

ParnassusData Knowledge Transfer Series

Oracle Recovery

- Recovering a Deleted Datafile Without a Backup

Parnassus
诗 檀

ParnassusData 诗檀软件

Author: Biot Wang
Creation Date: 30-Dec-2016
Revised Date: 30-Dec-2016



Revision Control

Update Logging

Date	Author & Reviser	Revision	Details
30-Dec-16	Biot Wang	1.0	Initial

Reviewer

Revision	Reviewer	Position	Remarks
1.0	Maclean Liu		

Distributions

分发号	文档名	分发位置
1	PD_KT_Oracle Recovery - Recovering a Deleted Datafile Without a Backup	Asana

Contents

Revision Control.....	2
Update Logging	2
Reviewer	2
Distributions	2
Contents.....	3
Document Declaration.....	4
Aim.....	4
Reference.....	4
Introductions	5
About ParnassusData.....	5
About Tech Training.....	5
Emergency Hotline	5
Preparations	Error! Bookmark not defined.
Oracle Sample Database.....	Error! Bookmark not defined.
Datafile Deleted.....	Error! Bookmark not defined.
PRMSCAN Using.....	Error! Bookmark not defined.
PRM-DUL Using.....	Error! Bookmark not defined.
Others	20
Un-resolved.....	20
Resolved.....	20
Final.....	21

Document Declaration

本文由 Parnassusdata 工作人员编写及使用，如有问题或更新汇报，请联系作者进行修订。

Aim

本文为学习总结，也可用于工作使用参考。

Reference

ParnassusData 诗檀软件

Introductions

About ParnassusData



诗檀软件（ParnassusData）是一家立足于上海的软件、服务、方案供应商。公司长于数据库相关技术咨询及全生命周期管理，并始终以数据为核心，致力于为客户提供更多样，更优质，更关键的产品及技术支持。



诗檀软件为 Oracle 公司合作伙伴，公司可提供专业的 Oracle 数据库资讯及技术支持服务。

About Tech Training



诗檀软件作为上海 Oracle University 授权培训机构，将长期为企业客户及院校学生提供专业的 Oracle 认证培训及高级 IT 课程服务。

Emergency Hotline

相关咨询及 Oracle 紧急服务国内热线电话：400-690-3643
或可拨打移动电话：(+86) 13764045638 / 18521710662

About RPMSCAN

Introduction

prmscan 是诗檀软件独立研发的 ORACLE 数据块碎片扫描合并工具, 其适用于以下的场景:

误手动删除了文件系统 (任意文件系统 NTFS、FAT、EXT、UFS、JFS 等) 或 ASM 上的数据文件

文件系统损坏, 导致数据文件大小变成 0 bytes 即数据文件被清零

文件系统损坏, 导致文件系统无法 MOUNT 加载

ASM 存储元数据损坏, 导致 diskgroup 无法 mount 加载

文件系统或 ASM 其中的 LV 或 PV 被物理破坏或丢失

以上场景均可以利用 prmscan 直接扫描文件系统或 ASM 对应的 PV、LV 中的残余未被覆盖的 oracle block, 来实现对这些 oracle 数据块的合并重组, 以达到数据恢复的目的。

PRMSCAN 是基于 JAVA 语言开发的, 可以跨一切支持 JDK 1.6 以后操作系统, 包括 Windows、Linux、Solaris、AIX、HP-UX。

prmscan 是诗檀软件独立研发的 ORACLE 数据块碎片扫描合并工具, 目前该产品不独立销售, 可以联系诗檀软件 (13764045638) 以服务形式提供恢复服务。

PRMSCAN is one tool developed by Parnassusdata Team for Oracle Data block fragment scanning and merging.

It is considered and used only in the worst scenarios that your files can not be found and maybe are physically destroyed:

- 1) File Systems (Systems like NTFS, FAT, EXT, UFS, JFS etc.)/ Data Files / ASM Files are removed by wrong operations.
- 2) File system MOUNT failures because of system destroyed.
- 3) Destroyed ASM storage metadata causing diskgroup mount errors.
- 4) LV and PV in ASM or file systems are destroyed, cause data missing.

For these mentioned scenarios, we can use PRMSCAN to scan physical devices / volumes directly.

