type in the user's new password.

Deleting a user

1 Select **View > Window > Command** to display the command window at the bottom of the screen.

2 In the command window, type:

admin delete user

3 Type in the username and password of the user you wish to delete.

Listing registered users

1 Select **View > Window > Command** to display the command window at the bottom of the screen.

2 In the command window, type:

admin state user

This lists all the users and their login details.

Colour schemes

You can define your own colour scheme for PowerSHAPE using information from the following sections:

Setting up a colour scheme for graphics (see page 10)

Colour definitions (see page 10)

Colours used in PowerSHAPE (see page 12)

Setting up a colour scheme for graphics

Not only can you edit the colours of the graphics using the **Main** and **Style** toolbars, you can also define your own colour scheme for your graphics. In this section, we show you just how to do that.

Colours are defined using two files. One file is the colour table and stores the colour definitions. The other file uses the colour definitions from the first file to define the colour scheme . You can edit both these files.

Colour definitions

Given below is the colour table file provided with PowerSHAPE. This contains the colour definitions.

```
46
Black 0 0 0
White 1 1 1
Red 1 0 0
Green 0 1 0
Blue 0 0 1
Yellow 1 1 0
Cyan 0 1 1
Magenta 1 0 1
Orange 1.0 0.6 0.1
GreenYellow 0.6 0.83 0.0
BlueCyan 0.0 0.5 1.0
Black 0.0 0.0 0.0
GreenCyan 0.0 1.0 0.6
BlueMagenta 0.55 0.0 1.0
RedMagenta 1.0 0.0 0.55
FireBrick 0.75 0.13 0.13
Moccasin 1 0.89 0.71
Orchid 0.85 0.24 0.88
Violet 0.73 0.21 0.93
Tan 0.82 0.71 0.55
Navy 0 0 0.5
OldLace 0.99 0.96 0.90
Chocolate 0.82 0.41 0.12
```

```
Charteuse 0.5 1 0
Azure 0.94 1 1
MidGreen 0.2 0.8 0.3
PaleGrey 0.8 0.8 0.8
LightGrey 0.6 0.6 0.6
MidBlueGrey 0.33 0.33 0.45
DarkBlueGrey 0.34 0.34 0.5
MidGrey .5 .5 .5
DarkGrey 0.4 0.4 0.4
DarkSlate 0.28 0.35 0.55
DarkOrange 1 0.3 0.0
Yellowish 1.0 1.0 0.5
RedOrange 0.8 0.4 0.0
Pink 1.0 0.6 0.6
Maroon 0.6 0.0 0.4
Mauve 0.7 0.55 0.8
BottleGreen 0 0.75 0
Claret 0.8 0.0 0.7
Amber 0.8 0.5 0.0
BrightPink 1.0 0.7 0.7
DarkMagenta 1 0 0.75
ElectricBlue 0 .9 .9
Peppermint .5 1 .5
```

We will now explain the contents of this file. On the first line is a number. This number is the number of colour definitions in the file. The remaining lines each contain a colour definition. Each colour definition is defined as colour code name followed by its RGB colour mix.

We have provided you with a variety of colour definitions, but feel free to add your own colours. For further details see Changing the colour definitions (see page 11).

Changing the colour definitions

1. Copy the following file to your home area:
2. c:\dcam\product\powershapeXXXX\sys\hci\colourscheme\colour.tab

   where XXXX is your version number of PowerSHAPE and C is the disk on which PowerSHAPE is installed.

Open your copy of colour.tab in a text editor.

3 In this file, change the colour definitions to your requirements.

Add a new colour by adding a unique code name of the colour and its RGB mix at the end of the file. For example,

```
RedOrange 0.8 0.4 0.0
```

Make sure the number on the first line matches the number of colour definitions in the file. Also make sure that there is an empty line at the end of the file.

Edit an existing colour by changing its RGB mix.

4 Save the file.

To use your copy of colour.tab, set the environment variable DCAM_COLOUR_TABLE to the path of your file. Details on how to do this are given below.

5 Click **Start** at bottom left of the screen.

6 Move the mouse to **Settings** and, from the pop-up menu, select **Control Panel** to display the **Control Panel** window.

7 In the **Control Panel** window, double click the **System** icon to display the **System Properties** dialog.

8 Click the **Environment** tab.

9 In the **Variable** text box, type:

DCAM_COLOUR_TABLE

10 In the **Value** text box, enter the pathname to your copy of colour.tab. For example,

c:\mypsfiles\colour.tab

11 Click **OK**.

12 Log out and into the system again for these changes to be effective.

Colours used in PowerSHAPE

Given below are the contents of the file of colour definitions used by PowerSHAPE.

```
# Colour table
# this supplies the colours displayed with
# the colour selector and used to show each
# line, etc displayed within the software.
#
# The last five colours are used for
# instrumentation.
#
# Note use of backslash is required.
#
# A colour entry like Red can also be
# replaced with its rgb entry e.g.
#   1 0 0
# or
#   1.0 0.0 0.0
# ===============================================

colour_table: \
16 \
Black \
White \
Red \
Green \
Blue \
ElectricBlue \
DarkMagenta \
Yellowish \
Amber \
BottleGreen \
Peppermint \
Yellow \
BlueCyan \
DarkOrange \
BrightPink \
MidGrey

# Now resources covering view background
# colour, etc
graduated: on
```

```
top_colour: 0.0 0.0 0.0
bottom_colour: 0.2 0.4 0.8

# Colour to use when rubber-banding to create
# new lines, etc Get some slight blues and
# reds with this colour but rubber-banding is
# at least always visible
xor_colour: 1.0 1.0 1.0

#
# Colours for dynamic editing of drafting
# text
text_cursor_colour: Pink
text_box_colour: Green
text_box_highlight_colour: Cyan
#
# Colours for the rotation tracker
#   These are the colours of the lines and
#   text of the rotation tracker controlled
#   under tools -> options -> views ->
#   rotation tracker
#
#   Resources control line and text colours
#   when displaying the world axes or a
#   workplane axes system.
#
#   Resources also control the position of
#   the axes also which defaults to
#   bottom_left, but can be one of right,
#   left, top, bottom top_right, top_left,
#   bottom_right, and bottom_left.
#
#   The attenuation resource controls how
#   much colours of line and text are reduced
#   when an axes points away from the viewer.
#
rotation_tracker_line_colour: 0.4 0.95 1.0
rotation_tracker_text_colour: 0.4 0.95 1.0
rotation_tracker_active_colour: 1.0 1.0 1.0
rotation_tracker_workplane_line_colour: 0.4 0.95 1.0
rotation_tracker_workplane_text_colour: 0.4 0.95 1.0
rotation_tracker_pos_type: bottom_left
rotation_tracker_colour_attenuation: 0.7
```

We will now explain the contents of this file. Any line beginning with # is a comment. At the start of the file is a group of comments.

The colour table defines 16 colours used in PowerSHAPE:

```
colour_table: \
16 \
Black \
White \
Red \
Green \
Blue \
ElectricBlue \
DarkMagenta \
Yellowish \
Amber \
BottleGreen \
Peppermint \
Yellow \
BlueCyan \
DarkOrange \
BrightPink \
MidGrey
```

Each line of the colour table must end with \ except the last entry. The first line states that the colour table is being defined. On the second line of the table is a number. This number is the number of colours used in PowerSHAPE and it must not change. The remaining lines each contain a colour code name that is defined in the colour table file called colour.tab. You can also define the colour as a RGB colour mix.

All of the colours are available in the **Main** and **Style** toolbars. You can use them to change the colour of objects.

Note that PowerSHAPE uses the first 2 and the last 5 colours for specific purposes (see the table below).

Colour number	Colour in default file	What this colour shows in PowerSHAPE
Colour 1	Black	Window background.
Colour 2	White	Default foreground colour for objects.
Colour 12	Yellow	Selected objects.
Colour 13	BlueCyan	Edit handles. Outside detail of the bright tracker ball.
Colour 14	DarkOrange	An object that can be selected when the intelligent cursor passes over. Selected surface curves. Surface normals. Inside detail of the bright tracker ball.
Colour 15	BrightPink	Secondary selections. Outside detail of the dull tracker ball.
Colour 16	MidGrey	A locked object. Detailed surface. Inside detail of the dull tracker ball.

