

A ParnassusData Technical White Paper

Dec 2013

ParnassusData is a software company

ParnassusData Recovery Manager For Oracle Database

The logo for ParnassusData, featuring the word "Parnassus" in a bold, blue, sans-serif font and "Data" in a white, sans-serif font with a blue outline, set against a light blue background.

Creation Date:	Feb 09, 2014
Last Update:	Apr 14, 2014
Version:	<Version 0.2>



Document Control

Author

ParnassusData Liu

Change Logs

Date	Author	Version	Change Log
Feb 09, 2014	ParnassusData Liu		Created.
Apr 14,2014	ParnassusData Liu	V0.2	

Reviewers

Name	Position
------	----------

Approvals

<Approver 1>	_____
<Approver 2>	_____

Distribution

Copy No.	Name	Location
----------	------	----------

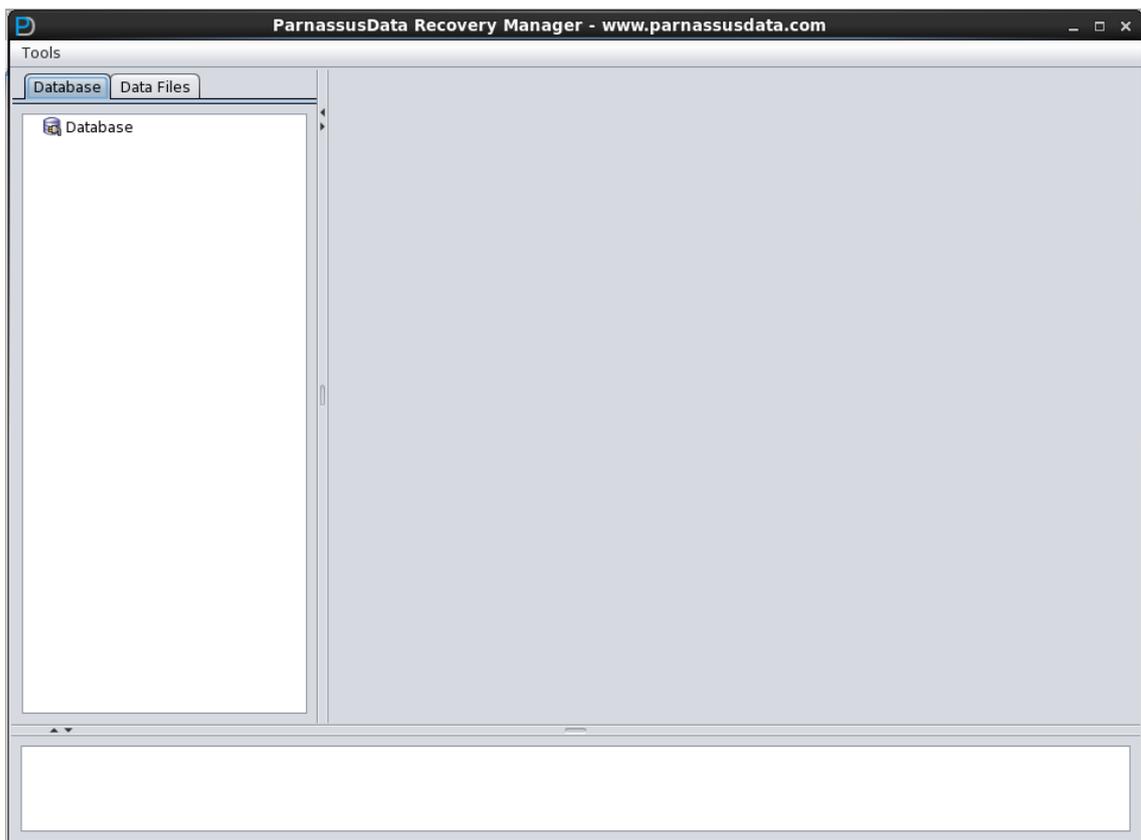
目录

Document Control	2
Author.....	2
Change Logs	2
Reviewers	2
Approvals.....	2
Distribution	2
概述.....	4
为什么要使用 PRM?	5
PRM 软件介绍.....	7
PRM 的安装与启动.....	13
Windows 平台下的启动方法.....	13
在 Linux/Unix 环境下的启动方法：	15
PRM 的许可证注册.....	17
基于不同的 Oracle 数据库恢复场景介绍如何使用 PRM.....	21
恢复场景 1 误 Truncate 表的常规恢复.....	21
恢复场景 2 误 Truncate 表的 DataBridge 数据搭桥恢复	39
恢复场景 3 ORACLE 数据字典受损导致数据库无法打开	46
恢复场景 4 误删除或丢失 SYSTEM 表空间	49
恢复场景 5 误删除了 SYSTEM 表空间和部分应用表空间数据文件.....	60
恢复场景 6 从被损坏的 ASM Diskgroup 中拷贝出数据库数据文件	61
恢复场景 7 ASM 下数据库无法打开	70
恢复场景 8 ASM 下误删或丢失 SYSTEM 表空间的恢复	74
FAQ 常见问题解答.....	77
Find More	79
Conclusion	79

概述

ParnassusData Recovery Manager(以下简称 PRM)是企业级 ORACLE 数据灾难恢复软件,可直接从 Oracle 9i,10g,11g,12c 的数据库数据文件(datafile)中抽取还原数据表上的数据,而不需要通过 ORACLE 数据库实例上执行 SQL 来拯救数据。ParnassusData Recovery Manager 是一款基于 JAVA 开发的绿色软件,无需安装,下载解压后便可直接使用。

PRM 采用 GUI 图形化界面(如图 1)简单方便。使用者无需额外学习一套命令,或者了解 ORACLE 的底层数据结构原理即可以通过恢复向导(Recovery Wizard)来恢复数据库中的数据。



图片番号 1

为什么要使用 PRM?

难道使用 RMAN 这个传统 ORACLE 恢复管理器的备份恢复还不够吗？为什么用户需要选择购买 PRM 呢？您的心头或许仍有这种疑惑！

在企业日益增长的 IT 系统中，数据容量正以几何级数扩展。Oracle DBA 在保证数据完整性的课题上正面临着现有磁盘存储系统容量不足以存放全量备份，基于磁带的备份在恢复数据时往往要求远远超过预期的平均修复时间等实际问题。

“对于数据库而言，备份重于一切”是所有 DBA 心中谨记的格言，但现实环境千差万别，企业的数据库环境中数据备份空间不足，采购的存储设备短期内无法到货，甚至于虽然进行了备份但是在数据恢复过程中发现备份实际不可用等问题均属常见的场景。

为了应对这些真实世界中常见的数据恢复困局，PRM 诗檀数据恢复管理软件充分发挥其对 ORACLE 数据库内部数据结构，核心启动流程等内部原理的理解，可以应对在完全没有备份情况下的 SYSTEM 表空间丢失、误操作 ORACLE 数据字典表、由于断电引起的数据字典不一致等数据库无法顺利打开的场景，也可以挽回误截断(Truncate)/删除(Delete)/业务数据表等人为的误操作，并从容恢复数据。

甚至于仅仅接触过 ORACLE 数据库几天的非 DBA 人员也可以轻松地使用 PRM，这得益于 PRM 简单的安装、和全称图形化的人机交互界面；实施恢复的人员不需要专业的数据库知识，不需要学习任何命令，更无需了解数据库底层的存储结构。仅仅需要轻轻点击几下鼠标就能从容恢复数据。

对比传统恢复工具 DUL, DUL 是 ORACLE 原厂内部恢复工具，其使用需要通过 ORACLE 内部流程，一般仅有购买了 ORACLE 原厂的现场服务的用户能够在原厂工程师的协助下使用该工具。PRM 打破了只有少数专业人士才能实施数据库恢复任务的限制，极大地缩短了从数据库故障到完整恢复数据的失败时间，降低了企业恢复数据的总成本。

通过 PRM 恢复的数据可以分为 2 种形式，传统抽取方式是将数据从数据文件中完整抽取出来并写入到平面文本文件中，之后使用 SQLLDR 等工具再加载到数据库中。传统方式简单易懂，但其缺点是需要 2 倍于现有数据容量的空间：即一份平面文本数据所占空间、以及之后将文本数据导入到数据库中所占空间；在时间上需要将原始数据从数据文件中抽取后，方

能导入到新建数据库中，往往又需要 2 倍的时间。

另一种是诗檀强烈推荐您使用的是 PRM 独创的数据搭桥方式，即通过 PRM 直接将抽取出来的数据加载到新建或者其他可用数据库中，这样避免了数据落地存储，对比传统方式有效节省了数据恢复所需要的空间和时间成本。

ORACLE 的 ASM 自动存储管理技术正被越来越多的企业采用，数据库采用 ASM 存储对比传统文件系统具有高性能、支持集群、管理方便等优势。但 ASM 的问题在于，对于普通用户而言 ASM 的存储结构过于黑盒了，一旦 ASM 中的某个 Disk Group 的内部数据结构发生了损坏导致 Disk Group 无法被成功 MOUNT，也就意味着用户重要的数据被锁死在这个 ASM 的黑盒中了。在这种场景中往往需要熟悉 ASM 内部数据结构的 ORACLE 原厂的资深工程师到达用户现场后通过手动修复 ASM 内部结构；而购买 ORACLE 原厂的现场服务对普通用户而言显得即昂贵又耗时。

基于 PRM 的研发人员(前 ORACLE 公司资深工程师)对 ORACLE ASM 内部数据结构的深入理解，PRM 中加入了特别针对 ASM 的数据恢复功能。

PRM 目前支持的 ASM 数据恢复功能包括：

1. 即便 Disk Group 无法正常 MOUNT，仍可以通过 PRM 直接读取 ASM 磁盘上的可用的元数据 metadata，并基于这些元数据将 Disk Group 中的 ASM 文件拷贝出来
2. 即便 Disk Group 无法正常 MOUNT，仍可以通过 PRM 直接读取 ASM 上的数据文件，并抽取其中的数据，支持传统抽取方式和数据搭桥方式。

PRM 软件介绍

ParnassusData Recovery Manager(PRM)基于 JAVA 开发, 这保证了 PRM 可以跨平台运行, 无论是 AIX、Solaris、HPUX 等 Unix 平台, Redhat、Oracle Linux、SUSE 等 Linux 平台, 还是 Windows 平台上均可以直接运行 PRM。

PRM 支持的操作系统平台 :

Platform Name	Supported
AIX POWER	✓
Solaris Sparc	✓
Solaris X86	✓
Linux X86	✓
Linux X86-64	✓
HPUX	✓
MacOS	✓

PRM 目前支持的数据库版本

ORACLE DATABASE VERSION	Supported
Oracle 7	✗
Oracle 8	✗
Oracle 8i	✗
Oracle 9i	✓
Oracle 10g	✓
Oracle 11g	✓
Oracle 12c	✓

考虑到部分陈旧服务器使用例如 AIX 4.3 Linux 3 等较早的操作系统, 这些操作系统上可能无法安装最新的 JDK 如 1.6/1.7; PRM 在研发过程中充分考虑了利旧性, 任何可以运行 JDK 1.4 的平台均可以运行 PRM。

此外由于 ORACLE 10g 数据库服务器软件自带了 JDK 1.4, 11g 自带了 JDK 1.5, 所以任何已安装 ORACLE 10g 及其以上版本的环境均可以顺利运行 PRM, 而且无需额外安装 JDK。

对于没有安装 JDK 1.4 版本的环境, 建议从以下地址下载

<http://www.oracle.com/technetwork/java/javasebusiness/downloads/java-archive-downloads-javase14-419411.html>

PRM 使用的最低 JAVA 软件环境为 JDK 1.4; 诗檀推荐您使用 JDK 1.6, 由于 JDK 1.4 以后对 JAVA 程序的性能做了很大优化, 所以 PRM 在 JDK 1.6 环境下的恢复速度要比 JDK 1.4 下快一些。

PRM 使用的最低硬件需求 :

CPU 中央处理器	至少 800 MHZ
物理内存	至少 512 MB
硬盘空间	至少 50 MB

PRM 推荐的硬件配置 :

CPU 中央处理器	2.0 GHZ
物理内存	2 GB
硬盘空间	2 GB

PRM 目前支持的多语言：

语言	字符集	对应的编码
中文 简体/繁体	ZHS16GBK	GBK
中文 简体/繁体	ZHS16DBCS	CP935
中文 简体/繁体	ZHT16BIG5	BIG5
中文 简体/繁体	ZHT16DBCS	CP937
中文 简体/繁体	ZHT16HKSCS	CP950
中文 简体/繁体	ZHS16CGB231280	GB2312
中文 简体/繁体	ZHS32GB18030	GB18030
日文	JA16SJIS	SJIS
日文	JA16EUC	EUC_JP
日文	JA16DBCS	CP939
韩语	KO16MSWIN949	MS649
韩语	KO16KSC5601	EUC_KR
韩语	KO16DBCS	CP933
法语	WE8MSWIN1252	CP1252
法语	WE8ISO8859P15	ISO8859_15
法语	WE8PC850	CP850
法语	WE8EBCDIC1148	CP1148
法语	WE8ISO8859P1	ISO8859_1
法语	WE8PC863	CP863
法语	WE8EBCDIC1047	CP1047
法语	WE8EBCDIC1147	CP1147
德语	WE8MSWIN1252	CP1252
德语	WE8ISO8859P15	ISO8859_15
德语	WE8PC850	CP850
德语	WE8EBCDIC1141	CP1141
德语	WE8ISO8859P1	ISO8859_1
德语	WE8EBCDIC1148	CP1148
意大利语	WE8MSWIN1252	CP1252

意大利语	WE8ISO8859P15	ISO8859_15
意大利语	WE8PC850	CP850
意大利语	WE8EBCDIC1144	CP1144
泰语	TH8TISASCII	CP874
泰语	TH8TISEBCDIC	TIS620
阿拉伯语	AR8MSWIN1256	CP1256
阿拉伯语	AR8ISO8859P6	ISO8859_6
阿拉伯语	AR8ADOS720	CP864
西班牙语	WE8MSWIN1252	CP1252
西班牙语	WE8ISO8859P1	ISO8859_1
西班牙语	WE8PC850	CP850
西班牙语	WE8EBCDIC1047	CP1047
葡萄牙语	WE8MSWIN1252	CP1252
葡萄牙语	WE8ISO8859P1	ISO8859_1
葡萄牙语	WE8PC850	CP850
葡萄牙语	WE8EBCDIC1047	CP1047
葡萄牙语	WE8ISO8859P15	ISO8859_15
葡萄牙语	WE8PC860	CP860

PRM 支持的表存储类型：

表存储类型	Supported
Cluster Table 簇表	YES
索引组织表, 分区或非分区	YES
普通堆表, 分区或非分区	YES
普通堆表 启用基本压缩	YES(Future)
普通堆表 启用高级压缩	NO
普通堆表 启用混合列压缩	NO
普通堆表 启用加密	NO
带有虚拟字段 virtual column 的表	NO
链式行、迁移行 chained rows 、 migrated rows	YES

注意事项：对于 virtual column、11g optimized default column 而言 数据抽取可能没问题，但会丢失对应的字段。这 2 个都是 11g 之后的新特性，使用者较少。

PRM 支持的数据类型

数据类型	Supported
BFILE	NO
Binary XML	NO
BINARY_DOUBLE	YES
BINARY_FLOAT	YES
BLOB	YES
CHAR	YES
CLOB and NCLOB	YES
Collections (including VARRAYS and nested tables)	NO
DATE	YES
INTERVAL DAY TO SECOND	YES
INTERVAL YEAR TO MONTH	YES
LOBs stored as SecureFiles	Future
LONG	YES
LONG RAW	YES

Multimedia data types (including Spatial, Image, and Oracle Text)	NO
NCHAR	YES
NUMBER	YES
NVARCHAR2	YES
RAW	YES
ROWID, UROWID	YES
TIMESTAMP	YES
TIMESTAMP WITH LOCAL TIMEZONE	YES
TIMESTAMP WITH TIMEZONE	YES
User-defined types	NO
VARCHAR2 and VARCHAR	YES
XMLType stored as CLOB	NO
XMLType stored as Object Relational	NO

PRM 对 ASM 的支持

功能	Supported
----	-----------

支持直接从 ASM 中抽取数据，无需拷贝到文件系统上	YES
支持从 ASM 中拷贝数据文件	YES
支持修复 ASM metadata	YES
支持图形化展示 ASM 黑盒	Future

PRM 的安装与启动

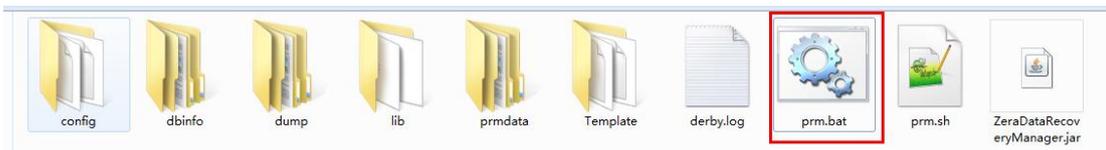
由于 PRM 是基于 JAVA 开发的纯绿色软件，所以无需额外安装，用户仅需要在下载软件 ZIP 包后解压即可用于恢复数据。

```
unzip prm latest.zip
```

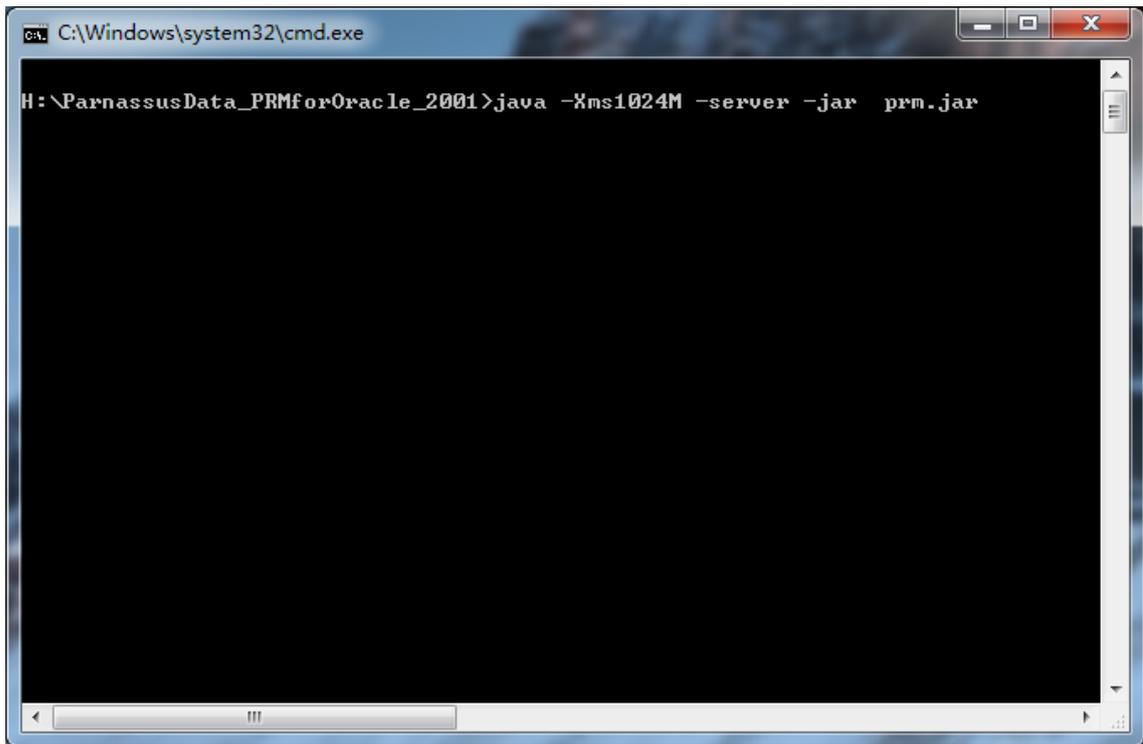
诗檀强烈推荐您使用命令行启动 PRM，这样可以从命令行中获得更多诊断信息。

Windows 平台下的启动方法

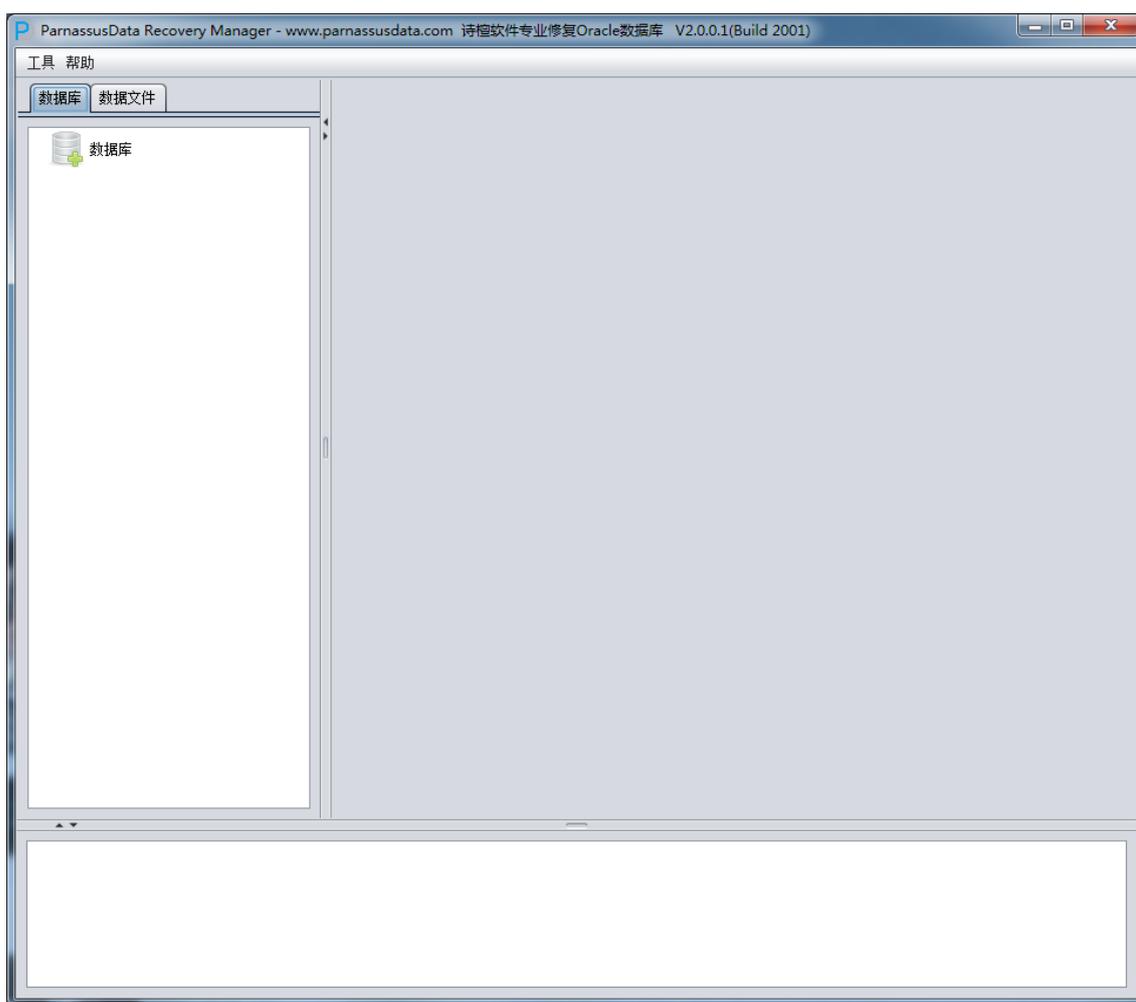
1. 首先保证 JDK 已正确安装且 java 已经加入到环境变量中：
2. 双击 PRM 解压目录下的 prm.bat



prm.bat 会在后台启动 RPM:



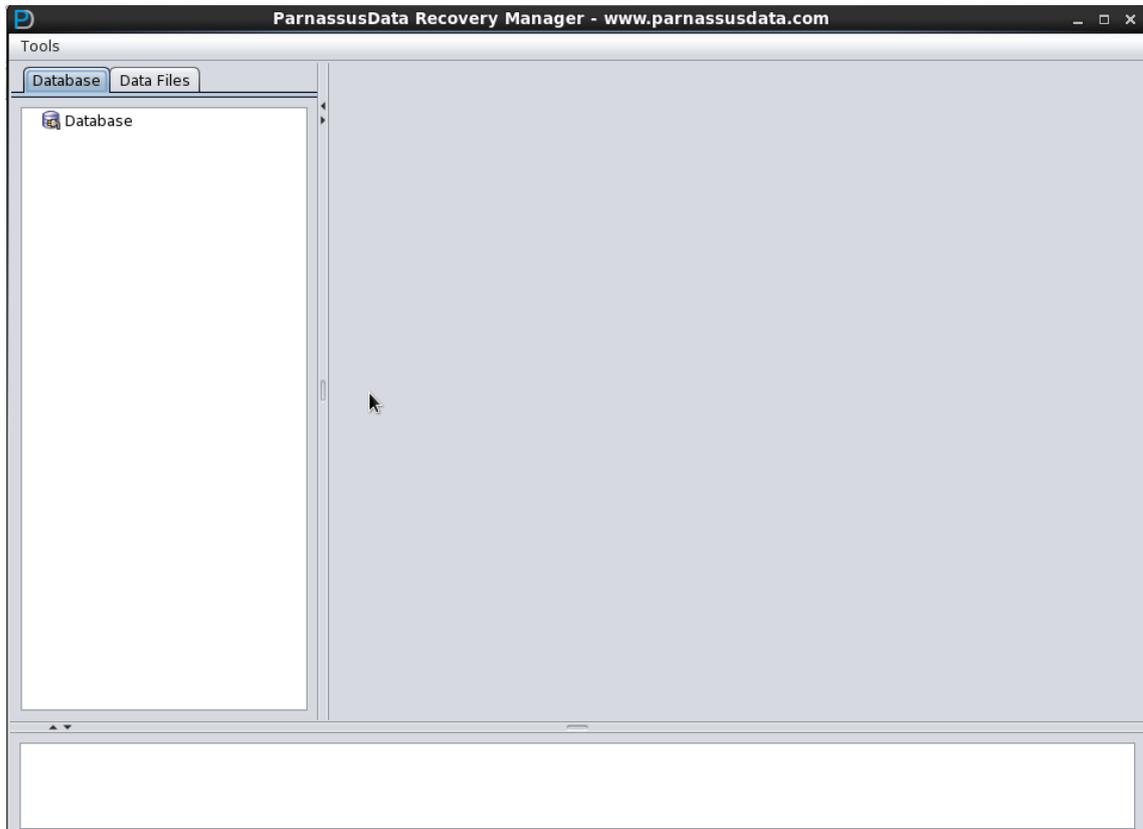
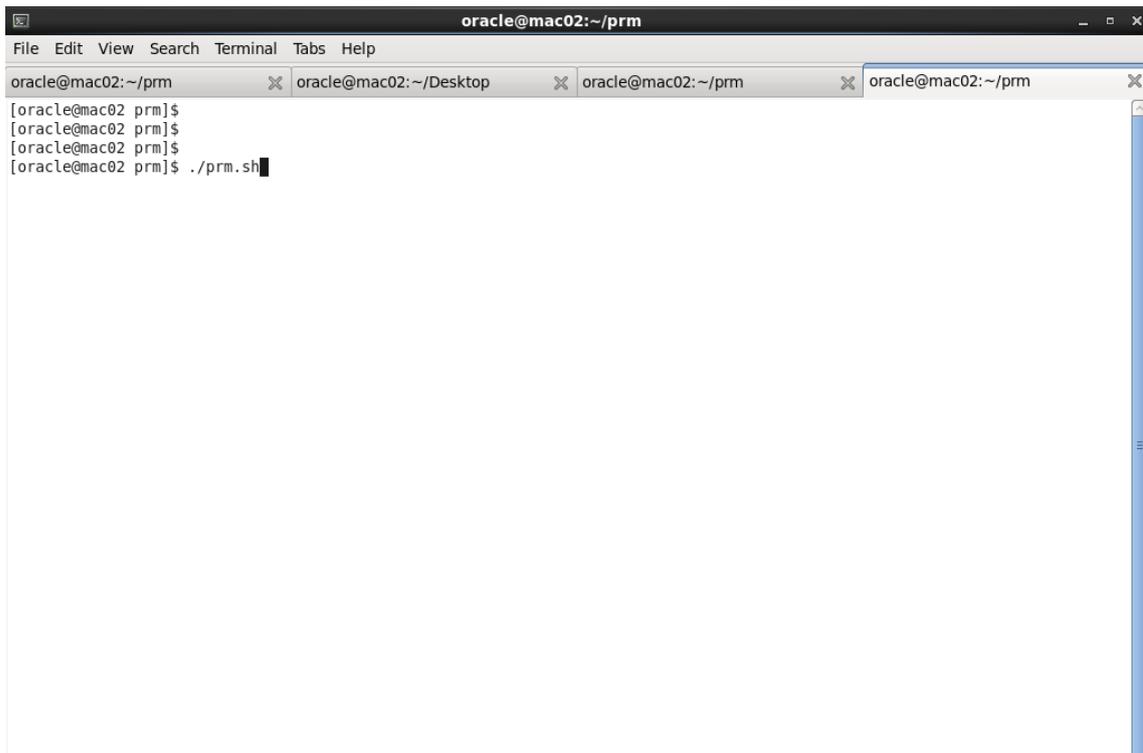
并启动 PRM 图形化主界面：