Rescuing those not overwritten Oracle blocks in PVs, LVs or filesystems. Searching, merging them byte by byte and finally to find the data all we want.

The PRMSCAN is Java based tool, so it can support all operation systems (including Windows, Linux, Solaris, AIX, HP-UX etc.) which supports JDK 1.6 and above.

Up till now, the tool is not for public, and it will be included in the Service provided by Parnassusdata.

For Parnassusdata Service, please contract the pre-sale (mobile: 13764045638).

In this sample case, we will show you how PRMSCAN works to find the missing data file and get the block merging.

ParnassusData 诗檀软件

RPMSCAN Practice

Demo Preparations

Prepare below software and installed them properly.

- Oracle Database 11.2.0.4
- PRMSCAN
- PRM-DUL

PRMSCAN and PRM-DUL are zipped files, unzip them and then we can use them directly.

In the sample case, we use Virtual Machine to install Oracle Database.

```
[oracle@hl ~]$ sqlplus / as sysdba

SQL*Plus: Release 11.2.0.4.0 Production on Thu Jan 5 12:08:51 2017
Copyright (c) 1982, 2013, Oracle. All rights reserved.

Connected to an idle instance.

SQL> startup
ORACLE instance started.

Total System Global Area  939495424 bytes
Fixed Size                  2258840 bytes
Variable Size              595593320 bytes
Database Buffers           335544320 bytes
Redo Buffers                 6098944 bytes
Database mounted.
Database opened.
SQL>
```



```
[oracle@h1 ~]$ lsnrctl status

LSNRCTL for Linux: Version 11.2.0.4.0 - Production on 05-JAN-2017 12:10:28

Copyright (c) 1991, 2013, Oracle. All rights reserved.

Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=h1.pd.com)(PORT=1521)))
STATUS of the LISTENER
-----
Alias                     LISTENER
Version                   TNSLSNR for Linux: Version 11.2.0.4.0 - Production
Start Date                05-JAN-2017 12:08:41
Uptime                    0 days 0 hr. 1 min. 47 sec
Trace Level               off
Security                  ON: Local OS Authentication
SNMP                      OFF
Listener Parameter File   /s01/app/oracle/product/11.2.0/dbhome_1/network/admin/
listener.ora
Listener Log File        /s01/app/oracle/diag/tnslsnr/h1/listener/alert/log.xml
Listening Endpoints Summary...
  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=h1.pd.com)(PORT=1521)))
  (DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=EXTPROC1521)))
Services Summary...
Service "PROD" has 1 instance(s).
  Instance "PROD1", status READY, has 1 handler(s) for this service...
Service "PROD1XDB" has 1 instance(s).
  Instance "PROD1", status READY, has 1 handler(s) for this service...
The command completed successfully
```

```
SQL> archive log list
```

```
SQL> archive log list
Database log mode          Archive Mode
Automatic archival        Enabled
Archive destination        USE_DB_RECOVERY_FILE_DEST
Oldest online log sequence 22
Next log sequence to archive 24
Current log sequence       24
SQL>
SQL> show parameter RECOVERY_FILE_DEST
NAME                                TYPE                                VALUE
-----                                -                                -
db_recovery_file_dest                string                               /s01/app/oracle/fast_recovery_
area
db_recovery_file_dest_size           big integer                          4977M
SQL> █
```

Oracle Sample Data

We need to create some tables and data for testing. Actually ASM files will be safer than File system data file storage by Oracle. Because for PRMSCAN, it can easily find data block in ASM storage device, but in the file system, more smaller data unit will be stored in separate places. This will cause data missing and merging corruptions.

