

Red Box Server Administration (UNIX Systems)

Copyright notice

Red Box Server Administration (UNIX Systems), M0060, Edition 2, February 2000. Red Box release 3.

This document is the copyright material of Ultracomp Ltd. It may not be copied without prior consent, in writing, from Ultracomp Ltd.

Ultracomp Ltd endeavours to ensure that the information in this document is correct, but does not accept liability for any error or omission. However, Ultracomp Ltd would be pleased to receive readers' views on the contents of this document.

The products described in this document are subject to licence agreements which govern their use. Statements in this document are not part of any licence or contract save insofar as they are incorporated into a licence or contract by express agreement. Issue of this document does not imply any entitlement to use of or access to any or all of the products or facilities it describes.

About this guide

This guide describes the components of the Red Box UNIX server software, and gives procedures for startup, shutdown, and regular maintenance, including backup and restore.

Chapter 10 "Troubleshooting" on page 75 describes problems that could stop users being able to connect to the Red Box server software. If you have a Red Box problem that is not covered in this chapter, please contact the Ultracomp Support Centre.

Technical Support

Make sure that you have as much information as possible to hand before contacting support.

General Enquiries

Ultracomp Limited
Ultracomp House
Pinehill Road
Crowthorne
Berkshire
RG45 7JD

Tel: +44 (0) 1344 779333

Fax: +44 (0) 1344 779385

Ultracomp Support Centre

Tel: +44 (0) 118 973 3456

E-mail

support@ultracomp.co.uk

Internet

<http://www.ultracomp.co.uk>

Contents

Chapter 1	Introduction	9
	Server software	9
	The Red Box database	9
	The batch utility	10
	Read-only views	10
	PC client	10
	Operations Control	10
Chapter 2	Red Box Processes	13
	Master server (listener)	13
	Periodic	14
	Delayed update process	15
	Oracle processes	15
Chapter 3	Red Box Files	17
	Configuration file	17
	Initialisation files	17
	Red Box initialisation file	17
	Oracle7 initialisation file	18
	Database files	18
	Oracle7 tablespaces	18
	Redo logs	18
	Control files	19
	Database file placement	19
	Log and trace files	20
	Oracle7 log files	21
	Red Box log and trace files	21
	redbox.log file	22
	Identifying trace files	23
Chapter 4	Red Box UNIX Users	25
	Key to figure 4.1	27
Chapter 5	Running Red Box	29
	Automatic startup and shutdown	29
	Manual startup and shutdown	30

	Manual startup	30
	Manual shutdown	30
	Checking for Red Box processes	31
	To check the master server	33
	To list Oracle processes	34
Chapter 6	Red Box Backups	35
	Red Box backup strategy	35
	UNIX backups	36
	Red Box backups	36
	Backup cycles	38
	Automatic archiving	38
	Running the backup commands	39
	Display during processing	40
	Backup sequence numbers	40
	Listing the contents of a backup tape	40
	Backup log files	41
	Backup.log files	41
	Backup dump files	42
	Handling ucrbbckonline failures	42
Chapter 7	Recovery	45
	Preparation	46
	Stage 1: Initial restore	47
	Restoring a UNIX backup	47
	Restoring a full Red Box backup (ucrbckfull)	48
	To give full permissions to create the oradata/ucrb directories	48
	To run a full restore	49
	To restore the Email process	49
	Stage 2: Restoring latest copies of files	50
	To run ucrbbckrestore	52
	To copy control files	54
	Stage 3: Database recovery	55
	Stage 4: Take a full offline backup	57
Chapter 8	Database Maintenance	59
	Database integrity (ucrbverify)	59
	Space allocation	60

	Tablespaces that may need to extend	60
	Monitoring free space (ucrbSPACE)	60
	Identifying tables or indexes that cannot extend (ucrbanalyze and ucrbtabre- port)	62
	ucrbanalyze	62
	ucrbtabreport	62
	Extending a tablespace (ucrbextend)	64
	Emptying the archive directory (ucrbtidylog)	65
	Moving the database (ucrbexport)	65
	Importing an exported database	66
	Running the commands	66
Chapter 9	Applying amendments	69
	Summary	69
	Applying the amendment	70
	Step 1: Shut down the Red Box server	70
	Step 2: Take a full backup	71
	Step 3: Apply the amendment	71
	Step 4: Restart the server	72
	Step 5: Remove temporary files	72
	Reversion	72
Chapter 10	Troubleshooting	75
	Connection failures	75
	All PCs lock, or all new connections fail	76
	A single PC cannot connect to the server	76
	File system full errors	77
Appendix A	Directory Structures	79
Appendix B	redbox.cfg	83
	Index	87

Introduction

This chapter describes the main server components of a Red Box system, which include:

- Server software that runs on a UNIX host
- A PC client, which is a Windows application
- Operations Control managers and agents (if the Red Box system includes Operations Control).

The rest of this guide describes the administration of the server software. For Red Box PC administration and Operations Control administration, refer to the *Red Box Administration Application Guide* and the *Red Box Operations Control Guide*.

Server software

The Red Box server software runs on a UNIX host that is accessed from the PC client. The main server software components are:

The Red Box database

The core feature of Red Box is its centralised Oracle7 database, known as the *Red Box database*. The database files contain records of all the components that are controlled and managed across all Red Box applications. For example, the database holds configuration items (CIs) created in Configuration Management, incidents created in the Help Desk, and events and actions created in Operations Control.

Users access the database by logging in to Red Box on PCs that have access to the server, and use the Red Box applications to view, insert and amend records. There are other Red Box facilities for batch updates and read-only views; these are described in the next two sections.

The batch utility

Use the Red Box batch utility, **ucrbbatch**, to insert and update large quantities of database records. You can use **ucrbbatch** to populate the database initially, and subsequently for ad hoc or regular transfers of data with other systems. The *Red Box Database Guide* describes **ucrbbatch**.

Read-only views

Read-only views allow direct read-only access to Red Box database records, held in database tables. You can use the views from Oracle7-compliant Windows applications that use SQL*Net or ODBC drivers. In the server, you can use the views via standard Oracle7 interfaces. The *Red Box Database Guide* describes read-only views.

PC client

The Red Box PC client is a Windows application from which Red Box users load Red Box applications and view and update database records. Each Red Box application supports a service management function, for example, Configuration Management, Help Desk, Change Management. The server software must be running in order for users to run the PC client.

Operations Control

Operations Control manages conditions reported by monitoring software. Depending on the type of monitoring software, these conditions may include alerts, events, network traps, and so on. Operations Control records each condition in an event record on the database, and initiates appropriate actions.

The Operations Control components are:

- **Agents.** These are instances of the process **ucrbagent**. An agent runs on a system being monitored and passes to its manager all conditions reported by the monitoring software. There are different types of agent for different types of system (UNIX and Windows).
- **Managers.** These are instances of the process **ucrbmanager**. One manager may control any number and type of agents. It may run on the same system as an agent, or on a different system (usually, it runs on the Red Box server).

A manager reads from and writes to the Red Box database, sending all relevant information to its agents. The server software must be running in order for the manager to access the database. However, managers are resilient to loss of the server and will run without it. In resilient mode, they postpone all database updates until the server connection is re-established.

- A Red Box Operations Control application, loaded from the PC client. Use it to set up and monitor the Operations Control system.

Red Box Processes

When the server software is running, a number of Red Box and Oracle processes start on the UNIX server. Some are permanent processes that run until the server software stops. These are:

- Master server (listener)
- Periodic
- Delayed update process (runs at regular intervals)
- Oracle processes.

In addition, for each successful database connection, the master server starts a pair of transient processes to service that connection.

This chapter describes the permanent processes and the transient processes that the master server starts. Also refer to [Figure 4.1 on page 26](#) which shows the relationship between the processes.

To check that the correct processes are running when Red Box is loaded, refer to the section [“Checking for Red Box processes” on page 31](#).

Master server (listener)

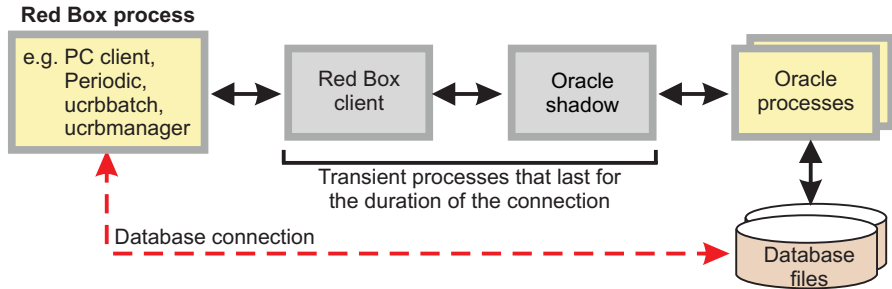
The master server (also known as the listener) listens for connection requests from processes that need to access the database. These processes include:

- PC clients. Each PC Red Box session requires one connection, from the display of the Red Box login screen and throughout the entire logged-in session until the user logs out, regardless of the number of Red Box applications that are loaded. A suspended Red Box session (which displays message counts only) also requires one connection.
- The Periodic process (described below).
- The delayed update process (described below).
- Each instance of **ucrbbatch**.
- Each instance of **ucrbmanager**.

For each successful connection, the master server starts a child process (a copy of itself), called a Red Box client. The client process in turn starts an Oracle shadow process. This pair of processes start when the connection request is successful and stops when the connection terminates.

Suspending a PC client session releases the shadow process; logging back in again starts another shadow process.

Figure 2.1 Access to Red Box database via client and shadow processes



The master server and its Red Box clients update the Red Box log file (**redbox.log**). For details, see the section “[redbox.log file](#)” on page 22.

Periodic

The Periodic process carries out regular processing required by PC clients, such as incident escalations and message distribution. The Red Box Administration application, loaded from the PC client, has options for controlling the frequency of Periodic processing.

Delayed update process

This process runs at regular intervals, defined in the Red Box configuration file (**redbox.cfg**) described in [Appendix B on page 83](#). The delayed update process:

- Checks for records whose names have been changed, and updates the amended names in records in which they are referenced.

- Checks for lost and timed out connections, and recovers slots for them. The *Red Box Administration Application Guide* describes slot usage.

Oracle processes

There are six Oracle processes that manage the database files and their online redo log and archived redo log files. The Oracle processes are:

ora_arch_ucrb
ora_lgwr_ucrb
ora_pmon_ucrb
ora_smon_ucrb
ora_dbwr_ucrb
ora_reco_ucrb

The section [“To list Oracle processes” on page 34](#) describes how to check that these processes are running.

Red Box Files

This chapter describes the main Red Box files, which include configuration, initialisation, database, log and trace files. It gives the file locations under their Red Box UNIX user names (**rboxsw**, **rboxdba** and **rboxuser**); these are described in [Chapter 4 on page 25](#).

Configuration file

The Red Box configuration file, **redbox.cfg**, is held in the **rboxsw** user's home directory, under the sub-directory **redbox/3.00**. The file is fully described in [Appendix B on page 83](#). Important values are:

- The port number of the server. For version 3.00 of the server software, the default value is 7300.
- The check interval and idle count, which among other things determine how frequently Red Box checks its PC client connections and reports connection failures.

Initialisation files

Each Red Box PC client uses an initialisation file **redbox.ini**, and Oracle7 uses the initialisation file **initucrb.ora**. These are described in the next two sections.

Red Box initialisation file

Each PC client maintains a **redbox.ini** file in its Red Box directory. The **redbox.ini** file holds screen colour, printer, caching and other options for the PC, and is fully described in the *Red Box Administration Application Guide*. Important values are the Service Name, Port, and Timer Frequency, as described on [page 76](#).

Oracle7 initialisation file

The initialisation file **initucrb.ora** contains a description of the Oracle7 system, for example, the maximum number of Oracle processes and the locations of Oracle files. Oracle reads but does not update the initialisation file. The only value that you may want to change is the Processes value, which specifies the maximum number of database connections allowed. For information on calculating and changing the Processes value, refer to “Oracle processes” section in the “Groups and Slots” chapter of the *Red Box Administration Application Guide*.

Database files

These files hold all the database information, and include:

- Oracle7 tablespaces
- Redo logs
- Control files.