在 Linux/Unix 环境下的启动方法:

在 Linux/Unix 环境下可以在本机图形化界面或者通过 Xmanager 等远程图形化工具使用

1. 首先保证 JDK 已正确安装且 java 已经加入到环境变量中
2. Cd 到 PRM 所在目录，并执行 ./prm.sh 启动程序主界面



PRM 的许可证注册

ParnassusData Recovery Manager(以下简称 PRM)是一款商业软件。ParnassusData 提供 PRM 的社区版供用户测试和学习（社区版中 ASM clone 功能无任何限制，今后社区版将加入更多免费新特性）。

如想无限制地使用 PRM 软件恢复 ORACLE 数据库则需要购买对应的 License 软件许可证，目前我们提供 2 种 License 类型：Standard Edition 标准版和 Enterprise Edition 企业版；其具体规格如下：

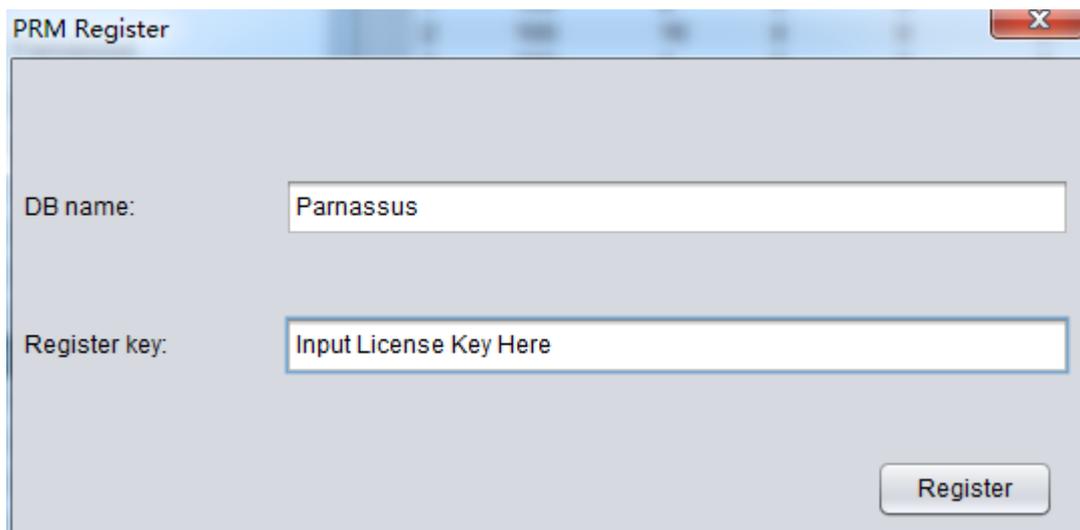
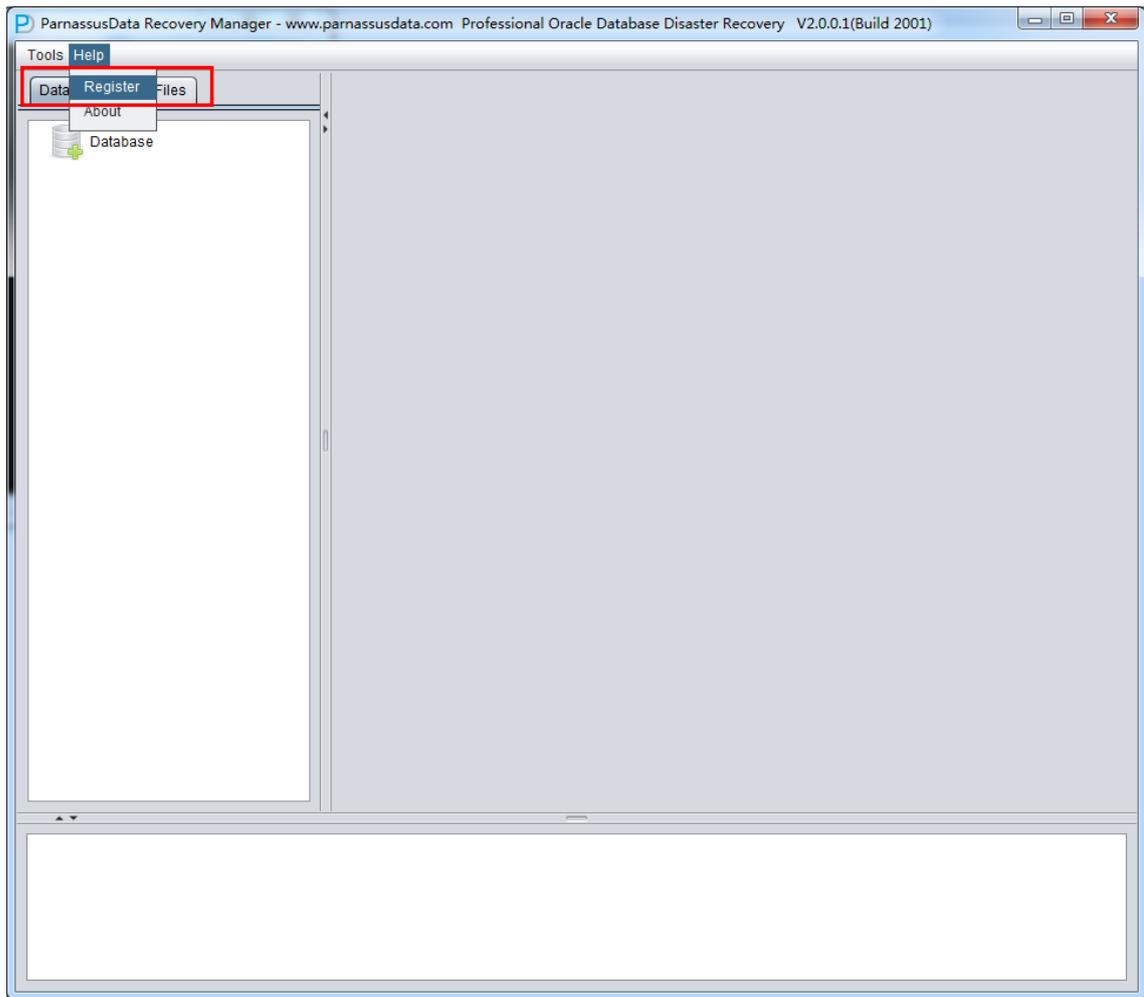
	社区版	标准版	企业版	企业服务
	FREE	\$299 PER DATABASE	\$999 PER DATABASE	面议
数据库支持大小	一万行限制	十万行限制	无限制	无限制
是否支持ASM功能	YES	YES	YES	YES
电话或者付费现场支持	NO	YES	YES	YES
	DOWNLOAD	BUY NOW	BUY NOW	BUY NOW

用户可以从 ParnassusData 官网 <http://www.parnassusdata.com/> 上购买 PRM License，购买时将需要您输入您要恢复的数据库名即 DBNAME，完成购买后您将受到一封邮件，该邮件中包括一个对应您输入的 DBNAME 和购买的许可证类型的 License Key。

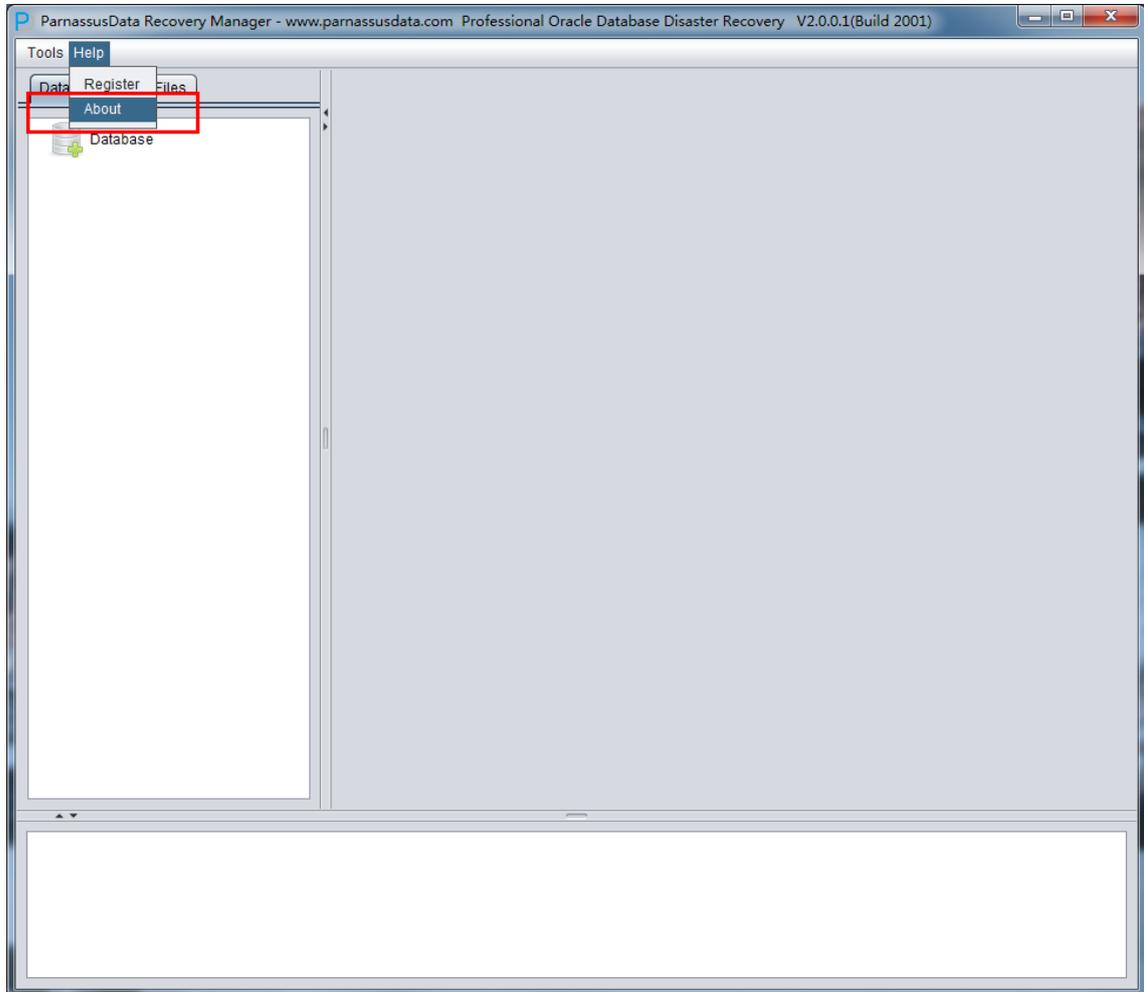
用户获得 License Key 之后可以自行在软件中注册 Register，具体使用方法为：

1. 在菜单栏 Help => Register
2. 输入 DB NAME 和发送给您的 License Key 并点击 Register 按钮即可

完成注册后，今后重新启动 PRM 将自动检测 License 注册信息，无需重复注册。



成功注册的信息可以在 Help=>About 中找到:





基于不同的 Oracle 数据库恢复场景介绍如何使用 PRM

恢复场景 1 误 Truncate 表的常规恢复

D 公司的业务维护人员由于误将产品数据库当做测试环境库导致错误地 TRUNCATE 了一张表上的所有数据，DBA 尝试恢复但是发觉最近的备份不可用，导致无法从备份中恢复出该数据表上的记录。此时 DBA 决定采用 PRM 来恢复已经被 TRUNCATE 掉的数据。

由于该环境中 所有数据库文件均是可用且健康的，用户仅需要 字典模式下加载 SYSTEM 表空间的数据文件以及被 TRUNCATED 表的数据文件即可，例如：

```
create table ParnassusData.torderdetail_his1 tablespace users as
select * from parnassusdata.torderdetail_his;
```

```
SQL> desc ParnassusData.TORDERDETAIL_HIS
```

Name	Null?	Type
SEQ_ID	NOT NULL	NUMBER(10)
SI_STATUS		NUMBER(38)
D_CREATEDATE		CHAR(20)
D_UPDATEDATE		CHAR(20)
B_ISDELETE		CHAR(1)
N_SHOPID		NUMBER(10)
N_ORDERID		NUMBER(10)
C_ORDERCODE		CHAR(20)
N_MEMBERID		NUMBER(10)

```
N_SKUID          NUMBER(10)
C_PROMOTION      NVARCHAR2(5)
N_AMOUNT         NUMBER(7,2)
N_UNITPRICE      NUMBER(7,2)
N_UNITSELLINGPRICE NUMBER(7,2)
N_QTY            NUMBER(7,2)
N_QTYFREE        NUMBER(7,2)
N_POINTSGET      NUMBER(7,2)
N_OPERATOR       NUMBER(10)
C_TIMESTAMP      VARCHAR2(20)
H_SEQID          NUMBER(10)
N_RETQTY         NUMBER(7,2)
N_QTYPOS         NUMBER(7,2)
```

```
select count(*) from
ParnassusData.TORDERDETAIL_HIS;
```

```
COUNT (*)
-----
      984359
```

```
select bytes/1024/1024 from dba_segments where
segment_name='TORDERDETAIL_HIS' and
owner='PARNASSUSDATA';
```

```
BYTES/1024/1024
-----
      189.71875
```

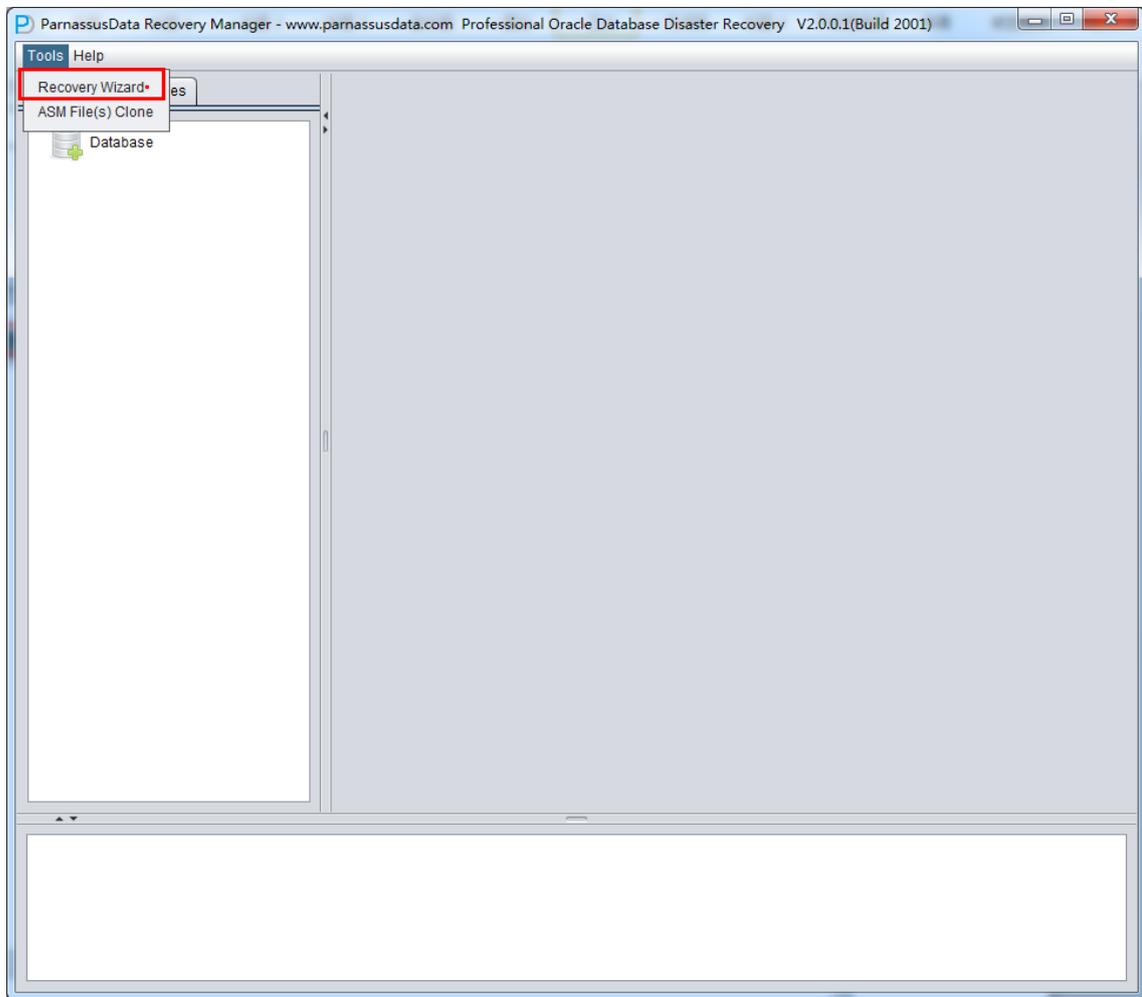
```
SQL> truncate table
ParnassusData.TORDERDETAIL_HIS;
```

```
Table truncated.
```

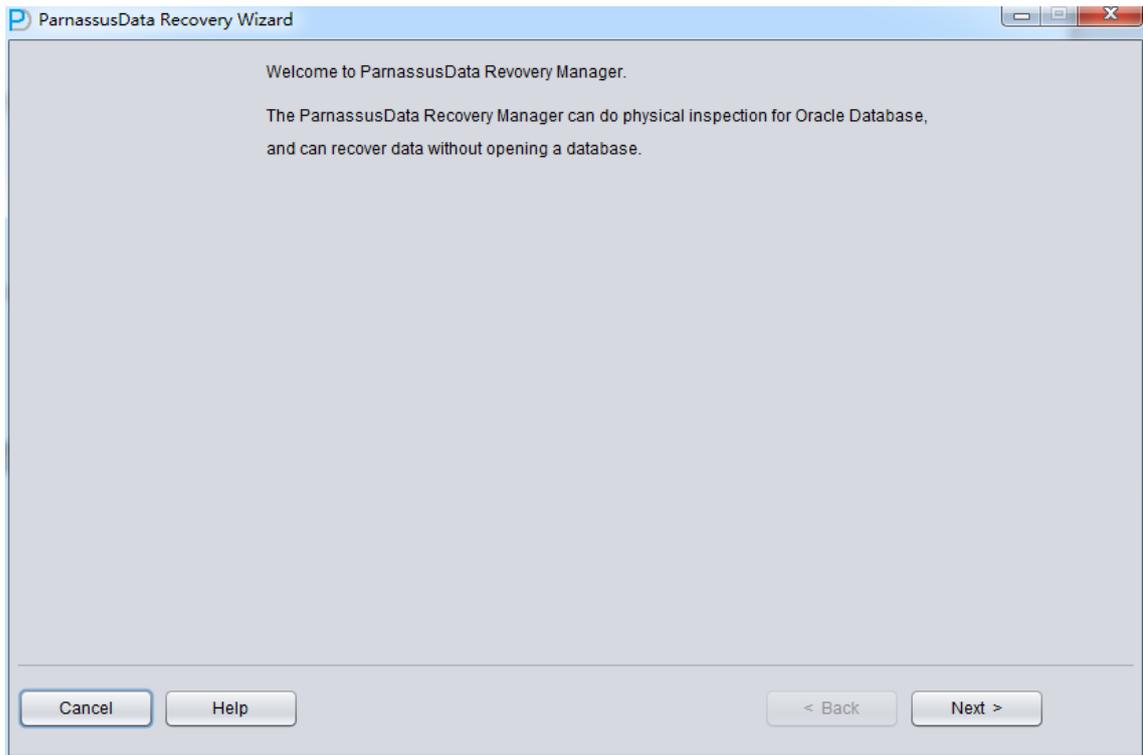
```
SQL> select count(*) from  
ParnassusData.TORDERDETAIL_HIS;
```

```
COUNT (*)  
-----  
0
```

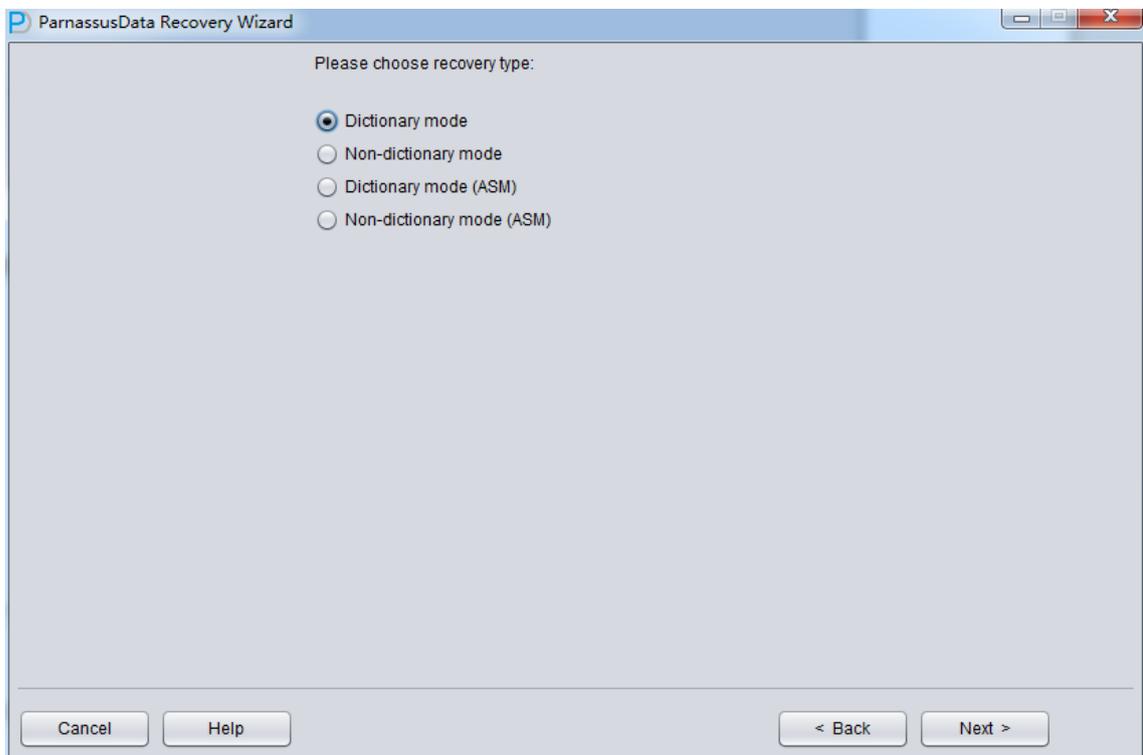
启动 PRM ， 并选择 Tools => Recovery Wizard



点击 Next



在此 TRUNCATE 场景中并未采用 ASM 存储，所以仅需要选择 《Dictionary Mode》字典模式即可：

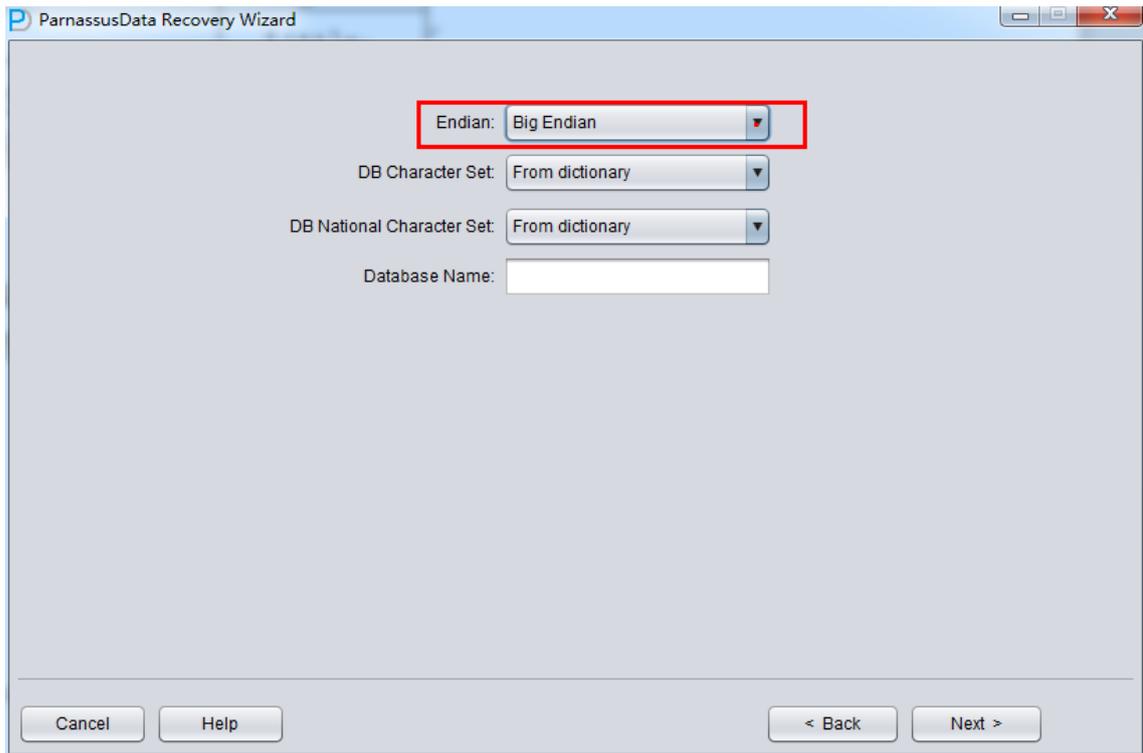


下一步骤 我们要选择几个参数 : 包括 Endian 字节序和 DB NAME

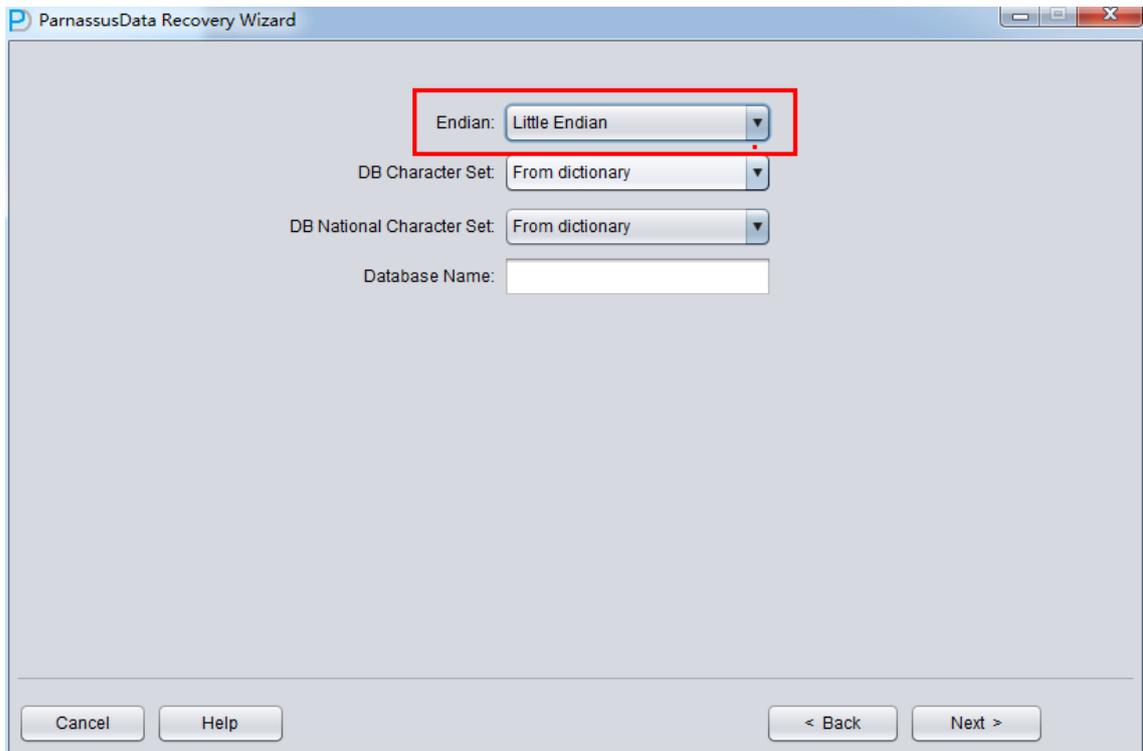
由于 ORACLE 数据文件在不同的操作系统平台上采用了不同的 Endian 字节序格式, 字节序和平台对应列表如下 :