Below is the script for us to generate the testing data.

```
BEGIN
  CREATE USER pd IDENTIFIED BY oracle DEFAULT TABLESPACE USERS QUOTA
  UNLIMITED ON USERS;
END;

BEGIN
  CREATE TABLE pd.positions(position_id VARCHAR2(10), position
  VARCHAR2(20)) TABLESPACE USERS;
  CREATE TABLE pd.employees(employee_id NUMBER(6),
    first_name VARCHAR2(20),
    last_name VARCHAR2(25),
    email VARCHAR2(25),
    hire_date DATE DEFAULT SYSDATE,
    position_id VARCHAR2(10) NOT NULL,
    manager_id NUMBER(6)) TABLESPACE USERS;
END;

BEGIN
  INSERT INTO pd.positions VALUES ('001', 'Group Leader');
  INSERT INTO pd.positions VALUES ('002', 'HR Manager');
  INSERT INTO pd.positions VALUES ('003', 'Team Leader');
  INSERT INTO pd.positions VALUES ('004', 'Developer');
  INSERT INTO pd.employees
  VALUES
    (1, 'Kart', 'Joe', 'kart.joe@pd.com', NULL, '001', NULL);
  INSERT INTO pd.employees
  VALUES
    (2, 'Hall', 'Tim', 'hall.tim@pd.com', NULL, '002', 1);
  INSERT INTO pd.employees
  VALUES
    (3, 'Maria', 'Cristino', 'maria.cris@pd.com', NULL, '003', 1);
  INSERT INTO pd.employees
  VALUES
    (4, 'Kumar', 'Akshay', 'kumar.akshay@pd.com', NULL, '004', 3);
  INSERT INTO pd.employees
  VALUES
    (5, 'King', 'John', 'king.john@pd.com', NULL, '004', 3);
  COMMIT;
END;
```

```
SQL> select * from pd.employees;
```

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	HIRE_DATE	POSITION_I	MANAGER_ID
1	Kart	Joe	kart.joe@pd.com			001

```

                2 Hall                Tim
hall.tim@pd.com                002                1

                3 Maria                Cristino
maria.cris@pd.com                003                1

EMPLOYEE_ID FIRST_NAME                LAST_NAME
-----
EMAIL                HIRE_DATE POSITION_I MANAGER_ID
-----
                4 Kumar                Akshay
kumar.akshay@pd.com                004                3

                5 King                John
king.john@pd.com                004                3

SQL> select * from pd.positions;

POSITION_I POSITION
-----
001        Group Leader
002        HR Manager
003        Team Leader
004        Developer

SQL>

```

```

SQL> select OWNER, TABLE_NAME, STATUS from dba_tables where TABLESPACE_NAME='USERS';

OWNER                TABLE_NAME                STATUS
-----
PD                EMPLOYEES                VALID
PD                POSITIONS                VALID

```

```
SQL> select FILE_NAME, FILE_ID, STATUS
2      from DBA_DATA_FILES
3      where TABLESPACE_NAME='USERS';

FILE_NAME
-----
FILE_ID STATUS
-----
/s01/app/oracle/oradata/PROD/users01.dbf
4 AVAILABLE
```

Datafile Deleted

Now we try to remove the data file (simulating the wrong operations).

```
[oracle@h1 ~]$ rm
/s01/app/oracle/oradata/PROD/users01.dbf
```

```
[oracle@h1 ~]$ sqlplus / as sysdba

SQL*Plus: Release 11.2.0.4.0 Production on Thu Jan 5 14:29:19 2017

Copyright (c) 1982, 2013, Oracle. All rights reserved.

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.4.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL> shutdown abort
ORACLE instance shut down.
```

'Shutdown abort' the database.

Try to restart the database and we can see the errors. And startup failed.

```
[oracle@h1 ~]$ sqlplus / as sysdba

SQL*Plus: Release 11.2.0.4.0 Production on Thu Jan 5 14:31:26 2017

Copyright (c) 1982, 2013, Oracle. All rights reserved.

Connected to an idle instance.

SQL> startup nomount
ORACLE instance started.

Total System Global Area 939495424 bytes
Fixed Size 2258840 bytes
Variable Size 595593320 bytes
Database Buffers 335544320 bytes
Redo Buffers 6098944 bytes
SQL>
```

```
SQL> alter database mount
2 ;

Database altered.

SQL>
SQL>
SQL>
SQL> alter database open;
alter database open
*
ERROR at line 1:
ORA-01157: cannot identify/lock data file 4 - see DBWR trace file
ORA-01110: data file 4: '/s01/app/oracle/oradata/PROD/users01.dbf'

SQL> shutdown abort;
ORACLE instance shut down.
SQL>
```

In more cases, error messages will seem quite vague, and DBAs must find more details in the trace files.