Oracle7 tablespaces

Red Box creates tables for holding Red Box records, and creates indexes for accessing the data in its tables. Oracle7 holds each table and index in a separate tablespace, where each tablespace is supported by a database file of a fixed size which is specified during installation. You can increase the size of a tablespace by adding a database file, and this may be necessary for the two tablespaces that hold user data and user indexes. You should follow the procedures in the section “[Space allocation](#)” on page 60 to check for free blocks in these tablespaces and to add new files as necessary.

Redo logs

Red Box updates three online redo log files (**redo01.rdo**, **redo02.rdo**, **redo03.rdo**) with all transactions (record creations, amendments, and so on) that can be re-applied to the database in the event of failure.

When an online redo log file is full, Red Box writes its contents to an archived redo log file. Archived redo log files are numbered sequentially (**rb_log_1**, **rb_log_2**, and so on), and accumulate on the file system until you run one of the backups **ucrbckoffline** or **ucrbckonline**, described in [Chapter 6 on page 35](#). These backups delete all archived redo log files after securing them.

On a daily basis, you should monitor the file system space used by the archived redo log files, and if necessary, increase the frequency of your backups to free up space more regularly.

Control files

A control file is essential for recovery after a file system failure, because it lists the online redo log files and archived redo log files that need to be re-applied to the database. Each file system that contains database files also contains a control file, named **cntrlrbox.ctl**. Red Box maintains its control files simultaneously across all file systems, so they contain identical information. If you lose the control file on one file system, you can use the control file on another file system to recover the database.

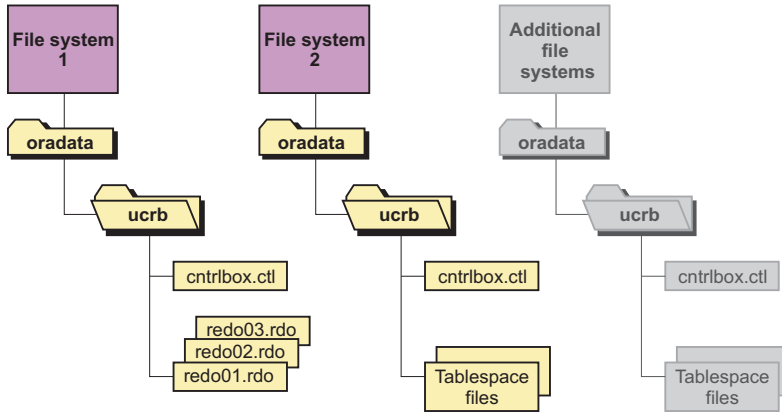
Database file placement

If your UNIX system has more than one disk drive, you should spread the Red Box database files correctly over at least two disks so that if one disk fails, you do not lose data.

The default installation uses two file systems, and places the redo logs on a separate file system to the Oracle7 tablespace files. This is the correct organisation of database files and is the basis of Red Box security strategy. If you use more than two file systems during installation, the directory structure for each additional file system is the same as for file system 2.

The following diagram shows the installation of database files across two or more file systems.

Figure 3.1 Placement of database files across file systems



If you choose to place all the Red Box files on a single file system (not recommended), all files will be held in the directory `oradata/ucrb`, and you will have a single control file.

Log and trace files

If a problem occurs during the startup, closedown or running of the Red Box system, you can check the relevant log file for error messages. You may also need to check trace files. You should report any error messages to the Ultracomp Support Centre or your product distributor. For details, see the section [“Technical Support” on page iii](#). Take copies of the relevant log files and retain them as evidence.



Note: Trace files contain all messages, not just those reporting fatal errors. Some messages simply report database constraints; for example, the message “unique constraint violated” indicates that a user has attempted to create a record with a duplicate name. You should ignore these messages.

Oracle7 log files

Oracle7 maintains independent startup, closedown and runtime log files.

Table 3.1 Oracle7 log files

File name	Description	Location
S88dbstart.log K50dbshut.log	Startup log (automatic startup). Closedown log (automatic closedown).	rboxdba user's home directory.
dbstart.log dbshut.log	Startup log (manual and automatic startup). Closedown log (manual and automatic closedown).	
alert_ucrb.log	Runtime log. You can access this file via the RBOX_ADMIN environment variable, as follows: \$RBOX_ADMIN/bdump/alert_ucrb.log	Under the UNIX user rboxsw .

Red Box log and trace files

Red Box maintains independent startup, closedown and runtime log files, and trace files for each Red Box process.

Table 3.2 Red Box log and trace files

File name	Description	Location
S89rbstart.log K49rbshut.log	Startup log (automatic startup). Closedown log (automatic closedown).	rboxuser user's home directory.
redbox.log rbprocessid.trc	Runtime log. Trace file for Red Box process, where <i>processid</i> is the process identifier.	

The following sections describe the **redbox.log** and trace files.

redbox.log file

The master server and its Red Box clients update the **redbox.log** file with details of each client connection to the Red Box database, including the process ID of the Red Box client started for each connection.

Figure 3.2 Extract from *redbox.log*

```
redbox: Red Box Server (3.00) Copyright © Ultracomp Limited 1992-1999
Compiled on Nov 27 1999 at 20:30:44
10:50:50.00 : 2190 : 1999/12/20 : redbox Starts (rb2190.trc)
10:50:52.00 : 2190 : Maximum data base connections 36
10:50:52.00 : 2190 : Maximum client connections 80
10:51:55.00 : 2190 : Listener ready : DB time = (19991220105155)
10:51:55.00 : 2219 : redbox Master Listener (rb2219.trc)
10:55:22.00 : 2578 : 1999/12/20 : redbox Server Responder (rb2578.trc)
10:55:23.00 : 2578 : *** New user Administrator(0) on unknown IP 127:0:0:1
10:56:31.00 : 3165 : 1999/12/20 : redbox Server Responder (rb3165.trc)
10:56:38.00 : 3165 : *** New user am(13238) on PC163 IP 172:16:1:163
13:42:05.00 : 24016 : 1999/12/20 : redbox Server Responder (rb24016.trc)
13:42:09.00 : 24016 : *** New user ndb(13237) on PC209 IP 172:16:1:209
```

The first line identifies the version of server software that is running, which in the example above is 3.00. Your server software may have a different version number.

Subsequent lines show the Red Box executable starting, then the master server, then three Red Box clients, as follows:

Table 3.3 Processes in the extract from *redbox.log*

Process ID	Description
2190,2219	These are the Red Box executable and master server. They log the number of database and client connections available, as shown on the Licence screen. The trace file rb2190.trc contains master server initialisation messages, and rb2219.trc contains runtime messages.
2578	This is the Red Box client started for the Periodic process (which connects to Red Box under the Administrator user). The number in brackets is the user's database identifier, which for the Administrator is always zero. Because the Periodic process runs on the server, it has a local IP address (127:0:0:1) and no PC identifier. The trace file rb2578.trc contains messages relating to Periodic.

Process ID	Description
3165	This is the Red Box client started for the Red Box PC user logged in as am , whose database connection id is 13238, PC identity is PC163, and IP address is 172:16:1:163. The trace file rb3165.trc contains messages generated during am 's Red Box session.
24016	This is the Red Box client started for the Red Box PC user logged in as ndb , whose database connection id is 13237, PC identity is PC209, and IP address is 172:16:1:209. The trace file rb24016.trc contains messages generated during ndb 's Red Box session.

Identifying trace files

The **redbox.log** file gives the name of the trace file for each Red Box process, including the Red Box client process started for each Red Box PC session. You can also identify the trace file for a particular PC session from any Red Box PC, as follows:

- 1 Click on the **General | Session Details** command.

The Session Details - Select screen appears.

- 2 Enter selection criteria to choose the required session. For example, enter the user's login name and click on the **Find** button.

The Session Details - View screen appears. Its **Process Id** field shows the process id of the Red Box client started for the chosen session. The trace file name on the server for that session has the format **rbprocessid.trc**.

Red Box UNIX Users

To secure against unauthorised access, Red Box installs the different server software components under four separate UNIX users: **rboxuser**, **rboxdba**, **rboxsw**, **rboxview**. By default, each user has a home directory of the same name. These directories are described in [Appendix A on page 79](#).

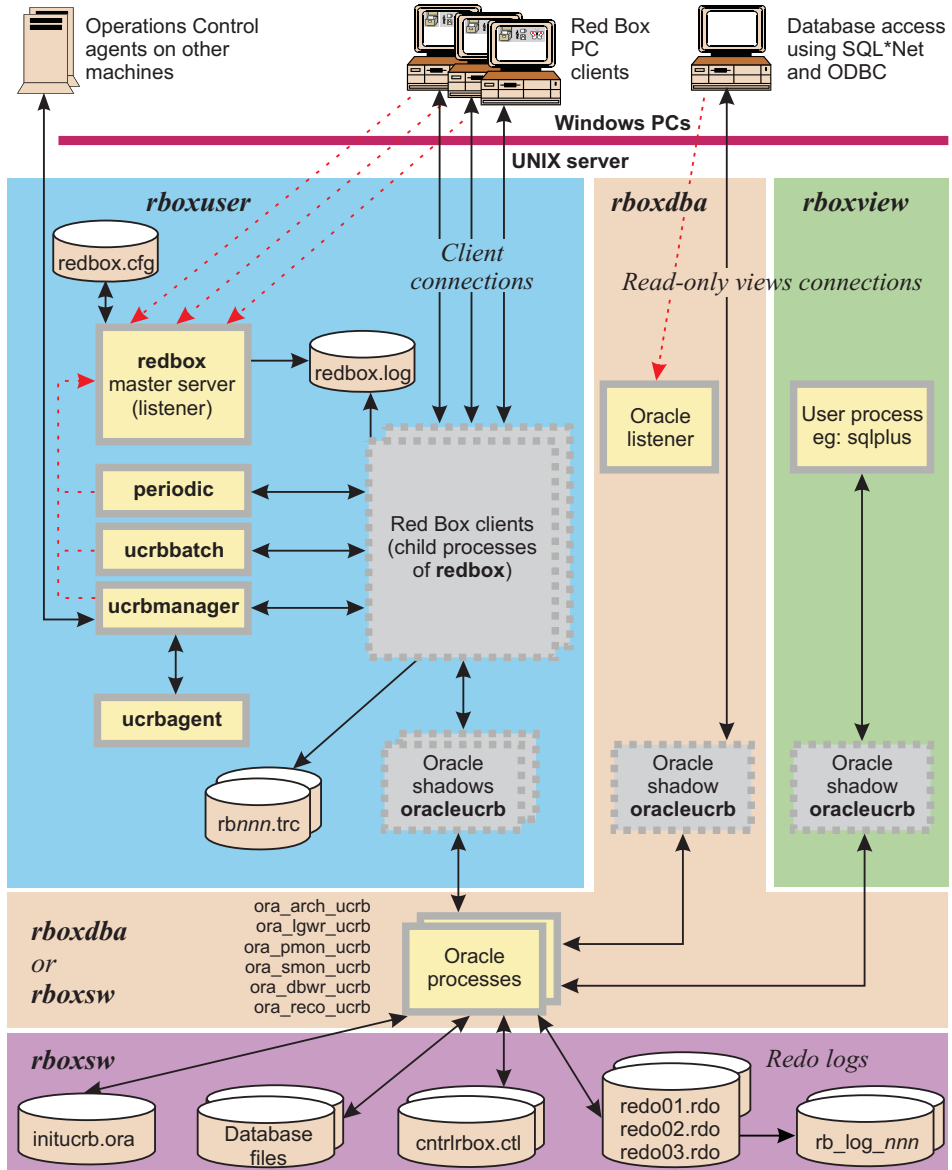
The following table describes the main functions of each Red Box UNIX user.

Table 4.1 Red Box UNIX users

User name	Description
rboxuser	<p>Server administrator</p> <p>Log in as this user to start and stop the:</p> <ul style="list-style-type: none"> • Red Box server software • Red Box batch process ucrbbatch • Operations Control managers and agents (ucrbmanager and ucrbagent).
rboxdba	<p>Database administrator</p> <p>Log in as this user for all database tasks, for example, starting and stopping the database, and for taking online backups, or partial backups.</p>
rboxsw	<p>Software owner</p> <p>This user holds the Red Box and Oracle7 software. Use it for taking full backups and for software upgrades.</p>
rboxview	<p>Read-only views</p> <p>Log in as this user for read-only view connections to the database.</p>

The following diagram shows how the Red Box processes and files are organised under the Red Box UNIX users (refer to the key on [page 27](#)).

