

PL/SQL Developer

使用手册

V7.0

1 介绍

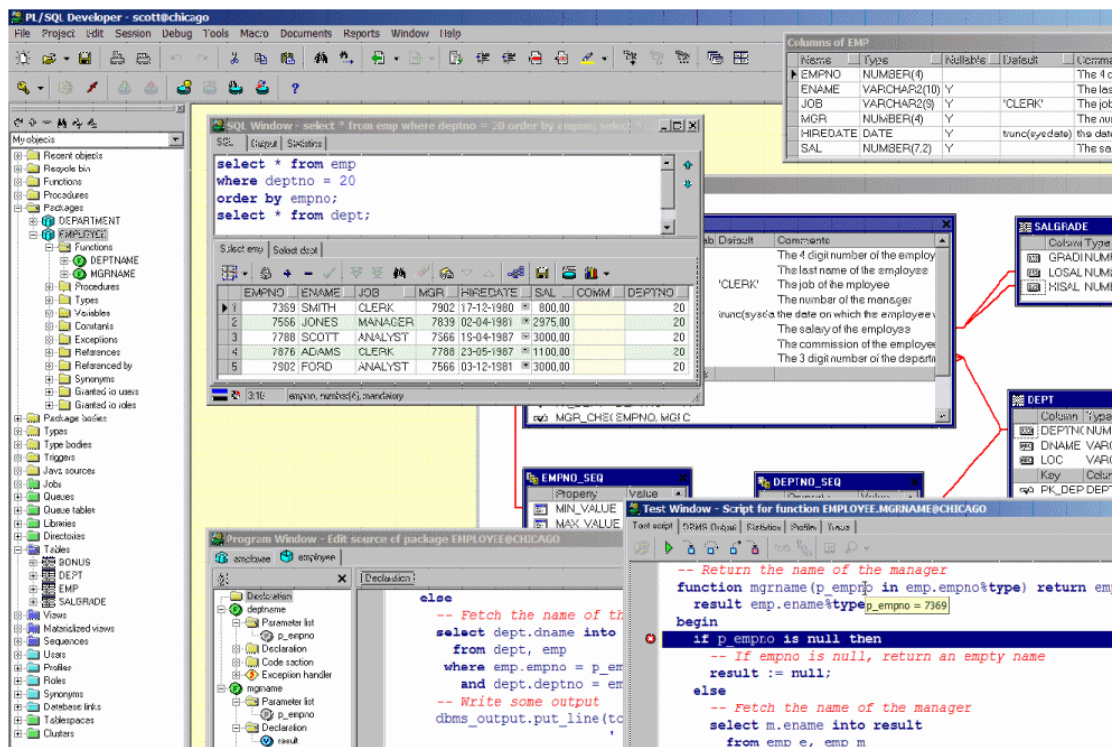
PL/SQL Developer 是一个开发 Oracle 数据库程序的集成开发环境 (IDE)。利用 PL/SQL Developer, 你可以很方便地创建 Client/Server 程序中的 Server 部分。

如果到现在你还像以下的方式开发是件很糟糕的事情:

- 使用文本编辑器写程序单元（存储过程，触发器，等等）。
- 使用 Oracle 的 SQL*PLUS 编译原文件。
- 如果编译错误，你必须在源代码文件中找到错误，修改后，切换到 SQL*PLUS 中重新编译，不料竟会发现其他错误。
- 使用 SQL*PLUS 或前端程序测试程序单元。
- 如果运行出错，你得又一次努力的寻找错误并改正它。
- 你使用 Explain Plan 或 tkprof 优化你的 SQL。
- 使用 SQL * Plus 或其他工具，来查看或修改数据库中的其他对象和数据。

这些工作——编辑、编译、改正、测试、调试、优化和查询——全部可以在 PL/SQL Developer 的集成开发环境中完成。

此外，PL/SQL Developer 提供了许多其他工具，这些工具对日常的 PL/SQL 开发是很有帮助的。



1.1 编辑 (Editing)

PL/SQL Developer 和其他重要的开发环境一样，假定你将源代码文件保存在磁盘上。其他工具只是让你在数据库中编辑源代码，但是这样不支持版本控制和部署方案。源代码文件可以在 SQL*PLUS 中运行，所以，不用 PL/SQL Developer 就可以在任何平台上部署。你可以通过多文档界面同时编辑多个文件。

编辑器为程序员提供了广泛的辅助功能，对于 SQL 语句和 PL/SQL 命令提供了上下文关系帮助，We've all been there: 你开始键入 substr，但是忘记了参数的准确含义，现在你只要简单地按 F1 键就能获得 SQL 参考手册中的相应主题。表、视图和程序单元都可以用同样的方式在编辑器中使用滚动窗口的形式为你描述出来。代码助手被集成在编辑器中，当键入数据库对象名时自动显示数据库对象信息，允许你查看和选择对象。对于大的包和类型体，代码编辑器为简化导航，提供了包含代码内容的树型视图，提供了后退和前进按钮，并提供超链接导航。查询设计器允许你图形化创建 select 语句。PL/SQL Developer 的可扩展模板可以方便地在他的程序中插入标准 SQL 和 PL/SQL 代码。所有的编辑器都使用适当的 SQL，PL/SQL 和 SQL*PLUS 语法高亮显示增加程序的可读性。

1.2 编译与纠错 (Compiling & correcting)

在编辑器中可以编译代码文件，当编译错误时自动定位到错误代码行。所有编译错误在编辑器下方列出，列表中可能包括程序错误或者用户命名规则错误的代码提示。因不保存代码就可以编译程序，你可以很安全地尝试不同的解决问题方法。文件修改后未保存或未编译时，编译器记住这个事实。

1.3 测试和调试 (Testing & debugging)

测试程序单元可以写一个测试脚本，在测试脚本主体包含的一个 PL/SQL 块中编写测试代码，在 PL/SQL 块中你想使用的任何变量都可以被声明、被赋输入值，还能在运行后查看。

当你运行一个测试脚本，可能发生一个运行错误，在这种情况下，PL/SQL Developer 允许你查看错误堆栈对应的源代码，运行错误涉及到的错误堆栈源代码行均被高亮显示，所以，你很容易跟踪到有问题的地方。

如果你使用的是 Oracle 7.3.4 或更高版本，你可以使用 PL/SQL Developer 的集成调试，可以单步执行，设置断点，查看和设置变量，查看调用堆等等。

Oracle8i 或更高版本，你可以另外使用 dbms_trace 功能记录选择的程序运行事件，可以帮助你分析程序流程和异常情况。

调用 dbms_output 和 PL/SQL Web Toolkit 包的输出结果被自动显示在测试窗口的相应标签页中。

作为回归测试，你可以使用 Test Manager 快速运行和成组校验测试脚本。

1.4 优化 (Optimizing)

为了优化程序单元中的 SQL 语句，Oracle 的 Explain Plan 应用提供了很大的帮助。所以，被集成到 PL/SQL Developer 的集成开发环境中。只需要简单地在源文件中选择 SQL 语句后按 F5 键，打开 Explain Plan 窗口并可视化显示该查询的执行计划。你可以直接修改语句优化查询，然后拷回到源文件中。

你也可以查看被执行的 SQL 语句和 PL/SQL 程序单元的统计信息，这些统计信息包括执行时间，CPU 时间，逻辑读，物理读，物理写等等。

Oracle8i 引进了一个 PL/SQL Profiler (PL/SQL 的一个调试优化跟踪方案的应用)，允许跟踪你的 PL/SQL 代码方案，对于已执行的每行代码，你可以测定运行时间和运行次数。

1.5 查询 (Querying)

你可以使用 SQL window 执行任何 SQL 语句查询数据库中的数据。所有已执行的语句都被保留在历史缓存中，所以，你可以很容易地再次执行。任何查询结果都显示在一个单独的表格中，随后你可以插入、更新和删除数据。数据结果表格还可以被用在 Query By Example 模式下，以便容易找到你所需要的数据。

你可以使用 Object Browser 查询数据库对象。数据库所有相关属性都可以查看，例如：表、视图、序号、函数、过程、包、类型和触发器，包括对象间的依赖关系。Browser 和 Windows 的浏览器一样使用树型视图支持鼠标的点击浏览方式。

1.6 运行 SQL 脚本 (Running SQL scripts)

PL/SQL Developer 包括一个可以运行 SQL 脚本或执行 SQL 语句的 Command Window。你还可以执行类似于你所熟悉的 SQL*PLUS 命令的命令。

1.7 创建和修改表定义 (Creating and modifying table definitions)

你可以不使用任何 SQL 语句创建和修改表定义，只在一个对话框中填写定义就可以修改字段，约束，索引，权限，存储空间，注释等等。这个定义可以提交到数据库，并查看，修改和保存对应的 SQL。

1.8 视图 (Diagrams)

你可以创建视图可视化你的应用或项目的数据库对象和它们之间的关系。一个视图可以被用来编制文档，并且还可以充当有关联对象的一个工作区。

1.9 报表 (Reporting)

PL/SQL Developer 带有许多基于 HTML 格式的报表，你可以在 PL/SQL Developer 中查看和打印这些报表，或保存为 HTML 文件。你也可以创建自己的报表。通过报表 (Report) 菜单可以很容易地创建报表。

1.10 图表 (Graphics)

在 SQL Window 和 Report Window 中可以打开 Graph Window 获得查询数据的图表显示。

1.11 工程 (Projects)

可以使用 PL/SQL Developer 的工程概念组织你的工作，一个工程包括很多文件和数据库对象。这些对象通过 Project Items Window 访问，并且可以简单地通过鼠标点击编译。

1.12 工具 (Tools)

PL/SQL Developer 提供许多工具帮助你的开发。这些工具包括在数据库对象中查找文字的查找数据库对象工具 (Find Database Object tool)、快速编辑在开发过程中变为非法对象的编译非法对象工具 (Compile Invalid Objects tool)、导入和导出工具、创建测试数据的数据生成工具 (Data Generator tool)、导出用户对象的 DDL 语句的导出用户对象工具 (Export User Objects tool)、比较两个用户对象定义的比较用户对象定义工具 (Compare User Objects tool)、会话信息工具、事件监视器、比较和均衡表容量的比较表数据工具 (Compare Table Data tool)。

除了这些标准的工具外，你可以在 PL/SQL Developer 的工具菜单下定义自己的工具项。

2 程序安装

有两个基本方法安装 PL/SQL Developer:

- 工作站安装。在这种情况下，你将在每个要使用的工作站上本地安装软件。
- 服务期安装。在这种情况下，你在一个服务器上的每个要使用的工作站可以访问的目录中安装软件。

两种安装类型将在这里解释。

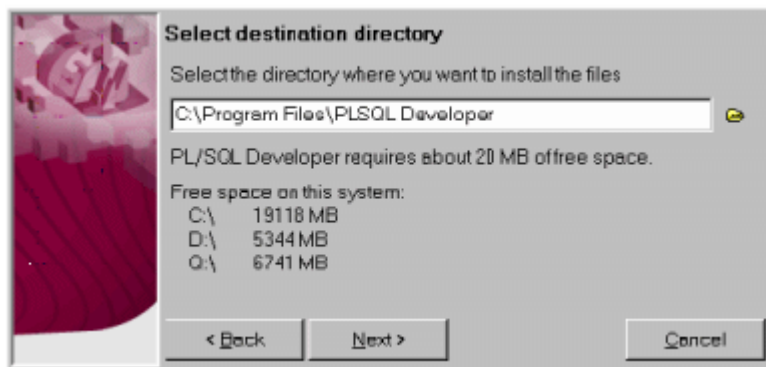
2.1 系统要求 (System requirements)

PL/SQL Developer 是一个 32 位的 Windows 应用程序。所以，你在工作站上必须至少安装了 Windows 95 或者 Windows NT 4。要连接一个 Oracle 数据库，PL/SQL Developer 需要

一个 32 位版本的 SQL*Net, Net 8, Net 9 或者 Net 10。

2.2 工作站安装 (Workstation installation)

要在工作站上本地化安装 PL/SQL Developer, 从安装介质上运行 setup 程序。这样做之后, 出现下面对话框:



你可以为程序文件选择一个目标目录, 一个在启动菜单中的文件夹, 一个 PL/SQL Developer 桌面快捷键位置, 和安装选项。在最后一页按完成键后, 程序文件被拷贝, 快捷键被创建。

2.3 基于服务器安装 (Server based installation)

对于多用户许可, 你可以在一个文件服务器上创建单个 PL/SQL Developer 安装并提供所有许可用户存取权限。要这样做, 你可以像前部分描述的相同方法在文件服务器上装配程序。你必须选择一个对于使用软件用户共享的目标目录。当 PL/SQL Developer 初次运行在一个工作站上, 这个安装类型被自动检测, 并在用户没有注意的情况下完成本地安装。

PL/SQL Developer 可以被安装到一个用户只有读取权限目录, 可是也有一些例外:

Macros 子目录必须可写, 因为这是全部宏被保存的地方。

Preferences 子目录必须可写, 因为这是放置个人参数选项集的地方, 详见参数选项集章节 (Preference sets)。

2.4 脚本安装 (Scripted installation)

除上面描述的安装过程外, 你也可以为无人值守安装创建一个脚本。阅读包含的 install.txt 文件了解更进一步信息。

2.5 卸载 PL/SQL Developer (Uninstalling PL/SQL Developer)

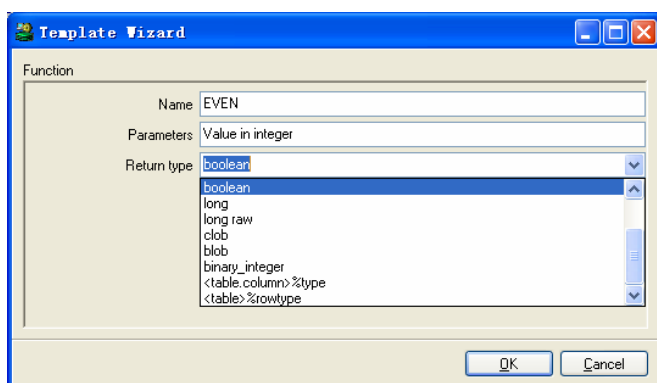
如果因为一些原因你想卸载 PL/SQL Developer, 你可以使用控制面板中的添加/移除程序 (Add/Remove Program) 移除它。

3 编写程序

在 Oracle 数据库中，存储过程分为五种类型：函数、过程、包、类型和触发器。这五种类型在编辑器中使用完全统一的方法创建和修改。

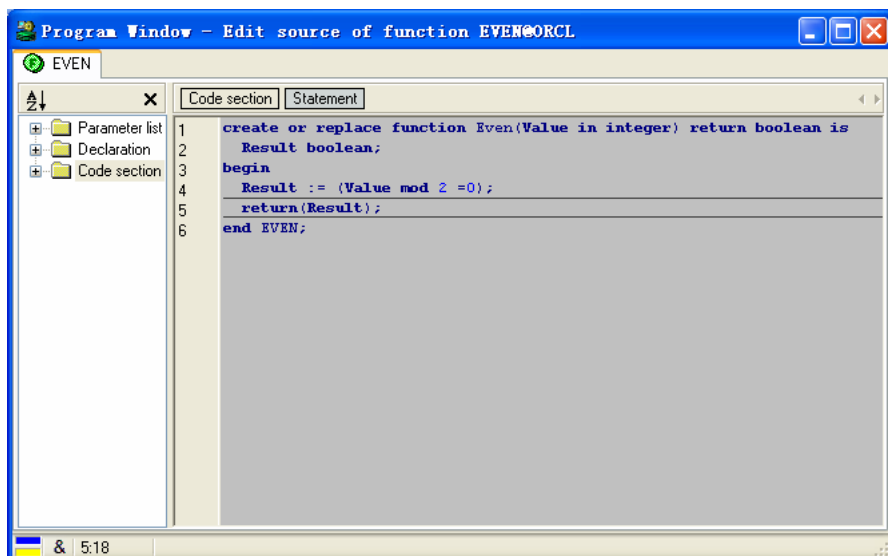
3.1 创建一个程序（Creating a program）

创建一个新程序，在工具条中按 New 键，选择 Program Window 菜单项，例如 Function 子菜单项。程序提示和程序类型有关的变量，当我们创建函数时包括：函数名、参数列表和返回类型：



从对话框标题中可以知道这些信息来自于模版。PL/SQL Developer 提供许多标准模版，你可以根据需要修改，你也可以自己创建新的模版，定义模版的有关信息参见模版章节（Templates）。

当你输入完变量并按 Ok 键后，出现程序编辑器窗口（Program Editor Window）并按函数模版生成代码。每个在程序编辑器中创建的程序，其单元开始必须使用 'create or replace' 的 SQL 语法。以上面为例，我们创建了 Even 函数，源文件看起来像这样：

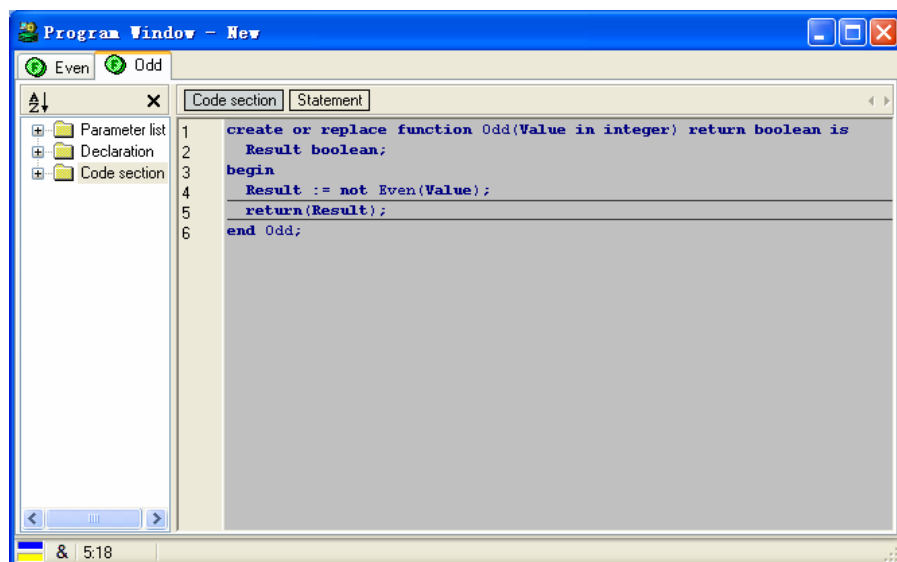


在编辑器的左侧你可以看到显示代码单元结构的代码目录，在包体和类型体这样的大程序中导航是非常有用的。详细描述见代码目录章节（Code Contents）。

在编辑器上方是代码层次，显示光标所在代码的层次，在当前情况下，程序显示你现在所处该函数中的顺序语句中的一行语句（返回语句）上。对于复杂程序来说，能够帮助你知道当前所在的具体位置，并且可以帮助你选择部分 PL/SQL 代码，具体描述见代码层次章节（Code Hierarchy）。

一个程序文件可以包含多个程序单元，右键点击程序编辑器，出现弹出菜单，可以添加和删除程序单元。你可以通过选择窗口顶端的页标签切换程序单元，这样你可以很方便地将有关联的程序单元组织在一个文件中，一个包说明和包体就是一个很好的例子。

程序单元必须放在引用程序单元的后面，如果你创建一个函数'odd'引用前面创建的'even'函数，程序编辑器应该像这样：



3.2 保存程序（Saving a program）

保存程序可以点击工具条中的 Save 键，弹出一个带程序扩展名的保存对话框：

程序类型	扩展名
函数	.fnc
过程	.prc
包说明和包体	.pck
包说明	.spc
包体	.bdy
类型说明和类型体	.typ
类型说明	.tps

类型体	.tpb
触发器	.trg
Java 代码	.jsp

你可以通过首选项来改变这些扩展名，详解文件-扩展章节（Files – Extensions）。如果程序只有一个函数、过程、包、类型、触发器或 Java 代码，则该文件名将自动由程序单元名和类型对应的扩展名构成。文件保存后，文件名在窗口标题栏中显示。

被保存文件和 Oracle SQL*Plus 格式兼容，包含 'odd' 和 'even' 函数文件如下：

```
create or replace function Even(Value in integer) return boolean is
    Result boolean;
begin
    Result := (Value mod 2 =0);
    return(Result);
end EVEN;
/

create or replace function Odd(Value in integer) return boolean is
    Result boolean;
begin
    Result := not Even(Value);
    return(Result);
end Odd;
/
```

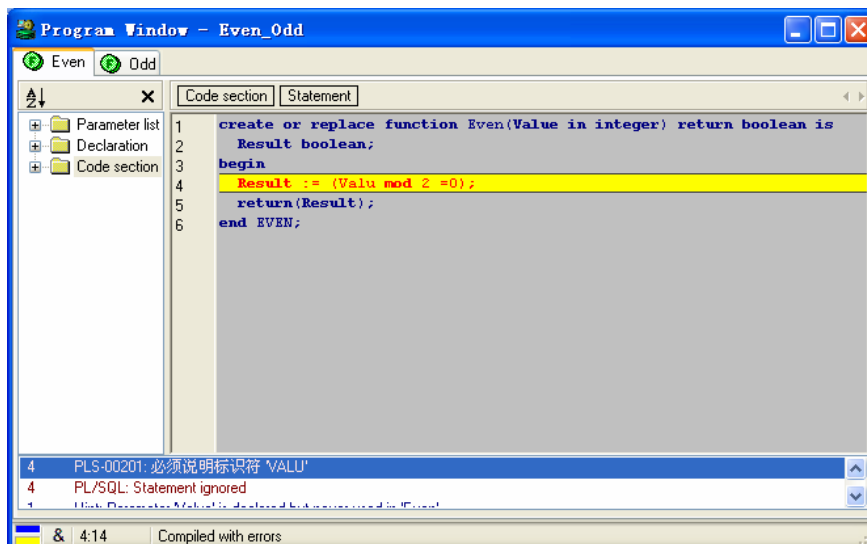
当文件在 SQL*PLUS 中执行后，两个函数将被创建。

3.3 修改程序（Modifying a program）

打开以前保存的程序文件可以在工具条中点击 Open 键并选择 Program file 菜单项，如果文件最近被使用过，可以在菜单中选择 File 下的 Reopen 或者点击工具条中 Open 键旁边的箭头。当你编辑时，在窗口下方亮起一个蓝色提示条，说明文件被修改后未保存，还有一个黄色提示条表示文件被修改后未编译。

3.4 编译程序（Compiling a program）

编译文件可以点击工具条中的 Execute 键，不管程序单元是否被选中，从文件中的第一个程序单元开始编译所有的程序单元，当出现编译错误时，编译自动终止，光标定位在产生错误的源代码行上：



如果不止一个错误并且你想纠正错误，可以在窗口下方点击下一个错误转到产生错误的地方。如果 PL/SQL Developer 已经配置了使用 HTML 手册，详见 HTML 手册章节（HTML Manuals），在编译错误上双击可以显示 Oracle 错误代码手册中的相应段落（Oracle Error Messages manual）。

编译错误列表中可能包括提示，提示不会造成编译失败，但是他们可能造成运行错误（例如空值比较，函数没有返回值，未使用变量等等），有关提示的更多信息见 Window Types - Program Window。

如果编译后在信息栏中出现错误信息，意味着源文件未被编译，创建失败。错误信息应该能解释错误原因，例如，如果你得到的 create or replace 语法错误，信息栏中会显示“ORA-00900: Invalid SQL statement”。

注意：如果你使用的是 Oracle Server 7.2 或更低版本，信息框中总是显示一个触发器编译错误。从 Oracle Server 7.3 版本开始，触发器的编译错误同其他程序类型。

3.5 替换变量（Substitution variables）

如果在程序中同 SQL*PLUS 一样（用与符合作为前缀）使用了替换变量，你可以点击窗口下方的&键，文件中的源程序将扫描替换变量，随后你可以对替换变量赋值。如果保存窗口状态首先项可以使用（见 Window Types - Program Window），这些替换变量值会被保存和在随后的程序文件或数据库对象被浏览和编辑时被还原（Restored）。

4 测试程序

成功编译程序后，需要对程序进行测试。要达到这个目的，你可以使用 PL/SQL Developer 的测试脚本，一个测试脚本允许你运行一个或多个程序单元、定义输入，输出、输入/输出变量和对变量的查看和赋值，如果在运行测试脚本期间出现一个运行错误，你可

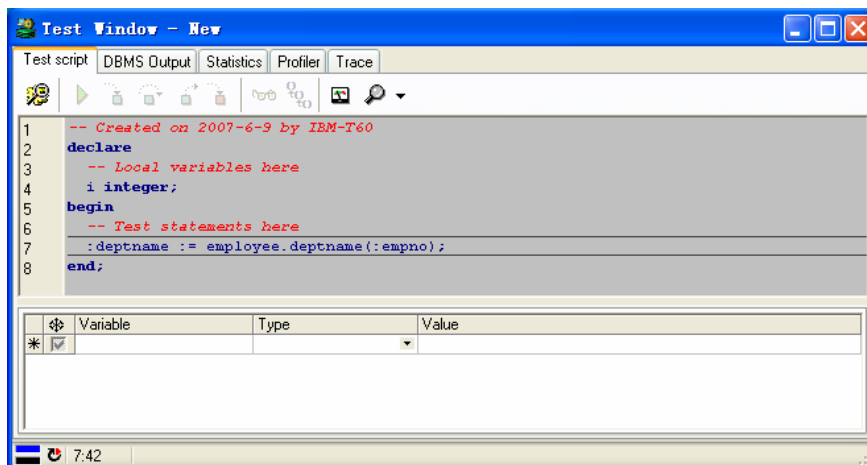
以查看造成错误的程序单元代码。

如果是 Oracle 7.3.4 或更高版本，你可以使用 PL/SQL Developer 的基础测试环境，单步跟踪代码、设置断点、查看和设置变量值、查看调研堆栈等等。

4.1 创建测试脚本（Creating a Test Script）

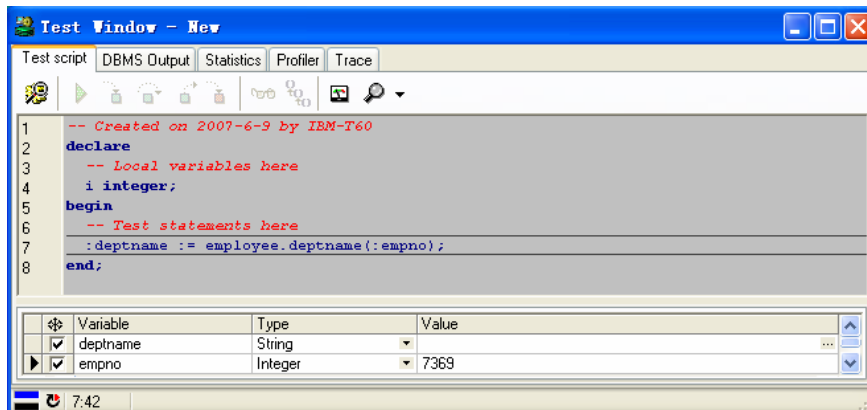
创建测试脚本有两种方法，右键点击浏览器中的程序单元（独立的函数或过程、打包的函数或过程、对象类型方法）选择 Test 菜单项，这样将创建一个带有参数和返回变量的测试脚本，为了说明问题，我们将手工从头开始创建测试脚本。

创建一个空的测试脚本，按工具条中的 New 键选择 Test Window 菜单项，你现在可以用类似于 declare...begin...end 语法在测试脚本中录入一个 PL/SQL 块。作为一个例子，我们将测试一个职工（employee）包，这个包实现了一些职工/部门演示表的职工函数。deptname 函数返回职工的部门名称，可以使用下面的 PL/SQL 块测试：



PL/SQL 块包含了一个简单的对于我们要测试函数的调用，并使用 deptname 和 empno 变量，我们必须给 empno 赋值，在运行后判断函数是否返回正确的 deptname。

使用冒号前缀的变量可以在测试脚本下方声明：



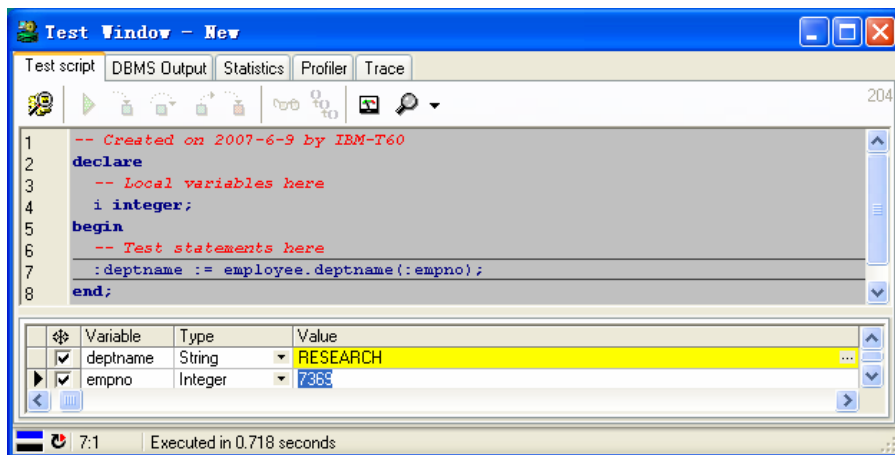
在变量列表左上角的扫描代码变量键（）可以快速拷贝代码中的变量，在此之后，

将整数类型和 7369 赋予 empno 变量。现在我们已经准备好执行脚本。

你也可以在 PL/SQL 块中的声明部分声明局部变量,但是你不能查看和修改这些变量值。这些局部变量的一个好处是可以用来记录类型、PL/SQL 表等等。

4.2 执行测试脚本 (Executing a Test Script)

按工具条上的 Execute 键, 执行测试脚本。脚本执行后, 变量的值被更新, 任何被更新的变量都用黄色背景显示。



函数的返回值是 'RESEARCH', 所以函数显然是正确的。在窗口的底部显示执行时间。这个信息可以用来优化你的代码性能。为了优化的目的, 你还可以选择统计页 (statistics) 查看 PL/SQL 块的执行统计信息。你还可以在脚本执行前按创建跟踪报表键 (Create Profiler report) 来创建每行 PL/SQL 代码的统计信息, 执行后你可以切换到优化跟踪页 (Profiler) 查看报表。统计和跟踪报表分别在自动统计 (Automatic statistics) 和 PL/SQL 跟踪 (The PL/SQL Profiler) 章节中解释。

你可以按 Break 键放弃执行测试脚本, 在程序进入死循环或者执行时间比预期要长时格外有用。需要注意的是按 Break 键并不总是成功。例如, 如果程序在等待锁时不响应终止信号。

执行脚本后, 一个事物可能被执行的程序单元打开, 如果这样, 工具条中的提交键和回滚键就可以使用, 对于事物的更多信息见窗口, 数据库会话和事物 (Windows, database sessions and transactions) 章节。

4.3 变量类别 (Variable types)

测试窗口 (Test Window) 中变量部分中的类型列表包含了全部可以使用的类型。这些类型的特性将在本章中解释。

类型	描述
Integer	能被用于数字值, 范围为 -2^{32} 到 $2^{32} - 1$ 。
Float	能被用于其它数字值。
String	相当于表里的 varchar2 数据类型, 能容纳 2000 (Oracle7) 或4000 (Oracle8) 个字符。此数据类型的值的单元格里有一个能调用文本编辑器的单元格按钮, 所以你能容易地查看和修改多行字符串。

Date	日期和时间的数据类型。
Long	数据类型不在表格里显示它的值，只显示 <Long>，按单元格按钮可以调出文本编辑器。
Long Raw	按 Long Raw 值的单元格按钮将调出一个导入/导出文件对话框，允许你载入或保存这个变量的内容。
Cursor	此数据类型用于PL/SQL 里你能使用游标变量的地方。运行了测试脚本后，你可以按值的单元格按钮以在 SQL 窗口里显示结果集。
CLOB, BLOB与 BFile	在你查看 LOB 数据以前，这些 LOB 定位变量必须在服务器上被初始化。CLOB 将调用文本编辑器， BLOB 与 BFile 将调用导入/导出文件对话框。
PL/SQL String	为 PL/SQL 使用此数据类型， varchar2 值可以达到 32000 个字符。
Char	定长字符串（空格被填补）。
Substitution	没有绑定变量的限制，置换变量就能被使用。在它被发送到服务器以前，PL/SQL 块里的变量名会被它在文本里的值置换。
临时CLOB与BLOB	在它们通过测试脚本被传送到服务器以前，这些 LOB 定位器已经可以持有数据了。

布尔变量

当你选择变量类型列表时，你会发现没有布尔数据类型。这是因为 SQL*PLUS 不支持这种数据类型。要使用布尔变量，你可以声明成一个整型变量并且使用 Oracle 提供的 sys.diutil.bool_to_int 和 sys.diutil.int_to_bool 函数。这些函数在 null/true/false 和 null/0/1 之间转化。如果你在浏览器中使用 Test 函数，这个转化自动为你完成。

4.4 保存测试脚本（Saving Test Scripts）

保存测试脚本按工具条中的 Save 键。保存对话框以 .tst 作为测试脚本的扩展名，然而你可以通过首选项修改默认的扩展名，详见文件-扩展名 (Files - Extensions) 章节。PL/SQL 块、所有变量（名称、类型和值）和调试监视（见调试 Debugging 章节）都被保存下来。由于保存了测试脚本，在程序被修改后可以很容易地进行再测试。

你可以通过按工具条中的 Open 键，选择测试脚本在新的测试窗口打开原保存的测试脚本。或者通过在已创建的测试窗口中点击右键选择 Load 菜单项，这将在当前的测试窗口中打开测试脚本。

Saving as SQL*Plus script

```
rem PL/SQL Developer Test Script
```

```
set feedback off
```

```
set autoprint off
```

```
rem Declare variables
```

```
variable result varchar2(2000)
```

```
variable p_empno number
```

```
rem Set variables
```

```
begin
```

```
  :result := null;
```

```
  :p_empno := 7369;
```

```
end;
```

```
/
```

```
rem Execute PL/SQL Block
```

```
begin
```

```
  :result := employee.deptname(:p_empno);
```

```
end;
```

```
/
```

```
rem Print variables
```

```
print result
```

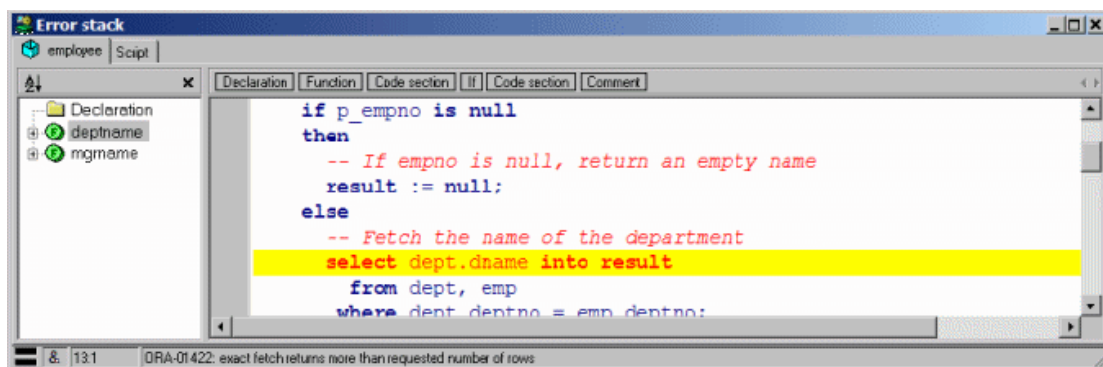
```
print p_empno
```

你可以选择将测试脚本保存为兼容 SQL*PLUS 格式。这样可以在 PL/SQL Developer 不能运行的环境下执行脚本（例如在 Unix 服务器上）。要做到这点，在保存对话框的文件类型中选择 SQL*Plus script (*.sql)。前面的脚本将被保存成：

注意：你不能再将一个 SQL*PLUS 脚本按测试脚本打开。如果你想在测试窗口重用它，一定要保证你同时将它保存成测试脚本！

4.5 跟踪运行时间错误

每当你的程序单元发生运行时间错误，PL/SQL Developer 将询问是否愿意查看错误堆栈的源代码。这样你可以快速地发现错误原因。如果在我们的 deptname 函数中有一个错误，错误堆栈看起来像这样：



在顶部你能看到每个和运行错误相关的程序单元页标签。这些程序单元按调用顺序从左到右排列。通过翻动标签页可以简单地发现导致错误的程序流程。在这里显示测试脚本已经调用了 employee.deptname 函数。这很明显可以获得更复杂，甚至通过触发器的信息。

注意：如果你使用的是 Oracle Server 7.2 或更早版本，触发器代码不在错误堆栈窗口中显示。

错误堆栈使用红色高亮显示和错误有关的代码行。在堆栈中的最后程序单元代码行产生错误。其他所有的程序单元只是错误发生时调用其他程序单元的代码行（意思是说其他程序单元红色标记的代码行只是错误发生时的调用关系——译者注）。

4.6 包状态和 Java 会话状态（Package states & Java session states）

当你编辑、编译和测试包时，包声明在每次编译后被 Oracle 清除。在包中的所有全局变量被重置，并且初始块将被重新执行。这可能导致意外结果。每当 PL/SQL Developer 检测到这种情况，在测试窗口的状态栏中提示一个警告：“Warning: ORA-04068: Existing state of packages has been discarded”。（包的现有状态被清除）

同样的，如果你编辑和编译一个 Java 代码，所有现有的 Java 会话状态将被清除。PLSQL Developer 也将控制这种情况，并在状态栏中显示“Warning: ORA-29549: class <class name> has changed, Java session state cleared”（类<类名>已改变，Java 会话状态被清除）。

4.7 查看结果集（Viewing result sets）

在测试脚本中，PL/SQL 块的数量不受限制，你也可以执行单个 SQL 语句，包括 select 语句。执行一个 select 语句将显示另一个结果标签页，以显示所有查询出的数据行。

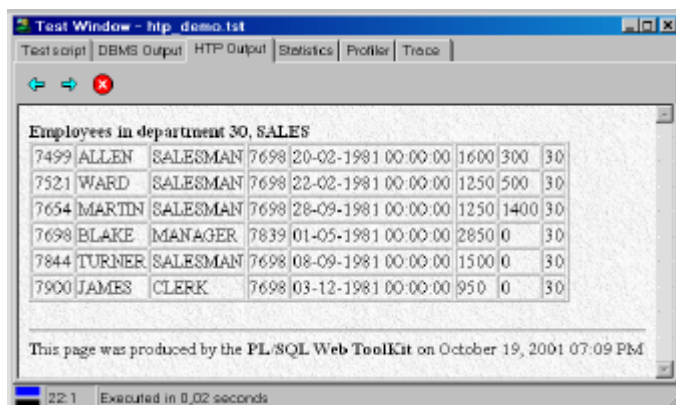
4.8 查看 dbms_output（Viewing dbms_output）

为了调试目的，有时必须在程序单元中“打印”一些信息在屏幕上。为了达到这个目的，Oracle 创建了 dbms_output 包。调用 dbms_output.put_line，信息被放在输出缓存中。

执行测试脚本后，你可以选择测试窗口上方的 Output 页查看输出缓存中的内容。在本页你还可以设置输出缓存的大小或者允许/禁止缓存。在 Output 页中的默认设置被输出参数选项控制，具体描述见 Oracle-输出（Oracle - Output）章节。

4.9 查看 HTP 输出（Viewing HTP output）

如果你要使用 PL/SQL Web 工具箱测试程序单元，生成任何的 HTP 输出将在 HTP Output 页中显示：




如果没有 HTP 输出，该页不显示。


4.10 调试（Debugging）


对于那些编程错误确实很难跟踪，测试窗口提供了一个集成调试器。在窗口上方可以看到一个和测试相关的工具条。要开始一个测试，只需按测试工具条左侧的 Start 键，而不是按主工具条中的 Execute 键。现在，其他按键可以使用，你可以准备测试了。


控制执行


启动调试器后，程序将暂停在测试脚本的第一句之前。在这之后，你可以使用测试工具条中的按键控制程序执行。

 执行脚本直到完成

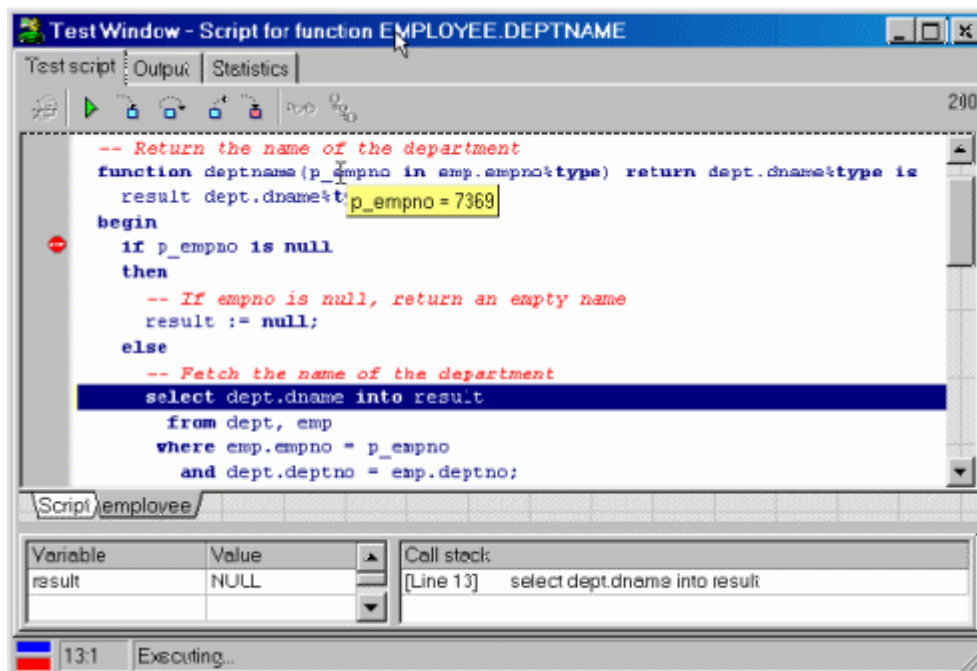
 单步执行进入一个过程、函数或者方法调用的下一行。如果调用的下一行包括更新、插入或删除而引起触发器，你将单步进入触发器。

 单步执行跳过下一行。该行将被执行，但是你不会进入该行。

 单步跳出当前程序单元。

 执行直到发生异常。执行将暂停在发生异常的行上。执行下一步将引发异常。

任何时候你单步进入一个程序单元，该单元将自动被载入测试窗口。浏览器底部显示该程序单元标签页，所以，你可以很方便地在程序单元中查看代码、设置/去除断点等等。当你对程序单元不再感兴趣的话可以在浏览器中点击右键将它移除：



查看和设置变量值

在调试期间查看变量，你可以在代码上移动鼠标。1/2 秒后，该值将被弹出显示。PL/SQL 块中的变量不能显示。虽然你可以看单独的字段值，但是复杂变量（例如记录 and 对象类型）不能显示值。在 Oracle 将来的版本可能去除这些限制。

要设置变量值，右键点击程序单元变量，从弹出菜单中选择 Set Variable 菜单项，出现一个输入字段，字段默认是当前变量值。你可以输入新值，按回车确认，按 ESC 取消操作：



在同样的弹出菜单中你也可以选择增加变量到监视列表中，这就意味着调试每执行一步都将在测试窗口左下方的监视列表（watch list）中实时显示变量值。

如果变量是一个标量数据类型（scalar data type）的集合（PL/SQL 表、数组或嵌套表），你可以通过右键点击它并选择 View collection variable 来查看整个集合。

注意只有程序单元使用调试信息编译（compiled with debug information）才能查看变量值。有一个优先选项可以让编译包含调试信息，你也可以手工增加调试信息，在浏览器中右键点击程序单元，在弹出菜单中选择 Add debug information 菜单项。

使用断点

断点可以让程序执行暂停在一个特定的 PL/SQL 代码行。当执行暂停时，你可以查看和设置变量值，跳过代码等等。你可以在断点上定义条件，当条件满足时执行暂停。每一个断

点都可以定义一个消息，当到达断点时消息被放到输出页中。

设置断点

有两种方法设置断点：在一个程序窗口（Program Window）或在一个测试窗口（Test Window）。在两种情况下，你只要在代码行所在的编辑器左侧边界点击，将出现一个断点标记，表示该行已设置了断点。当你在测试窗口以调试模式执行时，遇到断点则执行将被暂停。

如果你在程序窗口设置断点，可能这个特定程序单元还没有编译到数据库中，所以，这个断点还不能应用到数据库中，在这种情况下断点标记有一个不同的样式。当你编译了程序单元，断点被应用并且断点样式随之改变。只要程序单元没有被编译，任何先前在程序单元设置的断点都将有效。

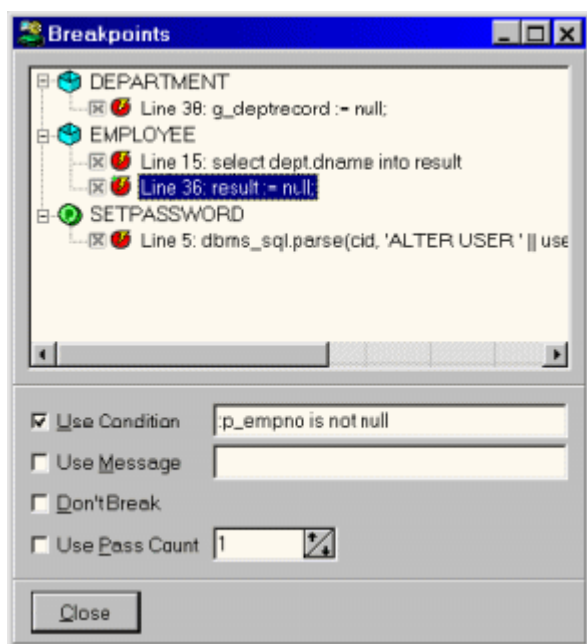
你不能在测试脚本中设置断点。

只要再次点击断点标记就可以删除断点。

断点条件

有时你设置的断点行经常被执行，就算你只对程序在特定的状态下感兴趣，如果是那样，你可以在断点上设置条件，只有满足条件时断点才起作用。

要为断点定义条件，右键点击断点标记，在弹出菜单中选择 Modify Breakpoints 菜单项，出现下面的对话框：



在这个对话框中你可以看到有断点的所有程序单元，断点列在该程序单元下面。每个断点有一个复选框，可以设置允许/禁止。下方部分显示以下字段：

- 使用条件（Use Condition）——复选框允许和禁止使用条件。条件必须是布尔表达式。当执行到该断点行时，只有条件为真时才暂停。你可以在条件中使用任何 SQL 表达式，可以使用任何在断点位置已知的变量，这些变量同在调试交互中可以查看

和设置的变量。变量必须以冒号为前缀，例如，如果 `ename` 在断点处是一个合理的变量，则 `upper(:ename) = 'SMITH'` 是一个正确的条件。

- 使用消息（Use Message）——复选框允许和禁止使用消息，如果执行到断点，并且满足条件的话，该消息被放到输出页。
- 不停止（Don't Break）——这个复选框只有在定义了消息时才有用。当选中时，执行不在该断点行上停止。这样方式是为了输出消息而设置断点。
- 使用执行次数（Use Pass Count）——复选框允许和禁止执行次数，该条件定义执行被暂停前该行必须已经执行了多少次。例如：如果你定义了 10 次，每经过断点行 10 次暂停一次。

查看调用堆栈

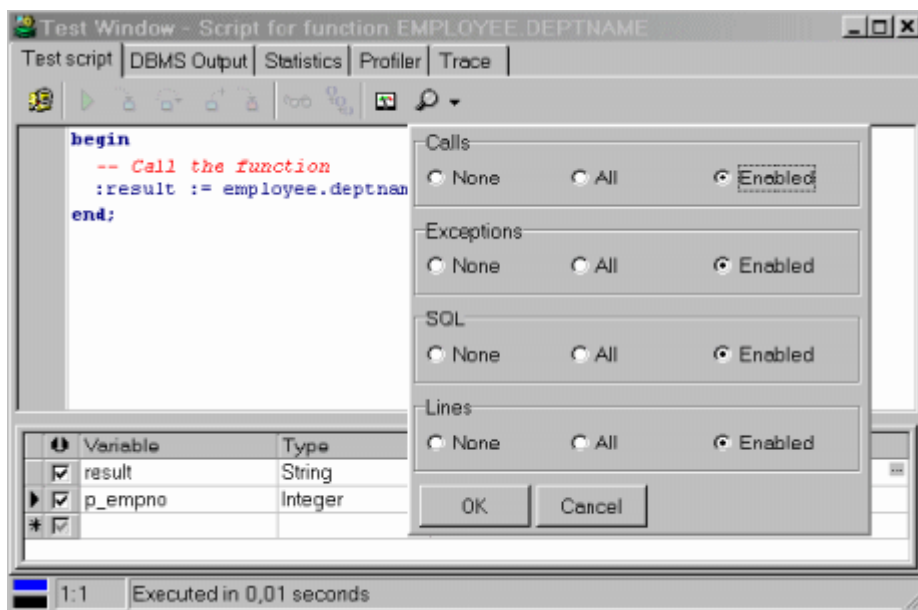
可以在测试窗口的右下方查看调用堆栈。每执行一步调试都自动更新。

调试器参数选择

在 Tools 菜单下的 Preferences 菜单项中你可以发现调试器相关参数选择部分。这些参数的详细描述见 Oracle-调试器（Oracle - Debugger）章节。

4.11 跟踪执行（Tracing execution）

如果你使用的是 Oracle8i 或者更高版本，你可以使用测试窗口的跟踪工具（Trace facility）来跟踪你的 PL/SQL 代码的执行。你可以设置你想跟踪的事件：调用、异常、SQL、甚至是 PL/SQL 代码行。点击创建跟踪报表（Create Trace report）键旁边的选择跟踪级别键（Select Trace Levels）出现下面的配置界面：



ALL 跟踪全部程序单元，Enabled 跟踪带调试信息编译的程序单元，None 不跟踪。

为了创建跟踪报表，点击测试窗口工具条中的创建跟踪报表（Create Trace report）键后执行你的测试脚本。执行后，你可以切换到跟踪标签页查看该报表和查看以前的测试跟踪报表：