If you change Colours 12 and 16, make sure Colour 12 is a bright colour and Colour 16 a shade of grey.

The next part of the file defines the background colour of view.

```
# Now resources covering view background
# colour, etc
graduated: on
```

```
top_colour: 0.0 0.0 0.0
bottom_colour: 0.2 0.4 0.8
```

It contains three variables: graduated, top_colour and bottom_colour. The variable graduated can be on or off. If it is off, the background colour is defined using the first colour of the colour table. If on, the views in PowerSHAPE have a graduated background using the colours defined by the top_colour and bottom_colour variables.

The code below defines the colour of the rubber banding.

```
# Colour to use when rubber-banding to create
# new lines, etc Get some slight blues and
# reds with this colour but rubber-banding is
# at least always visible
xor_colour: 1.0 1.0 1.0
```

Similarly, the code below defines the colours used to dynamically edit text in the drafting module.

```
# Colours for dynamic editing of drafting
# text
text_cursor_colour: Pink
text_box_colour: Green
text_box_highlight_colour: Cyan
```

The code below defines the colour of the rotation tracker.

```
# Colours for the rotation tracker
#   These are the colours of the lines and
#   text of the rotation tracker controlled
#   under tools -> options -> views ->
#   rotation tracker
#
#   Resources control line and text colours
#   when displaying the world axes or a
#   workplane axes system.
#
#   Resources also control the position of
#   the axes also which defaults to
#   bottom_left, but can be one of right,
#   left, top, bottom top_right, top_left,
#   bottom_right, and bottom_left.
#
#   The attenuation resource controls how
#   much colours of line and text are reduced
#   when an axes points away from the viewer.
#
rotation_tracker_line_colour: 0.4 0.95 1.0
rotation_tracker_text_colour: 0.4 0.95 1.0
rotation_tracker_active_colour: 1.0 1.0 1.0
rotation_tracker_workplane_line_colour: 0.4 0.95 1.0
rotation_tracker_workplane_text_colour: 0.4 0.95 1.0
rotation_tracker_pos_type: bottom_left
rotation_tracker_colour_attenuation: 0.7
```

Changing the colours used by PowerSHAPE

1. Copy the following file to your home area:
2. c:\dcam\product\powershapeXXXX\sys\hci\colourscheme\default.scm

   where XXXX is your version number of PowerSHAPE and C is the disk on which PowerSHAPE is installed.
3. Open your copy of default.scm in a text editor.
4. In this file, change the colours to your requirements by simply changing the code names of the colours.

   All code names must exist in the colour table file (colour.tab) used by PowerSHAPE.

   Make sure only 16 colours are used. Also make sure that there is an empty line at the end of the file.
5. Save the file with a new name.

Using the colour scheme in your copy of default.scm

To use the colour scheme in your copy of default.scm, do one of the following:

- Put it in the folder

  c:\dcam\product\powershapeXXXX\sys\hci\colourscheme

  where XXXX is your version number of the software and C is the disk on which the software is installed.

This adds the name of the file to the **Colour Scheme** selector on the **Properties** page of the **Options** dialog.

- Set the environment variable DCAM_COLOUR_SCHEME to the path of your file (see below for details).

If you want to add your file to our set, send us your file.

Setting the DCAM_COLOUR_SCHEME environment variable

1 Click **Start** at bottom left of the screen.

2 Move the mouse to **Settings** and, from the pop-up menu, select **Control Panel** to display the **Control Panel** window.

3 In the **Control Panel** window, double click the **System** icon to display the **System Properties** dialog.

4 Click the **Environment** tab.

5 In the **Variable** text box, type:

DCAM_COLOUR_SCHEME

6 In the **Value** text box, enter the pathname to your copy of default.scm. For example,

c:\mypsfiles\default.scm

7 Click **OK**.

8 Log out and into the system again for these changes to be effective.

Setting up printers/plotters

Printers and plotters are handled by drivers supplied by their manufacturers. You must have the correct driver installed for the printer or plotter to work correctly.

Installing a printer/plotter driver

1 Click the **Start** menu (on the Windows Taskbar) and move the cursor to **Settings**.

2 From the pop up menu, select **Printers** to display the **Printers** window.

3 From this window, double click **Add Printer** to display the **Add Printer** Wizard.

4 Use the Wizard to install your driver.

Once you have set up your printers/plotters in your operating system, printers/plotters are automatically available in PowerSHAPE.

If the driver for your pen plotter is inadequate or not available, you may have to purchase WinLine (see http://www.winline.com for further details).

Changing the default settings for the printer/plotter

Your installation of PowerSHAPE should automatically use the printer that is attached to you system. However, if necessary, you can change the default settings of the printer that you use with PowerSHAPE as follows:

1 Start up PowerSHAPE.

2 From the **File** menu, select **Page Setup** to display the **Page Setup** dialog.

3 On the dialog, click the **Printer** button to display the **Print Setup** dialog.

4 Change the printer, if necessary.

5 Use this dialog to change basic properties of the selected printer such as paper size.

6 For further options, click the **Properties** button (on the **Print Setup** dialog) to display the **Document Properties** dialog to change the settings.

Graphic formats

Versions before and including the 2012 product version:

All label images were stored as *.tifs*, that could result in large model files. The 2012 version of the product lets you, optionally, store the model version 23 images in the format that will used in the next model version:

- images that originated in *.jpg* files are stored as *.jpgs* in the model file
- images that originated in lossless files such as *.tif* or *.png* are stored as *.pngs*.

To enable this behaviour, uncomment the **lossy_in_23** resource in the *powershape.con* file and restart the program.

This change results in much smaller model files. This action is only required for version 23 models created with 2012 product version or later. Later model versions will incorporate this change when creating models.

The 2011 R3 product used version 23 models; images from this version store the images in the same way as they were in model version 22.

Models using this resource that are saved using model version 23 in the 2012 product, will not be readable by previous versions. However, exporting such models to model version 22 creates psmodels that are valid for all model version 23 and later.

Models that are written using this resource can be read by the 2012 product, irrespective of the settings in the .con *file.*

PowerSHAPE 2012 and later versions:

From PowerSHAPE 2012 onwards, you can force lossless images to be stored in the .tiff format. To enable this behaviour, uncomment the **use_tiff_for_labels: true** resource in the *powershape.con* file and restart the program.

Managing your data

Use the following sections to manage your data and solve typical data management problems:

- Naming your models (see page 21)
- Defining your naming scheme (see page 24)
- Setting up model database areas (see page 29)
- Limiting access rights for models (see page 32)
- Keeping your data safe (see page 33)
- Archiving your data

Naming your models

A sensible naming scheme is the key to successful management of models. You are not obliged to use any particular naming scheme. However, we strongly recommend that you set up your system so that models can be found by user, department and/or customer project.

Use the following sections to set up naming schemes suitable for your organisation:

Parts and models (see page 23)

Naming schemes used by companies (see page 24)

Choosing a naming scheme

There is no right or wrong answer on how to choose a naming scheme. This section provides guidelines to help you decide on a naming scheme for models within your company.

1 **List the items to include in the name**.

To illustrate this, you may have decided to include the following items in your model name:

- the part represented by the model
- the revision number of the geometry
- who produced the model
- the customer for whom the model is created
- the project in which the model is developed

2 **Code the items.**

If you have many items to include in your name, you may want to use codes to identify the items.

For our example, we have decided to use:

- the part number to show the part represented by the model
- a number to show the revision number of the geometry
- the initials of the person who produced the model
- the name of the customer for whom the model is created
- the number of the project in which the model is developed

3 **Order the items**.

- Fix the order in which the items appear in the name.
- Put the unique part number of the model name at the beginning. This is helpful because all the models connected with that component are grouped together in the model name lists. Therefore, you can easily find all the models connected with that component.
- In our example, we will leave the order as it is.

4 **Use separators between the items in the name**.

If the coded items have a fixed number of characters, they need no separators. A character such as a dot must separate the coded items if they are of variable length. However, we recommend you always use dot separators as they make the names easier to read. You can use other characters, but in practice a dot is the most convenient.

In our example, the model name

531.2.rpg.perkins.2500

indicates the following:

531 - This model is associated to a part with the unique part number *531*.