Figure 4.1 Red Box UNIX users and their Red Box components



Key to figure 4.1



The solid boxes are permanent processes that run until the server software stops.



The dotted boxes are transient processes that start and end with the start and end of each successful database connection.



The dotted lines indicate processes making connection requests on the master server. The master server creates a Red Box client for each successful request, then retires to listen for more connection requests. The ongoing connection between the process and the database is serviced by the pair of transient processes (the Red Box client and the Oracle shadow process). Connections via the Oracle listener work in a similar way, except that for these connections there is no Red Box client.

Running Red Box

The Red Box server software must be running on the UNIX host before you can run:

- Red Box PC clients
- The batch utility
- Read-only views.

This chapter gives the procedures for starting and shutting down the server software. For information on running the PC client and setting up automatic login at Red Box PCs, refer to the Red Box online help.

Automatic startup and shutdown

In the Red Box Installation Guide, the installation procedures for the server software include setting up automatic startup and shutdown. If these procedures have been followed on your site, the server software will:

- Start automatically when the UNIX host is powered on.
- Shut down as part of the UNIX closedown sequence. Before closing the UNIX system, ensure that none of your users are connected to Red Box, as follows:
 - a) Log in to Red Box at a PC.
 - b) Choose the **General | Session Details** command.

Red Box displays the Session Details - Select screen.

- c) Enter **Y** in the “Logged In” field.

Red Box lists any users who are logged in.

On a regular basis, you should check the Red Box and Oracle log files for error messages generated during startup and closedown. For details, see the section [“Log and trace files” on page 20](#).

Manual startup and shutdown

The procedures to start and shutdown the Red Box system manually are given below. If the expected messages do not appear, check the Red Box and Oracle log files for error messages. For details, see the section “[Log and trace files](#)” on page 20.

Manual startup

To start the server software manually:

- 1 Start the Oracle software:

Log in as **rboxdba**.

Enter the command **ucrddbstart**.

(Messages should be displayed to confirm that the database and software have started.)

Exit from **rboxdba**.

- 2 Start the Red Box server applications:

Log in as **rboxuser**.

Enter the command **ucrbstart**.

(Messages should be displayed to confirm that the server software has started.)

Exit from **rboxuser**.

Red Box users can now start the Red Box PC client, run read-only views and/or the batch utility.

Manual shutdown

To close the Red Box system manually:

- 1 Check there are no Red Box connections from Red Box PCs, read-only views and/or the batch utility.
- 2 Stop the server software:

Log in as **rboxuser**.

Enter the command **ucrbshut**.

(Messages should be displayed to confirm that the server software has stopped.)

Exit from **rboxuser**.

- 3 Stop the Oracle software:

Log in as **rboxdba**.

Enter the command **ucrbdbshut**.

(Messages should be displayed to confirm that Oracle7 has stopped.)

Exit from **rboxdba**.

Checking for Red Box processes

You can use standard UNIX commands to check that the correct Red Box server and Oracle processes are running:

- 1 Log in as **rboxuser**.
- 2 List the processes running under **rboxuser** by entering the command:

ps -fu rboxuser

Figure 5.1 Example Red Box process listing

UID	PID	PPID	C	STIME	TTY	TIME	CMD
rboxuser	23121	22908	0	02:58:51	?	00:00:07	redbox
rboxuser	23122	23121	0	02:58:51	?	00:00:06	oracleucrb (DESCRIPTION=(LOCAL=YES)(ADDRESS=(PROTOCOL=beq)))
rboxuser	22908	1	0	02:57:22	?	00:00:00	redbox
rboxuser	23113	22908	0	02:58:10	?	00:00:10	redbox
rboxuser	23114	23113	0	02:58:10	?	00:00:18	oracleucrb (DESCRIPTION=(LOCAL=YES)(ADDRESS=(PROTOCOL=beq)))
rboxuser	23115	1	0	02:58:18	?	00:00:03	periodic
rboxuser	23135	22908	0	03:03:10	?	00:00:02	redbox
rboxuser	23136	23135	0	03:03:11	?	00:00:01	oracleucrb (DESCRIPTION=(LOCAL=YES)(ADDRESS=(PROTOCOL=beq)))

The master server and Red Box client processes in the above listing logged the messages described in the section “[redbox.log file](#)” on page 22. The following table describes these processes.

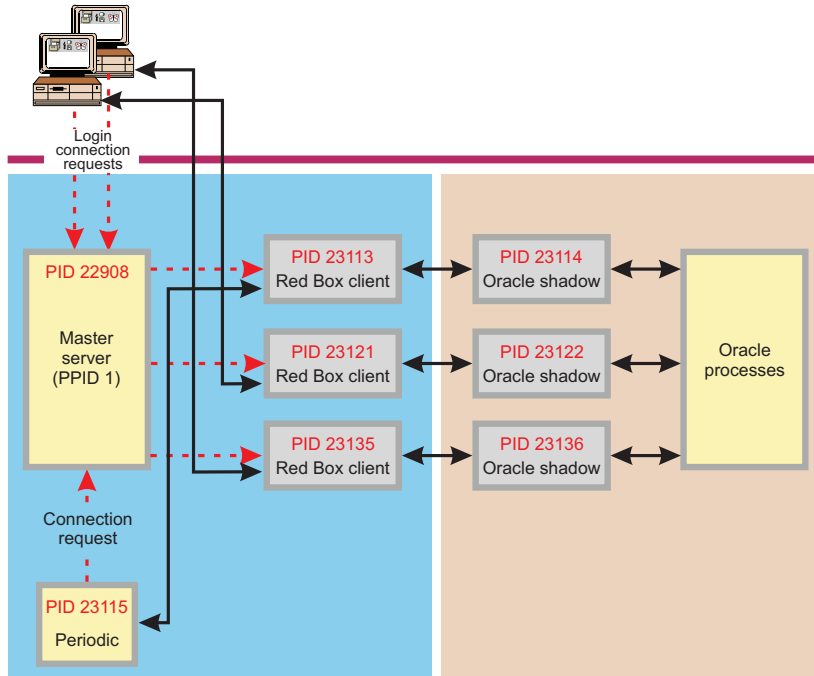
Table 5.1 Processes from example listing

PID	Description
22908	The master server whose PPID (parent process identifier) is always 1. PID 22908 is itself the PPID of the following Red Box clients.

PID	Description
23113	This is the Red Box client started for Periodic (PID 23115). PID 23113 starts the Oracle shadow PID 23114.
23121	This is the Red Box client started for a PC session. PID 23121 starts the Oracle shadow PID 23122. Because the PC client runs on a Red Box PC, it does not appear in the process listing. However, the Red Box client PID 23121 logs session details in the redbox.log file. If you look at the log file on page 22 , you will see that PID 3165 started for the Red Box user logged in as amm .
23135	This is the Red Box client started for another PC session. The Red Box client has logged session details in the redbox.log file, and started the Oracle shadow process 23136.

The following diagram shows the relationship between the processes in the example listing.

Figure 5.2 Example process listing (diagrammatically)



To check the master server

To check that the master server is running and listening for connection requests on the correct port number:

- 1 Log in as **rboxuser**.
- 2 Enter the command:

```
netstat -an | grep PortNumber
```

where *PortNumber* is **7300**, unless you changed the port number during Red Box installation.



Note: You may need to specify an absolute pathname for the **netstat** command.

The **netstat** display should include lines similar to the following example.

Figure 5.3 Example **netstat** display

Proto	Recv-Q	Send-Q	Local Address	Foreign Address	(state)
tcp	0	0	172.16.1.155.7300	172.16.1.229.1232	ESTABLISHED
tcp	0	0	127.0.0.1.7300	127.0.0.1.2489	ESTABLISHED
tcp	0	0	127.0.0.1.2489	127.0.0.1.7300	ESTABLISHED
tcp	0	0	*.7300	*.*	LISTEN

The first three lines of the example display show two established TCP/IP connections on port number **7300** (the port number of the server software). The local and foreign addresses are in the form *TCPIPAddress.Port Number*.

- Line 1 shows a connection from the server software to a PC client. The display does not show the other end of the connection, because this is in the PC whose TCP/IP address is 172.16.1.229.1232.
- Lines 2 and 3 show both ends of a local connection (the address 127.0.0.1 means “local”), so it must be for the Periodic process which runs in the server. Line 2 shows the connection from the server software to Periodic, and line 3 shows the connection from Periodic to the server software.

Line 4 shows the master server running and listening for connection requests. The local address is shown as **.PortNumber* above, but in some systems the address format may be *HostName.PortNumber*, where *HostName* is the name of the UNIX host. The foreign address is always shown as **.**.

To list Oracle processes

Depending on your version of UNIX, the Oracle processes may appear to run under the Red Box UNIX user **rboxdba** or **rboxsw**.

To list the Oracle processes:

- 1 Log in as **rboxdba**.
- 2 Enter the command:

```
ps -fu rboxdba (or ps -fu rboxsw)
```

The list of processes should include six Oracle processes, as shown in the following example:

Figure 5.4 Example Oracle process listing

UID	PID	PPID	C	STIME	TTY	TIME	CMD
rboxdba	18354	1	0	12:37:24	?	00:00:01	ora_pmon_ucrb
rboxdba	18355	1	0	12:37:27	?	00:00:02	ora_dbwr_ucrb
rboxdba	18356	1	0	12:37:30	?	00:00:00	ora_arch_ucrb
rboxdba	18357	1	0	12:37:33	?	00:00:01	ora_lgwr_ucrb
rboxdba	18358	1	0	12:37:36	?	00:00:01	ora_smon_ucrb
rboxdba	18359	1	0	12:37:39	?	00:00:00	ora_reco_ucrb

Red Box Backups

This chapter describes the recommended security procedures for the Red Box server software. You are recommended to stop the Red Box server software every day, take a full off-line backup, and restart the server software.



Note: The Red Box backup procedures require the use of a no-rewind tape device, and do not support backups that span more than one tape, or the use of disk files.

Red Box backup strategy

The recommended Red Box backup strategy is:

- Take regular UNIX system backups.
- Take a full backup (**ucrbckfull**) after installation and when there is a major change to the Red Box software. Red Box (including Oracle) must be closed.
- Take an offline backup (**ucrbckoffline**) when Red Box is closed, for example, overnight.
- Take online backups (**ucrbckonline**) while Red Box is running.

If you interrupt a run of **ucrbckonline** or the run fails, you must take special actions to continue running the Red Box service. For details, see the section [“Handling ucrbckonline failures” on page 42](#).

- Take partial backups (**ucrbckpartial**) for extra security while Red Box is running. Partial backups are recommended for single disk systems and can be as frequent as every hour.
- Verify the database (**ucrbverify**) at least once per cycle of backup tapes. The command requires Oracle to be running. For details, see the section [“Database integrity \(ucrbverify\)” on page 59](#).



Warning: If you use your own backup procedures instead of the Red Box ones to secure the Red Box data files, you must first shut down the database. Always run **ucrbtidylog** after backing up the archive directory. For details, see the section “[Space allocation](#)” on page 60.

UNIX backups

As part of normal UNIX security procedures, all files and sub-directories under each of the UNIX Red Box users (**rboxsw**, **rboxdba**, **rboxuser** and **rboxview**) should be backed up regularly, using standard UNIX commands such as **cpio**. You will need the latest **cpio** backup to recreate the Red Box users’ directory structures.

Red Box backups

Red Box is issued with a set of backup commands that you should use in addition to your regular UNIX backups, as described in the following table.