Comment	Event time	Unit name	Unit line	Unit line text	Exception
PL/SQL Virtual Machine started	19-10-2001 19:29:41				
Procedure Call	19-10-2001 19:29:41				
New line executed	19-10-2001 19:29:41	EMPLOYEE.DEPTNAME			
New line executed	19-10-2001 19:29:41	EMPLOYEE.DEPTNAME			
New line executed	19-10-2001 19:29:41	EMPLOYEE.DEPTNAME	9	if p_empno is null	
New line executed	19-10-2001 19:29:41	EMPLOYEE.DEPTNAME	15	select dept dname into result	
SELECT DEPT.DNAME FROM DEPT	19-10-2001 19:29:41	EMPLOYEE.DEPTNAME	15	select dept dname into result	
Exception raised	19-10-2001 19:29:41	EMPLOYEE.DEPTNAME	15	select dept dname into result	1403
Exception handled	19-10-2001 19:29:41	EMPLOYEE.DEPTNAME	27	when no_data_found then return(null);	1403
New line executed	19-10-2001 19:29:41	EMPLOYEE.DEPTNAME	27	when no_data_found then return(null);	
New line executed	19-10-2001 19:29:41	EMPLOYEE.DEPTNAME	28	end;	
Return from procedure call	19-10-2001 19:29:41	EMPLOYEE.DEPTNAME	28	end;	
PL/SQL Virtual Machine started	19-10-2001 19:29:41				

1:1 Executed in 0,01 seconds

在这个（简单）报表中你可以看到一个 select 语句在 employee.deptname 函数的第 15 行出现异常 1403（数据未查到），被本程序单元的第 27 行捕获。

对于每个跟踪事件，你可以在报表中包含像事件描述、时间、程序单元、行号、异常等等的信息。要设置这些信息点击跟踪工具条左上侧的参数设置（Preferences）键，弹出参数设置窗口，有关参数设置窗口的详细信息见 Oracle-跟踪（Oracle - Trace）章节。

要看过去的跟踪报表，从 Run 的选择列表中选择。按删除运行（Delete Run）键删除当前选择的运行。

Oracle 更多的跟踪工具信息参考 Oracle Supplied Packages Reference 手册中的 DBMS_TRACE。

4.12 回归测试（Regression testing）

为了用测试脚本进行回归测试，你可以使用测试管理工具（见测试管理）在测试脚本上定义和运行测试集（Test Sets）。

5 优化

在你的程序单元中优化 SQL 语句，Oracle 的计划解释工具（Explain Plan utility）通过显示语句的执行计划提供极大的帮助。所以，它被集成到 PL/SQL Developer's IDE 中。

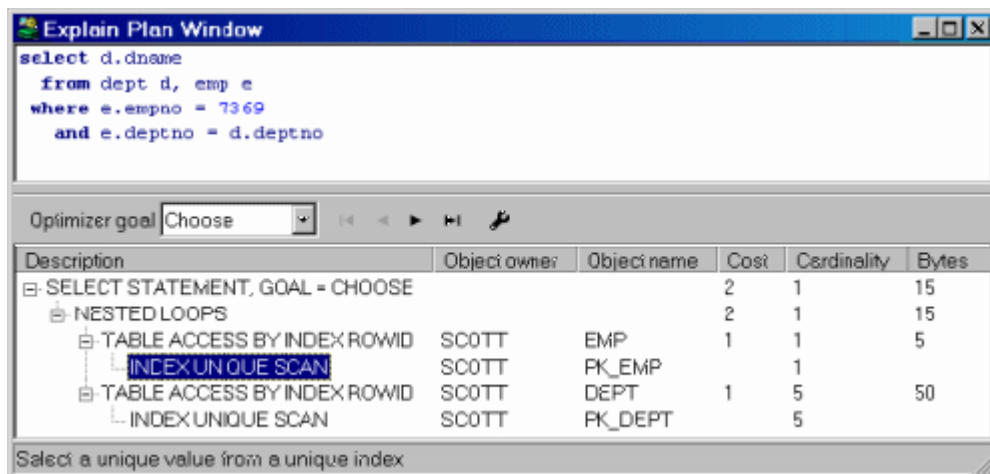
要查看 SQL 语句或者 PL/SQL 程序单元实际使用的资源，PL/SQL Developer 可以显示执行的统计信息。你可以设置哪些统计希望显示，包括执行时间，CUP 时间，逻辑读，物理读，物理写等等。

要判断每个 PL/SQL 代码行的执行时间，你可以使用 PL/SQL 优化跟踪器（PL/SQL Profiler），在 Oracle 8.0 或者更早版本不能使用。

最后，通过激活 SQL 跟踪，你可以使用 Oracle 的 tkprof 工具得到程序单元所有执行过的 SQL 语句使用的资源信息。

5.1 使用解释计划窗口（Using the Explain Plan Window）

要使用 Oracle 的解释计划窗口，点击工具条中的 New 键，选择 Explain Plan Window 菜单项，出现一个空的解释计划窗口。窗口的上半部分可以输入你想分析的 SQL 语句，按工具条中的 Execute 键后，执行计划被显示在窗口的下半部分：



你可以修改 SQL 语句后再次按 Execute 键来查看变化后的情况。更多有关执行计划的信息可以阅读 Oracle 的 Server Tuning 手册。

要查看不同优化器目标的效果，可以在 Optimizer goal 列表中选择相应的条目，执行计划立即按选择的优化器目标更新。

使用第一个、上一个、下一个和最后一个操作按键在执行计划中按顺序导航。在执行计划被确定后，第一个操作被高亮显示。

解释计划工具使用所谓的计划表存储执行计划，如果当前用户不能使用这张表，PL/SQL Developer 将询问是否在当前用户下创建这张表。要定义查看计划表中的哪些字段，查看顺序，按参数选项（Preferences）键，弹出相应的参数选项页（见窗口类型-计划窗口）。

注意：如果你使用 Oracle Server 7.2 或者更低版本，成本和集容量字段在计划表中无效。如果你使用的是 Oracle Server 7.3 或者更高版本，成本和集容量字段不显示，你或许需要升级计划表。

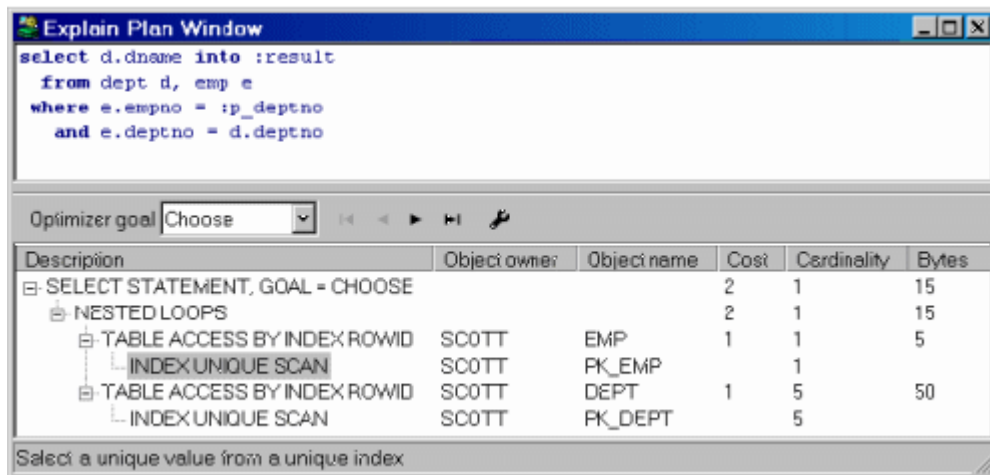
如果你右键点击对象名称（Object name）列，就显示选择对象的弹出菜单。

5.2 在程序单元中分析 SQL (Analyzing SQL in a program unit)

时常，你要分析的 SQL 语句都包含在程序单元中，要做这个，你可以在程序编辑器中选择 SQL 语句后选择 Tools 菜单中的 Explain Plan（或者按 F5），程序为选择的 SQL 语句

打开一个新的解释计划窗口，所有 PL/SQL 变量替换成绑定变量。

如果你是从程序单元中拷贝和粘贴带有 PL/SQL 变量的 SQL 语句到解释计划窗口执行，会收到一个“ORA-00904: Invalid column name”错误，解释计划工具错误地将这些变量解释为字段。每一个变量必须使用冒号作为前缀。如果你想分析 employee.deptname 中的 SQL 语句，result 和 p_empno 变量必须按下面方法替换：



现在，解释计划工具知道哪些代表字段，哪些代表变量了。

5.3 自动统计 (Automatic statistics)

当你在 SQL 窗口和测试窗口执行一条语句时，PL/SQL Developer 将自动为该执行产生统计报表，一个前提条件是你必须拥有动态性能表 v\$sqlsession, v\$sqlstatname 和 v\$sqlsesstat 的 select 权限（通过标准的 Plustrace 角色提供）。

可以通过 SQL 窗口或者测试窗口上方的统计标签页查看。执行 employee.deptname 函数后的统计表看起来如下：

Name	Last	Total
CPU used by this session	0	0
physical reads	5	12
physical writes	0	0
session logical reads	17	45
sorts (disk)	0	0
sorts (memory)	2	2
sorts (rows)	5	5
table fetch by rowid	1	6
table scan blocks gotten	3	5
table scan rows gotten	19	21
table scans (long tables)	0	0
table scans (short tables)	3	5

对于每一个统计指标都能看到最近一次执行值 (Last) 和当前会话值 (Total)。通过参数可以设置显示哪些统计数据，详解 Oracle-选项 (Oracle-Options)。默认的统计设置

和你的 SQL 以及 PL/SQL 有关，具体如下：

统计	含义
CPU used by this session	CPU 的使用，以百分之一秒为单位
Physical reads	从磁盘中读的数据块数
Physical writes	写磁盘的数据块数
session logical reads	从块缓存或者磁盘读的数据块数
sorts (disk)	在临时段执行排序数
sorts (memory)	在内存执行排序数
sorts (rows)	排序记录数
table fetch by rowid	使用 rowid 存取记录数，通常是索引存取的结果集
table scan blocks gotten	为全表扫描读的数据块数
table scan rows gotten	为全表扫描读的记录数
table scans (long tables)	为全表扫描读的大表数
table scans (short tables)	为全表扫描读的小表数

其他统计指标依赖于 Oracle 数据库版本，不在本手册中描述。如果你希望获得更多这方面的知识，有很多 Oracle 调优的书中都涉及这个话题。Oracle 服务参考 (Oracle Server Reference) 也简单地讲述了这些统计项。

统计表可以导出到 CSV 文件中，可以用表格应该程序打开。在统计表上右键点击，选择 Export 菜单项并且选择 CSV 文件项。你也可以选择 TSV, XML 或 HTML 格式，或者拷贝到剪贴板中。

5.4 The PL/SQL Profiler

The PL/SQL Profiler 是一个非常强大的工具，帮助你优化 PL/SQL 代码。对于每一个已执行的代码行，合计时间、最大时间、最小时间、平均时间和发生次数都将被确定。

The Profiler 在测试窗口中很容易得到。执行测试脚本前，只需要按测试窗口工具条中的 Create Profiler report 键，如果随后执行脚本，你可以切换到 Profiler 页查看报表。

下面的样例报表显示了 employee.deptname 函数查询语句用了 149 毫秒，调用 3 次 dbms_output 用了 57 毫秒：

Unit	Line	Total time	Occurrences	Text
ANONYMOUS BLOCK	4	27	2	2 :result := employee.deptname(p_empno => :p_empno);
EMPLOYEE	9	1	1	1 if p_empno is null
EMPLOYEE	12	0	0	0 result := null;
EMPLOYEE	15	149	1	1 select deptname into result
EMPLOYEE	20	57	3	3 dbms_output.put_line(to_char(sysdate, 'dd-mm-yyyy hh24:mi:ss'))
EMPLOYEE	24	5	1	1 return(result);
EMPLOYEE	27	0	0	0 when no_data_found then return(null);
EMPLOYEE	28	4	1	1 end;
EMPLOYEE	34	0	0	0 if p_empno is null then
EMPLOYEE	36	0	0	0 result := null;
EMPLOYEE	39	0	0	0 select m.ename into result
EMPLOYEE	44	0	0	0 dbms_output.put_line(to_char(sysdate, 'dd-mm-yyyy hh24:mi:ss'))
EMPLOYEE	48	0	0	0 return(result);
EMPLOYEE	50	0	0	0 when no_data_found then return(null);
EMPLOYEE	51	0	0	0 end;

1:1 Executed in 0 seconds

默认情况下，Profiler 页将显示最后一次运行报表，你可以从 Run 列表中选择之前的运行。Unit 列表允许你放大（zoom in）一个特定程序单元运行。

合计时间列用图形方式显示代码行的相对时间，对比的是最长时间。这可以让你快速找到最耗费资源的代码行。报表通过点击列表头排序。

如果一个代码行已红色显示，表示程序单元在创建报表后已被改变。因此，显示的代码行现在可能和创建报表时的不一致。

改变不同的 Profiler 样式，你可以按 Preferences 键，弹出相应的参数页面，详见 Oracle- Profiler 章节。

更多的有关 the PL/SQL Profiler 信息见“Oracle8i Supplied Packages Reference”手册中的 dbms_profiler。

注意：不是所有的平台都提供相同的精确时间信息。

5.5 SQL 跟踪（SQL Trace）

你可以按 SQL 跟踪（SQL Trace）键允许 SQL 跟踪。此后所有由 SQL 窗口或者测试窗口引起的服务进程都将被记录在数据库服务器中的跟踪文件中。有一个前提条件是你使用的数据库实例的定时统计参数必须为真（True）。

你可以再次点击 SQL 跟踪键禁止 SQL 跟踪。

你可以使用 Oracle 数据库服务器中的 tkprof 工具查看跟踪文件信息。它将生成一个关于 SQL 语句的逝去时间、CPU 时间、I/O 等等的报表。这样，你可以知道程序单元中的哪个语句最消耗资源。要学习更多有关 Oracle 的 tkprof 工具知识可以阅读“Oracle 7 Server Tuning”手册。

执行 employee.deptname 函数的 tkprof 报表看起来如下：


```

TKPROF: Release 7.2.2.3.1 - Production on Fri Sep 26 14:59:08 1997
Copyright (c) Oracle Corporation 1979, 1994. All rights reserved.
Trace file: ora07087.trc
Sort options: default
*****
count      = number of times OCI procedure was executed
cpu        = cpu time in seconds executing
elapsed    = elapsed time in seconds executing
disk       = number of physical reads of buffers from disk
query      = number of buffers gotten for consistent read
current    = number of buffers gotten in current mode (usually for update)
rows       = number of rows processed by the fetch or execute call
*****
begin
  :deptname := employee.deptname(:empno);
end;
call      count      cpu      elapsed disk      query  current    rows
-----
Parse    1          0.00    0.00    0         0        0         0
Execute  1          0.01    0.01    0         0        0         1
Fetch    0          0.00    0.00    0         0        0         0
-----
Total    2          0.01    0.01    0         0        0         1
Misses in library cache during parse: 0
Optimizer hint: CHOOSE
Parsing user id: 16
*****
SELECT DEPT.DNAME
FROM
  DEPT,EMP WHERE EMP.EMPNO = :b1 AND DEPT.DEPTNO = EMP.DEPTNO

Call      count      cpu      elapsed disk      query  current    rows
-----
Parse    1          0.00    0.00    0         0        0         0
Execute  1          0.00    0.00    0         0        0         5
Fetch    1          0.06    0.06    4         4        0         1
-----
Total    3          0.06    0.06    4         4        0         6
Misses in library cache during parse: 0
Optimizer hint: CHOOSE
Parsing user id: 16      (recursive depth: 1)
*****

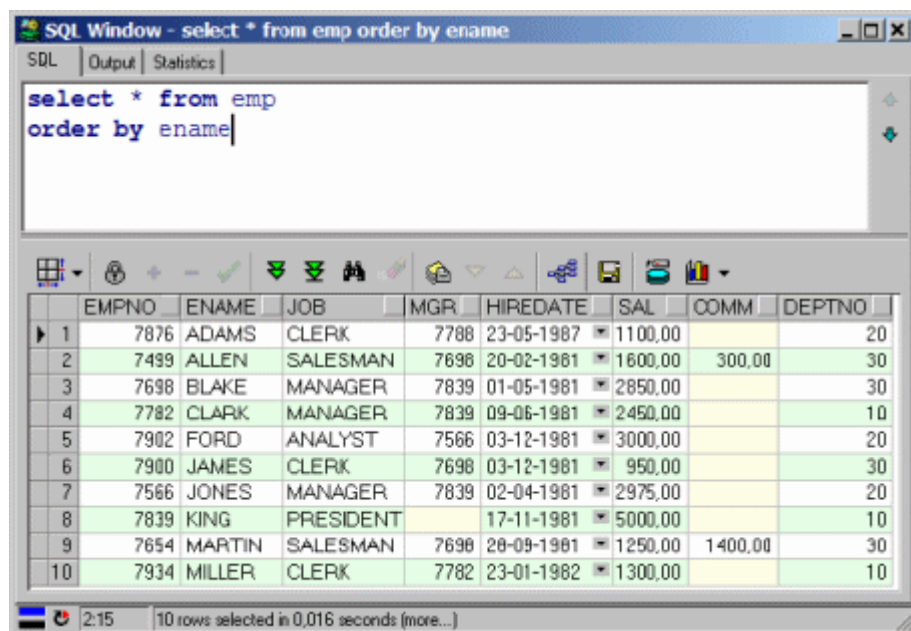
```

6 特殊的SQL(Ad hoc SQL)

经常在程序开发过程中，你需要执行一些 SQL 语句，不管是测试 SQL 语句，查看表数据还是更新数据，你可以在 PL/SQL Developer 使用 SQL 窗口。

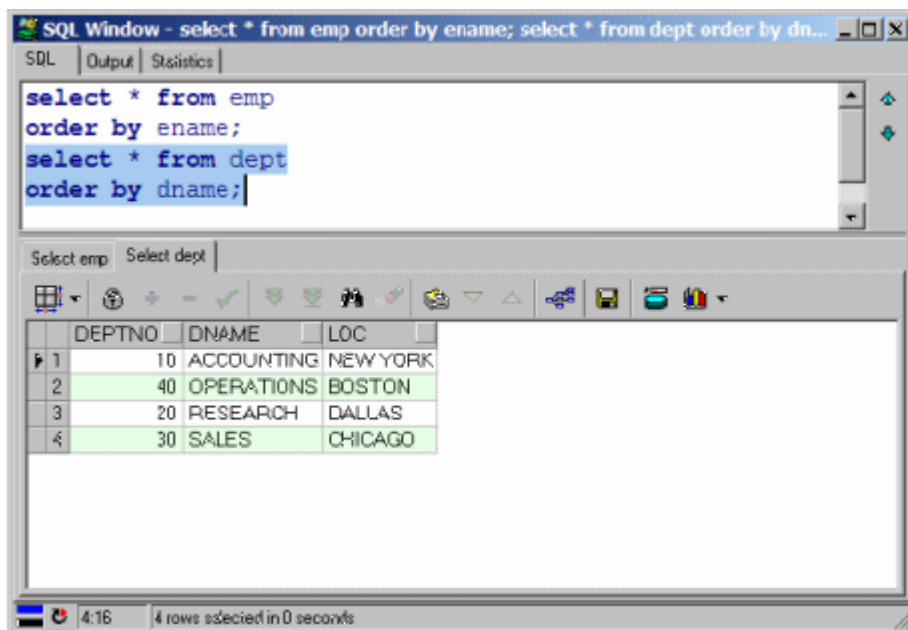
6.1 使用 SQL 窗口（Using the SQL Window）

打开一个新的 SQL 窗口，按工具条中的 New 键，选择 SQL 窗口（SQL Window），出现一个空的 SQL 窗口，你可以输入 SQL 语句。如果是查询语句，窗口被分为两部分，查询结果以表格形式在下半部分显示。如果我们想从 employee 表中查询所有数据，SQL 窗口应该像这样：



查询语句被执行并且存取了 10 行数据。下一页和最后页键亮说明结果集大于 10 行。初始存取 10 行是因为当前的表格只能显示 10 行数据。按下一页键将存取下一个 10 行，按最后页键将存取全部记录。

如果在 SQL 编辑区中有多个 SQL 语句，按执行（Execute）键将会执行所有语句，结果集分页显示：



SQL 语句必须以分号或者反斜线结束，PL/SQL 块就像在 SQL*Plus 中一样用斜线结束。如果你选择结果集页，相应的文字在 SQL 编辑区中被高亮显示。

如果你在编辑区中选择了部分文字，只有选中的文字被执行。这样就可以在编辑区中有多个语句，仍然可以一句一句地执行。

如果你执行的 SQL 语句需要很长时间才能结束，你可以按 Break 键放弃执行。注意，按 Break 键不能保证总会成功。例如，如果语句等待锁定就不响应 break 信号。当 SQL 窗口在提取数据时按 Break 键，它只简单停止并显示已提取的数据。你可以使用下一页和最后一页键继续提取数据。

注意 SQL Window 允许你通过参数设置确定查询语句初始返回的记录数，详解窗口类型 - SQL 窗口 (Window Types - SQL Window) 章节。

窗口右侧有两个键可以导航你在 SQL Window 中已输入的 SQL 语句。这样提供了一个快速方法可以再执行原已输入的语句。

为了优化的目的，你可以选择 Statistics 标签页查看 SQL 语句执行的统计表。统计表的解释见自动统计 (Automatic statistics) 章节。

当你打印 SQL 窗口，SQL 语句和表格中的结果将被打印，通过选择表格中的行限制打印的数据行数。

6.2 结果集表格处理 (Result grid manipulation)

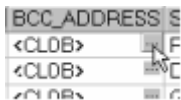
SQL 窗口中的结果集表格可以用不同的方法处理，一些单元格类型有特殊的行为，行、列和单元格的范围可以选择和打印，列可以移动，行可以排序，你可以切换到单记录查看等等。

识别空值 (Recognizing null values)

空值在单元格中以黄色背景显示，所以，你可以快速地和全部为空格值区别开来，这也对区别空值和不在单元格中直接显示的值有所帮助，例如 longs 和 LOB 值。你可以通过参数控制改变空值的背景（见窗口类型-SQL 窗口）。

查看大数据字段 (Viewing large data columns)

Long, Long Raw, CLOB, BLOB 和 BFILE 字段的值不在结果集中显示，代以显示<Long>, <Long Raw>, <CLOB>, <BLOB> 和 <BFILE>:

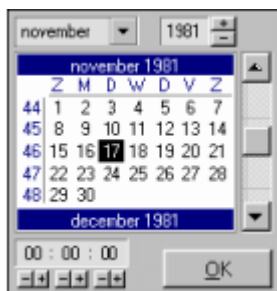


当你点击这个字段单元格的按钮时，调用大数据编辑器，允许你用不同的格式查看和编辑数据，详见大数据编辑器 (The Large Data Editor)。

你也可以点击字符型字段单元格的按钮调用大数据编辑器。单元格按钮只有在字段长度大于 20 个字符时才出现。

查看日期字段 (Viewing date columns)

一个日期字段有一个单元格按钮显示日历，当前日期高亮显示。如果日期带有时间，你在日历上同样可以看到时间值：



查看时间戳字段 (Viewing timestamp columns)

时间戳字段按照注册表中的 Oracle Home 下的 NLS_TIMESTAMP_FORMAT 和 NLS_TIMESTAMP_TZ_FORMAT 设置的显示格式显示。

查看 XML 数据 (Viewing XML data)

有很多方法把 XML 保存到数据库中：像 CLOB (Oracle8i), XMLTYPE (Oracle9i)，甚至 Varchar2 和 Long 字段。在任何情况下，如果以标准的 XML 头开始，大数据编辑器认定是 XML 格式，并将它转化为 XML 格式。详见大数据编辑器 (The Large Data Editor)。

查看嵌套游标字段 (Viewing nested cursor columns)

如果你在 select 语句的字段列表中包含了嵌套游标，字段值将初始显示为<Cursor>。点击单元格按钮，出现一个新的带有游标结果集的 SQL 窗口，这可以用作查看简单的嵌套查询，注意每个嵌套游标都暗含一个打开游标结果集，所以，对于大结果集很容易超过

OPEN_CURSORS 限制并得到“ORA-01000: maximum open cursors exceeded”错误。

选择多列、多行和多单元格 (Selecting columns, rows and cells)

在结果表格中选择多行或多列，只要点击行标题头或者列标题头拖动鼠标高亮选择：

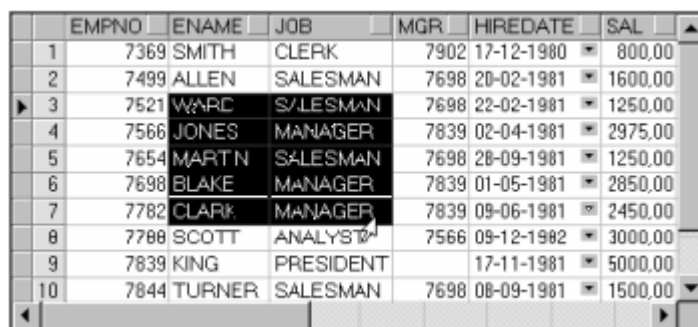


	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL
1	7369	SMITH	CLERK	7902	17-12-1980	800,00
2	7499	ALLEN	SALESMAN	7698	20-02-1981	1600,00
3	7521	WARD	SALESMAN	7698	22-02-1981	1250,00
4	7566	JONES	MANAGER	7839	02-04-1981	2975,00
5	7654	MARTIN	SALESMAN	7698	23-09-1981	1250,00
6	7698	BLAKE	MANAGER	7839	01-05-1981	2850,00
7	7782	CLARK	MANAGER	7839	09-06-1981	2450,00
8	7788	SCOTT	ANALYST	7566	09-12-1982	3000,00
9	7839	KING	PRESIDENT		17-11-1981	5000,00
10	7844	TURNER	SALESMAN	7698	08-09-1981	1500,00

	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL
1	7369	SMITH	CLERK	7902	17-12-1980	800,00
2	7499	ALLEN	SALESMAN	7698	20-02-1981	1600,00
3	7521	WARD	SALESMAN	7698	22-02-1981	1250,00
4	7566	JONES	MANAGER	7839	02-04-1981	2975,00
5	7654	MARTIN	SALESMAN	7698	23-09-1981	1250,00
6	7698	BLAKE	MANAGER	7839	01-05-1981	2850,00
7	7782	CLARK	MANAGER	7839	09-06-1981	2450,00
8	7788	SCOTT	ANALYST	7566	09-12-1982	3000,00
9	7839	KING	PRESIDENT		17-11-1981	5000,00
10	7844	TURNER	SALESMAN	7698	08-09-1981	1500,00

被选择列现在可以通过放开鼠标键，再次点击被选择列中的一个列标题头，拖动到新位置的方法移动。

为选择一个特殊区域的单元格，移动鼠标指针越过单元格左侧直到光标形状改变，按鼠标键拖动鼠标高亮选择：



	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL
1	7369	SMITH	CLERK	7902	17-12-1980	800,00
2	7499	ALLEN	SALESMAN	7698	20-02-1981	1600,00
3	7521	WARD	SALESMAN	7698	22-02-1981	1250,00
4	7566	JONES	MANAGER	7839	02-04-1981	2975,00
5	7654	MARTIN	SALESMAN	7698	23-09-1981	1250,00
6	7698	BLAKE	MANAGER	7839	01-05-1981	2850,00
7	7782	CLARK	MANAGER	7839	09-06-1981	2450,00
8	7788	SCOTT	ANALYST	7566	09-12-1982	3000,00
9	7839	KING	PRESIDENT		17-11-1981	5000,00
10	7844	TURNER	SALESMAN	7698	08-09-1981	1500,00

为选择所有的行和列，按左上角的单元格，或者右键点击表格，从弹出菜单中选择选择全部 (Select All) 菜单项。

通常被高亮选择的可以拷贝和打印。

导出数据 (Exporting data)

有很多方法导出结果集表格中的数据，执行一个 select 语句后，你可以像上面介绍的选择一个单元格区域，在上右键点击，从弹出菜单中选择导出结果集 (Export Results) 菜单项。这将显示一个子菜单，你可以选择导出的数据格式：CSV 格式 (Comma Separated

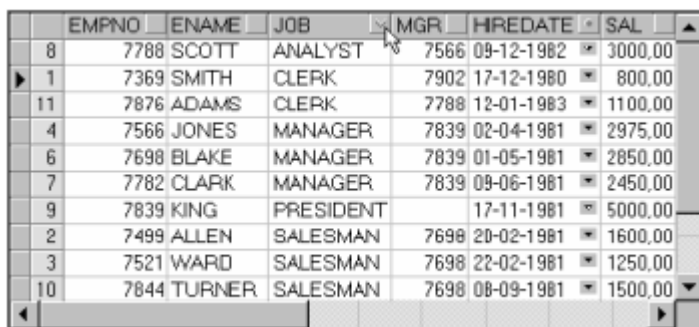
Values), TSV 格式(Tab Separated Values), HTML 格式, 或者 XML 格式, 选择格式后, 你可以指定导出文件。

另外一种方法是将内容拷贝到剪贴板中, 按 Ctrl_C 或者右键点击选择项后从弹出菜单中选择拷贝 (Copy) 或者带标题头拷贝 (Copy with Header) 菜单项。你可以随后粘贴数据到其他程序中, 像电子制表软件, 文字处理软件等等。

为了快速处理结果集到 Microsoft Excel, 选择 Copy to Excel 菜单项, 这将打开一个新的 Excel 实例, 而且所有选择数据被拷贝。

排序行 (Sorting rows)

为在结果集表格中排序行, 点击排序字段的标题头按键:



	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL
8	7788	SCOTT	ANALYST	7566	09-12-1982	3000,00
1	7369	SMITH	CLERK	7902	17-12-1980	800,00
11	7876	ADAMS	CLERK	7788	12-01-1983	1100,00
4	7566	JONES	MANAGER	7839	02-04-1981	2975,00
6	7698	BLAKE	MANAGER	7839	01-05-1981	2850,00
7	7782	CLARK	MANAGER	7839	09-06-1981	2450,00
9	7839	KING	PRESIDENT		17-11-1981	5000,00
2	7499	ALLEN	SALESMAN	7698	20-02-1981	1600,00
3	7521	WARD	SALESMAN	7698	22-02-1981	1250,00
10	7844	TURNER	SALESMAN	7698	08-09-1981	1500,00

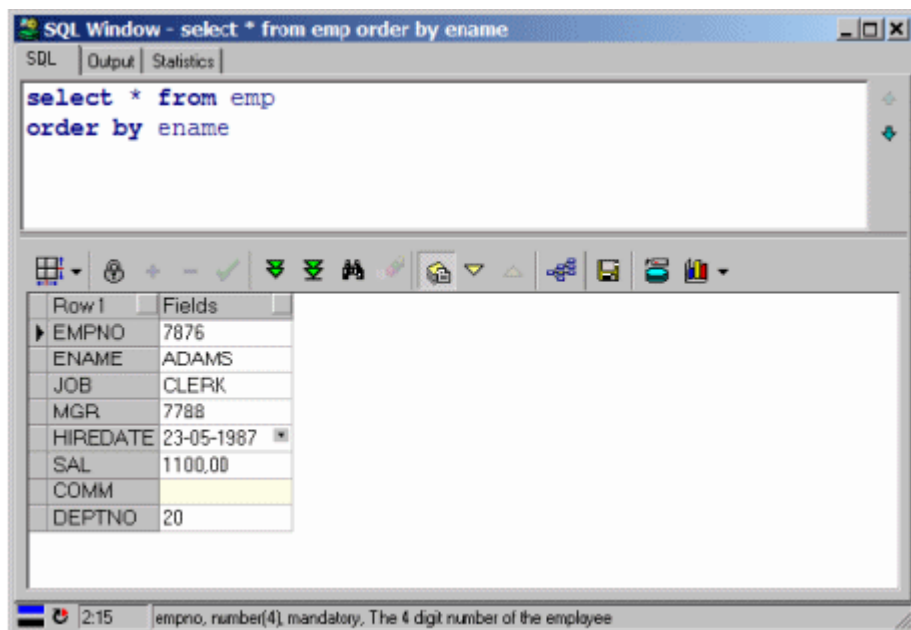
所有行将按升序排序, 像标题头按键显示的一样。再次按标题头按键将按降序排序, 点击第三次将取消排序。点击另外一个字段标题头按键将按这个字段排序, 但是, 原排序字段作为第二个排序字段 (标题头按键显示为标点)。在上面的例如中, job 字段是主排序字段, hiredate 字段是第二个排序字段。

注意排序在本地执行, 只对已经读取的行。如果排序后再读取数据, 这些新数据行将增加到结果集表格的最后, 没有任何排序。对于大结果集, 本地排序耗费大量时间。在这种情况下, 最好在 select 查询中使用 order by 语句, 让 Oracle 服务器执行排序。

单记录视图 (Single Record View)

如果结果集包含很多字段, 一条记录显示一行可能不方便, 你必须前后滚动查看关联字段 (虽然你可以移动字段), 并且不能一次看到所有字段。

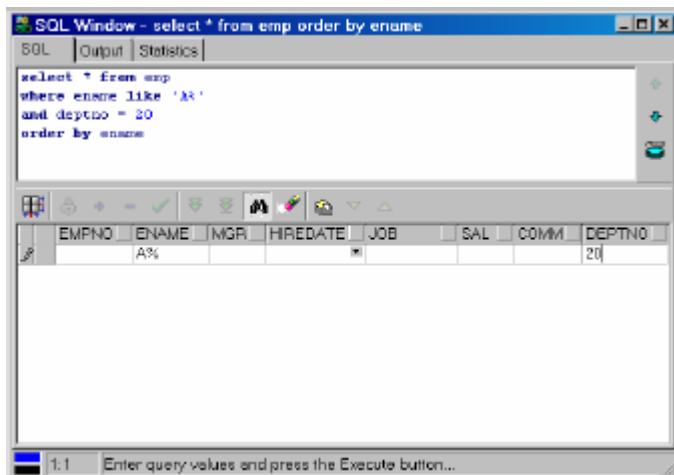
按表格工具条中的单记录视图 (Single Record View) 可以一次看一条记录:



现在每行显示一个字段名和字段值。表格工具条中的下一条记录和上一条记录可以导航结果集。再次按单记录视图 (Single Record View) 可以回到多记录视图。

6.3 实例模式查询 (Query By Example mode)

执行完一个 select 语句后,如果你想查找一个特殊记录,可以按结果集上方的 Query By Example 键。按这个键将清空表格, 留一个空记录让你输入查询值。例如, 你想找到姓名以字母 A 开始的, 在部门 20 的全部职工, 在 ename 字段中输入 A% 并且在 deptno 字段输入 20:



再次按 Query By Example 键或者按执行 (Execute) 键将执行被修改的查询并且显示受限的结果集。再次按 Query By Example 键继续前面的查询值, 按清除记录 (Clear record) 键清除查询值。

查询值并没有限制使用匹配符为单一值。你可以用下列表达式 (表达式用方括号括起):

- value [= value]

- value with wildcards [like value with wildcards]
- value
- < value
- != value [<> value]
- in (value1, value2, ...)
- between value1 and value2
- null [is null]
- not null [is not null]

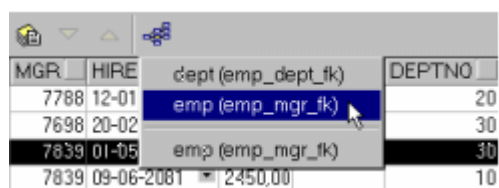
注意如果你使用文字值，可以省略引号（例如：SMITH 将自动转换为 SQL 文本中的'SMITH'）。对于其他所有的表达式，你必须使用 Oracle 服务器识别的文字值（例如：!= 'SMITH'），对于数值和日期值也是同样。如果你在一个表达式中使用，必须是 Oracle 服务器识别的格式。

存在许多参数允许你控制 Query By Example 功能的默认行为，详见窗口类型-SQL 窗口章节。

6.4 关联查询 (Linked Queries)

当查看一个结果集时，你经常想查询管理表，例如，当你查看部门表时，你可能想查询一个部门中所有的职工，或者相反：当查看职工表时，你可能想查询一个职工所在部门。

大部分情况下，主从表之间建立外键，在这种情况下，SQL 窗口可以自动为你生成和执行这些查询。如果你在结果集中定位在一个特殊的记录，按表格工具条中的关联查询(Linked Query) 键将显示一个弹出菜单，包括所有当前 select 语句表的主表和从表。例如，如果你查询 emp 表，应该出现下面的弹出菜单：

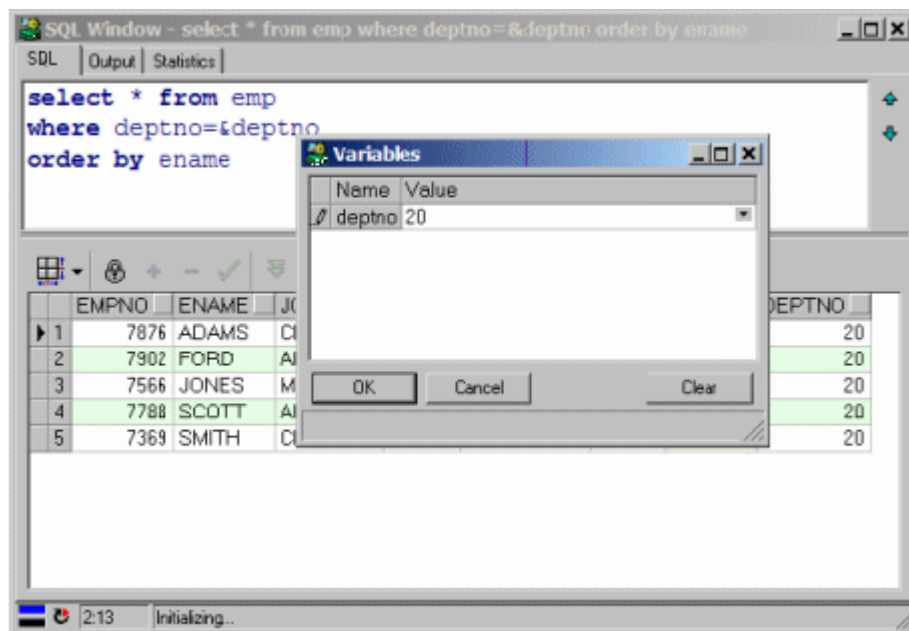


上部分显示主表（和外键名），下部分显示子表。dept 菜单项将生成一个该职工部门（30）的查询。第一个（主表）emp 菜单项将生成该职工主管(7839)的查询。第二个（从表）emp 菜单项将生成该职工管理的全部职工查询。

关联查询将在同一个 SQL 窗口中执行，除非你允许 SQL 窗口参数设置为管理查询在新窗口（见窗口类型-SQL 窗口），如果在按管理查询（Linked Query）时按 Ctrl 键，该参数将反向执行。

6.5 置换变量 (Substitution variables)

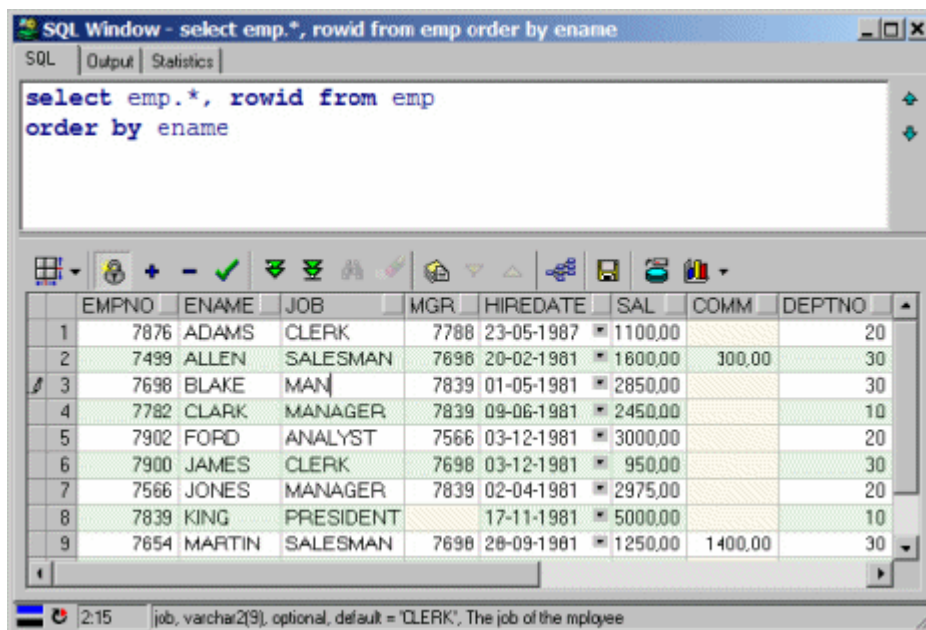
在你的 SQL 中可以使用置换变量，在执行时输入具体值。格式非常像你在 SQL*PLUS 中的格式：



对于 SQL 中的 &deptno，你将被提示输入该变量值。变量将在执行前被替换。你可以使用多个置换变量，同时，一个置换变量可以在多处使用。而且，你可以定义数据类型、默认值、选择列表、列表查询 (list queries)、复选框和其他高级选择。这些特性也可以使用在报表窗口。详见报表 (Reports)。

6.6 更新数据库 (Updating the database)

为更新、插入或者删除数据库中的记录，你可以在 SQL 窗口中执行相应的 DML 语句。在查询字段列表中包含 rowid 或者使用 select ... for update 语句的方法可以使结果集表格处于可修改状态：



要小心一个 `select ... for update` 语句将锁定所有查询记录，所以，为使结果集可编辑的最好办法是使用 `rowid` 查询。如果是关联查询，第一个表的字段可以修改，其他字段只读。使用别名字段也不能修改。

如果结果集是可修改的，你可以按结果集左侧的编辑数据（Edit data）按钮编辑记录。插入记录或者删除记录可以按插入记录（Insert record）或者删除记录（Delete record）按钮。如果你选择了多条记录，所有选择的记录将被删除。

修改结果集表格并没有真实地修改数据库，提交修改、插入或者删除记录到数据库按提交（Post changes）键，如果已禁止 SQL 窗口自动提交事物（AutoCommit SQL Window）选项，如果开始了一个事物，工具条中的提交和回滚键将被激活，更多有关事物信息见窗口、数据库会话和事物处理（Windows, database sessions and transactions）章节。

编辑大数据字段（Editing large data columns）

根据前章说明，你可以点击 Long, Long Raw, CLOB, BLOB 和 BFILE 字段的单元格键用不同格式查看数据值。当结果集表格是可修改的，你可以使用大数据编辑器修改字段值。

6.7 查看和编辑 XMLTYPE 字段（Viewing and editing XMLTYPE columns）

SYS.XMLTYPE 类型不支持 Oracle Net 9.0 或更低版本，所以，你不能直接存取该字段的 XML 数据。例如，如果 `xml_text` 是 SYS.XMLTYPE 字段，下面的查询将不能得到预期结果：

```
select id, xml_text from xml_table
```

只有 `id` 字段在结果集中显示，查看 XML 数据，使用 `getclobval()` 成员函数：

```
select id, t.xml_text.getclobval() from xml_table t
```

使用这种方法，CLOB 可以在文本编辑器中查看，并且将自动对 XML 语法高亮显示。要

编辑 XML 数据，将结果集设置成可编辑，通常方法是包含 rowid:

```
select id, t.xml_text.getclobval(), rowid from xml_table t
```

现在你可以编辑 CLOB 数据并将修改提交数据库。需要注意的是对于表和视图的查看数据 (View Data) 和编辑数据 (Edit Data) 函数将自动应用这些规则，所以，最简单的查看和编辑 XMLTYPE 字段方法是在表或者视图上右键点击，从弹出的菜单中选择相应的菜单项。

6.8 直接查询导出 (Direct Query Export)

对于大的结果集查询 (一万或更多)，先查询数据到结果集表格中，再导出到文件很不方便，这将耗费很长时间和大量内存资源。在这种情况下，更有效的方法是直接将结果集写入导出文件。要做到这点，你可以点击结果集工具条中的导出查询结果 (Export Query Results...) 键替代主工具条中的执行 (Execute) 键。这将显示一个弹出菜单，你可以选择导出数据到 CSV 格式 (Comma Separated Values), TSV 格式 (Tab Separated Values), HTML 格式，或者 XML 格式。之后确定导出的文件名，确认后查询将执行，查询的结果集不被显示，而是直接写到导出文件中。

6.9 保存 SQL 脚本 (Saving SQL Scripts)

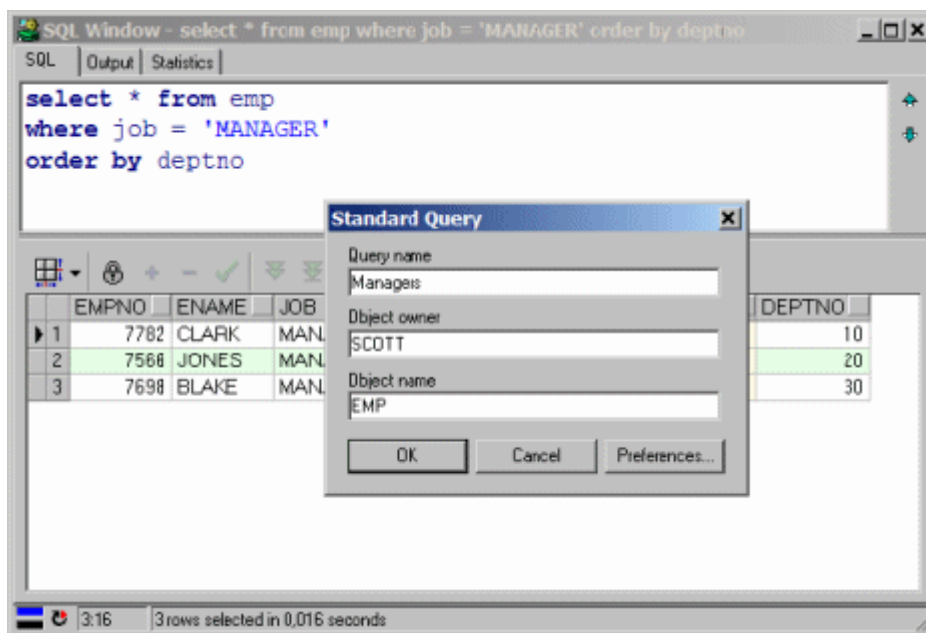
如果你创建了 SQL 语句还想后期再执行，你可以将它保存成 SQL 脚本。在工具条中按保存 (Save) 键，在对话框中输入相应的文件名，SQL 脚本的默认扩展名是 .sql，你可以通过参数修改默认扩展名，详见文件-扩展名 (Files - Extensions) 章节。需要注意的是只有当前的 SQL 语句被保存到文件中，不是所有已输入的语句。要想保存历史语句，在 SQL 窗口中右键点击，选择保存历史菜单 (Save with history) 项，所有语句将已斜线分割。

你可以按工具条中的打开 (Open) 键，选择 SQL 脚本文件后，在新创建的 SQL 窗口中打开原保存的 SQL 脚本。你也可以在已创建的 SQL 窗口中点击右键，选择导入 (Load) 菜单项，在当前的 SQL 窗口中打开 SQL 脚本。

6.10 创建标准查询 (Creating Standard Queries)

对于大多数表和视图，你将有不同的标准查询需要经常运行。例如：“显示所有管理人员”，或者“显示所有职工的工资总额，按部门代码排序”。你可以将这些查询保存成所谓的标准查询“Standard Query”，可以很容易地在这些表或视图的弹出菜单中调用。这个弹出菜单在对象浏览器 (Object Browser)、PL/SQL 或者 SQL 代码中右键点击表或者视图时出现。

当在 SQL 窗口中创建了标准查询后，在结果集表格工具条中按保存为标准查询 (Save as Standard Query) 键，出现下面的对话框：



你可以输入查询名称，这个名称将显示在弹出菜单上。你也可以修改标准查询显示时的宿主和表或者视图的名称。按参数选择（Preferences）键修改标准查询保存目录。

下列信息将保存到标准查询中：

- The SQL text
- The size of the SQL Window
- The grid mode: multi record or single record view

需要注意的是你可以使用置换变量（见置换变量章节）增加标准查询的灵活性。

7 命令窗口(Command Window)

命令窗口可以让你以一种很像 Oracle 的 SQL*Plus 方式执行 SQL 脚本。按工具条中的新建（New）键或者从文件（File）菜单中选择新建（New）菜单项创建一个命令窗口，不用离开 PL/SQL Developer 的集成开发环境就可以像你习惯的那样输入 SQL 和 SQL*Plus 命令：

```

Command Window - New
Dialog Editor
SQL> select * from emp
2  where deptno = 20
3  order by ename;

EMPNO  ENAME      JOB          MGR HIREDATE          SAL         COMM  DEPTNO
-----  -
7876  ADAMS      CLERK        7788 12-01-1983       1100,00          20
7902  FORD       ANALYST      7566 03-12-1981       3000,00          20
7566  JONES      MANAGER      7839 02-04-1981       2975,00          20
7788  SCOTT      ANALYST      7566 09-12-1982       3000,00          20
7369  SMITH      CLERK        7902 17-12-1980        800,00          20

SQL> exec dbms_output.put_line(employee.deptname(7876))
RESEARCH

PL/SQL procedure successfully completed

SQL> desc dep
Object dep does not exist.

SQL> desc dept
Name      Type          Nullable Default Comments
-----
DEPTNO    NUMBER(2)          No      The 2-digit identification number of the department
DNAME     VARCHAR2(14)      Yes     The name of the department
LOC       VARCHAR2(13)      Yes     The location of the department

SQL>

```

ECH TRM FDB VER APR HDG TIM PL/SQL procedure successfully completed in 0 seconds

7.1 输入 SQL 语句和命令（Entering SQL statements and commands）

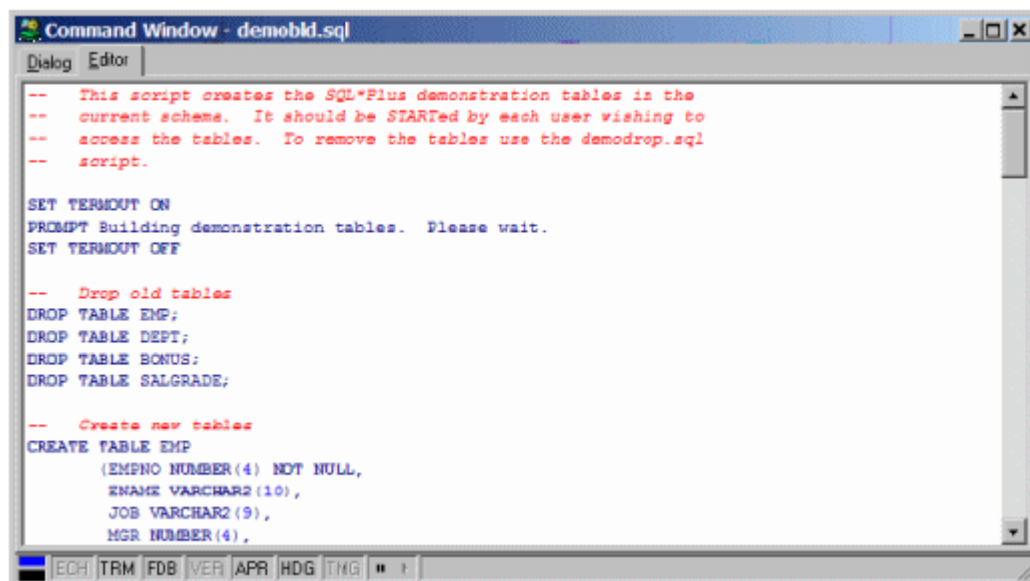
就像在 SQL*Plus 中一样，你可以多行输入 SQL 语句，并以分号或者斜线结束。你可以使用左右光标键编辑命令行，用上下光标键选择原已输入行。