Solaris[tm] OE (32-bit)	Big
Solaris[tm] OE (64-bit)	Big
Microsoft Windows IA (32-bit)	Little
Linux IA (32-bit)	Little
AIX-Based Systems (64-bit)	Big
HP-UX (64-bit)	Big
HP Tru64 UNIX	Little
HP-UX IA (64-bit)	Big
Linux IA (64-bit)	Little
HP Open VMS	Little
Microsoft Windows IA (64-bit)	Little
IBM zSeries Based Linux	Big
Linux x86 64-bit	Little
Apple Mac OS	Big
Microsoft Windows x86 64-bit	Little
Solaris Operating System (x86)	Little
IBM Power Based Linux	Big
HP IA Open VMS	Little
Solaris Operating System (x86-64)	Little
Apple Mac OS (x86-64)	Little

例如在传统 Unix AIX-Based Systems (64-bit) 、HP-UX (64-bit) 上使用的是 Big Endian 大端字节序, 则这里要选为 Big Endian:

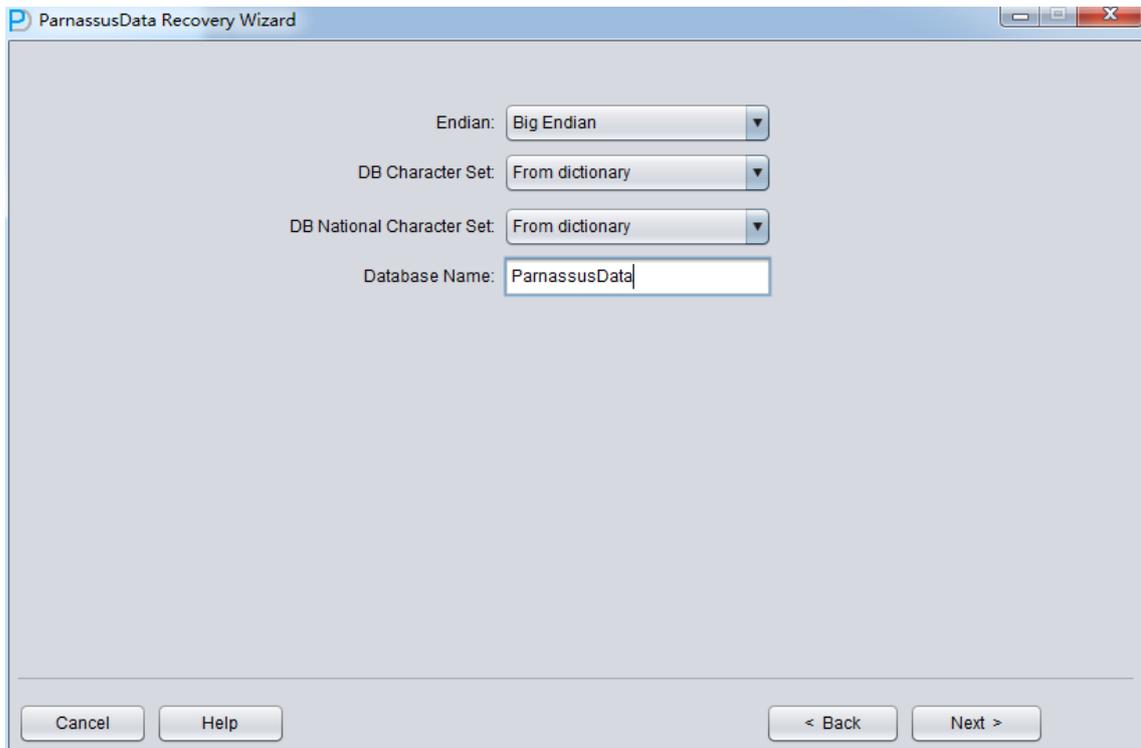


否则例如常见的 Linux x86-64 、 Windows 都保持为默认的 Little Endian:



注意事项：如果你的数据文件是在 AIX(即 Big Endian 的)上生成的，你为了方便而将这些数据文件拷贝到 Windows 服务器上并使用 PRM 来恢复数据，那么你仍应当选择其原生的 Big Endian 格式。

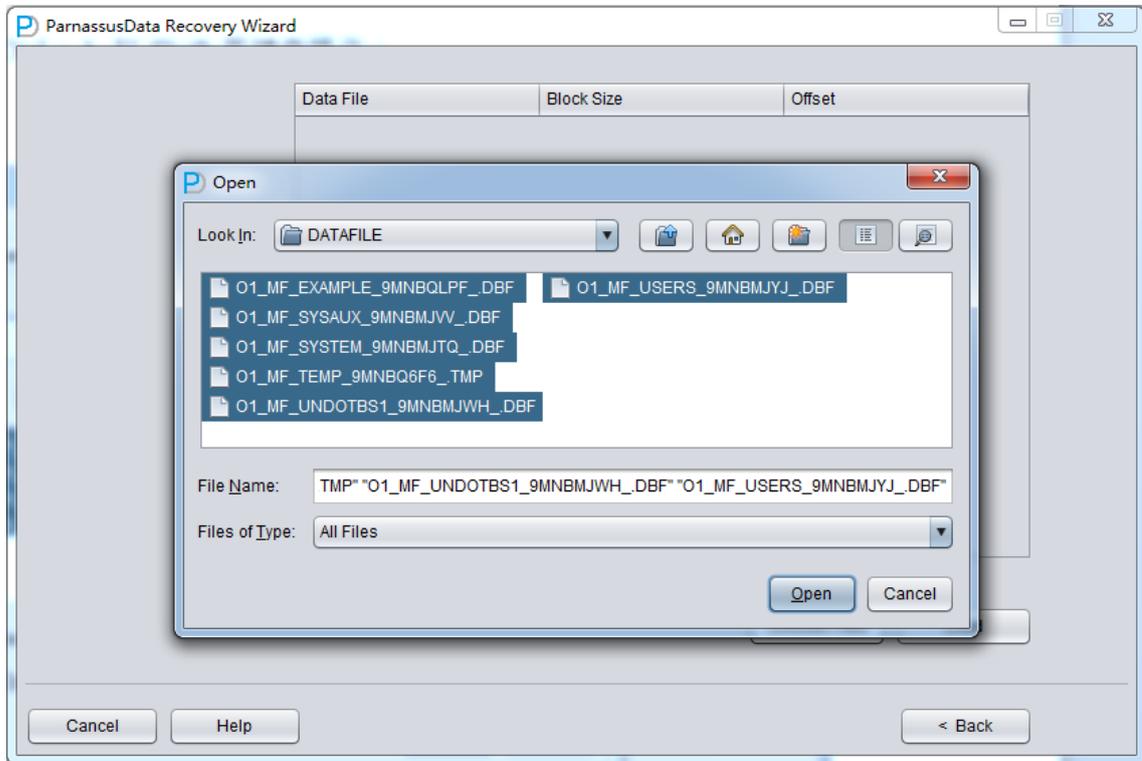
这里由于我们的数据文件是在 Linux x86 上所以我们选择 Endian 为 Little，并输入 Database name 数据库名字(注意这里输入的数据库名仅仅是一个别名，它不代表这个数据库真实的 DBNAME，PRM 的 LICENSE 检测机制使用的是真实的 DBNAME，而非此处输入的 Database Name)：

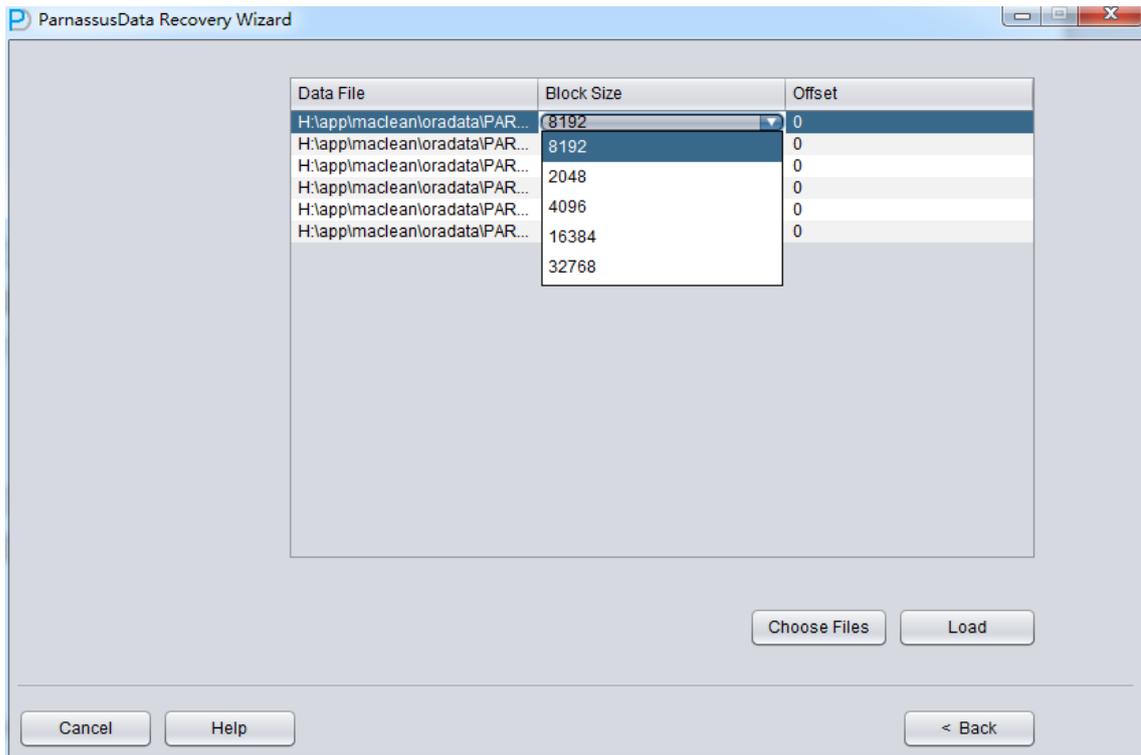


点击 Next

点击 Choose Files，一般我们推荐 如果数据库不大，那么将该库所有的数据文件都选择进来；如果你的数据库很大，且你了解你的数据表位于哪些数据文件上，则你可以仅仅选择 SYSTEM 表空间的数据文件(必须！)以及数据所在的数据文件。

注意 Choose 界面支持 Ctrl + A 和 Shift 等键盘操作 :





之后需要为指定的数据文件指定其 Block Size 即 ORACLE 数据块的大小，这里根据实际情况修改即可，例如你的 DB_BLOCK_SIZE 是 8K，但是部分表空间指定 16K 作为数据块大小的，仅仅需要为那些不是 8k 的数据文件修改 BLOCK_SIZE 即可。

这里的 OFFSET 参数主要是为了那些采用裸设备存放数据文件的场景，例如在 AIX 上基于普通 VG 的 LV 作为数据文件，则存在 4k 的 OFFSET，需要在此处指定。

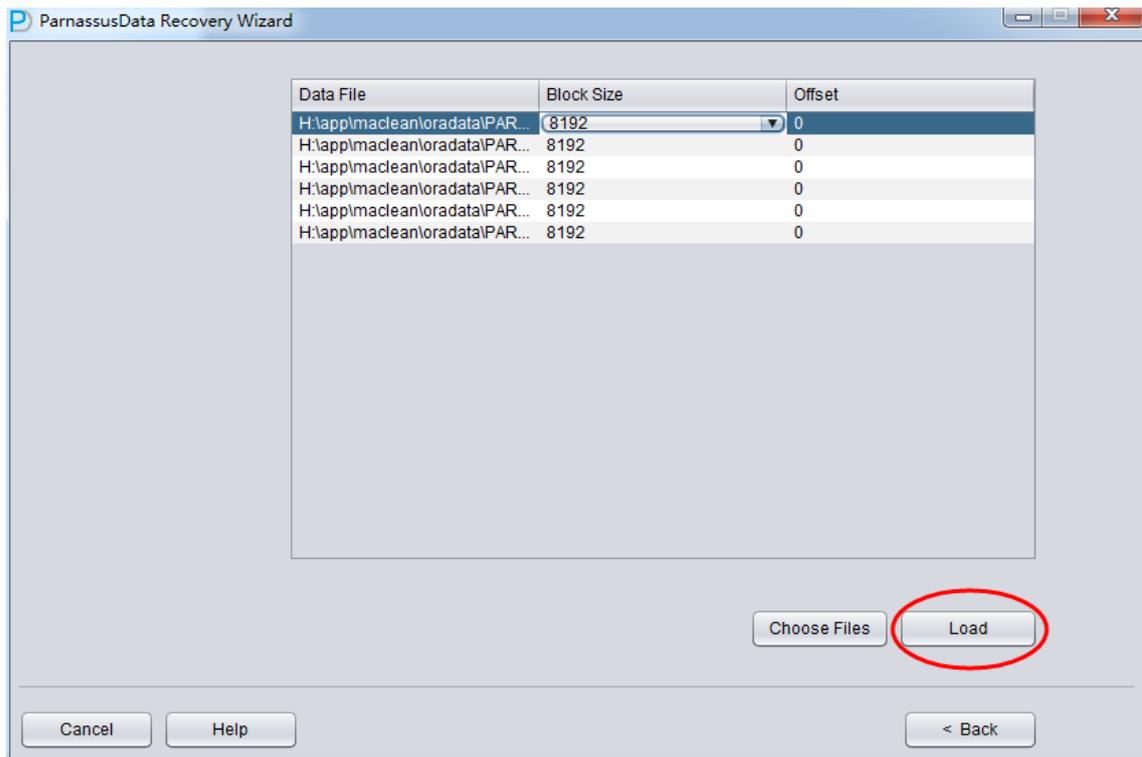
如果你恰巧正在使用裸设备数据文件，而又不知道 OFFSET 到底是多少？则可以使用 \$ORACLE_HOME/bin 下自带的 dbfsize 工具查看，如下面的例子高亮部分显示该裸设备具有 4K 的 OFFSET

```
$dbfsize /dev/lv_control_01

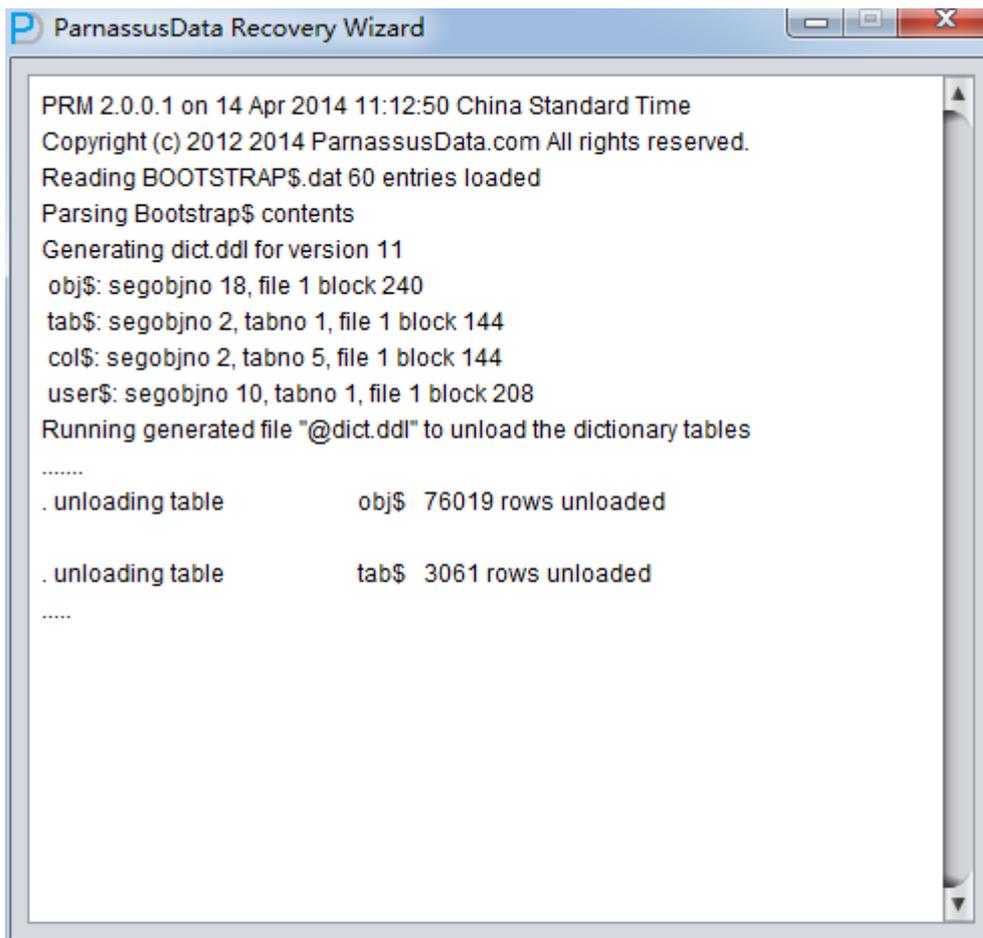
Database file: /dev/lv_control_01
Database file type: raw device without 4K starting offset
```

Database file size: 334 16384 byte blocks

由于此场景中所有数据文件均为 8K 的 BLOCK SIZE，且基于文件系统所以均没有 OFFSET，点击 Load



Load 阶段 PRM 会从 SYSTEM 表空间中读取 ORACLE 数据字典信息，并在自带的 Derby 中自建一个数据字典，这让 PRM 有能力操作 ORACLE 数据库中的各种数据。



Load 完成后会在后台输出数据库 字符集和国家字符集等信息：

```

oracle@mlab2:~/prm
File Edit View Terminal Tabs Help
k file header:
DB Software version: 00 00 00 00
DB Compatibility version: 0B 20 00 00
DB id: 2823240832
DB Name: ASMME
File logical blocks count: 49603
Parsing /s01/oradata/ASMME/data_D-ASMME_I-2823240832_TS-SYS_UNDOTS_FNO-3_06oubs4
d.bak file header:
DB Software version: 00 00 00 00
DB Compatibility version: 0B 20 00 00
DB id: 2823240832
DB Name: ASMME
File logical blocks count: 25888
Parsing /s01/oradata/ASMME/data_D-ASMME_I-2823240832_TS-USERS_FNO-5_07oubs4s.bak
file header:
DB Software version: 00 00 00 00
DB Compatibility version: 0B 20 00 00
DB id: 2823240832
DB Name: ASMME
File logical blocks count: 192000
Database character set is US7ASCII
Database national character set is AL16UTF16
Current character set for decoding is ASCII
Current national character set for decoding is UTF16

```

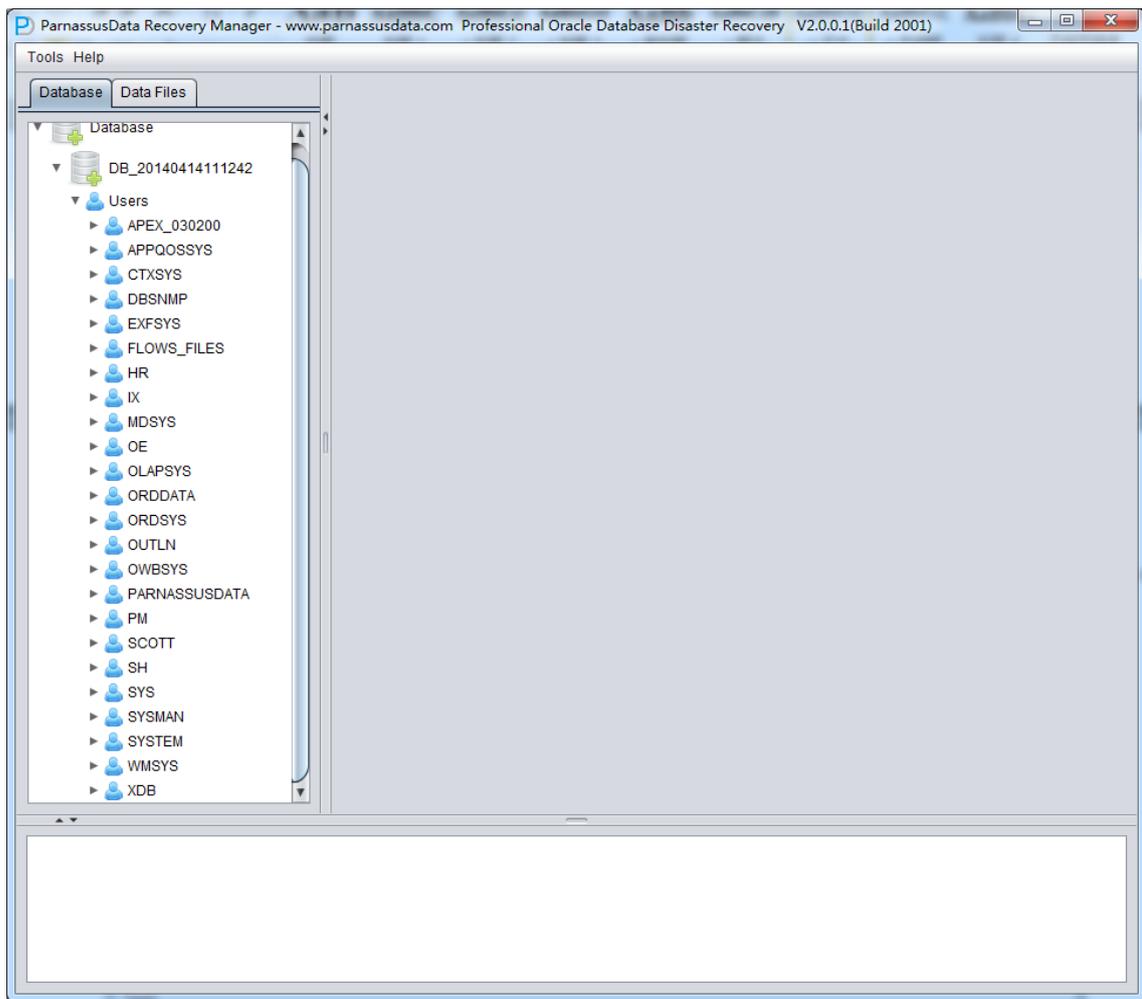
注意 PRM 是支持 多语言和 ORACLE 数据库的多字符集的，但是前提是实施 PRM 数据恢复的操作系统要求已经安装了对应的语言包；例如在 Windows 操作系统上没有安装中文语言包，但是由于 ORACLE 数据库字符集是独立于操作系统语言的，即 ORACLE 数据库的字符集可以为 ZHS16GBK 字符集，但是操作系统并不支持中文，此场景中不在本服务器上部署的 ORACLE 客户端并不受影响，可以正确显示数据库中的中文数据。

但是使用 PRM 则要求实施 PRM 数据恢复的操作系统已经安装了对应的语言包，例如用户要恢复 ZHS16GBK 的中文字符集数据库，则需要操作系统上已经安装了中文语言包才可以。