PRMSCAN Using

Now we use PRMSCAN to do the device scanning. If your system has not installed java yet. Please install it first.

```
[oracle@h1 ~]$ mkdir prmscan_tl
[oracle@h1 ~]$ cd prmscan_tl/
[oracle@h1 ~]$ su - root
[root@h1 ~]# cd ~oracle/prmscan_tl
[root@h1 prmscan_tl]# ls
lib      PRMScan.0.2.8.zip    PRMScan.jar    prmscan.key
[root@h1 prmscan_tl]# java -version
java version "1.8.0_111"
Java(TM) SE Runtime Environment (build 1.8.0_111-b14)
Java HotSpot(TM) 64-Bit Server VM (build 25.111-b14, mixed
mode)
```

We must be clear where is the lost file. In this case, it is under root. And the device mapping is /dev/mapper/VolGroup00-LogVol100

```
[root@h1 prmscan_tl]# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/mapper/VolGroup00-LogVol00
                92G   12G   75G   14% /
/dev/sda1       99M   23M   71M   25% /boot
tmpfs           2.0G   0     2.0G   0% /dev/shm
[root@h1 prmscan_tl]#
```

Scan the Logical Volume:

```
[root@h1 prmscan_tl]# java -jar PRMScan.jar --scan
/dev/mapper/VolGroup00-LogVol00 --guess 8k
```

PRMSCAN will store all related metadata in derby db after scanning.

```
[root@h1 prmscan_tl]# java -jar PRMScan.jar --scan /dev/mapper/VolGroup00-LogVol00 --guess 8k
[root@h1 prmscan_tl]# ls
db  derby.log  lib  PRMScan.0.2.8.zip  PRMScan.jar  prmscan.key  prmscan.log
[root@h1 prmscan_tl]#
[root@h1 prmscan_tl]#
```

See the prmscan.log

```
Scanning device /dev/mapper/VolGroup00-LogVol00, pos: 0, elapsed: 0.007s
Scanning device /dev/mapper/VolGroup00-LogVol00, pos: 1016320, elapsed: 0.062s
Scanning device /dev/mapper/VolGroup00-LogVol00, pos: 2032640, elapsed: 0.085s
Scanning device /dev/mapper/VolGroup00-LogVol00, pos: 3048960, elapsed: 0.095s
Scanning device /dev/mapper/VolGroup00-LogVol00, pos: 4065280, elapsed: 0.105s
Scanning device /dev/mapper/VolGroup00-LogVol00, pos: 5081600, elapsed: 0.107s
Scanning device /dev/mapper/VolGroup00-LogVol00, pos: 6097920, elapsed: 0.112s
Scanning device /dev/mapper/VolGroup00-LogVol00, pos: 7114240, elapsed: 0.115s
Scanning device /dev/mapper/VolGroup00-LogVol00, pos: 8130560, elapsed: 0.122s
Scanning device /dev/mapper/VolGroup00-LogVol00, pos: 9146880, elapsed: 0.151s
Scanning device /dev/mapper/VolGroup00-LogVol00, pos: 10163200, elapsed: 0.164s
Scanning device /dev/mapper/VolGroup00-LogVol00, pos: 11179520, elapsed: 0.176s
Scanning device /dev/mapper/VolGroup00-LogVol00, pos: 12195840, elapsed: 0.201s
Scanning device /dev/mapper/VolGroup00-LogVol00, pos: 13212160, elapsed: 0.254s
Scanning device /dev/mapper/VolGroup00-LogVol00, pos: 14228480, elapsed: 0.269s
Scanning device /dev/mapper/VolGroup00-LogVol00, pos: 15244800, elapsed: 0.281s
Scanning device /dev/mapper/VolGroup00-LogVol00, pos: 16261120, elapsed: 0.31s
Scanning device /dev/mapper/VolGroup00-LogVol00, pos: 17277440, elapsed: 0.346s
Scanning device /dev/mapper/VolGroup00-LogVol00, pos: 18293760, elapsed: 0.381s
Scanning device /dev/mapper/VolGroup00-LogVol00, pos: 19310080, elapsed: 0.393s
```