通过输入编辑（edit）命令，你可以使用简单的文本编辑器来编辑整个的输入缓冲区。编辑缓冲区后，你可以在命令行上输入斜线执行。编辑器有一个所有已执行命令的历史缓冲区，以便于你可以快速地执行已修改的命令。

命令窗口的状态行显示响应（Echo），结果显示（Termout），反馈（Feedback），校验（Verify），自动打印（Autoprint），标题（Heading）和耗时（Timing）选项的状态，通过在选项上双击可以改变该选项的开和关。

7.2 开发命令文件（Developing command files）

为开发一个带多个 SQL 语句和命令的命令文件，你经常需要编辑文件，执行，再编辑，再运行等等，为了方便操作，命令窗口有一个内建的编辑器使用 SQL，PL/SQL 和 SQL*PLUS 的高亮语法：



7.3 打开、执行和保存命令文件(Opening, Executing, and Saving a command file)

创建一个命令文件，切换到编辑页（Editor）输入命令。要执行编辑器中的命令，按工具条中的执行（Execute）键或者按 F8。命令窗口将切换到对话（Dialog）页并执行所有命令。

要编辑一个已存在的命令文件，按工具条中的打开（Open）键，选择命令文件（Command File）菜单项，命令文件将在新的命令窗口中载入。你也可以在编辑器中右键点击，在弹出菜单中选择导入（Load）菜单项。保存一个修改后的命令文件，按工具条中的保存（Save）键。

7.4 单步执行一个命令文件（Stepping through a command file）

如果你在执行前或者执行当中按状态行中的暂停执行（Pause Execution）键，你可以通过控制面板单步执行脚本。下一个执行命令将在编辑器中高亮显示，并且在你按状态行中的执行下一条命令（Execute Next Command）键时被执行。如果你再次按暂停执行（Pause Execution）键后，将从上一次执行的命令行后继续执行。你可以在脚本中使用 SET EXECPAUSE 命令强制执行暂停。

7.5 支持的命令（Supported commands）

除了所有的 SQL 语句外，命令窗口在本版本中支持以下标准 SQL*Plus 命令：

命令	含义
/	执行 SQL 缓冲区
? [Keyword]	提供 keyword 的 SQL 帮助
@[@] [Filename] [Parameter list]	执行指定命令文件，传递指定参数

ACC[EPT] Variable [DEF[AULT] Value] [PROMPT Text NOPR[OMPT]]	运行用户输入置换变量
CL[EAR] [SCR[EEN]]	清屏
CL[EAR] SQL	清 SQL 缓冲区
COL[UMN] [Column] [Format] [NEW_ VALUE Variable]	定义一个字段格式，显示一个字段格式或者显示所有字段格式
CON[NECT] [username/password@database]	用指定用户连接数据库
DEF[INE] [Variable] [= Text]	定义一个置换变量，显示一个变量或者显示所有置换变量
DESC[RIBE] Object	给指定对象一个描述
DISC[CONNECT]	断开数据库连接
EDIT	显示一个编辑 SQL 缓冲区的文本编辑器
EXEC[UTE] Procedure	执行指定的存储过程
EXIT [APPLICATION]	退出一个执行的脚本或者关闭命令窗口。加 APPLICATION 参数将关闭 PL/SQL Developer
GET [Filename]	装载一个命令文件到编辑器中
HOST [Command]	执行主机命令
HELP [Keyword]	提供 keyword 的 SQL 帮助
PAUSE [Message]	暂停并显示消息直到用户按确认或取消
PRI[NT] [Variable]	显示绑定变量值或者全部绑定变量值
PROMPT [Text]	显示 text
QUIT [APPLICATION]	退出一个执行的脚本或者关闭命令窗口。加 APPLICATION 参数将关闭 PL/SQL Developer
R[UN]	执行 SQL 缓冲区
REM[ARK] [Text]	注释行
SET AUTOP[RINT] [ON OFF]	设置执行一个 SQL 语句或者 PL/SQL 块后是否自动显示绑定变量
SET COLSEP [Separator OFF]	设置列分隔符（默认=" "）
SET CON[CAT] [Character ON OFF]	设置置换变量的结束符（默认=,）
SET DEF[INE] [Character ON OFF]	设置置换变量起始符（默认=&）
SET ECHO [ON OFF]	设置是否显示脚本中已执行的命令
SET ESC[APE] [Character ON OFF]	设置避免置换变量起始符的字符（默认=\\）
SET FEED[BACK] [ON OFF]	设置是否显示涉及 SQL 语句的行数
SET HEA[DING] [ON OFF]	设置是否在结果集的上方显示字段标题
SET LONG [Width]	设置 long 字段显示的最大宽度
SET NUM[WIDTH] [Width]	设置不带精度的数字字段显示的最大宽度
SET PAGES[IZE] [Size]	设置结果集在显示多少行后字段标题重复显示
SET PROMPT [Prompt]	替换标准的 SQL> 提示。替换文字可以使用变量 [user], [db] 或者 [connection]。而且可以包括绑定变量(set prompt :bind_var_name)
SET SCAN [ON OFF]	设置是否扫描置换变量

SET SERVEROUT[PUT] [ON OFF] [SIZE n]	设置是否显示 dbms_output.put_line 的输出结果和输出缓冲区的大小
SET SPOOL*DIRECTORY [Directory]	设置如果 SPOOL 命令没有指定一个绝对目录时，spool 文件的保存目录
SET TERM[OUT] [ON OFF]	设置是否显示执行的 SQL 语句输出
SET TIMI[NG] [ON OFF]	设置是否显示 SQL 语句的执行时间
SET VER[IFY] [ON OFF]	设置当置换变量在 SQL 语句或者 PL/SQL 块中是否显示
SHO[W] ERR[ORS] [Type Name]	显示上一个编译或者指定对象的错误
SHO[W] REL[EASE]	显示当前连接的 Oracle 版本
SHO[W] SQLCODE	显示已执行 SQL 语句的结果代码
SHO[W] USER	显示当前连接的用户名
SPO[OL] [Filename OFF]	开始或结束假脱机
STA[RT] [Filename] [Parameter list]	运行指定的命令文件，传递指定参数
STORE SET [Filename]	将全部选项值保存到文件，以后可以执行这个文件导入这些选项
UNDEF[INE] Variable	不定义指定的置换变量
VAR[IABLE] [Variable] [Datatype]	定义一个绑定变量，显示一个绑定变量或者显示全部绑定变量
WHENEVER [OSERROR SQLError] [Action]	指定一个当系统错误或者 SQL 错误发生时执行的动作。动作可以是 EXIT 或者 CONTINUE，可以选择后跟 commit 或者 rollback

这些命令功能同 SQL*Plus。下面的命令是 PL/SQL Developer 特有的：

命令	含义
BEAUT[IFY] File Object	使用当前规则或者由 SET BEAUTIFIERRULES 命令指定的规则美化指定文件或者数据库对象
BROWSE Object	在对象浏览器中选择对象
EDIT Object	打开一个可编辑的对象定义窗口
EDITD[ATA] Table View	打开一个表或者视图并且包含可编辑结果集的 SQL 窗口
EXPORT[DATA] Table	打开指定表的导出工具
INFO	显示本连接的有关信息
PROP[ERTIES] Object	显示一个指定对象的属性窗口
QUERY[DATA] Table View	打开一个表或者视图并且包含只读结果集的 SQL 窗口
REC[OMPILE] Object	重编译对象
SET BEAUT[IFIERRULES] [File]	从指定文件中临时使用美化规则，这个可以被 BEAUTIFY 命令使用
SET COL[WIDTH] [Width]	设置结果集最大字段宽度，如果宽度是 0，则宽度没有限制，默认值是 80
SET EXEC[PAUSE] [ON OFF]	在下一句命令上暂停执行（ON），或者继续下一句命令正常执行（OFF）
SQLPLUS	用当前文件调用 SQL*Plus with the current file.
TEST ProgramUnit	对于指定程序单元打开一个带测试的测试窗口
VIEW Object	打开一个只读的对象定义窗口

8 创建与修改非 PL/SQL 对象(Creating and modifying non-PL/SQL objects)

在 PL/SQL 编程期间，你会发现经常需要创建表、修改限制或者索引，序列复位等等。

PL/SQL Developer 有很多功能让你创建和修改表（以及相关元素）、序列、同义词、库、目录、用户和角色。这里不讲解这些对象的功能，仅仅讲解如何创建、修改和查看他们的属性。更多有关每个对象的功能信息见 Oracle 文档，如 SQL 参考指南（“SQL Reference Guide”）。

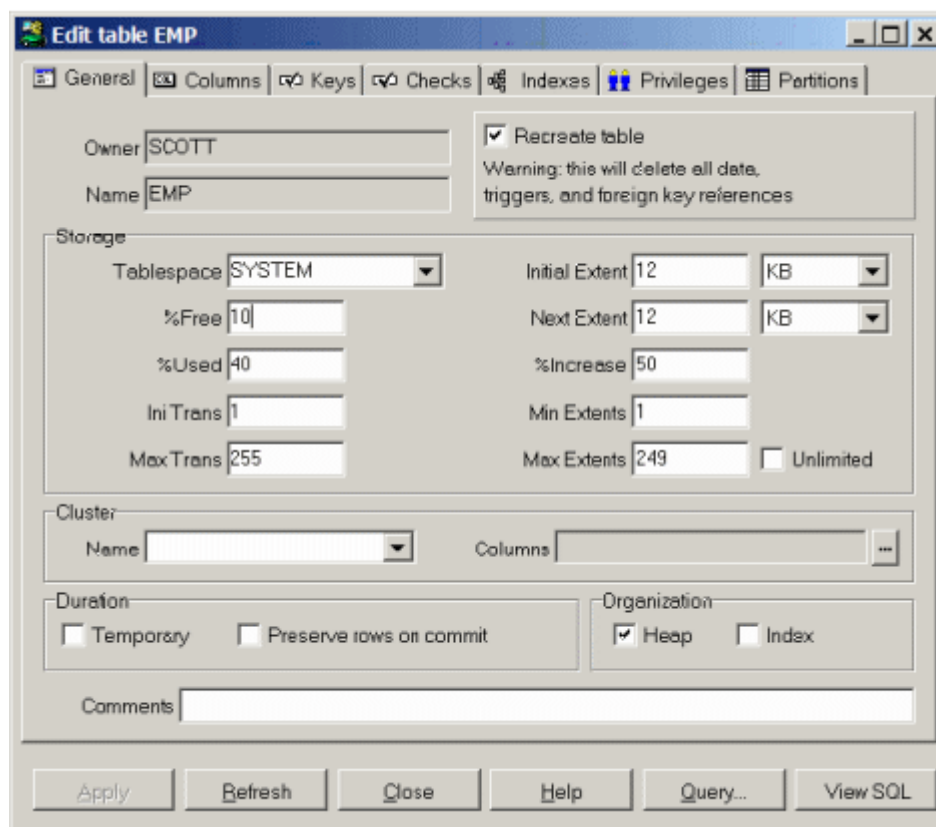
为了创建一个对象，你既可以按工具条中的新建（New）键后选择相应的对象类型，或者选择文件菜单（File）中的新建（New）菜单项。也可以右键点击对象浏览器中对象类型的根文件夹，选择弹出菜单中的新建（New）菜单项。

为了修改一个对象，从对象浏览器中选择该对象，右键点击，从弹出菜单中选择编辑（Edit）菜单项。这种方法不能修改对象名称，必须明确地使用对象浏览器中的更改名称功能完成（如果对象类型支持的话）。如果你只是想查看对象，从弹出菜单中选择查看（View）菜单项。

你可以在对象浏览器的弹出菜单中选择复制（Duplicate）菜单项复制对象，出现的对象编辑器填写除了名称和宿主（owner）（如果可以应用）以外的所有属性。这样你可以快速创建一个类似的对象去探索一个可选择方案、执行一些危险或者破坏性实验、或者其他任何事情。

8.1 表定义编辑器（The table definition editor）

表定义编辑器针对一个表的不同方面分了 7 个标签页：



在编辑器的底部，有下面 6 个按键：

- 应用——将你在编辑器中做的所有修改应用到数据库中。
- 刷新——重新从数据库中获取定义，你做的任何改动都被放弃。
- 关闭——关闭编辑窗口。
- 帮助——显示在线式帮助。
- 查询——调用一个带查询的 SQL 窗口，允许你查看和编辑表数据。
- 查看 SQL——显示一个带有 SQL 语句的文字编辑器，SQL 语句是针对在表定义编辑器中所作的改变而创建的。如果你还没有做任何改变，它将显示完整的创建表 SQL。

下面章节将描述表定义编辑器的不同页。

常规页 (General page)

常规页显示在上节，包含表宿主和名称、存储信息、簇信息和注释。对于一个新表，除了名字外，其他属性都可以为空，所有其他属性将取默认值：

- 宿主——当前登录用户
- 表空间——当前用户的默认表空间
- Free——10
- Used——40
- 初始事务数 (Initial transaction entries) ——1
- 最大事务数 (Maximum transaction entries) ——255

段属性的默认值 (initial extent, next extent, %increase, minimum extents, maximum extents) 依赖于表空间相应的默认值。

簇信息，存储信息和周期属性相互排斥，因为簇隐含定义了表的存储特性，并且临时表不能簇化也有隐含的存储特性。

如果你输入或者选择了一个簇名称，存储属性和时间属性将变成只读，簇字段可以输入。清空簇名称的效果正好相反。

如果你定义表为临时的，簇和存储信息将变为只读，你可以定义提交事物后行是否被保护。

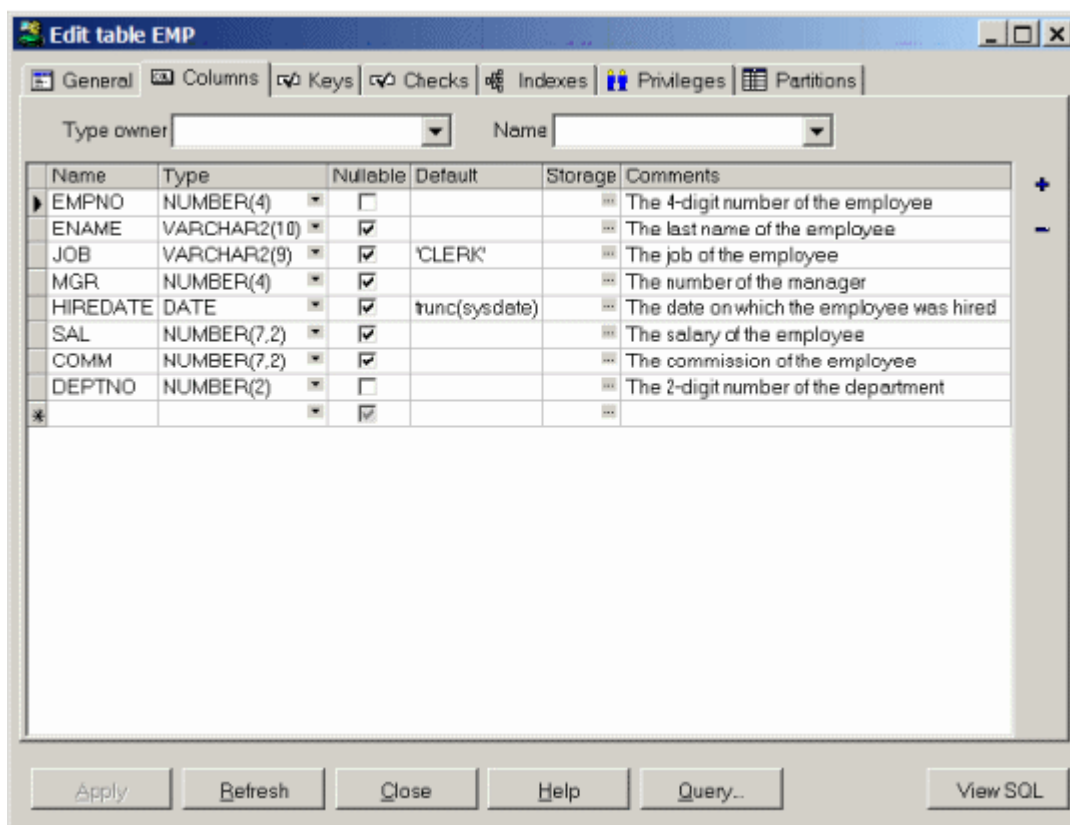
如果你定义一个索引结构表，你必须定义一个主键。对于堆栈结构表（默认）不需要。

如果你修改一个已存在的表，不是所有属性都能修改。如表空间，初始区，最小区，簇信息，时间属性和组织属性。如果你想修改这些属性，勾选窗口上面的重建表 (Recreate table) 复选框。这样，表先被删除后按新定义重新建立。所有数据，触发器，和外键都将丢失，所以，如果用这个选项要相当小心！

字段页 (Columns page)

在字段页，你可以查看、增加、删除、移动和修改表的字段。如果你创建的是对象表

(object table)，你也可以在本页中选择对象类型：



要增加字段，你可以按增加字段（Insert column）键，或者你在最后一行中直接输入新字段信息（这将创建一个新的空的行）。在类型（Type）列有一个包含大部分标准数据类型的列表，你也可以在这里输入其他数据类型。默认列需要你按相应格式输入值，例如，字符串必须用引号括起来。

要删除一个字段，按删除字段（Delete）键。这既可以删除当前光标所在字段，也可以删除所有选择的字段。你可以通过点击行标题头选择一个或者多个字段，拖动鼠标将选择一个行区间。点击行标题头时按 Control 键可以选择多个独立行。

要移动一个或者多个字段，按上节讲的方法选择它们，现在你可以点击其中一个选择的标题头，移动到希望的位置。

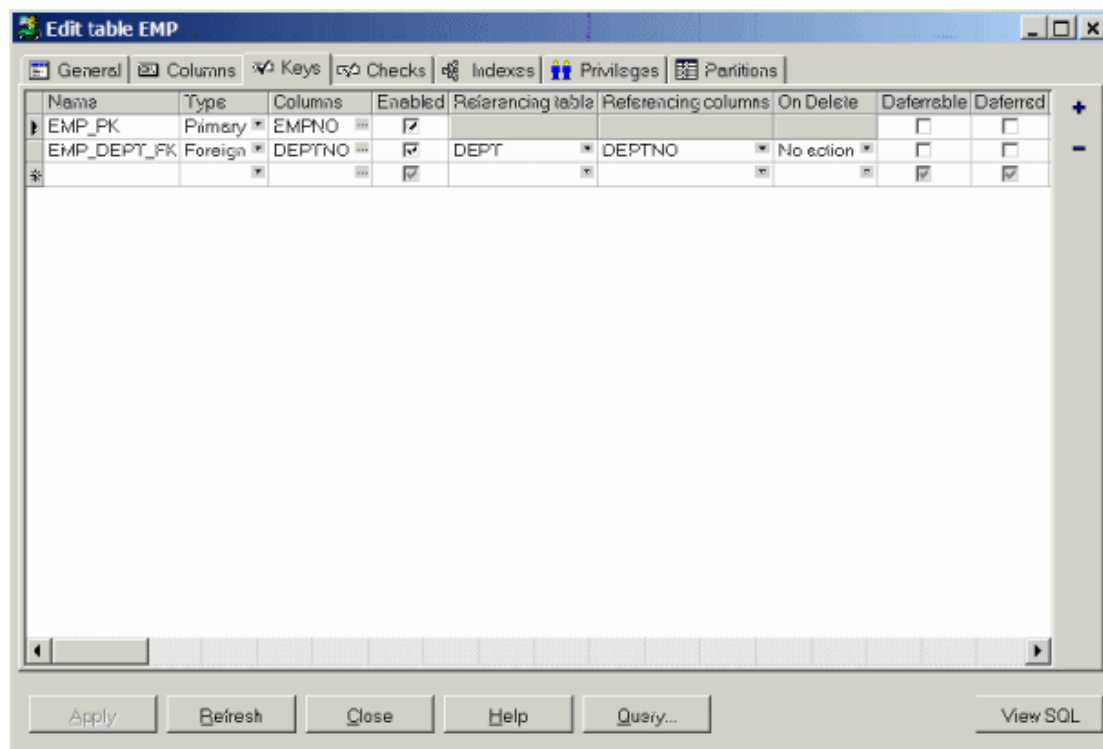
对于已存在的表，在 Oracle9.2 或者更高版本上你可以修改字段的名称，在 Oracle8i 或者更高版本你可以删除字段。为了克服这些限制，你可以在常规页中勾选重建表选项。这个选择的结果在前章节已经描述。

注意修改字段名称将自动传播到限制字段，索引字段和簇字段。

在这个标签页的上方如果你选择类型宿主和类型名称，你将创建一个基于这个对象类型的对象表。字段名称和类型将被对象类型定义和限制。你也可以定义一个字段是否可为空，默认值，它的存储和注释。如果你后期想修改这个对象类型，你必须在常规页中勾选重建表选项。

键页 (Keys page)

在键页你可以查看、增加、删除和修改表的主键、唯一和外键限制：

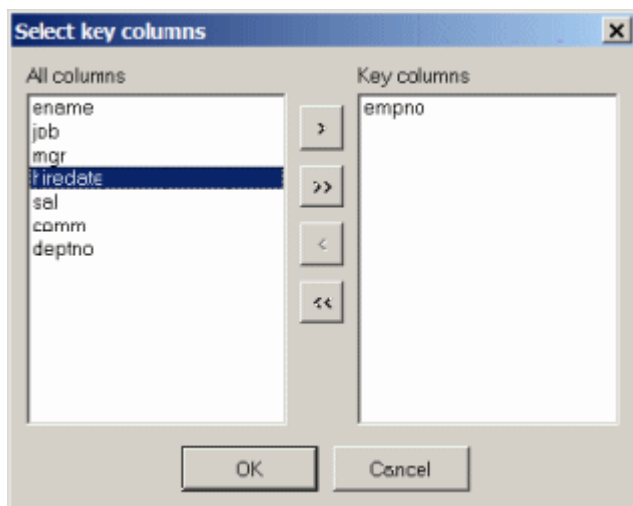


这个表格的操作方法和前面章节中字段表格一样。你可以增加、删除和修改键限制。移动限制没有任何影响，因为他们不是排序的。

注意，在用于强制执行一个主键或者唯一键限制下的索引的存储信息在这页没有被定义。如果启用限制，一个相应的同名索引将自动在索引页中。对于这个索引你可以定义存储。禁止或者删除主键或者唯一键限制将隐含删除对应的索引。

类型列有一个列表，允许你只能选择主键、唯一或者外键类型。你可以输入首字母快速选择正确的类型 (P, U 或者 F)。注意一个表只能有一个主键，并且这个字段的默认值（主键或者唯一）依赖它（and the default value for this column (Primary or Unique) depends on this）。修改这个类型将影响到关联表，关联字段和级联（Cascade）字段，因为外键限制只能引用主表的外键或者唯一索引。修改类型成外键也将隐含删除索引。

定义组合字段限制，你可以简单地在相应单元格输入字段名（用逗号分隔），或者按单元格按键，这将调用一个字段选择屏幕，你可以简单增加、删除或者移动（排序）键字段：

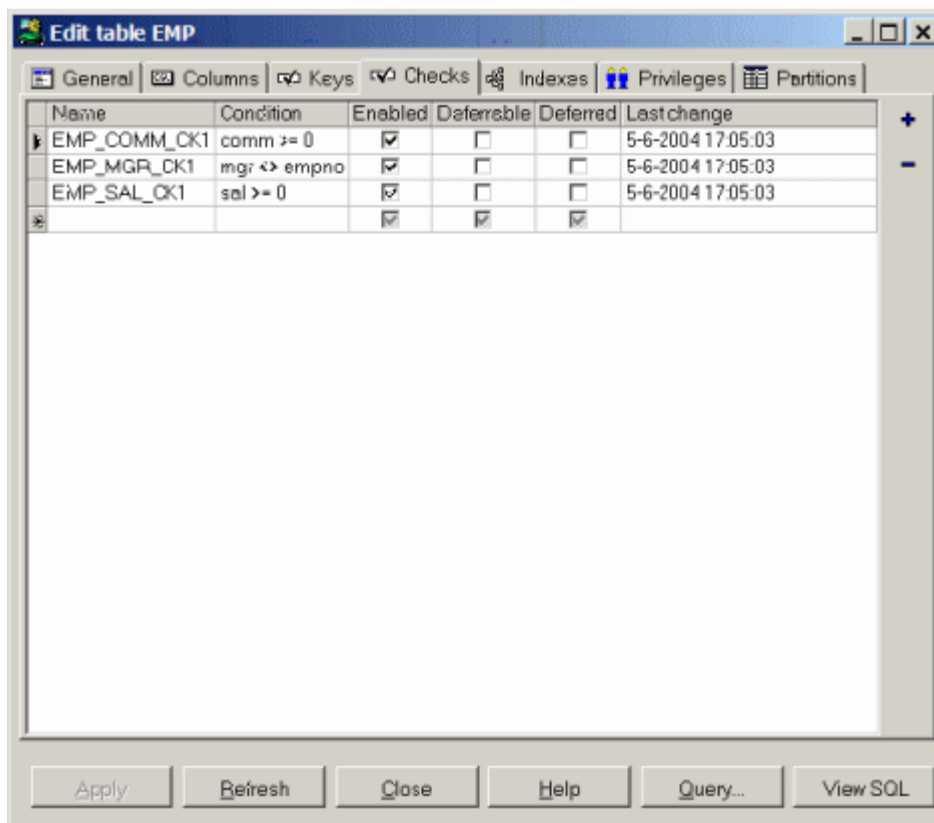


你可以输入或者选择一个外键限制的参引表。这样做将自动根据该表的主键或者唯一索引选择默认参引字段。这个字段的选择栏中显示参引表中的主键或者唯一索引的全部字段集，因此这也是唯一的候选人。在删除时列表框（On Delete Listbox）中，你可以定义当主表记录删除时，子表记录应该采取的操作。在 Oracle8i 和更高版本中可以为空。

如果你正在使用 Oracle8，你可以额外定义是否延缓限制和是否初始延缓。这有 2 个完全对立的选择：你不可能有非暂缓限制而其初始却是暂缓的。

检查页（Checks page）

在检查页你可以查看、增加、删除和修改表的检查约束：



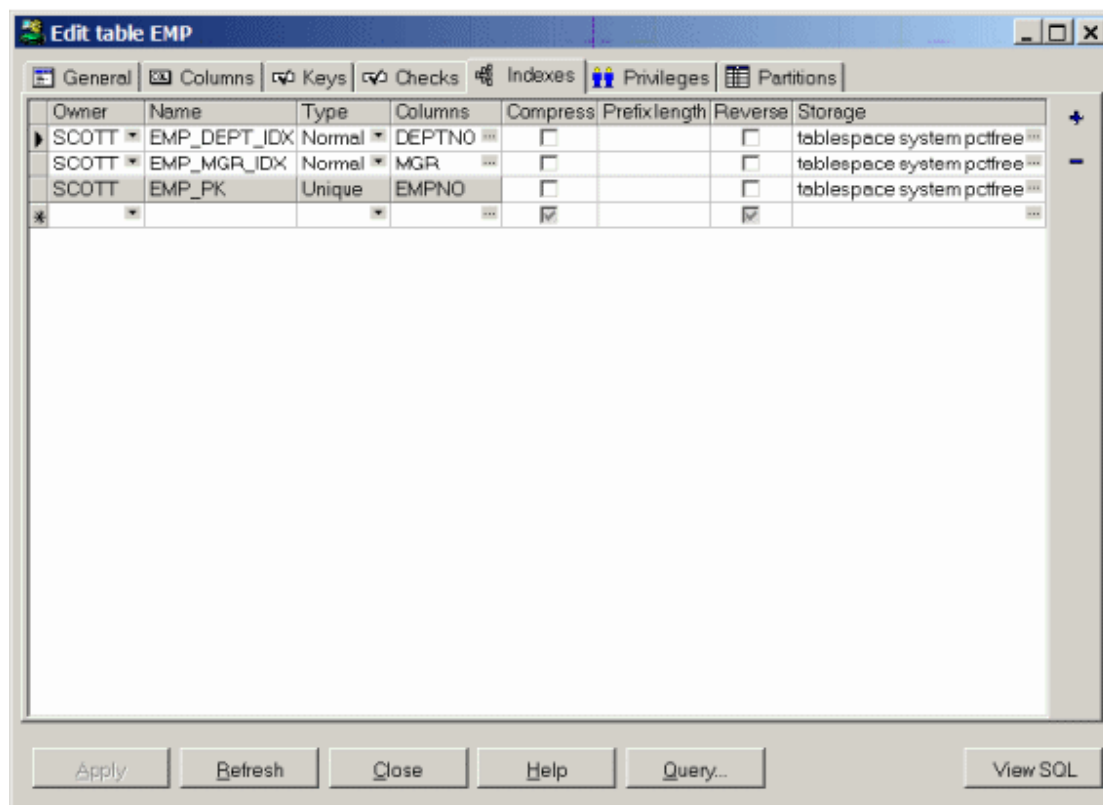
这个表格操作同字段表格，你可以增加、删除和修改检查约束。

条件列应该包含对于表中所有行必须为真（或空）的布尔表达式。如果一个或多个记录违反条件，你就不能应用检查限制，除非你未选可用（Enable）选项。在修改了违反条件记录后，你可以运行检查限制。

如果你正在使用 Oracle8 或更高版本，你可以额外定义是否延缓限制和是否初始延缓。这有 2 个完全对立的选择：你不可能有非暂缓限制而其初始却是暂缓的。

索引页（Indexes page）

在索引页中你可以查看、增加、删除和修改表的索引：



表格的操作同字段表格操作。你可以增加、删除和修改索引。

索引的宿主可以为空，默认的宿主是当前登录的用户。如果你想使用不同的用户，只要从列表中选择就可以了。

索引的类型可以是普通、唯一或者位图 (Bitmap)。最后一个索引类型只适合 Oracle7.3 和更高版本。注意，只有索引字段在数据库中确实唯一的情况下才能创建唯一索引。只要不是这种情况，你不能在数据库中应用这个索引。

要定义索引字段，你可以简单地录入字段名在相应的单元格（用逗号隔开），或者按单元格按键，这将调用一个字段选择屏幕，在这里你可以方便地增加、移除或者排序索引字段。

对于函数索引，你可以指定函数，用逗号隔开。对于降序索引，你可以在字段后加 DESC。

你可以额外允许索引的键压缩，包括压缩的前导字段数和定义索引为反向索引。

如果索引是主键或者外键，宿主、名称、类型和字段不能修改。这些属性都来自于键限制。改变键限制的名称将自动改变相应的索引名称，改变限制字段将改变相应的索引字段。宿主将总是同表的宿主，并且索引类型总是唯一类型。

存储参数可以按单元格按键定义或者查看，这将调用一个存储编辑器：

The image shows a 'Storage parameters' dialog box with the following fields and values:

Field	Value	Unit
Tablespace	USER_DATA	
Initial Extent	10	KB
%Free	10	
Next Extent	10	KB
%Used		
%Increase	50	
Ini Trans	2	
Min Extents	1	
Max Trans	255	
Max Extents	121	
<input type="checkbox"/> Unlimited		

Buttons: OK, Cancel

对于新索引，你不必输入任何信息，每个属性都有默认值：

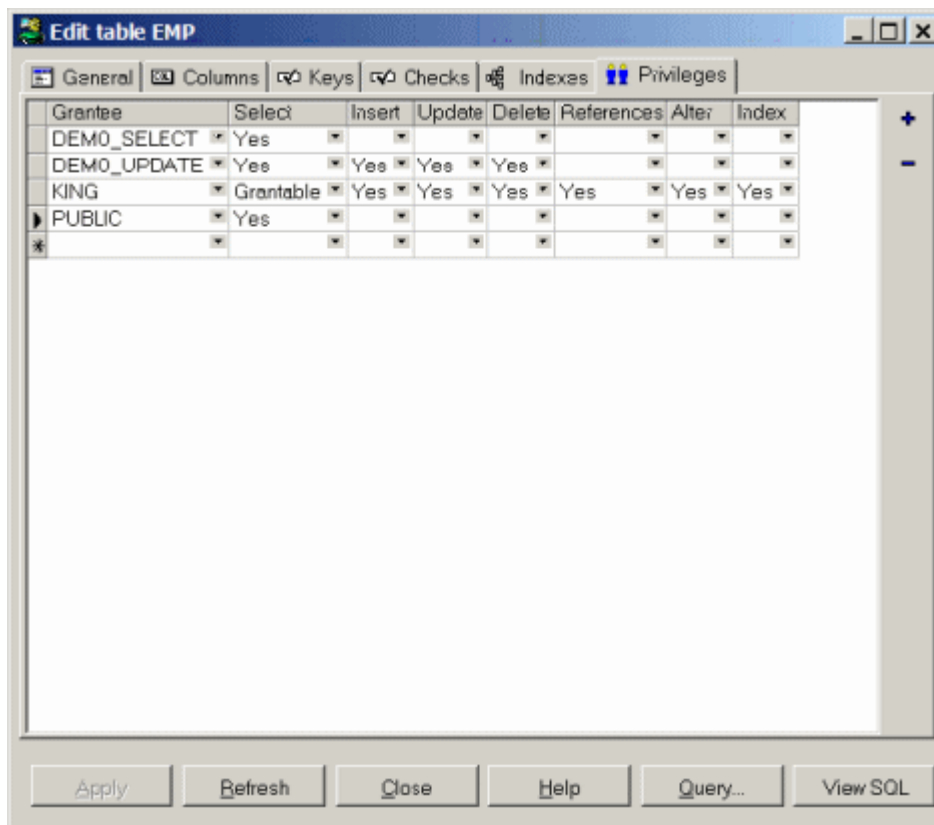
- 表空间——当前用户的默认表空间
- %Free——10
- 初始事务数——2
- 最大事务数——255

区属性的默认值（初始区（initial extent），下一区（next extent），增长率（%increase），最小区（minimum extents）和最大区（maximum extents））依赖于相应表空间的默认值。索引不使用%Used 属性。

注意如果你修改一个已存在索引的%free，初始区和最小区属性，这将暗含着索引（和限制）被删除并按这个改变重新创建。对于大索引，这将很耗时。

权限页 (Privileges page)

在权限页中你可以授予/收回用户和角色在表上的权限：



这个表格的操作同字段表格操作。你可以增加、删除和修改一个授予者。删除一个授予者等于收回他的所有权限。

授予者是你想授予特定权限的用户或者角色。注意权限授予一个角色经常不能满足开发存储过程程序单元的需要。如果一个用户通过角色授予一个表的查询权限并且在存储过程程序单元中引用了这个表，编译会因为不知道表而失败。用户必须单独授予查询权限才能在他（她）自己的编程单元中使用。

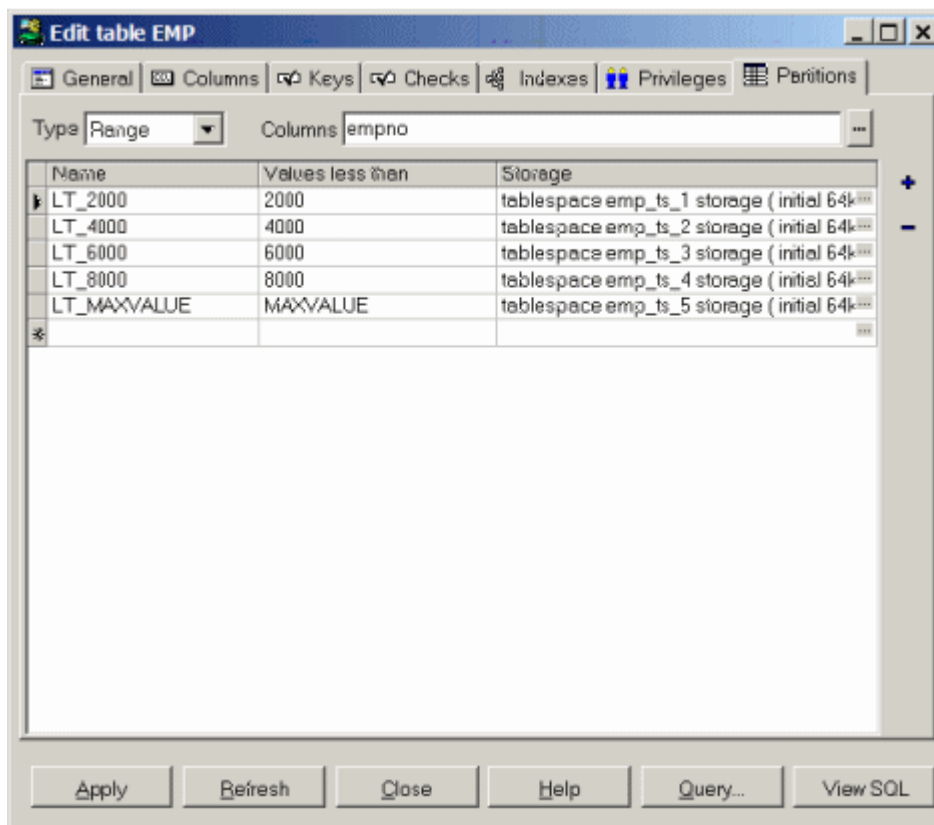
个人权限列有 3 个值：

- 空 (Blank) —— 授予者没有权限
- 是 (Yes) —— 授予者有权限
- 可授予的 (Grantable) —— 授予者有权限并且可以将权限授予其他的用户或角色

查询、插入、更新和删除权限允许授予者在表中对记录执行这些操作。参数权限允许授予者创建表的外键。修改权限允许授予者通过 `alter table` 命令修改表定义。索引权限允许授予者创建表索引。

分区页 (Partitions page)

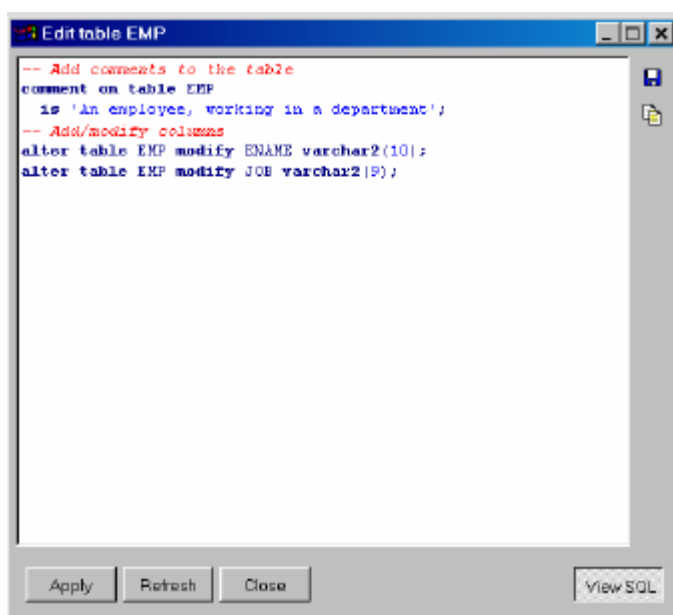
分区页只有在（重）创建一个表或者你编辑一个已存在的分区表时可用：



在页上方你可以选择 Rang, Hash, 或 List 分区, 分区字段也一样。在表格中你可以定义每个分区的名称, 字段值 (如果可应用的话) 和存储。如果你定义了一个以上的分区字段, 你必须使用逗号分割。

查看和修改 SQL (Viewing and modifying the SQL)

创建一个表或者修改一个已存在的表后, 你可以按查看 SQL (View SQL) 键查看对应的 SQL。增加一个表注释并且修改 ename 和 job 字段为 varchar2 数据类型后, 对应的 SQL 如下:



按窗口右上角的相应按键可以将 SQL 保存到文件, 或者拷贝到剪贴板。工具条中的打

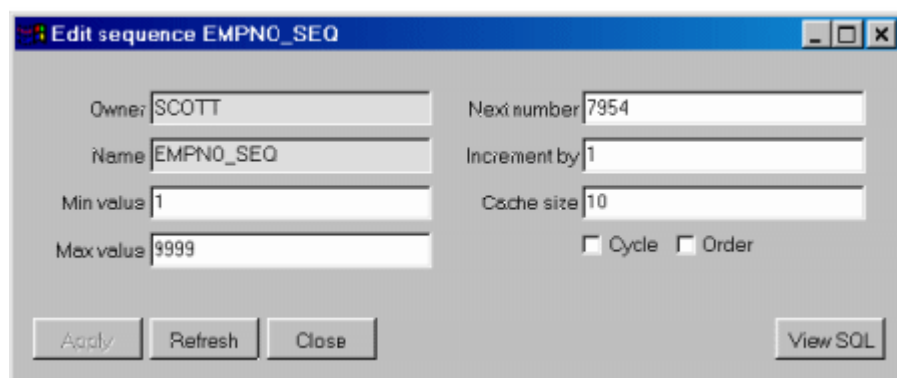
印 (Print) 键可以打印 SQL。你还可以修改 SQL 并应用 (Apply) 修改的 SQL，应用修改的 SQL 后，窗口将返回到表单模式并刷新表定义。SQL 产生的变化立刻反映在表单中。

如果你再次按查看 SQL (View SQL) 键，窗口将返回表单模式，任何修改后未被应用的 SQL 将被丢弃。

如果你是在对象浏览器中选择查看 (View) 来查看表定义，查询 SQL (View SQL) 键将显示创建该表的 SQL，在它之前不包括一条删除表命令。这种方式你可以保存一个表定义到文件中，或者拷贝定义到其他数据库或用户。

8.2 序列定义编辑器 (The sequence definition editor)

序列编辑器让你方便地创建或者修改一个序列：

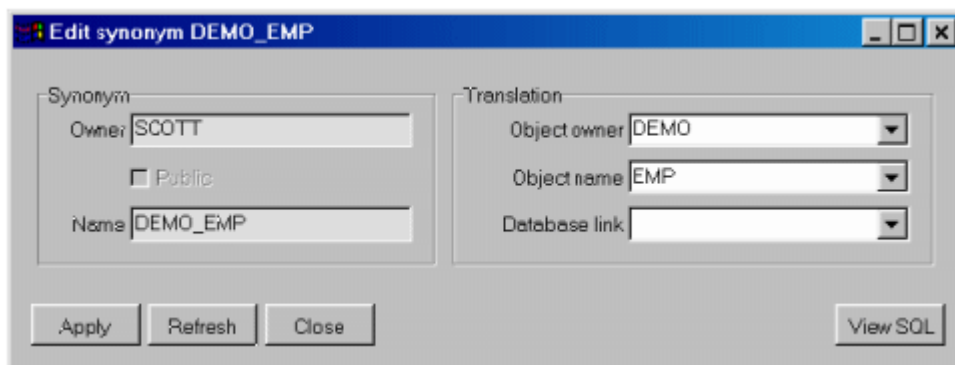


多数属性都有默认值，创建一个新的序列时你只需要提供一个名称：

- 宿主——当前登录的用户
- 最小值——1
- 最大值——无限制
- 初始值——1
- 增量——1
- 缓冲大小——20
- 循环——关闭
- 排序——关闭

8.3 同义词定义编辑器 (The synonym definition editor)

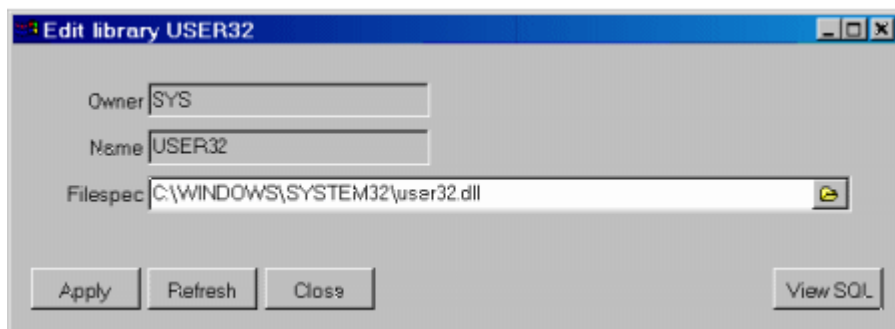
同义词编辑器让你方便地创建或者修改一个同义词：



默认的宿主值是当前登录的用户。勾选公共（Public）选项将创建一个公共同义词，将使宿主为只读。对象宿主和对象名称域为了方便选择一个代表对象的同义词提供了一个建议列表，如果对象宿主域为空，对象名称列表将显示所有对象。

8.4 库定义编辑器(The library definition editor)

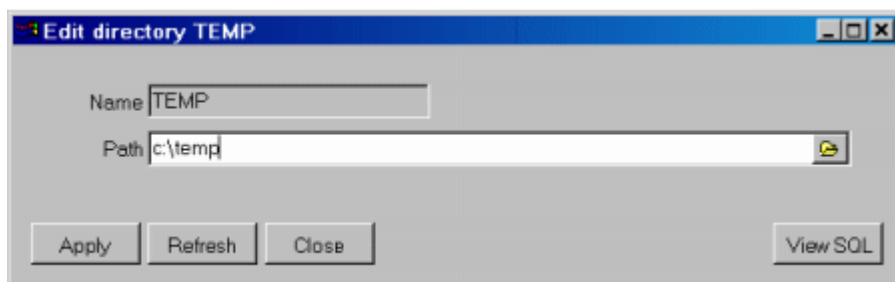
库编辑器让你方便地创建和修改一个库：



默认的宿主值是当前登录的用户。你可以使用建议列表创建一个带有不同宿主的库。Filespec 是你想使用的 3GL 库的全路径名。如果你正在数据库服务器上开发，你可以使用选择 DLL 文件（Select DLL file）按钮选择动态连接库。否则，你必须输入 DLL 名（Windows）或者共享对象（UNIX）。Filespec 是否有效不被验证。

8.5 目录定义编辑器(The directory definition editor)

目录编辑器让你方便地创建和修改一个目录，这个目录随后被用在 BFILE's：



如果你正在数据库服务器上开发，你可以用选择目录（Select directory）按钮选择目录路径。目录路径是否有效不被验证。

8.6 作业定义编辑器(The job definition editor)

作业编辑器让你创建和修改一个作业：

The 'Edit job 21' dialog box contains the following fields and values:

- Submitter: SCOTT
- Job: 21
- What: begin clear_transaction_log; end;
- Next date: 28-8-2002
- Interval: sysdate + 2
- Priv user: SCOTT
- Schema user: SCOTT
- Last date: (empty)
- This date: (empty)
- Total time (sec): 0
- Failures: 0
- NLS environment: NLS_LANGUAGE='DUTCH' NL
- Broken: ☐

Buttons: Apply, Refresh, Close, Help, View SQL

创建一个新作业等于调用 `dbms_job.submit`。修改一个已存在的作业等于调用 `dbms_job.change`。更多有关作业信息见 Oracle Supplied Packages Reference 手册中的 DBMS_JOB。

在 What 域旁边的打开 PL/SQL 编辑器 (Open PL/SQL Editor) 按钮调用一个更方便的 PL/SQL 编辑器，该编辑器带有语法高亮，代码助手等等。

8.7 队列定义编辑器(The queue definition editor)

队列编辑器让你创建和修改一个高级队列：

The 'Create queue' dialog box contains the following fields and values:

- Owner: SCOTT
- Name: DEPT_QUEUE
- Table name: DEPT_QUEUE_TABLE
- Queue type: Normal queue
- Comment: (empty)
- Max retries: 5
- Retry delay (sec): 0
- Retention time (sec): 0

Buttons: Apply, Refresh, Close, Help, View SQL

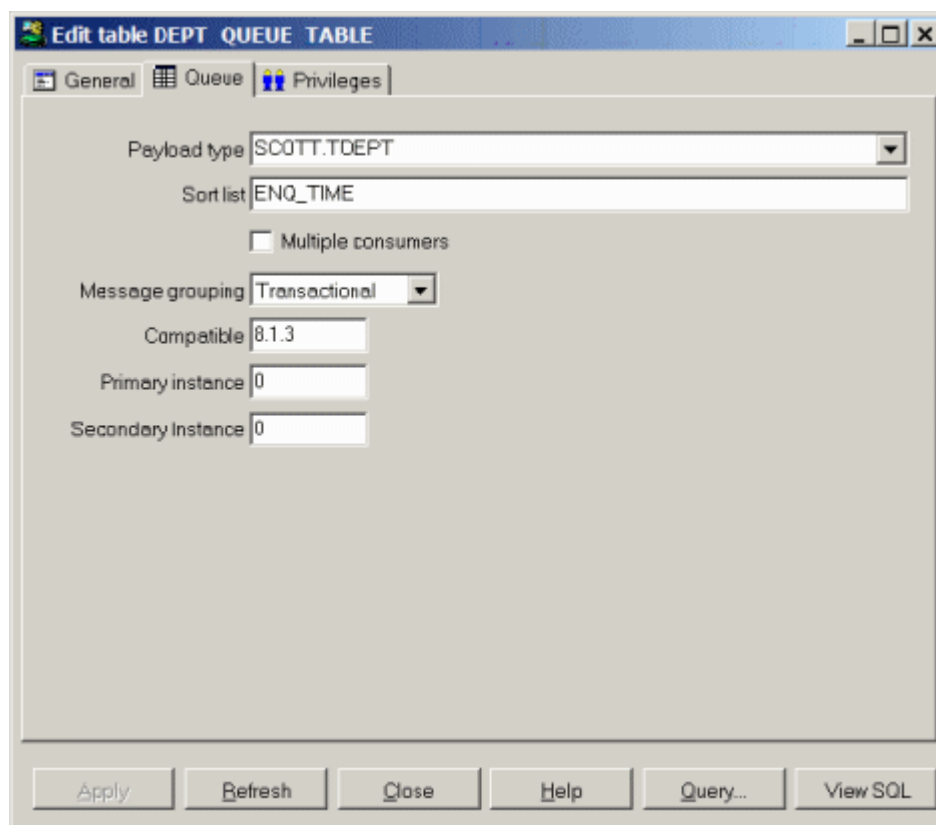
对于队列表名你可以选择一个已存在的队列表，或者你可以按创建 (Create) 键创建一个新队列表（见队列表定义编辑器）。

更多有关 Oracle 高级队列的信息见下面的 Oracle 文档：

- 应用开发指南－高级队列（Application Developer's Guide - Advanced Queuing）
- 代理的 PL/SQL 包和类型参数(Supplied PL/SQL Packages and Types Reference)

8.8 队列表定义编辑器(The queue table definition editor)

队列表编辑器让你创建和修改一个高级队列表：



在常规标签页中你可以定义表的名称、存储参数和注释。在权限标签页你可以授予表的权限给用户和角色。参见表定义编辑器（The table definition editor）。