类似的 在 Linux 上需要安装 fonts-chinese 中文字体包。

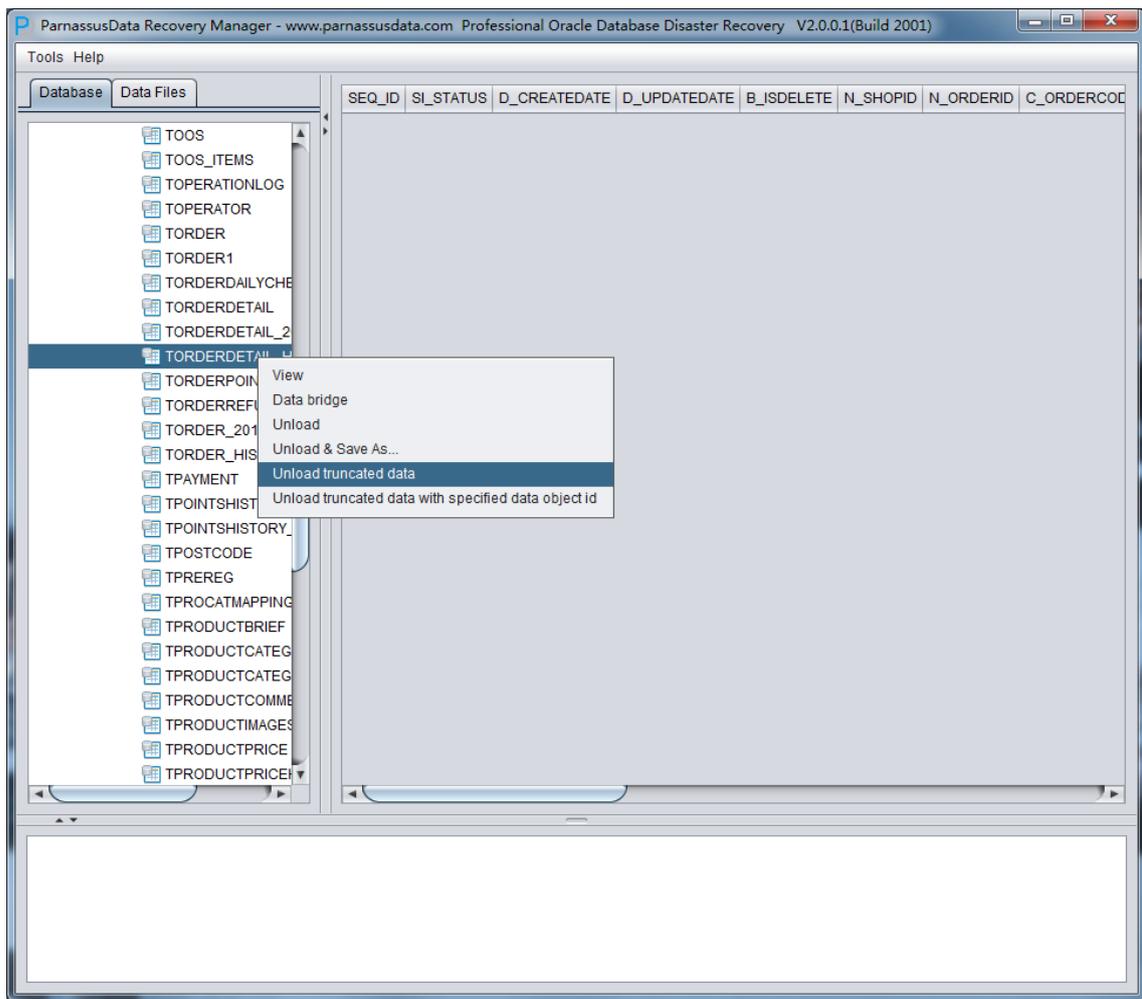
Load 完成后 PRM 界面左侧出现按照数据库用户分组的树形图

点开 USERS，可以看到多个用户名，例如用户需要恢复 PARNASSUSDATA SCHEMA 下的一张表，则点开 PARNASSUSDATA，并双击表名：

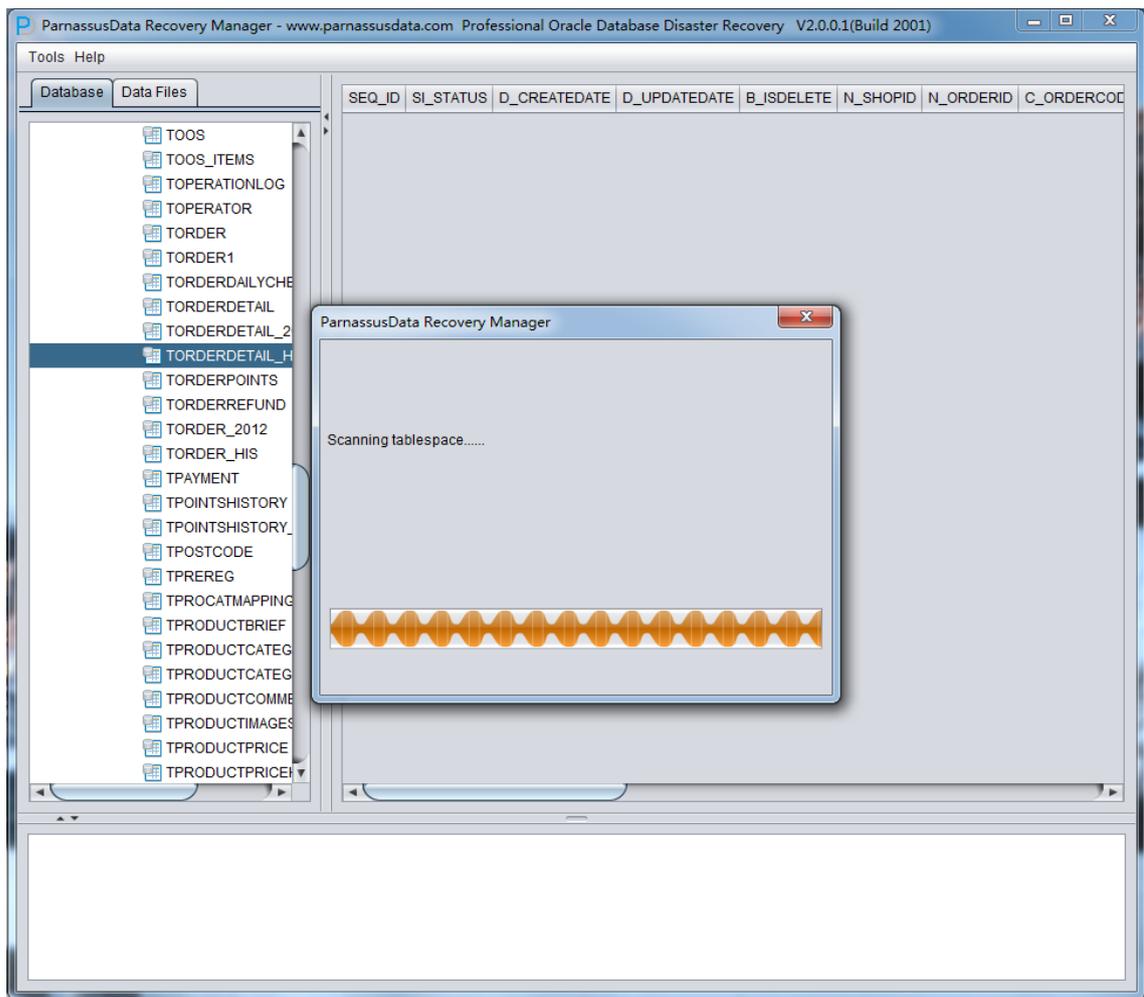


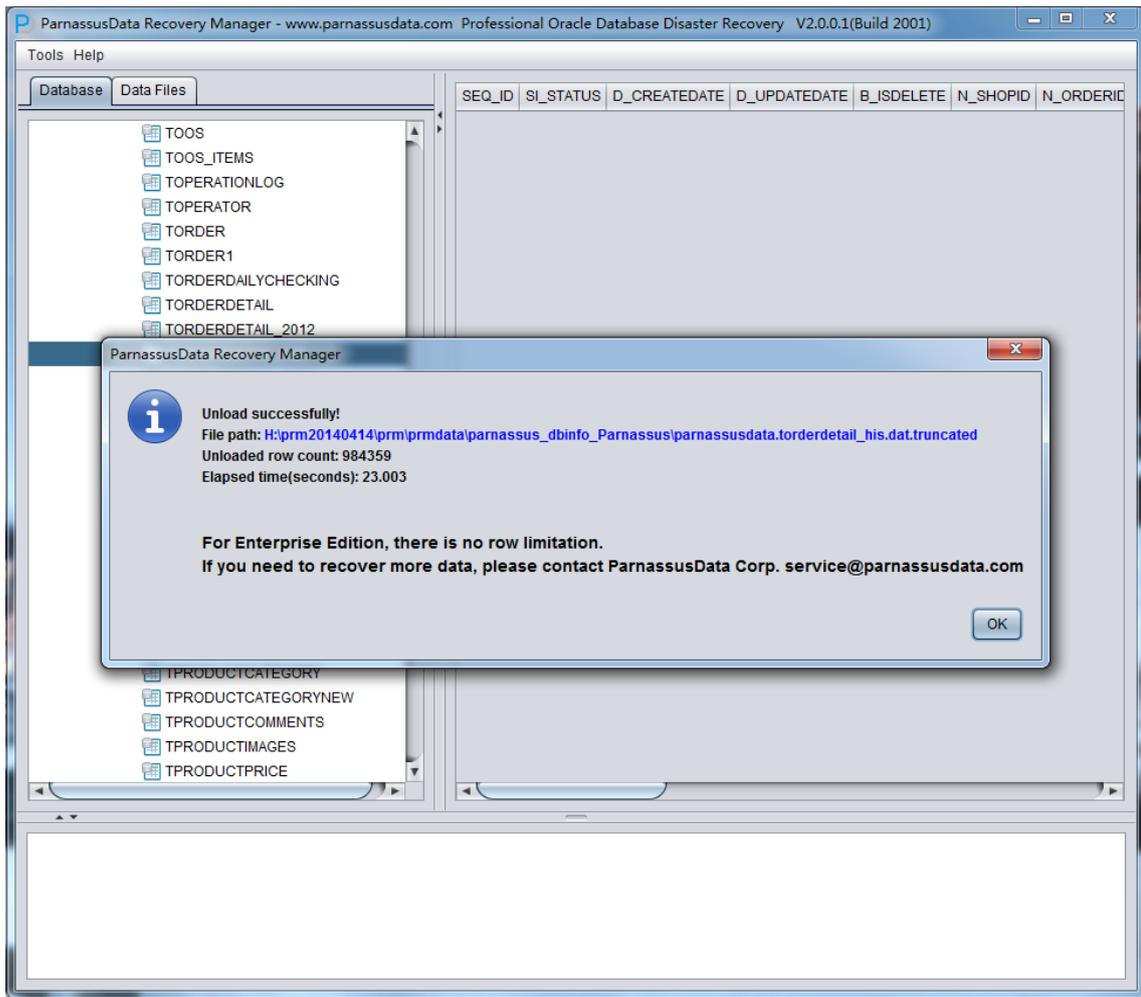
由于该 TORDERDETAIL_HIS 表之前已经被 TRUNCATED 掉了，所以双击没有显示有数据，此时在表上右键选择 Unload truncated data

:



PRM 将尝试扫描该表所在表空间并将已经 truncated 掉的数据抽取出来：





如上图所示从已经被 TRUNCATE 过的 TORDERDETAIL_HIS 表中抽取出完整的 984359 条记录，并存放在提示指定的路径下。

这里还自动生成了将文本数据导入到数据库中使用的 SQLLDR 控制文件。

```
[oracle@mlab2 ~]$ cd
/home/oracle/prm/prmdata/parnassus_dbinfo_PARNASSUSDATA/

[oracle@mlab2 parnassus_dbinfo_PARNASSUSDATA]$ ls -l ParnassusData*
-rw-r--r-- 1 oracle oinstall      495 Jan 18 08:31 ParnassusData.torderdetail_hisctl
-rw-r--r-- 1 oracle oinstall 191164826 Jan 18 08:32
ParnassusData.torderdetail_his.dat.truncated
```

```
[oracle@mlab2          parnassus_dbinfo_PARNASSUSDATA]$          cat
ParnassusData.torderdetail_his.ctl
LOAD DATA
INFILE 'ParnassusData.torderdetail_his.dat.truncated'
APPEND
INTO TABLE ParnassusData.torderdetail_his
FIELDS TERMINATED BY ' '
OPTIONALLY ENCLOSED BY '"'
TRAILING NULLCOLS (
"SEQ_ID" ,
"SI_STATUS" ,
"D_CREATEDATE" ,
"D_UPDATEDATE" ,
"B_ISDELETE" ,
"N_SHOPID" ,
"N_ORDERID" ,
"C_ORDERCODE" ,
"N_MEMBERID" ,
"N_SKUID" ,
"C_PROMOTION" ,
"N_AMOUNT" ,
"N_UNITPRICE" ,
"N_UNITSELLINGPRICE" ,
"N_QTY" ,
"N_QTYFREE" ,
"N_POINTSGET" ,
"N_OPERATOR" ,
"C_TIMESTAMP" ,
"H_SEQID" ,
"N_RETQTY" ,
"N_QTYPOS"
)
```

将数据导入到源表中(注意 ParnassusData 强烈建议你修改该 SQLLDR 控制文件中导入的表名字为一个临时表, 这样不会覆盖原环境)。

```
[oracle@mlab2          parnassus_dbinfo_DB_20140119002120]$          sqlldr  
control=ParnassusData.torderdetail_his.ctl direct=y  
Username:/ as sysdba
```

```
select * from ParnassusData.torderdetail_his minus select * from  
parnassus.torderdetail_his;
```

```
no rows selected
```

测试 TRUNCATE 用例表与源数据表对比，发现记录完全一致。
说明 PRM 完整、丝毫不差地恢复了被 TRUNCATE 表上的记录。

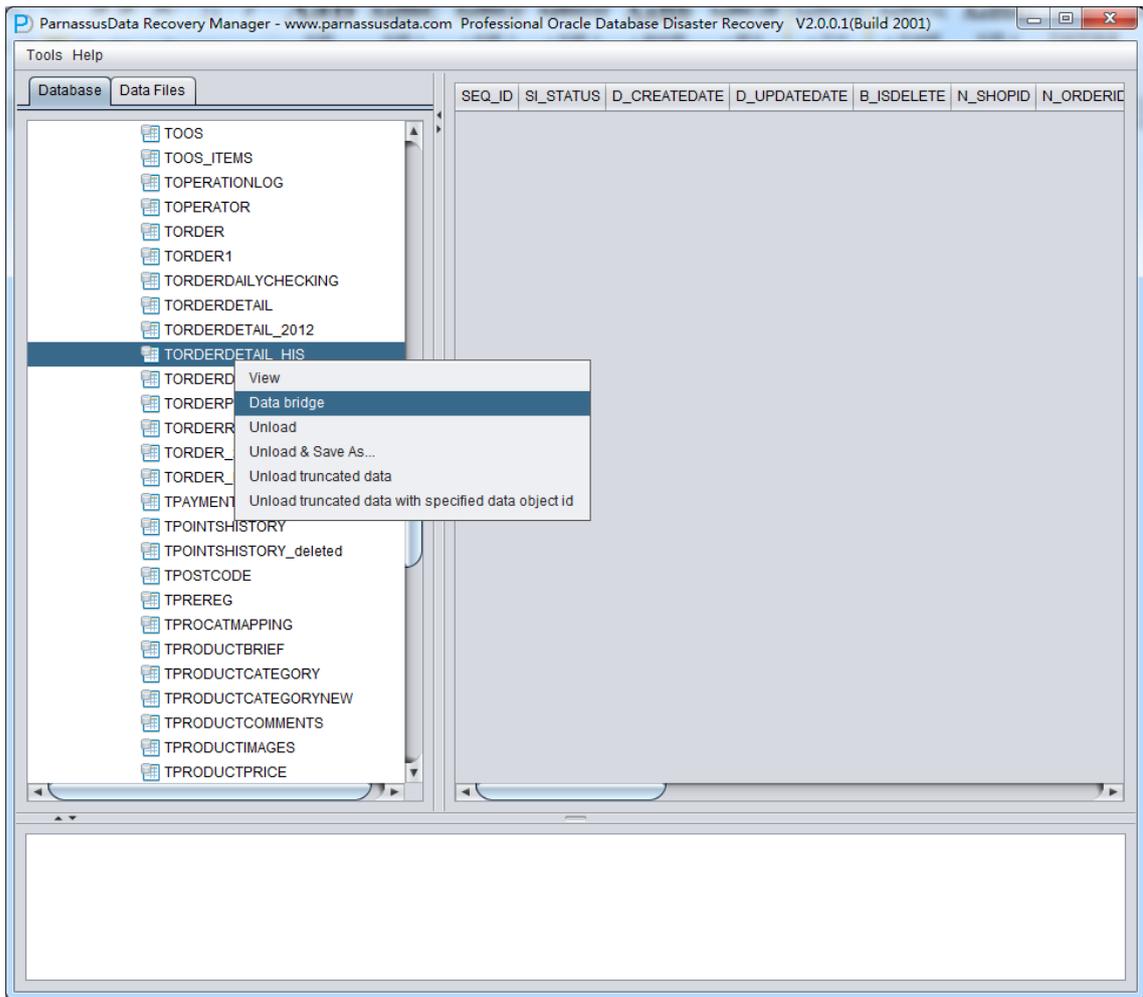
恢复场景 2 误 Truncate 表的 DataBridge 数据搭桥恢复

恢复场景 1 中我们采用了常规 unload+sqlldr 的恢复方式；但实际上 ParnassusData 原厂更推荐您使用我们精心设计的 DataBridge 数据搭桥模式。

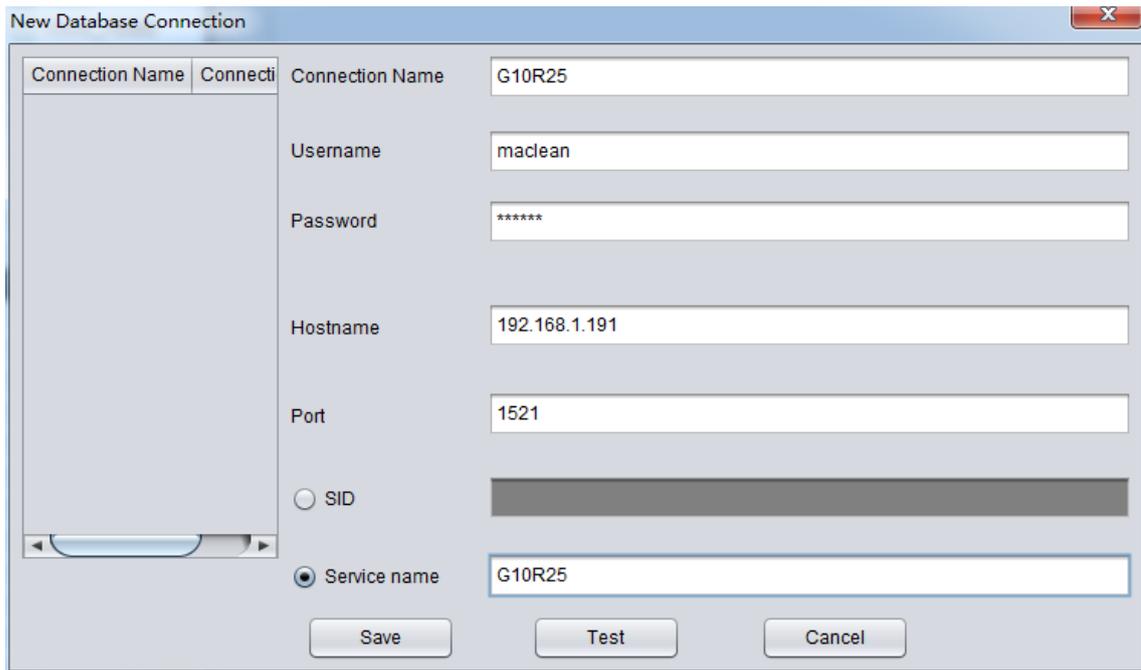
为什么要引入数据搭桥模式呢？

- 普通的 unload+sqlldr 恢复方式意味着要保存一份源数据，一份抽取数据，和一份目标数据，即在恢复过程中可能需要扩容 2 倍于原来的存储空间，这对于甚至无法腾出备份空间的企业来说十分困难
- 数据搭桥与普通 unload+sqlldr 模式的最大区别在于，数据搭桥直接从源库中抽取数据并传送到目标数据库中，无需在文件系统中保留一份抽取数据
- 通过数据搭桥传送到目标数据库中的数据本身就是结构化的，可以立即使用 SQL 语句来验证其完整性和一致性
- 如果数据搭桥的目标数据库位于异机上，那么源数据库上仅仅做读取操作，读写 IO 将分布于 2 台服务器上，PRM 恢复的速度将更快
- 如果用户所需要恢复的是 Truncate 数据的话，那么可以马上搭桥回到源库中，恢复仅仅是鼠标点几下的工作

使用数据搭桥模式也十分简便，通常模式一样，在左侧树形图中选中你需要的表，右键选择 DataBridge 选项：

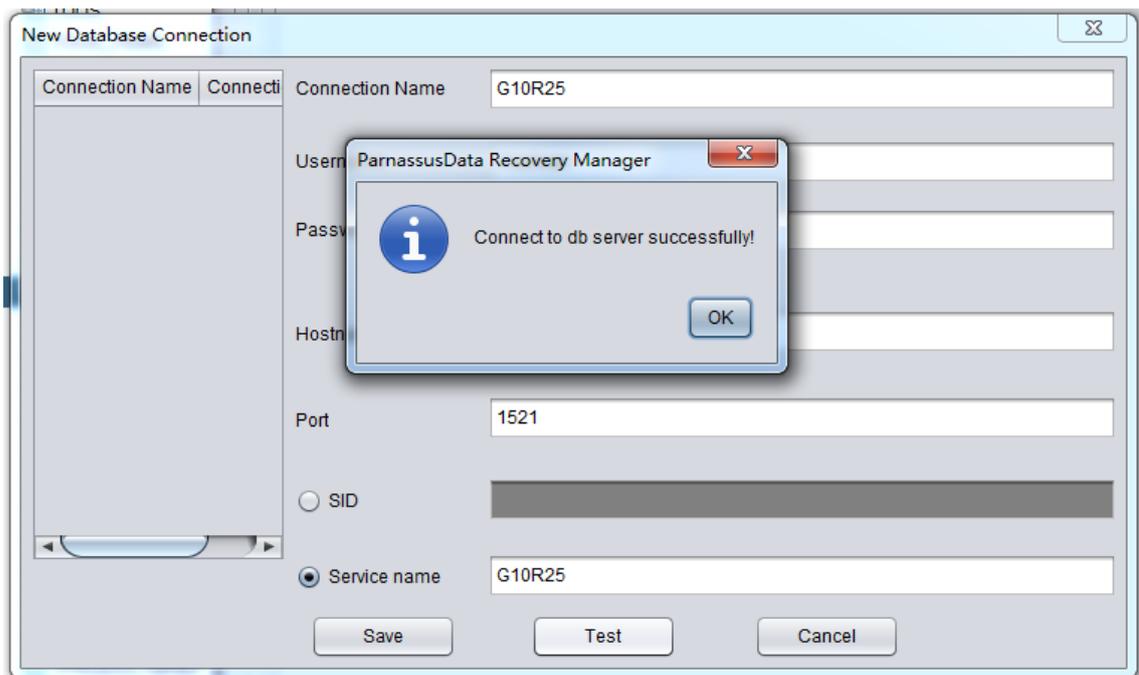


首次使用数据搭桥模式时需要先创建目标数据库连接信息，这就和我们在 SQLDEVELOPER 中创建一个 Connection 是类似的工作，包括目标数据库的 Host、端口、Service_Name 以及用户登录信息；注意这里填选的用户信息，将会是稍后数据搭桥使用的目标数据库的 User 用户，即从源库这里抽取出来的表会传输到目标数据库中此处所指定的用户名下。

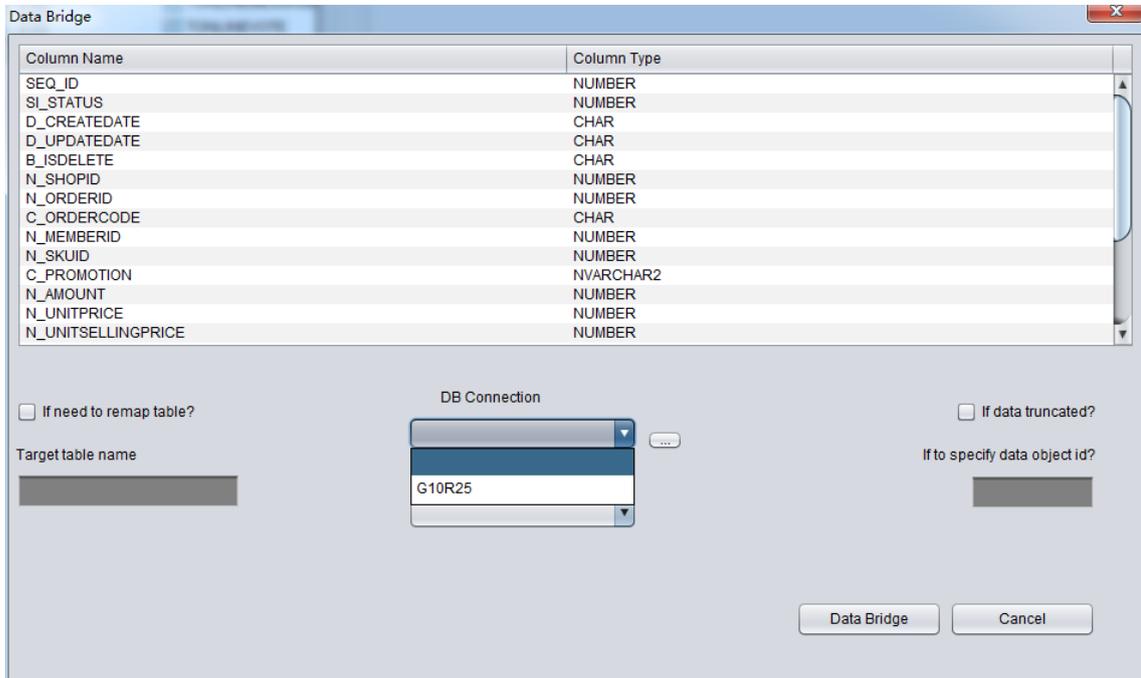


如上述建立了一个 G10R25 的连接，用户为 maclean，对应的 oracle Easy Connection 连接串为 192.168.1.191:1521/G10R25。

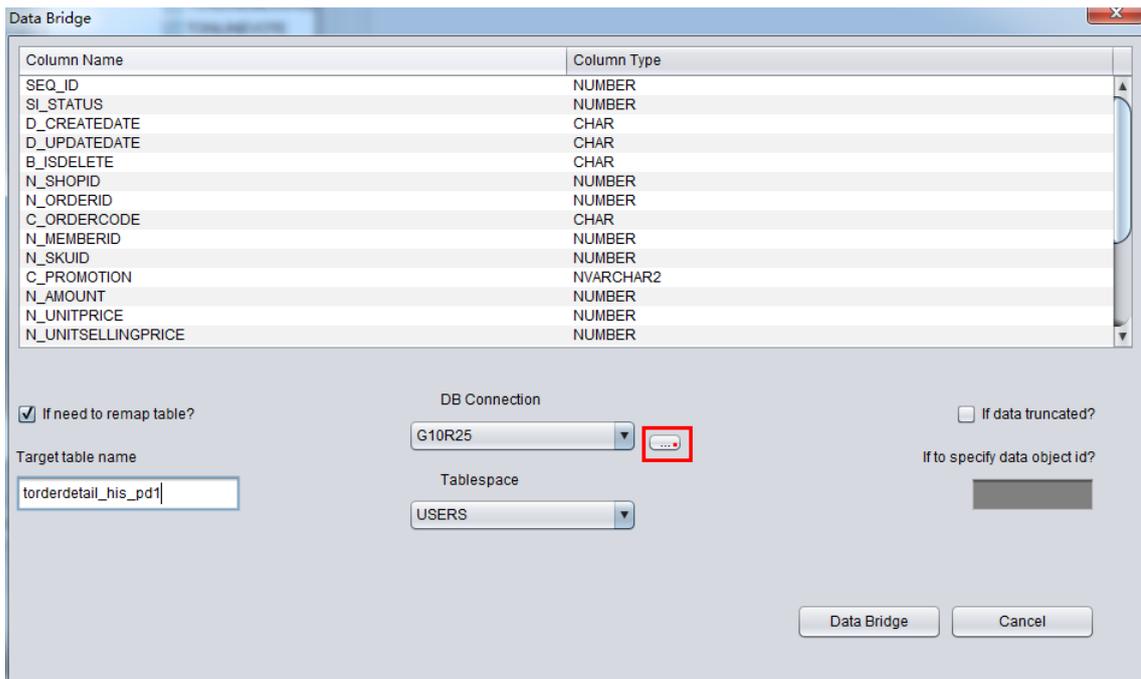
完成上述数据库连接信息填写后可以点击 Test 按钮来测试该连接配置是否正确可用，如果返回 “Connect to db server successfully” 则说明连接可用，点击 Save 按钮保存即可。



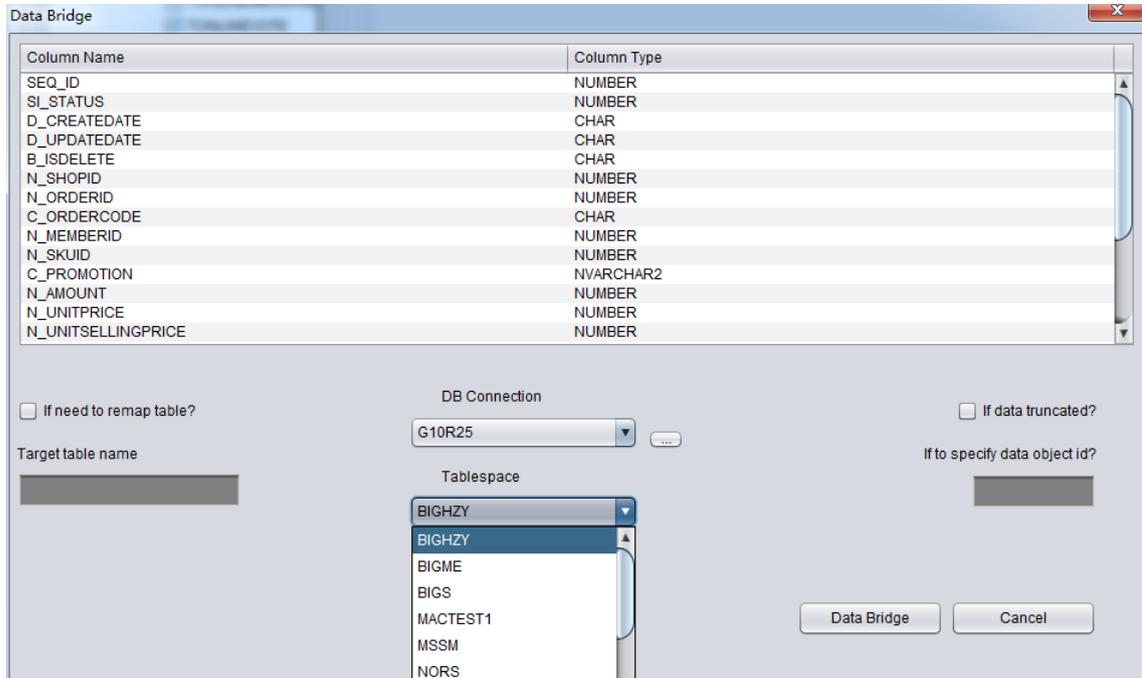
Save 后进入 DataBridge 主界面，首先在 DB Connection 下拉框中选择刚刚加入的 Connection G10R25:



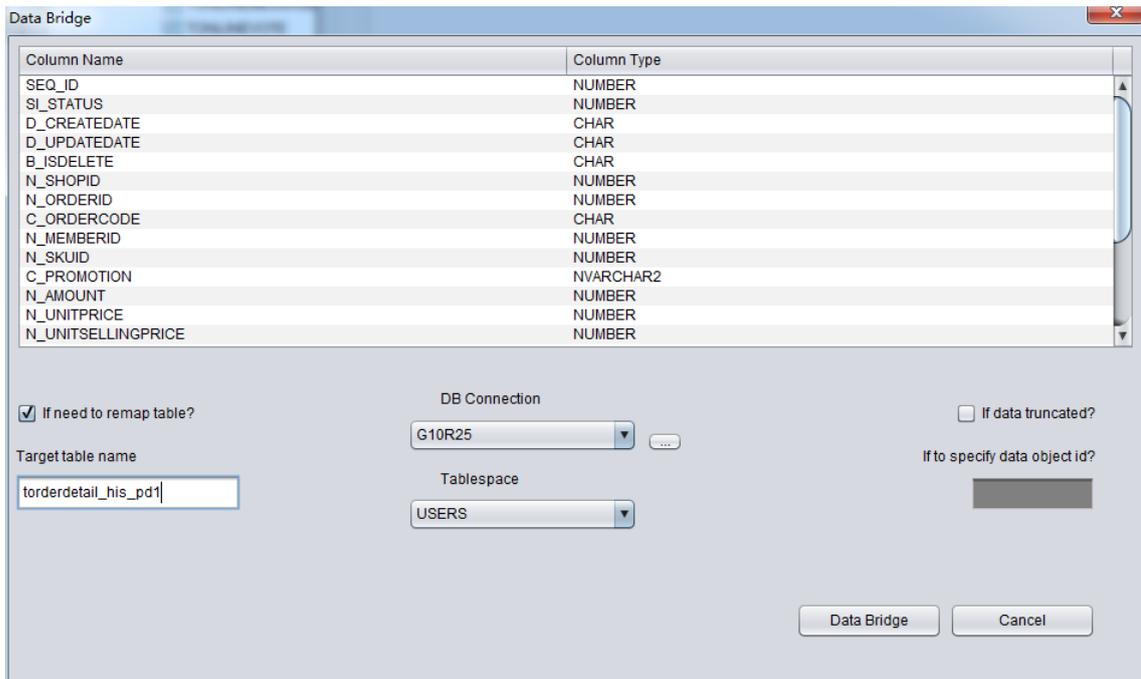
此处如果所需用的数据库连接并未在 DB connection 下拉框中出现，则需要点击 DB connection 旁的”...”按钮添加 DB Connection:



正确选择 DB Connection 后可以 Tablespace 的下拉框将变得可用，选中合适的表空间：



用户可以选择是否要将从源库传输到目标库的表的表名做映射修改，例如我们在源库中 Truncate 掉了一张表，现在通过 DataBridge 将数据恢复回源库中，但是不想使用原来的表名字，如原来的表名为 torderdetail_his，现在希望将恢复的数据以别的表名存放，则可以选中 “if need to remap table” 并填入合适的目标表名，如下图所示：



注意： 1) 对于目标库中已经存在对应表名的情况，PRM 不会重建表而是会在现有表的基础上插入所需恢复的数据，由于表已经建立了所以指定的表空间将无效 2) 对于目标数据库中还不存在对应表名的情况，PRM 会尝试在指定表空间上建表并插入恢复数据

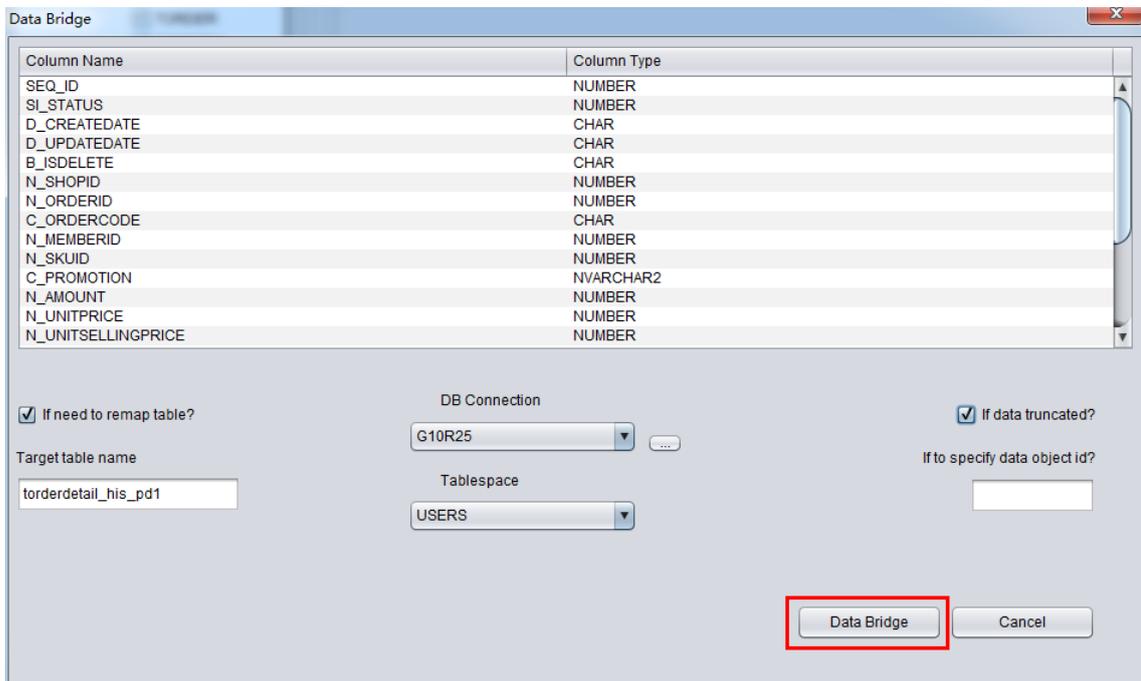
此场景中由于我们是恢复 Truncate 掉的数据，所以需要选中“if data truncated”选项，否则 PRM 将以常规模式抽取数据，将无法抽取到已经被 Truncate 掉的数据。

Truncate 数据的大致机理是，ORACLE 会在数据字典和 Segment Header 中更新表的 Data Object ID，而实际数据部分的块则不会做修改。由于数据字典与段头的 DATA_OBJECT_ID 与后续的数据块中的并不一致，所以 ORACLE 服务进程在读取全表数据时不会读取到已经被 TRUNCATE 但是实际仍未被覆盖的数据。

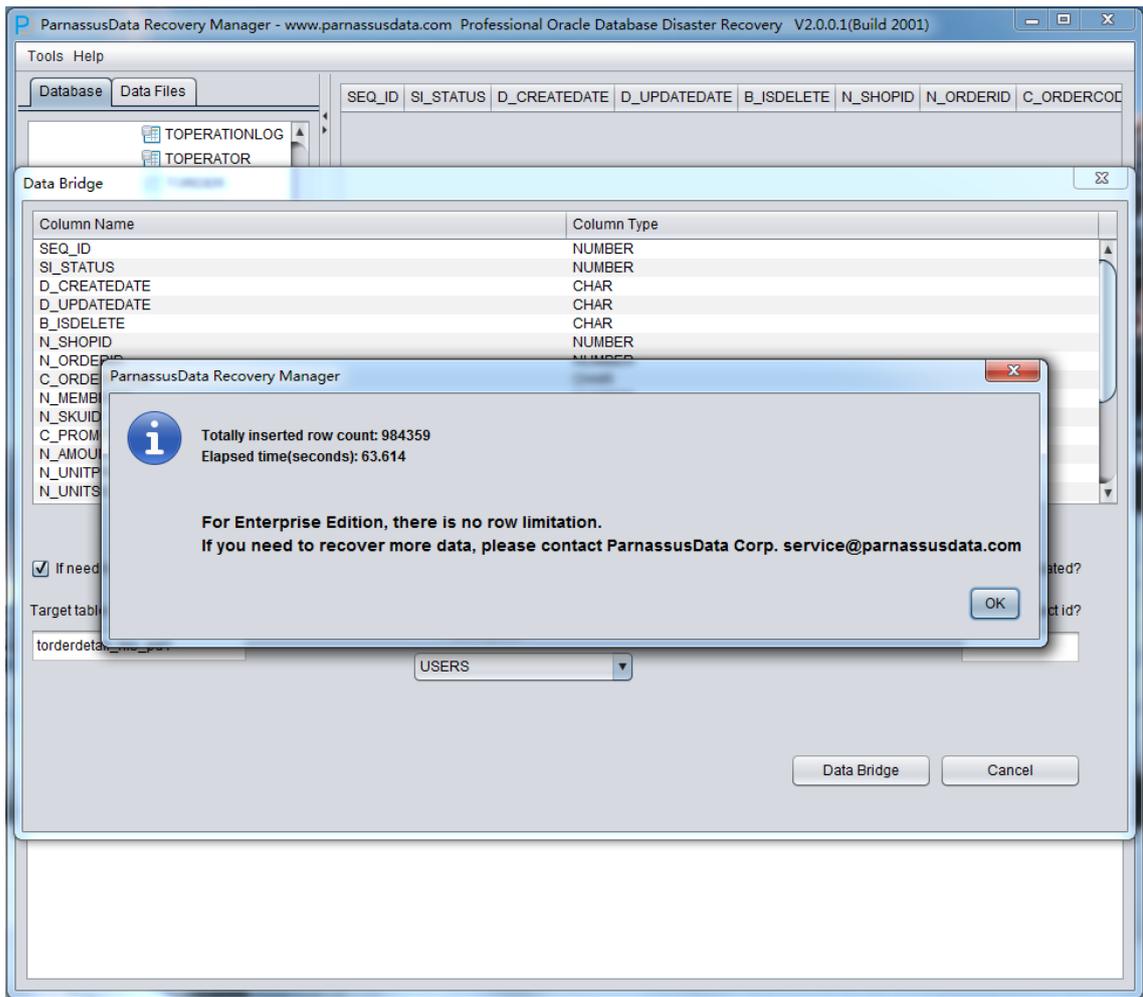
PRM 通过自动扫描被 TRUNCATE 掉数据段头 Segment Header 后续的数据块智能判断 TRUNCATE 前数据段的 DATA_OBJECT_ID，并根据字典中的表字段定义和自动获得的原始 DATA_OBJECT_ID 来抽取数据。

此处还存在一个”if to specify data object id”输入框，该输入框可以让用户指定要恢复的数据的 Data Object ID。一般情况下不需要指定任何值，除非你发现恢复 Truncate 数据不成功时，建议在 ParnassusData 原厂工程师的帮助下指定该值。

如上正确完成 DataBridge 配置后即可证实开始数据搭桥，只需要点击 DataBridge 按钮即可：



数据搭桥完成后会显示成功传输的数据行数，以及耗时。



恢复场景 3 ORACLE 数据字典受损导致数据库无法打开

D 公司的 DBA 由于误操作删除了 TS\$数据字典基表导致数据库无法启动

Oracle Database 11g Enterprise Edition Release 11.2.0.3.0 - 64bit Production
With the Partitioning, Automatic Storage Management, OLAP, Data Mining

and Real Application Testing options

INSTANCE_NAME

ASMME

SQL>

SQL>

SQL> select count(*) from sys.ts\$;

COUNT(*)

5

SQL> delete ts\$;

5 rows deleted.

SQL> commit;

Commit complete.

SQL> shutdown immediate;

Database closed.

Database dismounted.

ORACLE instance shut down.

Database mounted.

ORA-01092: ORACLE instance terminated. Disconnection forced

ORA-01405: fetched column value is NULL

Process ID: 5270

Session ID: 10 Serial number: 3

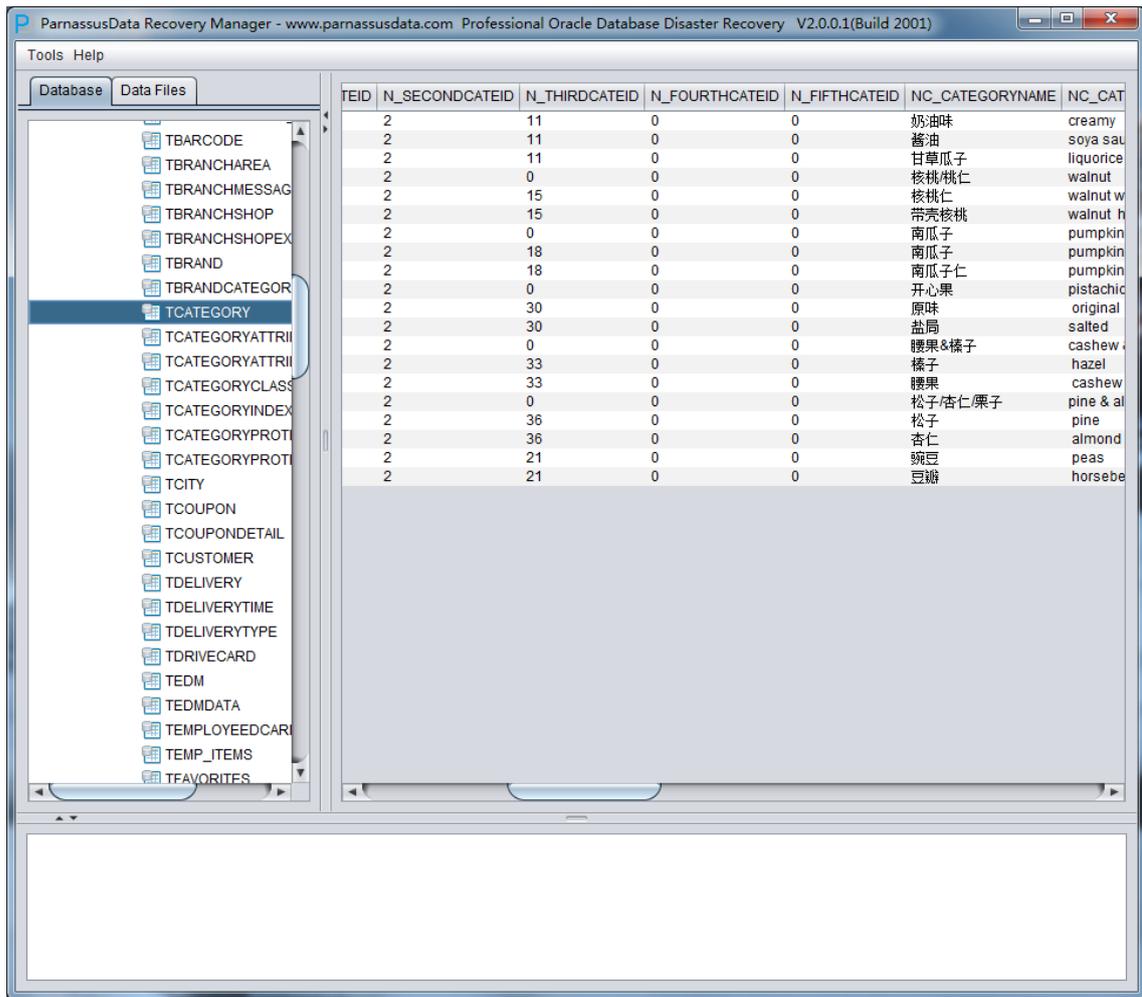
Undo initialization errored: err:1405 serial:0 start:3126020954 end:3126020954 diff:0
(0 seconds)

```
Errors in file /s01/diag/rdbms/asmme/ASMME/trace/ASMME_ora_5270.trc:
ORA-01405: fetched column value is NULL
Errors in file /s01/diag/rdbms/asmme/ASMME/trace/ASMME_ora_5270.trc:
ORA-01405: fetched column value is NULL
Error 1405 happened during db open, shutting down database
USER (ospid: 5270): terminating the instance due to error 1405
Instance terminated by USER, pid = 5270
ORA-1092 signalled during: ALTER DATABASE OPEN...
opiodr aborting process unknown ospid (5270) as a result of ORA-1092
```

此场景中由于数据字典已经损坏，所以想要正常打开数据库是十分困难的。

此时则可以使用 PRM 来抽取数据库中的数据。具体步骤与场景 1 中的相似，用户仅仅需要输入该数据库的所有数据文件即可，其简要步骤如下：

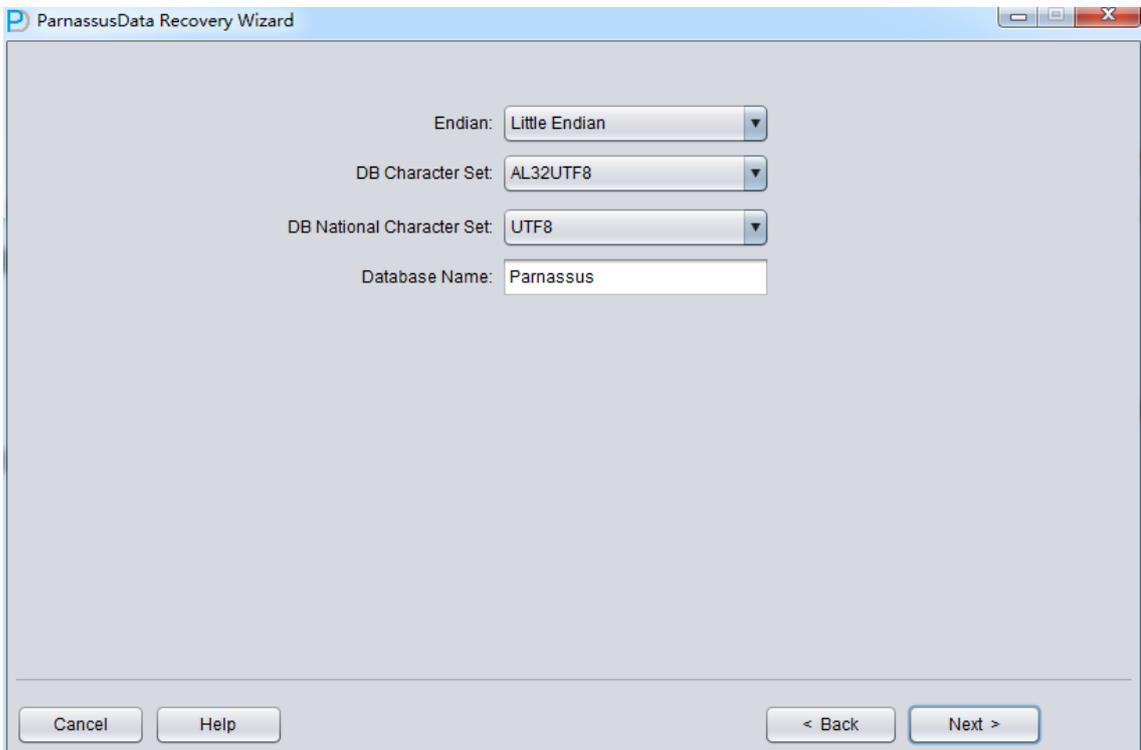
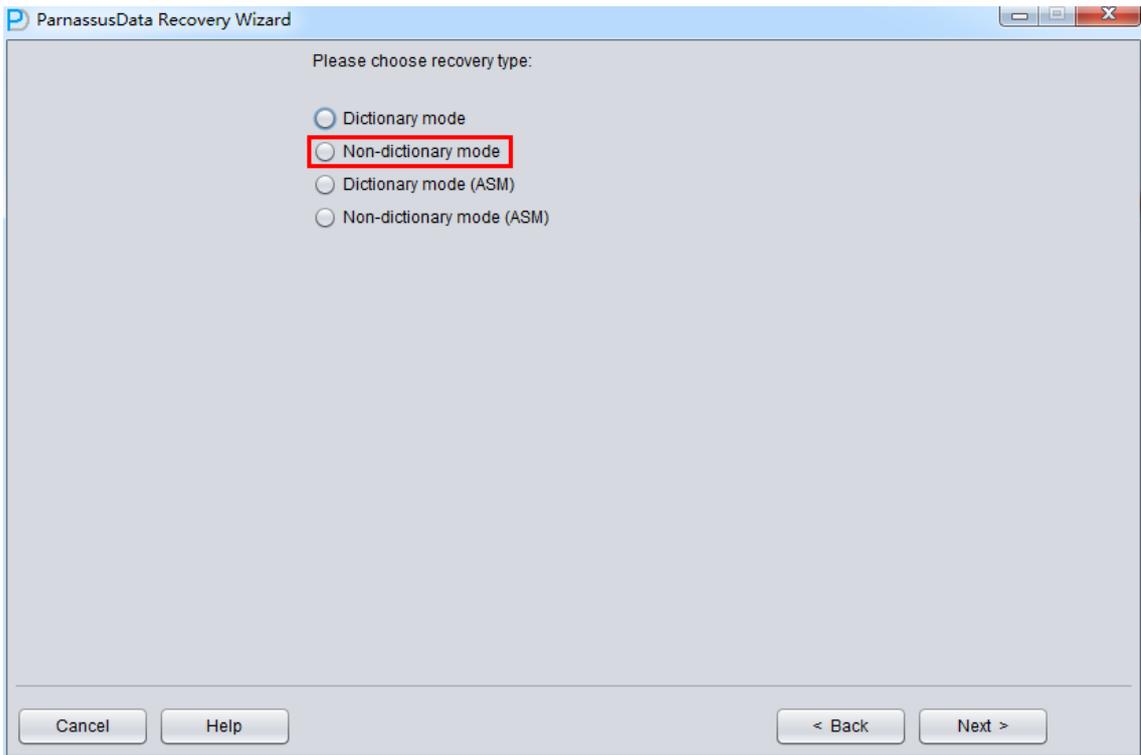
1. Recovery Wizard
2. 选择字典模式 Dictionary Mode
3. 合理选择 Big 或者 Little Endian
4. 加入数据文件并点击 Load
5. 根据实际需求恢复表中的数据



恢复场景 4 误删除或丢失 SYSTEM 表空间

D 公司的 SA 系统管理员误删除了某数据库的 SYSTEM 表空间所在数据文件，这导致数据库完全无法打开，数据无法取出。在没有备份的情况下，可以使用 PRM 恢复接近 100% 的数据。

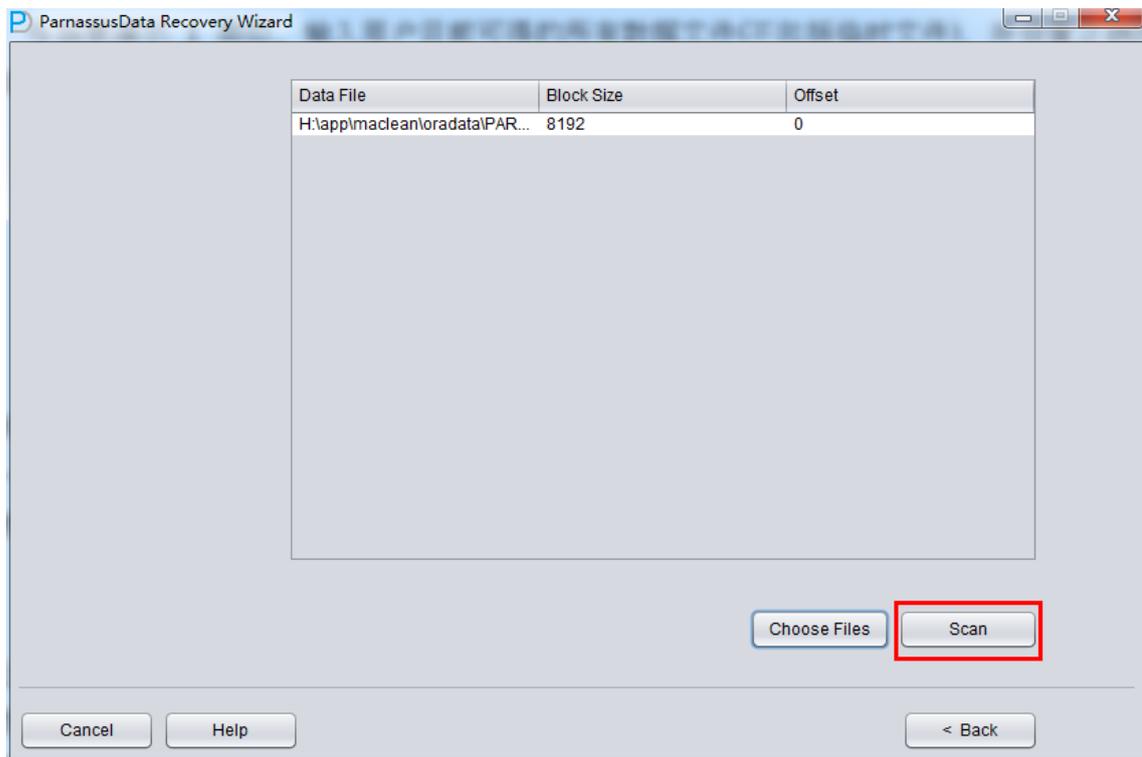
此场景中启动 PRM 后，进入 Recovery Wizard 后选择《Non-Dictionary mode》非字典模式：



No-dictionary 模式下需要用户指定 字符集和国家字符集, 这是因为丢失了 SYSTEM 表空

间后，数据库的字符集信息无法正常获得，所以需要用户的输入。只有输入正确的字符集设置以及安装了必要的语言包才能保证 No-Dictionary 模式下正常抽取多国语言。

与场景演示 1 类似，输入用户目前可得的所有数据文件(不包括临时文件)，并设置正确的 Block Size 和 OFFSET:

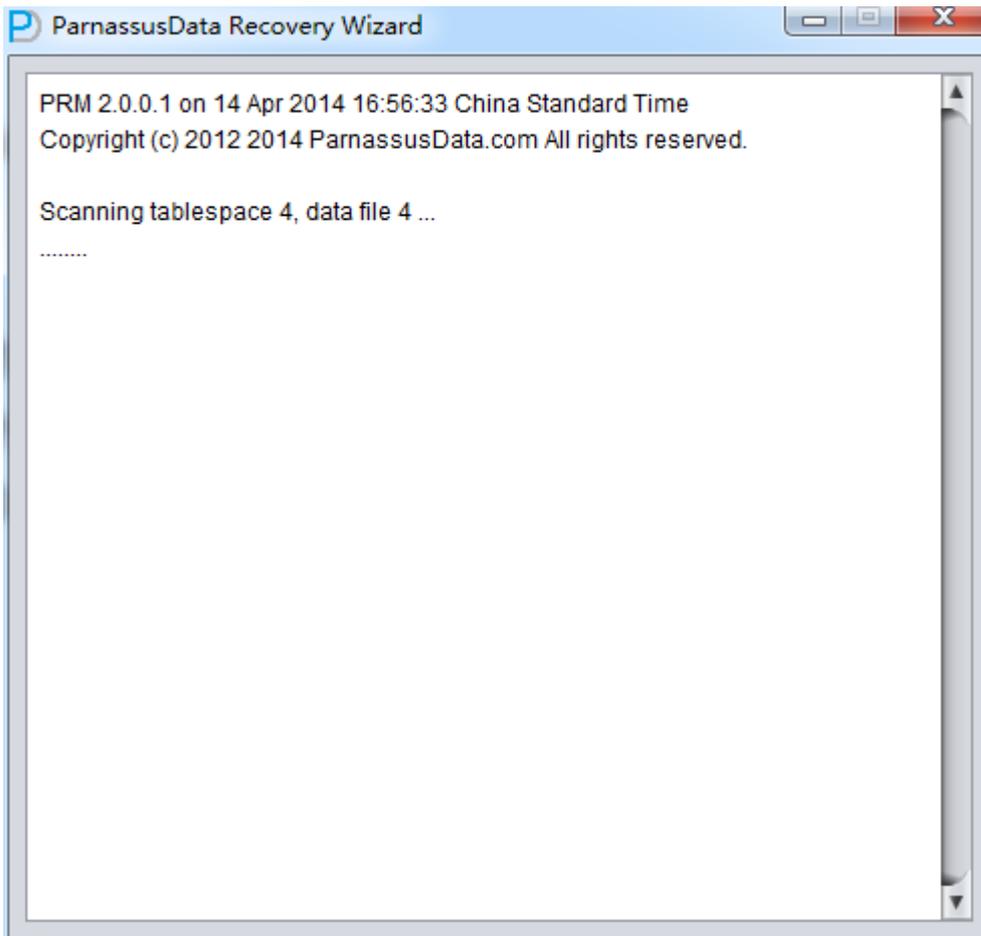


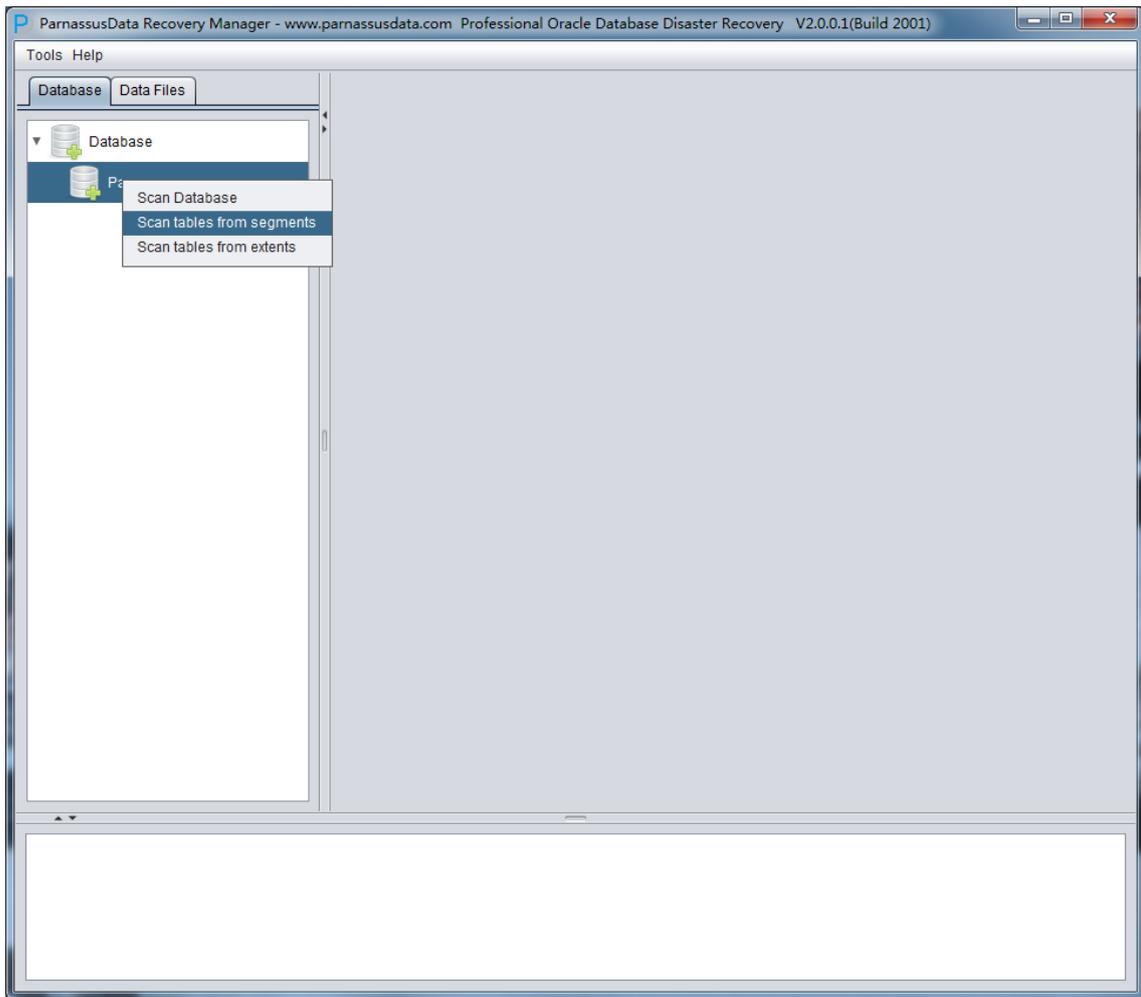
之后点击 SCAN，SCAN 的作用是扫描所有数据文件上的 Segment Header，并记录到 SEG\$.DAT 和 EXT\$.DAT 中；在 ORACLE 中一个非分区表或者一个分区表的分区都对应着一个 SEGMENT HEADER 数据段头，只要能找到 SEGMENT HEADER 就可以获得整个表数据段的盘区 EXTENT MAP 信息，通过 EXTENT MAP 可以获得该表上的全部记录。