2 - This is the *second* revision of this model.

rpg - This model was produced by a user whose initials are *rpg.*

Perkins - This model was designed for the customer *Perkins.*

2500 - This model was developed within project number *2500.*

5 **Inform all users of the naming scheme.**

Everyone must be aware of the naming scheme for it to be successfully used.

Parts and models

It is important to understand the distinction between a *part* and a *model*.

A *part* is a physical component that can be purchased or manufactured.

A *model* is a computer-generated representation built by PowerSHAPEto describe some aspects of the part.

Although the part can be represented by one model, it is usually represented by several models. Some of the reasons for this are:

- The inside and outside geometry is created as separate models.
- Several sub-components are individually modelled.
- Models can contain drawings and the drawings are stored as individual models.
- It is important to name your models so that you know which *part* they represent.

Naming schemes used by companies

Most companies tend to use one of two naming schemes:

1. A part naming scheme that does not begin with the project name.

   Each new part modelled by the company is given a new number sequentially. Two parts with successive numbers might be for different projects.

   In this case it is useful for the model name to contain a code (immediately following the part number) to identify the model/drawing sheet. Further fields identify the project name and the designer. For example:

   123.1.proj1.ano

2. A part naming scheme that begins with the project name.

   For example, a project to build a mold tool for the front casing of a Nokia model 310 mobile phone might be named **NK310F**. A number of models might exist to describe this mold assembly. They may be denoted by names or numbers following the model name (which are only unique within a single project). Finally the designers initials are added. For example:

   NK310F.10.ano

Defining your naming scheme

You can store permanent revisions of models by setting up the naming convention macro naming_convention.mac (see page 25). You must have a copy of this macro file in your shareddb folder for your models to save as temporary and vault models. For further details, see Temporary models (see page 25).

With the macro, models are stored as two types:

- vault models (see page 24)
- temporary models (see page 25)

Vault models

Vault models are models you want to store permanently at certain stages of your work and are stored in a *vault*. This is a specified folder on your network. Once a model is stored in a vault, you cannot change or delete it. When working with vault models, you should understand that:

- The names of the vault models comply with the naming conventions defined by you. For further details, see Creating your own macro file (see page 25).

- Each vault model has a revision number. When you save it, a copy is saved with a new revision number.
- You can open any one of the revisions of the model from the vault.
- You cannot save a vault model to itself. You must save it as a model with the next revision number.

Temporary models

The naming conventions for your temporary and vault models are set in the macro file called naming_convention.mac (see page 25).

You must have a copy of this macro file in your shareddb folder for your models to save as temporary and vault models.

An example file is available from:

C:\dcam\product\PowerSHAPEXXXX\file\pdm

where XXXX is the version number of the software and C is the drive on which the software is installed.

Use this example to create your own file. For further details, see Creating your own macro file (see page 25).

If the macro file is in your shareddb, the following dialogs change:

- **Open Model** dialog

  A button called **Search** is displayed on this dialog. Click to display the **Search for files** dialog. This dialog helps you to set up filters on the **Open Model** dialog to quickly search for temporary and vault models.

- **Save As Model** dialog

  This contains extra fields to help you name temporary and vault models.

Creating your own macro file

There are two main types of codes for the fields used to name the models.

- Field to indicate temporary or vault models (see page 27)
- Model naming field (see page 27)

Use the following example *naming_convention.mac* file to create your own macro file.

```
PDM

# These prefixes indicate whether a model is
# a temporary or vault model.
ConstFieldName = Constant
TempValue = Temp
VaultValue = Vault

# Project number is associated in the vault
FieldName = ProjectNumber
SeparatorOrFieldWidth = .
DBArea = TRUE
Revision = FALSE
VaultNamesOnly = TRUE
NonVaultNamesOnly = FALSE

# The revision number
FieldName = RevisionNumber
SeparatorOrFieldWidth = .
DBArea = FALSE
Revision = TRUE
VaultNamesOnly = TRUE
NonVaultNamesOnly = FALSE

# Who is doing the work
# This is available for only
# temporary models.
FieldName = Assignee
SeparatorOrFieldWidth = 3
DBArea = FALSE
Revision = FALSE
VaultNamesOnly = FALSE
NonVaultNamesOnly = TRUE

# The part number
FieldName = PartNumber
```

```
SeparatorOrFieldWidth = .
DBArea = FALSE
Revision = FALSE
VaultNamesOnly = FALSE
NonVaultNamesOnly = FALSE

END
```

Field to indicate temporary or vault models

The following three lines of code are used to define the field which indicates whether a model is a temporary or vault model.

```
ConstFieldName = Constant
TempValue = Temp
VaultValue = Vault
```

Each item is discussed below:

ConstFieldName - This appears in the **Search for files** and the **Save As Model** dialogs next to the text box where you input the code to indicate whether a model is a temporary or vault model.

TempValue - This is the code to indicate the model is a temporary model.

VaultValue - This is the code to indicate the model is a vault model.

Creating your own macro file

The fields used to name the models are defined using the following six values:

1. FieldName = PartName
2. SeparatorOrFieldWidth = 3.
3. DBArea = FALSE
4. Revision = FALSE
5. VaultNamesOnly = FALSE
6. NonVaultNamesOnly = FALSE

Set the six values depending on what you want the field to represent.

FieldName - This name appears next to the text box for this field in the **Search for files** and **Save As Model** dialogs.

SeparatorOrFieldWidth - You can specify the total number of characters in this field or a separator.

In the following field definition, **SeparatorOrFieldWidth** is set to *3*. Only three characters will be recognised for this field.

```
# Who is doing the work
# This is available for only
# temporary models.
FieldName = Assignee
SeparatorOrFieldWidth = 3
DBArea = FALSE
Revision = FALSE
VaultNamesOnly = FALSE
NonVaultNamesOnly = TRUE
```

In the following field definition, SeparatorOrFieldWidth is set to a dot character. The dot will automatically appear at the end of the field in the model name.

```
# The part number
FieldName = PartNumber
SeparatorOrFieldWidth = .
DBArea = FALSE
Revision = FALSE
VaultNamesOnly = FALSE
NonVaultNamesOnly = FALSE
```

DBArea - This field determines where to store model. Models with the same name in this field are stored in the same area. You should set this for vault models only, that is, set VaultNameOnly to true too. For example,

```
FieldName = Anyname
SeparatorOrFieldWidth = .
DBArea = TRUE
Revision = FALSE
VaultNamesOnly = TRUE
NonVaultNamesOnly = FALSE
```

Revision - This is the field which contains the revision. Make sure only 1 field is the revision field. You should set this for vault models only, that is, set VaultNameOnly to true too. For example,

FieldName = RevisionNumber

SeparatorOrFieldWidth = .

DBArea = FALSE

Revision = TRUE

VaultNamesOnly = TRUE

NonVaultNamesOnly = FALSE

VaultNamesOnly - This field is valid for names of vault models.

NonVaultNamesOnly - This field is valid for names of temporary models.

Set **VaultNamesOnly** and **NonVaultNamesOnly** to false if you want to use the name for both vault and temporary models.

Search patterns

If you have a naming convention for your models that uses dot separators between each component of the name, you can conduct sophisticated searches to find model names using patterns.

You can also use patterns to decide which models are stored where. For example, all model names matching the pattern **.abc* are saved in a particular database area. This may result in you having a model name which matches more than one pattern. Where a string of characters does match more than one pattern, the match is determined by the most closely matching pattern.

For example, *top.124* matches both the following patterns:

- ***1***
- ***.124**

Since *top.124* matches more characters in **.124*, that is the deciding pattern.

Setting up model database areas

Disks inevitably fill up. If the models for several users are stored centrally, it can be difficult to agree what data to move or delete when a disk becomes full. The solution is to provide a separate data area for each user. This allows each person to manage their own disk space.

Ideally, users should have their own workstation and store their data locally. This has the advantage that PowerSHAPE runs more quickly and network traffic is reduced for all users.

When using the **Store in database** option, PowerSHAPE places models with names matching specific patterns into specific areas of the network. By default, all such models are placed in the default model database area. The **Store in database** option is set on the Save Model As dialog.