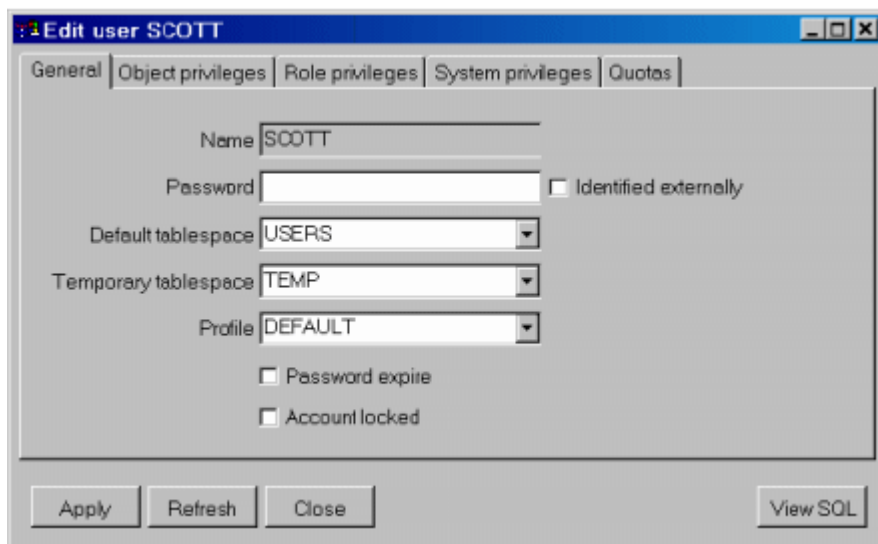
在队列页你可以提供队列表的特殊信息。对于有效负荷类型（payload type），你可以选择一个已存在的对象类型，或者 RAW 创建一个裸队列。

更多有关 Oracle 高级队列的信息见下面的 Oracle 文档：

- 应用开发指南－高级队列（Application Developer's Guide - Advanced Queuing）
- 代理的 PL/SQL 包和类型参数(Supplied PL/SQL Packages and Types Reference)

8.9 用户定义编辑器(The user definition editor)

用户编辑器让你创建和修改一个用户，以及用户的权限和表空间限额：



注意你需要特殊系统权限（象 DBA 角色提供的权限）来查看和修改用户信息。

常规页（General page）

在常规页中，你基本上只需要输入姓名和密码。所有其他属性都有一个默认值：

默认表空间——SYSTEM

临时表空间——SYSTEM

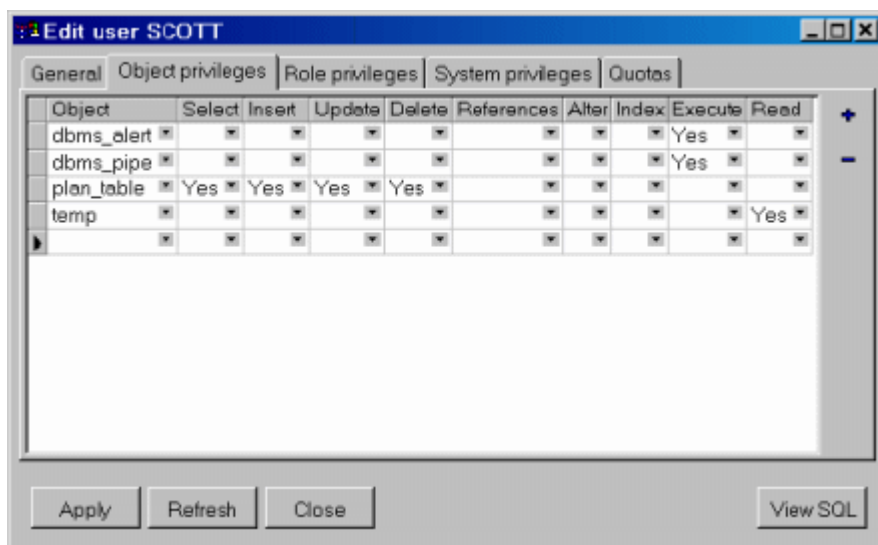
Profile——DEFAULT

密码过期——关闭

账户锁定——关闭

如果你修改一个已存在的用户，当前密码不被显示。修改密码可以输入一个新的密码，或者设置为空不修改密码。如果用户是外部识别(identified externally)，你不能输入一个密码。

对象权限页（Object privileges page）



这页让你查看和修改授予用户的对象权限。这不包括通过角色而间接授予的权限。

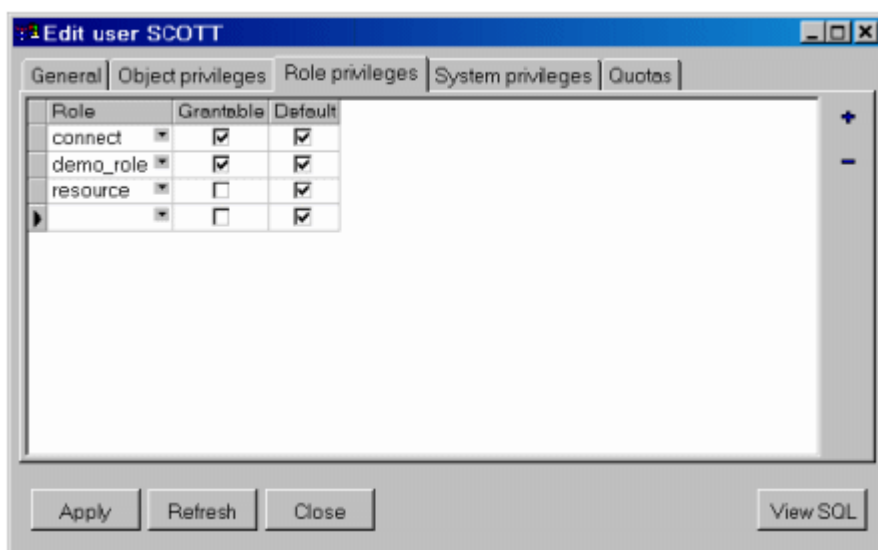
在第一列你可以看到授予用户权限的对象，其他列表示一个特殊权限。每个权限可以有下面的值：

空——用户没有权限

是（Yes）——用户有权限

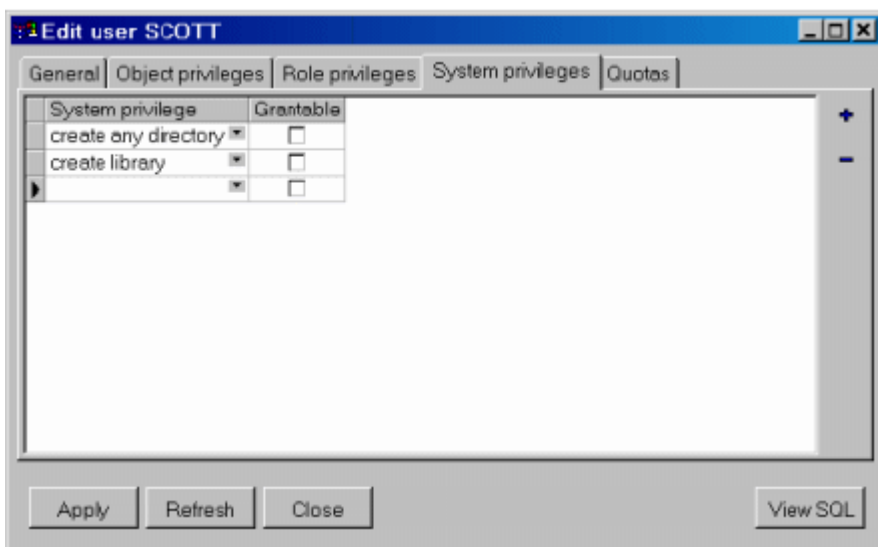
可授权（Grantable）——用户有权限并且可以授予其他用户或者角色权限

角色权限页（Role privileges page）



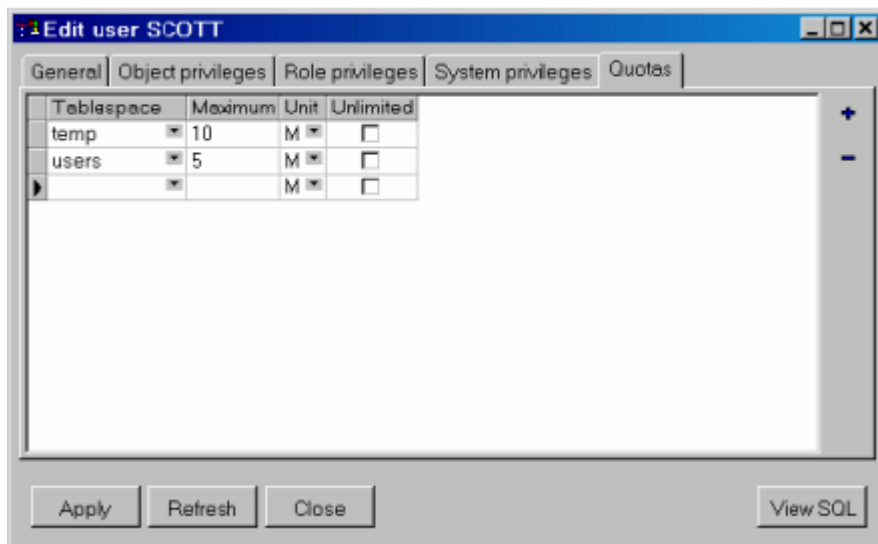
这页显示已经授予用户的角色，每个角色的权限是可授予的，所以用户可以将它授予其他用户和角色。如果一个角色权限是默认的，当用户登录，角色将有效，如果它不是默认的，角色权限必须通过用户登录后执行一个 `set role` 命令才能在会话中实际有效。

系统权限页（System privileges page）



这页显示授予用户的系统权限。每个系统权限是可授予的，因此用户可以将它授予其他用户和角色。

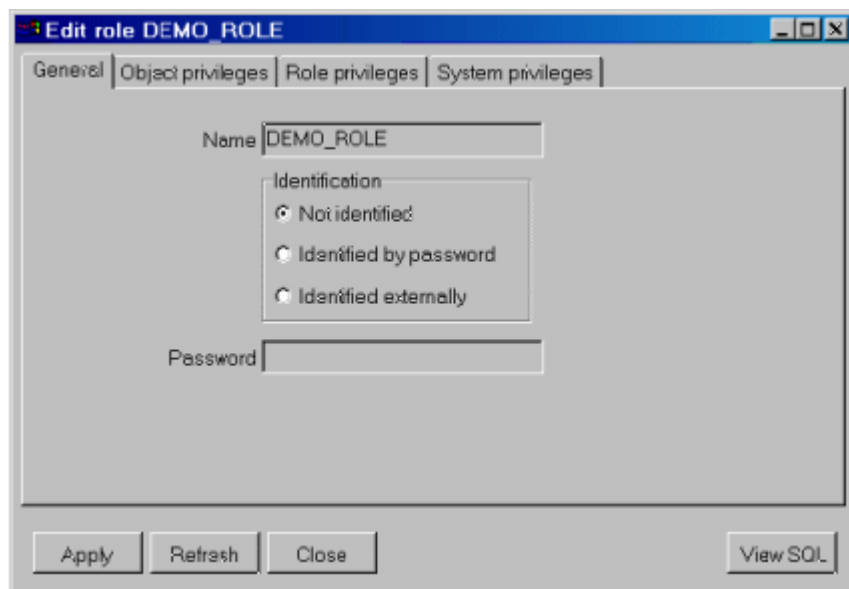
限额页 (Quotas page)



这页显示用户的表空间限额。你可以输入一个用户可以使用的表空间的最大字节数、K 字节数或者兆字节数，你也可以指定不限制。

8.10 角色定义编辑器(The role definition editor)

角色编辑器让你创建和修改一个角色和它的权限：

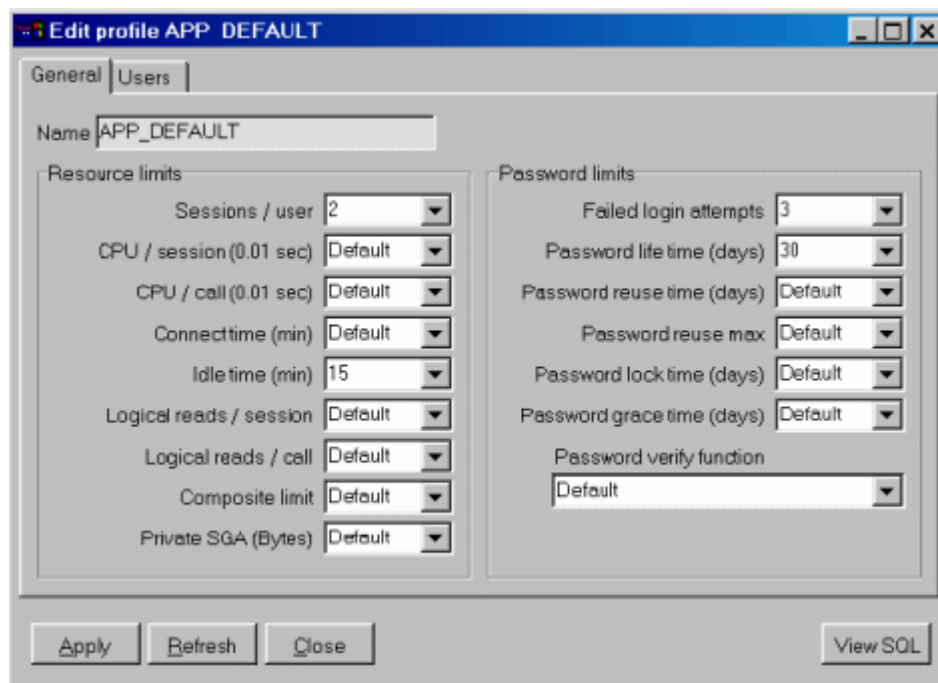


默认情况下，一个角色不需要被确认。你也可以对一个角色定义密码确认或者外部确认。密码确认需要你输入一个密码，当一个角色没有通过默认授予用户时将非常有用。象这样的角色需要用户登录后明确设置，并且需要密码。

其他 3 个标签页（对象权限，角色权限和系统权限）和用户定义的相关页相同。

8.11 profile 定义编辑器(The profile definition editor)

profile 编辑器让你编辑资源限制，密码限制和 profile 的用户：

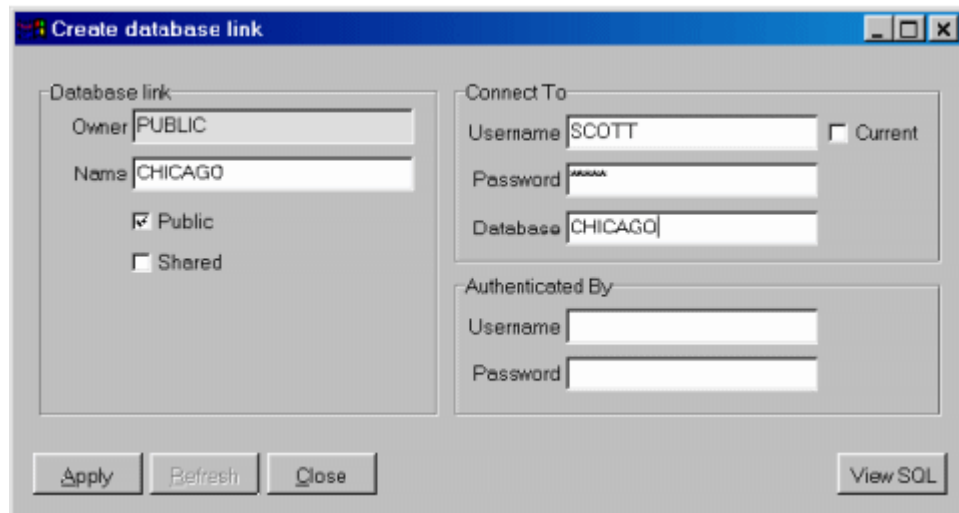


每个限制可以设置成默认值（Default），在这种情况下，值可以从标准的 profile DEFAULT 中继承，还可以设置成无限制或者设置一个具体值。

在用户标签页中，你可以定义这个 profile 应用的用户。注意，如果你从 profile 中移除一个用户，该用户将隐含地修改为 DEFAULT profile。

8.12 数据库连接定义编辑器(The database link definition editor)

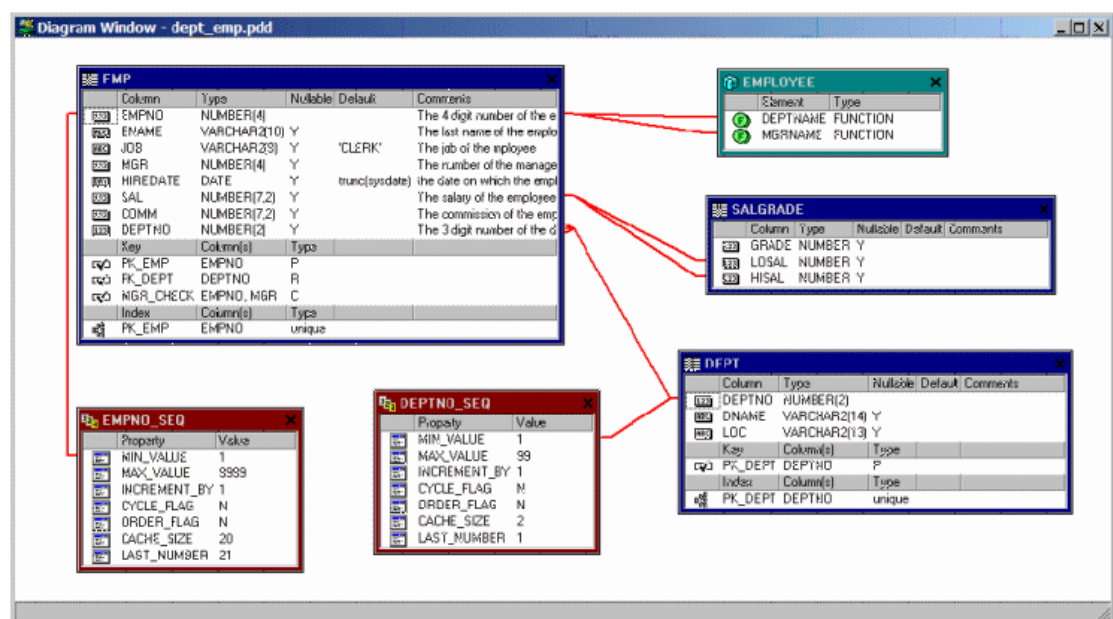
数据库连接编辑器让你创建和编辑一个数据库连接：



注意，密码（Password）和验证（Authenticated By）属性在你编辑一个已存在的数据库连接时不被显示。在编辑数据库连接时都需要输入这个信息。

9 图表(Diagrams)

图标窗口允许你创建一个选择对象的图展现。这样你可以很方便地可视化（一部分）应用程序或者工程中数据库对象关系：



你可以在图表中增加关系对象类型和他们之间包括的关系。表之间的外键关系自动包含。对于大的图标，你可以额外增加包含对象组的结构。

一个图表可以用于文档，但也可以作为一个工作区，右键点击一个对象提供存取所有对象功能，双击一个对象将调用对象类型默认动作。

9.1 创建一个图表(Creating a diagram)

为创建一个新的图表，选择文件（File）菜单中的新建（New）子菜单中的图标窗口（Diagram Window），出现一个空的图表。要在图表中增加数据库对象，从对象窗口拖动对象到图表窗口。你可以一次拖拉多个对象。

移动一个图表对象，点击标题栏后拖动到新的位置。调整图表对象大小可以拖动边缘或者一个角。从图表中删除一个对象，按对象的右上的 X。这将只影响图表，不删除数据库中的对象。

显示和隐藏项(Showing and hiding items)

显示或者隐藏一个图表对象的特殊项，右键点击它，进入项目（Items）子菜单。这里你将看到一个被选择对象可应用的所有项目的子菜单。对于一个表，你可以选择字段、键或者索引的哪些项目可以显示。你也可以完全隐藏键或者索引部分。

在默认情况下，显示全部项目，但是如果你右键点击一个对象，并且选择按默认值设置（Set as default），该对象的可显示项目将应用到所有后期增加的对象。对于对象类型可以使用同样方法。

增加关联（Adding relations）

要增加 2 个对象之间的关联，点击原对象的元素的图标，拖动到目标元素，2 个对象元素之间将画一条线。你可以选择线并上下移动选择点来将线移动到其他元素上。为了移除一条线，选择它并按删除键。

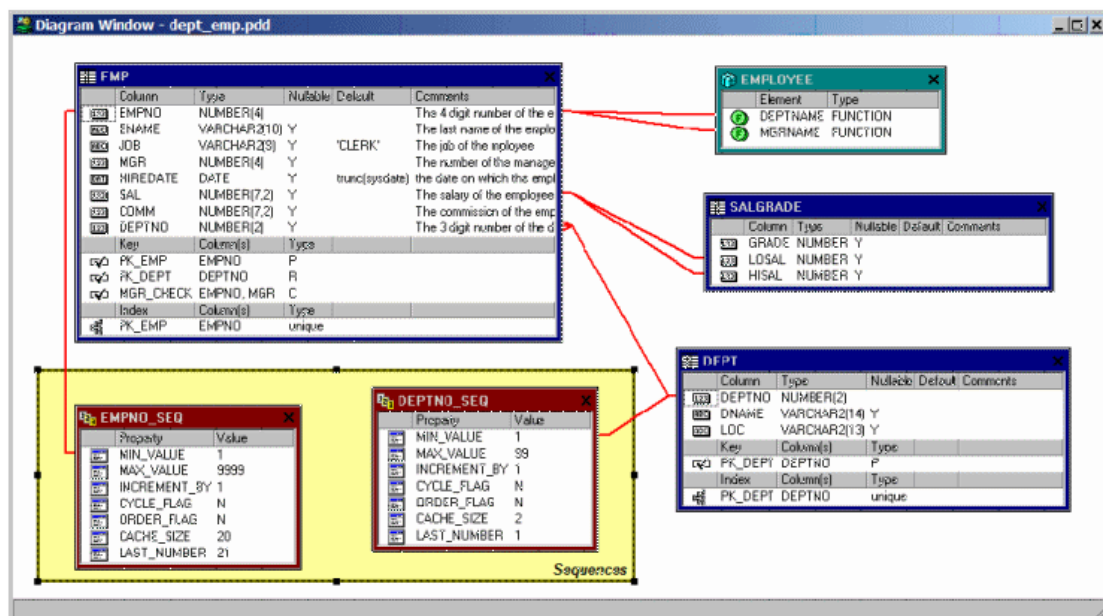
如果你在图表中增加一个表，将自动包含对于图表中已有表的外键关系。如果需要，这些关联可以删除。

默认情况下，关联将划一个直线。你可以在线上右键点击并从弹出菜单中选择插入弯曲来增加额外的弯曲，作为结果，一个新选择点被增加到线上。

为了创建一对多关联，右键点击线的端点，从弹出菜单中选择改变线端点（Change line end）。

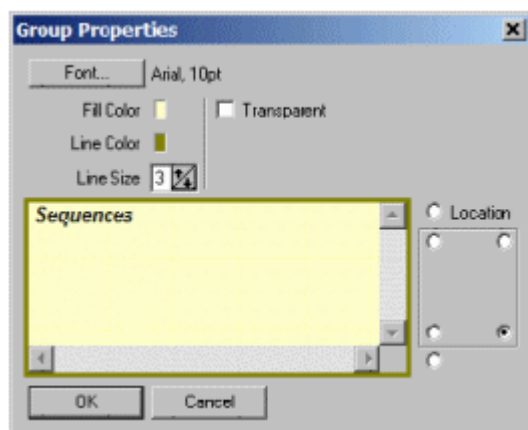
增加组（Adding groups）

为了一个大图表增加附加结构，你可以创建一个组。组可以有不同的颜色，可以在上下左右的位置上放置标题：



要增加一个组，在图表的空白处右键单击，从弹出菜单中选择增加组（Add group），在这个位置处将出现一个组，你可以调整到适合的尺寸。任何被组完全覆盖的对象都暗含成为这个组的一部分，移动组键同样移动这些被附加的对象。如果你在移动组时同时按下 Ctrl 键，附加对象将不被移动。

为修改一个组的外观，在它上面右键单击，从弹出菜单中选择属性（Properties）项。你也可以双击组。出现下面屏幕：



你可以修改组的字体、填充颜色、线颜色和线大小，组透明。你可以输入组的标题，指定标题的位置。

为了创建一个默认组外观，在组上右键单击，选择设置为默认。这将使外观应用到所有后增加的组。

为了从图表中删除一个组，选择它并按删除键。这将只删除组，不删除被附加的对象。

9.2 保存和打开一个图表文件(Saving and opening a diagram file)

要保存一个图表到一个文件中，从文件（File）菜单中选择保存（Save）菜单项。一个图表将被保存到带.pdd 扩展名的文件中，以后可以再次打开。为定义一个图表文件的目录，进入目录参数（Directories preference）页设置相应的选项。

9.3 更新一个图表(Updating a diagram)

一个图表是数据库的一个静态展现。如果数据库对象在将来改变，你将需要更新图表反映这些变化。做到这点，右键点击一个图表对象，从弹出菜单中选择从 DB 更新（Update from DB）。为更新图表中的所有对象，右键点击它选择全部从 DB 更新（Update All from DB）。

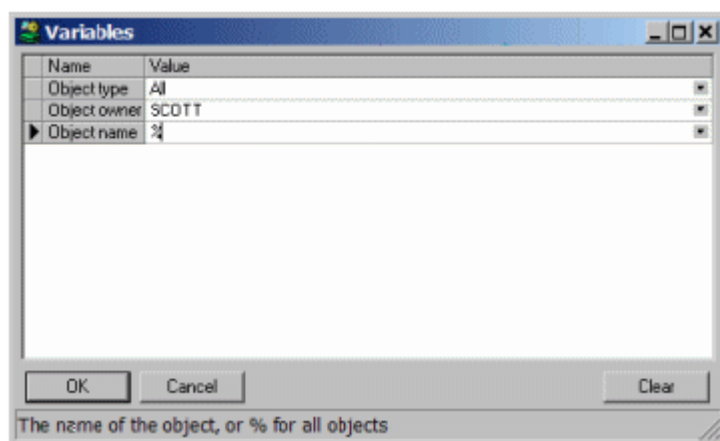
如果一个对象从数据库中删除，你也需要从图表中删除。如果一个对象在数据库中被重命名，你将需要从图表中删除后再用新名称增加它。

10 报表(Reports)

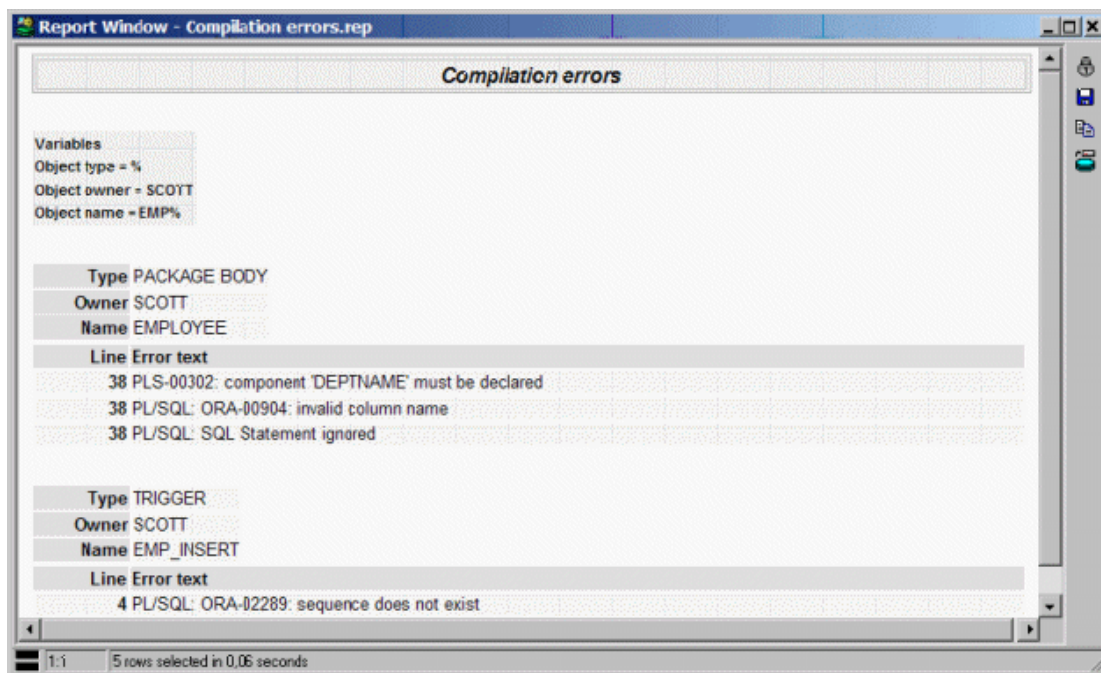
要运行你的程序数据或者 Oracle 字典报表，你可以使用 PL/SQL Developer 内置的报表功能。有很多标准的报表，你也可以很容易地创建你自己的报表。这些自定义报表可以保存到一个报表文件中，并被依次归入到报表菜单中，这使你很容易地经常运行自定义报表。

10.1 标准报表（Standard reports）

标准报表可以通过报表（Reports）菜单得到。例如，如果选择编译错误报表，你首先被提示想要显示的当前编译错误对象的类型，宿主和名称：



当你输入适当的值并按确认键后，报表将被显示：



结果是由 PL/SQL Developer 内置的 HTML 阅读器或者由 Internet Explorer ActiveX control 生成的一个 HTML 文档，这依赖于参数（见用户界面-选项）。

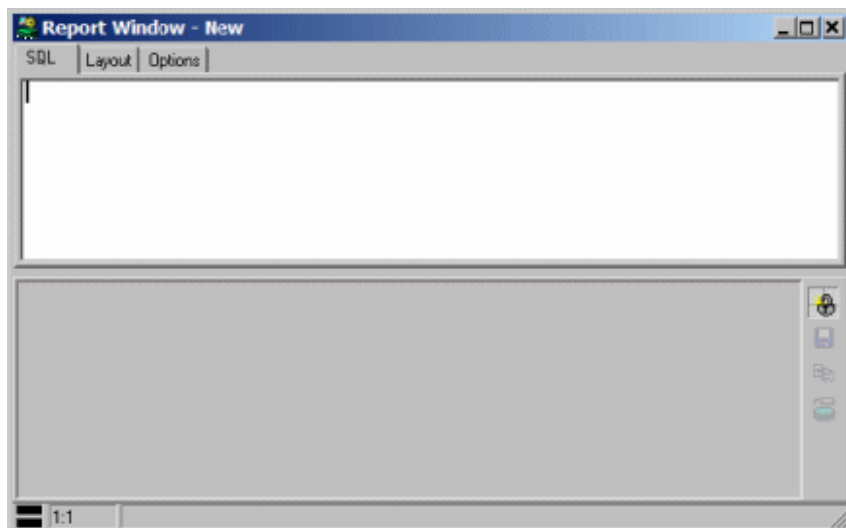
接下来你可以按工具条上的打印键打印报表或者按报表窗口右侧的保存结果（Save results）键以 HTML 格式保存报表。拷贝为 HTML（Copy as HTML）键拷贝报表结果到粘贴板。导出报表（Export Results）键允许你导出结果为 CSV, TSV 或 XML 格式，或者直接导出到 Excel。

如果报表没有被锁定，编辑报表（Edit report）键提供访问报表定义。

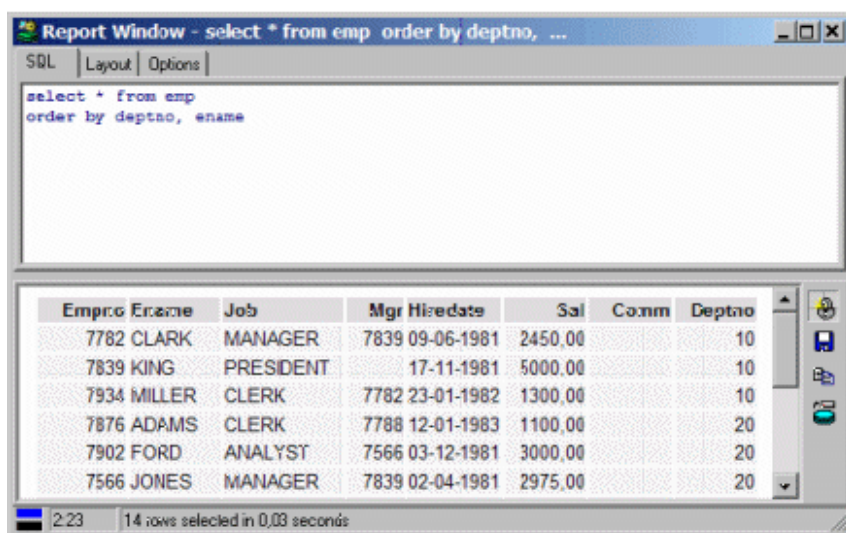
你可以在报表结果中右键点击，选择额外的选项。

10.2 自定义报表（Custom reports）

创建自定义报表十分简单。首先按工具条中的新建（New）键，创建一个新的空报表，然后从弹出菜单中选择报表窗口菜单项。将出现一个空报表窗口：



在 SQL 编辑器中你可以为报表输入一个简单的 SQL 查询语句。对于主/从报表，你可以使用关联查询语句（这将在后面描述）。让我们开始一个例子，使用我们熟悉的 `emp` 表，输入查询语句后按工具条中的执行（Execute）按钮：



这是一个非常基础的报表，只使用了默认风格。下一个章节描述如何精制报表的功能。

10.3 变量（Variables）

经常你的报表在运行前需要一些额外的输入值，让我们假设你想限制上节的查询只是一个部门职工。这样你需要在 SQL 中提供一个置换变量：

```
select * from emp
where deptno = &department
order by ename
```

当这个报表运行，在查询语句执行前，用户被提示输入部门。这当然不很安全，如果用户没有输入会如何？语句将导致一个 `ORA-00936: missing expression` 异常。如果用户输入

一个不是数字的值会如何？或者数字不在部门表中又如何？为防止这些情况，你可以使用在下章讲述的参数选项。这些选项必须指定在括号之间，就象 HTML 或者 XML 文档中的属性。

名称选项 (Name option)

变量的名称被用做提示在变量输入表单中。所以，你需要确保它表达了变量的意思。如果名字是唯一选项，名字是唯一不用指定在括号中的选项。如果名字包含空格或者其他特殊字符，用双引号括起来。下面 2 个例子是一样的：

```
select * from emp
where deptno = &"Department number"
order by ename
```

和

```
select * from emp
where deptno = &<name="Department number">
order by ename
```

提示选项 (Hint option)

名称之外，你可以额外指定一个提示文字。这个提示在变量获得焦点时被显示在变量输入表单的状态栏中：

```
select * from emp
where deptno = &<name="Department number"
                    hint="Only employees from this department will be listed">
order by ename
```

类型选项 (Type option)

类型选项限制用户可以输入的信息，并且也控制如何将值插入到最终的 SQL 文字中：

```
select * from emp
where deptno = &<name="Department number"
                    type="integer">
order by ename
```

这种情况下用户只能输入有效的整型数字的值。输入值将原封不动地替换 SQL 文字。然而对于字符串，你想将值括在引号中，考虑下面的查询语句：

```
select * from emp
where ename = &<name="Employee name"
```



```
type="string">
```

如果你输入 SCOTT（没有引号）在变量输入表单中，最终 SQL 将是：

```
select * from emp
where ename = 'SCOTT'
```

有效类型是：none, string, integer, float, 和 date。日期值没有有效性判断，所以，用户可以额外提供日期表达式，象 sysdate, trunc(sysdate, 'MM')等等。

必需选项（Required option）

对于某个变量，如果你需要一个值，设置它的必须选项为"yes"或"true"：

```
select * from emp
where deptno = &<name="Department number"
              required="yes">
order by ename
```

默认值选项（Default option）

为变量提供一个默认值，指定默认值选项：

```
select * from emp
where deptno = &<name="Department number"
              default="10">
order by ename
```

默认值也可以是一个查询语句，这个查询语句必须只返回一条记录并且只有一个字段。例如：

```
select * from emp
where deptno = &<name="Department number"
              default="select min(deptno) from dept">
order by ename
```

这种情况下，默认值将是最小的部门代码。

如果空选择（Ifempty option）

使一个变量成为必需或者提供一个默认值的一种选择是你可以指定 ifempty 值，以免用户没有指定值：

```
select * from emp
where ename like &<name="Employee name"
              type="string"
```

```
ifempty="% ">
```

如果用户没有指定一个值，百分号将被使用，所有职工将被存取。这个选项不能和必需选项混合使用。

大写选项 (Uppercase option)

如果你有一个字符串变量，并且想暗中将值转换成大写，设置它的大写选项为"yes"或"true":

```
select * from emp
where ename = &<name="Employee name"
              type="string"
              uppercase="yes">
```

值按输入显示，但是被转换成大写到最终 SQL 文字中。

只读选项 (Readonly option)

指定 readonly="yes"可以将变量定义为只读。用户可以查看变量和它的值，但是不能修改它。

隐藏选项 (Hidden option)

你可以通过指定 hidden="yes"隐藏变量。

列表选项 (List option)

列表选项提供用户一个可以使用的值列表。你可以指定一个用逗号分隔的项目列表:

```
select * from emp
where deptno = &<name="Department number"
              list="10, 20, 30, 40">
order by ename
```

这可以让用户选择四个中的一个值，或者输入一个不同的值。既然这样，使用查询语句更有意义:

```
select * from emp
where deptno = &<name="Department number"
              list="select deptno from dept order by deptno">
order by ename
```

你可以为每个项目提供额外的描述:

```
select * from emp
where deptno = &<name="Department number"
```

```

list="10, ACCOUNTING,
      20, RESEARCH,
      30, SALES,
      40, OPERATIONS"

description="yes">

order by ename

```

现在每个项目跟随一个描述，并且增加描述选项。列表将只显示描述，但是在最终 SQL 中使用实际的值。

你可以使用一个含 2 个字段的查询语句做同样的事情：

```

select * from emp

where deptno = &<name="Department number"

list="select deptno, dname from dept order by dname"

description="yes">

order by ename

```

要限制用户只使用列表中的项目，指定限制选项：

```

select * from emp

where deptno = &<name="Department number"

list="select deptno, dname from dept order by dname"

description="yes"

restricted="yes">

order by ename

```

现在用户只能在列表中选择值，不能手工输入其他值。

项目列表有时可以依赖其他参数值。设想你想创建一个报表，显示一个指定表的所有字段，一个表被宿主和名称标识。对于宿主你可以用数据库中所有用户的列表，对于表名称，你可以通过选择宿主查询 all_tables 视图：

```

select * from all_tab_columns

where owner = &<name="Owner"

type="string"

list="select username from all_users order by username">

and table_name = &<name="Table"

type="string"

```

```
list="select table_name from all_tables
      where owner = :owner
      order by table_name">

order by column_id
```

你可以看到，第二个（表）列表的查询语句通过 `owner` 绑定变量引用了第一个变量。无论何时宿主值发生变化，表列表将随之调整。

注意，绑定变量的名称不能超过 30 个字符，也不能包括空格或者特殊字符。在变量中的空格将被转变为下划线。如果变量名是 `Owner of the table`，则绑定变量必须是 `:owner_of_the_table`。

还要注意如果一个变量值是空，依赖的列表也将是空。在这种情况下，没有查询被执行。

复选框选项（Checkbox option）

如果用户可以选择两个不同可能性中的一个，你可以使用复选框选项。这个选项需要跟随勾选和未选值：

```
select * from emp
order by hiredate &<name="Descending sort order"
checkbox="desc, asc">
```

即使你总要指定 2 个值，但其中一个值可以为空：

```
select * from emp
order by hiredate &<name="Descending sort order"
checkbox="desc,">
```

因为默认排序是升序，这 2 个例子是相同的。

前缀和后缀选项（Prefix and suffix options）

如果一个变量是空，这可能暗示 SQL 文字的某个固定部分必须也被忽略。考虑一个例子，假设你想提供一个选择排序字段，如果没有字段被指定，排序子句将被全部忽略。在这种情况下，你可以将这个文字指定为该变量的前缀或者后缀。如果变量值为空，前缀和后缀也将同样被忽略：

```
select * from emp
&<name="Descending sort column"
list="empno, ename, hiredate"
prefix="order by "
```

```
suffix=" desc">
```

如果变量没有被指定值，结果将同样是空：

```
select * from emp
```

如果一个字段被指定，结果将是：

```
select * from emp
order by empno desc
```

注意前缀后和后缀前的空格。这些很重要，因为前缀，值和后缀按指定紧密连接。

换码符（Escape character）

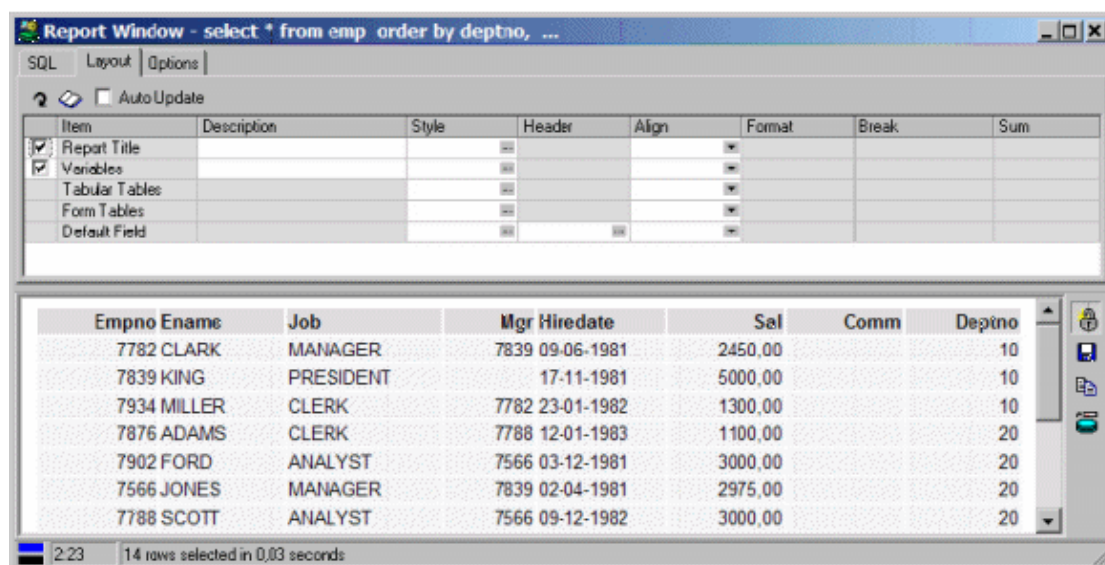
如果你想在 SQL 文本中使用一个&符号，并且不被认为是置换变量，用一个双&字符代替。下面例子将存取 R&D 部门的所有职工：

```
select * from emp
where emp.deptno in (select dept.deptno from dept
                    where dname = 'R&&D')
order by empno desc
```

如果使用'R&D'代替，你将被提示输入 D 变量。

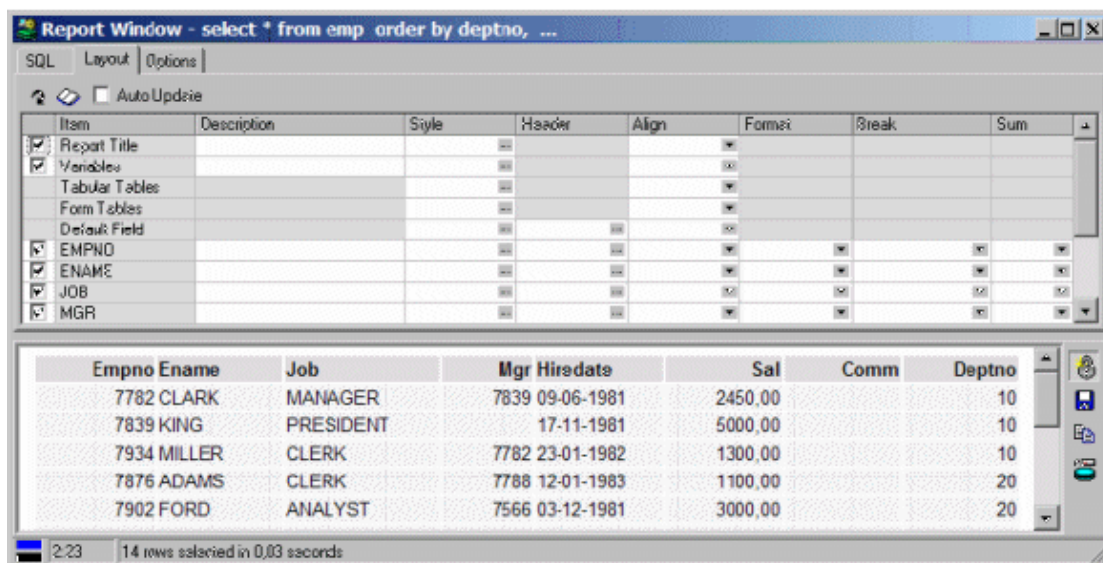
10.4 精制布局（Refining the layout）

打开一个新的报表窗口后，输入一个查询语句，执行报表，布局将使用默认风格属性。此风格有一个平面布局，标准的字体属性，table 风格，颜色等等。为精制标准布局，切换到布局（Layout）标签页。对于一个新报表，页显示如下：



你可以看到，你可以定义所有布局项目为不同布局属性（报表标题，变量，表，字段）。

如果你想定义字段属性，首先需要按工具条中的刷新字段列表（Refresh Fieldlist）键：



现在布局表格中包含查询语句结果集中的全部字段，你可以为每个字段设置布局属性。如果你不填写字段的样式（Style），标题（Header）或者排列（Align）属性，将使用相应的默认字段属性。

修改一个布局属性后，你可以再次执行报表查看对结果的影响。你也可以在工具条中启用自动更新（Auto Update）选项，在这种情况下每个改变将立刻反应到结果集上。

下面章节详细描述各布局属性：

显示（Displayed）

布局表格中最左侧的复选框列控制布局项是否被显示，这允许你禁止报表标题或者变量，并且允许你控制一个指定字段是否显示。

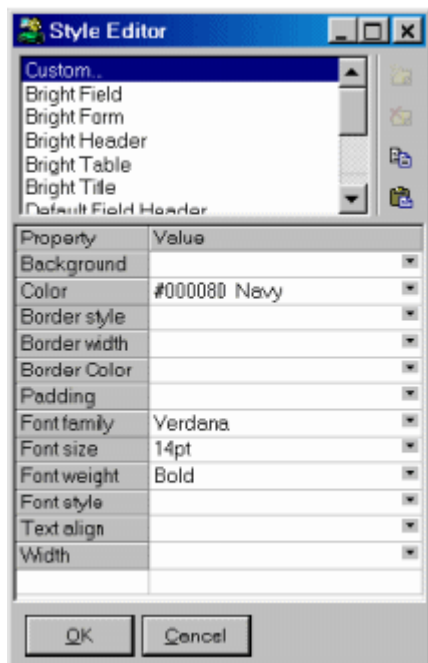
描述（Description）

报表标题的描述将显示在报表和每个分页的上部。变量的描述将显示在变量的上方。

对于一个字段，你可以定义一个描述覆盖标准字段名，这个描述将作为列的标题头。

样式（Style）

样式控制布局项的外观。按单元格按钮（...）显示格式编辑器：



在样式编辑器上方的列表框中你可以看到所有标准样式和一个自定义样式。你可以从样式库中选择一个标准样式（见样式库），或者你可以选择自定义样式，这样，你可以改变当前布局项的不同样式属性。

注意这个样式系统是基于Cascading Style Sheets (CSS)标准。你在样式编辑器中看到的属性是经常使用的，你也可以在标准属性下增加CSS属性和值。例如，你可以增加一个值是20pt高度的样式属性来设置单元格的高度为20像素。更多有关CSS信息见<http://www.w3.org/TR/REC-CSS2>。

拷贝和粘贴键允许你拷贝一个样式属性，因此可以快速地从已有样式获取一个新的样式。

如果你没有定义字段样式，将使用样式库中的默认字段样式，你可以通过定义默认字段样式覆盖它。你可以在字段级指定样式覆盖相应的默认格式。注意，每个字段标题头从默认标题头样式中继承它的样式属性，所以，你只需要覆盖你想修改的属性。例如，假设你定义了一个默认字段属性：Verdana，14像素，蓝色，并且你想显示 empno 列为 Verdana，14像素，红色，你只需要在 empno 级定义颜色。字体和字体大小将自动从默认字段级别中继承。

不同布局项的默认的样式在样式库中描述。

标题头 (Header)

就像你可以设置字段的数据样式一样，你也可以设置它们的标题头。如果你没有定义，将使用样式库中的默认标题头样式。你可以通过定义布局项的默认标题头样式覆盖它。你可以在字段级指定样式覆盖相应标题头的默认格式。注意，每个字段标题头从默认标题头样式中继承它的样式属性，所以，你只需要覆盖你想修改的属性。

对齐 (Align)

你可以使用以下方式对齐布局项：

- 左——项目在单元格中左对齐
- 右——项目在单元格中右对齐
- 中间——项目在单元格中居中对齐
- 默认——数字将右对齐，其他的左对齐
- 无——不指定对齐

注意，你可以为布局项设置文字对齐样式属性。这将优先于你在布局表格中指定的对齐选项。还需要注意对齐效果还依赖于布局项的宽度样式属性。

显示格式 (Format)

默认情况下，一个字段的显示格式是由数据类型，范围和精度控制。你经常使用计算字段或者汇总字段，在这种情况下不知道范围和精度，你可以指定一个显示格式。

显示格式属性只对日期和数字字段有效。对于数字字段你可以用一个 9 代表一个零制止数字，0 代表一个普通数字，G 代表千分符，D 代表一个小数点，E 代表科学计数法。你可以额外用双或者单引号指定文字字符。一些例子：

Format	Value = 1234	Value = 5	Value = 0.1
999G990D00	1,234.00	5.00	0.10
9G990	1,234	5	0
0000	1234	0005	0000
0"%"	1234%	5%	0%
0D000e+00	1.234e+03	5.000e+00	1.000e-01
0D000e-0	1.234e3	5.000e0	1.000e-1

对于日期字段，你可以用标准的 Windows 日期和时间显示格式。

中断 (Break)

中断布局属性允许你组织你的报表结果。假设你想显示部门和它们的职工。你可以指定以下查询：

```
select d.*, e.*
from   emp e, dept d
where  e.deptno = d.deptno
order by d.deptno, e.empno
```

这当然形成一个简单的表格报表：

Report Window - select d.*, e.* from emp e, dept ...