Table 6.1 Red Box backup commands

Command	Description
ucrbbckfull	<p>Full backup</p> <p>Take a full backup after installation and after a major change, such as the addition of a new database file or the application of a software amendment.</p> <p>Run ucrbbckfull under the user rboxsw when Red Box (including Oracle) is closed. The command backs up all the files owned by rboxsw. From a full backup, you can recreate all your Red Box server files.</p>

Command	Description										
<p>ucrbckoffline ucrbckonline</p>	<p>Daily backups</p> <p>ucrbckoffline and ucrbckonline are equivalent commands, except an offline backup is run with Red Box closed, and an online backup is run with Red Box running. Both commands run under the user rboxdba and carry out the following:</p> <ol style="list-style-type: none"> 1. They back up the following: <table border="0" data-bbox="382 431 1012 565"> <tr> <td><u>ucrbckoffline</u></td> <td><u>ucrbckonline</u></td> </tr> <tr> <td>Database files</td> <td>Database files</td> </tr> <tr> <td>Control files</td> <td>Control files</td> </tr> <tr> <td>Archived redo log files</td> <td>Archived redo log files</td> </tr> <tr> <td>Online redo log files</td> <td></td> </tr> </table> <p>An online backup does not include online redo log files because these files are open while the service is running.</p> 2. They delete the secured archived redo log files. <p>You should close Red Box regularly (if possible, daily) and run an offline backup. Ultracomp recommend offline backups because they return the database to a known point, so are easier to restore.</p> <p>While Red Box is running, you should take online backups. Online backups do not return the database to a known point, so the restore of an online backup requires a database recovery.</p> <p>You can include the ucrbckoffline and ucrbckonline commands in automatic jobs that are run regularly on your system, as described in the section “Automatic archiving” on page 38.</p>	<u>ucrbckoffline</u>	<u>ucrbckonline</u>	Database files	Database files	Control files	Control files	Archived redo log files	Archived redo log files	Online redo log files	
<u>ucrbckoffline</u>	<u>ucrbckonline</u>										
Database files	Database files										
Control files	Control files										
Archived redo log files	Archived redo log files										
Online redo log files											
<p>ucrbckpartial</p>	<p>Partial backup</p> <p>Partial backups secure the archived redo log files only, and are for protection against disasters, such as the loss of the disk or disks containing all the Red Box software. They are quick to run and can be taken hourly if required. Partial backups run under the user rboxdba while Red Box is running.</p> <p>If Red Box is installed on a single disk, you should take partial backups because you will lose data if the disk fails (you cannot recover data from a second disk). The frequency of partial backups depends on how much data you are prepared to lose.</p>										

Backup cycles

The following table shows an example daily backup cycle using five generations of backup tapes. You should keep at least three generations in case you need to return to more than one backup. Before overwriting the first tape in the cycle, you should run **ucrbverify** to validate the database.

Table 6.2 Recommended backup cycle between runs of **ucrbverify**

Day	ucrbverify	backup1	backup2	backup3	backup4	backup5
1	✓	✓				
2			✓			
3				✓		
4					✓	
5						✓
6	✓	✓				

Automatic archiving

The Red Box backup commands are designed to be included in automatic jobs that are run regularly on the UNIX system. The commands log their processing in a backup log file, and the content of each log file is written to a separate dump file when the procedure completes.

If a backup fails, the backup command returns a non-zero error code and writes any error messages at the end of the log file.



Note: You must include the *TapeDeviceName* parameter for each backup command included in an automatic job. If the parameter is omitted, the backup command displays an interactive prompt asking you which tape device to use.

Running the backup commands

The procedures to run the Red Box backup commands are:

- 1 Set up the correct environment for running the command.

Backup command	Environment
ucrbckfull ucrbckoffline	Red Box (including Oracle) must be closed.
ucrbckonline ucrbckpartial	Red Box and Oracle must be running.

- 2 Log in as the Red Box user under which the command is run.

Backup command	Login username
ucrbckfull	rboxsw
ucrbckoffline ucrbckonline ucrbckpartial	rboxdba

- 3 Enter the command name; one of:

ucrbckfull [*TapeDeviceName*]
ucrbckoffline [*TapeDeviceName*]
ucrbckonline [*TapeDeviceName*]
ucrbckpartial [*TapeDeviceName*]

where *TapeDeviceName* is the name of a no-rewind tape device. You must use a no-rewind device; use of a rewind device (or file) will invalidate the backup and may result in loss of data.

If the command is included in a backup job, you should always include the *TapeDeviceName* parameter. If omitted, the command uses the device selected the last time the command was run, and if none prompts for a device name:

Specify no-rewind device for use with backups? [/dev/nrStp0]

Choose <Return> without making an entry to select the default name in brackets.

If you omit the *TapeDeviceName* parameter, the command also prompts for confirmation of the tape device:

Is this device *TapeDeviceName* correct?

Enter **Y** to accept the tape device, or enter **N** to supply a different device name.

Display during processing

During processing, the **ucrbckfull**, **ucrbckoffline**, **ucrbckonline** and **ucrbckpartial** commands display:

CommandName: (logfile backup.log)

and on completion, they display:

(completes OK) or (completes unsuccessfully)

The message “completes (or ends) OK” confirms that the backup (or database verification) was successful, and you do not need to take further action.

The message “completes (or ends) unsuccessfully” indicates that the backup or database verification failed. In this case, you should check the backup log and/or dump files, described in the section “**Backup log files**” on page 41.

Backup sequence numbers

For recovery purposes, Red Box gives each **ucrbckoffline**, **ucrbckonline** or **ucrbckpartial** backup a sequence and sub-sequence number, and you should include these numbers in the information on the backup tape label. Use the **ucrbcklist** command to display the sequence number of a backup on a tape and the contents of the backup, as described in the next section.

Listing the contents of a backup tape

Run the Red Box **ucrbcklist** command to display a tape’s sequence number and to list the files backed up on the tape.

- 1 Put the backup tape in the tape drive.
- 2 Set up the environment variable **RBOX_HOME**, by entering the commands:

```
RBOX_HOME=rboxswHomeDirectory/redbox/3.00  
export RBOX_HOME
```

where *rboxswHomeDirectory* is the **rboxsw** user’s home directory.

- 3 Enter the command:

```
ucrbcklist
```

- 4 The command displays the date and time of the backup and its dump sequence number and sub-sequence number. Transfer this information to the tape label.

Red Box prompts:

Do you wish to continue?

- 5 Enter **Y** for a list of backed-up files.



Warning: `ucrbcklist` will only work with `ucrbckoffline`, `ucrbckonline` and `ucrbckpartial` backup tapes. It will not work with backup tapes written using `ucrbckfull`.

Backup log files

The backup commands maintain a **backup.log** file during processing, and write a history of their processing to a permanent dump file.

Backup.log files

You can view the progress of the backup while it is running, by viewing the **backup.log** file which the backup commands update. The file is held in the home directory of the user from which the command is run.

Figure 6.1 Example backup.log file for run of `ucrbckpartial`

```
ucrbckpartial : (Version 1.6.2.0 - 14/Oct/1999)
Backup of red tape information
starts at    02:07:33 PM Thu 02 Dec 1999 gmt
Using ucrbckcpio on /dev/nrStp0 to back files
( !!REDBOX_INFO_FILE )
( /u1/rboxsw/admin/ucrb/adhoc/rbcntrl.bak )
Ends successfully at 02:08:00 PM Thu 02 Dec 1999 gmt

Backup of archive log files
starts at    02:08:01 PM Thu 02 Dec 1999 gmt
Using ucrbckcpio on /dev/nrStp0 to back up files
( /u1/rboxsw/admin/ucrb/arch/rb_log_9 )
( /u1/rboxsw/admin/ucrb/arch/rb_log_10 )
Ends at ok   02:08:24 PM Thu 02 Dec 1999 gmt
```

The next run of the same backup command overwrites the **backup.log** file. Red Box preserves the log for each backup in a dump file for that backup.

Backup dump files

You can identify the dump file for a particular backup by the date and time held in the filename. The dump file is held in the **\$RBOX_ADMIN/logbook** directory:

```
dumpyyymmddhhmm
```

To view a dump file:

- 1 Log in as the Red Box user **rboxsw**.
- 2 Change to the log directory:

```
cd $RBOX_ADMIN/logbook
```

- 3 Use the UNIX **ls** command to display a list of dump files.
- 4 View the dump file that has the date and time of the relevant backup. The file will contain a list of the files selected for backup, followed by any messages resulting from the backup. Backup error messages are listed at the end of the file and you should report them to the Ultracomp Support Centre. For details, see the section **“Technical Support”** on page iii.



Note: With full backups, the list of selected files at the start of the dump file will contain error messages reporting access permission failures when changing directories. You should ignore these error messages.

Handling ucrbbckonline failures

During an online backup, Oracle places the Red Box tablespaces in a backup state. It returns the tablespaces to their normal state when they are backed up.

If an online backup fails or is interrupted, one or more tablespaces may still be in the backup state, and Oracle will fail to load. If this happens, reset the tablespaces as follows:

- 1 Log in as **rboxdba**.
- 2 Enter the command:

sqlplus rboxdba/Password

where *Password* is the **rboxdba** user's password.

- 3 At the **SQL>** prompt, enter the command:

select tablespace_name from dba_data_files;

Oracle lists the Red Box tablespaces, for example:

TABLESPACE_NAME

SYSTEM

RBS

TEMP

ADMDATA

ADMINDEX

USRDATA

USRINDEX

Tablespaces are described in the section **“Oracle7 tablespaces”** on page 18.

- 4 For each tablespace, enter the command:

alter tablespace TablespaceName end backup;

where *TablespaceName* is the name of the tablespace, for example, **system**.

Oracle will reset any tablespaces that are in the backup state. For those that are already reset, it will display a message similar to:

ERROR at line 1:

ORA-01142: cannot end online backup - none of the files are in backup

Ignore this message.

- 5 When all tablespaces are reset, Oracle (and Red Box) should start normally.

Recovery

You can recover your Red Box data up to the point of failure, if:

- You lose a disk or file system containing some or all of the Red Box software
- You lose a database file (Oracle will not load)
- A database file becomes corrupt (reported by the Red Box **ucrbverify** command described in the next chapter).



Note: Red Box and Oracle must be closed before you run any recovery or restore procedures.

Red Box recovery consists of four stages which are summarised below.

Table 7.1 Stages in a database recovery

Recovery stage	Summary
1	<p>Initial restore (necessary only if you have lost a disk or file system)</p> <p>This stage restores your file system directory structures and software from the latest UNIX and full Red Box backups.</p> <p>If you are restoring database files and the Red Box directory structure is intact, you must omit this stage to avoid overwriting your current online redo and archived redo log files.</p>
2	<p>Restore latest copies of lost or damaged database files</p> <p>This stage restores the database, online redo, and/or archived redo files that you require from offline, online and/or partial backups.</p>

Recovery stage	Summary
3	<p>Recover the database up to the point of the failure</p> <p>This stage recovers all database files to the same update level. It updates the database, if necessary, with transactions in the online redo log files and archived redo log files, and updates control files with the current state of the database.</p> <p>Omit this stage only if you are sure that all your database files are at the same recovery level and consistent with the update level held in the control file.</p>
4	<p>Take a full offline backup</p> <p>This stage is essential to secure a copy of the recovered state of the database files and to allow subsequent Red Box recoveries.</p>

Preparation

The following backup tapes should be available:

Table 7.2 Backups required for a restore

Backup	Description
Last full UNIX backup Last full Red Box backup (ucrbckfull)	Required to restore the Red Box directory structures and files.
Last offline backup (ucrbckoffline)	Required to restore the latest database files, online redo log files and archived redo log files.
Last online backup, if any (ucrbckonline)	Required only if you have taken online backups since the last offline backup, and you need to restore the latest database files and/or archived redo log files. (The online redo log files can be restored only from an offline backup.)
Last partial backup, if any (ucrbckpartial)	Required only if you have taken partial backups since the last offline/ online backup, and you need to recover the latest versions of your archived redo log files.

Stage 1: Initial restore

This step creates the correct Red Box directory structures and restores the Red Box files ready for a restore of up-to-date data (stage 2).



Note: Omit this step if you are only recovering database files. This step overwrites the current online redo log files which cannot then be used to recover data.

The steps in an initial restore are:

- Restore from the last UNIX backup
- Restore from the last full Red Box backup (**ucrbbackfull**)
- Restore the Red Box Email process.

Restoring a UNIX backup

If you have replaced a disk or file system and are recovering from a UNIX backup, it is important that the replacement disk has the same file system mount points, or where a new server is being used, the “old” Red Box users and directory structures are duplicated as far as practicable.

Red Box and Oracle have embedded directory structures which are identified during installation and remain fixed thereafter. You can change them manually, but you should consult the Ultracomp Support Centre for advice. For details, see the section “**Technical Support**” on page iii.