Setting up a database area for each user

To set up model databases for each user, you must arrange for a naming convention for models to include the designer's initials such as **model.abc**, where **model** is the model name and **abc** is the user's initials. Create a folder on each user's hard disk and give it a suitable name. This folder is the database area and will hold the user's models.

Example

Assume that you create a user called *abc* whose workstation is called *ws1.*

If we create a folder *parts* in the shared disk *local1*:

1 You can tell PowerSHAPE to store all models with names defined by the pattern **.abc* in the folder \\ws1\local1\parts. Log on to the software as a user with administrative privileges (for example, *admin).*

2 Select **View > Window > Command** to display the command window at the bottom of the screen.

3 In the command window, type:

```
admin create dbarea
```

4 Type in the pattern of the model name you wish to store in a particular area. Make sure you put the pattern in single quotes ('), for example,

```
'*.abc'
```

5 Type in the path, again in single quotes, of where the models are to be stored. For example:

```
'\\ws1\local1\parts\'
```

All models with names ending **.abc** will be put here. Note the \ at the end of the path.

An example of this command is:

```
admin create dbarea '*.abc' '\\ws1\local1\parts\'
```

All model names ending with *.abc* are saved in \\ws1\local1\parts.

Make sure paths are of the form:

\\machine\folder\parts

and not of the form:

k:\parts

Setting up a database area for projects

Many companies manage their design work as a set of projects. A project may, for example, be a made-to-order design for a particular customer or it could be an internal project to design a new version of an existing model. These projects come into being, have a finite life span and once complete can be saved to tape and removed from disk for long-term storage.

It is often useful, when starting a project, to create a new folder on the network. All models for this project can then be stored in this folder. To minimise network traffic, place the folder on the workstation that will be used for this project.

The commands to set up a project are the same as those described in Setting up a database area for each user (see page 30) except the pattern for model name uses the project name.

Setting up projects

1. Create a folder in a suitable location. Use a meaningful name, which easily allows you to find the project. This will hold the project's PowerSHAPE models.
2. Log on to PowerSHAPE as a user with administrative privileges (for example, *admin).*
3. Select **View > Window > Command** to display the command window at the bottom of the screen.
4. In the command window, type:

   admin create dbarea
5. Type in the pattern of the model name you wish to store in a particular area. Make sure you put the pattern in single quotes ('), for example,

   '*.proj1.*'
6. Type in the path, again in single quotes, of where the models are to be stored. For example:

   '\\ws1\local1\project1\'

   All models with names matching ***.proj1.*** will be put here.

An example of this command is:

```
admin create dbarea '*.proj1.*' '\\ws1\local1\project1\'
```

All models containing the string .proj1. in their name are saved in \\ws1\local1\project1.

> *Make sure paths are of the form:*
>
> *\\machine\folder\project*
>
> *and not of the form:*
>
> *k:\project*

Moving a model from one database area to another

To move a model from one database area to another database area, you simply rename it, using **File-SaveAs**, to match the name pattern of the other database area.

When a model is completed, it can be renamed if required, using **File-SaveAs**, to move its files to a central project location.

Access rights

Access rights define the operations that a user or a group of users can use on a specific model or group of models.

For example, you might want to allow all users full access only to the models they create, but read-only access to models created by other users. For a particular combination of user and model, the access rights that can be granted are:

- create
- delete
- read
- write

Granting access rights

You can restrict how users can access the database areas by forcing them to create model names that end with their initials.

1. Log on tothe software as a user with administrative privileges (for example, *admin).*
2. Select **View > Window > Command** to display the command window at the bottom of the screen.
3. In the command window, type:

   ```
   admin create access
   ```

Type in the pattern of the model names which the user can operate on, again in single quotes. For example:

'*.abc'

4. Type in the initials of the user in single quotes. For example:

   'abc'

   You can type this string as a pattern so that it can match more than one user. If the user's initials match more than one pattern, then the access rights are determined by the most closely matching pattern.

5. Type in the user's access rights to the models. You can choose from:

   create delete read write

   If you wish no access to any models, then type:

   none

An example of this command is:

admin create access '*' 'abc' read

admin create access '*.abc' 'abc' create delete read write

User **abc** can only read everyone else's models, but can create, delete, read and write to any model ending with the name **.abc.**

Keeping your data safe

PowerSHAPE models represent a considerable investment of time and money to your company. This could easily be lost if, for example, a database area is accidentally deleted from a disk, corrupted, or if the disk is damaged by fire or mechanical failure.

Even though such events are rare, the consequences can be serious. We, therefore, strongly recommend that you protect your company by backing-up daily all your models and databases. This involves copying the shared database, parts, models and drawings to tape or other backup media. Then, if data is damaged, a recent copy can be copied back to disk.

There is no single best way to organise backing-up. In large companies, data may be backed-up as a general procedure organised by the systems department. Otherwise, backing-up should be the responsibility of the PowerSHAPE system administrator. Delcam provides back-up scripts on the program distribution DVD.

For further details, see

- Backing-up (see page 34)

- Types of backup (see page 35)
- Storing or retrieving data using backup (see page 35)
- Which folders should be backed up? (see page 36)

Backing-up

We assume that you are using some type of tape storage system, but this does not mean that you cannot use a different removable media system, for example CD. These provide benefits such as random data access, faster read write times and often a longer data life.

However, there is still the risk of physical damage, so you should always use the ten tape (cartridge) system as discussed here.

We recommend the following backup routine which keeps data for the last two weeks.

1. Label 10 tapes (which store the data) as follows:

   WEEK 1 DAY 1

   WEEK 1 DAY 2

   WEEK 1 DAY 3

   WEEK 1 DAY 4

   WEEK 1 DAY 5

   WEEK 2 DAY 1

   WEEK 2 DAY 2

   WEEK 2 DAY 3

   WEEK 2 DAY 4

   WEEK 2 DAY 5

   If you work a six or seven day week, you will need additional daily tapes to ensure every working day is covered.

2. Insert a tape in a drive. For example, if today is Wednesday, then insert the tape with label WEEK 1 DAY 3.

   Use a backup software to store your data. For further details, see Storing or retrieving data using Backup (see page 35)

   We strongly recommend that you do a full back up on a daily basis. For further details, see Types of back up (see page 35).

   For further details on what data to backup, see Which PowerSHAPE folders should be backed up? (see page 36)

3. Each day insert the next tape and backup the data.

Types of backup

There are two types of backup:

- Full

  This copies all specified data. This is the most secure type of backup because everything is copied.

- Incremental

  This only copies data that has changed since the last full backup. This uses less space than a full backup, but it can be more complicated to restore the data as changes to a single model could spread over several days of incremental backups.

We strongly recommend that you use full backup on a daily basis.

If you wish to use incremental backup, then we recommend that full backups are made every week, and incremental backups every working day that a full back-up is not done. This provides a secure system that optimises the number of tapes in use and the time required to run the backup routines.

***Warning**: If a model or drawing gets deleted or irretrievably corrupted, and this is not noticed for two weeks, it will be lost completely. If you wish to increase the "safe" time period, you must use additional sets of weekly tapes. You might also consider making full monthly backup tapes on a semi-permanent basis.*

Storing or retrieving data using Backup

Microsoft Backup is a backup program, provided as standard with your Windows XP operating systems.

To run Microsoft Backup, from the **Start** menu, select **Accessories** from the **Programs** options, then **Backup** from the **Systems Tools** options.

For more information on how to use Backup, press the Help button on the Backup window.

***Warning:** If models are open when using this software, they are not backed up. If you want to backup open models, you should purchase a backup software that allows you to do so, for example, Backup Exec (the full version of Microsoft Backup) from Veritas.*

Which folders should be backed up?

On rare occasions, models can become damaged after a power failure, loss of network connection or similar problem. To detect damaged models, we have added extra checks that are automatically made during Model Save. These checks ensure that the file saved on disk is complete.

If your company performs daily backups of your models and your model becomes damaged, you can get a copy of your model that saved on the previous day. If you lose your work during the day, that could mean several hours of lost work. The routine backup will not provide you with that.

You can protect your work further by using the backup facilities available in PowerSHAPE. This could minimise the time lost.

To recover work after a model has been accidentally deleted or has become damaged, you must have backup facilities set before the problem occurs.

If your models regularly become damaged, discuss whether to turn on the backup facilities with your local Delcam Support.