SQL | Layout | Options

```
select d.*, e.*
from emp e, dept d
where e.deptno = d.deptno
order by d.deptno, e.empno
```

Deptno	Dname	Loc	Empno	Ename	Job	Mgr	Hiredate	Sal	Comm
10	ACCOUNTING	NEW YORK	7782	CLARK	MANAGER	7839	09-06-1981	2450,00	
10	ACCOUNTING	NEW YORK	7839	KING	PRESIDENT		17-11-1981	5000,00	
10	ACCOUNTING	NEW YORK	7934	MILLER	CLERK	7782	23-01-1982	1300,00	
20	RESEARCH	DALLAS	7369	SMITH	CLERK	7902	17-12-1980	800,00	
20	RESEARCH	DALLAS	7566	JONES	MANAGER	7839	02-04-1981	2975,00	
20	RESEARCH	DALLAS	7788	SCOTT	ANALYST	7566	09-12-1982	3000,00	
20	RESEARCH	DALLAS	7876	ADAMS	CLERK	7788	12-01-1983	1100,00	
20	RESEARCH	DALLAS	7902	FORD	ANALYST	7566	03-12-1981	3000,00	
30	SALES	CHICAGO	7499	ALLEN	SALESMAN	7698	20-02-1981	1600,00	300,00
30	SALES	CHICAGO	7521	WARD	SALESMAN	7698	22-02-1981	1250,00	500,00
30	SALES	CHICAGO	7654	MARTIN	SALESMAN	7698	28-09-1981	1250,00	1400,00

4.27 14 rows selected in 0.03 seconds

通过在 LOC 列上指定一个 Break，你将得到更加结构化的报表：

Report Window - select d.*, e.* from emp e, dept ...

SQL | Layout | Options

☐ Auto Update

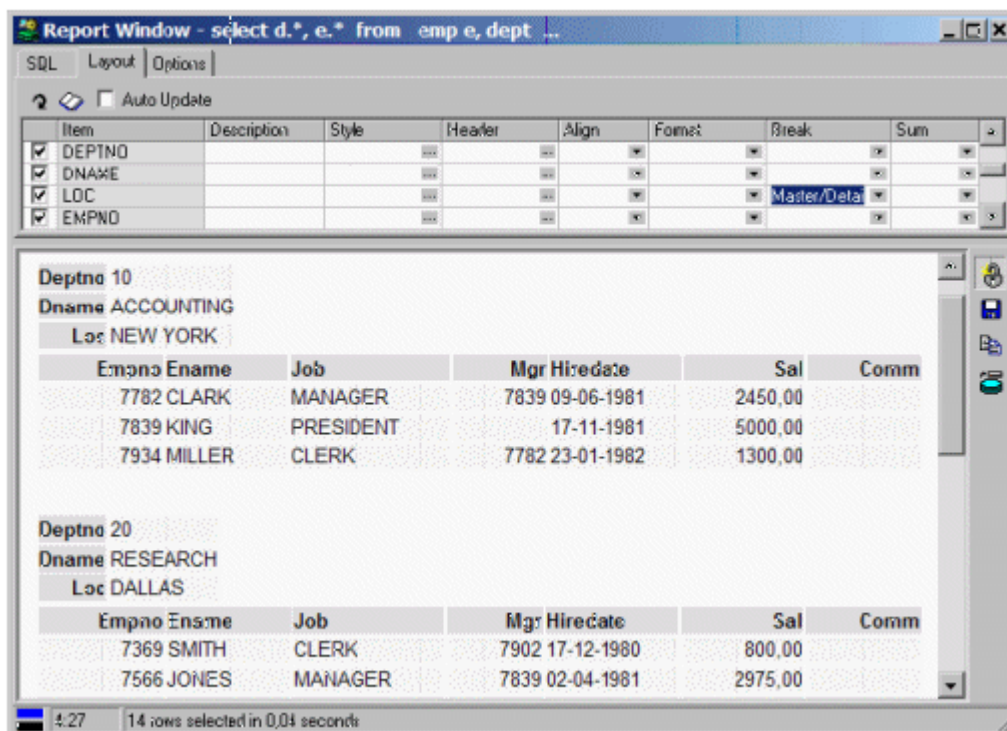
Item	Description	Style	Header	Align	Format	Break	Sum
<input checked="" type="checkbox"/> DEPTNO							
<input checked="" type="checkbox"/> DNAME							
<input checked="" type="checkbox"/> LOC						Break	
<input checked="" type="checkbox"/> EMPNO							

Deptno	Dname	Loc	Empno	Ename	Job	Mgr	Hiredate	Sal	Comm
10	ACCOUNTING	NEW YORK	7782	CLARK	MANAGER	7839	09-06-1981	2450,00	
			7839	KING	PRESIDENT		17-11-1981	5000,00	
			7934	MILLER	CLERK	7782	23-01-1982	1300,00	
20	RESEARCH	DALLAS	7369	SMITH	CLERK	7902	17-12-1980	800,00	
			7566	JONES	MANAGER	7839	02-04-1981	2975,00	
			7788	SCOTT	ANALYST	7566	09-12-1982	3000,00	
			7876	ADAMS	CLERK	7788	12-01-1983	1100,00	
			7902	FORD	ANALYST	7566	03-12-1981	3000,00	
30	SALES	CHICAGO	7499	ALLEN	SALESMAN	7698	20-02-1981	1600,00	300,00
			7521	WARD	SALESMAN	7698	22-02-1981	1250,00	500,00

4.27 14 rows selected in 0.03 seconds

从开始到（包括）LOC 的字段重复值将被禁止。简而言之：每个部门记录将只显示一次。在这种情况下，按部门排序是必须的。如果你想在每个中断后重复头，使用 Break + Header 选项。

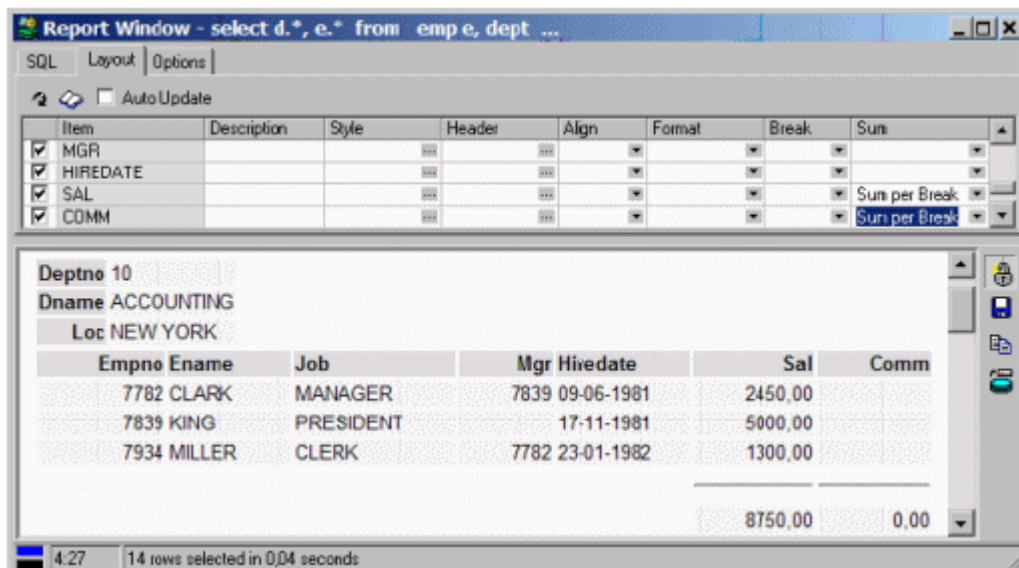
你也可以选择一个主/从布局：



你可以有多个中断在一个报表中。查询中的 order by 子句必须保证所有中断列必须在结果集中排序。

汇总 (Sum)

汇总属性允许你包含对一个字段的汇总：



你可以指定在中断、报表或两者的级别上汇总。

页标题 (Page headers)

要定义一个页标题，你首先需要设置页尺寸（见选项）。报表标题（Report Title）在分页后重复，并且可以包括下面变量：

- OSUser——操作系统用户

- DBUser ——连接数据库的用户。
- Database——用户连接的数据库。
- Date——当前日期。
- Time——当前时间。
- Page——当前页号

为在页标题中包含这些参数，之前使用与（&）。例如：

```
All employees (&dbuser@&database)
```

如果你运行报表时用 scott 连接 chicago 数据库，标题将如下面：

```
All employees (scott@chicago)
```

因为报表是 HTML 格式，你也可以包括 HTML 代码。为了包含一个图像，你可以包含 标签：

```
All employees <img src=logo.gif>
```

要包含左对齐、居中和右对齐信息，你可以使用一个单行表带 3 个单元，并且在相应单元包含信息：

```
<table width="100%"> <tr> <td>&dbuser</td> <td align="center">All Employees</td>
<td align="right">&page</td> </tr> </table>
```

这将显示为：

Scott	All Employees	1
-------	---------------	---

注意你可以使用这些变量和 HTML 代码在所有布局项的描述中，但是他们在报表标题中更有用。

10.5 样式库（The Style Library）

替代重复来重复去地定义和再定义同样的样式，你可以在工具条中按样式库（Style Library）键创建和修改你的标准样式。例如，如果你想包含 SQL 表达式的全部字段显示为 Courier New, 12 像素，蓝色，简单的在库中定义一个带这些属性的 SQL 样式并适当地应用它。

这里有许多有一个特殊意义的样式名：

样式名	描述
Default Report Title	将被应用到报表标题
Default Variables	将被应用到变量
Default Tabular Tables	将被应用到全部表格表（例如明细记录）
Default Form Tables	将被应用到全部表单表（例如主记录）
Default Field Style	将被应用到全部没有定义样式的字段

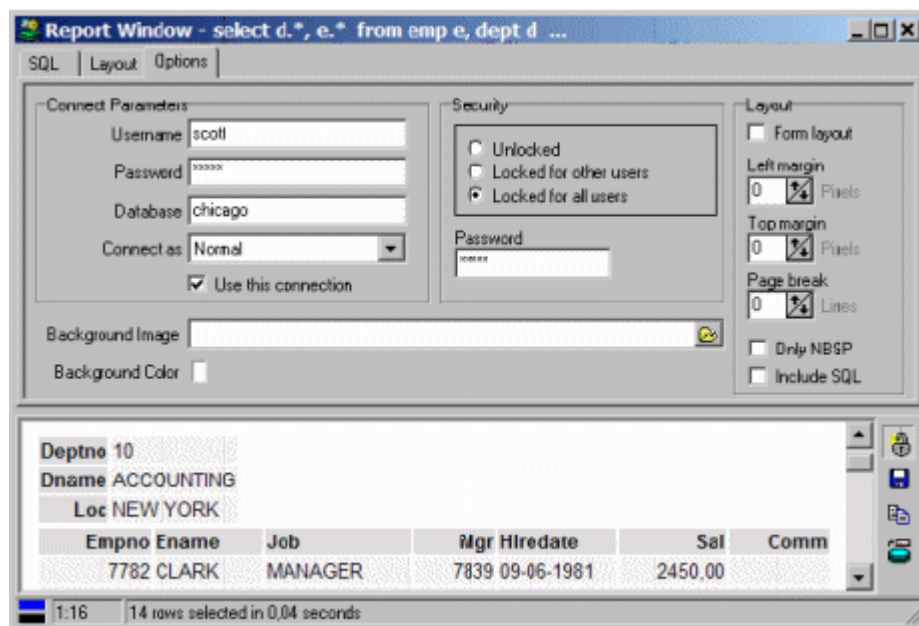
Default Field Header	将被应用到全部没有定义样式的字段标题
----------------------	--------------------

这些是默认样式，所以你可以在一个独立报表中通过定义相应的布局项样式覆盖它们。

注意全部使用的样式（标准和自定义）都保存在报表文件中。当一个报表运行并且需要应用一个样式时，PL/SQL Developer 将试着在样式库中查找它。当它无效，使用报表中的样式。这将确保你的报表将总是正确显示，但是那个样式同样可以被否决。

10.6 选项（Options）

你可以为你的报表指定不同选项：



连接参数（Connect Parameters）

连接参数决定如何运行报表查询。例如，如果你的报表必须在 SYS 账户下执行，你可以在用户名域中输入这个名字。你也可以输入密码，或者不输入。如果你没有输入密码，在报表执行时将提示输入密码。如果报表总是在同一个数据库实例中执行，你也可以指定数据库。你也可以在数据库域中输入*，表示报表将在当前数据库中执行。

使用这个连接（Use this connection）选项允许你禁止或者允许连接参数应用到一个报表。如果被禁止，使用当前 PL/SQL Developer 连接。

安全（Security）

安全选项允许你锁定报表定义（SQL、布局 and 选项），使得用户不能查看和修改它。如果连接参数指定一个带密码的有特权账户（如 SYS）时非常有用。如果一个用户可以修改 SQL，如果报表又没有锁定，他或她可以在 SYS 账户下执行任何语句。

安全可以设置为以下值之一：

- 不锁定——任何人可以修改报表定义。

- 锁定其他用户——只有知道报表密码才能修改报表定义，或者你使用由连接参数指定的账户连接 PL/SQL Developer.
- 锁定所有用户——不论你现在如何登录 PL/SQL Developer，在修改报表定义前，你总被提示输入报表密码。

如果报表被锁定，报表文件(.rep)被加密，所以，文字编辑器不能查看和修改。

布局选项 (Layout options)

- 表单布局 (Form layout) ——默认情况下，除了主记录在表单样式中显示，所有其他记录以表格布局显示，启用这个选项，所有的记录以表单布局显示。这对于每个记录有很多和/或宽字段不能在一行显示时很有用。
- 左边界 (Left margin) ——报表左边界的像素数。这个选项只有在使用 Internet Explorer 作为 HTML 浏览器时才起作用。
- 上边界 (Top margin) ——报表上边界的像素数。这个选项只有在使用 Internet Explorer 作为 HTML 浏览器时才起作用。
- 分页 (Page break) ——多少行以后开始分页。在分页后，报表头将重复。
- Only NBSP——当允许时，数据中所有空格在 HTML 输出时将被转化成 NBSP (non-breaking spaces)。当禁止时，使用通常的空格。这将影响布局，因为 HTML 浏览器将连续空格视作一个空格。

包含 SQL (Include SQL) ——当允许时，报表的 SQL 文本将包含在 HTML 输出中。

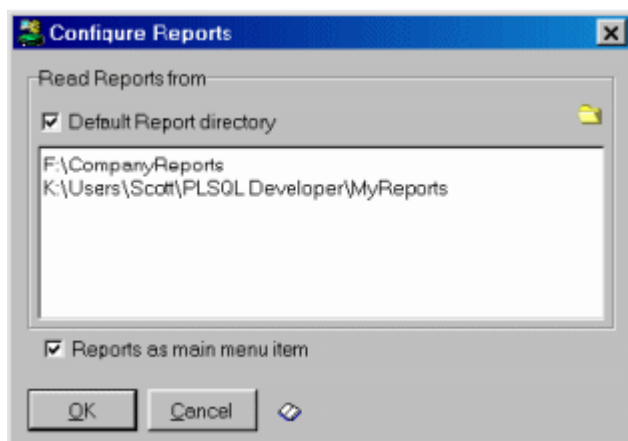
其他选项 (Other options)

- 背景图形 (Background image) ——图形 (.jpg or .gif) 可以显示在报表的背景中。你可以使用选择文件 (Select file) 键查找你的图形文件。
- 背景颜色 (Background color) ——报表要显示的背景颜色。

10.7 报表菜单 (The Reports menu)

默认情况下报表菜单项在主菜单中，只包括标准的报表。这些标准报表存放在 PL/SQL Developer 安装目录下的报表目录下。菜单只简单地反映了目录结构。从报表菜单中选择一个菜单项将执行那个报表。

你可以在目录中增加或者删除报表。但是你也可以通过工具菜单中的配置报表 (Configure Reports) 菜单项配置报表菜单。这将弹出一个对话框：



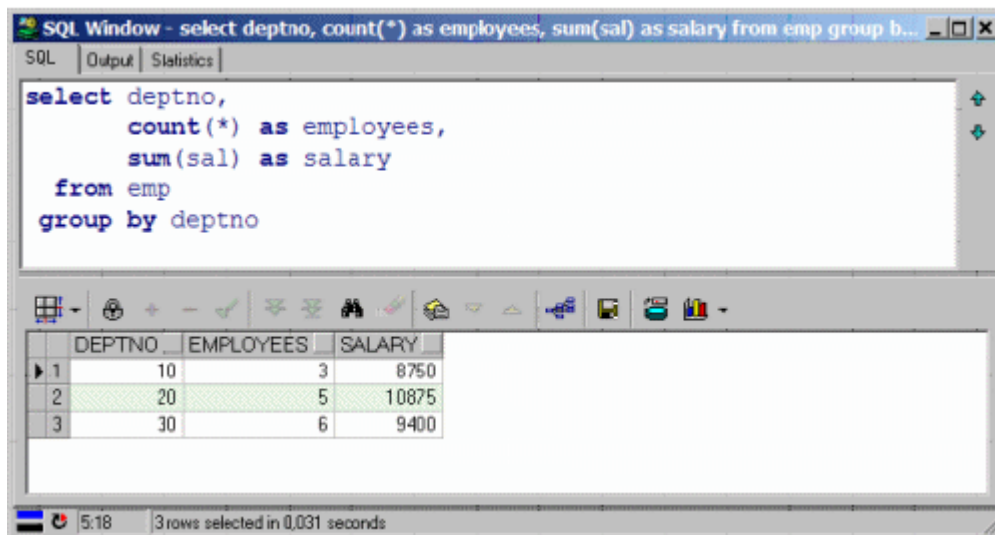
默认报表目录 (Default Report directory) 选项指出是否报表菜单包含 PL/SQL Developer 安装目录下的 Reports 子目录。这个目录包含标准报表。选项下面的编辑区是菜单所包含的目录列表。你可以手工输入目录，或者你可以按右侧的增加目录键选择一个目录。

注意菜单文件和子目录将被合并并在报表菜单中。你应该在目录列表中防止重复目录和报表文件，否则，在报表菜单中你将得到不明确的菜单项。

样式库也可以从报表配置对话框中获得（见样式库）。

11 图形 (Graphics)

为了快速地显示一个在 SQL 窗口或者报表窗口查询出的字段数据的图形展现，你可以使用图形窗口，考虑在 SQL 窗口中的以下查询：

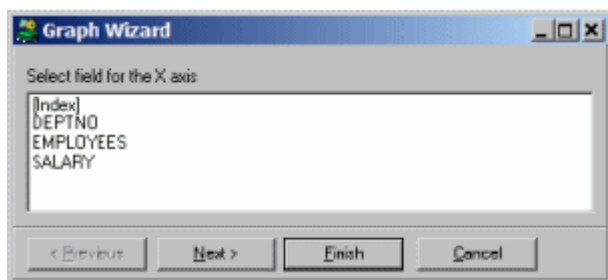


要看一个部门工资的图形展现，你可以按图形键：

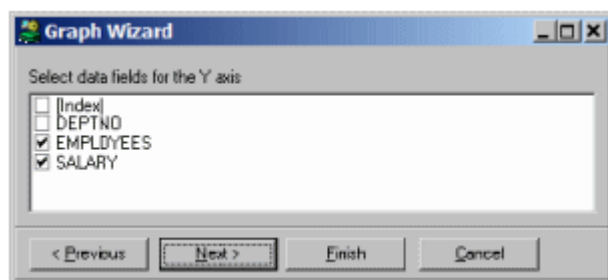


在报表窗口中也有同样的键。这可以开始一个下面描述的图形向导，或者你可以选择一个事先定义好的图形版式或者模板，并且相应的字段将被选择在 X 和 Y 轴上。

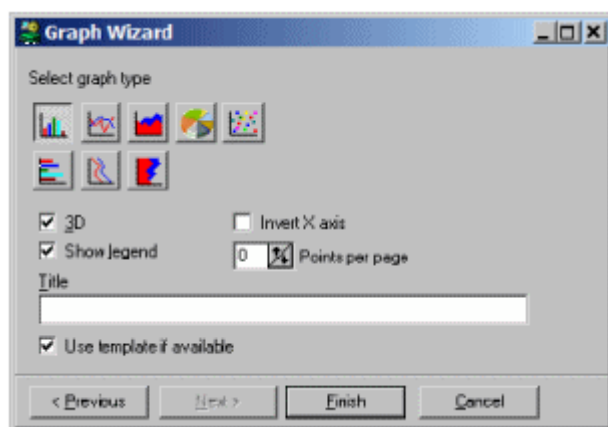
图形向导首先让你选择 X 坐标数据：



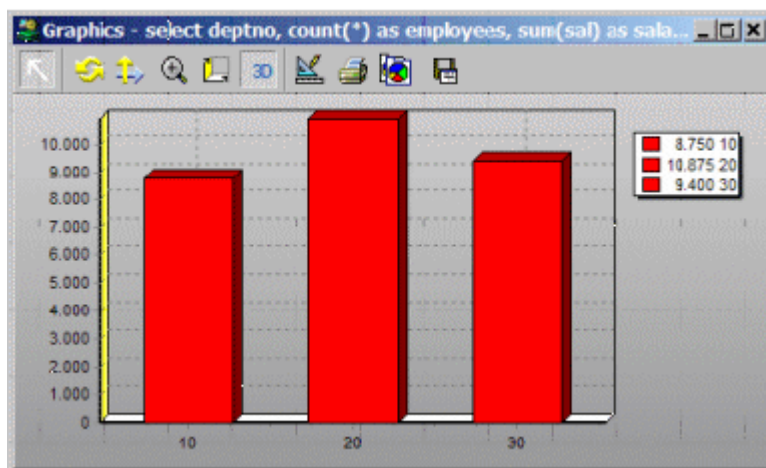
这显示所有字段和索引项，对于没有 X 坐标值但又想按顺序显示 Y 坐标数据时非常有用。在本例中，我们用部门号显示工资，所以我们选择 DEPTNO 按下一步键选择 Y 坐标的数据：



选择 SALARY 作为 Y 坐标后按下一步选择图形样式属性：



按完成后，图形窗口的结果将被显示：



图形窗口的工具条提供一下功能：

旋转（Rotate）——按鼠标左键移动鼠标沿着 X 坐标和 Y 坐标旋转图形。

移动（Move）——按鼠标左键移动鼠标在窗口中移动图形。当按鼠标右键时这个功能总是有效的，不论工具条在什么功能状态下。

缩放（Zoom）——按鼠标左键移动鼠标放大和缩小图形。

深度（Depth）——按鼠标左键移动鼠标增加或者减少图形深度。

三维（3D）——图形二维和三维的开关。

编辑（Edit）——编辑图形数据和布局。

保存成模版（Save as template）——为以后图形基于相同的字段名，你可以将图形的布局保存成模版。如果你以后创建一个图形，从图形键的弹出菜单中选择模版，就可以使用模版中的布局信息。

你也可以按相应的工具条按键打印图形或者将图形拷贝到粘贴板。

12 工程 (Projects)

你可以使用 PL/SQL Developer 内建的工程概念组织你的工作。一个工程包括一组源文件，数据库对象，注释和选项。允许你在一个指定项目集中工作，而不是全部的数据库或者纲要（schema）。这样可以很容易地查找你需要的工程项，编译全部工程项目，或者移动一个工程从一个位置或者数据库到另外一个。

一个工程将跟踪你桌面设置。如果你打开一个工程，所有的项目将按你之前关闭时的位置打开，这要求自动保存桌面（AutoSave Desktop）参数必须打开。这个信息没有保存在工程定义文件中（Project.prj），而是保存在工程桌面文件中（Project.dsk）。

注意：使用项目必须允许使用工程（Use Projects）参数。如果不是这样，主菜单将不包含工程菜单项。对于所有的工程菜单项都有对应的工具条按键，可以通过工具条参数页增加相应的工具条按键（见用户界面—工具条）。

12.1 保存一个工程 (Saving a project)

设置选项后，你可以通过从工程菜单中选择保存 (Save) 菜单项保存工程。这将创建一个以 .prj 为扩展名的工程文件。工程文件包括项目描述，选项和注释。所以在增加或者移除项目、修改选项或者修改注释后都要保存工程。

保存一个工程并不保存工程中的源文件。

12.2 增加文件到工程 (Adding files to a project)

有 2 个方法增加一个文件到你的工程：

如果文件已经在集成环境中打开，你只要从工程菜单中选择增加到工程 (Add to Project) 菜单项。对于源程序文件 (包含 PL/SQL 代码的程序单元)，项目隐含可以编译。这意味着当你制造 (make) 或者构造 (build) 工程时，这个文件将包含在编译过程中。其他文件，如测试脚本，SQL 脚本，报表文件等等将不能编译。

如果文件没有在集成环境中打开，你可以在工程项目工具条中按增加文件 (Add file(s)) 键并选择增加的文件。

注意如果可能，文件目录是相对于工程文件的路径，例如，如果文件保存在工程文件所在目录，路径不被保存。如果文件保存在工程文件下的子目录中，只有子目录被保存。如果文件保存在其他位置，全路径将被保存。这使工程容易‘轻便地’从一个目录到另外一个目录。

12.3 增加数据库对象到工程 (Adding database objects to a project)

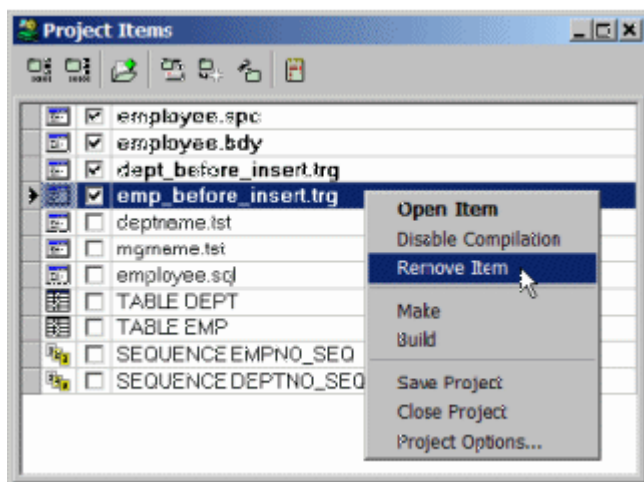
为增加一个数据库对象到工程，在对象浏览器中右键点击它，从弹出菜单中选择增加到工程 (Add to Project) 菜单项。就象程序文件一样，一个 PL/SQL 程序单元将隐含可以编译，其他对象，如表和序列不能编译。

你也可以在对象浏览器中使用拖动/放下一个数据库对象到工程项目窗口 (见下)。

如果当前用户拥有这个数据库对象，宿主将不被保存在工程文件中，如果其他用户拥有这个对象，宿主将作为对象描述被保存。这使工程很‘轻便地’从一个用户到另外一个用户。

12.4 用工程项目工作 (Working with project items)

增加项目到工程后，你可以从工程 (Project) 菜单中选择工程项目 (Project Items) 来工作。工程项目窗口看起来应是：



前面 4 个程序文件可以编译。2 个测试脚本，SQL 脚本、表和序列不能编译。

你可以保持工程项目窗口在打开状态，可以快捷地访问全部的工程项目。这个窗口是工程桌面的一部分，在重新打开工程时重新打开它。

要编辑一个项目，你可以简单地双击它，如果项目是一个代码源文件，它将在相应的窗口中被打开（程序窗口、测试窗口等等）。如果项目是一个数据库对象，带有对象定义的相应编辑器将被打开。如果一个项目是一个外部文件，相应的程序将被载入（如一个 PDF 文件使用 Acrobat Reader）。

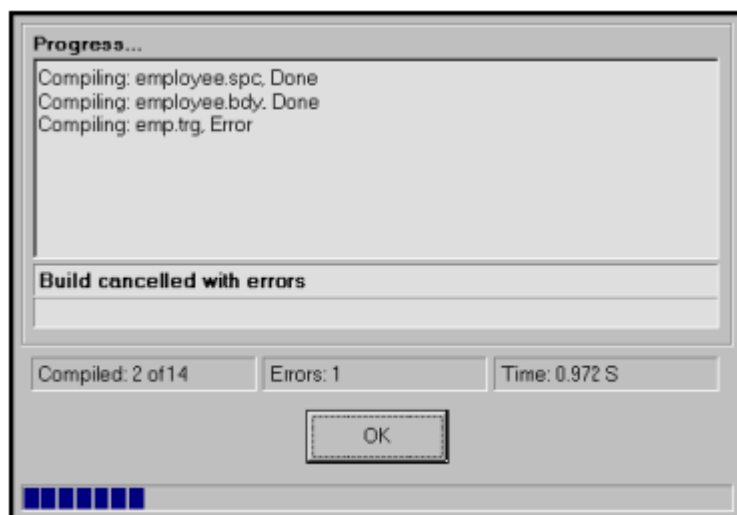
你也可以右键点击一个项目，象上面显示的会出现一个弹出菜单。你可以允许/禁止项目编译，或者移除工程。如果一个项目是数据库对象，将作为一个弹出菜单的子菜单。

为修改工程项目的顺序，你可以点击行头拖动到新位置。顺序不仅仅对显示重要，还决定编译的顺序。

12.5 编译一个工程（Compiling a project）

为编译一个工程，你可以从工程菜单中选择 **Build** 或者 **Make** 菜单项。

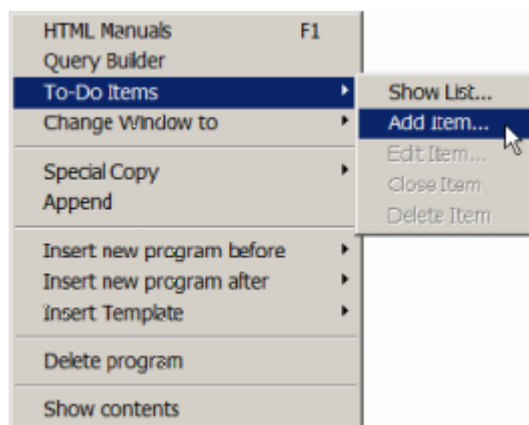
Build 功能将只编译工程中所有可以编译的项目。项目将按在工程项目列表的显示顺序编译。如果一个项目失败，第一个错误后暂停（Halt after first error）的工程选项将决定是否继续编译：



Make 功能只编译从上次编译后改变的项目。如果一个 PL/SQL 程序单元作为一个数据库对象，并且它在数据库中是无效的，它也被编译。编译的时间戳被保存在工程的桌面文件中。项目允许编译但还是新的（are up-to-date），将在编译进程窗口中被列为可跳过的。

13 创建一个 To-Do 项目 (Creating a To-Do Item)

为创建一个 To-Do 项，你首先需要将光标定位到文件的指定位置，这里将插入一个注释，表示在这附近将有重要工作要做，下一步你可以右键点击这个位置选择弹出菜单中的 To-Do Items 子菜单中的增加菜单项：



将出现上面说到的 To-Do 项目编辑器，你可以为新的 To-Do 项目输入信息。按确认键后，注释将在光标所在处插入。

13.1 编辑一个 To-Do 项目 (Editing a To-Do Item)

为编辑一个已存在的 To-Do 项目，你既可以在源文件中右键点击相应注释从 To-Do Items 子菜单中选择编辑（Edit）菜单项，也可以在你远离要编辑的 To-Do 项目时选择显示列表（Show List）菜单项，这样你可以从列表中选择 To-Do 项目后按编辑键。

现在你可以修改 To-Do 的不同属性，按确认键更新源文件的注释。

13.2 关闭一个 To-Do 项目 (Closing a To-Do Item)

通过右键点击源文件中的注释，从 To-Do Items 子菜单中选择关闭菜单项关闭一个 To-Do 项。

14 窗口、数据库会话和事务 (*Windows, database sessions and transactions*)

PL/SQL Developer 提供了一个多会话/多线程环境。

多会话的意思是即使你只登录一次 PL/SQL Developer，隔离的数据库会话可以用在测试窗口、SQL 窗口和命令窗口。此外，程序单元也将在一个隔离的数据库会话中编译。毕竟一个编译是一个 DDL 语句，将隐含提交当前事务，如果和其他窗口共享同一个会话将出现不必要的影响。

多线程的意思是在测试窗口、SQL 窗口和命令窗口的执行可以同时运行，不需要相互调动。好处是当长时间运行 SQL 语句时可以继续做其他工作，如你可以中断或者杀一个会话，你可以测试程序单元的锁定行为等等。

14.1 会话模式 (Session mode)

多线程暗示你需要一个多会话环境，毕竟 Oracle 服务器在给定的数据库会话中一次只能执行一个语句。所以必须在不同的会话中才能同时运行。不是所有的开发环境都允许无限制地使用会话，所以，有一个参数设置（见 Oracle-选项）控制 PL/SQL Developer 可以使用的会话数。这个参数叫会话模式，可以在下面值中选择一个：

多会话——每个测试窗口、SQL 窗口和命令窗口将有自己的会话，并且编译使用其他会话。这是最灵活的设置，很明显将使用最大量的数据库会话。另外一个缺点是更新窗口 X 只有在提交后才能在窗口 Y 中看到。

双会话——测试窗口、SQL 窗口和命令窗口将共享一个会话，编译使用另外一个会话。这个模式的缺点是一次只有一个窗口可以执行。

单会话——所有窗口和编译都使用同一个会话，使得事务管理很困难。在这种模式下不能调试。如果你想限制必须使用一个数据库会话时才使用这个设置。

14.2 在多会话或双会话模式下执行 (Execution in Multi session or Dual session mode)

当你在多会话或双会话模式下使用一个测试窗口、SQL 窗口或命令窗口执行时，在工具条中的中断 (Break) 键将可以在任何时间中断执行。中断不一定全部成功，例如在锁等待状态下。如果你有足够的权限，你可以借助在会话菜单中的 Kill 菜单项。

如果你将鼠标光标移动到一个正在执行窗口，它将改变形状提示正在运行，不能修改窗口中的任何内容。

执行结束后，如果开启了一个事务，提交和回滚键将可以使用。在多会话中，每个窗口有自动的会话，因此也拥有自己的事务。提交和回滚键将反映最上层窗口的事务状态。

15 浏览对象 (Browsing objects)

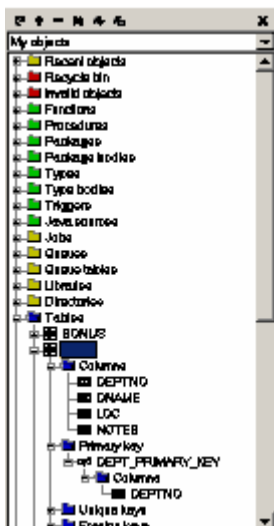
当你在 Oracle 数据库中开发一个程序单元，它总会存取其他数据库对象。毕竟如果不是这样，程序单元就没有在数据库存在的必要。要写这样一个程序单元，你必须要知道其他数据库对象是如何定义的。因此，下面所列的问题对于数据库开发者来说十分熟悉：

- 这个字段的数据类型是什么？
- 表的主键是什么？
- 包函数的确切名称是什么？
- 参数的默认值是什么？
- 触发器执行的前提条件是什么？
- 哪些对象调用我正在工作的函数？
- 等等...

所有这些信息都在 Oracle 数据字典中，并且可以使用 PL/SQL Developer 的浏览器获得。

15.1 使用浏览器 (Using the Browser)

你可以通过在工具菜单选择浏览器菜单项使浏览器有效。浏览器显示和开发程序单元有关的所有数据库基础对象类型。你可以象在 Windows 的资源管理器中操作目录一样通过这个树形视图操作。打开表文件夹显示从当前会话获取的所有表。打开一个指定表文件夹显示这张表的所有相关属性。打开一个表下的外键文件夹显示表的所有外键，等等。近期对象文件夹包含你最近在 PL/SQL Developer 中使用的全部对象。



每当你打开一个文件夹，为使随后的访问尽可能快速，从字典中查询出的信息被读入内存中。在某些东西改变后要强制浏览器重新查询信息，你需要选择修改的对象，按窗口左上方的刷新（Refresh）键。你也可以从弹出菜单选择相应的菜单项刷新一个对象。要刷新整个浏览器，按刷新键的同时按 **Ctrl** 键。

你可以从对象浏览器拖动和放下对象到工作区。这将调出一个带对象定义的编辑器。你可以设置一个参数准确控制要执行的功能（见用户界面-浏览器）。

你也可以从对象浏览器中拖动和放下一个对象到一个编辑器中。这将调出一个弹出菜单，菜单内容依据于对象。你可以选择拷贝名称（如果需要，前面带宿主）、属性、描述和其他相关文本到编辑器，位置将在你放下它的地方。如果你拖动和放下一个文件夹到编辑器中，文件夹中全部对象名将被拷贝。这些对象名被用逗号分开。你可以设置一个参数控制对象名放置在一行或者多行（见用户界面-编辑器）。

在一个对象图表上按鼠标右键，出现一个上下文敏感的弹出菜单，允许你在这个对象上执行其他功能。这个弹出菜单中的功能在以下段落中描述。

你可以在选择时使用 **Control** 或者 **Shift** 键选择多个对象。如果多个对象被选中，对象弹出菜单将只显示对于全部选中对象有效的条目，并且拖动和放下功能将同时在所有对象上起作用。

你可以按 **Ctrl -**和 **Ctrl +**快速排列浏览器的宽度以适合内容。

浏览器的参数（见用户界面-浏览器）允许你定义哪些文件夹可见和显示的排序。它还允许你定义在最近对象文件夹中可见的最大对象数。今后你可以设置在一个浏览器对象上双击的动作。默认情况下，这将展开相应的节点，但是你也可以定义这将调出一个对象定义编辑器或者属性窗口。

创建一个新对象 (Creating a new object)

要创建一个新的程序单元、表、视图、序列或者同义词，在一个同类对象上或者在这个对象类型的根文件夹上右键点击，从弹出菜单中选择新建（New）菜单项。

使用程序单元的新建功能将创建一个新的程序窗口，初始内容来自一个指定的模版，描述见模版（Templates）。

视图可以在一个 SQL 窗口中创建，初始内容来自一个指定模版，就像程序单元一样。

表、序列和同义词将显示在一个指定的编辑器窗口。表、序列和同义词的编辑器在创建和修改非 PL/SQL 对象中详细描述。

复制一个对象（Duplicate an object）

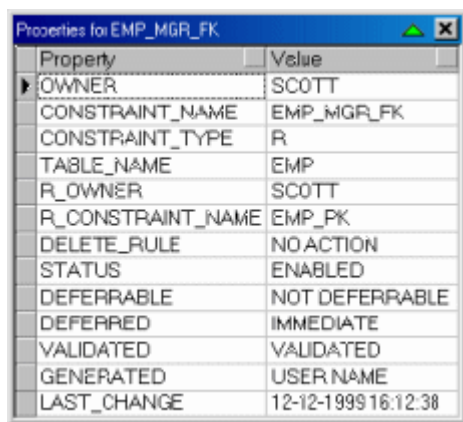
你可以通过弹出菜单中选择复制（Duplicate）菜单项复制一个表、序列或者同义词。这将调用一个除名称外，带所选对象所有属性的编辑器窗口。这对于当你想创建一个相似表来做一些测试时很有用。

拷贝一个对象或者文件夹（Copying an object or folder）

为拷贝一个对象或者在一个文件夹下的对象名称，选择拷贝以逗号分割(Copy comma separated)菜单项。这将把名称放到粘贴板中。你可以设置一个参数控制对象名被放置一行或者多行（见用户界面-编辑器）。

查看对象属性（Viewing object properties）

为查看一个指定对象的属性，从弹出菜单中选择属性（Properties）菜单项。你看到的实际属性依赖对象类型和你的 Oracle 服务器版本。



Property	Value
OWNER	SCOTT
CONSTRAINT_NAME	EMP_MGR_FK
CONSTRAINT_TYPE	R
TABLE_NAME	EMP
R_OWNER	SCOTT
R_CONSTRAINT_NAME	EMP_PK
DELETE_RULE	NOACTION
STATUS	ENABLED
DEFERRABLE	NOT DEFERRABLE
DEFERRED	IMMEDIATE
VALIDATED	VALIDATED
GENERATED	USER NAME
LAST_CHANGE	12-12-1999 16:12:38

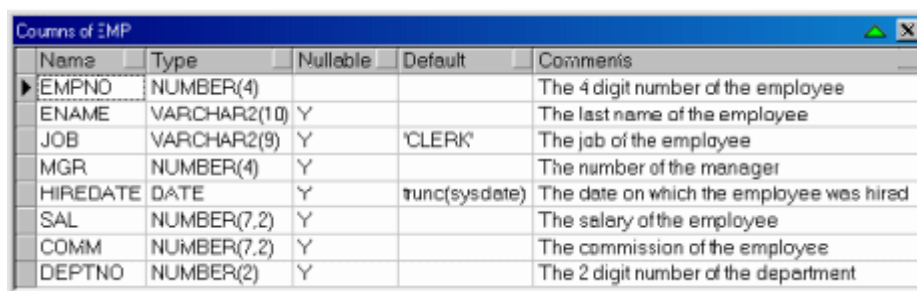
属性被显示在一个模式窗口中，你可以放置在屏幕的任何地方。你可以保持属性窗口可见（总是顶层窗口）并且继续编辑其他窗口。你也可以按窗口右上方的按键卷起和放下窗口。这样可以方便地给其他东西让出空间并且保持属性可用。

查看对象描述（Viewing object descriptions）

表、视图、函数、过程、包、类型和方法可以用和 SQL*Plus 相似的方法描述，在弹出菜单中选择描述（Describe）菜单项，描述将显示在一个模式窗口中，和前面提到的属性窗口一样。

对于表和视图：全部字段和它们各自的属性显示在一个属性窗口中；对于函数、过程和方法：参数和它们的属性被显示；对于包：函数、过程、参数、类型、变量、常数和异常

被显示；对于类型：方法和属性被显示。



Name	Type	Nullable	Default	Comments
EMPNO	NUMBER(4)			The 4 digit number of the employee
ENAME	VARCHAR2(10)	Y		The last name of the employee
JOB	VARCHAR2(9)	Y	'CLERK'	The job of the employee
MGR	NUMBER(4)	Y		The number of the manager
HIREDATE	DATE	Y	trunc(sysdate)	The date on which the employee was hired
SAL	NUMBER(7,2)	Y		The salary of the employee
COMM	NUMBER(7,2)	Y		The commission of the employee
DEPTNO	NUMBER(2)	Y		The 2 digit number of the department

你可以在描述窗口中选择多行、多列或者单元格的范围。为选择多行或者多列，只要在行标题或者列标题点击后拖动鼠标指针高亮选择。要选择一个指定范围的单元格，移动鼠标指针通过单元格左边界直到指针形状发生改变，按鼠标键并拖动鼠标高亮选择。选中可以拷贝或者打印。做一个用逗号间隔的选中单元格拷贝，在窗口中右键点击，从弹出的菜单中选择相应的菜单项。为导出描述到一个 CSV, TSV, HTML 或 XML 文件，从导出子菜单中选择相应菜单项。为打印描述窗口内容，选择打印菜单项。

在描述窗口中双击一个单元格，单元格的内容将被拷贝到最顶层的编辑器窗口中。

为在描述窗口中排序行，点击在你想排序的列标题按键。

查看一个对象定义 (Viewing an object definition)

所有对象类型都有一些定义。为查看这个定义，你可以从弹出菜单中选择查看 (View) 菜单项。这个菜单项对于全部程序单元、视图、表、序列和同义词都有效。如果你对一个表元素，如一个字段或者索引选择查看 (View) 菜单项，将显示表定义，并且导航到选择的元素。

程序单元将显示在一个程序窗口，视图将显示在 SQL 窗口，表、序列和同义词将显示在一个特殊编辑器窗口。表、序列和同义词编辑器的详细描述见创建和修改非 PL/SQL 对象 (Creating and modifying non-PL/SQL objects) 章节。

编辑一个对象定义 (Editing an object definition)

你可以通过弹出菜单选择编辑 (Edit) 菜单项来编辑直接来自数据库的一个对象定义。你可以通过一个参数禁止这个特性，详见 Oracle-选项 (Oracle-Options) 章节。

重新命名一个对象 (Renaming an object)

表、视图、序列和同义词可以通过选择弹出菜单中的重新命名 (Rename) 菜单项重新命名。如果你正在使用的是 Oracle9.2 或者更高版本，你也可以重新命名字段。一个带有原名称对话框出现，你可以修改和应用。

删除一个对象 (Dropping an object)

有时需要删除一个对象。它可能已被废弃，或者必须从一个类型修改为另外一个（如从函数修改为过程）。为实现删除，只要从弹出菜单中选择删除 (Drop) 菜单项。

浏览一个相关对象 (Browsing a related object)

大部分对象都有引用和被引用对象。通过选择浏览（Browse）菜单项，你可以在浏览器中快速定位一个相关对象。

重编译一个对象（Recompiling an object）

对象经常引用数据库中其他对象。每当这些被引用对象修改，引用对象可能变为无效。在这种情况下，这个对象的图标上显示一个红色标记。为重新编译一个无效对象，从弹出菜单中选择重新编译（Recompile）菜单项。如果编译成功，标记消失，如果编译失败，你可以从弹出菜单中选择查看（View）菜单项查看错误。

增加一个对象代码到一个编辑器（Adding an object source to an editor）

可以通过选择增加源代码到编辑器（Add source to editor）菜单项将一个程序单元的源代码文件增加到一个程序窗口或者测试窗口。

对于一个程序窗口，源代码将放在当前程序单元后增加的一个新页中。这样，你可以创建一个程序文件包括已在数据库中存在的程序单元。

对于一个测试窗口，源代码将被增加到一个调试阶段的当前选择代码之后。这样，你可以在测试窗口能够包含一个源代码，随后在实际要单步进入程序单元前设置或者移除中断。

增加调试信息到一个对象（Adding debug information to an object）

如果你想在调试阶段查看或者修改变量，包含这些变量的对象必须包含调试信息。你可以通过选择增加调试信息（Add debug information）菜单项手工增加调试信息。这个菜单项有一个检查记号提示是否对象已经包含了调试信息。调试器参数部分包含一个选项当一个程序单元编译时自动增加调试信息，这样，只有不在开发环境中的程序单元才需要使用这个菜单项。

重新编译引用对象（Recompiling referencing objects）

当你改变一个被其他程序单元或者视图引用的对象，这些引用对象将变为无效。重新编译它们将使它们有效或者发现任何不兼容的问题，这可以很容易地通过选择重新编译引用对象（Recompile referencing objects）菜单项完成。这将调出一个即将被编译的无效对象窗口。更多有关这个工具的信息可以在编译无效对象（Compile Invalid Objects）章节中找到。

测试一个程序单元（Testing a program unit）

为测试一个程序单元（一个独立函数或者过程，一个包函数或者过程，或者一个对象类型方法），你可以使用测试脚本，这个在测试程序（Testing programs）章节中详细讲解。一个测试脚本通常包含一个简单的函数或者过程调用。你需要设置输入参数和在执行测试脚本后查看返回参数和函数结果。从浏览器中你可以选择测试（Test）菜单项创建这个测试脚本。

查询表和视图（Querying tables and views）

为查询一个表或者视图中的数据，从弹出菜单中选择查询数据（Query data）菜单项。一个 SQL 窗口被创建，包含在浏览器中选择的表或者视图的查询语句。如果你允许自动执行 SQL 窗口(AutoExecute SQL Window)参数，查询语句将被立即执行。

编辑表数据（Editing table data）

如果你想编辑一个表中的数据，当表已在浏览器中选，你可以选择编辑数据(Edit data)菜单项。一个 SQL 窗口被创建，带一个含 rowid 的查询语句。如果你允许自动执行 SQL 窗口(AutoExecute SQL Window)参数，查询语句将被立即执行。在获得记录后，SQL 窗口立即进入编辑状态。

导出表（Exporting tables）

为导出一个表的定义和数据，选择导出数据(Export data)菜单项。这将调出选中表的导出表工具。详见导出表(Export Tables)章节。

允许和禁止触发器和限制（Enabling and disabling triggers and constraints）

触发器和限制可以从对象浏览器中通过在弹出菜单中选项启用或者禁用菜单项被启用或者禁用。

禁用的触发器和限制被灰色显示。

执行一个工作（Running a job）

对于工作，通过 dbms_job 包提交，你可以从弹出菜单中选择执行（Run）菜单项显示执行。

启用和禁用排队和出队（Enabling and disabling enqueueing and dequeuing）

对于队列，你可以从弹出菜单中选择相应的菜单项启用或者禁用排队和出队操作。一个在菜单项前检查标记提示该操作当前是否启用。

增加一个对象到一个用户定义的文件夹（Adding an object to a user defined folder）

为增加一个对象到用户定义的文件夹（见浏览器文件夹），选择增加到文件夹(Add to folder)菜单项，随后从子菜单中选择一个文件夹。

恢复或者清除一个对象（Restoring or purging an object）

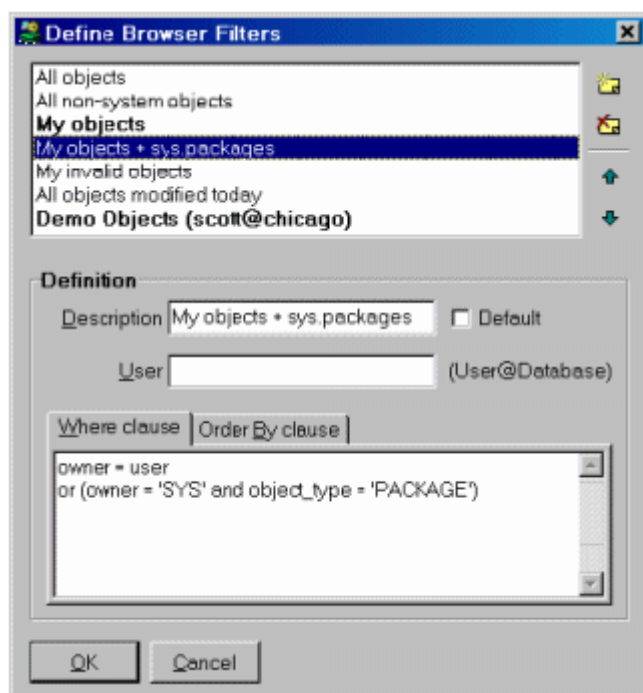
如果你使用 Oracle10g 或者更高版本，你可以恢复或者精炼一个原已删除表，如果这个特性在数据库中被启用。从回收站文件夹中的一个对象你可以选择清除（Purge）菜单项永久删除表和它的数据，或者选择恢复（Restore）菜单项在原始名称或者在新的名称下恢复。

15.2 浏览器过滤器（Browser Filters）

你可以定义过滤器决定在对象浏览器的系统文件夹中哪些对象可见，和以什么顺序显示这些对象。在浏览器的上方有一个查询列表允许你快速地在不同过滤器中切换。

在工具菜单中选择浏览器过滤器(Browser Filters)菜单项或者按浏览器上方的过滤器按

键可以定义浏览器过滤器。在这个对话框中，你可以修改，创建和删除过滤器：



一个过滤器由下面属性组成：

- 描述（Description）

这个描述在浏览器顶部，并在选择列表上方显示。

- 用户（User）

定义这个过滤器适用于哪个 Oracle 用户。如果你未填写这个字段，这个过滤器将适用全部 Oracle 用户。如果你指定了一个用户和数据库（如 scott@chicago），该过滤器将只对使用这个用户和数据库登录的用户可见。