Then we can use PRMSCAN metadata to generate the output script file. The script can help us to recover the files under the scanned volume.

```
[root@h1 prmscan_tl]# java -jar PRMScan.jar
--output ./output_raw.txt
[root@h1 prmscan_tl]# java -jar PRMScan.jar
--outputsh ./output_sh.txt
```

```
dd if="/dev/mapper/VolGroup00-LogVol100" of=PD_DBF0.dbf bs=512 count=16 skip=1779
42355 seek=0 conv=notrunc
dd if="/dev/mapper/VolGroup00-LogVol100" of=PD_DBF0.dbf bs=512 count=16 skip=1779
37487 seek=0 conv=notrunc
dd if="/dev/mapper/VolGroup00-LogVol100" of=PD_DBF0.dbf bs=512 count=16 skip=1779
35721 seek=0 conv=notrunc
dd if="/dev/mapper/VolGroup00-LogVol100" of=PD_DBF0.dbf bs=512 count=32 skip=1779
11723 seek=0 conv=notrunc
dd if="/dev/mapper/VolGroup00-LogVol100" of=PD_DBF0.dbf bs=512 count=16 skip=1779
08680 seek=0 conv=notrunc
dd if="/dev/mapper/VolGroup00-LogVol100" of=PD_DBF0.dbf bs=512 count=16 skip=1779
08602 seek=0 conv=notrunc
dd if="/dev/mapper/VolGroup00-LogVol100" of=PD_DBF0.dbf bs=512 count=64 skip=1779
01711 seek=0 conv=notrunc
dd if="/dev/mapper/VolGroup00-LogVol100" of=PD_DBF0.dbf bs=512 count=16 skip=1779
00287 seek=0 conv=notrunc
dd if="/dev/mapper/VolGroup00-LogVol100" of=PD_DBF0.dbf bs=512 count=16 skip=1778
98492 seek=0 conv=notrunc
dd if="/dev/mapper/VolGroup00-LogVol100" of=PD_DBF0.dbf bs=512 count=16 skip=1778
70843 seek=0 conv=notrunc
dd if="/dev/mapper/VolGroup00-LogVol100" of=PD_DBF0.dbf bs=512 count=16 skip=1778
65975 seek=0 conv=notrunc
dd if="/dev/mapper/VolGroup00-LogVol100" of=PD_DBF0.dbf bs=512 count=16 skip=1778
64209 seek=0 conv=notrunc
dd if="/dev/mapper/VolGroup00-LogVol100" of=PD_DBF0.dbf bs=512 count=16 skip=1778
28811 seek=0 conv=notrunc
```

Run the output script.

```
[root@h1 prmscan_tl]# mkdir output
[root@h1 prmscan_tl]# cd output
[root@h1 output]# sh ./output_sh.txt
```

And that is it.

名字	大小	已改变	权限	拥有者
..		1/5/2017 4:58:52 PM	rxr-xr-x	oracle
PD_DBF1023.dbf	33,034,...	1/5/2017 5:00:59 PM	rw-r--r--	root
PD_DBF1020.dbf	1,052,6...	1/5/2017 5:00:59 PM	rw-r--r--	root
PD_DBF144.dbf	2,142,2...	1/5/2017 5:00:59 PM	rw-r--r--	root
PD_DBF80.dbf	12,582,...	1/5/2017 5:00:59 PM	rw-r--r--	root
PD_DBF28.dbf	3,684,4...	1/5/2017 5:00:59 PM	rw-r--r--	root
PD_DBF16.dbf	1,572,8...	1/5/2017 5:00:59 PM	rw-r--r--	root
PD_DBF11.dbf	33,036,...	1/5/2017 5:00:59 PM	rw-r--r--	root
PD_DBF9.dbf	7,868,4...	1/5/2017 5:00:59 PM	rw-r--r--	root
PD_DBF8.dbf	1,052,8...	1/5/2017 5:00:59 PM	rw-r--r--	root
PD_DBF4.dbf	3,672,1...	1/5/2017 5:00:59 PM	rw-r--r--	root
PD_DBF3.dbf	276,480...	1/5/2017 5:00:57 PM	rw-r--r--	root
PD_DBF2.dbf	195,264...	1/5/2017 5:00:43 PM	rw-r--r--	root
PD_DBF1.dbf	363,640...	1/5/2017 5:00:14 PM	rw-r--r--	root
PD_DBF0.dbf	2,621,4...	1/5/2017 4:59:32 PM	rw-r--r--	root