PowerSHAPE can backup your models in five ways:

1. Updating a model (see page 36)
2. Opening a model (see page 37)
3. Manually using the File Doctor (see page 38)
4. Running File Doctor when saving a model (see page 38)
5. Running File Doctor when using Model SaveAs (see page 39)

Updating a model

When a model is updated to a new model version, a backup model called *Modelname_vN.psmodel_backup* is created by default, where *N* is the model version number of the model before it was updated.

- The backup is stored by default in the temp folder, but you can set an alternative backup folder using the *update_backup_path* setting in the *powershape.con* file.

You can turn off this backup facility using one of the following:

- In the *powershape.con* file, use the setting:

  update_backup

  In the *login* macro, use:

  File OnUpdate Backup Off

The backup model does not appear in the shareddb, and can be deleted when no longer required.

Summary:

Backup Name	Modelname_vN.psmodel_backup
Folder	Temp folder
Default action	Always make the backup
powershape.con	update_backup update_backup_path
Command	File OnUpdate Backup On/Off

Opening a model

When a model is opened, you can automatically create a backup model called *Modelname_safe.psmodel*. If the model becomes unusable while working on it due to problems such as loss of network connection, you can revert to the backup copy.

The backup model is stored in the temp folder. This folder is usually on the same drive as the software.

You can turn on this backup facility using one of the following:

- In the powershape.con file, use the setting:

  open_backup

- In the login macro, use:

  File OnOpen Backup On

If your model becomes damaged, you can revert to the backup model as follows:

- Close the damaged model in PowerSHAPE.
- Select the damaged model in Windows explorer and delete it.
- Copy the safe model from the temp folder into the same folder as the damaged model.
- Rename the safe model to the name of the model just deleted.
- Open the model in PowerSHAPE.

Summary:

Backup Name	Modelname_safe.psmodel
Folder	Temp folder
Default action	Do not make the backup
powershape.con	open_backup
Command	File OnOpen Backup On/Off

Manually using File Doctor

If File Doctor is run with **Check & Fix** and the model is updated, the following three files can be automatically created:

- *Modelname_datetime.psmodel* - the model as it was the last time it was opened/saved
- *Modelname_datetime.mac* - a macro of the commands run since the last time the model was opened/saved
- *Modelname_datetime.log* - the File Doctor log

These files are stored in the temp folder. This folder is usually on the same drive as the software.

If these files are created, please send them to your local Delcam Support for analysis.

You can turn on this backup facility using one of the following:

- In the powershape.con file, use the setting:

  manual_filedoctor_backup

- In the login macro, use:

  File OnFileDoctor Backup On

Summary:

File Names	Modelname_datetime.psmodel Modelname_datetime.mac Modelname_datetime.log
Folder	Temp folder
Default action	Do not create the files
powershape.con	manual_filedoctor_backup
Command	File OnFileDoctor Backup On/Off

Running File Doctor when saving a model

If you run the File Doctor during Model Save and there are errors, three files can be automatically created:

- *Modelname_datetime.psmodel* - the model as it was the last time it was opened/saved
- *Modelname_datetime.mac* - a macro of the commands run since the last time the model was opened/saved
- *Modelname_datetime.log* - the File Doctor log

These files are stored in the temp folder. This folder is usually on the same drive as the software.

If these files are created, please send them to your local Delcam Support for analysis.

You can turn on this backup facility using one of the following:

- In the powershape.con file, use the setting:

  Save_backup

- In the login macro, use:

  File OnSave Backup On

Summary:

File Names	Modelname_datetime.psmodel Modelname_datetime.mac Modelname_datetime.log
Folder	Temp folder
Default action	Do not create the files
powershape.con	Save_backup
Command	File OnSave Backup On/Off

Running File Doctor when using Model Save As

The File Doctor is now run by default during Model Save As. If there are errors, the model can be saved as *modelname_unsafe.psmodel*, fixed by the File Doctor and resaved as *modelname.psmodel*.

The unsafe model is stored in the same folder as the model. If the model appears in the shareddb, so does the unsafe model.

Also, the following three files can be created:

- *Modelname_datetime.psmodel* - the model as it was the last time it was opened/saved
- *Modelname_datetime.mac* - a macro of the commands run since the last time the model was opened/saved
- *Modelname_datetime.log* - the File Doctor log

These three files are stored in the temp folder. This folder is usually on the same drive as the software.

If these files are created, please send them to your local Delcam Support for analysis.

You can turn on this backup facility using one of the following:

- In the powershape.con file, use the setting:

  Saveas_backup

- In the login macro, use:

File OnSaveAs Backup On

Summary:

Backup Name	Modelname_unsafe.psmodel
Backup folder	Same folder as the model
File Names	Modelname_datetime.psmodel Modelname_datetime.mac Modelname_datetime.log
Folder	Temp folder
Default action	Do not create the files
powershape.con	Saveas_backup
Command	File OnSaveAs Backup On/Off

Trouble-shooting

This section describes typical problems that can be encountered on a PowerSHAPE network and the appropriate solutions to those problems.

Recognising the symptoms

There are three classes of symptom that indicate problems with your software.

Class	Symptom
Crashing (see page 41)	Returns unexpectedly to the operating system.
Hanging (see page 41)	Being left in the program, but unable to do anything.
Error conditions (see page 44)	Shown by certain error messages. These may or may not eventually lead to crashing or hanging.

When you start up PowerSHAPE, the powershape_startup_log file is added to the tmp directory. This contains details about the checks performed when starting up PowerSHAPE. It also contains any warnings and errors. Errors or warnings are also displayed in a dialog on the screen when you start up PowerSHAPE.

Crashing

For PowerSHAPE crashing, a list of symptoms, their possible causes and solutions are given in the table below.

Symptom	Possible cause	Solution
Crash when trying to start PowerSHAPE.	Incorrect installation.	See Installation problems (see page 48)
Crashing when trying to create a new graphics window. Crashing when trying to enter a "!" to issue an operating system command within the PowerSHAPE command window.	Insufficient virtual memory.	See Virtual memory problems (see page 48).
Program crashes when performing an action on a particular model, but other models behave normally.	Damaged models	See Corrupt databases (see page 51).
Program crashes for no particular reason.	Software error.	If possible, capture any error messages displayed in the operating system command window and pass on to Delcam Software Support. See Software errors (see page 49).

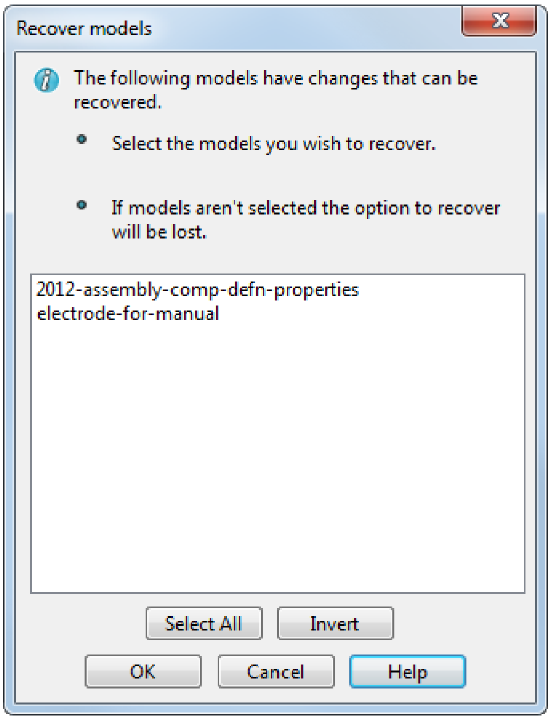
Hanging

If PowerSHAPE hangs, use the following list of symptoms, their possible causes to identify a solution to the problem:

Symptom	Possible cause	Solution
Hangs on start up.	Incorrect dates in: \dcam\product\powershapeXXXX\sys\hci where XXXX is the version number of the software..	While access rights are not the cause of the problem, the solution involves changing them. See Operating system access rights (see page 54).
	Incorrect installation.	See Installation problems (see page 48).
	Insufficient virtual memory.	See Virtual memory problems (see page 48).
Hangs when trying to create a new graphics window. Hangs when trying to enter a "!" to issue an operating system command within the command window.	Insufficient virtual memory.	See *Virtual memory problems* (see page 48).
Hangs, usually with a message in the terminal window about semaphore files.	Locked shared database tables.	See Locked shared database tables. (see page 51)
Hangs when trying to create or save a model and you do not get an error message.	Possible causes are insufficient disk space or a problem with the network.	See Network breakdowns (see page 54).
Hangs when performing an action on a particular model, but other models behave normally.	The model is damaged.	See Corrupt databases (see page 51).
Hangs for no particular reason.	Software error (see page 49).	If possible, capture any error messages displayed in the operating system command window and send to Delcam Software Support (see page 50).