- 默认（Default）

如果你选中这个选项，当你启动 PL/SQL Developer 时，选择的过滤器将被激活。你可以为每个用户定义一个默认过滤器。默认过滤器显示为粗体。

- 查询子句（Where clause）

一个过滤器定义一个在 all_objects (或 dba_objects) 视图上的查询语句。使用查询子句可以限制这个查询语句的结果集。你可以使用下面字段：

- 宿主（owner）——对象的拥有者
- 对象名（object_name）——对象的名称
- 对象类型（object_type）——对象的类型（表、视图、包等等）
- 状态（status）——对象的状态（有效或者无效）
- 创建（created）——对象创建的日期/时间

- 最近 DDL 时间 (last_ddl_time) ——最近编译对象日期/时间
- 排序子句 (Order by clause)

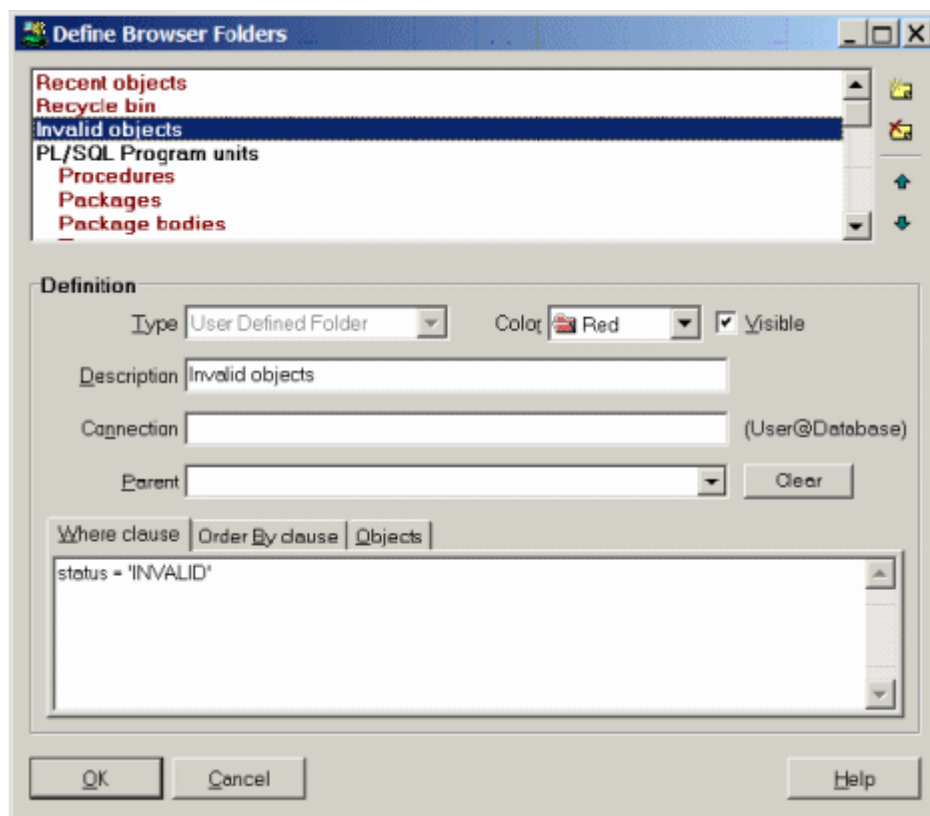
使用 all_objects 视图字段排序浏览器中的对象。

15.3 浏览器文件夹 (Browser Folders)

对象浏览器的文件夹可以用多种方式自定义。你可以修改浏览器中的文件夹顺序，每个文件夹的颜色，一个文件夹的层级，根据你当前的连接隐藏指定的文件夹。

你也可以定义你自己的文件夹和为它们填充对象。你可以明确增加指定的对象到文件夹中，或者定义一个查询填充它，或者结合两种方式。当该文件夹在对象浏览器中打开时查询被执行。

浏览器文件夹可以通过选择工具菜单中的浏览器文件夹 (Browser Folders) 或者按浏览器上方的文件夹按键定义。在对话框中你可以修改，创建和删除文件夹：



在这个对话框的上方你可以看到全部的文件夹和它们的层级关系。红色代表系统文件夹，黑色代表用户定义文件夹。系统文件夹有一些限制。

文件夹右侧有 4 个按键，可以在当前选择的文件夹位置增加一个新的文件夹，删除一个文件夹，或者在列表中上下移动。在浏览器中的顺序同列表中的顺序。

在底部你可以修改当前选择文件夹的定义：

- 类型 (Type) ——显示文件夹类型：用户定义或者系统定义

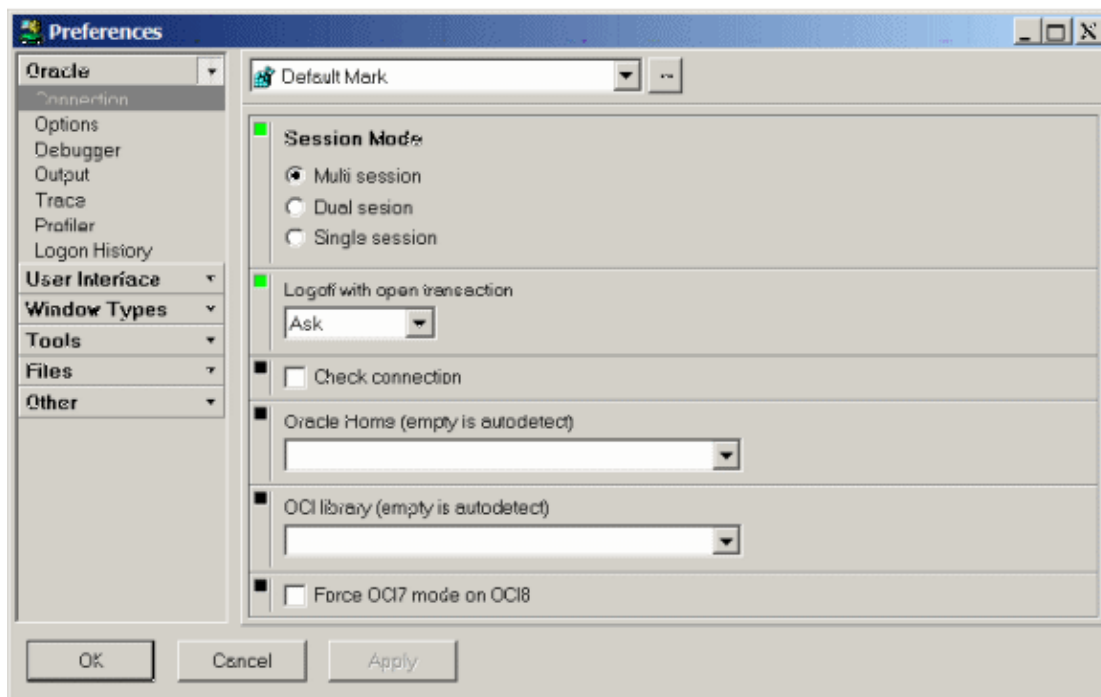
- 颜色 (Color) —— 选择文件夹的颜色。子文件夹将继承相同的颜色
- 可视 (Visible) —— 禁用这个选项使该文件夹在浏览器中不可见。这可以隐藏特殊的系统文件夹并覆盖连接选项
- 描述 (Description) —— 对象浏览器中文件夹的描述。系统文件夹不能修改
- 连接 (Connection) —— 定义对于哪些 Oracle 用户可以显示这个文件夹。如果你没有填写，这个文件夹将对全部 Oracle 用户显示。如果你指定一个用户和数据库（如 `scott@chicago`），只有使用这个用户和数据库登录的用户才能看见这个文件夹。你可以使用 `?` 和 `*` 通配符匹配多个用户。例如 `*@prd` 表示 `prd` 数据库中的全部用户。
- 父 (Parent) —— 为分级放置一个文件夹到另外一个文件夹中，你可以从列表中选择它的父。要清除父可以按清空键。
- 查询子句 (Where clause) —— 条件子句可以使用在基于查询的文件夹上，限制对象的选择。在上面的例子中，条件子句设置为 `status='INVALID'`，则该文件夹将只包含无效的对象。
- 排序子句 (Order By clause) —— 排序子句可以用于基于查询的文件夹上，定义查询对象的排序顺序。
- 对象 (Objects) —— 对于手工的文件夹，你可以定义对象列表。你可以在每行以 `owner.name` 格式放置一个对象。例如：`SCOTT.EMP`。注意更方便的方法是从对象浏览器中右键点击一个对象，从弹出菜单中选择增加到文件夹 (Add to folder) 菜单项来增加一个对象。

查询子句和排序子句可以包括任何 SQL 表达式和任何 `all_objects` 视图的字段。详见浏览器过滤器 (Browser Filters) 章节。

注意过滤器不影响用户定义文件夹的对象。

16 参数选择 (Preferences)

参数选择 (Preferences) 工具允许你设置多种 PL/SQL Developer 参数。当你选择这个菜单项后，出现下面的对话框：



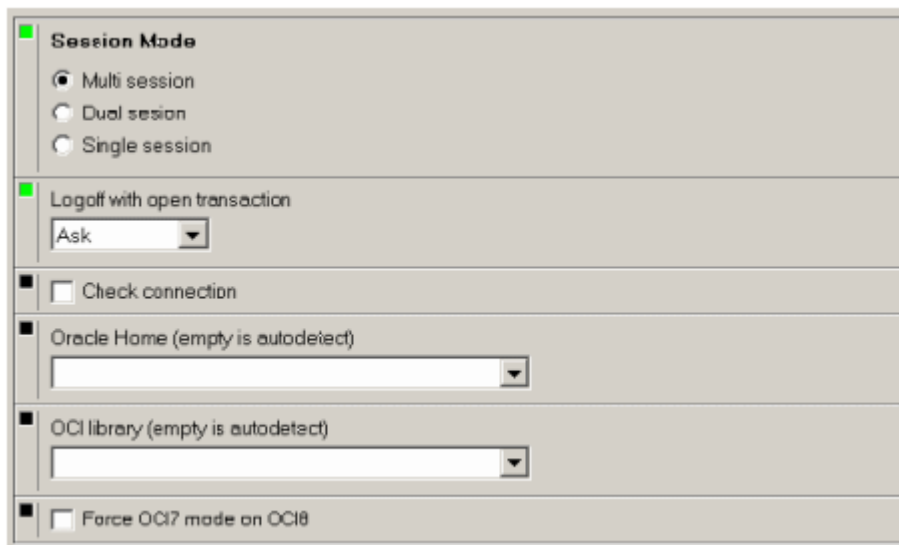
在左侧你可以从 6 大类之一选择一个参数页（Oracle，用户界面，窗口类型，工具，文件和其他）。

在上方你可以选择一个参数集，在一个特定级别上定义了一组参数。默认情况下，你将定义你的 Windows 用户的个人参数（上面屏幕截图是用户 Mark），但是你也可以定义一个指定 Oracle 连接的个人参数。任何时候你使用这个连接登录数据库，相应的参数将被使用。此外，你的系统管理员可以定义全局和 Oracle 连接的默认参数。有关参数集详见参数集（Preference sets）。

在参数集下是当前选中参数页的参数（上面的屏幕截图是 Oracle-连接页）。每个参数都有当前值。参数左侧的一个绿色指示器预示这个值是当前参数集的设置。如果指示器是黑色，数值是从另外一个参数集中继承（将鼠标光标停在指示器上看参数集名称）。为将一个参数恢复到继承值（也可以说：移除当前参数设置），你可以在绿色指示器上点击。

下面章节将详细描述每个参数页。

16.1 Oracle-连接 (Oracle – Connection)



- 会话模式 (Session mode)

控制 PL/SQL Developer 将使用多少个隔离数据库会话。这个特性已经在窗口、数据库会话和事务章节中描述。

- 开启事务时注销 (Logoff with open transaction)

当一个会话被注销并且这个会话已经开启了一个事务，默认动作是提交这个事物。你也可以通过这个参数定义回滚这个事务，或者将提示确认。

- 检查连接 (Check Connection)

当勾选这个参数时，PL/SQL Developer 每 60 秒检查一次你的数据库连接是否还活动 (alive)。如果你的会话被杀掉或者服务器关闭，PL/SQL Developer 自动注销。

- Oracle 起始位置 (Oracle Home)

默认情况下，PL/SQL Developer 将使用你的 PC 机中的 Primary Oracle Home。你可以使用 Oracle Home Selector(一个标准 Oracle 工具)定义 Primary Oracle Home。你可以通过从这个列表中选择来强制 PL/SQL Developer 使用一个特殊的 Oracle 起始位置。

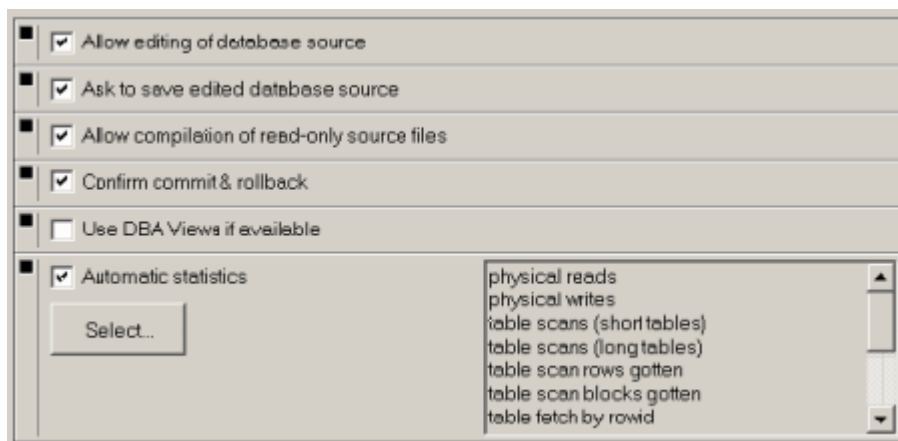
- OCI 库 (OCI Library)

PL/SQL Developer 将使用已安装的最新版本 SQL*Net 或者 Net8。如果这个版本的 SQL*Net 或者 Net8 出现问题，你可以强制 PL/SQL Developer 使用另外的版本。DLL 名如 ora72.dll (SQL*Net 2.2), ora73.dll (SQL*Net 2.3), oci.dll (Net8 8.1.x)，等等。这些 DLL 的位置在 Oracle 的 bin 目录下。

- 在 OCI8 上强制使用 OCI7 模式（Force OCI7 mode on OCI8）

有些版本的 Net8（8.0.3 和 8.0.4）存在 bugs，这将在 PL/SQL Developer 中产生问题。设置这个参数就是使用更稳定的 SQL*Net 2 替代 Net8。一个 Net8 缺陷的例子是当描述过程和函数时报表错误的参数模式。设置这个参数的缺点是你不可以查询 BLOB，CLOB 和 BFILE 字段，并且预先包装的包不能被描述。

16.2 Oracle-选项（Oracle – Options）



- 允许编辑数据库源代码（Allow editing of database source）

这个选项控制是否一个程序单元或者视图的源代码可以从对象浏览器中直接编辑，或者需要从文件系统中打开。

- 询问保存编辑的数据库源代码（Ask to save edited database source）

如果你编辑一个保存在数据库字典中程序单元源代码（在程序单元上右键点击，从弹出菜单中选择编辑菜单项），你可以通过禁用这个选项防止询问保存这些修改到一个文件中。如果你编辑的一个程序单元是从文件系统打开，你将总被询问保存这些变化。

- 允许编译只读源文件（Allow compilation of read-only source files）

如果一个源文件是只读，当被载入一个程序单元窗口时，它不能被编译。当使用版本控制系统时这个参数很有用，因为在工作目录中经常是一个只读的拷贝源文件。这个拷贝一定不能编译，因为这将干扰其他项目成员的工作。

- 确认提交和回滚（Confirm Commit & Rollback）

当你按相应键提交或者回滚一个事务时，你将被要求确认。禁用这个选项将直接提交或者回滚，不再要求确认。

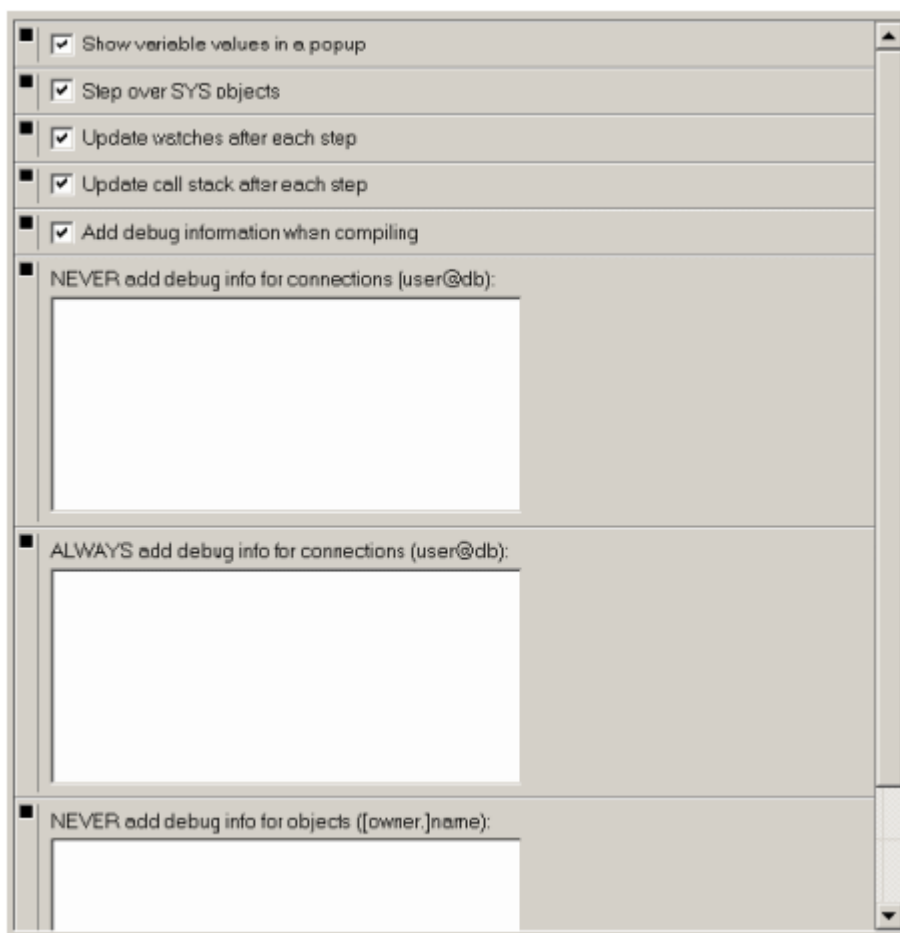
- 如果可用就使用 DBA 视图（Use DBA views if available）

如果设置这个参数, PL/SQL Developer 将试着使用 DBA 视图 (dba_objects, dba_triggers, 等等) 替代所有视图 (all_objects, all_triggers, 等等)。如果开发者具有某些系统权限, DBA 视图比 ALL 视图提供更多的信息。

- 自动统计 (Automatic statistics)

这个选项控制是否每执行一个 SQL 语句和测试脚本都形成一个统计报表。如果你禁用这个选项, 统计报表将空。通过点击 **Select...** 键, 打开一个当前连接可使用的全部统计列表。你可以选择你希望包含在一个统计报表中的统计。如果你没有存取动态性能表的权限, **Select...** 键禁用。

16.3 Oracle-调试 (Oracle – Debugger)



- 弹出显示变量值 (Show variable values in a popup)

如果你禁用这个参数, 在测试窗口中当你移动鼠标到变量名上将不自动弹出显示变量值。你必须在右键点击变量名后从弹出菜单中选择设置变量 (Set Variable) 菜单项。

- 单步跳过 SYS 对象 (Step over SYS objects)

如果禁用这个选项, 当你有查看 SYS 对象源代码的权限时, 调试器可以

单步进入这些程序单元。通常你不想这样做，所以这个选项默认是禁用。

- 每步后更新监视 (Update watches after each step)

如果你禁用这个选项，你可以按测试窗口的测试工具条中的更新监视 (Update watches) 键手工刷新监视。这样可以在较慢的配置环境下加快调试性能。

- 每步后更新调用堆栈 (Update call stack after each step)

如果你禁用这个选项，你可以按测试窗口的测试工具条中的更新调用堆栈 (Update call stack) 键手工刷新调用堆栈。这样可以在较慢的配置环境下加快调试性能。

- 编译时增加调试信息 (Add debug information when compiling)

当启用这个选项，每个在 PL/SQL Developer 中的编译都自动增加调试信息，这样，在一个调试期间，变量值总可以查看和设置。如果禁用这个选项，你可以手工在浏览器中增加调试信息。

- 从不对连接增加调试信息 (NEVER add debug info for connections)

对于匹配这个列表的连接，永不自动添加调试信息，不管“编译时增加调试信息”的参数设置。当你确定不想为生产数据库或者为某个用户增加调试信息时，这个选项有用。你可以使用熟悉的 `user@database` 语法，可以使用 * 和 ? 通配符（如任何数据库的 sys 用户可以使用 `sys@*`）。

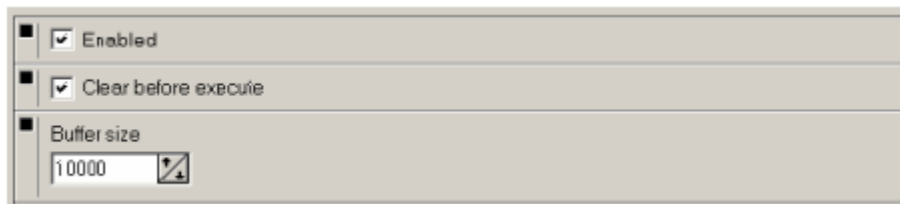
- 总为连接增加调试信息 (ALWAYS add debug info for connections)

对于匹配这个列表的连接，总是自动添加调试信息，不管“编译时增加调试信息”的参数设置。当你确定总是想为开发数据库或者为某个用户增加调试信息时，这个选项有用。你可以使用熟悉的 `user@database` 语法，可以使用 * 和 ? 通配符（如任何 devdb 数据库的用户可以使用 `*@devdb`）。

- 从不为对象添加调试信息 (NEVER add debug info for objects)

这个列表中的对象将永不增加调试信息。一些 Oracle 服务器版本在特定情况下可能在添加调试信息时产生错误。你可以看到在编译时有代表性的 ORA-00600, ORA-03113 或 PLS-00801 错误。如果出现这个问题，在 Oracle 提供修补前，你可以增加相应的对象到这个列表中。

16.4 Oracle-输出 (Oracle – Output)



- 启用 (Enabled)

在 SQL 窗口和测试窗口中, 调用 `dbms_output.put_line` 将被缓存并在脚本执行后被显示。在每个独立的 SQL 窗口和测试窗口的输出页上可以否决 (be overruled) 这个参数。

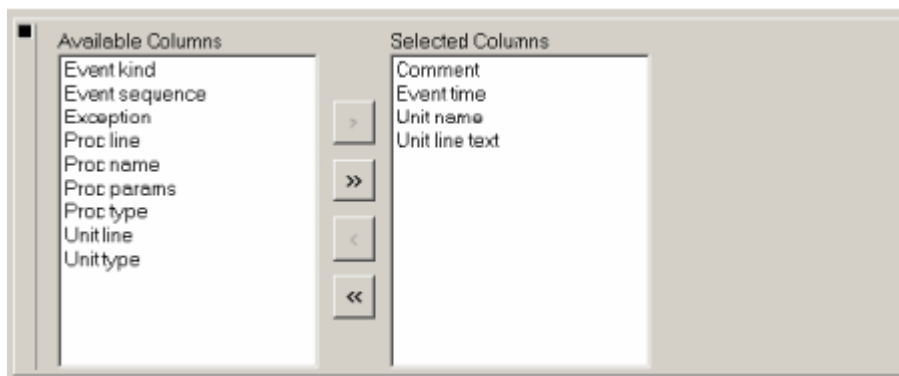
- 执行前清空 (Clear before execute)

执行一个 SQL 窗口或者测试窗口前清空当前的输出页, 这样, 输出页总显示最后执行的输出信息。

- 缓存大小 (Buffer size)

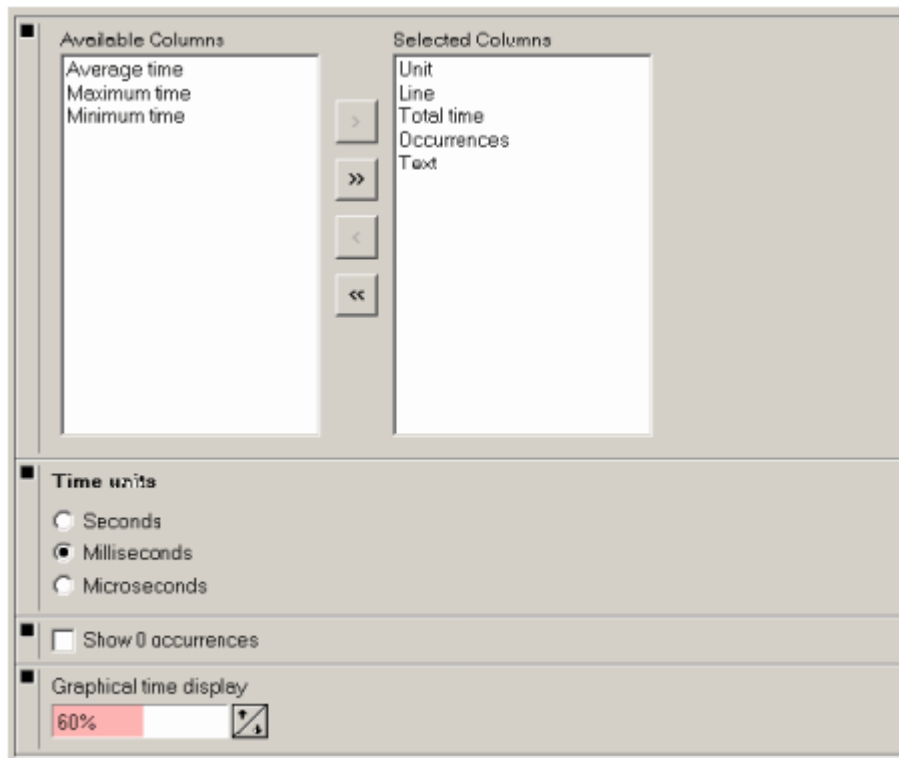
`dbms_output` 缓存的字节大小, 当一个执行输出超过这个数字时, 将产生一个异常。

16.5 Oracle-跟踪 (Oracle – Trace)



这些参数控制测试窗口的跟踪页 (见跟踪执行)。你可以控制哪些被跟踪表的字段被显示和排序方式。

16.6 Oracle - Profiler



这些参数控制测试窗口的 Profiler 页（见 The PL/SQL Profiler）。

- 字段（Columns）

控制哪些 profile 表字段被显示和排序方法。

- 时间单位（Time units）

定义总时间（Total time），最小时间（Maximum time），最大时间（Minimum time）和平均时间（Average time）字段的时间单位。

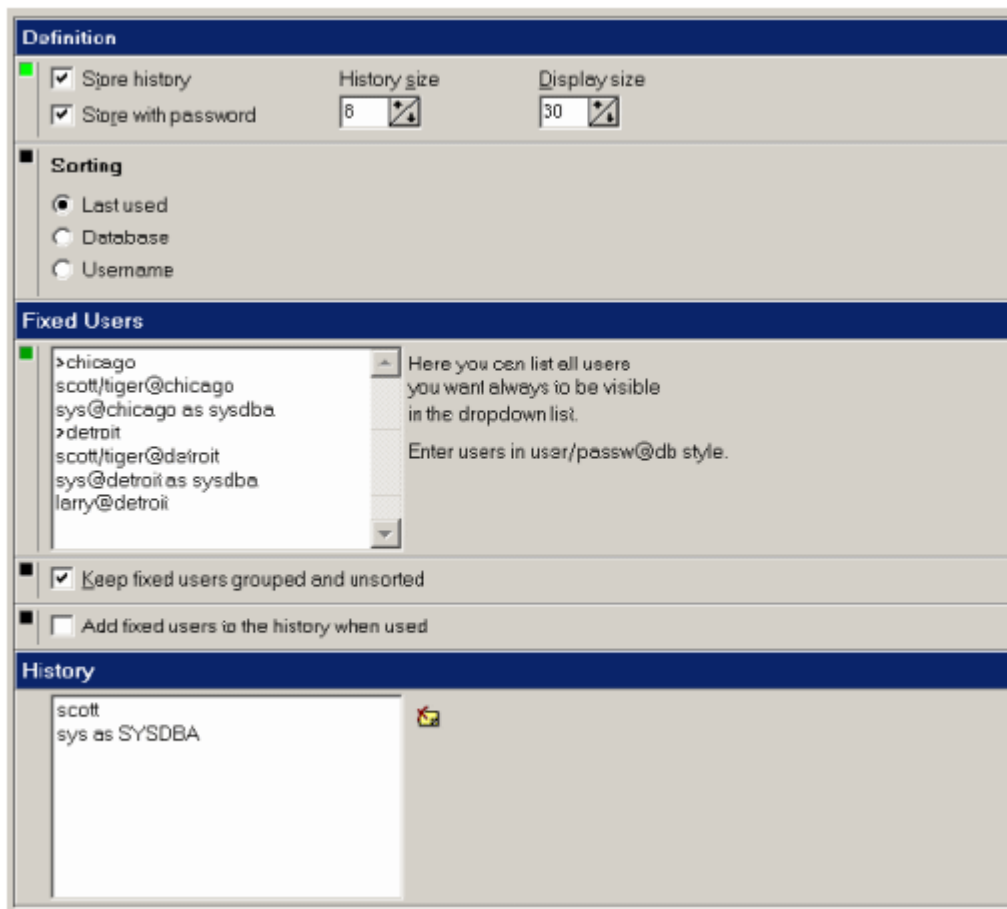
- 显示 0 执行（Show 0 Occurrences）

决定如果一个被报表的源程序代码行未执行是否被显示。

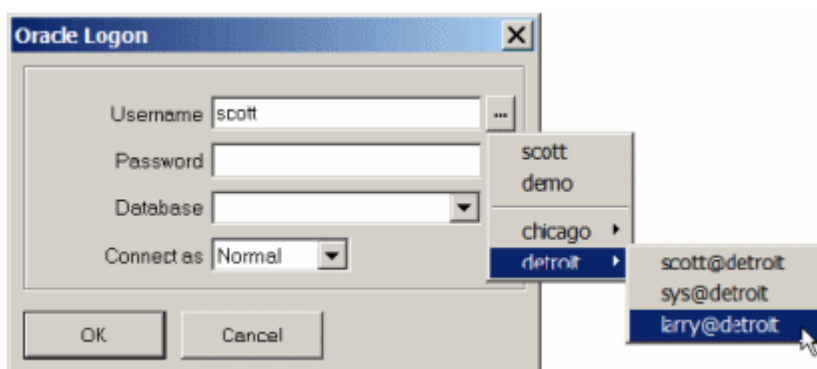
- 图形化时间显示（Graphical time display）

图形化展现所有时间字段的相对时间。你可以控制颜色亮度或者使其不可见。

16.7 Oracle-登录历史 (Oracle - Logon History)



如果你启用保存历史参数，PL/SQL Developer 将在登录历史中保存用户名、密码和数据库信息。当你今后重新启动 PL/SQL Developer 时，登录对话框允许你快速选择原来使用的用户账户：



即使 PL/SQL Developer 对密码进行了加密，但是为了安全的原因，你不能在历史中保存密码，这意味着如果任何人获得你机器上的 PL/SQL Developer 将可以使用你的账户登录数据库。因此，你应该禁用保存密码选项。历史大小和显示大小参数控制保存多少连接（最早的将被从历史中移除），在屏幕上显示多少连接。排序选项控制列表中的连接顺序。

注意即使你已经选择了在登录历史中保存密码，你仍然可以通过按回车时或者按确认

键登录时按下控制（Control）键禁用该选项。这样，你可以防止特殊账户（如 SYS 或者 SYSTEM）的密码被召回（be recalled）。

你也可以在固定用户部分定义一些带密码或者不带密码的固定账户。这些固定账户将总被显示，不会被自动移除。你还可以通过增加>menuname 的方式为固定用户定义子菜单项。上面的例子中，创建了一个 **chicago** 和 **detroit** 子菜单：

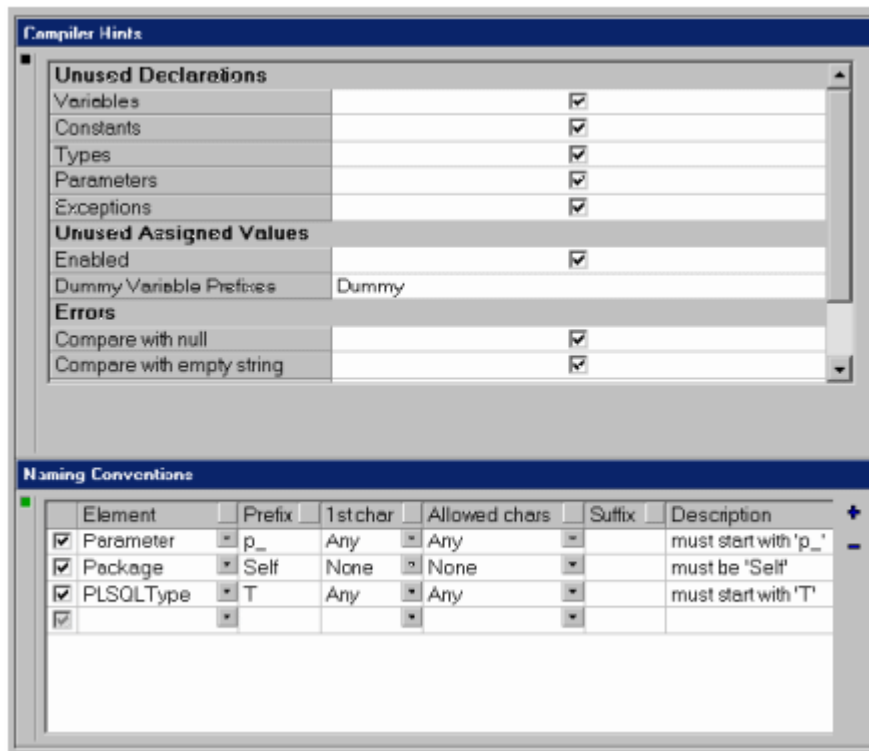
```
>chicago  
  
scott/tiger@chicago  
  
sys@chicago as sysdba  
  
>detroit  
  
scott/tiger@detroit  
  
sys@detroit as sysdba  
  
larry@detroit
```

在这种情况下，创建了一个数据库的一个子菜单，你可以创建希望的任何种类。例如开发、测试、生产。

如果启用了保持固定用户成组和不排序(Keep fixed users grouped and unsorted)选项，固定用户将显示在历史中的一个分开部分，并且不排序。如果启用当固定用户被使用到历史（Add fixed users to the history when used）选项，选择一个固定用户登录时将添加到历史中。

在历史部分，作为登录 PL/SQL Developer 的结果你可以看到哪些用户被增加到历史中。为删除一个指定项，选择它并按列表旁边的删除按钮。

16.8 Oracle-提示 (Oracle – Hints)



编译提示 (Compiler Hints)

你可以启用和禁用每个单独的提示，并且在适当的情况下控制其他提示属性。每个提示的解释和属性在提示列表下显示。

命名规则 (Naming Conventions)

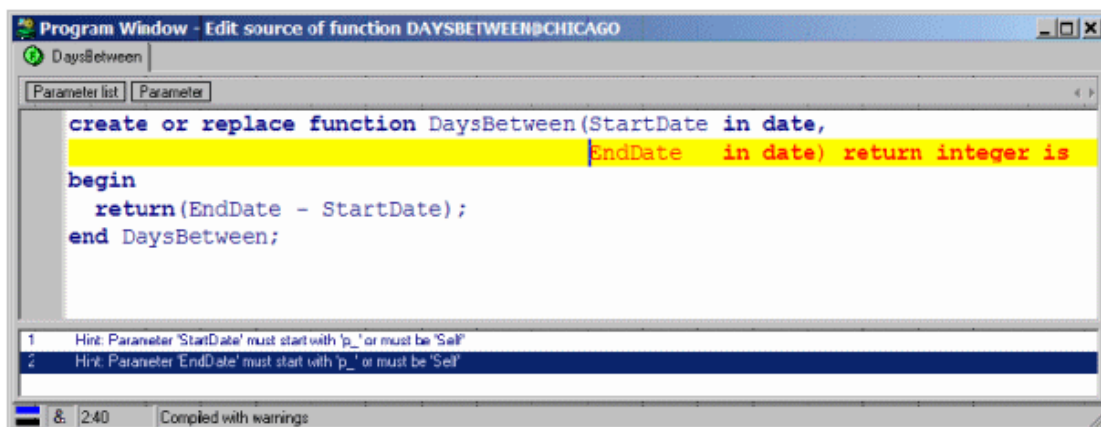
在这个列表中你可以定义命名规则，当一个程序单元被编译或者当调用 Show Compiler Hints function 功能时将检查这些命名规则。对于程序单元代码中的每一个不符合命名规则的元素显示一个提示信息。

对于每一个命名规则，你可以定义元素类型（参数、变量等等），元素类型的前缀，第一个和随后字符的可能值（前缀后），需要的后缀，和违反命名规则需要显示的提示描述。

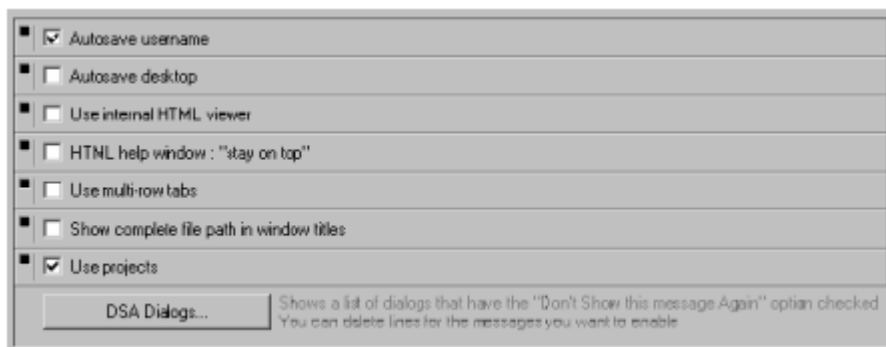
描述应该符合上面例子显示的形式。实际提示信息将是<元素> <名称> <描述>。例如：参数 'StartDate' 必须以 'p_' 开始（Parameter 'StartDate' must start with 'p_'）。

如果一个元素必须符合多个命名规则，则你可以简单地相同的元素类型增加多行。多行的描述将被显示，使用 'or' 分开。对于上面的例子，参数提示消息将是：

Parameter 'StartDate' must start with 'p_' or must be 'Self'



16.9 用户界面-选项 (User Interface – Options)



- 自动保存用户名 (Autosave username)

这个选项可以勾选自动保存每个 PL/SQL Developer 会话的用户名。当你将来重新启动 PL/SQL Developer 时，最后使用的用户名将是登录对话框的默认值。

- 自动保存桌面 (Autosave desktop)

当启用这个选项时，当你中止 PL/SQL Developer 时打开的全部文件将自动在重新启动时打开。当前打开的命令窗口或者解释计划窗口不和一个文件相关也被恢复。此外，主窗口和对象浏览器的尺寸将被恢复。

- 使用内部 HTML 浏览器 (Use internal HTML viewer)

在默认情况下，PL/SQL Developer 将使用一个内部 HTML 浏览器显示 HTML 帮助文档。如果你有 Microsoft Internet Explorer 4 或者更高版本，你可以禁用这个选项并使用 Microsoft 的 HTML 组件替代。这将在弹出菜单中增加一些额外的特性，并且将继承在 Internet Explorer 中的参数选择。

- HTML 帮助窗口：“固顶” (HTML help window: "stay on top")

如果启用这个选项，HTML 帮助将固顶直到你关闭或者最小化。如果被禁用，HTML 帮助窗口如一个独立任务显示，在窗口的任务条中带有自己的

按键。这样，你可以根据自己的需要在前端显示 PL/SQL Developer 主窗口或者帮助窗口。

- 使用多行标签 (Use multi-row tabs)

如果一个窗口包含多个标签页，这个选项控制是否当这些标签页在超出显示范围后拆分成多行。如果禁用这个选项，当标签超出范围后，将出现 2 个滚动键。当前这个功能只应用在程序窗口。

- 窗口标题使用完整路径 (Use complete path in window titles)

在默认情况下，一个窗口标题只显示文件名。如果启用这个选项，将显示完整路径。

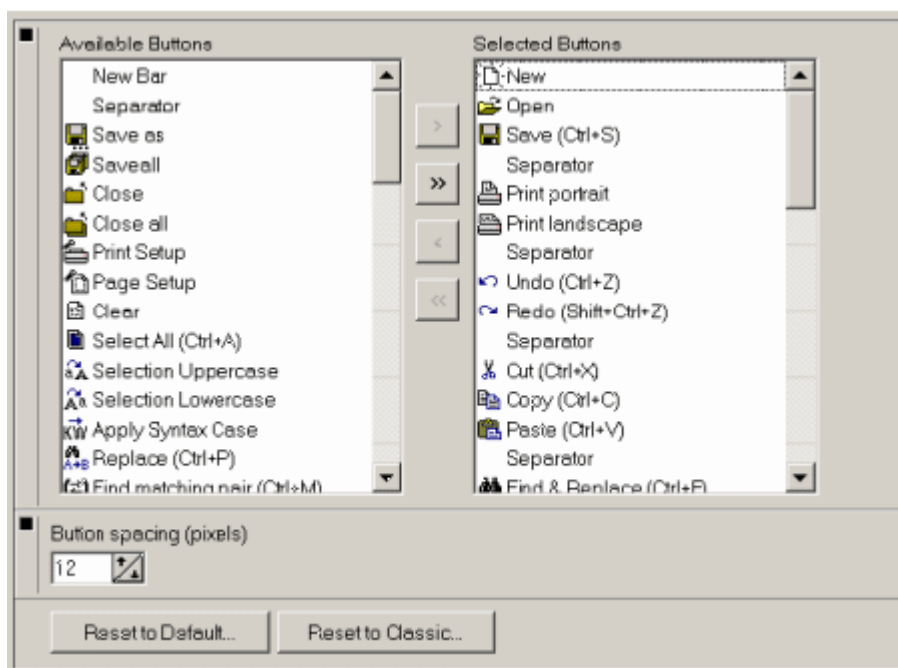
- 使用工程 (Use projects)

如果启用这个选项，工程主菜单项才可见。如果你没有使用工程就禁用它，这样不占用主菜单空间。有关工程的描述详见工程 (Projects) 章节。

- DSA 对话框 (DSA Dialogs)

只要你在一个消息框上选择 "Don't show this message again" 选项，一个条目将添加到你的参数的 DSA 部分。如果你随后想再启动这个消息，按 DSA 对话框 (DSA Dialogs) 键编辑这个部分。这将调出一个文本编辑器，允许你移除相应行。

16.10 用户界面-工具条 (User Interface – Toolbar)

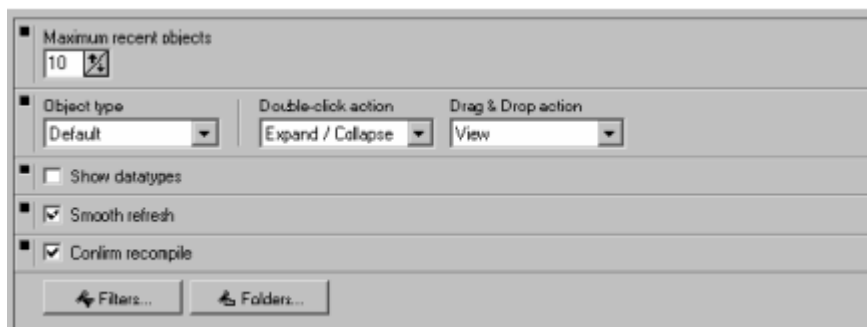


你可以控制哪些键显示以及排列顺序。用隔离 (Separator) 项隔离一组相关按键。新

建条（New Bar）项可以在前一个工具条下创建一个新工具条。

Reset to Default 键重置工具条成默认状态。Reset to Classic 键重置工具条成 PL/SQL Developer 3 布局。使用 Button spacing 值控制每个按键之间的间隔。

16.11 用户界面-浏览器（User Interface – Browser）



这些参数控制对象浏览器的外观和行为（见浏览对象章节）。

- 最近对象的最大值（Maximum Recent objects）

控制有多少个对象在最近对象文件夹中显示。当最近对象超过这个数值后，最早的对象将被移除。

- 双击和拖放动作（Double-click and Drag & Drop actions）

控制在浏览器中双击一个对象，或者从浏览器拖放到工作区时的执行动作。你可以指定一个默认动作，并且对于列表中的每个对象类型可以覆盖或者继承这个默认动作。

- 显示数据类型（Show data types）

当启用这个选项时，字段的数据类型（表和视图），属性（对象）和参数（函数和过程）将显示在浏览器中。

- 平滑刷新（Smooth Refresh）

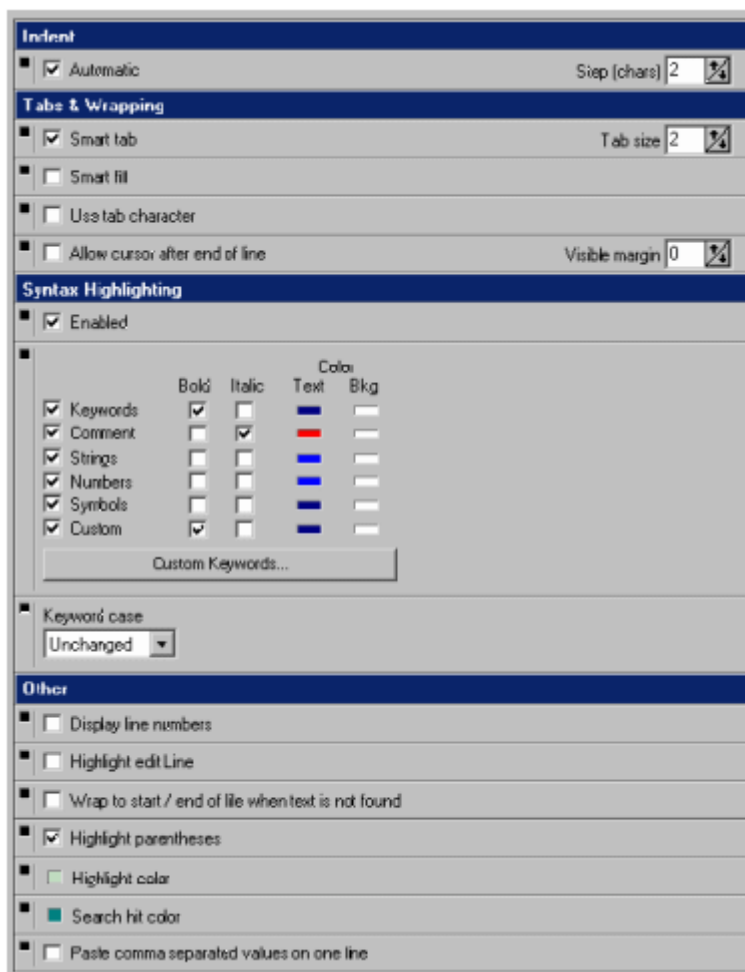
在以下系统中这个选项可能降低浏览器的刷新操作（如：修改浏览器过滤器，切换另一个用户登录，等等）。如果是这种情况，即使浏览器在刷新时屏幕有些闪烁，你也应该禁用这个选项。

- 确认重新编译（Confirm Recompile）

当启用这个参数时，在对象浏览器中重新编译一个程序单元将提示成功或者失败信息，当禁用这个参事时，只有编译失败时才显示。

过滤器（Filters（见浏览器过滤器章节））键和文件夹（Folders（见浏览器文件夹章节））键将调用相应的对话框。

16.12 用户界面-编辑器 (User Interface - Editor)



缩进 (Indent)

- 自动 (Automatic)
决定是否按回车后编辑器自动缩进。
- 步长 (字符) (Step (chars))
编辑器缩进或者取消缩进一个选中行的字符数。

跳格和换行 (Tabs & Wrapping)

- 智能跳格 (Smart tab)
将导致编辑器结合上一行确定一个跳格位置。
- 跳格大小 (Tab size)
决定跳格需使用的字符数。
- 智能填充 (Smart fill)
编辑器在可能的情况下使用跳格代替空格。

- 使用制表符 (Use tab character)

当禁用时，跳格和缩进使用空格。

- 换行 (Wrap lines)

行比窗口宽度大时延续到下行。

- 允许光标在行尾后 (Allow cursor after end of line)

当启用时，你可以将光标放置在一行的结束后。当你在此插入文本，本行将首先添加空格。

- 边界可见 (Visible margin)

在指示位置显示一个竖线。0=禁用。

语法高亮 (Syntax Highlighting)

这组参数控制编辑器使用的语法高亮。你可以启用或者禁用语法高亮，并且定义主关键字、注释、字符串、数字和符号 (:=, =>, ||, 等等) 的字体和颜色。对于关键字你也可以控制转换大写、小写或者首字母大写。你也可以自定义一个语法高亮，并且定义需要高亮的词。按自定义关键字(Custom Keywords)键将调出一个允许你定义这些词的编辑器。

其他 (Other)

- 显示行号 (Display line numbers)

勾选这个选项在程序编辑器和测试窗口中的左边界上显示行号。

- 高亮编辑行 (Highlight edit line)

控制是否高亮当前光标所在行。

- 当文本未找到时转动文件的开始/结束 (Wrap to start / end of file when text is not found)

当使用查找功能时，如果文本未找将继续从文件开始（当向下查找）或者从文件结束处（向上查找）查找。

- 高亮圆括号 (Highlight parentheses)

启用选项，当光标定位在圆括号的开始或者结束时，编辑器高亮圆括号对。

- 高亮颜色 (Highlight color)

确定所有高亮的颜色，如圆括号，变量，等等。

- 查找提示颜色 (Search hint color)

