

ODM 管理介绍

在 AIX 上, 通过 ODM 来维持 RISC/6000 硬件配置、连接信息、操作系统配置和软件产品信息。这是一种比传统 UNIX 所使用的 ASCII 文件更具活力、更安全、并且可分享的资源管理方式。

ODM 管理的数据信息包括:

- 设备配置信息
- 用于 SMIT 的显示信息 (菜单、选择框和对话框)
- 安装和更新用到的软件产品信息
- 通讯配置信息
- 系统资源信息

大多数的 ODM 中定义类和对象存储在 /usr/lib/objrepos 目录; 其他存储 ODM 信息的目录有:

- /usr/lib/objrepos
- /usr/share/lib/objrepos
- /etc/objrepos

ODM 的基本内容就是对象类和对象。通过使用 ODM 专用的命令和子程序来管理这些对象类和对象。具体地说, 就是利用这些接口程序为数据存储和管理创建对象类和对象。

※ ODM 概念

ODM 概念摘要

项目	定义	C 程序中类似的定义	文件中类似的定义
对象类	一个具有相同定义的对象 的存储集合	一个 C 语言结构数 组	一个固定格式的文件
ODM 对象	一个已定义的 ODM 类的 成员, 一个要求管理和存 储数据的实体	结构数组中的一个单 元	一条固定格式的记录
ODM 对象数据库	ODM 对象类的存储集合	结构数组的集合	包含文件的目录

一个对象类由一个或更多的描述符组成。当对象被加到一个对象类时, 每个对象的描述符均带有相应的值。每个对象的描述符和他们相关的值, 能通过 ODM 专用程序定位和修改。

所有物理卷、卷组和逻辑卷的信息都保存在 ODM 中。这些信息相当于 VGDA 信息的拷贝。例如, 导入卷组(importvg)的过程, 包含了将所导入卷组的 VGDA 数据拷贝进 ODM 的过程。相反地, 当导出卷组 (exportvg)时, 关于该卷组在 ODM 中所保持的数据将从 ODM 数据库删除。

※ ODM 命令

这些命令实现创建、增加、变化、检索、显示和删除与 ODM 相关的对象和对象类等功能。

注意：

仅仅在传统的系统维护方法（例如 SMIT）无效时，才可以使用 ODM 命令。作为一个刚入门的系统管理员，强烈推荐你在使用这些命令前多查阅资料。不正确的使用这些命令可能导致系统不可用。这里所描述 ODM 命令仅为介绍目的。

这些命令包括：

- **Odmadd**：把对象加到一个对象类中。odmadd 命令以一个 ASCII 节文件作为输入参数，然后把在节文件中定义的对象组合成对象类。
- **Odmchange**：改变在指定的对象类中特定的对象。
- **Odmcreate**：创造空对象类。odmcreate 命令把描述对象类的 ASCII 文件作为输入参数，并生成存取这些对象类的 C 语言应用程序所用的.h 和.c 文件。
- **Odmdelete**：从一个对象类中删除对象。
- **Odmdrop**：删除整个对象类。
- **Odmget**：从对象类中检索对象，并以 ASCII 节文件的格式（与 odmadd 命令输入参数的格式相同）输出。
- **Odmshow**：显示一个对象类的描述。odmshow 命令以一个对象类名字作为输入参数，并把对象类信息以 ASCII 格式（与 odmcreate 命令输入参数的格式相同）输出。

※ ODM 数据库中对象类的例子

下面是一个关于当前配置设备数据库的对象类定义的例子（CuDv）：

odmshow CuDv

```
class CuDv {
char name[16];                      /* offset: 0xc ( 12) */
short status;                      /* offset: 0x1c ( 28) */
short chgstatus;                   /* offset: 0x1e ( 30) */
char ddins[16];                    /* offset: 0x20 ( 32) */
char location[16];                 /* offset: 0x30 ( 48) */
char parent[16];                   /* offset: 0x40 ( 64) */
char connwhere[16];                /* offset: 0x50 ( 80) */
link PdDv PdDv uniquetype PdDvLn[48]; /* offset: 0x60 ( 96) */
};
/*
descriptors: 8
structure size: 0x98 (152) bytes
data offset: 0x20001cd8
population: 50 objects (50 active, 0 deleted)
*/
```