通过存在这样一种情况，例如一张非分区的单表存放在某个由 2 个数据文件组成的表空间上，其 SEGMENT HEADER 以及一半的数据存放在 A 数据文件上，另一半数据存放在 B 数据文件上。但是由于某些原因，SYSTEM 表空间和存放有 SEGMENT HEADER 的 A 数据文件均丢失了，只剩下 B 数据文件了，此时若希望仅仅恢复 B 数据文件上该表的数据，则不能依赖于 SEGMENT HEADER，而只能依赖于从 B 数据文件上扫描盘区图 EXTENT

MAP 信息了。

为了同时满足 基于 SEGMENT HEADER 和 EXTENT MAP 数据的 NO-Dictionary 模式恢复需要，所以 SCAN 操作在这里会填充 SEG\$.DAT 和 EXT\$.DAT2 个文件(文本文件仅仅为了便于诊断，所有程序实际依赖于 PRM 自带嵌入数据库 DERBY 的数据)，并记录到 DERBY 数据库中。





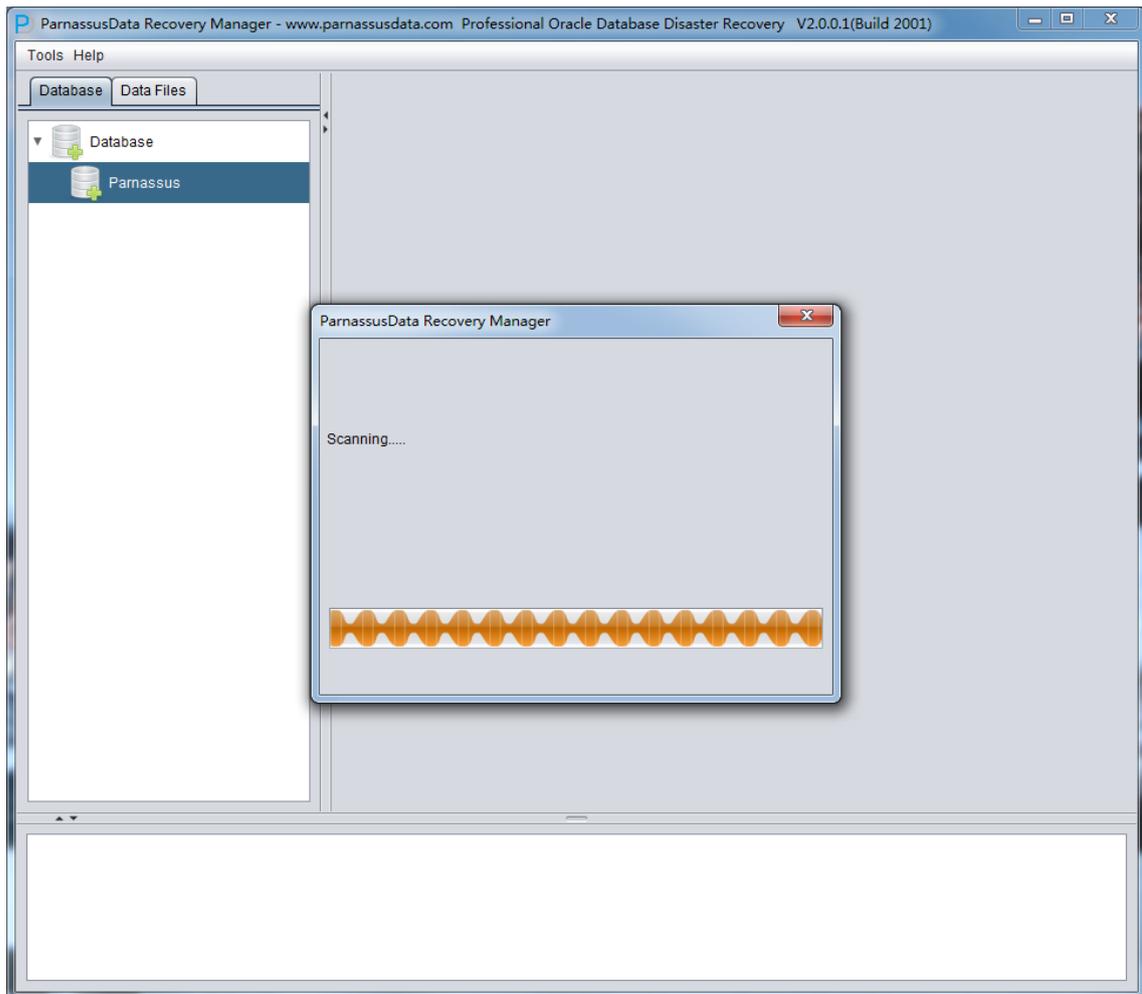
完成 SCAN 后，主界面左侧出现数据库图标。

此时可以选择 2 种模式：

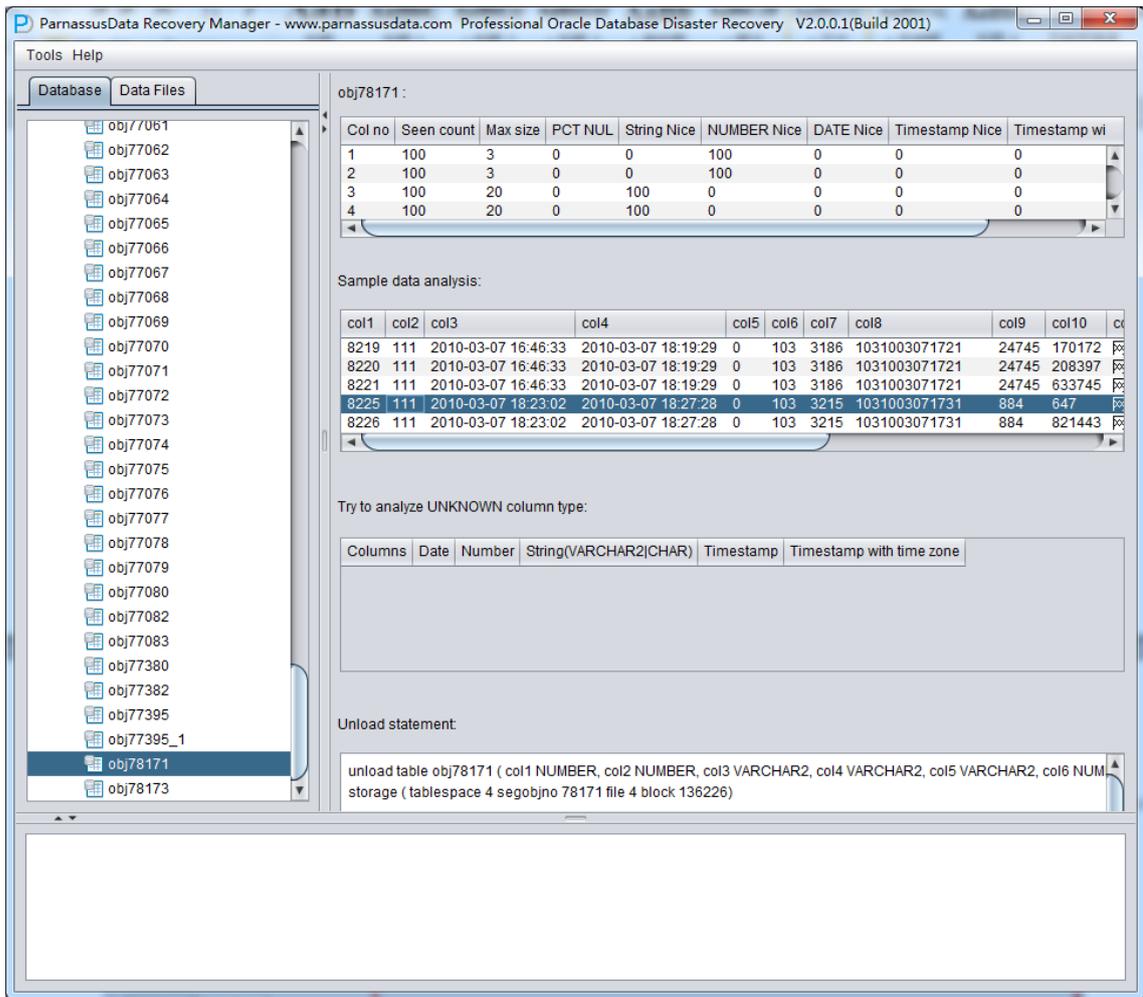
- 1、Scan Tables From Segments ，此模式适用于
 - 丢失了 SYSTEM 表空间，但是所有的应用数据表空间均存在
 -
- 2、Scan Tables From Extents
 - 不适用于 Dictionary 模式的 Truncate 表数据恢复
 - 丢失了 SYSTEM 表空间，而且丢失了 SEGMENT HEADER 所在数据文件

通俗地说 除非你无法使用场景 2 中的方式来恢复已经 TRUNCATE 掉的数据, 否则总是优先使用 Scan Tables From Segments 模式, 如果发现 Scan Tables From Segments 下找不到你要的数据, 再考虑使用 Scan Tables From Extents 模式。

我们优先采用 Scan Tables From Segments 模式

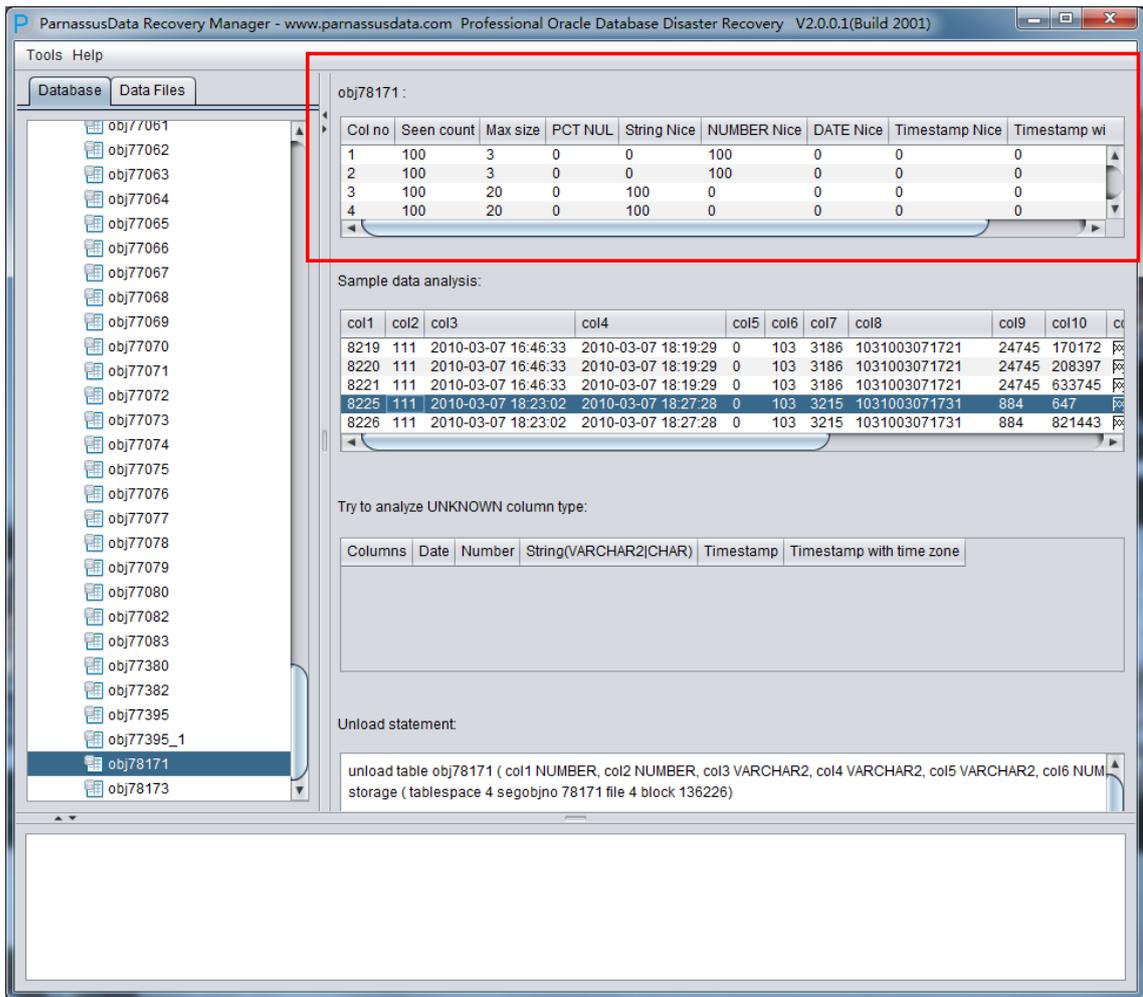


Scan Tables From Segments 完成后可以点开主界面左边的树形图：



Scan Tables 操作基于 SEG\$ 中的 SEGMENT HEADER 信息来构建数据表信息，树形图上每一个节点表示一个数据表段，其名字为 obj+ 数据段上记录的 DATA OBJECT ID 。

点击一个节点 并观察主界面右侧边栏：



智能字段类型解析

由于丢失了 SYSTEM 表空间，故 NO-Dictionary 模式下缺乏数据表的结构信息，这些结构信息包括表上的字段名字和字段类型，而且在 ORACLE 中这些信息均只保存为字典信息，不会在数据表上存放。当用户只有应用表空间时，需要基于数据段上的 ROW 行数据来猜测每一个字段的类型，PRM 采用先进 JAVA 类型预判技术，可以解析多达 10 来种主流数据类型；、

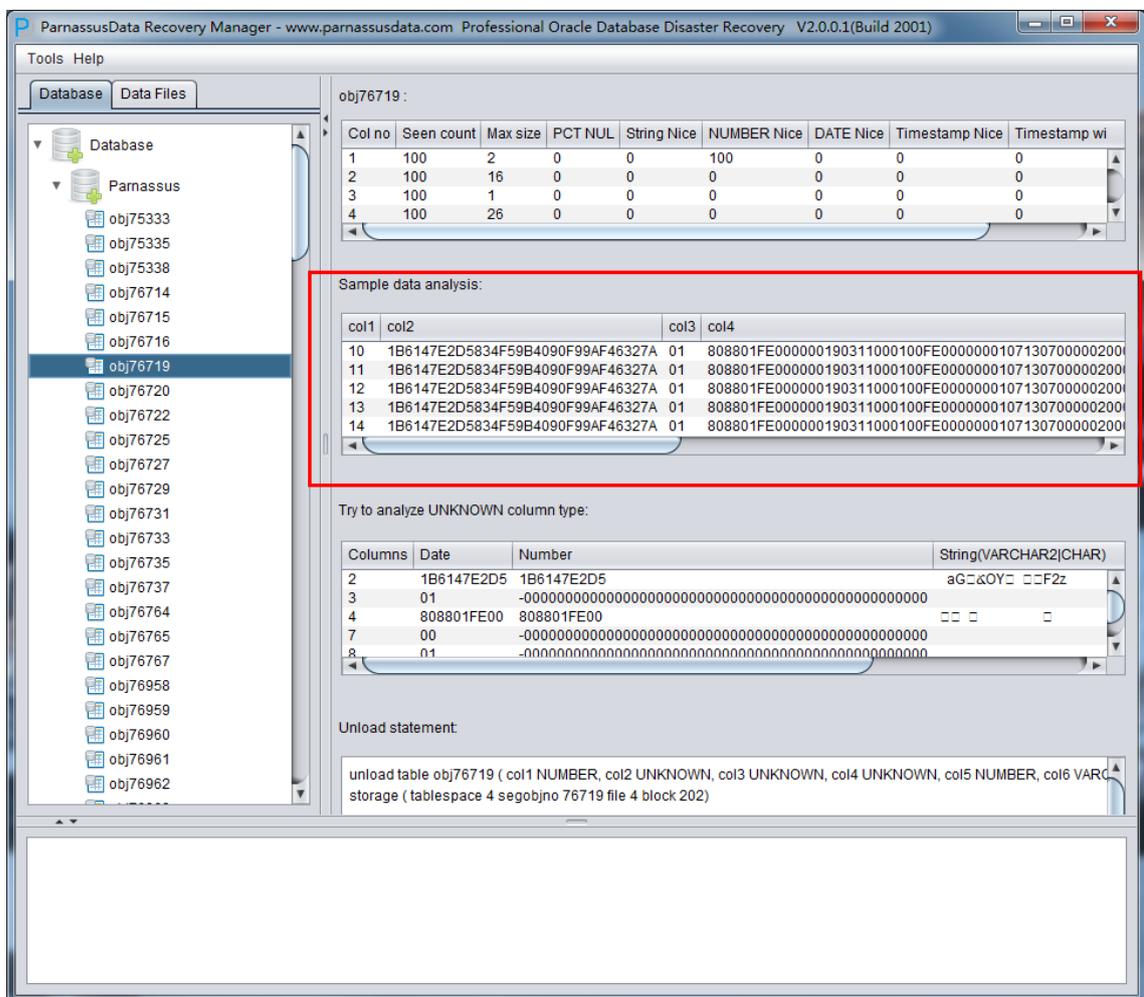
智能解析准确度超过 90%，可以自动解决大部分场景。

右侧边栏 上部各字段的含义：

- Col1 no 字段号

- Seen Count: 取到的行数
- MAX SIZE: 最大长度, 单位为字节
- PCT NULL: 采样到的 NULL 的比例
- String Nice: 将该字段解析为字符串, 并成功的比例
- Number Nice: 将该字段解析为数字, 并成功的比例
- Date Nice: 将该字段解析为 Date, 并成功的比例
- Timestamp Nice: 将该字段解析为 Timestamp, 并成功的比例
- Timestamp with timezone Nice: 将该字段解析为 Timestamp with timezone Nice, 并成功的比例

示例数据分析 Sample Data Analysis:

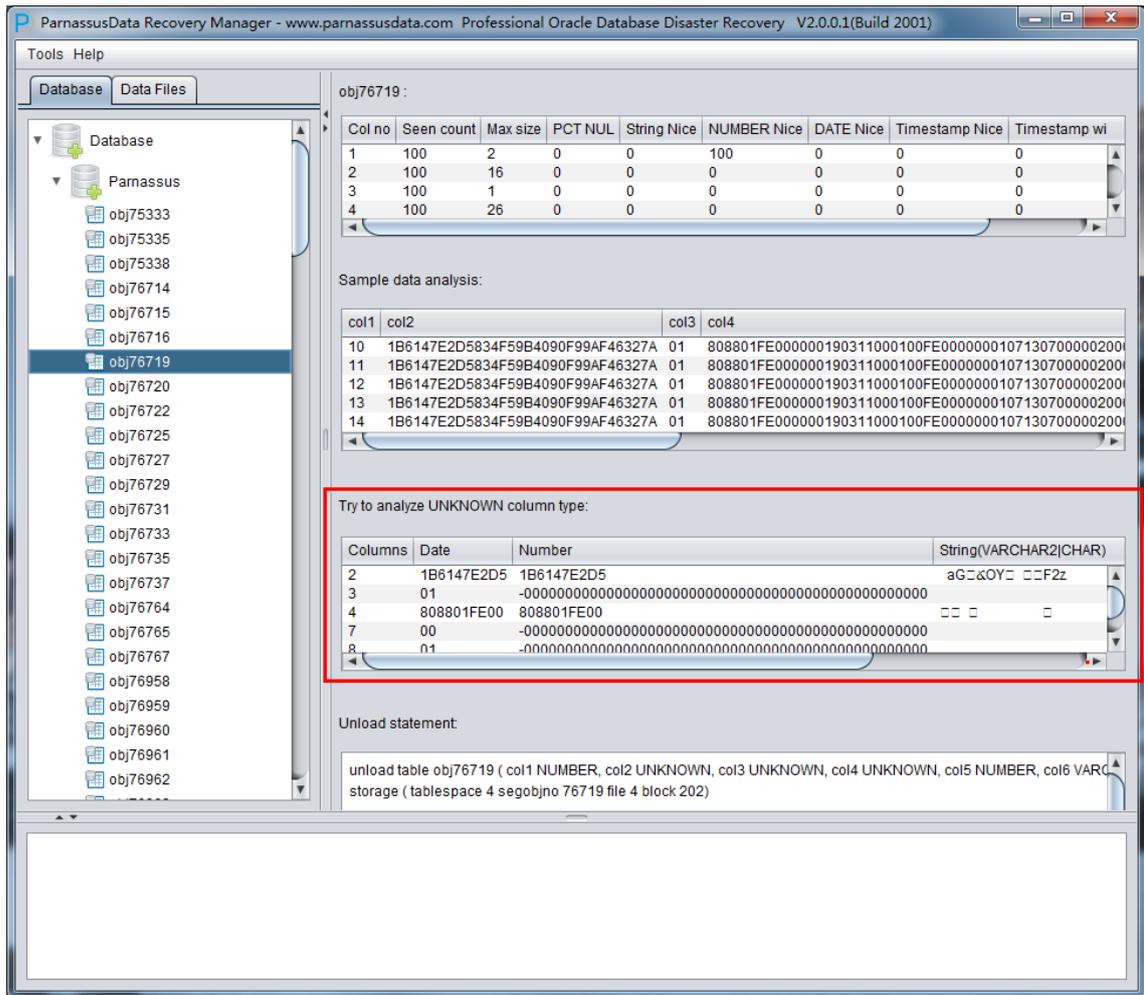


该部分依据智能字段类型解析的结果来解析 10 条数据, 并显示解析结果。通过示例数据可

以帮助用户了解实际该数据段中存放数据的情况。

如果数据段上记录条数不足 10 条，则显示所有记录。

TRY TO ANALYZE UNKNOWN column type:



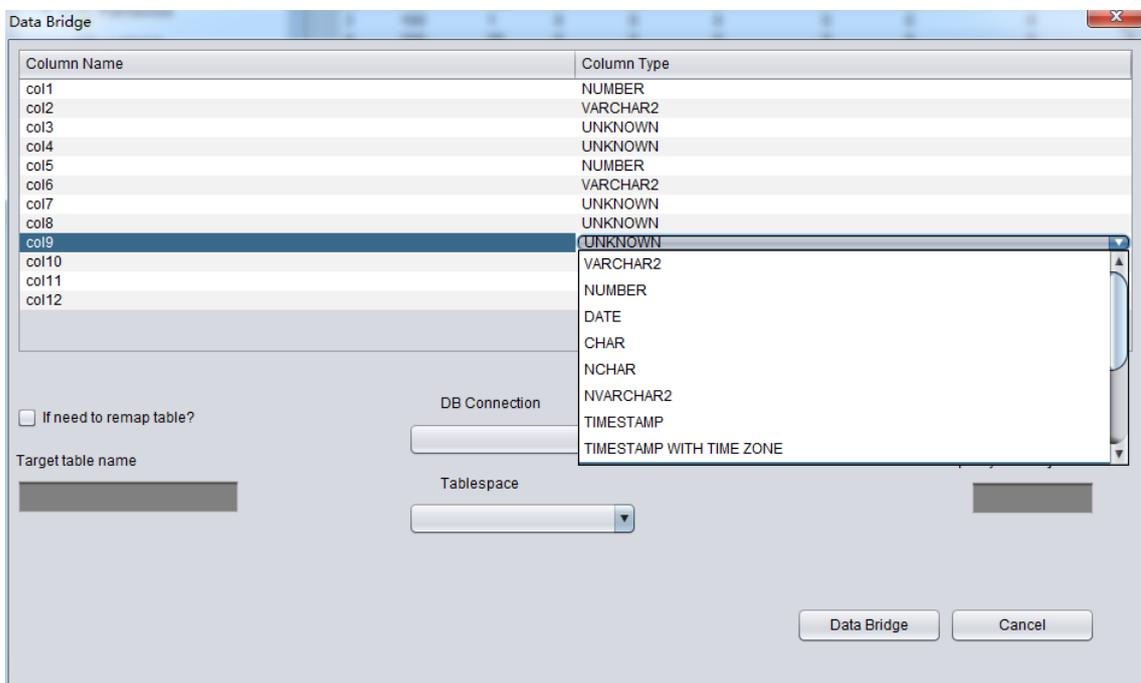
该部分是针对智能字段类型分析不能 100%确认的字段，尝试用各种字段类型来解析，并呈现给用户，以便用户自行判断其究竟是什么类型。

目前 PRM 还不支持的类型包括：

XDB.XDB\$RAW_LIST_T、XMLTYPE、用户自定义类型等

在此《Non-Dictionary Mode》非字典模式下同样可以采用常规和数据搭桥模式，与字典模式相比，主要的区别在于在非字典模式下数据搭桥时用户可以自行执行字段的类型，如下图中中部分字段类型为 UNKNOWN，即未知的。这些字段可能是 PRM 目前还不支持的例如 XML 字段，也可能是 PRM 的智能解析没有顺利分析器类型。

如果用户知道这张表设计时的结构（也可以来源于应用开发商的文档），那么可以自行去填选正确的 Column Type 类型，以便 PRM 顺利将该表数据搭桥到目标数据库。



恢复场景5 误删除了SYSTEM表空间和部分应用表空间数据文件

D 公司的 SA 由于误操作将在线业务数据库的 SYSTEM 表空间上的数据文件，以及部分应

用表空间数据文件意外删除了。

此场景中由于部分应用表空间数据文件被删除了，这其中可能包括含有数据表的 SEGMENT HEADER 的数据文件，所以使用 Scan Tables From Segment Header 可能不如使用 Scan Tables From Extents 来的合适。