Model recovery

If your PowerSHAPE session crashes, it may be possible to recover the model you were working on. When you restart PowerSHAPE, after a crash, the following dialog is displayed:



The following models have changes that can be recovered.

- Select the models you wish to recover.
- If models aren't selected the option to recover will be lost.

2012-assembly-comp-defn-properties
electrode-for-manual

1. Select the models that you wish to recover. You can use **Select All** and **Invert** to select multiple models.
2. Click one of the following:
   - **OK** to recover all selected models in the list.
   - **Cancel** to cancel the recovery process. All recoverable changes will be lost.
3. Following a model recovery, the recovered model must be saved to allow recovery from a second crash.

If your model does not contain any post-version 8 solid data (Parasolid), the models will be recovered to the command before the crash.

If your model does contain post-version 8 solid data (Parasolid), models can only be recovered to the state at which the last autosave occurred.

*The time between autosaves is defined using the **Time between autosaves** option on the **Tools>Options>File>Model** dialog.*

Error conditions

When PowerSHAPE encounters an error condition, it displays an error message in a dialog (see page 44) or in the command window. The PowerSHAPE message may be followed by a message in the operating system window, for example:

invalid argument to routine

IDTUP = 8764864

<....DBFAVA

<..DBGACH

-CLPRIN<out>

List of error messages

This section lists the error messages that may be displayed and indicates possible causes.

Error message:

`Another user is working on this model, please try later`

Possible cause	Solution
Sometimes lock files can be left in place in a model folder causing this message to appear even when no other person is using the model.	See Locked model database tables (see page 52).

Error message:

Assertion : <error>
… at line <number> of <file>

Attempt to change an implicit attribute

Attempt to get/put off end of an array

Attempt to get/put wrong type of attribute

Invalid or unset list problem

Possible cause	Solution
If the error occurs repeatedly when performing an action on a particular model, but other models behave normally, this may indicate the model is corrupted.	See Corrupt databases (see page 51). Also remember that corrupted models can also corrupt the program held in memory, so if the program does not crash you should exit and restart the program. Save your changes first (but to a new file).
Software error.	See Software errors (see page 49).

Error message:

Unable to open database file

Possible cause	Solution
There could be missing files or folders.	See Missing files or folders (see page 53).
There could be insufficient disk space.	See Insufficient disk space (see page 53).
The operation system access rights could be over restrictive.	See Operating system access rights (see page 54).

Error message:

Unrecognised database format

Out of date database format - update first

The database is corrupt

Possible cause	Solution
Could be one or more damaged database files.	See Corrupt databases (see page 51).

Error message:

Unable to open virtual memory page file

Possible cause	Solution
Operation system access rights may be preventing writes or deletes in the temporary or current folder.	See Operating system access rights (see page 54).

Error message:

Attempt to read/write an invalid block number

Possible cause	Solution
One or more database tables are probably corrupted. This could have happened some time ago or just before the message was raised.	See Corrupt databases (see page 51).

Error message:

Attempt to update in read mode ignored

Cannot find table …

Unknown table of type …

Possible cause	Solution
These indicate that there is probably a serious program error.	See Software errors (see page 49).

Error message:

Database table is open for read by another user - please wait

Database table is open for write by another user - please wait

These messages may repeat.

Possible cause	Solution
Some shared database tables are locked.	See Locked shared database tables (see page 51).
There are over-restrictive operating system access rights in the shared database area.	See Operating system access rights (see page 54).
There is inadequate disk space in the shared database area.	See Insufficient disk space (see page 53).

Error message:

Unable to create part database file

Unable to create model database file

Unable to create drawing database file

Possible cause	Solution
A network problem.	See Network breakdowns (see page 54).
The operating system access rights are over-restrictive in the model database area.	See Operating system access rights.
Inadequate disk space in the shared database area.	See Insufficient disk space (see page 53).

Error message:

Cannot find the details file

Unable to find database files

Possible cause	Solution
There is a network problem.	See Network breakdowns (see page 54).
The required files or folders are missing.	See Missing files or folders (see page 53).
The operating system access rights are over-restrictive in the model database area.	See Operating system access rights.
Inadequate disk space in the model database area.	See Insufficient disk space (see page 53).

Error message:

Unable to open file <filename>

Unable to open part database file

Unable to open model database file

Unable to open drawing database file

Possible cause	Solution
A network problem.	See Network breakdowns (see page 54).

The required files or folders are missing.	See Missing files or folders (see page 53).
The operating system access rights are over-restrictive in the model database area.	See Operating system access rights.

Solving typical problems

Use the following sections to identify and solve typical problems:

Recognising the symptoms (see page 40)

Installation problems (see page 48)

Virtual memory problem (see page 48)

Software errors (see page 49)

Corrupt databases (see page 51)

Locked shared database tables (see page 51)

Locked model database tables (see page 52)

Missing files or folders (see page 53)

Insufficient disk space (see page 53)

Network breakdowns (see page 54)

Operating system access rights (see page 54)

Installation problems

Installation problems may happen if you the copy the software files manually, rather than using the installation program.

The installation program completes additional processes, such as setting up registry entries, and should always be used in preference to manual copying of files.

1. Uninstall the software
2. Re-install it using the installation procedure provided with your downloaded files or on your DVD.

Virtual memory problem

Windows keeps a temporary copy of every program you are running, together with space for its data, in a special temporary file. It uses this file to swap information in and out of your computer's physical memory. This temporary file is referred to as 'virtual memory'.

Windows normally manages your virtual memory files automatically, but needs available physical disk space to do this. Your virtual memory will always be limited to the amount of physical disk space available on the computer's local disk. You must ensure that you have enough disk space available to hold the memory 'image' of PowerSHAPE plus its data, before starting the program. A good guide is around 500Mb of free disk.

Windows can split virtual memory over local disks so that one running program image is stored on one disk, and another program is stored on a second disk. However, if a program starts on one disk and needs to expand in size, it must stay on the same disk. Thus you can find situations where virtual memory becomes exceeded (on one disk) whilst a lot of space still exists on another disk.

If you have problems that indicate possible exhaustion of virtual memory, check how you computer's virtual memory is configured (see Microsoft's documentation) and check for availability of disk space on all disks.

Software errors

Software errors are caused by faults in the PowerSHAPE program code. Delcam is extremely careful to only release reliable software. However, on a program this complex, problems caused by code errors are inevitable, but hopefully rare events.

Although they are (usually) rare occurrences, you should never ignore program errors. In many cases, PowerSHAPE will continue operating as normal. However, the program (held in memory) might be damaged and causing corruption in your model. We recommend that you should log out immediately without saving your model.

If the program hangs, you will not be able to exit normally. In this case, use the following technique to exit the program:

1. Press Alt Ctrl Del.
2. Select **Task Manager** to display the **Task Manager** window.
3. Click the **Applications** tab.
4. Select **PowerSHAPE** from the list of tasks.
5. Click **End Task**.

When you restart PowerSHAPE, you run an undamaged copy of the software. If you have not saved the model since the problem began the stored version is undamaged. Unfortunately, you will now have to repeat any unsaved work.

If you suspect a model may be damaged:

1. Open the model in PowerSHAPE
2. Use the **Tools > File Doctor**.

   This detects and repairs most types of model corruption.

*If logging out without saving means that you might lose much work, you could always save the model under a new name (**Save as** in the File menu) and then run the **File Doctor** on the new version. This allows you to keep your most recent work, without damaging work you know to be clean.*

If **File Doctor** finds and corrects any faults or finds no faults, you can continue working. In these situations, we recommend that you regularly save your model under a new name each time. If any model turns out to be corrupt, you can always return to an older version.