If you cannot replicate the original Red Box directory structures (for example, because the time required to obtain a replacement disk necessitates using space on an existing undamaged system) you should make temporary use of symbolic links to make the structures look the same.



Warning: The use of symbolic links can cause confusion, so for a long term solution, you should consult the Ultracomp Support Centre who will advise on changing the embedded directory structures.

Restoring a full Red Box backup (ucrbckfull)

If the failed disk or file system contained files belonging to any of the Red Box UNIX users **rboxsw**, **rboxdba**, **rboxuser** or **rboxview**, you must restore files from the last full Red Box backup.

The Red Box file permissions may mean that you cannot create the **oradata/ucrb** directories, and this will cause the restore to fail. To avoid this, do one of the following:

- Recreate the **oradata/ucrb** directories manually before running the restore
- Temporarily give full permissions to all users to each lost file system's mount point directories, as described in the next section.

To give full permissions to create the oradata/ucrb directories

Follow these steps to give full permissions for creating the **oradata/ucrb** directories, run a full restore, and restore normal permissions.

- 1 Log in as root.
- 2 For each file system that has been lost, make a note of the current permissions.
- 3 Enter the command:

```
chmod 777 /FileSystemName
```

where *FileSystemName* is the name of the file system.

- 4 Exit from root.
- 5 Run a full restore, as described in the next section.
- 6 For each file system that has been given full permissions, restore the normal permissions recorded in step 2, for example, by entering the command:

```
chmod 755 /FileSystemName
```

where *FileSystemName* is the name of the file system.

To run a full restore

Before running a full restore, ensure that the Red Box server software and Oracle service are closed, as described in [Chapter 5](#), and that the last full backup tape (taken using **ucrbckfull**) is in the tape drive.

- 1 Log in as **rboxsw**.
- 2 Ensure that the files to be created have the correct permissions, by entering the command:

```
umask 002
```

- 3 Change to the root directory:

```
cd /
```

- 4 Enter the command:

```
cpio -idvB < /TapeDeviceName “/FileSystemName1/*” .. “/FileSystemNamen/*”
```

where:

- a *TapeDeviceName* is the name of the tape drive, for example, **/dev/rct0**
- b *FileSystemName₁*.. *FileSystemName_n* optionally name the file system(s) that have been lost. **cpio** will restore files on the backup tape for the specified file systems only. Omit all the file system names to restore all files from the backup tape.

The **cpio** command restores each directory and file in turn, and displays a list of restored files. If a file exists that is the same age or newer than one on the backup tape, **cpio** reports that it has not restored that file.

- 5 Ensure that the environment is set up by the **.profile** copied in step 4, by logging out and logging back in as **rboxsw**.
- 6 Establish the correct ownerships by entering the command:

```
chmod 6751 $ORACLE_HOME/bin/oracle
```

To restore the Email process

When the Email process is restored, it is no longer owned by the user root. Re-establish the correct ownership as follows:

- 1 Log in as root.
- 2 Set up the environment variable **RBOX_HOME**, by entering the commands:

```
RBOX_HOME=rboxswHomeDirectory/redbox/3.00
```

```
export RBOX_HOME
```

where *rboxswHomeDirectory* is the **rboxsw** user's home directory.

- 3 Enter the commands:

```
cd $RBOX_HOME/ucbin
```

```
./ucrbemail
```

- 4 Exit from root.

Stage 2: Restoring latest copies of files

This stage uses the Red Box **ucrbckrestore** command to restore the database, online redo, and/or archived redo log files that you require from the latest offline, online and/or partial backups.



Note: The **ucrbckrestore** procedure does not overwrite existing files that are more recent than files on the backup tape. However, to ensure that you do not overwrite the wrong files and prevent data from being recovered, you may wish to take security copies of all undamaged files before running the restore. If you want **ucrbckrestore** to replace an existing file that is corrupt, you must remove the corrupt file before running **ucrbckrestore**.

The following table summarises the backup(s) to be restored, and the processing instructions to supply to **ucrbckrestore**. For instructions on running **ucrbckrestore**, see the section [“To run ucrbckrestore” on page 52](#).

Table 7.3 Restore procedures for different types of data loss

Data loss	Restore procedure
<p>Lost database files. Oracle will not run, or you have lost a disk or file system containing database files.</p>	<ol style="list-style-type: none"> 1 Restore all database files from the latest offline or online backup, including those on file systems that are not lost. 2 If the online redo log files and archived redo log files are undamaged, retain these files; otherwise, restore them as described below. 3 Make sure that a copy of the latest control file exists in each database directory, as described below. 4 Go through Stage 3: Database recovery” on page 55.
<p>Corrupt database file(s). The control file, online redo log files and archived redo log files are intact.</p>	<p>If you have a corrupt database file, you must go back to a point before the file corruption:</p> <ol style="list-style-type: none"> 1 Restore all database files from the most recent offline or online backup <i>taken before the last successful run of ucrbverify.</i> 2 Restore all archived redo log files backed up since then, starting with the oldest backup. 3 Process the restored archived redo log files by running a database recovery. <p>The following example shows ucrbverify reporting a database corruption after tape 3 in a sequence of five backup tapes.</p> <div style="text-align: center;"> </div> <p>The corruption reported after tape 3 may exist in any of tapes 1, 2 and 3. To recover the database, you must return to backup 5, taken before the last successful run of ucrbverify. The recovery sequence for this example is:</p> <ol style="list-style-type: none"> 1 Restore the database files from backup tape 5. 2 Restore the archived redo log files from each backup since backup 5, starting with the oldest (1, then 2, then 3). Each requires a separate run of ucrbckrestore. 3 Go through Stage 3: Database recovery” on page 55.

Data loss	Restore procedure
On-line redo files	If you have lost the file system containing the on-line redo files, restore them from the latest offline backup.
Archived redo log files	If you have lost the file system containing the archived redo log files, restore them from the latest offline, online or partial backup.
Control file	<p>The control file cntrlrbox.ctl contains details of the current state of the Oracle service, including which online redo log file is current. Red Box maintains an identical copy of the control file in each database directory (<i>/FileSystemName/oradata/ucrb</i>). This control file must be present in each database directory for Oracle to start.</p> <p>If you have lost a control file, if possible retrieve a copy of the current file from an undamaged file system and copy it to each database directory. The database directories are listed in the Oracle initialisation file \$ORACLE_HOME/dbs/initucrb.ora.</p> <p>If this is not possible, you can use the control file restored from the last online or offline backup. This file will have been copied to the directory \$RBOX_ADMIN/adhoc.</p>

To run ucrbbckrestore

Each run of **ucrbckrestore** restores one backup. If you need to restore multiple backups, you must repeat the command for each backup.

- 1 Red Box and Oracle must be closed. If necessary, go through the closedown procedures described in [Chapter 5](#).
- 2 Log in to the user **rboxsw**.
- 3 Enter the command:

```
ucrbckrestore
```

- 4 **ucrbckrestore** displays the following prompts. Choose <Return> to select the default responses in brackets.

```
Specify no-rewind device for use with backups ? [/dev/nrStp0]
Is this device /dev/nrStp0 correct ? [Y]
```

Enter **N** to retype the device name.

- 5 The procedure then displays the message:

Retrieving tape header information

Followed by a display identifying the backup (offline, online or partial), its date and time and sequence number, and a prompt for confirmation:

Do you want to continue ? [Y]

- 6 If the backup tape information is correct, choose <Return> to continue. Otherwise enter **N**, find the correct tape and repeat from step 2.

ucrbckrestore now displays a series of prompts. The prompts displayed depend on the type of backup tape being processed, and your answers will depend on the files that you need to restore, as described in the following table.

Table 7.4 *ucrbckrestore* prompts

Prompt	Description
<p>Database file questions</p> <p><i>For offline backups:</i> Do you want to restore files for file system fs ? [Y]</p> <p><i>For online backups:</i> Do you want to restore files for tablespace SYSTEM ? [Y]</p>	<p>For offline and online backups, ucrbckrestore prompts for the database files to restore. In either case, choose <Return> to restore the database files, or enter N if you do not want the files restored.</p> <p>ucrbckrestore repeats the question for each tablespace or file system on which data can be restored. Unless advised otherwise by Ultracomp, supply the same answer to all questions.</p> <p>If you enter N to a question, the procedure displays a message confirming that it has skipped the tablespace or file system.</p>

Prompt	Description
<p>Online redo log file questions (<i>offline backups only</i>)</p> <p>Do you want to restore on-line redo log files [Y]</p>	<p>This prompt appears for offline backups only. Press <Return> if the failed disk or file system contained online redo log files. Otherwise, enter N.</p> <p>The procedure confirms that it is restoring or skipping the online redo log files. If the files are being restored, it displays a list of the files.</p> <p>On completion, the procedure displays: “Above are restored on-line redo log files” or “Skipped on-line redo log files”.</p>
<p>Archived redo log file questions (<i>offline, online and partial backups</i>)</p> <p>Do you want to restore archive log files [Y]</p>	<p>Enter Y for database corruptions only, or where you have lost both database and archived redo log files.</p> <p>The procedure confirms that it is restoring or skipping the online redo log files, and displays a list of files.</p> <p>On completion, the procedure displays: “Above are restored archive log files” or “Skipped archive log files”.</p>

To copy control files

If any of the control files (**cntrlrbox.ctl**) in the database directories (*/FileSystemName/oradata/ucrb/*) are undamaged, copy them over any damaged files. The required copies are listed on the **control_files** line in the file **\$RBOX_ADMIN/pfile/initucrb.ora**.

If there are no undamaged versions of the control file on your file systems (that is, all file systems containing control files had to be restored), you can copy the control file from the backup tape as follows:

- 1 Copies of backed-up control files are restored automatically when a backup tape is listed or processed. After the last backup has been restored you should copy the control file to each of the damaged files. The restored copies of the control files are held in:

\$RBOX_ADMIN/adhoc/cntrlrbox.ctl for an offline backup.

\$RBOX_ADMIN/adhoc/rbctrl.bak for an online backup.

- 2 The copies which are required are listed on the **control_files** line in the file **\$RBOX_ADMIN/pfile/initucrb.ora**.

Stage 3: Database recovery

This stage restores all database files to the same update level. It updates the database, if necessary, with transactions in the online redo and archived redo log files, and updates control files with the current state of the database.

Omit this stage only if you are sure that all your database files are at the same recovery level and consistent with the update level held in the control file.

The main steps in recovering the database are:

- 1 Invoke the Oracle executable **svrmgrl**
- 2 Mount the database
- 3 Roll forward the data base using archived redo log files
- 4 Reset the on-line redo log files and open the data base.

These steps are described below.

- 1 **Invoke the Oracle executable.**

Log in as **rboxsw**

Enter the command:

```
svrmgrl
```

This will return a prompt

```
SVRMGR>
```

which indicates the procedure is ready to accept the next command.

- 2 **Mount the database.**

At the SVRMGR prompt, enter the commands:

**connect internal
startup mount****3 Roll forward the database.**

This step is required even where the database files have not been restored. Enter one of the following commands at the SVRMGR prompt, depending on the files restored.

- a If only the online redo log files have been restored, enter the command:

recover database until cancel;

The response should be

```
Media recovery complete
```

- b If only the database files have been restored, enter the command:

recover database;

The response should be similar to:

```
ORA-00279: Change nnn generated at 12/01/96 09:07_48 needed for thread 1
ORA-00289: Suggestion : /u1/rboxsw/admin/ucrb/arch/rb_log_141
ORA-00289 : Change nnn for thread 1 is in sequence #141
Specify log : { <RET>=suggested | filename | AUTO | CANCEL }
```

Check that the name of the suggested log file is correct, and if so press <Return>.

If all the archived redo log files required are present, enter AUTO to prevent further prompts. The next display should be similar to:

```
ORA-00278 :Logfile '/u1/rboxsw/admin/ucrb/arch/rb_log_141'
no longer needed for this recovery
```

Oracle repeats the messages and prompts until it has processed all the required archived redo log files.