其简要步骤如下：

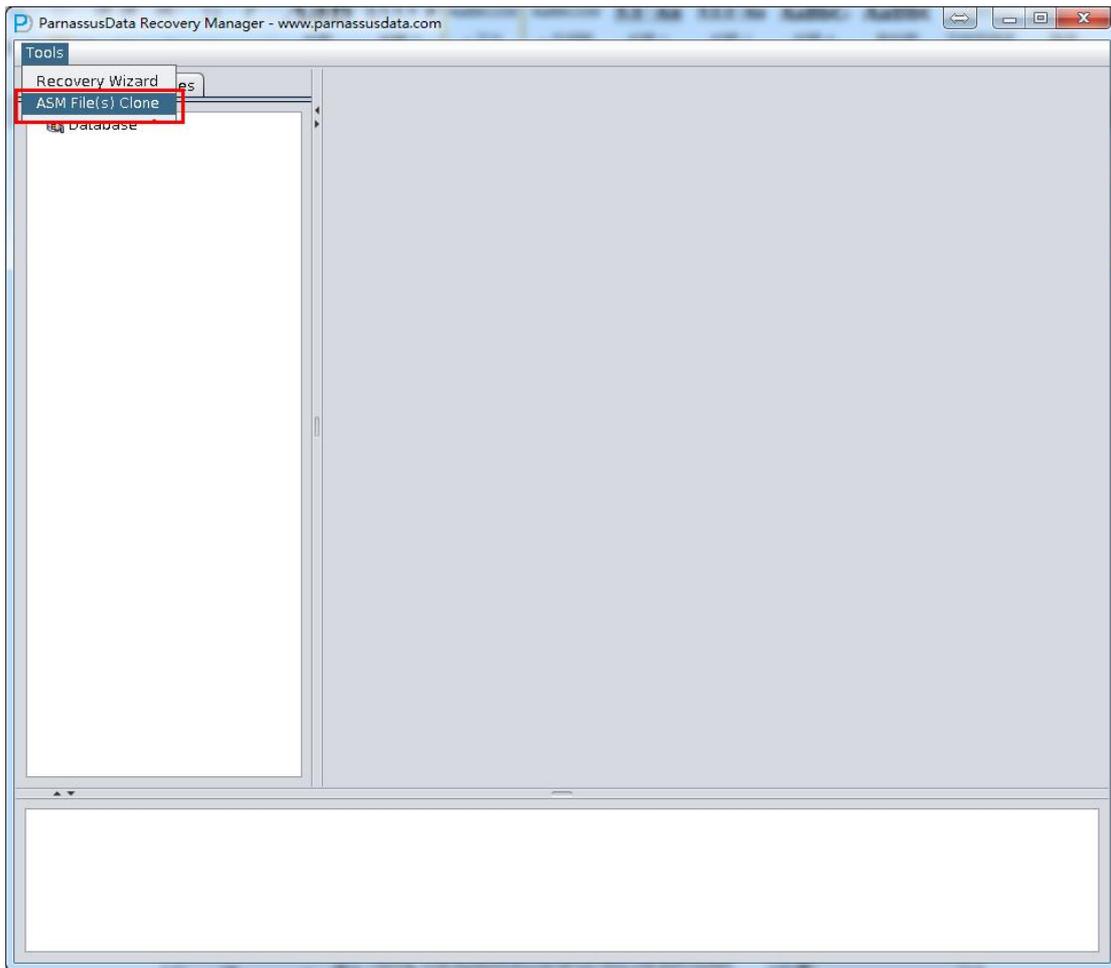
1. 进入 Recovery Wizard ，选择 No-Dictionary 模，加入所有可用的数据文件，执行 Scan Database
2. 选中数据库，并右键 Scan Tables From Extents
3. 对于 PRM 主界面上生成的对象树形图中的数据进行分析 and 导出/数据搭桥
4. 其余操作与恢复场景 4 中一样

恢复场景 6 从被损坏的 ASM Diskgroup 中拷贝出数据库数据文件

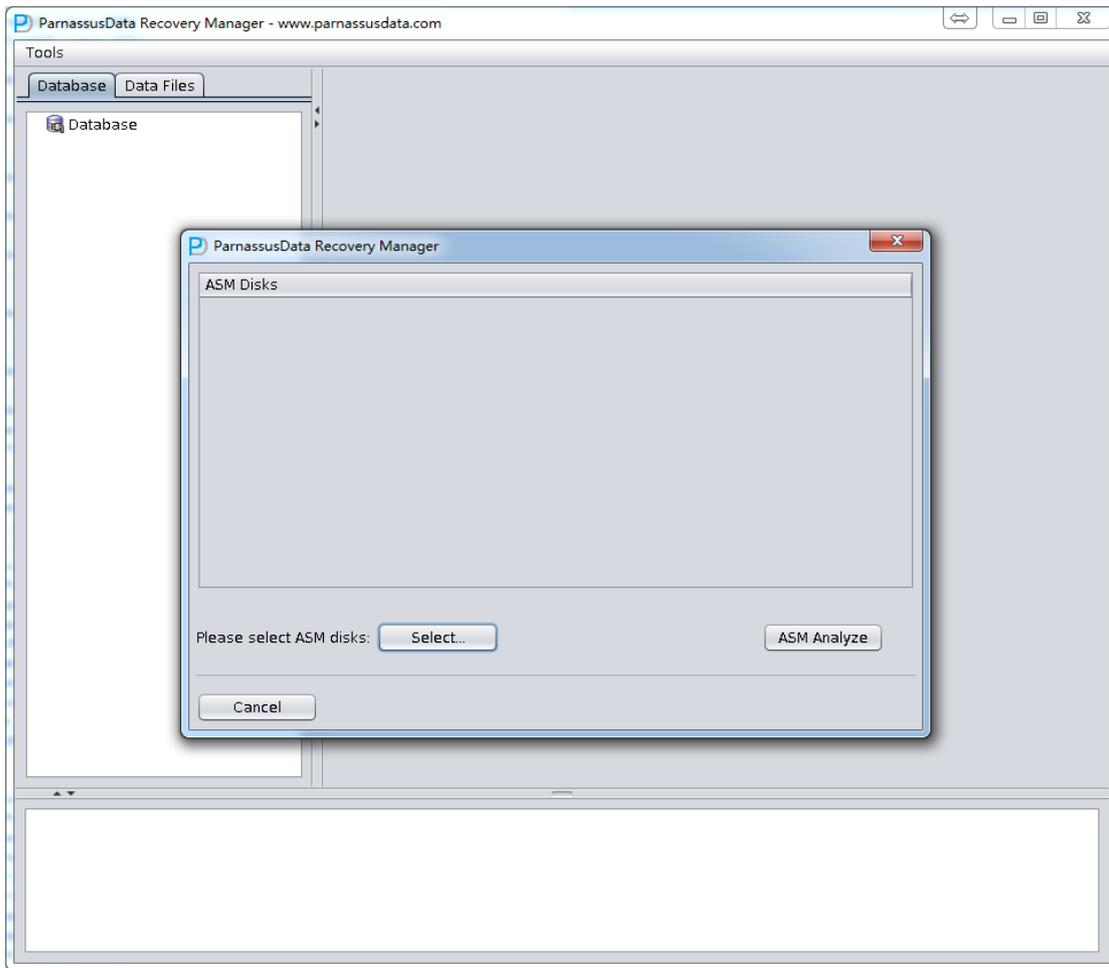
D 公司开始采用 ASM 方案来替代文件系统和裸设备，但是由于使用的 11.2.0.1 版本 ASM 上 Bug 较多导致 ASM DISKGROUP 磁盘组无法加载 MOUNT，通过多方修复 ASM Disk Header 无果。

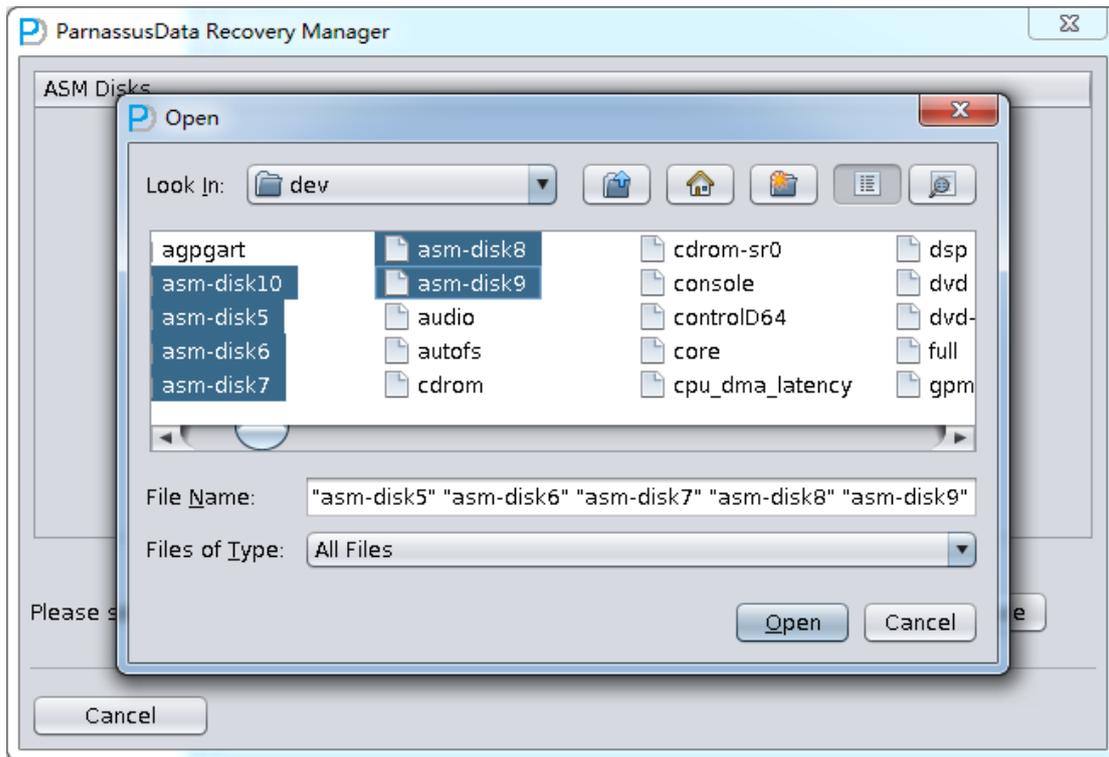
此场景可以使用 PRM 的 ASM Files Clone 文件克隆功能从受损的 ASM Diskgroup 中拷贝出数据库数据文件。

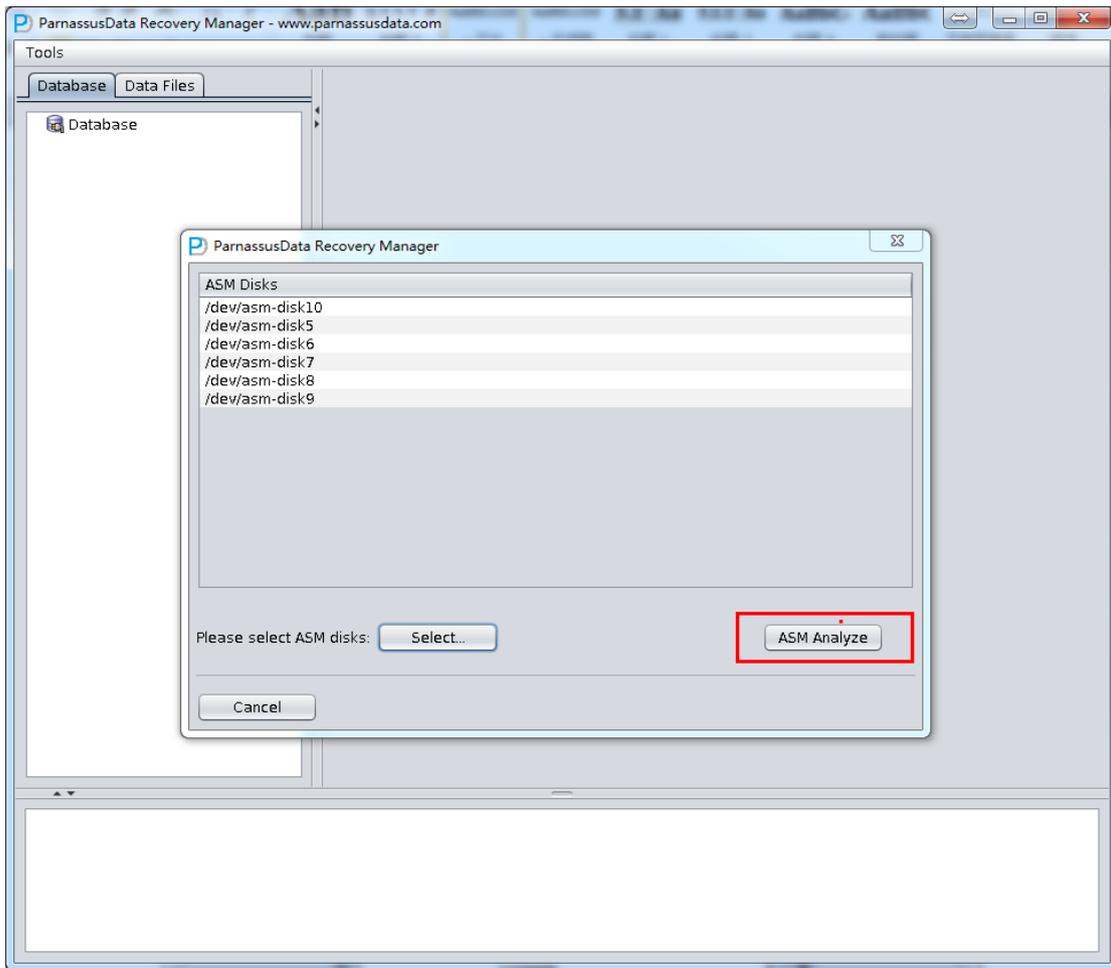
1. 打开主界面，菜单栏 Tools 选择 ASM File(s) Clone:



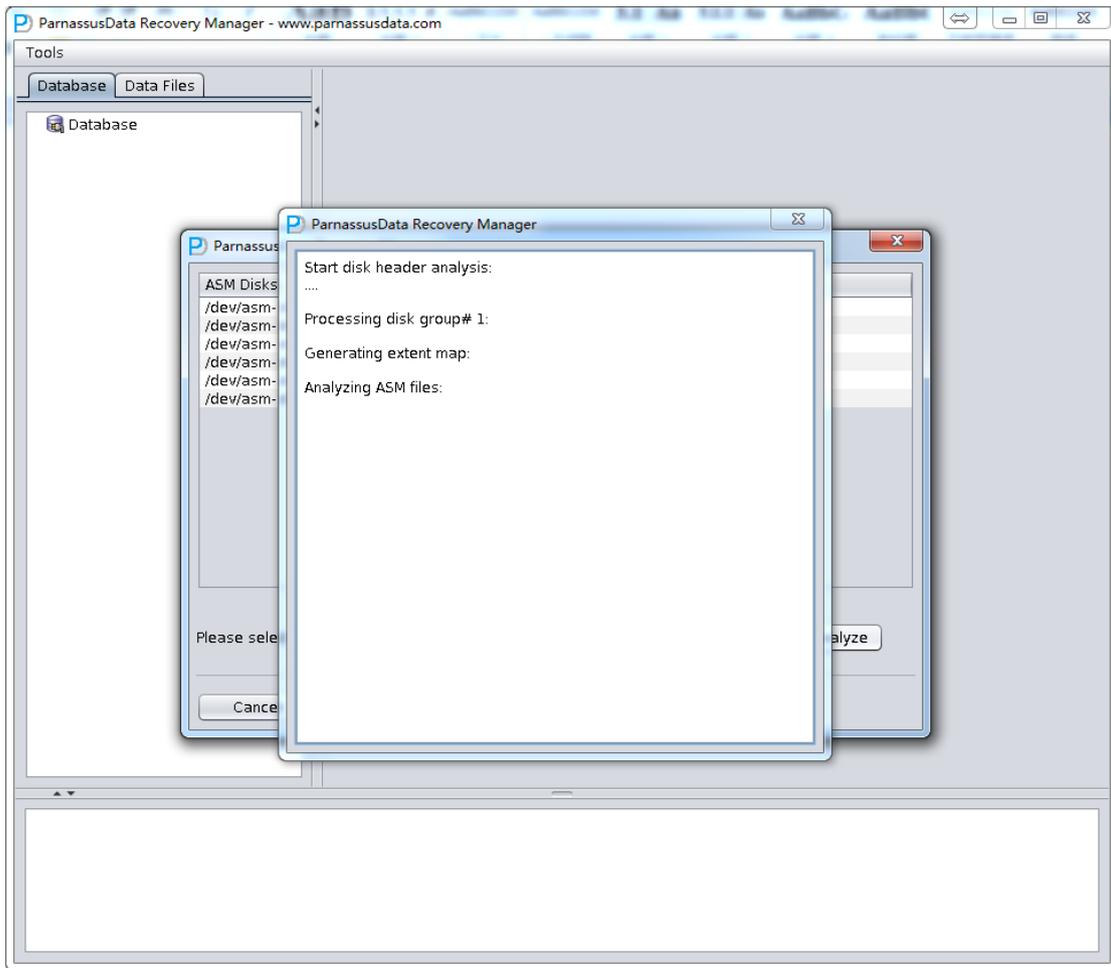
2. 进入 ASM Disks 界面，点击 SELECT...按钮加入仍可用的 ASM Disks，如 /dev/asm-disk5(linux);确保加入所有可用 LUN 后，点击 ASM analyze 按钮





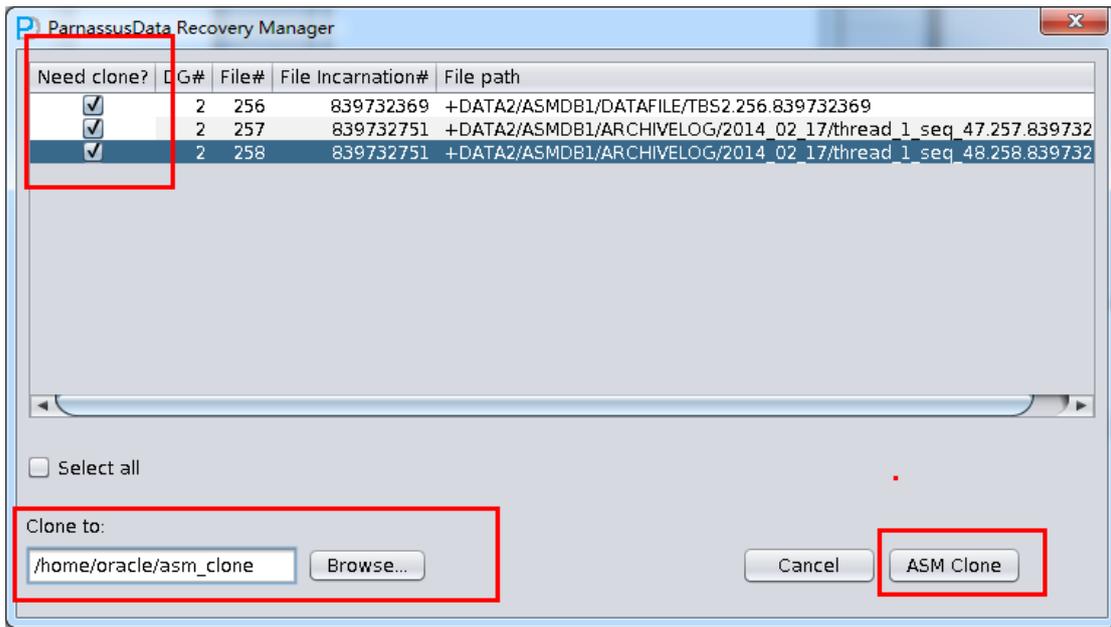


3. ASM Files Clone 将分析指定的 ASM Disk 的磁盘头，以便找出对应 Disk group 磁盘组中的文件，以及这些文件的分布位置(File Extent Map)；这些信息均将记录到 Derby 数据库中以便今后使用；可以说 PRM 将 ASM 的所有 Metadata 元数据均收集、分析、并存储起来，并通过各种形式来完善 PRM 的基本功能，并以图形化地方式展现给用户。

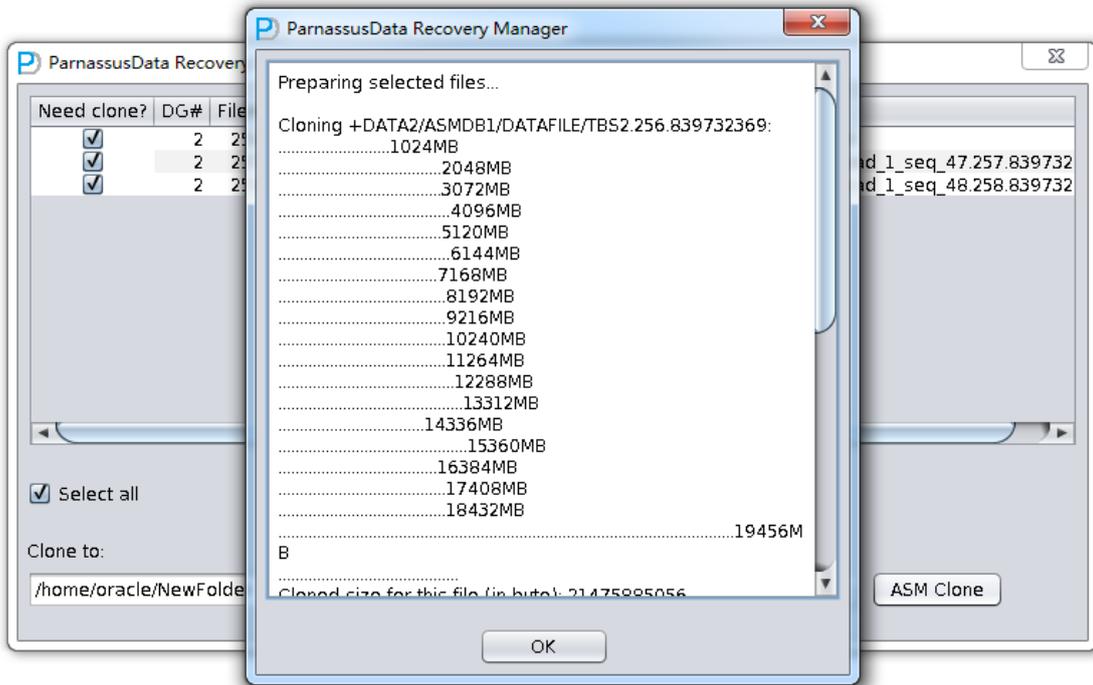


4. ASM Analyze 分析完成后，PRM 将列出找到的 ASM 上文件的列表，用户可以勾选那些文件需要被克隆，并指定文件克隆的目标文件夹。

之后点击 ASM Clone 按钮，进入文件克隆阶段。



文件克隆阶段中，将列出 ASM File 的克隆进度，克隆完成后点击 OK。



克隆阶段的进度日志输出如下：

```
Preparing selected files...

Cloning +DATA2/ASMDB1/DATAFILE/TBS2.256.839732369:
.....1024MB
.....2048MB
.....3072MB
.....4096MB
.....5120MB
.....6144MB
.....7168MB
.....8192MB
.....9216MB
.....10240MB
.....11264MB
.....12288MB
.....13312MB
.....14336MB
.....15360MB
.....16384MB
.....17408MB
.....18432MB
.....
.....19456MB
.....
Cloned size for this file (in byte): 21475885056

Cloned successfully!

Cloning
+DATA2/ASMDB1/ARCHIVELOG/2014_02_17/thread_1_seq_47.257.839732751:
.....
Cloned size for this file (in byte): 29360128

Cloned successfully!

Cloning
+DATA2/ASMDB1/ARCHIVELOG/2014_02_17/thread_1_seq_48.258.839732751:
.....
Cloned size for this file (in byte): 1048576

Cloned successfully!

All selected files were cloned done.
```

5. 可以通过 dbv 或者 rman validate 命令来验证克隆出来的数据文件，例如：

```

rman target /

RMAN> catalog datafilecopy
'/home/oracle/asm_clone/TBS2.256.839732369.dbf';

cataloged datafile copy
datafile copy file
name=/home/oracle/asm_clone/TBS2.256.839732369.dbf RECID=2
STAMP=839750901

RMAN> validate datafilecopy
'/home/oracle/asm_clone/TBS2.256.839732369.dbf';

Starting validate at 17-FEB-14
using channel ORA_DISK_1
channel ORA_DISK_1: starting validation of datafile
channel ORA_DISK_1: including datafile copy of datafile 00016 in backup
set
input file name=/home/oracle/asm_clone/TBS2.256.839732369.dbf
channel ORA_DISK_1: validation complete, elapsed time: 00:03:35
List of Datafile Copies
=====
File Status Marked Corrupt Empty Blocks Blocks Examined High SCN
-----
16 OK 0 2621313 2621440 1945051
File Name: /home/oracle/asm_clone/TBS2.256.839732369.dbf
Block Type Blocks Failing Blocks Processed
-----
Data 0 0
Index 0 0
Other 0 127

Finished validate at 17-FEB-14

```

对于使用 ASMLIB 的 ASM 环境要如何使用 PRM 呢？

其实也很简单，asmlib 相关的 ASM DISK 在 OS 操作系统上会以 `ll /dev/oracleasm/disks` 的形式存放，例如：直接将 `/dev/oracleasm/disks` 下的文件加入到 PRM ASM DISK 中即可

```

$ll /dev/oracleasm/disks
total 0
brw-rw---- 1 oracle dba 8, 97 Apr 28 15:20 VOL001
brw-rw---- 1 oracle dba 8, 81 Apr 28 15:20 VOL002
brw-rw---- 1 oracle dba 8, 65 Apr 28 15:20 VOL003
brw-rw---- 1 oracle dba 8, 49 Apr 28 15:20 VOL004
brw-rw---- 1 oracle dba 8, 33 Apr 28 15:20 VOL005
brw-rw---- 1 oracle dba 8, 17 Apr 28 15:20 VOL006
brw-rw---- 1 oracle dba 8, 129 Apr 28 15:20 VOL007
brw-rw---- 1 oracle dba 8, 113 Apr 28 15:20 VOL008