Reporting program errors

It may be possible to work-around the problem in the short-term, but you should report all program errors to Delcam Customer Support or your local Delcam Sales Partner. They will try to ensure that any error can be removed from future release versions of the software..

When reporting errors, you should send as much information as possible:

- Send the error messages that appeared on your screen. Error messages, sych as the one shown below, help software developers pin-point a problem exactly.

  ```
  Assertion: <message> in <file> line <line number>
  ```

- Send a sequence of commands that will duplicate the problem and the models you were working on. The more information you send, the more likely it is that the developers can find and fix the problem.

Reporting program errors

You should report all program errors to Delcam Customer Support or your local Delcam Sales Partner if applicable. They are often able to show you how to achieve the same results by a different route which "works around" the problem. Delcam will always try to ensure the error is removed from future release versions of PowerSHAPE.

When reporting errors the following will help identify the source of the error:

- Always send the error messages that appeared on your screen. Messages of the form **Assertion: <message> in <file> line <line number>** pin-point the problem exactly to the software developers.
- Send a sequence of commands (macro) that will duplicate the problem and the models you were using. The more information you send, the more likely it is that the developers can find and fix the problem.

  If you send a macro that crashes the program when run, Delcam will make their best efforts to fix the fault within the next two releases.
- The *.dmp* files that are created in the *temp* directory if PowerSHAPE crashes are named using the following format:

  PowerSHAPEXXXXXx32_yyyymmdd-hhmmss.dmp (32-bit PowerSHAPE)

  *PowerSHAPEXXXXXx64_yyyymmdd-hhmmss.dmp(*64-bit PowerSHAPE)

  Please send these files to Delcam support when reporting problems.

Corrupt databases

This problem can be caused by a program error or a lack of disk space when modifying the database or perhaps a human error when manually copying folders.

The **File Doctor** in the Tools menu can detect and repair most types of corruption, both in the model and the shared database files. This is mostly done by deleting the corrupt data so sometimes even if the corruption is fixed, there can be a significant loss of data. In such a case, you will need to restore the affected model or shared database from the backups. For further details, see Keeping your data safe (see page 33).

Locked shared database tables

Lock files (called semaphores) are written and removed when users access the database. The purpose is to prevent more than one user from accessing the same files at the same time. Sometimes however, lock files can be left in the database or locked unnecessarily by the operating system.

To clear locks:

1. Ask all users to log out of PowerSHAPE.
2. Type the following in the operating system command window:

check_dblocks -delete

This deletes unwanted semaphore files and operating system locks in the shared database areas.

It is always a good idea to investigate why the locks are still in place. It may be that a user has crashed out of PowerSHAPE or perhaps someone has found a program error. If either is true you should investigate further. For further details, see Software errors (see page 49).

Locked model database tables

This problem can be caused by an unwanted lock file in a model folder or that another user is unnecessarily locking the model.

To clear locks:

1. Ask all users to log out of PowerSHAPE.
2. Type the following in the operating system command window:

   check_dblocks -models -delete

Using this method can take a long time as it unlocks all the files and folders in the model database areas.

If you ar an experienced user of the operating system, us the following method:

1. Enter the following command in the PowerSHAPE comand window:

   admin state dbarea

   This displays the pathname of the offending model/drawing folder.
2. Check that no users have this model open.
3. Delete the lock file as follows:
   a. Use Windows Explorer to find the locked folder.
   b. Select the lock file and press the Delete key or select Delete from the File menu.

As with all problems, it is always a good idea to investigate why the locks are still in place. It may be that a user has crashed out of PowerSHAPE or perhaps someone has found a program error. If either is true, you should investigate further. For further details, see Software errors (see page 49).

Missing files or folders

The most likely problem is that files from a particular model are missing. In which case, restore a complete version from the back-up tapes.

There are several reasons why files or folders may not be present:

- The files or folders may have been removed manually. You should actively discourage users from working in folders containing database files in case they accidentally delete the files. Only expert support staff should deliberately delete files in the database folders.
- A relative pathname was used when creating a new database area. This means that the folder can be accessed from where it was created, but from nowhere else. Always use the full pathname, for example, \\ws1\disk\models\
- A database area may have been created, but the corresponding folder, might not have been created or it was subsequently removed.
- Files might be missing if at any stage PowerSHAPE had issued a message that it was unable to create them. This could be due to access rights, insufficient disk space or due to other missing folders (as explained above).

Insufficient disk space

As you create new models, you will inevitably fill up your data storage disks. However, on a network installation, you will find that some disks fill up more quickly than others. If you have errors caused by lack of space, check all the disks on the network and attempt to find which has caused the problem. Bear in mind that files are created in the following areas:

- the model and drawing database areas
- the shared database area
- the temporary and current folders

The following are possible solutions:

- Delete temporary files. These start with the characters *'w_'*. Make sure all PowerSHAPE users exit the program first.
- Archive some of the models.
- Create some new database areas to redistribute the available space.
- Add additional storage hardware to your network and create some new database areas.

Network breakdowns

Network breakdowns are usually caused by:

- disconnected cables
- failure of the network server program
- crashing of the network server program
- crashing of a connected workstation

Open the Windows Explorer and select each workstation. A warning is raised for those that cannot be reached. If you cannot see all the workstations, assign another drive to the missing one(s). If the drive cannot be assigned, check the network cables are connected properly to that workstation. If this does not correct the fault, reboot the workstation and try again. If the fault persists, contact your Delcam Sales Partner or Delcam Customer Support.

Operating system access rights

On any system, it is normal to have different access rights in the various parts of the system. The table below shows the normal access rights for the PowerSHAPE:

Files/folders	Access rights
The contents of \dcam\config\	**read** and **execute** rights.
The contents of \dcam\exec\	**read** and **execute** rights.
The contents of all database areas and the temporary (or current) folders.	**read**, **write** and **execute** rights.
All folders in the product tree.	**read-only** access rights.

A shared database table consists of three files:

table.db

table.sh

table.pf

If a table is correctly accessed, the dates of the files show that the .db file is accessed most recently, followed by the .pf file, followed by the .sh file.

Dates/times in a different order indicate a problem and PowerSHAPE attempts to regenerate the files to correct the fault. To do this the program needs read and write access to all three files, otherwise it will hang as you try to enter the program.

The files in the shared database area should already have suitable access, but database tables also exist in the PowerSHAPE product tree which normally has read-only access rights. The files are in

\dcam\product\powershape<version>\sys\hci

and all the sub-folders below this.

If you have a problem with the program hanging on start-up, provide **read**, **write** and **execute** rights to all files below

\dcam\product\powershape<version>\sys\hci

The next time PowerSHAPE is run, it will correct the files and the hanging problem should correct itself. You should then return the access rights to read-only.

This problem results either from an incorrect installation or from subsequent 'tampering' with the installed files. Always use the installation procedure provided.

Commands and scripts

This section contains the commands and scripts to manage your PowerSHAPE system. It is split into two sections:

- PowerSHAPE commands (see page 55)
- Script commands

Commands

Commands are entered in thecommand window.

1 Select **View > Window > Command** to display the command window at the bottom of the screen. The command window is displayed at the bottom of the screen.

2 Type the commands in the command window

For further details, see:

Access rights (see page 56)

Model database areas (see page 58)

Users (see page 61)

Access rights

Sites that have several departments using PowerSHAPE and sharing the same database, or sites where models are confidential, may wish to create access restrictions to prevent certain users from accessing or deleting models created by other users. This task is the responsibility of the System Administrator, and is done using the **Create Access** and **Delete Access** commands.

In addition to the more obvious inter-user access restrictions, it may be useful to restrict access in such a way that once a model is complete, it cannot be deleted by anybody until the access restriction is removed (to avoid accidental deletion). Similarly models could be protected from write access once they have been approved, to ensure that each revision of a model thereafter can only be done by copying the model and altering the copy (thus ensuring that all revisions of a model are maintained).

For further details, see:

- Actions on Access (see page 56)
- Create Access (see page 56)
- Delete Access (see page 57)
- State Access (see page 58)

Actions on Access

The following actions may be performed on **Access**:

Create Access

Delete Access

State Access

Create Access

This creates an access restriction for users matching user_name_pattern accessing models matching model_name_pattern.