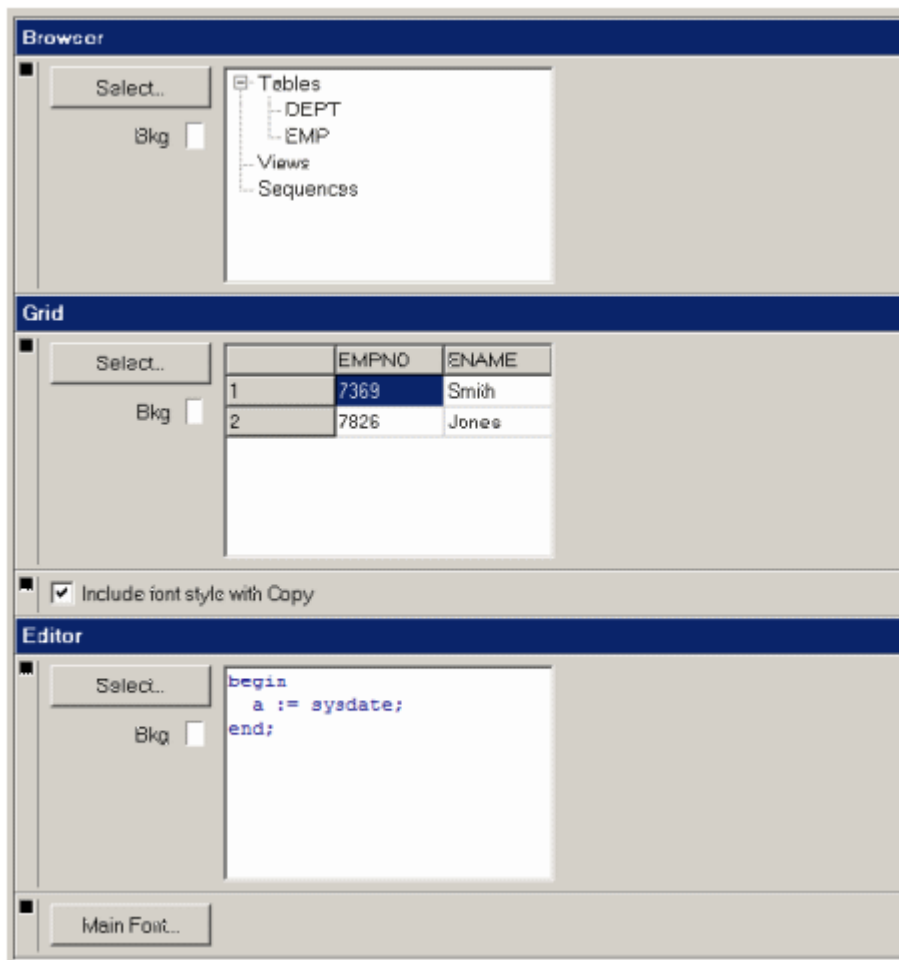
确定由查找发现的查找提示颜色。

- 在一行上粘贴由逗号分割的项目 (Paste comma separated items on one line)

控制从对象浏览器中或者一个描述窗口中选择多个项目粘贴到一个编辑

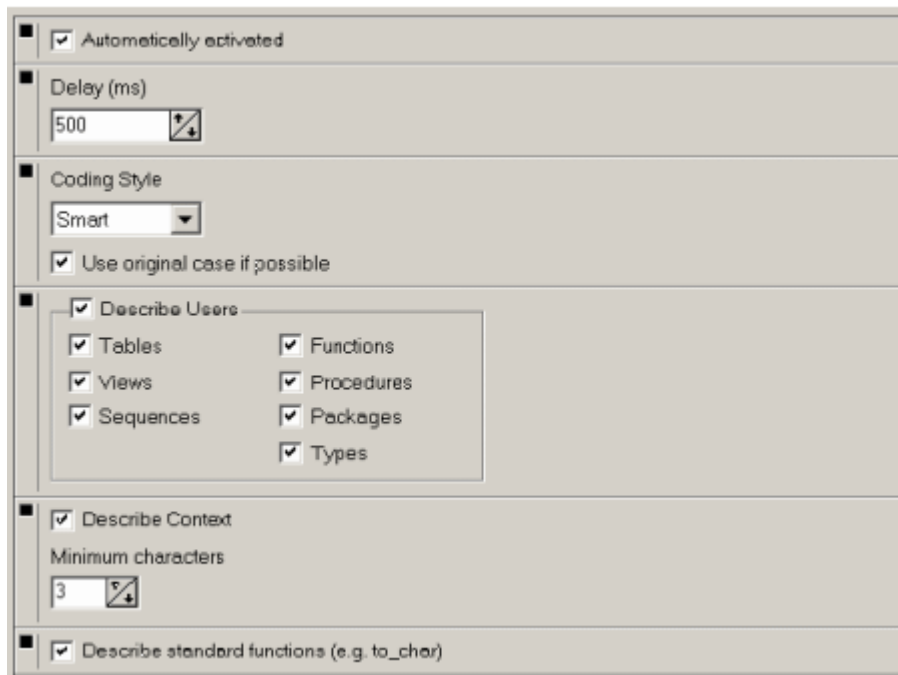
器里的一行还是多行上。

16.13 用户界面-字体 (User Interface – Fonts)



字体参数选项控制浏览器字体、编辑器字体和像在 SQL 窗口、属性窗口等等中的不同表格使用的字体。通过按选择 (Select) 键你可以修改字体的名称、格式和颜色。你也可以为所有对话框设置默认字体。

16.14 用户界面-代码助手 (User Interface - Code Assistant)



代码助手在你输入数据库对象名称时自动显示它们的信息（见代码助手章节）。这个参数选项页允许你定义这些特征的行为。

- 自动激活 (Automatically activated)

代码助手可以在一个确定的延时后自动调用（见下）。你也可以选择通过一个功能键手工激活代码助手。

- 延时 (Delay)

编辑器显示代码助手列表前将等待的毫秒数。

- 代码风格 (Coding Style)

控制当你选择它们时，被选择项目如何插入编辑器：

- Smart——代码助手将看被描述对象决定风格。
- Init Caps——每个词（被下划线分割的）的首字母大写。
- Lowercase——所有字符转换成小写。
- Uppercase——所有字符转换成大写。
- 尽可能使用原风格 (Use original case if possible)

当启用这个选项时，代码助手将尽可能依据保存在数据库字典中的源代码确定标识符的大小写。这个选项应用到全部程序单元和它们的元素（参数，类型，等等）以及视图字段上，并且覆盖上面描述的代码风格参数选项。如果原大小写不能确定，将使用代码风格 (Coding Style)。

你可以因为性能问题禁用这个特性。

- 描述用户（Describe users）

确定当你输入一个用户名后跟一个句点时是否列出这个用户拥有的对象。如果这个选项被启用，你可以额外定义你想在列表中包括的对象类型。

- 描述上下文（Describe context）

确定代码助手是否描述当前用户，编辑器和程序单元的上下文信息。

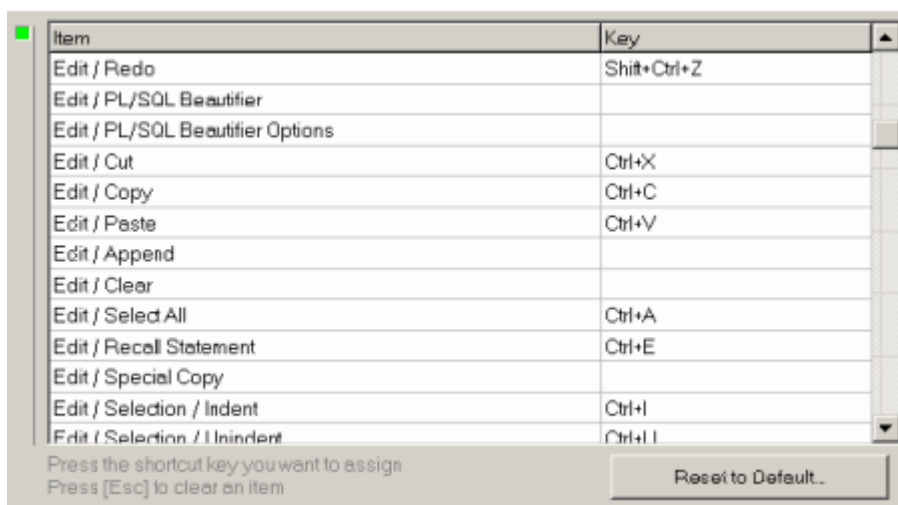
- 最少字符（Minimum characters）

确定自动调用上下文描述之前需要输入一个词的多少个字符。注意即使你没有输入设置的字符数也可以手工调用代码助手。

- 描述标准函数（Describe standard functions）

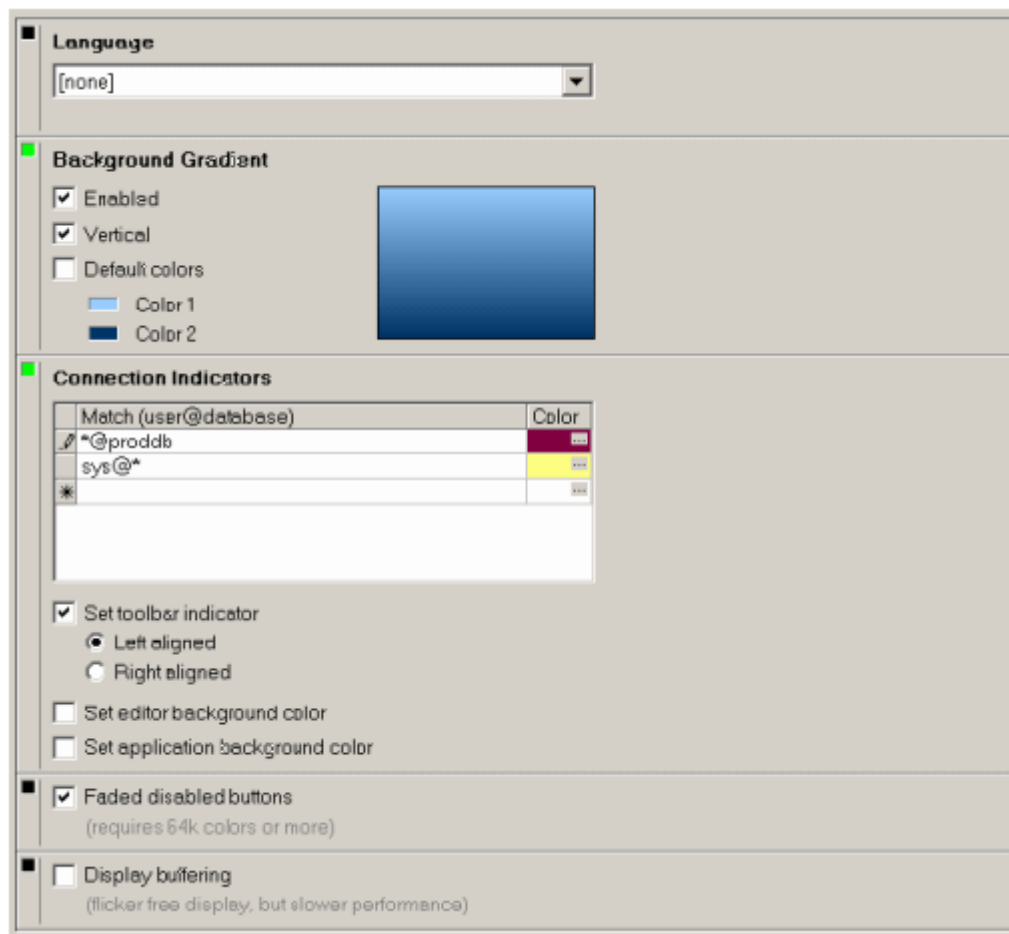
在默认情况下，代码助手将描述标准函数，如 `to_char`, `add_months`, 等等。如果你非常熟悉这些函数，你可以禁用这个选项。

16.15 用户界面-键配置（User Interface - Key Configuration）



在这个页，你可以为 PL/SQL Developer 的所有功能定义自己的键配置。只要从列表中选择功能，按下你想为这个功能使用的键组合。要清除一个特定功能的热键，从列表中选择后按 Esc 键。

16.16 用户界面-外观 (User Interface – Appearance)



- 语言 (Language)

如果是多国语言版本的 PL/SQL Developer，你可以在这里选择语言。所有标题和信息将随即按所选择的语言显示。

- 背景梯度 (Background Gradient)

允许你定义 PL/SQL Developer 工作区的颜色和梯度。如果你只想使用单一颜色而不想使用梯度，在颜色 1(Color 1)和颜色 2 (Color 2) 中设置相同值。

- 连接指示器 (Connection Indicators)

这些选项允许你提供当 PL/SQL Developer 使用特殊连接时的可视化提示。你可以定义多个符合 user@database 格式的连接匹配和相关颜色。这个颜色可以关联随后的左对齐或者右对齐工具条指示器，编辑器背景或者应用程序背景（覆盖背景梯度）。

一个连接匹配可以包含*和? 通配符。在上面的例子中，任何用户连接到 PRODDb 数据库时，编辑器背景和右对齐工具指示器将为红色。当使用 SYS 账户连接任何数据库时为黄色。在列表中的第一项有更高的优先级，所以当使

用 SYS 账户连接 PRODDb 数据库时，指示器为红色。

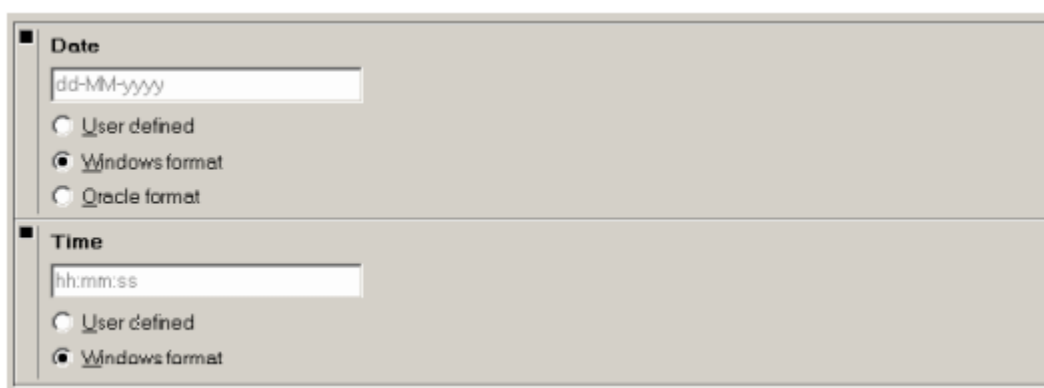
- 淡色禁用按钮（Faded disabled buttons）

当启用时，禁用按钮以淡色显示。当禁用时，禁用按钮以标准的灰色显示。对于使用 256 色显示的屏幕，禁用按钮总是灰色显示。

- 显示缓冲（Display buffering）

当窗口移动或者调整大小时，屏幕闪烁少一些，但是显示性能可能会略微降低。

16.17 用户界面-日期/时间（User Interface - Date/Time）



在这页，你可以定义在 PL/SQL Developer 中使用的日期和时间格式。

日期格式有 3 种方法定义：

- 用户定义（User defined）

一种明确定义的日期格式，遵循 Windows 区域设置的相同规则。

- Windows 格式（Windows format）

格式定义同 Windows 区域设置。

- Oracle 格式（Oracle format）

格式定义同你的 Oracle 环境（NLS_DATE_FORMAT）。

时间格式有 2 种方法定义：

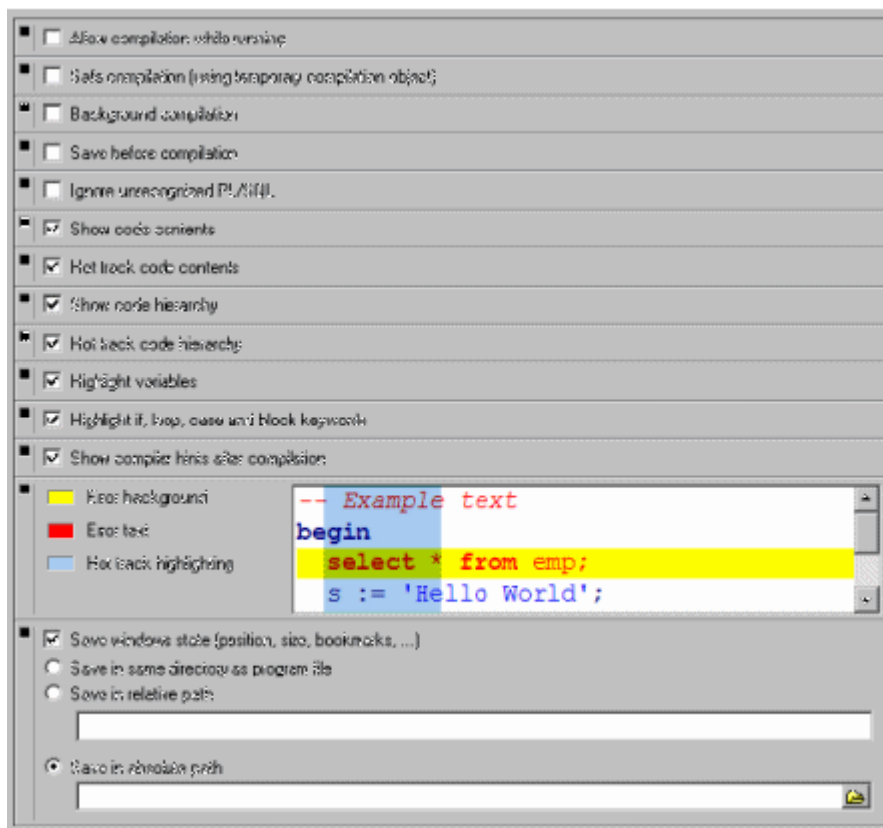
- 用户定义（User defined）

一种明确定义的时间格式，遵循 Windows 区域设置的相同规则。

- Windows 格式（Windows format）

格式定义同 Windows 区域设置。

16.18 窗口类型-程序窗口（Window Types - Program Window）



这些参数选项影响程序窗口的行为（详见编写程序章节（Writing programs））。

- 允许运行时编译（Allow compilation while running）

在多或者双会话模式下，从技术上讲，你可以在运行测试脚本或者 SQL 脚本时编译程序单元，这将引发锁定而防止这个参数设置。

- 安全编译(使用临时编译对象)(Safe compilation (using temporary compilation object))

当被启用，每个编译首先通过使用一个临时编译对象进行测试。当这个测试成功时，编译实际的对象。如果在一个因无效对象可能造成问题的环境中编译对象，这个选项可能有用。

- 后台编译（Background compilation）

当被启用，编译使用一个独立会话在后台进程中执行。这种方式可以在编译时继续其他工作。这将为每一个程序窗口引发一个额外的数据库会话。注意这个参数只有在会话模式参数（见 Oracle-连接章节）设置为多会话时才有效。

- 编译前保存（Save before compilation）

如果你想确保数据库对象和在文件系统中的源程序文件保持同步，你可以启用这个选项。

- 忽略未被承认的 PL/SQL（Ignore unrecognized PL/SQL）

当被启用，程序窗口将忽略所有不是以"create or replace <program unit type>"开始的 SQL。在程序文件中的其他 SQL 或者 SQL*Plus 命令将被忽略。

- 显示代码目录 (Show code contents)

这个参数控制是否显示代码目录窗格，详见代码目录 (Code Contents) 章节。

- 热跟踪代码目录 (Hot track code contents)

控制是否在代码目录窗格中当鼠标移动到一个项目上时，自动地在编辑器中高亮相应 PL/SQL 代码。

- 显示代码层级关系 (Show code hierarchy)

这个参数控制是否显示代码层级窗格，详见代码层级 (Code Hierarchy) 章节。

- 热跟踪代码层级 (Hot track code hierarchy)

控制是否在代码层级窗格中当鼠标移动到一个项目中，自动地在编辑器中高亮相应 PL/SQL 代码。

- 高亮变量 (Highlight variables)

当光标定位在一个变量时，出现这个变量的地方都被高亮。这将方便查看变量的使用 and 修改变量可能造成的影响，如果禁用这个选项，你必须使用编辑菜单中的查找菜单项。

- 高亮 if, loop, case 和块关键字 (Highlight if, loop, case and block keywords)

当光标定位在一个 if / then / else / elsif / end if 关键字，一个 loop / end loop 关键字，一个 for / loop / end loop 关键字，一个 while / loop / end loop 关键字，一个 case / when / then / else / end case，和 declare / begin / end 关键字。如果这个选项禁用，你必须使用编辑菜单中的查找菜单项。

- 编译后显示编译提示 (Show compiler hints after compilation)

编译提示提供有关未使用的声明，未使用的赋值，可能错误和警告信息。当启用这个参数，这些提示信息将自动在编译后显示在编译错误后（如果有）。如果你禁用这个参数，你必须从工具菜单或者工具条中选择显示编译提示 (Show Compiler Hints) 菜单项。

- 错误背景 (Error background)

选择编译错误文字的背景颜色。

- 错误文字 (Error text)

选择编译错误文字的颜色。

- 热跟踪高亮 (Hot track highlighting)

选择高亮文字的背景颜色，包括代码目录热跟踪，代码层次热跟踪，和 if, loop,

case 和 block 关键字。

- 保存窗口状态 (Save window state)

当被启用，程序窗口的位置，大小，书签，颜色标记，和置换参数被保存在一个.pfi 文件中，并当文件或者数据库对象被再次浏览或者编辑时恢复。

16.19 窗口类型-SQL 窗口 (Window Types - SQL Window)

☐ AutoCommit SQL
☐ AutoCommit posted records
☒ AutoExecute queries
☐ AutoSelect statement
☒ Ask to save new windows
☐ Linked query in new window
☐ Number fields to_char
☐ Date fields to_char
☒ Show dictionary info in result grid
☐ Null value cell color
Records per Page
☒ Automatically determined
☐ Fixed

☐ All records
Query By Example
☐ Case insensitive
☐ Automatic contains
☐ Ignore time fraction
Number Layout
☐ Left aligned
☐ Right aligned
☒ Formatted
☐ Formatted with thousand separator

这些参数选项影响 SQL 窗口的行为（见特殊的 SQL (Ad hoc SQL) 章节）。

- 自动提交事务 SQL (AutoCommit SQL)

这个选项控制 DML 语句在 SQL 窗口执行后是否自动提交事务。

- 提交的记录自动提交事务 (AutoCommit posted records)

这个选项控制通过结果集表格提交的记录是否自动提交事务

- 自动执行查询 (AutoExecute queries)

如果启用这个选项，当在一个表或者视图上选择查询数据 (Query Data) 或者编辑数据 (Edit Data) 时，一个 SQL 窗口被创建并立即执行。如果禁用这个选项，你

可以在执行前修改生成的查询语句。

- 自动选择语句 (AutoSelect statement)

如果你在 SQL 编辑器中有一个以上的 SQL 语句 (分号隔开), SQL 窗口在执行前将自动选择当前光标所在的语句。这样, 你不必自己选择要执行的语句。

- 询问保存新窗口 (Ask to save new windows)

当你关闭一个刚创建的 SQL 窗口时, 在通常情况下, 你将被询问是否保存。如果你禁用这个选项, 这个确认信息将被省略。如果你更新了一个以前打开的 SQL 文件, 无论这个选项如果设置, 你将总被询问是否保存。

- 关联的查询在新窗口 (Linked query in new window)

当这个参数被设置, 执行一个关联的查询 (见关联查询 (Linked Queries)) 将创建一个新 SQL 窗口, 如果没有设置, 关联查询将在当前 SQL 窗口中执行。

- 数字字段 to_char (Number fields to_char)

当启用这个参数, 数字字段将在服务器上被转换成字符串。会话的 NLS 参数将决定数值的格式。PL/SQL Developer 不执行任何附加的数字格式或者确认有效性。

- 日期字段 to_char (Date fields to_char)

当启用这个参数, 日期字段将在服务器上被转换成字符串。会话的 NLS 参数将决定数值的格式。PL/SQL Developer 不执行任何附加的日期格式或者确认有效性, 在编辑日期域时, 日期选择工具(date picker)将不能使用。

- 在结果集表格中显示字典信息 (Show dictionary info in result grid)

当被启用, 下面的字典信息将在结果集表格中显示:

- 字段数据类型, 可选项, 和注释将被显示在状态行中。
- 对于受限制的字段, 当限制是特定值时显示一个查找列表 (如 col in (value1, value2, ...))
- 对于字段带一个外键限制, 并且关联到少于 1000 行的小表时显示一个查找列表。

- 空值单元格颜色 (Null value cell color)

这个颜色将被使用提示空值。这使空值和空格以及不在表格中直接显示 (如 LOB's 和 Longs) 的容易区分开来。

- 每页记录数 (Records per page)

这个参数定义在一个 SQL 窗口的结果集表格中从查询语句初次和每一次按下一页键提取的记录数:

- 自动确定 (Automatically determined) — 根据结果集表格大小确定读取的记录

数。

- 固定 (Fixed) —按指定记录数读取。
- 全部记录 (All records) —当查询执行时立即读取全部记录。
- 数字布局 (Number layout)

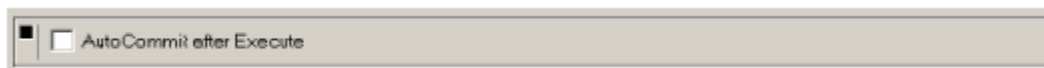
这个参数控制数字字段如何显示：

- 左对齐 (Left aligned) —数值左对齐显示，没有任何格式。
- 右对齐 (Right aligned) —数值右对齐显示，没有任何格式。
- 格式化 (Formatted) —数值右对齐显示，带一个格式符合字段的精度和范围。
- 格式化带千分符 (Formatted with thousand separator) —类似前一个选项，但是带一个千分符 (如：1,277.65)
- 通过例子查询 (Query By Example)

这些参数控制 SQL 窗口在 QBE 模式中的默认行为(见通过例子模式查询章节)：

- 大小写不敏感 (Case insensitive) —查询字符字段值时大小写不敏感。
- 自动包含 (Automatic Contains) —字符字段只需要包含查询值。
- 忽略时间小数 (Ignore Time fraction) —如果查询值未包含一个时间小数，则忽略日期字段的时间小数。

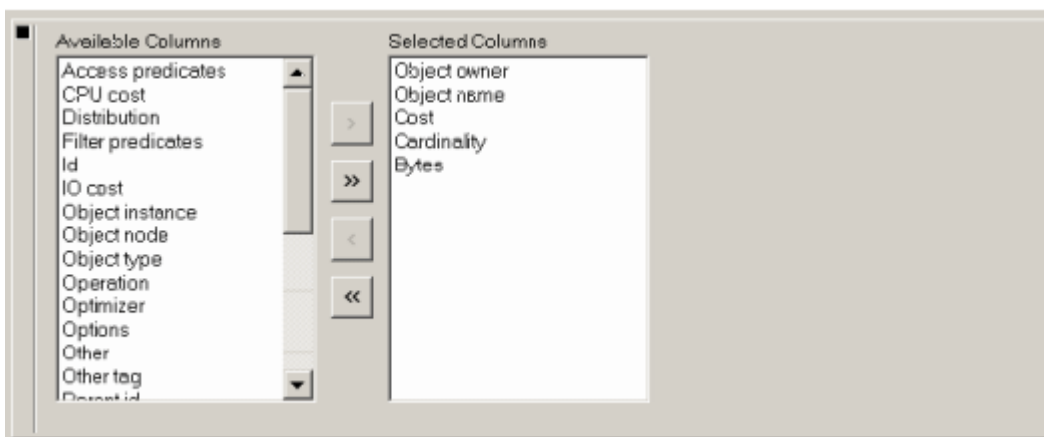
16.20 窗口类型-测试窗口 (Window Types - Test Window)



- 执行后自动提交

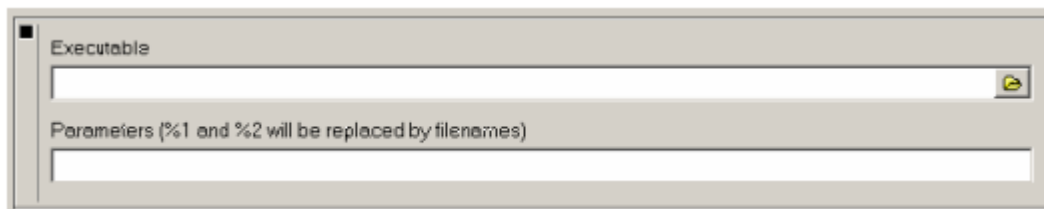
这个选项控制是否执行测试脚本自动提交事务。

16.21 窗口类型-计划窗口 (Window Types - Plan Window)



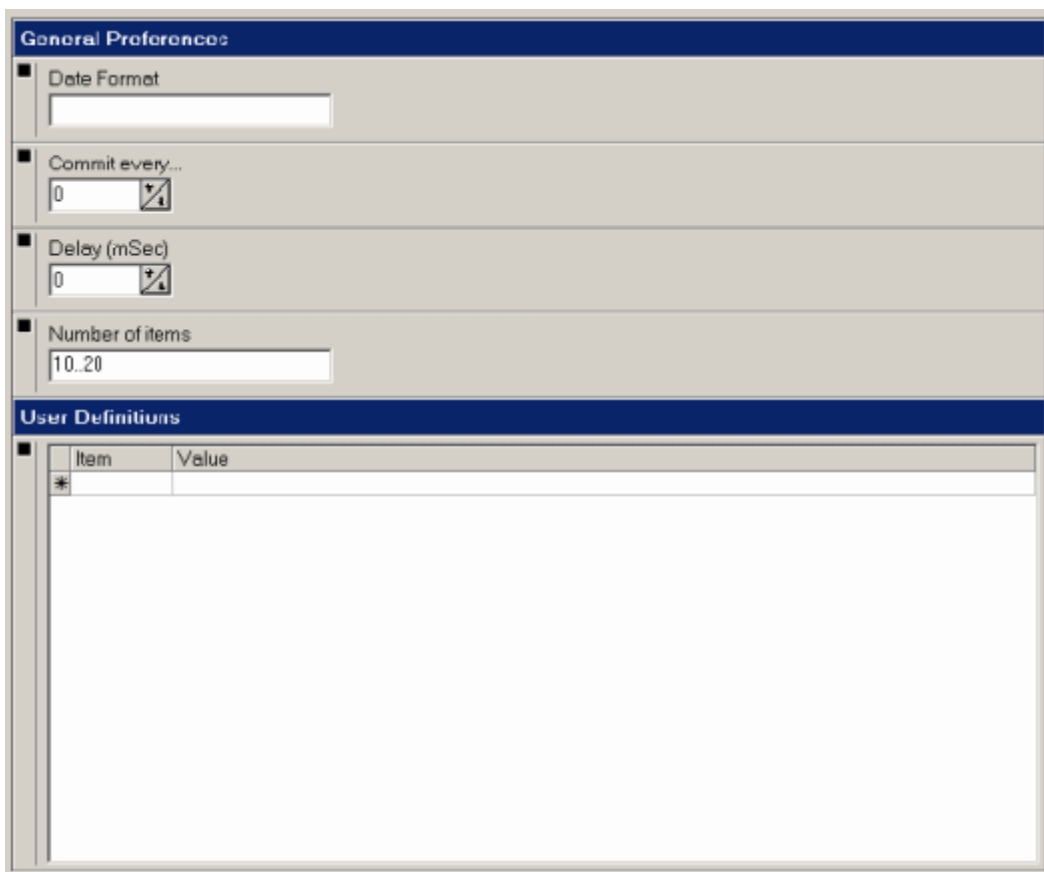
这些参数控制在计划窗口中计划表（plan table）中的哪些字段需要显示（见使用解释计划窗口），并且他们显示的顺序。

16.22 工具-差异（Tools – Differences）



在这个参数页,你可以选择外部差异工具作为比较用户对象(Compare User Objects)工具（见比较用户对象章节(Compare User Objects)）。

16.23 工具-数据生成器（Tools - Data Generator）



这个参数页允许你为数据生成器工具定义一个区间默认集。这些默认可以被每个数据生成器定义覆盖。更多关于这些参数的意义见数据生成器（Data Generator）。

16.24 工具- To-Do 列表 (Tools - To-Do List)



这些参数控制在 To-Do 项目编辑器中的优先级，种类和宿主值选择明细表。对于优先级，它可以对每个值前带一个数值，这样明细表可以使用优先级排序有用。详见 To-Do 项目(To-Do Items)章节。

16.25 工具-召回语句 (Tools - Recall Statement)



在这个参数页，你可以定义在语句召回缓存中的最大语句数，和它保存的目录。详见召回语句(Recalling statements)章节。

16.26 文件-目录 (Files – Directories)



- 程序文件 (Program files)

定义打开和保存程序文件的起始目录。如果你没有填写，你的起始目录将总是上次打开或者保存一个程序文件的目录。

- 测试脚本 (Test scripts)

定义打开和保存测试脚本的起始目录。如果你没有填写，你的起始目录将总是上次打开或者保存一个测试脚本的目录。

- SQL 脚本 (SQL Scripts)

定义打开和保存 SQL 脚本的起始目录。如果你没有填写，你的起始目录将总是上次打开或者保存一个 SQL 脚本的目录。

- 报表文件 (Report files)

定义打开和保存报表文件的起始目录。如果你没有填写，你的起始目录将总是上次打开或者保存一个报表文件的目录。

- 命令文件 (Command files)

当从菜单或者通过输入@file 打开一个命令文件，这个将是默认目录，如果你没有填写，将使用工作目录。

- 宏 (Macros)

这个目录将被用作保存你的宏库。如果你没有填写，将使用 PL/SQL 目录下的 Macro 目录。

- 模版 (Templates)

这个目录将被用于保存程序文件模版，如果你没有填写，将使用 PL/SQL 目录下的 Template 目录。

- 插件 (Plug-Ins)

这个目录在查询插件时使用。如果你没有填写，将使用 PL/SQL 目录下的 PlugIn 目录。

- 工程 (Projects)

定义打开和保存工程文件的起始目录。如果你没有填写，你的起始目录将总是上次打开或者保存一个工程文件的目录。

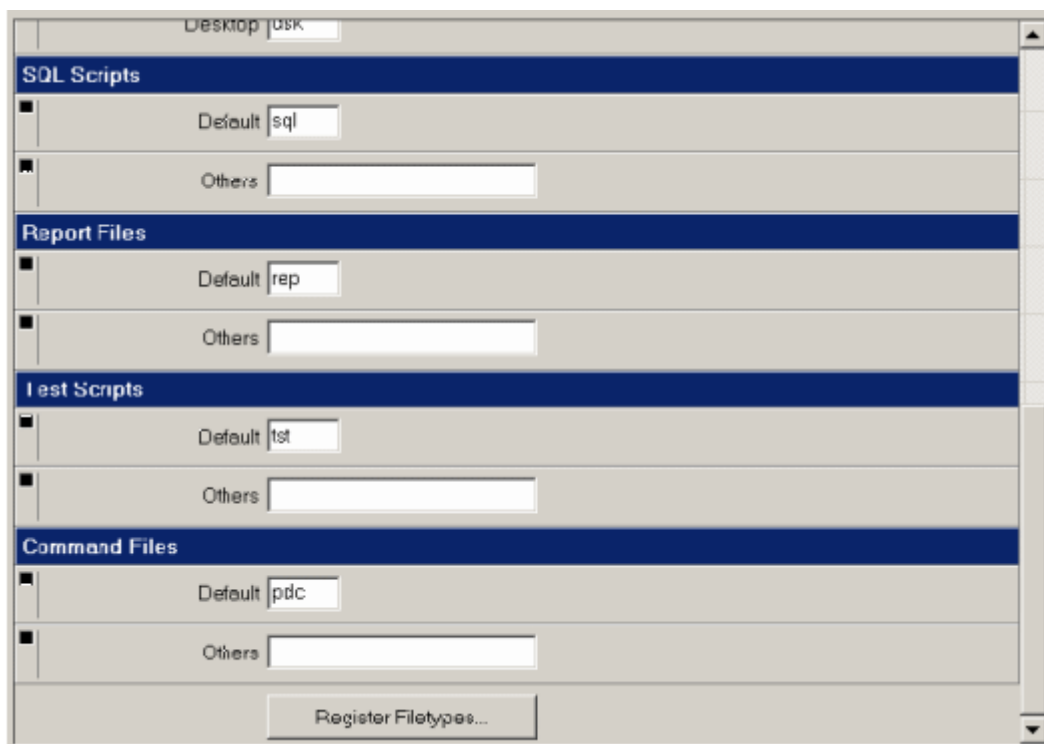
- 标准查询 (Standard queries)

定义保存标准查询的目录。如果你没有填写，标准查询将保存在你的 profile 的应用数据目录。

- OFS Locations

定义 OFS Location Directory File (ofs.ldr)。OFS 管理者(OFS Manager)按键允许你调用 OFS 管理者查看或者编辑这个文件。另见 OFS 管理者 (OFS Manager)

16.27 文件-扩展名 (Files – Extensions)



在这页，你可以定义所有文件类型的扩展名。当保存或者装载这样一个文件时，如果你在文件对话框中没有指定一个扩展名时，采用默认扩展名。对于每个文件类型，你可以额外指定'other'扩展名，这同样将被显示在文件对话框中。多个扩展名用逗号隔开。

通过按注册文件类型 (Register Filetypes) 键，你可以使这些默认扩展名和 PL/SQL Developer 关联。在 Windows Explorer 中双击带这个扩展名的文件将自动调用 PL/SQL Developer。一个对话框将显示让你选择哪个扩展名和 PL/SQL Developer 关联。

16.28 文件-格式 (Files – Format)

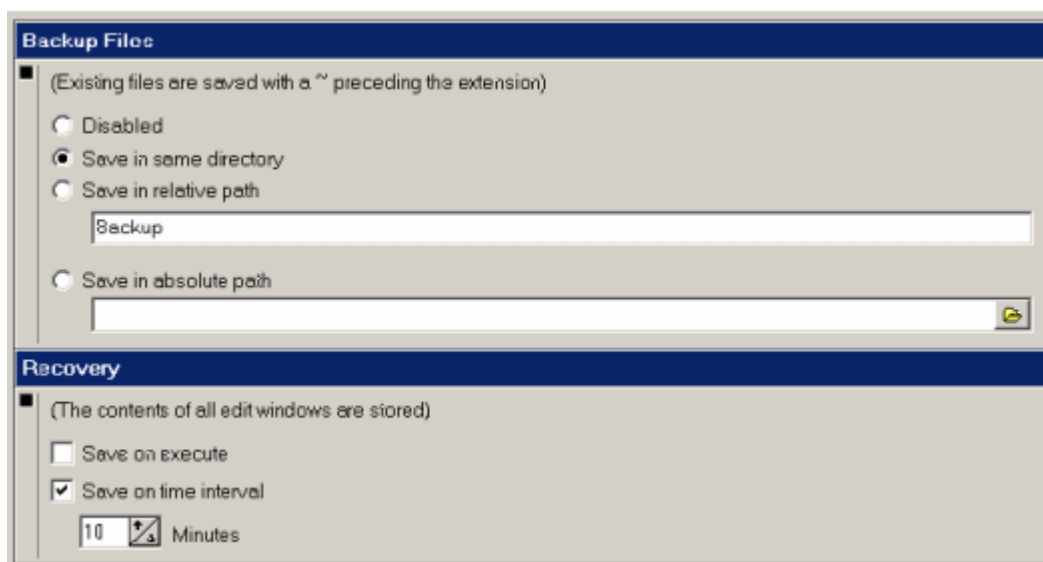


Linebreaks 选项控制在文本文件中的行是如何终止的。行既可以按 Windows 的样式以 CR/LF 字符对终止，也可以按 Unix/Linux 样式以 LF 字符终止。你也可以指定如果文件在特殊位置的是 Unix 格式，否则是 Windows 格式。你可以在不同行指定多个目录，按增加位置 (Add location) (...) 键选择目录。注意不是只保存在这个目录中的文件是 Unix 格式，而且下

面的子目录文件也都是 Unix 格式。

精准引号导出 (Export with smart quotes) 控制数值是如何导出到 CSV 格式。当被禁用, 所有值都使用引号括起来, 当启用, 只有需求的才使用引号括起来。需要括起来的是含逗号或者引号的值。

16.29 文件-备份 (Files – Backup)



这个参数选项也包含备份和恢复你的源文件选项。

创建备份文件 (Create backup files)

在 PL/SQL Developer 保存一个文件前, 原文件被覆盖前将被保存成一个备份文件。这个备份文件和原文件同名, 扩展名的前缀是~字符 (如 employee.pck 将被保存为 employee.~pck)。这样你就有一个以前文件的备份。

所有允许你打开一个文件的文件对话框都有一个 "Backup files (*.~*)" 过滤器允许你轻易地打开一个备份文件。

恢复 (Recovery)

PL/SQL Developer 可以在特定事件时保存所有打开的文件到一个临时位置。当一个服务器问题 (如断电, 软件崩溃, 等等), 当 PL/SQL Developer 重新启动后允许你恢复这些文件。你可以指定以下事件:

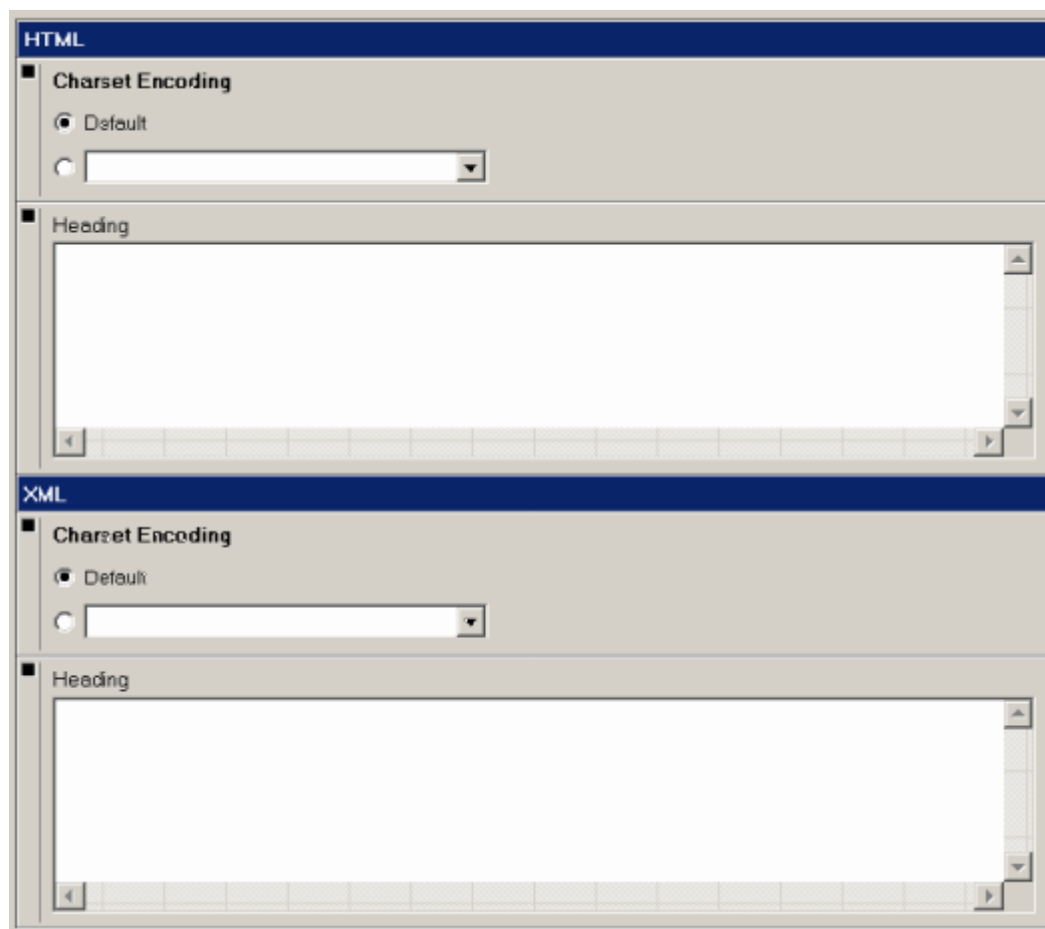
- 执行时保存 (Save on Execute)

当你执行一个程序窗口, 测试窗口, 命令窗口, SQL 窗口或者报表窗口前所有打开文件将被保存在临时目录中。

- 定时保存 (Save on time interval)

所有打开文件将在指定时间间隔被保存到临时目录中。

16.30 文件—HTML/XML (Files - HTML/XML)



这些参数允许你控制报表窗口和导出功能生成的 HTML 和 XML 输出。

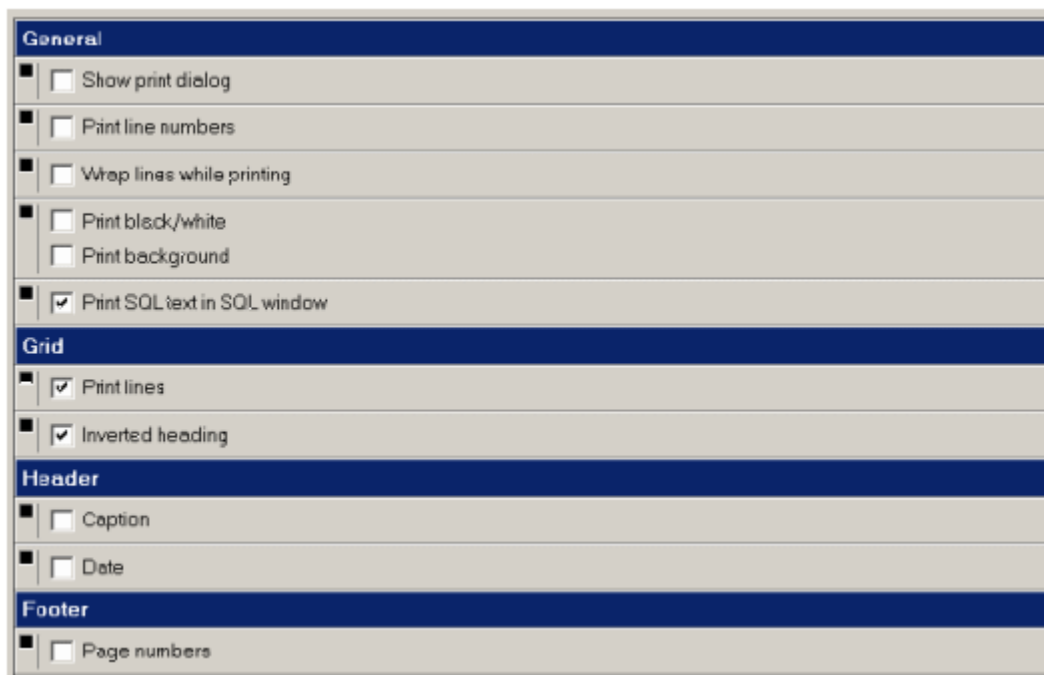
- 字符集翻译 (Charset encoding)

允许你定义 HTML 或者 XML 文档的字符集翻译。这必须符合你的 Oracle 环境字符集才可以保证数据被翻译正确。

- 头 (Heading)

这个文本将插入 HTML 文件的<HEAD>部分或者一个 XML 文件的标准 XML 头中。

16.31 其他—打印（Other – Printing）



常规（General）

- 显示打印对话框（Show print dialog）

当按打印键，将首先显示打印对话框。
- 打印行号（Print line numbers）

当打印 SQL 或者 PL/SQL 代码时，每行前显示一个行号。
- 当打印时自动换行（Wrap lines while printing）

当行超出页宽度时将在下一行继续。
- 黑白打印（Print black/white）

所有打印输出将是黑白的，禁用这个选项将以彩色或者灰度创建输出。
- 后台打印（Print background）

当被启用，背景颜色将被打印。
- 在 SQL 窗口中打印 SQL 文本（Print SQL text in SQL Window）

确定当打印一个 SQL 窗口内容时，是否在打印查询结果集前先打印 SQL 文本。

表格（Grid）

- 打印线（Print lines）

确定是否打印表格线。
- 反转标题（Inverted heading）

确定是否反转颜色打印表格标题。

报头 (Header)

- 标题 (Caption)

当被启用，将在每页报头左侧打印文件类型和文件名。

- 日期 (Date)

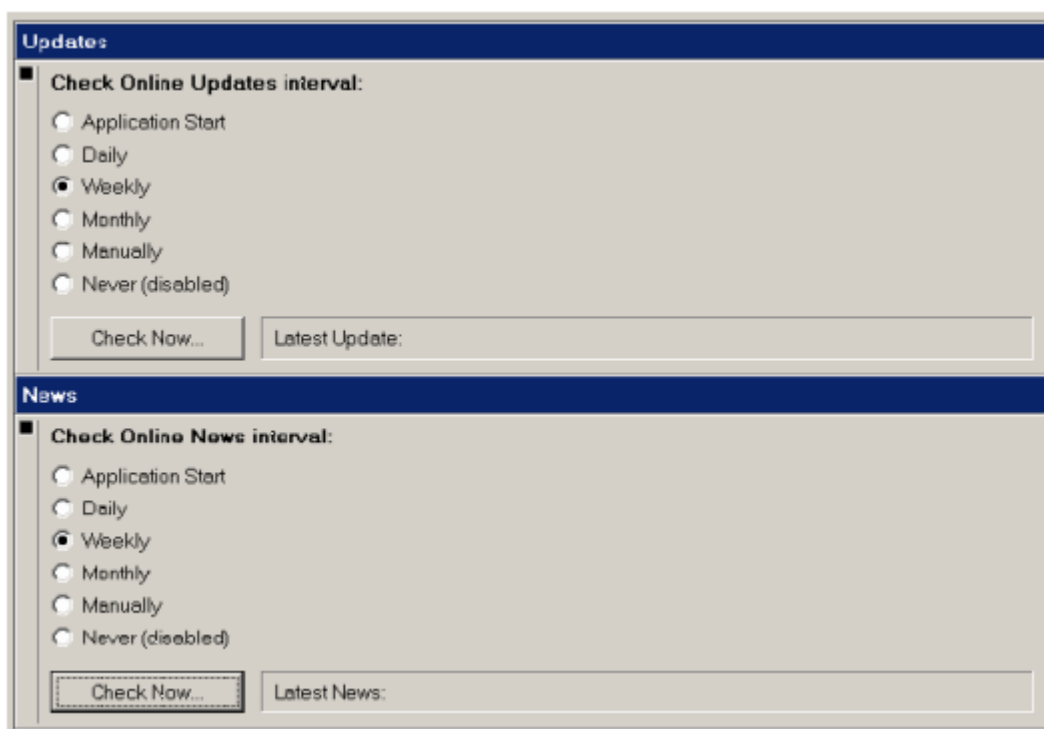
当被启用，将在每页报头右侧打印当前日期和时间。

页脚 (Footer)

- 页码 (Page numbers)