```

直接将/dev/oracleasm/disks 下的文件加入到 PRM ASM DISK 中即可。

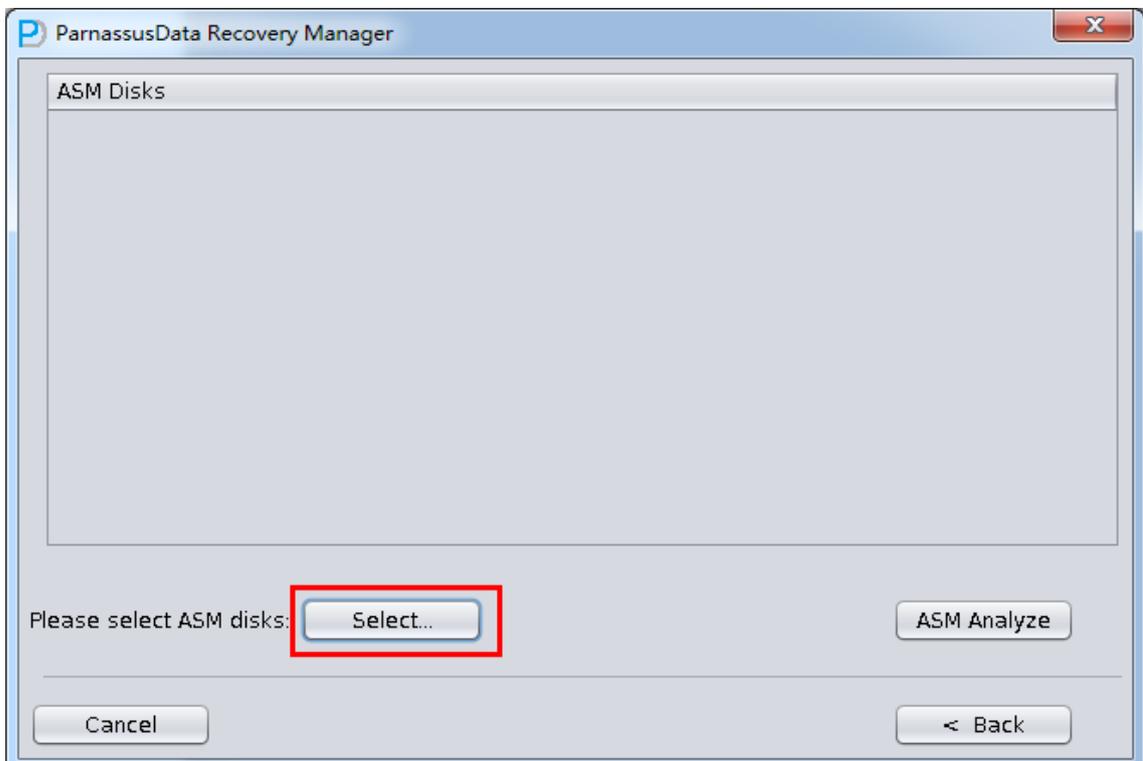
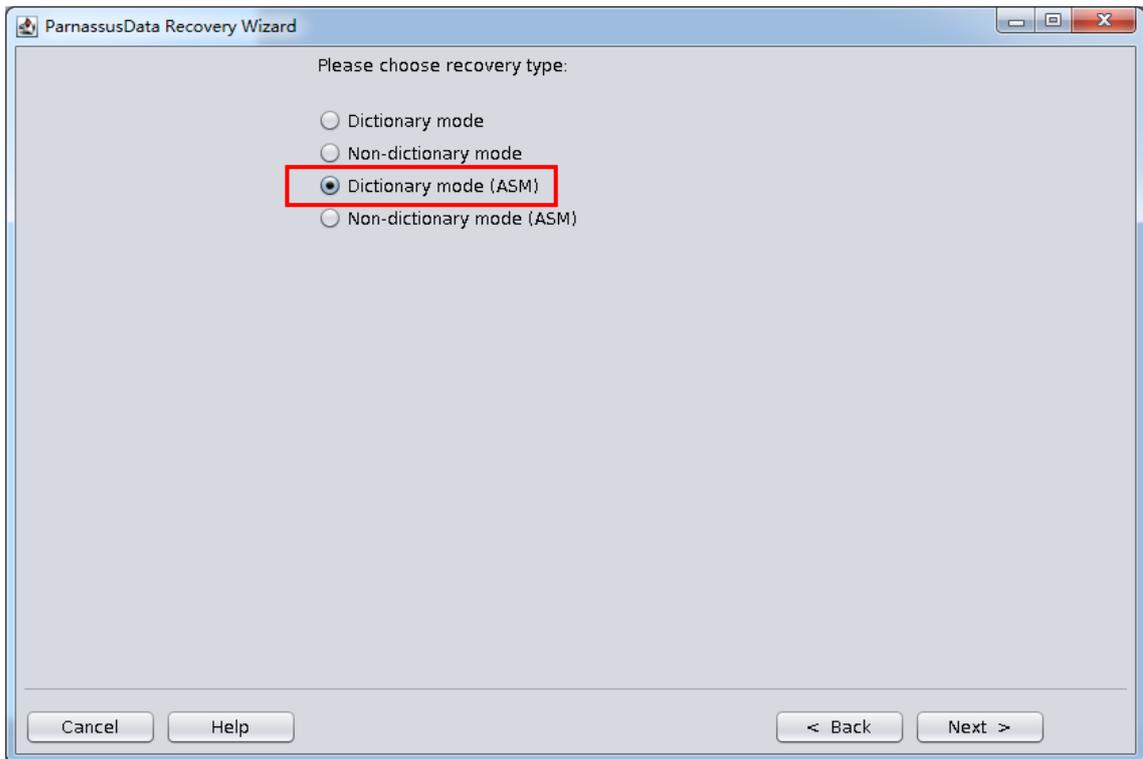
恢复场景 7 ASM 下数据库无法打开

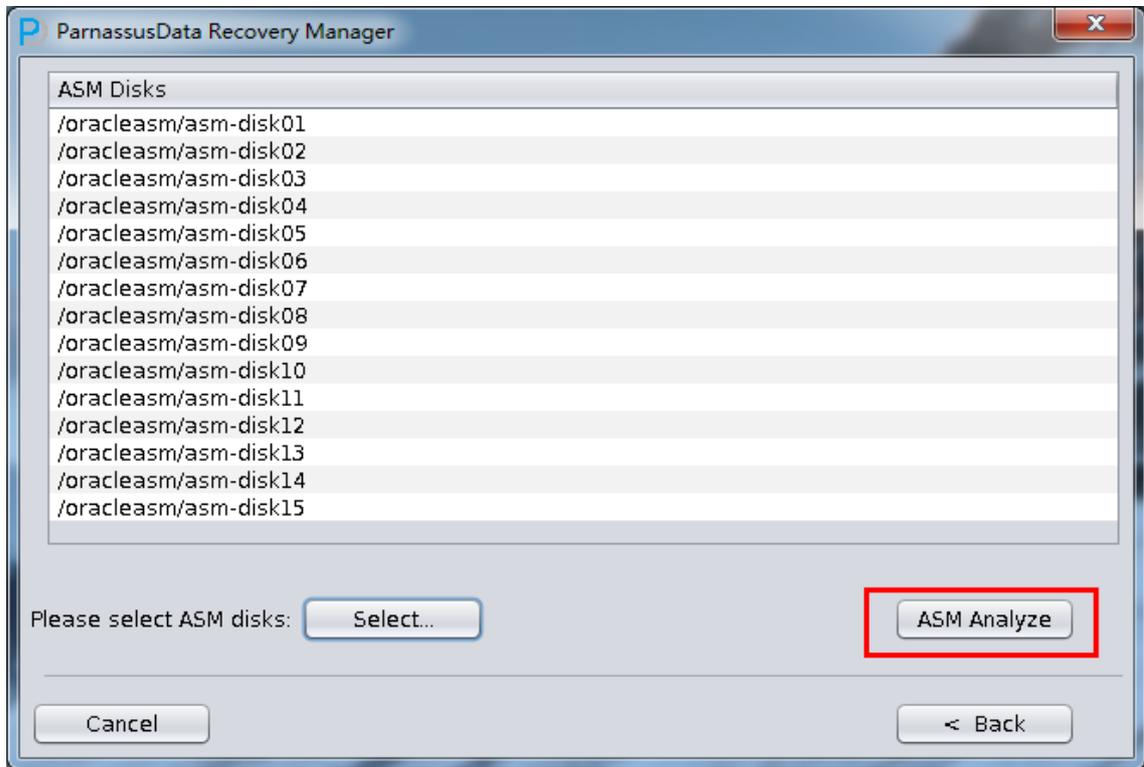
D 公司的某套核心 CRM 库由于加入到 ASM Diskgroup 中的少量磁盘存在 I/O 问题，导致 SYSTEM 表空间的 DBF 数据文件发生讹误，导致数据库无法打开。

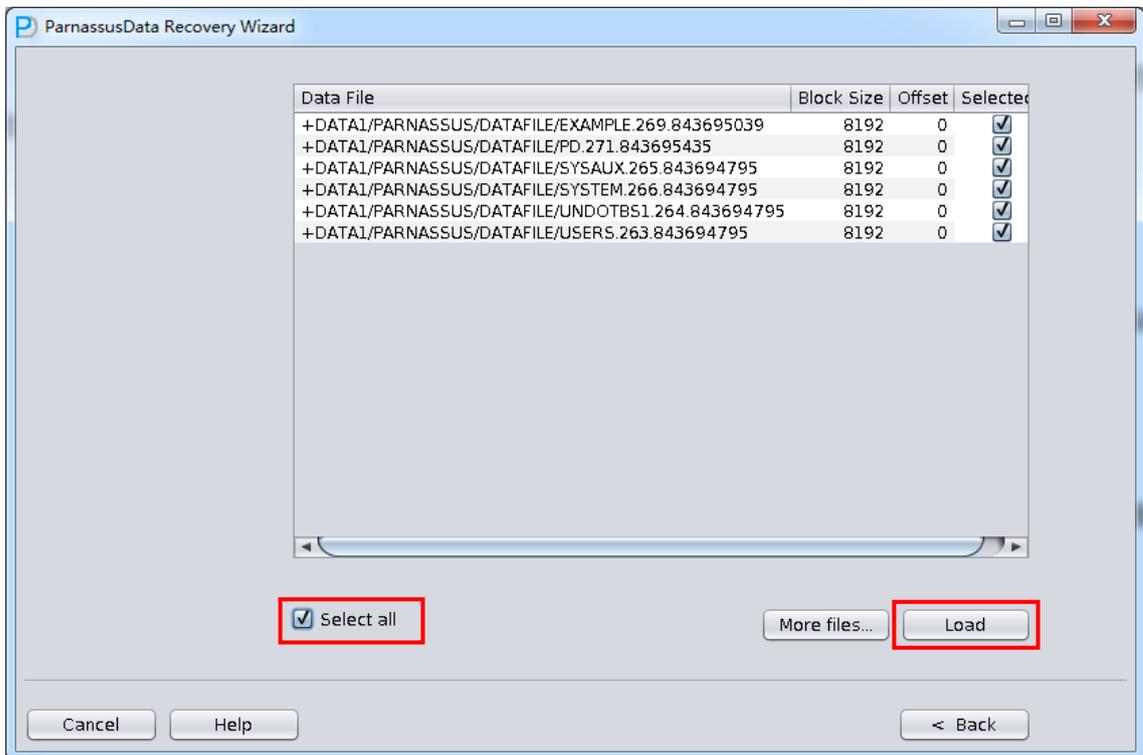
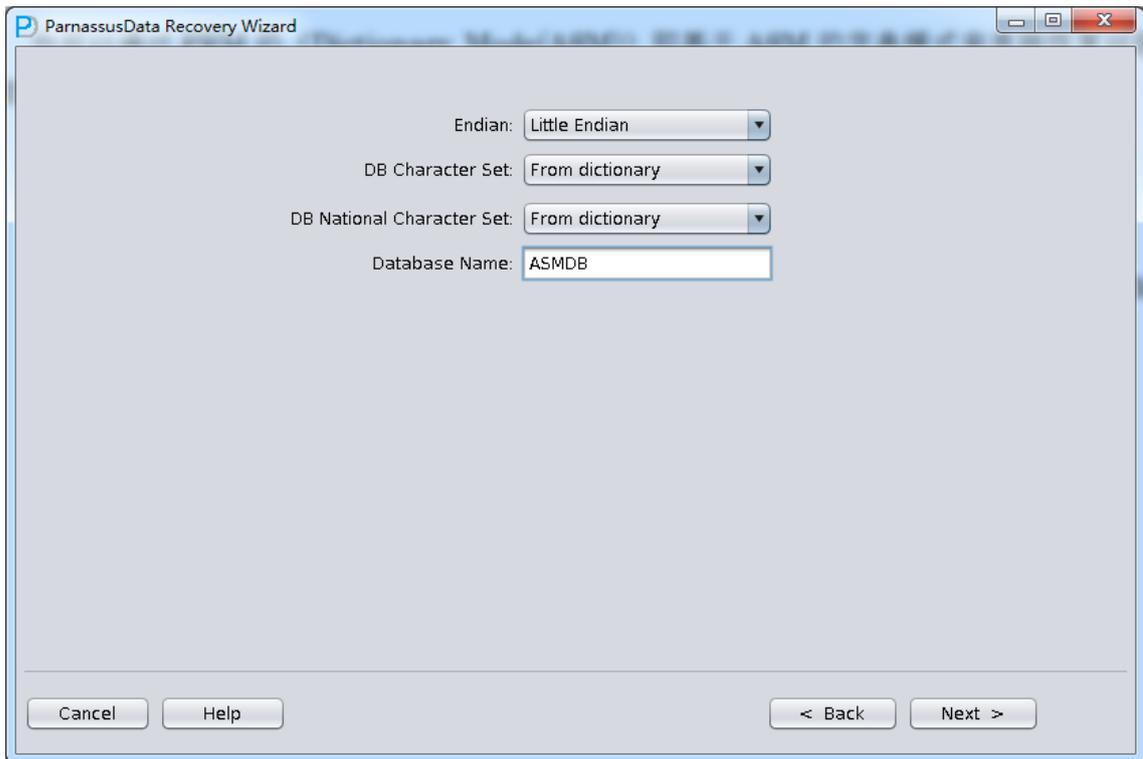
此时即可以通过 PRM 恢复软件从 ASM Diskgroup 中将 DATAFILE 全部克隆到文件系统上，如恢复场景 6 中所述，并进一步修复数据库。

也可以通过 PRM 的《Dictionary Mode(ASM)》即基于 ASM 的字典模式来直接恢复问题数据库。其简要步骤如下：

1. Recovery Wizard
2. Dictionary Mode(ASM)
3. 加入必要的 ASM DISK(你所要恢复数据库的所在的 ASM Disk Group 的所有 ASM DISK)
4. 点击 ASM analyze
5. 为后面的数据文件选择合适的 Endian
6. 在 ASM analze 给出的数据文件列表中选中需要的数据文件，如果嫌麻烦且只有一套库，那么可以勾选”Select all”
7. 点击 load 按钮，后续的恢复与《场景 3》中类似







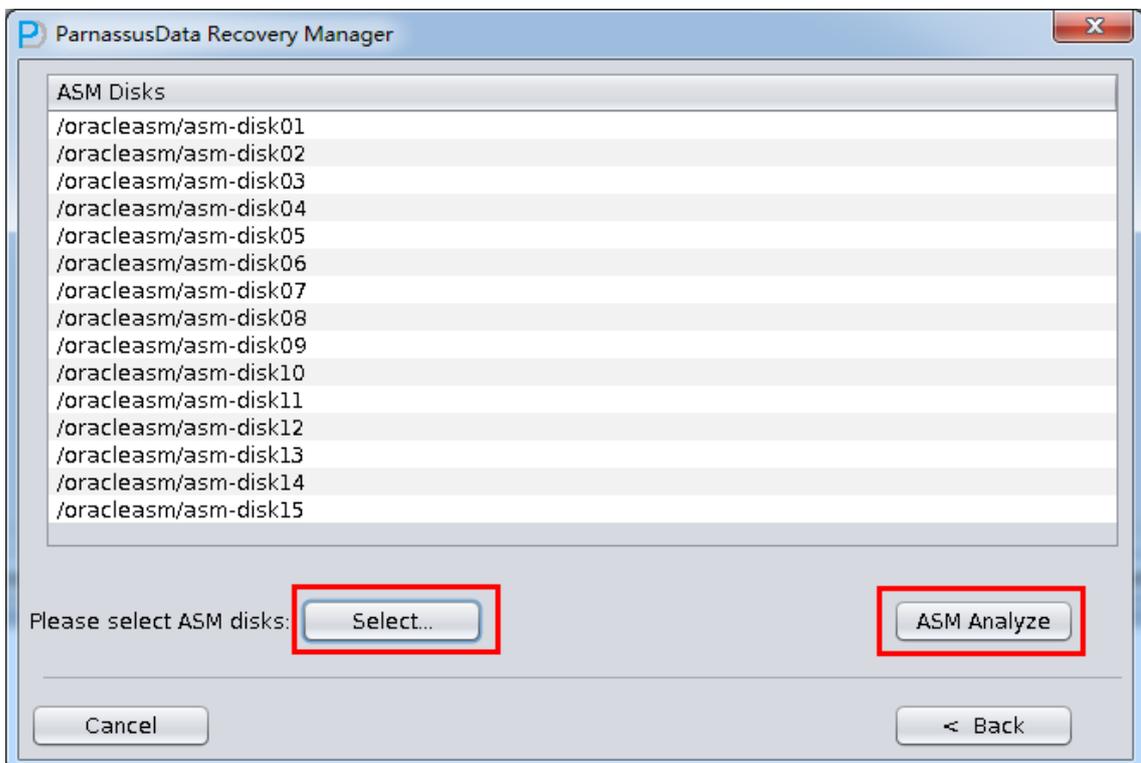
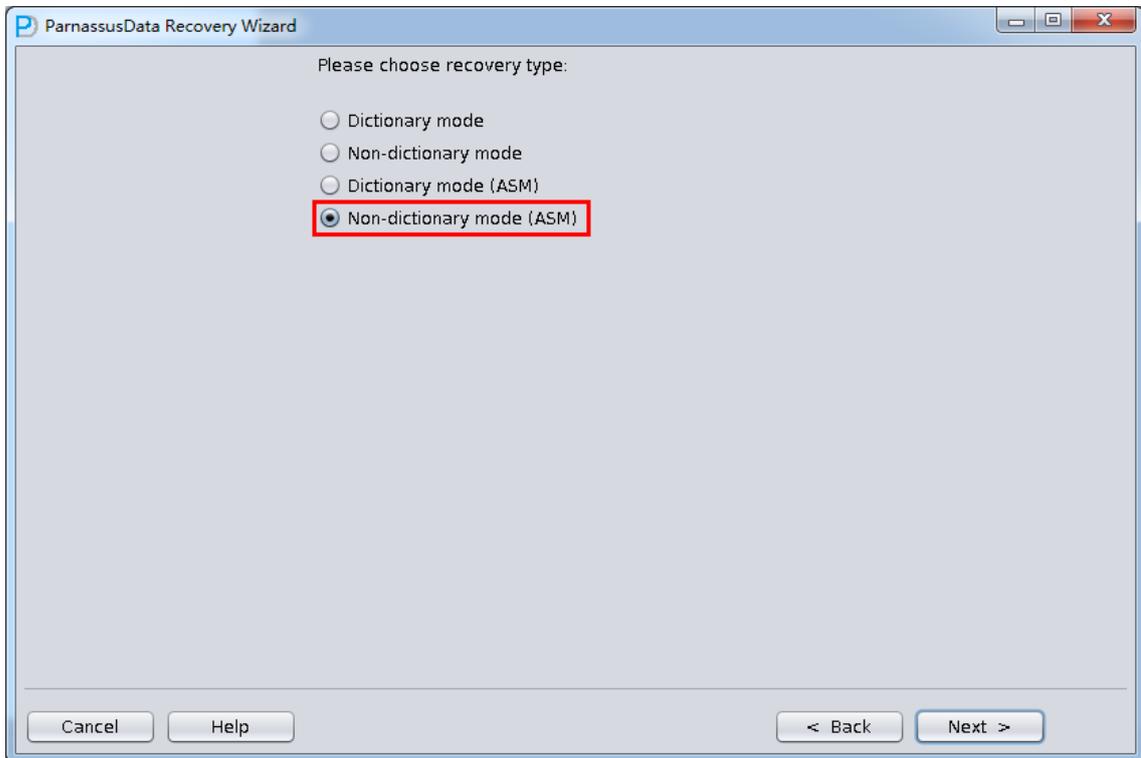
恢复场景 8 ASM 下误删或丢失 SYSTEM 表空间的恢复

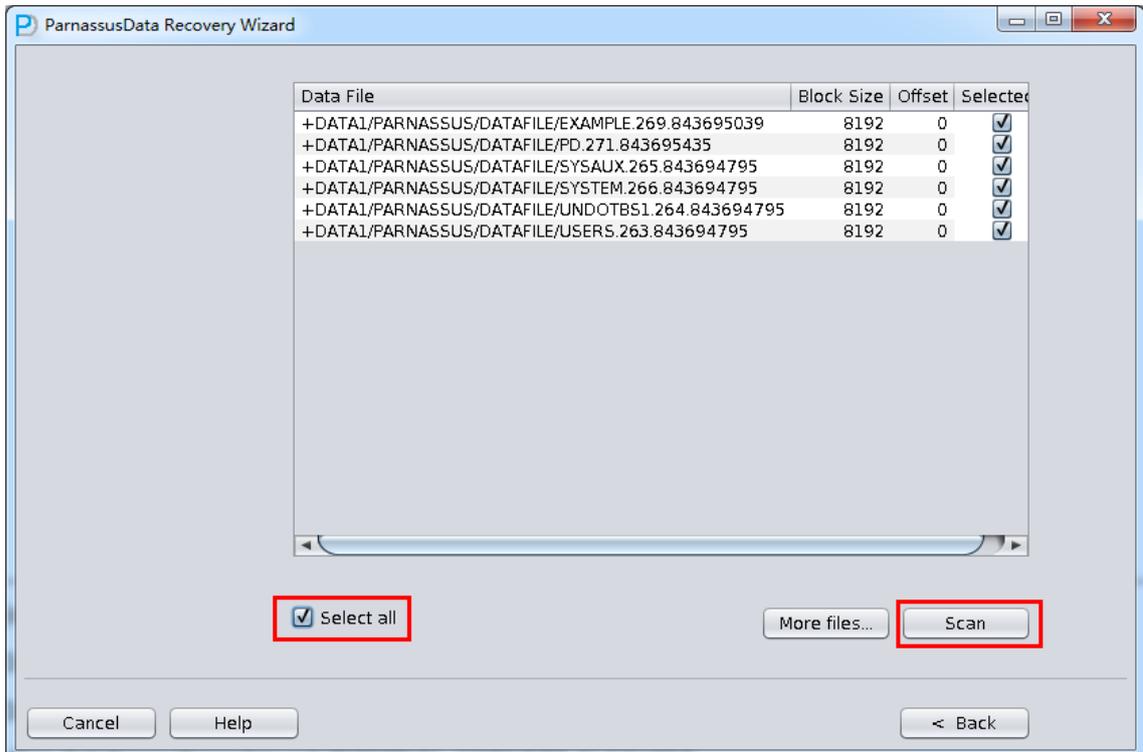
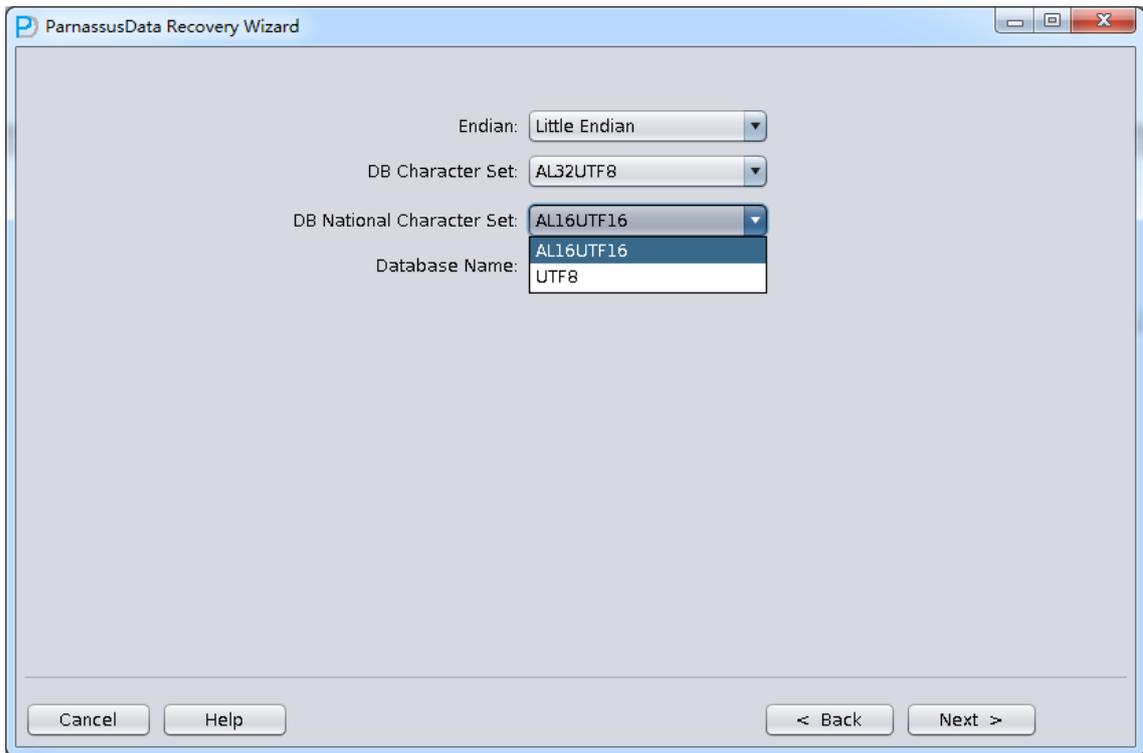
D 公司的运维人员误操作删除了核心数据库的 SYSTEM 表空间 FILE#=1 的数据文件以及部分应用表空间，导致数据库无法正常打开。

此场景下可以通过 PRM 的《Non-Dictionary Mode(ASM) 》ASM 下的非字典模式基于现有的数据文件尽可能恢复出数据。

其简要的流程如下：

1. Recovery Wizard
2. Non-Dictionary Mode(ASM)
3. 加入必要的 ASM DISK(你所要恢复数据库的所在的 ASM Disk Group 的所有 ASM DISK)
4. 点击 ASM analyze
5. 为后面的数据文件选择合适的 Endian 以及字符集(由于是非字典模式所以需要手动选择字符集)
6. 在 ASM analyze 给出的数据文件列表中选中需要的数据文件，如果嫌麻烦且只有一套库，那么可以勾选”Select all”
7. 点击 scan 按钮，后续的恢复与《场景 5》中类似





FAQ 常见问题解答

1. 我不知道我的数据库的字符集信息怎么办？

你可以通过 ORACLE 告警日志 alert.log 来大致了解你的数据库字符集信息，例如：

```
[oracle@mlab2 trace]$ grep -i character alert_Parnassus.log
Database Characterset is US7ASCII
Database Characterset is US7ASCII
alter database character set INTERNAL_CONVERT AL32UTF8
Updating character set in controlfile to AL32UTF8
Synchronizing connection with database character set information
Refreshing type attributes with new character set information
Completed: alter database character set INTERNAL_CONVERT AL32UTF8
alter database national character set INTERNAL_CONVERT UTF8
Completed: alter database national character set INTERNAL_CONVERT UTF8
Database Characterset is AL32UTF8
Database Characterset is AL32UTF8
Database Characterset is AL32UTF8
```

2. 为什么我使用 PRM 总是闪退或者报一些例如” gc warning: Repeated allocation of very large block (appr.size 512000)”的 GC 报错？

就目前 ParnassuData 看到的案例而言，绝大多数此类问题都是由于使用了非推荐的 JAVA 环境所造成的；特别是在 Linux 平台上使用了 redhat gcj java 的话很容易造成该问题。ParnassusData 建议用户使用 ORACLE 提供的 JDK1.6 以上环境运行 PRM，可以直接使用 \$JAVA_HOME/bin/java -jar prm.jar 来启动 PRM。

各个操作系统平台的 JDK 1.6 的下载连接如下：

<http://www.oracle.com/technetwork/java/javasebusiness/downloads/java-archive-downloads-javase6-419409.html#jdk-6u45-oth-JPR>

3. 如果我发现了 PRM 的 bug，我应当如何 report bug 给 ParnassusData?

ParnassusData 欢迎任何人给我们 report bug，只需要发邮件到 report_bugs@parnassusdata.com 就好了，建议提交 bug 时附上您详细的运行环境 包括操作系统、JAVA 运行环境和 ORACLE 数据库版本的环境信息。

4. 我启动 PRM 出现下面的错误怎么办?

```
Error:      no      'server'      JVM      at      'D:\Program      Files
(x86)\Java\jre1.5.0_22\bin\server\jvm.dll'.
```

这是由于用户的环境中安装的是 JAVA Runtime Environment JRE，而没有安装 JDK。而启动 PRM 的脚本中加入了-sever 的选项，该选项在 JRE 版本 1.5 之前是没有的，所以会出现该错误。

ParnassusData 建议用户使用 ORACLE 提供的 JDK1.6 以上环境运行 PRM。

各个操作系统平台的 JDK 1.6 的下载连接如下：

<http://www.oracle.com/technetwork/java/javasebusiness/downloads/java-archive-downloads-javase6-419409.html#jdk-6u45-oth-JPR>

5. 为什么我在使用 PRM 时中文显示乱码?

目前已知有 2 种可能导致该中文显示乱码问题：

- PRM 所运行操作系统环境没有安装中文语言包，例如 PRM 运行在没有安装中文语言包的 Linux 平台上，那么 PRM 作为一个软件是无法正常显示中文的。
- 操作系统已经安装了必要的语言包，但启动 PRM 使用的是 JDK 1.4 的 JAVA 运行环境，同样可能出现中文乱码，该问题建议使用 JDK 1.6 或以上版本来解决

Find More

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

Technical Support: service@parnassusdata.com

Sales: sales@parnassusdata.com

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

Contact: <http://www.parnassusdata.com/zh-hans/contact>

Conclusion

ParnassusData

ParnassusData Corporation , Shanghai , GaoPing Road No. 733 . China

Phone: (+86) 400-0625-198

ParnassusData.com

Facebook: <http://www.facebook.com/parnassusData>

Twitter: <http://twitter.com/ParnassusData>

Weibo: <http://weibo.com/parnassusdata>

Copyright © 2013, ParnassusData and/or its affiliates. All rights reserved. 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, 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.

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

AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. UNIX is a registered trademark licensed through X/Open Company, Ltd. 0410

Copyright © 2014 ParnassusData Corporation. All Rights Reserved.