On completion the following message is displayed:

```
Media recovery complete.
```

- c If you have restored both database and online redo log files then no recovery is required unless archive files from a partial backup are available. In this case, enter the command:

recover database using backup controlfile until cancel;

The response should be similar to:

```
ORA-00279: Change nnn generated at 12/01/96 09:07_48 needed for thread 1
ORA-00289: Suggestion : /u1/rboxsw/admin/ucrb/arch/rb_log_141
```

```
ORA-00289 : Change nnn for thread 1 is in sequence #141
Specify log : { <RET>=suggested | filename | AUTO | CANCEL }
```

Check that the name of the suggested log file is correct, and if so press <Return>. The next display should be similar to:

```
ORA-00278 :Logfile '/u1/rboxsw/admin/ucrb/arch/rb_log_141'
no longer needed for this recovery
```

Oracle repeats the messages and prompts until it has processed all the available files. After the last file, Oracle prompts for the next file, and the attempt to retrieve the file fails with an error message. Choose AUTO to go on to step 4. (Alternatively, you can choose CANCEL to go on to step 4.)

4 **Reset the on-line redo log files.**

Once media recovery is complete, the database should be opened by entering the following command at the SVRMGR prompt. The command depends on the type of recovery, as follows:

- a If the online redo log files have not been recovered, enter:

```
alter database open;
```

- b For all other types of recovery, enter:

```
alter database open resetlogs;
```

In either case, the confirmation message is:

```
Statement processed
```

Stage 4: Take a full offline backup

This stage is essential to secure a copy of the recovered state of the database files and to allow subsequent Red Box recoveries.

- 1 Close the service to allow a full offline backup to be made. Do this at the SVRMGR prompt, by entering:

```
shutdown;  
disconnect;
```

- 2 Press CTRL/D to return to the UNIX prompt.
- 3 Follow the instructions for **ucrbckoffline** in the section **“Running the backup commands”** on page 39.

Database Maintenance

This chapter describes the database maintenance commands, which can be grouped as follows:

- Database integrity (**ucrbverify**)
- Space allocation (**ucrbanalyze**, **ucrbextend**, **ucrbspace**, **ucrbtabreport**, **ucrbtidylog**)
- Moving the database (**ucrbexport**).

Instructions for running the commands are given in the section “[Running the commands](#)” on [page 66](#).

Database integrity (**ucrbverify**)

The database can become corrupt because of hardware problems, so you are recommended to run **ucrbverify** regularly (at least once per cycle of backup tapes) to check for database corruptions.

ucrbverify checks that the database is intact by exporting the database tables, ensuring that every block is read. The command then checks the Oracle log file for any errors that might indicate a block corruption.

If **ucrbverify** encounters an Oracle error, it displays an Oracle error message identifying the block number and file number of the file that could not be read. It holds all messages displayed during running in the log file **verify.log.1** in the directory in which it is run. It holds messages from the run of the command itself in the log file **verify.log.2**.

You should report any errors to the Ultracomp Support Centre who will advise whether you need to recover the database. For details, see the section “[Technical Support](#)” on [page iii](#).

Space allocation

As the use of Red Box extends within your organisation and you add new Red Box facilities, the database will grow in size beyond your original expectations. For this reason, you should monitor the size and usage of the database tablespaces. The database files that support each tablespace have a fixed file size, but you can increase the size of a tablespace by adding a file.

Tablespaces that may need to extend

The Oracle7 tablespaces are described in the section “Oracle7 tablespaces” on page 18, and are listed below.

admdata
adminindex
rbs
system
temp
usrdata
usrindex

- **admdata** and **adminindex** will not grow in size, so you will not need to extend them.
- **system**, **temp** and **rbs** may need extending because they are affected by your use of the Red Box system.
- **usrdata** and **usrindex** increase when Red Box creates records and indexes for records. When there are no more free blocks in a tablespace, a table or index cannot increase, and users will get “Failure to extend” errors when they try to create new records.

Monitoring free space (ucrbSPACE)

Use **ucrbSPACE** to monitor free space. This command is quick to run and reports the free and used space in each tablespace. By running the command regularly you can monitor the growth rate of the **usrdata** and **usrindex** tablespaces, and extend these before users get “Failure to extend” errors. If a Red Box user does get a “Failure to extend” message when inserting a record, run **ucrbSPACE** to check which tablespace is full, prior to running **ucrbextend**.

ucrbSPACE displays two tables which it also writes to the file **space.log** in the directory in which it is run.

- Table 1 lists the database tablespaces, the files that hold them and the file sizes in 4 Kb blocks and in Mbytes.
- Table 2 shows for each tablespace:

Total - the size in Mbytes of the tablespace

Free - the number of Mbytes free

The number of Mbytes used.

Examples are shown below:

*Figure 8.1 Example table 1 displayed by **ucrb**space*

Tablespace	File Name	Blocks (4096)	Size
ADMDATA	/u1/oradata/ucrb/admdata01.dbf	7,680	30 Mb
ADMINDEX	/u1/oradata/ucrb/adminindex01.dbf	1,280	5 Mb
RBS	/u1/oradata/ucrb/rbs01.dbf	10,240	40 Mb
SYSTEM	/u1/oradata/ucrb/system01.dbf	5,120	20 Mb
TEMP	/u1/oradata/ucrb/temp01.dbf	12,800	50 Mb
USRDATA	/u1/oradata/ucrb/usrdata01.dbf	12,800	50 Mb
USRINDEX	/u1/oradata/ucrb/usrindex01.dbf	12,800	50 Mb

*Figure 8.2 Example table 2 displayed by **ucrb**space*

Tablespace	Files	Fragments	Total	Free	Used
ADMDATA	1	1	30 Mb	8.18 Mb	21.82 Mb
ADMINDEX	1	2	5 Mb	2.89 Mb	2.11 Mb
RBS	1	3	40 Mb	3.57 Mb	36.43 Mb
SYSTEM	1	1	20 Mb	3.24 Mb	16.76 Mb
TEMP	1	3	50 Mb	50.00 Mb	4.00 Kb
USRDATA	1	1	50 Mb	40.62 Mb	9.38 Mb
USRINDEX	1	3	50 Mb	20.95 Mb	29.05 Mb

Depending on the size of each new record, many new records may be created before the “Used” value for **usrdata** and **usrindex** increases and the “Free” value decreases.

Identifying tables or indexes that cannot extend (**ucrbanalyze** and **ucrbtabreport**)

Run **ucrbanalyze** and **ucrbtabreport** to identify any tables that cannot extend. These commands take longer to run than **ucrbspace**, but are more useful when identifying impending problems. **ucrbanalyze** produces the statistics that **ucrbtabreport** uses, so always run **ucrbanalyze** before **ucrbtabreport**.

ucrbanalyze

ucrbanalyze does not produce reports itself, but gathers statistics for **ucrbtabreport** to report on. **ucrbanalyze** holds messages from its run in the files **analyze.log.1** (for admin tables) and **analyze.log.2** (for user tables) in the directory in which it is run.



Warning: The time that **ucrbanalyze** takes to run depends on the number of database records in your system. We recommend that you run the command out of hours, because it can adversely affect performance at Red Box PCs.

ucrbtabreport

ucrbtabreport identifies any tables or indexes that cannot extend. The command produces reports from the latest statistics collected by **ucrbanalyze**. The report names have the format **anatab*n*.lst** (where *n* is in the range 1 to 8) and are held in the directory in which **ucrbtabreport** is run. **ucrbtabreport** holds messages from its run in log files named **tabrep.log,*n***.

The reports from **ucrbtabreport** are for Ultracom's use mainly, but you should check whether the following two reports are empty:

- **anatab2.lst**. This report lists any Red Box tables that Oracle cannot extend. The list includes the table name and the size (in 4096 byte blocks) of the next extent.
- **anatab5.lst**. This report lists any Red Box indexes that Oracle cannot extend. The list includes the index name and the size (in 4096 byte blocks) of the next extent.

For the **anatab2.lst** and **anatab5.lst** reports, if **ucrbtabreport** displays:

```
WARNING: There are tables/indexes which cannot expand  
you should:
```

- 1 List the reports to see which tables or indexes are full. The following examples show an **anatab2.lst** report for two tables that cannot extend, and an **anatab5.lst** report.

Figure 8.3 Example **anatab2.lst** report

Fri Oct 11		page 1
Red Box - Tables Which Will Not Extend		
Table Name	Logical Name	Increase (blocks)
-----	-----	-----
r102	Report Section	1065
r74	History	210

Figure 8.4 Example **anatab5.lst** report

Fri Oct 11		page 1	
Red Box - Indexes Which Will Not Extend			
Table Name	Logical Name	Index Name	Increase (blocks)
-----	-----	-----	-----
r00	Name	UC_IND_418_NAME	27
r102	Report Section	UC_CON_325_FULL_IDENTITY	41
r74	History	UC_IND_705_PARENT	27

- 2 Calculate the required size for new data file from the information in **anatab2.lst** or **anatab5.lst**, as follows:

$$\frac{\text{TotalBlocks} \times 4096}{1048576} \text{ rounded up to the nearest 10 megabytes}$$

where *TotalBlocks* is the total blocks required for all the tables named in **anatab2.lst** or **anatab5.lst**.

This calculation gives the *minimum* file size. There is a limit to the number of database files that you can add, so if you have sufficient space, create a larger file. The limit to the number of database files depends on your Oracle7 installation, but usually is 16.

- 3 Run **ucrbextend** to add a data file to the relevant tablespace (**usrdata** for tables, **usrindex** for indexes), as described in the next section. **ucrbextend** will prompt for the file size in megabytes.

Extending a tablespace (**ucrbextend**)

Run **ucrbextend** when **ucrbspace** or **ucrbtabreport** indicate tablespaces that have no free space and tables or indexes that won't extend. The command adds a new data file for a named database tablespace. You can place the new file on a different file system and/or directory, but take care with file placement to maintain balanced disk utilisation.

First, **ucrbextend** lists the current data files and file sizes in each of the Oracle7 tablespaces.

*Figure 8.5 Oracle7 tablespaces files displayed by **ucrbextend***

Tablespace	Full File Name	File Size
admdata	/u5/oradata/ucrb/admdata01.dbf	3 Mb
adminindex	/u5/oradata/ucrb/adminindex01.dbf	3 Mb
rbs	/u5/oradata/ucrb/rbs01.dbf	40 Mb
rbs	/u5/oradata/ucrb/rbs02.dbf	1 Mb
system	/u5/oradata/ucrb/system01.dbf	10 Mb
temp	/u5/oradata/ucrb/temp01.dbf	72 Mb
usrdata	/u5/oradata/ucrb/usrdata01.dbf	256 Mb
usrindex	/u5/oradata/ucrb/usrindex01.dbf	185 Mb

ucrbextend then displays the following prompts. Where it displays a default, example values are shown in brackets below. Press <Return> without making an entry to select a default value.

Tablespace Name :

Enter the name of the tablespace to be extended, as listed in **Figure 8.5** above. For example, to extend tablespace **usrdata**, enter **usrdata**.

Directory Name (Default /u5/oradata/ucrb) :

Enter the directory name for the data file to be created. By default, **ucrbextend** creates it in the same directory as the last file for the tablespace. You can choose a different file system and/or directory.

Data File Name (Default usrdata02.dbf) :

Enter the file name of the new data file to be created. By default, **ucrbextend** adds one to the last filename.

Data File Size in Mb (Default 1) :

Enter the size (in megabytes) of the new data file, calculated from information in the **anatab2.lst** or **anatab5.lst** reports, as described in the previous section.

Having created the new data file, **ucrbextend** displays a confirmation message. For example:

```
ALTER TABLESPACE usrdata ADD DATAFILE 'u5/oradata/ucrb/usrdata02.dbf' SIZE 1M
```

Emptying the archive directory (**ucrbtidylog**)

Run **ucrbtidylog** to delete all archived redo log files (**rb_log_nnn**) in the Red Box archive directory, by default **\$RBOX_ADMIN/arch**. If you are not running the Red Box backup procedures **ucrbbackoffline** or **ucrbbackonline** regularly, or you are using your own backup procedures instead of the Red Box ones, you should run **ucrbtidylog** daily to tidy the archive directory.



Caution: The archived redo log files are required for recovery after a disk failure or corruption. If you delete them other than as part of your backup process, *you risk losing your data.*

Moving the database (**ucrbexport**)

You may need to move the Red Box database from one machine to another, to create a parallel database for testing, or to change the layout of the Red Box disk. The simplest way of doing any of these is to use **ucrbexport** to create database export files which can then be used by the Red Box setup program to recreate the database.

ucrbexport creates three export files (**rbadmCIR.300**, **rbusrC.300**, **rbusrR.300**) on disk or tape, suitable for use by the Red Box setup program to create a copy of the database.