PRM-DUL Using

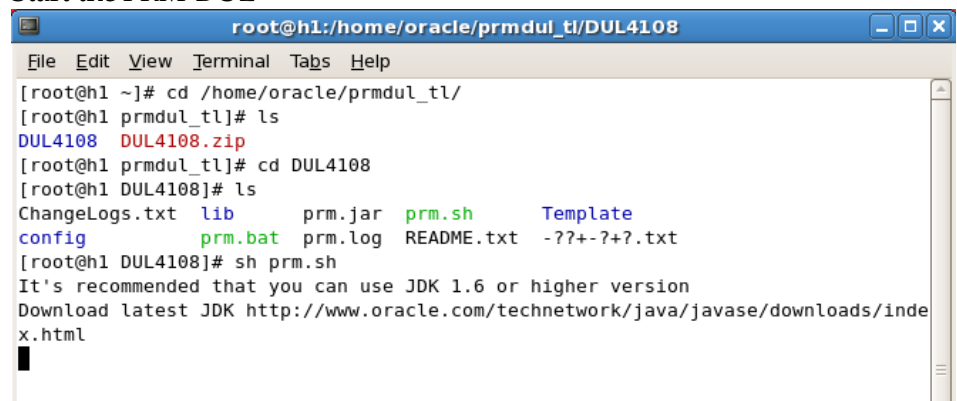
After done, PRM-DUL can be used then for data searching

and extracting.

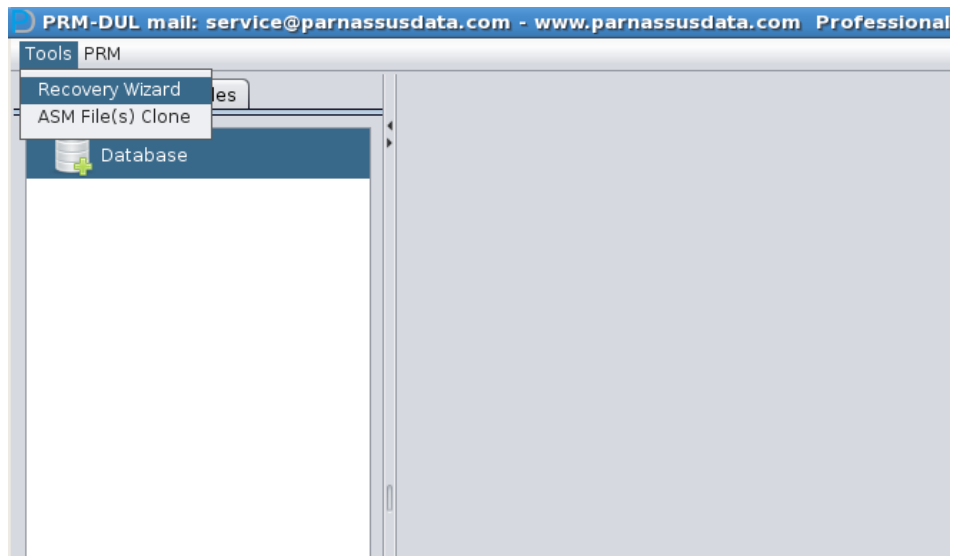
```
[root@h1 oracle]#  
[root@h1 oracle]# mkdir prmdul_tl  
[root@h1 oracle]# ls  
Desktop oradiag_oracle prmdul_tl prmscan_tl  
[root@h1 oracle]# cd prmdul_tl  
[root@h1 prmdul_tl]#
```

```
[root@h1 prmdul_tl]# ls  
DUL4108.zip  
[root@h1 prmdul_tl]#  
[root@h1 prmdul_tl]# unzip DUL4108.zip  
Archive:  DUL4108.zip  
  creating: DUL4108/  
  inflating: DUL4108/ChangeLogs.txt  
  creating: DUL4108/config/  
  inflating: DUL4108/config/bootstrap$.template  
  extracting: DUL4108/config/prm.conf  
  inflating: DUL4108/config/sys.col$.template  
  inflating: DUL4108/config/sys.obj$.template  
  inflating: DUL4108/config/sys.tab$.template  
  inflating: DUL4108/config/sys.user$.template  
  creating: DUL4108/lib/  
  inflating: DUL4108/lib/commons-io-2.4.jar  
  inflating: DUL4108/lib/derby.jar  
  inflating: DUL4108/lib/ojdbc6.jar  
  inflating: DUL4108/lib/prm_core.jar  
  inflating: DUL4108/lib/SQLFunctionCollection.jar  
  inflating: DUL4108/lib/swing-layout-1.0.4.jar  
  inflating: DUL4108/prm.bat  
  inflating: DUL4108/prm.jar  
  extracting: DUL4108/prm.log  
  inflating: DUL4108/prm.sh  
  inflating: DUL4108/README.txt  
  creating: DUL4108/Template/  
  inflating: DUL4108/-\ +-\ ? .txt
```