In PowerSHAPE's command window, type:

ADMIN CREATE ACCESS 'model_name_pattern' 'user_name_pattern'

and then type in the access rights. Choose from:

- **CREATE**
- **DELETE**
- **READ**
- **WRITE**

These access rights may be specified in any order.

If no access is required, type:

```
NONE
```

For further details on patterns, see Search patterns (see page 29)

When PowerSHAPE is first installed, all users have full access to all models.

To give users only limited access to models, first prevent all users from accessing any model by typing:

```
ADMIN CREATE ACCESS '*' '*' NONE
```

The rights to access specific groups of models can be given to specific groups of users, for example

```
ADMIN CREATE ACCESS 'SS' '*.SS' CREATE READ WRITE
```

When a user-model access check matches more than one pattern the access rights are determined by the most closely matching pattern. (For example, the highest number of exact character matches and ? wildcard matches in model_name_pattern and user_name_pattern).

Delete Access

This deletes an access restriction created by the **CREATE ACCESS** command.

In the PowerSHAPE command window, type:

```
ADMIN DELETE ACCESS 'model_name_pattern' 'user_name_pattern'
```

The exact model_name_pattern and user_name_pattern used to create the access restriction must be given to delete it. If the patterns are not given on the same line as **Delete Access**, you are prompted for them in turn.

For further details on patterns, see Search patterns (see page 29)

An example of a complete **Create - Delete Access** sequence:

```
... > ADMIN CREATE ACCESS
Pattern matching models to be protected  > 'me*'
Pattern matching users to be allowed access > 'ano*'
Access rights (CREATE/DELETE/READ/WRITE) > CREATE
DELETE
...
...
... OTHER COMMANDS
...
... > ADMIN DELETE ACCESS
```

```
Model name pattern (as per STATE ACCESS) > 'me*'
User name pattern (as per STATE ACCESS) > 'ano*'
```

State Access

This displays access rights in general or for specific models.

In the command window, type:

```
ADMIN STATE ACCESS ALL
```

to display the access rights for all the models. This is the same as typing:

```
ADMIN STATE ACCESS
```

To display the access rights of those model names that match a pattern, type:

```
ADMIN STATE ACCESS 'model_name_pattern'
```

For further details on patterns, see Search patterns (see page 29)

State Access outputs the restrictions:

- set up by **Create Access**
- on the models specified in the command

For example, typing:

```
ADMIN STATE ACCESS 'fig*'
```

produces an output similar to:

```
Access restrictions
Model patterns Users Create Delete Read Write
FIG*          *     Yes    No     Yes  No
SYM*          RPG   Yes    No     Yes  Yes


Model names Delete Read Write
Fig1        No     Yes  No
Fig2        No     Yes  No
```

In this example any user creating a model called FIG followed by any other characters will give that model read-only access. If user RPG creates a model SYM23 it cannot be deleted.

Model database areas

For details on model database areas, select from the following:

- Actions on Model (see page 59)

- Archive Model (see page 59)
- Retrieve Model (see page 59)
- Print Model (see page 60)
- State Model (see page 61)

Actions on Model

The following actions may be performed on **Model**:

- **Archive Model**
- **Retrieve Model**
- **Print Model**
- **State Model**

Archive Model

This requests a model for archive or cancels the request.

To request a single model for archiving, in the command window type:

ADMIN ARCHIVE MODEL model_name

A file called **doc.put** is added to the directory of the named model. This file indicates that the model is requested for archiving.

To request a pattern of model names for archiving, in the command window type:

ADMIN ARCHIVE MODEL 'model_name_pattern'

All model names matching the pattern are given an archive request.

For further details on patterns, see Search patterns (see page 29)

To delete an archive request, in the command window type:

ADMIN ARCHIVE MODEL CANCEL model_name

For a pattern of model names, type:

ADMIN ARCHIVE MODEL CANCEL 'model_name_pattern'

If a model has an archive request recorded when the archiving script **archive_put**is run, it is transferred to tape for long term storage, freeing the disk space it once occupied.

Use **State Model** to list models; the output displayed shows whether they have been archived or not, and gives their archive status.

Retrieve Model

This requests a model for retrieval from archive or cancels the request.

To request a single model for retrieval, in the command window type:

```
ADMIN RETRIEVE MODEL model_name
```

A file called **doc.get** is added to the directory of the named model. This file indicates that the model is requested for retrieval.

To request a pattern of model names for retrieval, in the command window type:

```
ADMIN RETRIEVE MODEL 'model_name_pattern'
```

The request is added to a list of archive requests which is processed by the System Administrator.

For further details on patterns, see Search patterns (see page 29).

To delete a retrieval request, in the command window type:

```
ADMIN RETRIEVE MODEL CANCEL model_name
```

For a pattern of model names, type:

```
ADMIN RETRIEVE MODEL CANCEL 'model_name_pattern'
```

If a model has an archive retrieval request recorded when the archiving script **archive_get** is run, it will be transferred from tape to disk.

Use **State Model** to list models; the screen display shows whether they have been archived or not, and gives their archive status.

Print Model

This prints detailed information about the named model model_name.

In the command window, type:

```
ADMIN PRINT MODEL model_name password
```

Any password given to the model when it was created must be given after the model name.

The information printed out is the name of the model, any description, password, and standard given when the model was created. Also included in the printout are the models minimum and maximum size, and if it is archived.

The format of the displayed table is:

```
Model details

============

Name Description Std Min corner Max corner Archived Password

TB1  Test Piece  BSI -10 -10 -10 10 10 10  No       none
```

State Model

This display information about one, some or all models.

In the command window, type:

```
ADMIN STATE MODEL ALL
```

to print information about all the models. This is the same as typing:

```
ADMIN STATE MODEL
```

To display the information of models matching a pattern, type:

```
ADMIN STATE MODEL 'model_name_pattern'
```

A list of the models in the database is displayed on the screen, together with any description given when the model was created and whether the model has been archived.

The information is output in a table format as shown below :-

```
Model details

============

Name          Description                 Archived

AutoPart       A description of auto part        No

ARL                                    Yes

DEMO2          Zylinderkopf                    No

 ...         ...                    ...

 ...         ...                    ...

 ...         ...                    ...
```

For further details on patterns, see Search patterns (see page 29).

Users

PowerSHAPE maintains a list of users who are allowed to login to the program. The System Administrator may update this list and allocate a security rating to each user to control access to data. Each user may have an individual password.

When PowerSHAPE starts up, it checks the name of the user currently logged into the host computer. If this user name is not already known then PowerSHAPE will ask for a login name. Your System Administrator may create a new user at this point.

Security ratings are used to control access to model information and to other user's account information. For example, **Print User** and **State User**print the passwords of users who have lower security ratings than that of the user who is logged in.

For further details on model database areas, select from the following:

- Actions on User (see page 62)
- Create User (see page 62)
- Delete User (see page 62)
- Edit User (see page 63)
- Print User (see page 63)
- State User (see page 63)

Actions on User

The following actions may be performed on **User**:

- **Create User**
- **Delete User**
- **Edit User**
- **Print User**
- **State User**

Create User

This creates a new user in the PowerSHAPE user database.

In PowerSHAPE's command window, type:

```
ADMIN CREATE USER user_name password full_name
security_rating config_file_name
```

If no password, or config file, is required the word NONE must be typed.

Only a security rating equal to, or lower, than the current user's is accepted.

Delete User

This deletes the user user_name from the list of users allowed to use PowerSHAPE.

In the PowerSHAPE command window, type:

```
ADMIN DELETE USER user_name password
```

If a password is defined then you must enter it.

Edit User

This edits the user's password.

In PowerSHAPE's command window, type:

```
ADMIN EDIT USER user_name PASSWORD new_password
```

Print User

This prints the details about a user.

In the command window, type:

```
ADMIN PRINT USER user_name password
```

If you enter *ME* as the *user_name*, then the details are printed of the user who is currently logged in.

State User

This prints the user database in table format.

In the command window, type:

```
ADMIN STATE USER
```

An example of the print out is given below.

```
User details
Login name  Password   Full name       Security level
RPG                    Razia Ghani     Sys.  Admin
KBH         ninka      Karen Heesom    Draughtsperson
```

Passwords are only printed if the user is a System Administrator.