You can use **ucrbexport** to back up the database, but in this case, ensure that the Red Box server is closed (by running **ucrbshut**) and copy the export files to offline storage media.

ucrbexport displays the following prompts.

```
Directory for export files ? [/U1/rboxsw/redbox/3.00/dbs]
```

If you allow the default path, you will overwrite files copied from your installation tape. Enter a different path name if required.

ucrbexport holds messages from its run in the log files **export.log.n** (where *n* is in the range 1 to 3) in the directory in which **ucrbexport** is run.

Importing an exported database

You can import an exported system into a newly-created Red Box system or into an existing Red Box system. In both cases, the database is overwritten by the exported data.

- 1 Transfer the three binary files created by **ucrbexport** (**rbadmCIR.300**, **rbusrC.300** and **rbusrR.300**) to the following directory on the target Red Box system.

```
$RBOX_HOME/dbs
```

- 2 Run the Red Box command **ucrbsetup**, as described in step 5 of the “Server installation” section of the *Red Box Installation Guide*.

Running the commands

Follow the steps in this section to run the database maintenance commands.

- 1 Set up the correct environment for running the command, as shown in the following table.

Table 8.1 Environment in which commands can be run

Command	Environment	Automatic/interactive running
ucrbanalyze	Oracle must be running.	Can be run from an automatic job.
ucrbexport	Red Box must be closed. Oracle must be running.	Must run interactively because ucrbexport prompts for the directory to hold the export files.

Command	Environment	Automatic/interactive running
ucrbextend	Oracle must be running.	Must run interactively because ucrbextend prompts for the tablespace to be extended and the directory and file name of the new data file.
ucrbspace	Oracle must be running.	Can be run from an automatic job.
ucrbtabreport	Oracle must be running.	Can be run from an automatic job. You must precede ucrbtabreport with a run of ucrbanalyze .
ucrbtidylog	Red Box and Oracle may or may not be running.	Can be run from an automatic job.
ucrbverify	Oracle must be running.	Must run interactively.

- 2 Log in as the Red Box UNIX user under which the command is run.

Table 8.2 Users under which commands are run

Login username	Command
rboxsw	ucrbexport and ucrbextend .
rboxdba	ucrbanalyze , ucrbspace , ucrbtabreport , ucrbtidylog and ucrbverify .

- 3 Enter the command name; one of:

ucrbanalyze
ucrbexport
ucrbextend
ucrbspace
ucrbtabreport
ucrbtidylog
ucrbverify

Applying amendments

When the Red Box server software requires an amendment, Ultracomp issues a patch tape containing the amendment, together with an Amendment Notice that describes the purpose of the amendment and gives the version number of Red Box server software to which it applies.

This chapter gives instructions for applying amendments and is relevant for most patch tapes. For some tapes, the Amendment Notice may give replacement instructions.

A patch tape contains:

- An amendment identifier
- The latest amendment.



Note: You must follow step 5 below to remove the temporary files created by one amendment before you can apply a second amendment. Amendments must be applied in sequence. When you apply an amendment, the Red Box patch script checks its identifier and prevents you applying an earlier amendment after a later one.

Before applying amendments to the Red Box server software, you should read through all the instructions and make sure that you understand them. If you have any queries, please contact the Ultracomp Support Centre. For details, see the section [“About this guide” on page ii](#).

Summary

The steps involved in applying an amendment are summarised below. You must shut down your Red Box system while you apply the amendment.

You will need:

- The issued patch tape. Check that this is the correct tape for your type of UNIX system.
- Access to the Red Box server.

- The name and password of the UNIX user in which Red Box was installed. Typically, the username will be **rboxsw**.

The steps in applying an amendment are:

- 1 Shut down the Red Box server.
- 2 Take a full backup.
- 3 Apply the amendment.
- 4 Restart the Red Box server.

Before going on to step 4, you should run Red Box normally until you know whether the amendment has been successful or not. If not, at this stage you can still revert to the previous amendment level.

- 5 Remove temporary files created when applying the amendment.



Warning: After removing temporary files, you cannot revert to the previous amendment level.

Applying the amendment

Follow the instructions for each step, given below. You may also need to refer to [Chapter 5](#) for details of the Red Box **ucrbshut**, **ucrbdbshut**, **ucrbdbstart** and **ucrbstart** commands. This chapter also explains how to use the UNIX **ps** and **netstat** commands to check whether the Red Box server processes are running.

Step 1: Shut down the Red Box server

If Red Box is not already closed down:

- 1 Use the UNIX **netstat** command to check that there are no users connected to Red Box.
- 2 Close down the Red Box system. For details, see the section [“Manual shutdown” on page 30](#).

Troubleshooting

This chapter describes Red Box server problems that prevent PC users being able to connect to Red Box.

If you have problems that are not covered here, contact the Ultracomp Support Centre. For details, see the section “[Technical Support](#)” on page iii. Before making the call:

- Have access to the contents of the relevant log and trace files described in the section “[Red Box log and trace files](#)” on page 21.
- Check the following:
 - a Is the problem influenced by your network environment?
 - b Are there relevant messages on the Red Box server console screen?
 - c If the problem concerns the use of a Red Box function, has the function previously been used successfully on your system? If the function has been used before, review any changes that may have affected the function.
 - d If performance is a problem, refer to the server installation chapters of the *Red Box Installation Guide*. If your workload or the number of Red Box PCs or Red Box users have increased, you may need to change your UNIX kernel parameters and swap space settings. Refer to your UNIX System Administrator or ask Ultracomp for a complete UNIX audit and tuning service.
 - e If performance has deteriorated gradually, check for a build-up of unread messages.

Connection failures

Failed connections between PCs and the Red Box server can have various causes. The lists below describe the checks that you should make if one or all of the Red Box PCs lose their connections with the Red Box server.

All PCs lock, or all new connections fail

- Check that the correct Red Box and Oracle processes are running. For details, see the section “[Checking for Red Box processes](#)” on page 31.
- If your TCP/IP package includes a **ping** application, use it to test the communication between individual PCs and the server. Use of the ping application imposes an overhead on the network, so do not use it over a long period or from several PCs at once.
- UNIX must be able to create two processes for each PC connection. This will not be possible if the system process limit is reached or your Red Box user process limit is reached. To check the number of processes running in the system and in your user, use the UNIX **ps** command piped to the **wc** command. For example, enter the commands:

```
ps -e | wc -l    to display the number of processes running in the system.  
ps | wc -l      to display the number of processes running in your user.
```

Ask your UNIX System Administrator to ensure that the process limits are high enough and that there is sufficient swap space for the number of processes required.

- Check the Oracle **alert_ucrb.log** file to see whether the database is running out of space for redo log files, and if necessary, run **ucrbtidylog** to delete old archived redo log files. For details, see the section “[File system full errors](#)” on page 77.

A single PC cannot connect to the server

- Check for duplicate internet addresses. This is a common problem when a new PC is installed. You should check:
 - a The PC’s internet address in the TCP/IP configuration file.
 - b The server’s server name or internet address, specified in the Service Name entry of the PC’s **redbox.ini** file (described in the *Red Box Administration Application Guide*). If entered as an address, it must be exactly as shown by a UNIX **netstat** command. Inserting leading zeros, for example, may cause errors with certain TCP/IP implementations.

If your TCP/IP package includes a **ping** application, you can check for a duplicate internet address as follows:

- a Power off the suspect PC.
- b Use **ping** from the server to the suspect internet address.

If there is a response, one of your other PCs has the same internet address.

- Check the Port entry in the PC's **redbox.ini** file. This must be the same as the port number in the server's **redbox.cfg** file.
- Check the Timer Frequency entry in the PC's **redbox.ini** file. For the PC to send a heartbeat to the server, the PC's Timer Frequency, and the `check_interval` and `idle_count` in the server's **redbox.cfg** file must be set so that the PC sends a heartbeat at least as often as the server checks for one. You may need to increase the `check_interval` and `idle_count` values to allow this.

The **redbox.cfg** file is described in [Appendix B](#). A PC user should return to Windows in normal mode within the time:

`check_interval x idle_count seconds`



Warning: Increasing the `check_interval` and `idle_count` values will decrease their effectiveness at dealing with genuinely lost connections, and will also affect more than just the timeout of inactive PCs. For example, the `check_interval` also controls how often messages are counted.

File system full errors

During running, the Red Box server software creates archived redo log files in the archive directory established during installation (by default **\$RBOX_ADMIN/arch**). Archived redo log files accumulate in the archive directory until you run an online or offline backup using the Red Box commands **ucrbckonline** or **ucrbckoffline**, described in [Chapter 6](#).

The online and offline backup commands automatically tidy the archive directory after each backup, by deleting the archive files the backup has secured. We recommended that you run daily backups, both to secure the system and to tidy the archive directory.

The rate of accumulation of archived redo log files in the archive directory depends on the size of the Red Box workload. If you do an unusually large number of Red Box transactions, or if you do not run daily backups, there is a risk of your file system filling. In this case, the following may happen:

- Red Box locks up on current PC connections.
- No new Red Box connections are accepted.

- The ORACLE log file **alert_ucrb.log** contains a message similar to:
ORA-00255 error occurred during archival of log 'xx', sequence #nn

This file is held in the **\$RBOX_ADMIN/bdump** directory.

For most Red Box systems, running **ucrbckonline** or **ucrbckoffline** daily should be sufficient to prevent an accumulation of archived redo log files. To check that there are no problems, you can use a standard UNIX command such as **df** to display the usage of the file system holding the archive file directory. If the usage is high regularly, review the frequency of your backups or make more file system space available.



Note: Depending on how your UNIX system is set up (for example, with 10% reserved for supervisor use) a value of 90% usage may indicate that the file system is full as far as Red Box is concerned.

If you use your own backup procedures (for example, you copy all of your filestore to tape each night), and not the Red Box backup commands, your archive redo files will not be cleared. Eventually, filestore will fill up and Oracle will not run. You will be unable to get further connections to the database, or having closed it down, it will not restart. In this case, run the Red Box procedure **ucrbtidylog** to delete old archived redo log files, then restart Red Box.

On a regular basis, check that filestore allocated to the Red Box user has not become full, or nearly full. Log files created before the last filestore backup can safely be deleted. Normally, these files are in the directory **\$RBOX_ADMIN/arch**, with names of the form **rb_log_nnn**, where *nnn* is a number.

Directory Structures

This appendix shows the Red Box directories and files under each of the Red Box UNIX users, **rboxsw**, **rboxdba**, **rboxuser** and **rboxview**. For descriptions of these, refer to [page 25](#).