Start the PRM-DUL



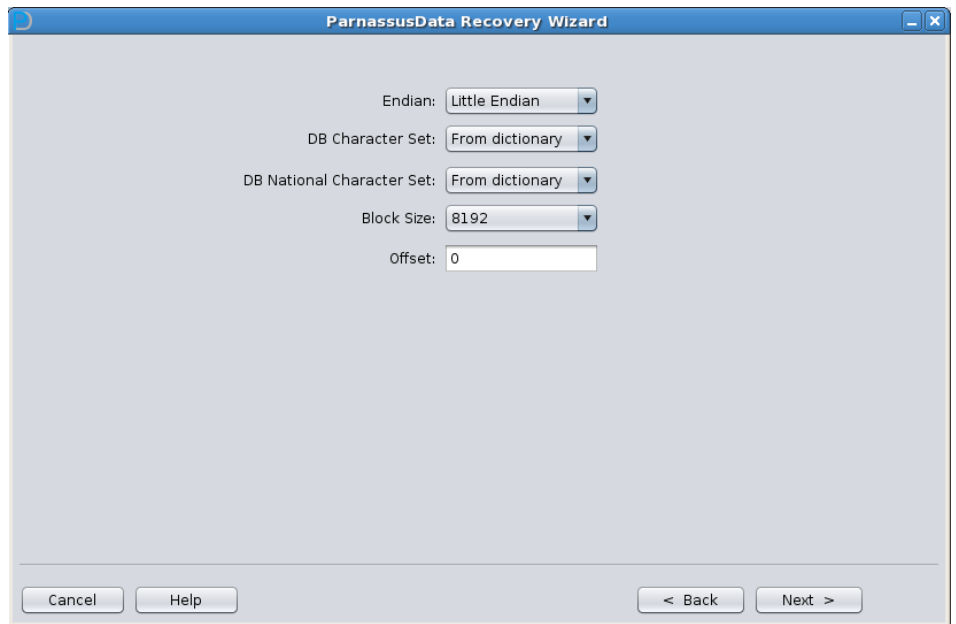
```
root@h1:/home/oracle/prmdul_tl/DUL4108  
File Edit View Terminal Tabs Help  
[root@h1 ~]# cd /home/oracle/prmdul_tl/  
[root@h1 prmdul_tl]# ls  
DUL4108 DUL4108.zip  
[root@h1 prmdul_tl]# cd DUL4108  
[root@h1 DUL4108]# ls  
ChangeLogs.txt lib prm.jar prm.sh Template  
config prm.bat prm.log README.txt -?-?-?.txt  
[root@h1 DUL4108]# sh prm.sh  
It's recommended that you can use JDK 1.6 or higher version  
Download latest JDK http://www.oracle.com/technetwork/java/javase/downloads/index.html
```

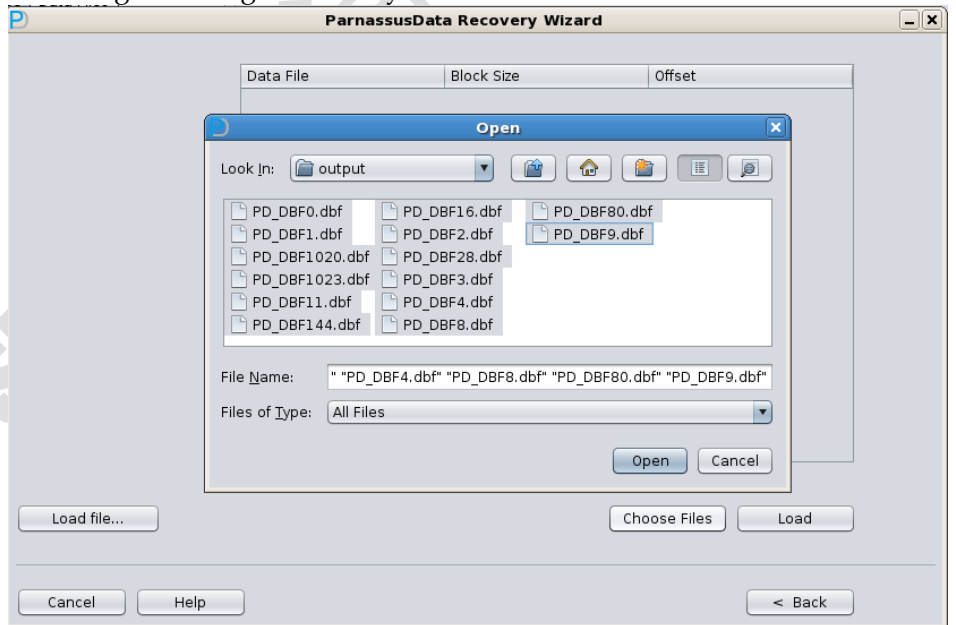
Because Oracle system tablespace files are also under the root.
So we can try Dictionary Mode.

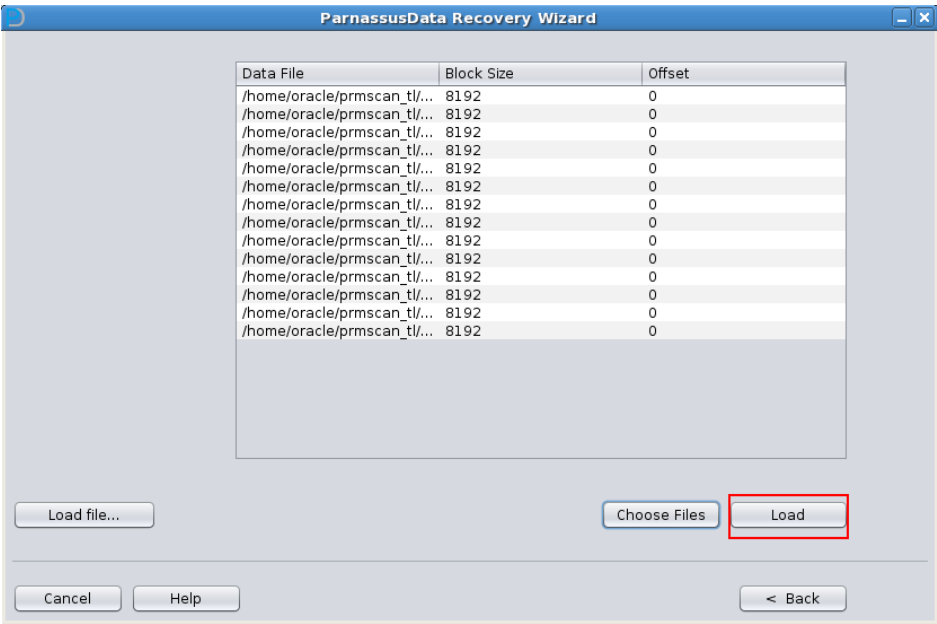


Linux is Little Endian system.



Including the files generated by PRMSCAN before.





Start data scanning ..

ParnassusData 诗檀

Others

Un-resolved

Q#	Details	Solutions	Revised Date

Resolved

Q#	Details	Solutions	Revised Date

Final

ParnassusData

ParnassusData Corporation, Shanghai, China

Phone: (+86) 13764045638

<http://www.parnassusdata.com/>

This document is provided for information purposes only and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, or including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

Parnassusdata and dbDao are registered trademarks of Parnassusdata and/or its affiliates. Other names may be trademarks of their respective owners.

Copyright © 2019 ParnassusData Corporation. All Rights Reserved.