Figure A.1 *rboxsw* directory structure

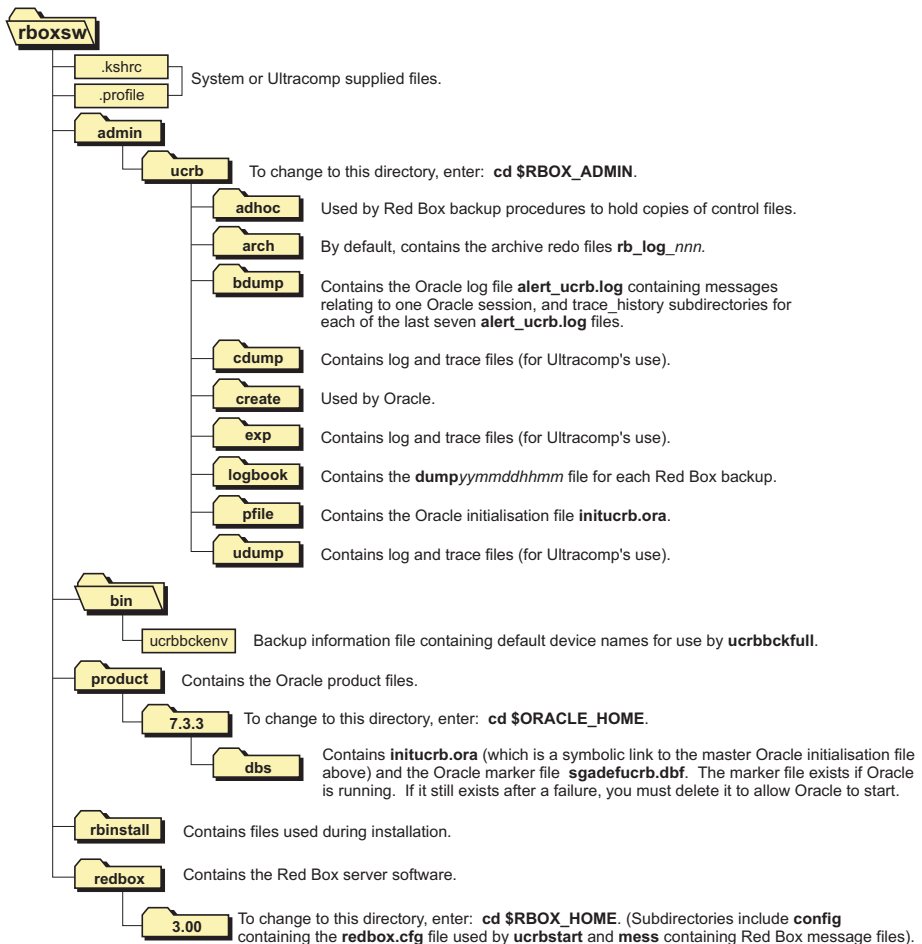


Figure A.2 *rboxdba* directory structure

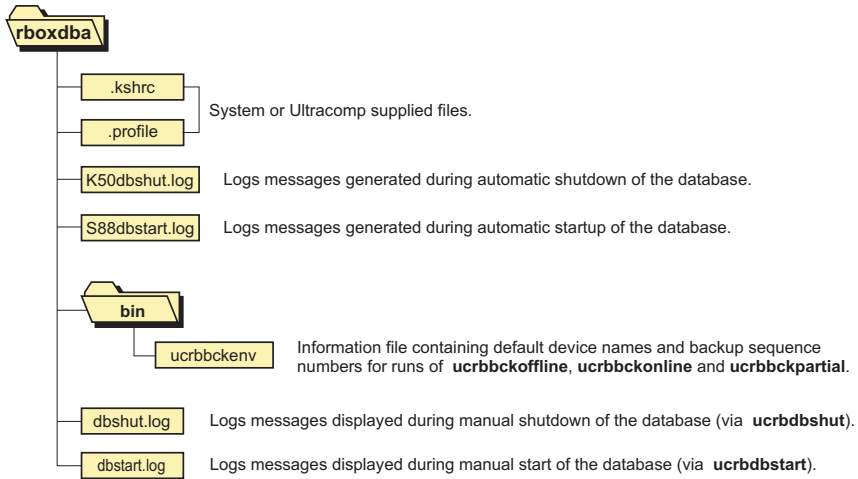


Figure A.3 *rboxuser* directory structure

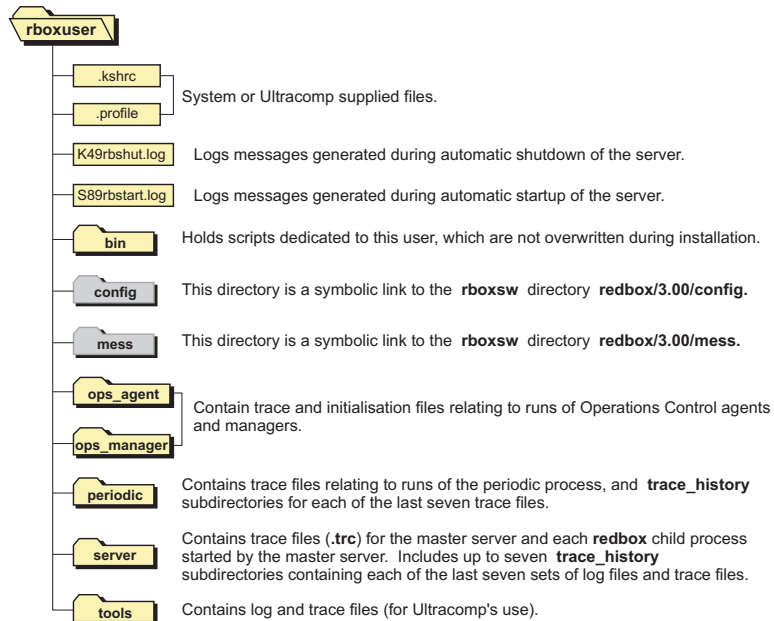
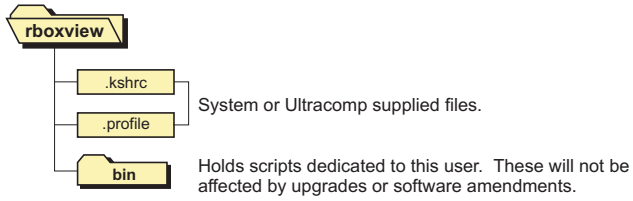


Figure A.4 *rboxview* directory structure



redbox.cfg

The configuration file, `/config/redbox.cfg`, is held in the `rboxsw` user's home directory, under the sub-directories `redbox/n.nn`, where `n.nn` is the Red Box version number, for example, `3.00`. You can access the file via the `$RBOX_HOME` environment variable:

`$RBOX_HOME/config/redbox.cfg`

Red Box stores the intervals at which update processes are run, and other options in `redbox.cfg`. Red Box creates this file at installation; it includes values supplied to the setup process.

You can customise the setup by changing the values in the `redbox.cfg` file, using a standard editor such as UNIX `vi`. You must re-load Red Box before the changes take effect.

Table B.1 Contents of the redbox.cfg file.

Entry (with default setting)	Description
<code>server=</code>	Optional server name, defaults to all interfaces on host.
<code>port=7300</code>	Port number of Red Box server. The issued number is 7 followed by a three-digit version identifier. For example, the port number for Red Box version 3.00 is 7300. The port number must match the port number supplied in PC <code>redbox.ini</code> files.
<code>text_table = text.tab</code> <code>message_table = dcmess.tab</code>	Issued files needed for the server software to run. The filenames are inserted correctly by the setup process and should not be changed.
<code>data_user=rboxuser</code>	The name/password of an Oracle7 user created by the setup process that Red Box will use to access user data.
<code>dba_user=rboxuser</code>	The name/password of an Oracle7 user created by the setup process that Red Box will use to access system data.

Entry (with default setting)	Description
admin_user=rboxadmin	The name/password of an Oracle7 user created by the setup process that Red Box will use to access administration data.
view_user=rboxview	The name/password of an Oracle7 user created by the setup process that Red Box will use to access read-only views of the user data.
check_interval=60	<p>The frequency in seconds with which the master server runs the delayed update process to:</p> <p>Check that PCs are active. A server Process is started for each Red Box user who connects to Red Box. The process checks regularly for a heartbeat or other message from the connected PC, and ensures that its server process is still running.</p> <p>Count the number of waiting messages for each Red Box user, team and user/department.</p> <p>Perform the delayed update processing, i.e. checking for records whose names have been changed, and copying the amended names to records in which they are referenced.</p>
idle_count=5	The number of times that the check interval can pass without response before the user's server process is abandoned, causing a "Process <i>nnnn</i> timed out" message to be written to the master server redbox.log file and a "Connection failed" message to be displayed at the PC.
trace_level=	Switches on tracing for the server and periodic, the syntax for each, with the full sets of implemented switches are:

Entry (with default setting)	Description
	<p>trace_level=redbox,0,1,2,3,4,5,6,7,8,9,10,11</p> <ul style="list-style-type: none"> 0 Trace startup actions, as the datamodel information is loaded. 1 Show all SQL and database accesses. 2 Trace use of fields and tables in SQL generation. 3 All tli object method calls and results. 4 Full trace of comms messages. 5 SQL performance trace (TKPROF). 6 List tracing. 7 Lock tracing. 8 Switches off FIRST_ROWS optimizer mode on select statements. 9 Set switches on BEFORE database initialisation. Normally, the tracing starts only after the initialisation is complete. 10 Message counts for Red Box users, teams and user/ departments. 11 Show all SQL without terms and results. 12 Trace application of domains to searches and updates. <p>trace_level=periodic,0,1,2,6</p> <ul style="list-style-type: none"> 0 Show major actions. 1 Show records processed. 2 Copies of RFC Next Stage processing messages. 6 Trace message distribution.

A

- Amendments 69–73
 - reversion 72
- Archived redo logs. *See* Redo logs, archived
- Archiving. *See* Backup

B

- Backup 35–43
 - automatic 38
 - before amendment 70, 71
 - commands explained 36
 - cycles 38
 - deleting archived redo logs 77
 - failures 42
 - log files 41
 - running commands 39
 - sequence numbers 40
 - strategy 35
 - tapes required for recovery 46
- backup.log. *See* Backup, log files
- Batch. *See* ucrbbatch

C

- check_interval 77, 84
- Closing Red Box. *See* Shutdown
- cntrlrboxctl. *See* Control file
- Configuration file 17, 26, 77
 - listed 83–85
- Connection failures 17, 75–77
- Control file 19, 26
 - backed up 37
 - restoring 52, 54

D

- Database
 - integrity 59
 - maintenance 59–67
 - maintenance commands 66–67
 - moving 65–66
 - recovery. *See* Recovery
 - space allocation 60–65
- Database files 18–20, 26
 - backed up 37
 - placement 19–20
- Delayed update process 13, 15, 84
- Directory structures 79–81
- Domains 85

E

- Email process, restoring 49

F

- File system full errors 77–78

I

- idle_count 77, 84
- Initialisation files 17–18
- initucrb.ora. *See* Oracle, initialisation file
- Internet address 76

L

- Listener. *See* Master server
- Log files 20–23

M

- Master server 13–14, 26–27, 31, 33–34

O

ODBC 26

Online redo logs. *See* Redo logs, online

Operations Control 10–11

Oracle

initialisation file 18, 26

listener 26–27

log file 76

processes 15, 26, 34

shadow processes 14, 26–27, ??–32, 32–??

 tablespaces. *See* Tablespacesoracleucrb. *See* Oracle, shadow processes**P**

PC

client 10, 13, 26, 32, 33

 connection failures. *See* Connection failures

Periodic process 13, 14, 26, 32

Port number 17, 77, 83

Processes, checking for 31–34

R

rboxdba 25–26

rboxsw 25–26

rboxuser 25–26

rboxview 25–26

Read-only views 10

Recovery 45–57

restoring a full backup 48

restoring database files 55–57

restoring latest files 50

summarised 45

Red Box

clients 14, 26–27, ??–32, 32–??

 configuration file. *See* Configuration file

database 9

 database files. *See* Database files

initialisation file 17, 76–77

log file 22–23, 26, 31

 processes. *See* Processes

UNIX users 25–27, 36, 79–81

redbox.cfg. *See* Configuration fileredbox.ini. *See* Red Box, initialisation fileredbox.log. *See* Red Box, log file

Redo logs 18–19, 26

archived 19, 37, 76

filling file system 77–78

tidying 65

online 18, 37

resetting 57

restoring 51–52

S

Server software 9–10

Shutdown

automatic 29

manual 30

SQL*Net 26

Startup

automatic 29

manual 30

Suspended sessions 13, 14

T

Tablespaces 18

extending 60–65

TCP/IP 76

Trace files 20–23

Troubleshooting 75–78

U

ucrbagent 10, 25–26

ucrbanalyze 62, 66–67

ucrbbatch 10, 13, 25–26
ucrbbackfull. *See* Backup
ucrbbacklist 40
ucrbbackoffline 19
ucrbbackoffline. *See* Backup
ucrbbackonline 19
ucrbbackonline. *See* Backup
ucrbbackpartial. *See* Backup
ucrbbackrestore 50–54
ucrbdbshut 31
ucrbdbstart 30
ucrbexport 65, 66–67
ucrbextend 63, 64–65, 66–67
ucrbmanager 10, 14, 25–26
ucrbpatch 71
ucrbpatchtidy 72
ucrbpatchundo 72
ucrbshut 30
ucrbspace 60–61, 66–67
ucrbstart 30
ucrbtabreport 62–63, 66–67
ucrbtidylog 36, 65, 66–67, 76, 78
ucrbverify 35, 38, 51, 59, 66–67

UNIX

- backups 36
- backups restored 45, 47
- closedown 29
- kernel parameters 75
- process limit 76
- swap space 75
- users. *See* Red Box, UNIX users