当被启用，页脚中间包含页码。

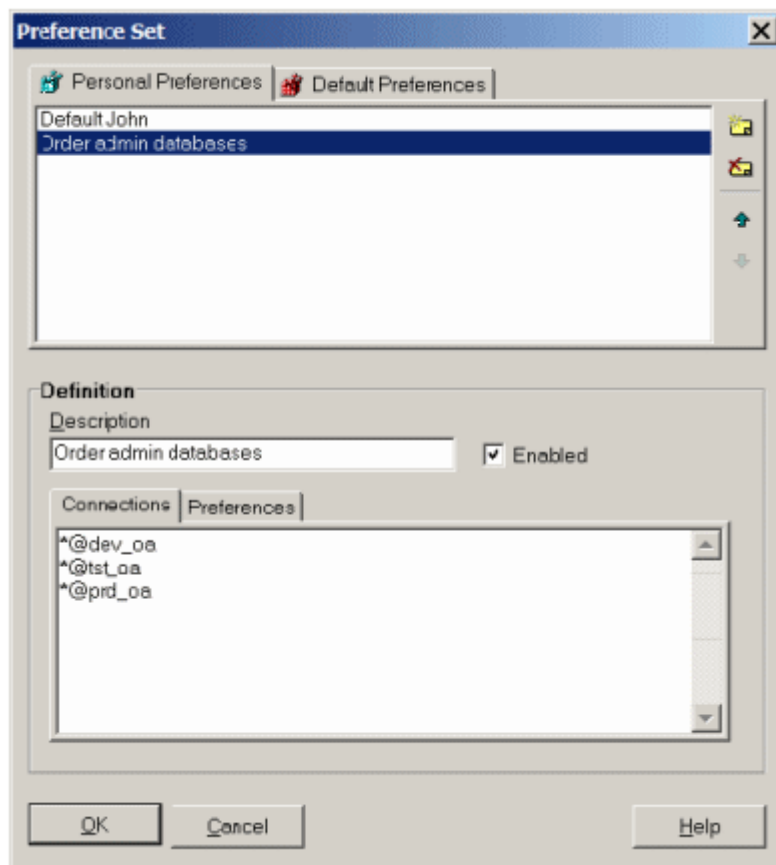
16.32 其他—更新和新闻 (Other - Updates & News)



允许你控制 PL/SQL Developer 是否和何时检查官方网站上有关 PL/SQL Developer 的更新和新闻。如果你禁用这个选项，你可以按相应的选项或者通过帮助菜单中的在线检查更新和在线检查新闻菜单项手工完成。

16.33 优先选择集 (Preference sets)

像在优先选择 (Preferences) 章节中提到的，你可以在不同级别上定义优先选择。要定义一个选项，在优先选择集选择列表旁边按设置优先选择集 (Configure preference sets) 键。将出现下面的对话框：



个人优先选择（Personal Preferences）定义当前 Windows 用户（在该情况下是 John）。系统管理员可以定义默认优先选择（Default Preferences），提供所有 Windows 用户的默认优先选择，这个优先选择既是一个全局级别的也是一个连接级别的。

个人优先选择（Personal preferences）

在默认情况下全部用户都有一个默认<用户>的优先选择集。这是你在默认情况下获得的优先选择集。要为一个或者多个 Oracle 连接额外创建一个优先选择集，在优先选择列表右侧按创建（New）键。现在你可以输入一个描述并增加一个或者多个连接。对于每个连接，你可以使用通配符来匹配多个用户或者数据库。在上面的例子中，John 定义了一个针对 dev_oa, tst_oa, 和 prd_oa 数据库的 Order admin database 优先选择集。如果一个优先选择没有在这个集中定义，将使用默认用户优先选择集中相应的优先选择。

默认优先选择（Default preferences）

一个系统管理员可以为全部用户 and 所有连接定义全局默认优先选择，可以额外为一个或者多个连接定义一个默认的优先选择集。要这样做，切换到默认优先选择（Default Preferences）标签页,在优先选择列表旁边按创建（New）。如果你没有输入任何连接，新的优先选择集将定义为全部用户的默认优先选择集。如果你输入一个或者多个连接，优先选择集将定义为这些连接的默认优先选择集。

优先选择集的优先级（Preference set precedence）

如果一个优先选择被定义多个级别，最高级别将得到优先：

1. 个人连接优先选择集（最高优先级）
2. 个人 Windows 用户优先选择集
3. 默认连接优先选择集
4. 默认全局优先选择集（最低优先级）

优先选择集文件和目录（Preference set files and directories）

一个优先选择集被保存在 PL/SQL Developer 安装目录下的 Preferences 子目录中的一个单独文件中。在这个优先选择集目录中，个人优先选择被保存在以 Windows 用户名命名的子目录中。例如：

```
C:\Program Files\PLSQL Developer\Preferences\John
```

默认优先选择被保存在 Default 子目录中，例如：

```
C:\Program Files\PLSQL Developer\Preferences\Default
```

如果 PL/SQL Developer 安装在文件服务机，你必须提供 Preferences 目录的读写权限，和 Preferences\Default 子目录的读权限。另一种方法是，你可以提供 PREFPATH 命令行参数，指明用户优先选择的目录：

```
plsqldev.exe prefpath=u:\userdata
```

如果 Windows 用户 John 使用 PL/SQL Developer，他的个人优先选择集将被保存在 u:\userdata\john。默认优先选择已经保存在 PL/SQL Developer 安装目录下的 Preferences\Default 子目录中。

为确保全部用户使用一个相似的优先选择路径，你可以在 params.ini 文件中设置这个参数（见命令行参数章节（Command-line parameters））。

17 工具 (Tools)

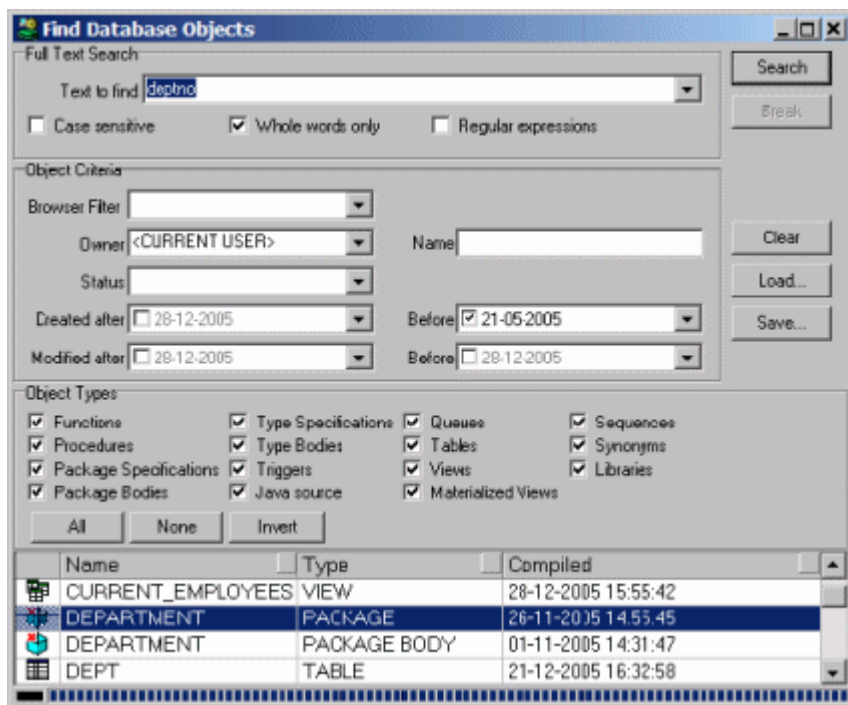
许多工具在工具菜单中可以使用。这些工具将在本章描述。

17.1 浏览器 (Browser)

浏览器工具已经在浏览对象 (Browsing objects) 章节中详细描述。在工具菜单中，你可以通过选择浏览器 (Browser) 菜单项显示或者隐藏浏览器。通过这个菜单，你也可以定义浏览器过滤器，在浏览对象 (Browsing objects) 章节中也已详细描述。

17.2 查找数据库对象 (Find Database Objects)

这个工具允许你基于指定的限制条件查询数据库对象，如对象名称、类型、宿主或者状态，以及附加的针对对象 DDL 源代码的全文搜索限制：



你可以在这个表单上 3 部分输入查询限制。

全文搜索（Full Text Search）

这个对话框的选项或多或少等同于文本编辑器中的查找对话框，可以选择区分大小写，全字匹配，和规则表达式。如果你忽略全文搜索，则符合其他限制条件的全部对象将返回。

对象限制（Object Criteria）

在这里你可以输入以下限制：

- 浏览器过滤器（Browser Filter）— 根据选择的过滤器限制对象。
- 宿主（Owner）— 对象的拥有者，你可以使用 SQL 通配符（如 sys%）。默认值是<CURRENT USER>，这将限制只查找当前用户的对象。
- 名称（Name）— 对象名。你可以使用 SQL 通配符（如%common）。
- 状态（Status）— 限制查找有效或者无效对象。
- 被创建之后/之前（Created after/before）— 限制查找创建日期在指定日期之前或者之后。使用日期字段的左侧复选框启用或者禁用这些限制。
- 被修改之后/之前（Modified after/before）— 限制查找对象修改日期在指定日期之前或者之后。使用日期字段的左侧复选框启用或者禁用这些限制。

对象类型（Object Types）

在这里，你可以选择要查找的对象类型范围。

查找结果（Search results）

按查找键后，查找操作在后台被执行并且填写查找结果列表

如果你在查找结果中右键点击，你将看到一个对象的弹出菜单。弹出菜单中的第一个菜单项是定位文本（Locate text），在一个程序窗口中打开源代码，并且在文本中定位第一个出现查询文本的位置。你也可以在查找结果中通过双击调用定位文本（Locate tex）。

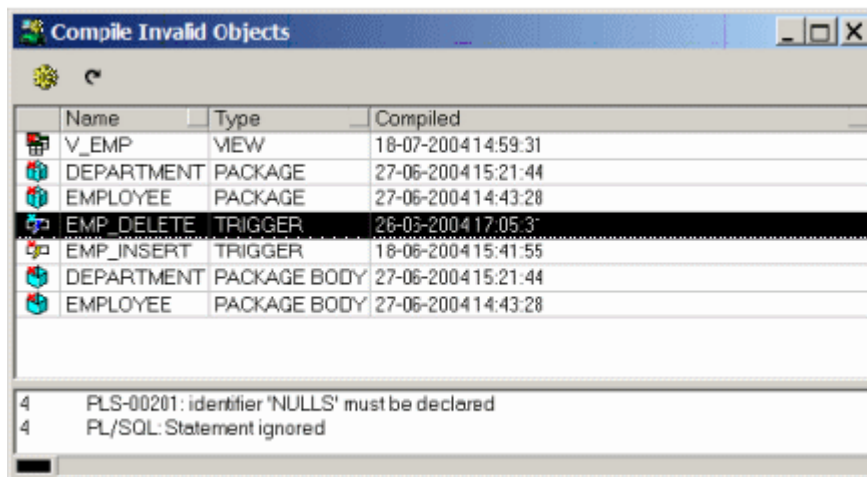
要保存查找结果，你可以右键点击它们并且从弹出菜单中选择导出（Export）菜单项，选择一个相应的格式。

保持经常使用的查找限制（Saving frequently used search criteria）

要保持经常使用的查找限制，你可以按保存（Save）键并输入一个目录和文件名。你可以按载入（Load）键重新调用这些查找限制。

17.3 编译无效对象（Compile Invalid Objects）

使用这个工具你可以很容易地重新编译数据库中所有无效的对象。当你选择它，将显示一个全部无效对象的对话框。



注意，当前活动的浏览器过滤器（Browser Filter）控制包含的无效对象。对话框的底部将显示当前选择的对象上次编译的全部错误。

要重新编译对象，按工具条上的编译无效对象（Compile invalid objects button）键或者主工具条中的执行（Execute）键。对象间的依赖关系将被分析来确定编译顺序。对象列表左侧的图标指示哪些对象被成功编译。编译错误部分将在编译后被更新。

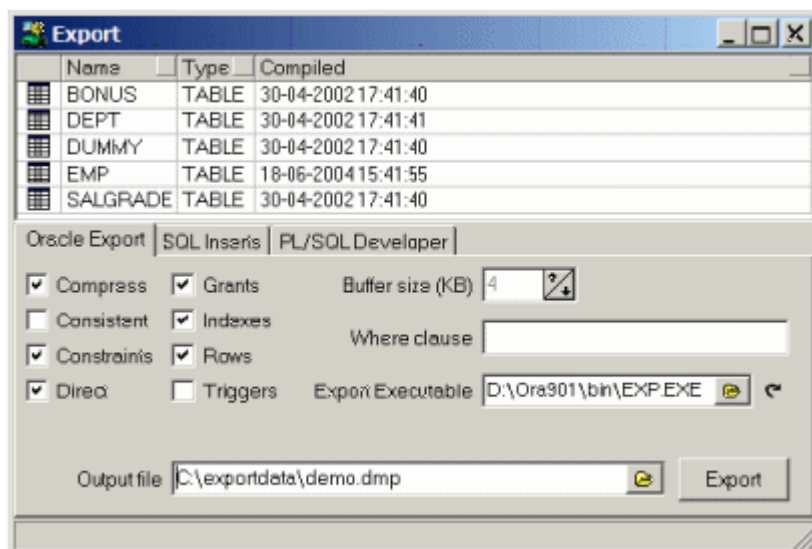
要刷新无效对象列表，按工具条中的刷新对象列表（Refresh object list）键。这将根据当前所有无效对象更新列表。

在对象列表中你可以右键点击一个对象获得它的弹出菜单。

17.4 导出表（Export Tables）

导出表工具允许你导出一个或者多个表定义和它们的数据到一个文件中，这样就可以在以后导入表。启动导出表工具后，你可以选择你希望导出的表，选择一个导出方法（Oracle

Export, SQL Inserts, 或者 PL/SQL Developer），并且设置不同的选项应用到导出方法：



导出功能分析表之间可能存在的外键依赖关系，并按正确的顺序写到输出文件。

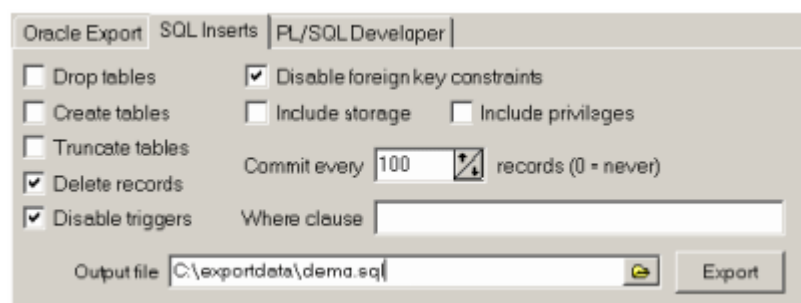
Oracle 导出方法（Oracle Export method）

上面的屏幕截图显示 Oracle Export method 的选项。用这种方法让你指定 Oracle 命令行导出工具的选项（见"Oracle Server Utilities"手册），并将随后使用它执行导出。导出执行程序（Export Executable）输入框允许你选择一个指定版本的 Oracle 导出工具。默认情况下，当前 Oracle 主目录（Oracle Home）中的最新版本将被使用。

导出完成后，将显示一个新的日志（Log）标签页。这页包含导出工具创建的日志。

这种方法的优点是速度和导出文件的兼容性。毕竟，你可以在任何地方使用 Oracle 的导入工具都可以导入它。

SQL 插入方法（SQL Inserts method）



这种导出方法创建一个标准的 SQL 脚本，包含插入语句和（可选）DDL 语句重新创建表定义。标签页包含以下的选项：

- 删除表（Drop tables）— 生成的 SQL 脚本中包含在重建表和载入数据之前的删除表语句。在这种情况下，创建表（Create tables），截去表（Truncate tables）和删除记录（Delete records）选项被禁止，并有一个固定值。

- 创建表（Create tables）— 生成的 SQL 脚本包含载入数据前的创建表语句。包括限制，索引和授权。
- 截去表（Truncate tables）— 生成的 SQL 脚本在载入数据前使用截去表（truncate table）语句清空表。这个选项比 Delete 表快，但是如果有外键存在时不能使用。
- 删除记录（Delete records）— 生成的 SQL 脚本在载入数据前使用删除语句清空表。
- 禁用触发器（Disable triggers）— 生成的 SQL 脚本在载入数据前禁用表中的所有触发器，并在之后启用。这可以改善性能，有时也是必须的，如触发器包含了导入过程不支持的检查。
- 禁用外键限制（Disable foreign key constraints）— 生成的 SQL 脚本在载入数据前禁用表的所有外键，并在之后启用。这对于自引用外键是必须的。不同表之间的外键不会冲突，因为表被按正确顺序导出。禁用外键也可以改善性能。
- 包含存储（Include storage）— 生成的 SQL 脚本包含表创建语句的原存储信息，如表空间名和初始大小。在跨数据库时这些信息可能不一致，所以，这些信息不一定总是合适的。
- 包含权限（Include privileges）— 生成的 SQL 脚本包含授予其他用户和角色的对象权限。当对象在不同的数据库中创建时，这些用户和角色必须存在。
- 提交每（Commit every）— 控制生成的 SQL 脚本在插入多少记录后执行提交事务操作。如果这个值是 0，在 SQL 脚本最后，全部记录插入后一次性提交事务。对于大的导出文件和小的回滚段，在这里必须提供一个合适值。
- 条件子句（Where clause）— 只有满足条件子句的记录将被导出。条件子句必须应适用于所有选择的表！

这种方法的优点是 SQL 脚本的兼容性，并且可以使用一个文字编辑器编辑。

PL/SQL Developer 方法（PL/SQL Developer method）



这种导出方法创建一个自定义，压缩的 PL/SQL Developer 导出文件，包含表定义和数据。标签页包含以下选项：

- 条件子句（Where clause）— 只有满足条件子句的记录将被导出。条件子句必须满足所有选中的表。

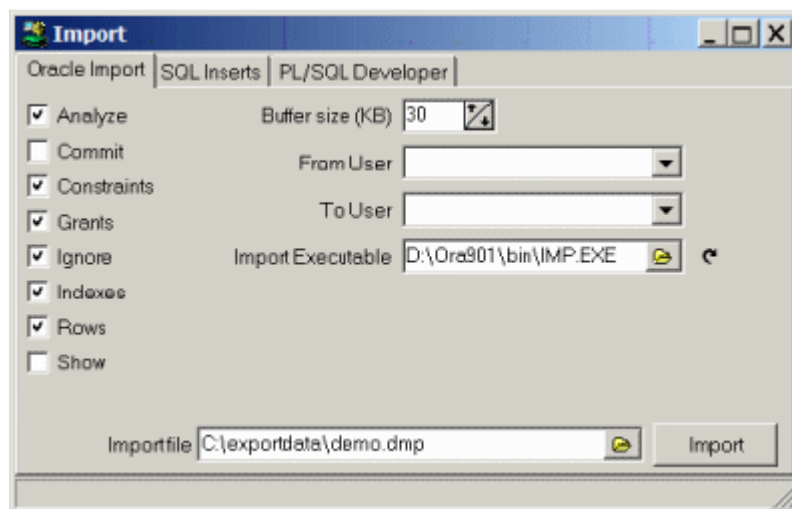
- 压缩文件（Compress file）— 禁用这个选项将禁用压缩。这可以改进性能如果数据不能被很好压缩（如对于 long raw 或者 BLOB 数据，这些数据已经在数据库中被压缩）
- 包含存储（Include storage）— 生成的 SQL 脚本包含表创建语句的原存储信息，如表空间名和初始大小。在跨数据库时这些信息可能不一致，所以，这些信息不一定总是合适的。
- 包含权限（Include privileges）— 生成的 SQL 脚本包含授予其他用户和角色的对象权限。当对象在不同的数据库中创建时，这些用户和角色必须存在。

这种方法的优点是极大地减少了导出文件的大小，事实上，所有选项被延期到导入阶段，并且在导入时可以方便地选择表。

17.5 导入表（Import Tables）

导入表工具允许你从以前使用导出表工具导出的文件导入表定义和数据，导出表工具在上节描述。就像导出表工具，有 3 种方法导入表，每种有自己的文件格式。

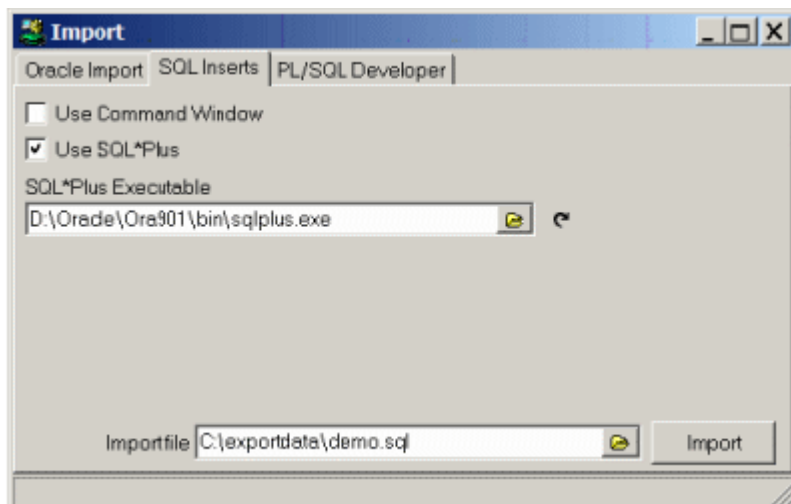
Oracle 导入方法（Oracle Import method）



这种方法使用一个 Oracle 导出工具创建的导出文件，并且调用 Oracle 导入工具导入数据。用这种方法让你指定 Oracle 命令行导入工具的选项（见"Oracle Server Utilities"手册），导入执行程序（Import Executable）输入框允许你选择一个指定版本的 Oracle 导入工具。默认情况下，当前 Oracle 主目录（Oracle Home）中的最新版本将被使用。

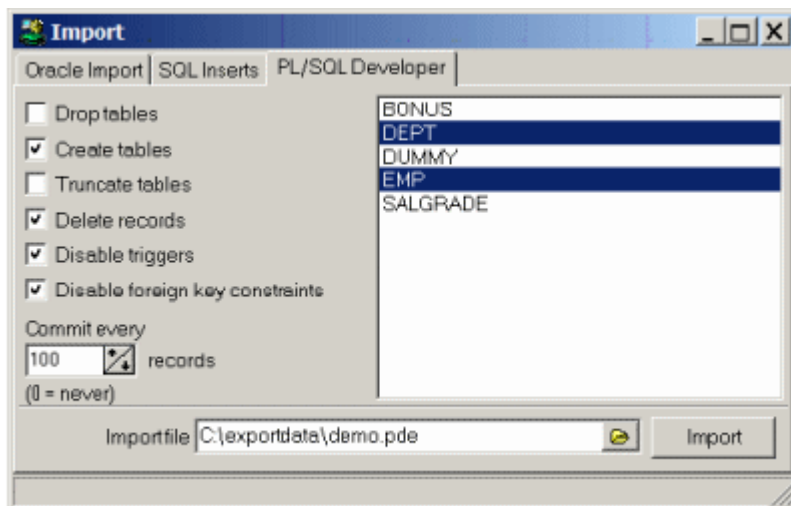
导入完成后，将看到一个新的日志（Log）标签页。这个页包含导入工具创建的日志。

SQL 插入方法（SQL Inserts method）



这种方法执行导出表工具生成的 SQL 脚本。你可以选择装载 SQL*Plus 运行脚本，或者用一个内部命令窗口。对于 SQL*Plus 你可以选择一个特殊 SQL*Plus 执行程序。默认情况下，当前 Oracle 主目录中的版本将被使用。

PL/SQL Developer 方法（PL/SQL Developer method）



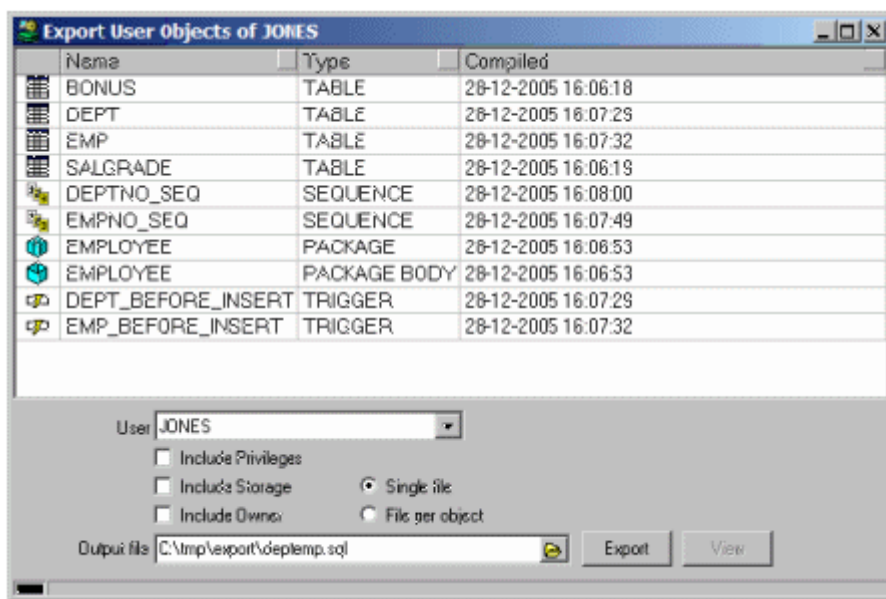
选择一个由 PL/SQL Developer 方法导出的导入文件后，你可以从列表选择一个或者多个文件。可以指定下面的选项：

- 删除表（Drop tables）— 生成的 SQL 脚本中包含重新创建表和载入数据前删除表的语句。在这种情况下，创建表（Create tables），截去表（Truncate tables）和删除记录（Delete records）选项被禁止，并有一个固定值。
- 创建表（Create tables）— 生成的 SQL 脚本包含载入数据前的创建表语句。包括限制，索引和授权。
- 截去表（Truncate tables）— 生成的 SQL 脚本在载入数据前使用截去表（truncate table）语句清空表。这个选项比 Delete 表快，但是如果有外键存在时不能使用。

- 删除记录 (Delete records) — 生成的 SQL 脚本在载入数据前使用删除语句清空表。
- 禁用触发器 (Disable triggers) — 生成的 SQL 脚本在载入数据前禁用表中的所有触发器，并在之后启用。这可以改善性能，有时也是必须的，如触发器包含了导入过程不支持的检查。
- 禁用外键限制 (Disable foreign key constraints) — 生成的 SQL 脚本在载入数据前禁用表的所有外键，并在之后启用。这对于自引用外键是必须的。不同表之间的外键不会冲突，因为表被按正确顺序导出。禁用外键也可以改善性能。
- 提交每 (Commit every) — 控制生成的 SQL 脚本在插入多少记录后执行提交事务操作。如果这个值是 0，在 SQL 脚本最后，全部记录插入后一次性提交事务。对于大的导出文件和小的回滚段，在这里必须提供一个合适值。

17.6 导出用户对象 (Export User Objects)

为导出一个用户的所有对象的 DDL(数据定义语言 (Data Definition Language)) 语句，你可以使用导出用户对象工具。这样你可以很容易地为其他用户，或者在一个不同数据库重建对象。在工具菜单选择导出用户对象后，当前用户的所有对象将在一个表格中呈现：



选择一个输出文件后，你可以按导出 (Export) 键导出对象。如果在表格中没有明确选择的对象，所有对象将被导出。你可以点击对象选择它们，并使用 Control 和 Shift 键选择多个对象和对象范围。

导出文件是一个 SQL 脚本，和 Oracle 的 SQL*Plus 兼容，并由 PL/SQL Developer 的命令窗口使用。导出操作后你可以按查看 (View) 键在一个命令窗口中打开生成的文件。

选择一个不同用户，如果当前用户有权限，可以导出其他用户拥有的对象。

如果你想包含授予其他用户和角色的对象权限，可以启用包含权限选项。当对象在一

个不同数据库中重建，这些用户和角色必须存在。

类似的，你可以启用包含存储（Include Storage）选项包含如一个表空间名和初始大小的存储信息。这些信息在跨数据库时可能不同，所以这些信息可能不是总适合。

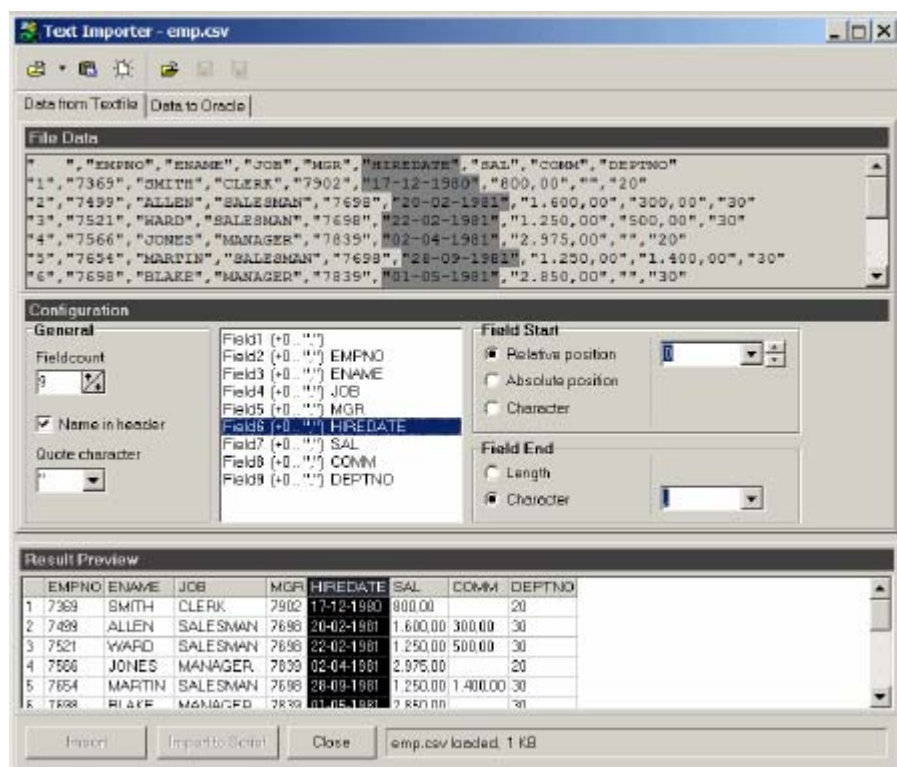
包含宿主（Include Owner）选项控制在作为结果输出文件中的对象是否带对象拥有者前缀（如 SCOTT.EMP 代替 EMP）。

单文件和每个对象一个文件（Single file and File per object）选项控制是否一个单一 SQL 脚本包含所有对象的 DDL，或者每个对象导出一个特殊文件（文件名同对象名，并且扩展名是.sql）。这样，输出文件将包含一系列调用这些对象的特殊文件。所有文件将被写入输出文件目录。

17.7 文本导入器（Text Importer）

你可以使用文本导入器导入 ASCII 文件到数据库中。支持多数行定位格式，像逗号和制表符分割字段。导入器将自动设法确定文件格式，所以大多情况下你不需要定义任何事情，只要选择文件，选择一个表，仅此而已。

当你打开文本导入器并装载一个文本文件，你将获得如下一些东西：



导入器首先从文件中装载 100 行，原始数据在上方的“文件数据”（"File Data"）部分显示。中间部分允许你对文本文件作出定义。底部“结果预览”（"Result Preview"）有一个将要被导入的数据表格。

工具条中有选择一个文本文件或者从剪贴板粘贴文本的按钮。文件没有大小限制，它

们并不是整体被读入内存。"New"键将清空文本导入器中的任何事物。打开和保存定义按键允许你重用定义。

一旦你载入一个文件，设置就自动被确定，但是如果设置不正确，你可以自己创建和修改。设置正确的字段数，从字段列表选择一个字段来定义一个字段定义。在文件数据部分，高亮指定字段数据，允许你检查字段定义是否正确。你有以下选项配置文本导入定义：

-常规—字段数 (General—Fieldcount)

每个记录的字段数。

-常规—标题名 (General—Name in Header)

标识第一行是否为字段名

-常规—引号字符 (General —Quote Character)

标识字符串字符，一般为单引号或者双引号。

-字段开始—相对位置 (Field Start—Relative Position)

字段开始于前一个字段结束的相对位置，0 表示字段开始于前一个字段的结束。

-字段开始—绝对位置 (Field Start—Absolute Position)

标识字段开始于固定位置。

-字段开始—字符 (Field Start—Character)

标识字段起始于一个指定字符。这是相对于前一字段的结束。

-字段结束—长度 (Field End—Length)

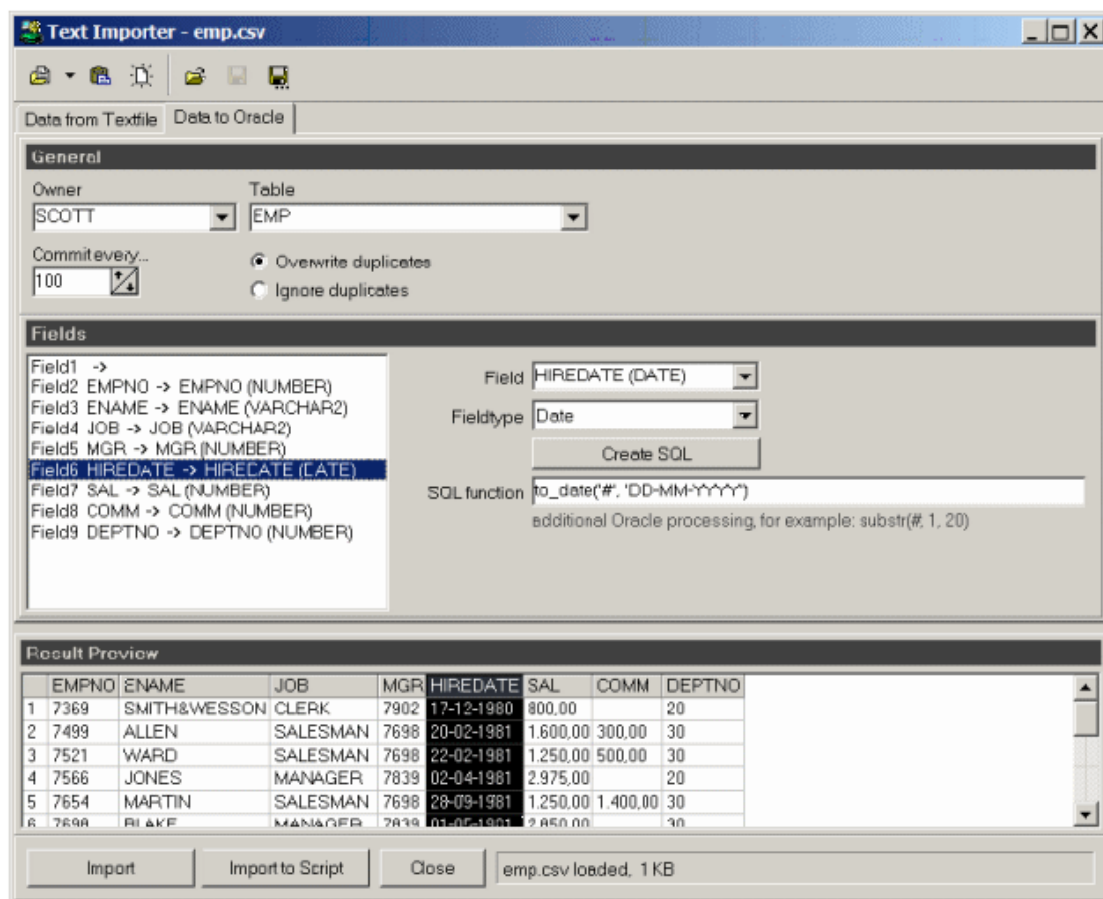
指出字段有一个固定长度。

-字段结束—字符 (Field End—Character)

标识字段结束于一个指定字符。

对于标准的使用逗号分开的文件 (csv)，所有字段将有一个 0 的相对起始位置 (relative start position)，结束字段的结束字符应该是一个逗号。行结束(cr/lf 或 lf 或 cr)将也标识字段结束，所以，对于一行的最后字段，结束字符并不重要。

当你有一个正确的文件定义，你需要选择一个 Oracle 表和指出哪些文本字段应该导入到哪些 Oracle 字段。你可以在第二个标签页中做这个事情：



在顶部有一些常规的导入参数。"Fields"部分允许你关联文本字段和 Oracle 字段。底部是你的信息结果预览。

你可以设置下面的常规参数：

-表 (Table)

你想导入数据到 Oracle 表 (或视图)。选中一个表后，导入器自动尝试确定字段和字段类型。只有在文本文件有一个标题时才可以进行。

-提交事务每 (Commit every...)

指明多少记录以后你想提交事务。如果你设置 0，全部导入后全部数据被提交。

-覆盖重复 (Overwrite Duplicates)

在数据库中重复数据将被更新。

-忽略重复 (Ignore Duplicates)

重复数据将被忽略。

在字段定义中，你看到在文本文件中的字段列表。对于每个字段你可以定义以下信息：

-字段 (Field)

Oracle 字段要对应的文本字段。如果你不想导入这个字段，可以不填写这个信息。

-字段类型 (Fieldtype)

基础字段类型（The basic fieldtype）：字符串（String），数字（Number）或日期（Date）。

-SQL 函数（SQL function）

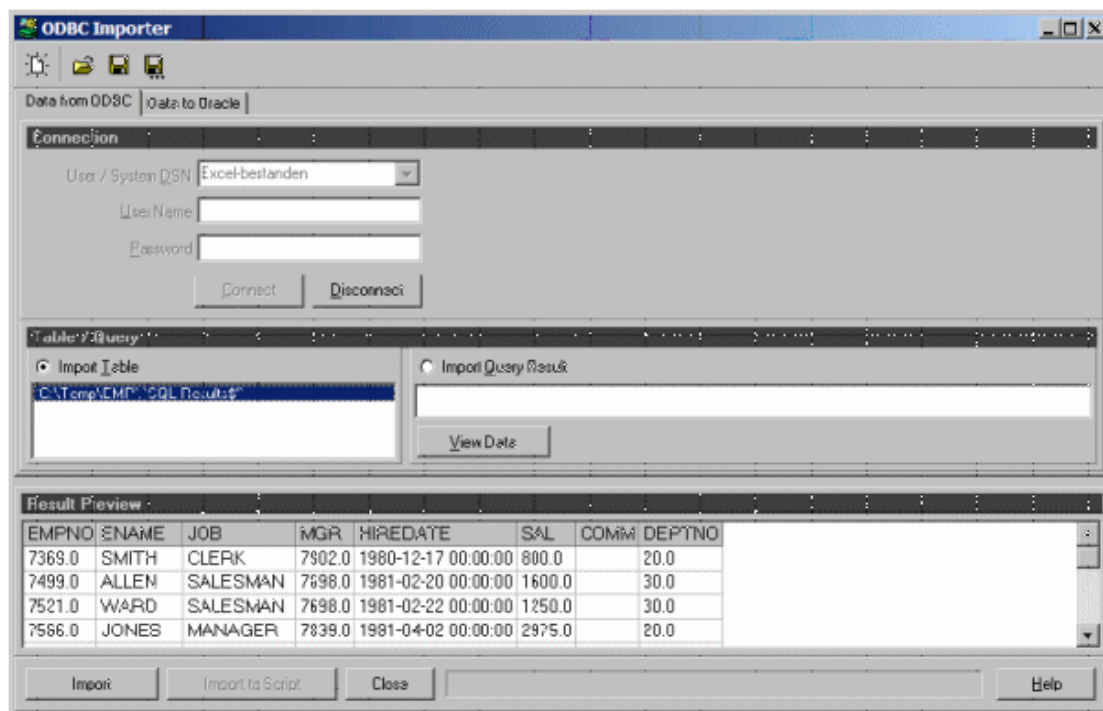
这个选项允许你定义额外的 SQL 处理。对于日期字段，一个 to_date 函数被自动添加。这是一个非常强大的选项，允许你转化导入数据。基本上，你可以输入 Oracle 可以处理的任何事情。你可以添加一个#代表数据。创建 SQL（Create SQL）按钮将为日期型字段填充 to_date 函数。

当定义完成后，你可以决定保存它。工具条有按钮保存和载入定义文件。文本导入器将记住哪个定义文件被哪个文本文件使用，当下次打开相同的文本文件时，定义文件将被自动载入。

你有两个导入键，"Import" 和 "Import to Script"。第一个键将导入数据到选中的表。第二个将使用插入语句创建一个 SQL 脚本。

17.8 ODBC 导入器（ODBC Importer）

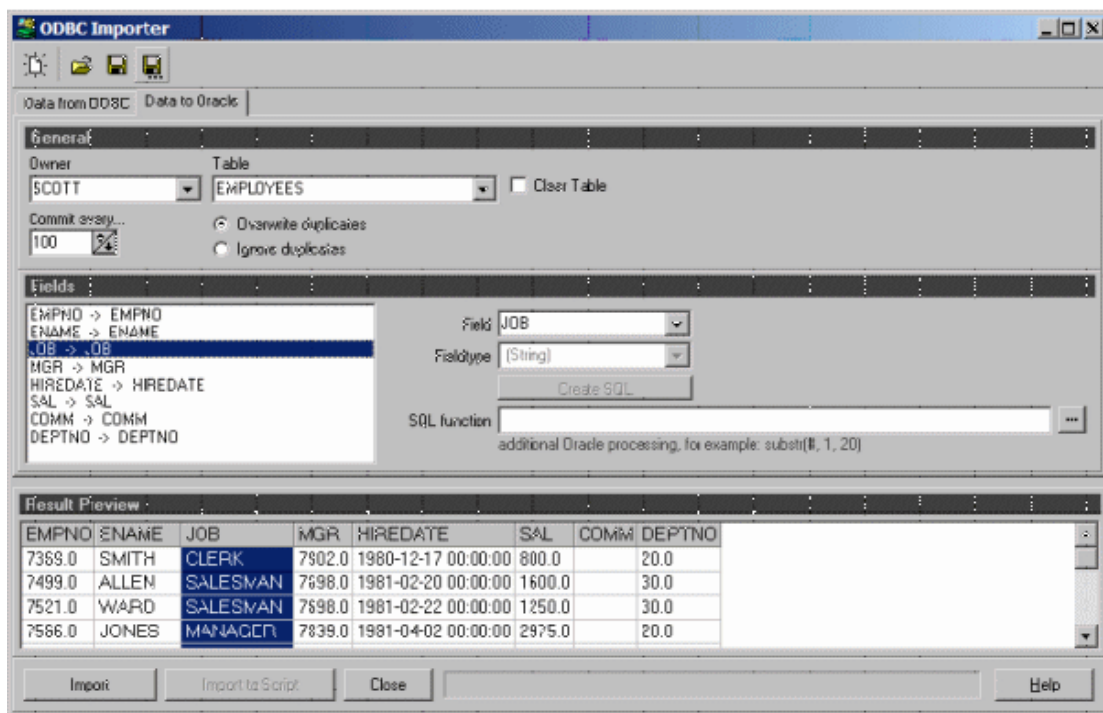
ODBC 导入器工具允许你从任何 ODBC 数据源导入数据到 Oracle 表：



在 ODBC 导入器第一页你需要选择 ODBC 数据源，并且可选的提供一个用户名和密码。你可以随后连接，之后，数据源中的全部表将装载到表列表中。选择一个表将填充结果预览窗格。替代选择一个表，你也可以选择导入查询结果（Import Query Result）选项，输入一个查询文本。这对于你想限制记录行或者字段，或者从多张表查询结果很有用。

输入数据源规格后，你可以切换到 Oracle 的数据（Data to Oracle）标签页，选择目的

表和确定字段对应：



在常规部分，你可以选择将被导入的一个宿主和一张表。你也可以选择导入前清空表，选项一个提交事务间隔（0=结束时提交事务）和忽略重复字段或者覆盖他们。

在字段（Fields）部分，你可以对应 ODBC 数据源（左）和 Oracle 表（右）的字段。每个字段可以额外增加一个 SQL 函数。例如，如果你想转换一个字符串字段成大写，输入 upper(#)。这个#号将被每条记录的字段数据所替代，并且表达式结果应该插入到语句中。

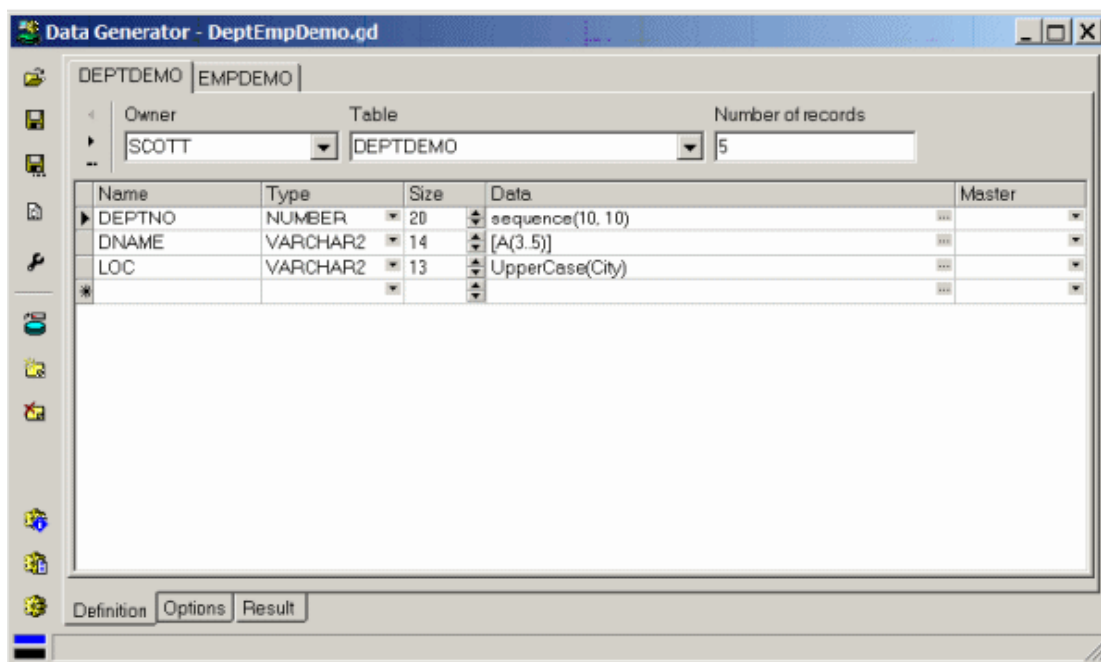
完成字段对应后，你可以按导入键导入数据到 Oracle 表中。如果你按键时按 Ctrl 键，一个插入语句例子将被显示，这样可以检验你指定的字段对应和 SQL 函数。

以后要重用这个导入定义，你可以按工具条中的保存定义（Save Definition）键，这个定义今后可以按打开定义（Open Definition）键打开。

17.9 数据生成器（Data Generator）

数据生成器允许你创建 demo 和测试数据。这可以帮助测试应用并看出在大数据量时的运行情况。你可以在工具菜单下找到数据生成器。

基本上，定义由一个或者多个表、你想生成的记录数、和字段数据定义组成。



你从上面可以看到提供的 deptemp 演示，创建 deptdemo 和 empdemo 表数据，和熟悉的 dept 和 emp 表类似。在这页你看到以下项：

- 表（Table）— 表名。
- 记录数（Number of records）— 你想生成的记录数。这可以是一个数字，也可以是范围，如 10..100。
- 名称（Name）— 字段名称。
- 类型（Type）— 字段的数据类型。
- 大小（Size）— 字段大小（适当的）。对数字字段，这将是范围，精度。
- 数据（Data）— 一个字段的数据定义（见下）
- 主表（Master）— 如果这张表是其他表的子表，你可以设置主表。对应每个生成的主表记录，指定的子表记录将被生成。

你可以使用 "Add existing table" 键添加一个已存在表。你也可以从对象浏览器中拖放一个表。

数据定义（Data definition）

数据定义决定生成数据。如果你想创建简单字符，你可以在方括号中输入字符定义：[数据]

数据可以混合下面的预定义集：

- a: a..z (小写字符)
- A: A..Z (大写字符)
- @: A..Z 和 a..z (所有字符)

- #: A..Z 和 a..z 和 0..9 (所有字符和数字)
- *: #33..#126 (所有 ASCII 字符)
- 0: 0..9 (所有数字)
- 1: 1..9 (除 0 外所有数字)
- 9: 0..9 (所有数字)

例如:

[Aaa00]生成的字符串像: Gxe21, Liy05, 等。

你也可以在单引号中增加文字正文 (literal text)。

例如:

[AA '-' 1000] 生成的字符串像: CX-4903, SY-1044, 等。

在定义中的空格字符被忽略, 除非在引号中。

如果你想一个字符重复多次, 你可以在字符后增加数字, 用括号括起来(n)。你也可以增加一个随机数(min..max)。

例如:

[Aa(5..15) ' ' Aa(8..20)] 结果像: "Masfae Qwwecdsadif"

在定义中的文本正文不必用括号括起来, 换句话说, ['hello'] 和 'hello'是一样的。没有引号的正文被解释为一个函数。

有许多可用的特殊函数:

- Signal(Min, Max, Delta, Noise)

返回技术测量数据 (像温度)。Min 和 Max 确定范围, Delta 是最大变化, 你也可以增加一些噪声。例如: Signal(-10, 20, 0.1, 0.1)

- Random([Min], Max)

返回 Min 和 Max 之间的随机数。如果只指定了 Max, Min 设置为 0。对于日期字段你可以在 Min 和 Max 上输入日期。

- Sequence(Start, [Inc], [WithinParent])

返回序列数字, 以 Start 开始, 以 Inc 递增 (默认 1)。对应一个明细表 (子表), 你可以额外指定 WithinParent 关键字表示对于每个父表记录重置序列。

- List('item'(weight), 'item'(weight), ...)

随机返回一个指定项目。weight (权重) 可以在括号中增加, 允许指定特殊项目比其他项目更经常发生。

例如: List('CLERK'(50), 'SALESMAN'(30), 'MANAGER'(10))

- List(select statement)

如前面的 List 函数，但是项目是由 SQL 查询语句返回的。

- Text([Style], MaxCharacters, [WordsPerLine, LinesPerParagraph])

这个函数返回文字。Style 参数可以是 LorumIpsum (默认), English, German 或 Japanese。它也可以是一个字符集像[aA]，这样的词从指定集中生成。

MaxCharacters 确定生成文字的最大值大小，选项 WordsPerLine 和 LinesPerParagraph 确定一行和一段的大小。指定大小可以是一个数字范围 (min..max)。

- File(path, path, ...)

从给定的目录随机选择一个文件，并插入内容。允许你输入二进制数据（像图形）到数据库。Path 可以有通配符，像 d:\images*.bmp。

你可以使用下面函数改变上面函数的文字结果：

- Uppercase()
- LowerCase()
- InitCaps()

例如：InitCaps(List(select ename from emp))

还有一些预定义的数据集适用于生成或多或少数据，你可以使用下面定义：

- Firstname — 名一览表
- Lastname — 姓一览表
- Company — 公司名（存在公司的随机列表）
- Address1 — 地址线 1
- Address2 — 地址线 2
- ZIP — 邮政编码
- City — 城市
- State — 省
- Country — 国家
- Email (对应名，姓，和国家)

还有一些可适用的用户定义数据集范例：

- Components.Code — 普通商品项目：商品代码
- Components.Description — 商品描述（计算机部分）
- Components.Price — 商品价格
- Elements.Name — 化学元素（名称）

- Elements.Symbol—化学元素（符号）

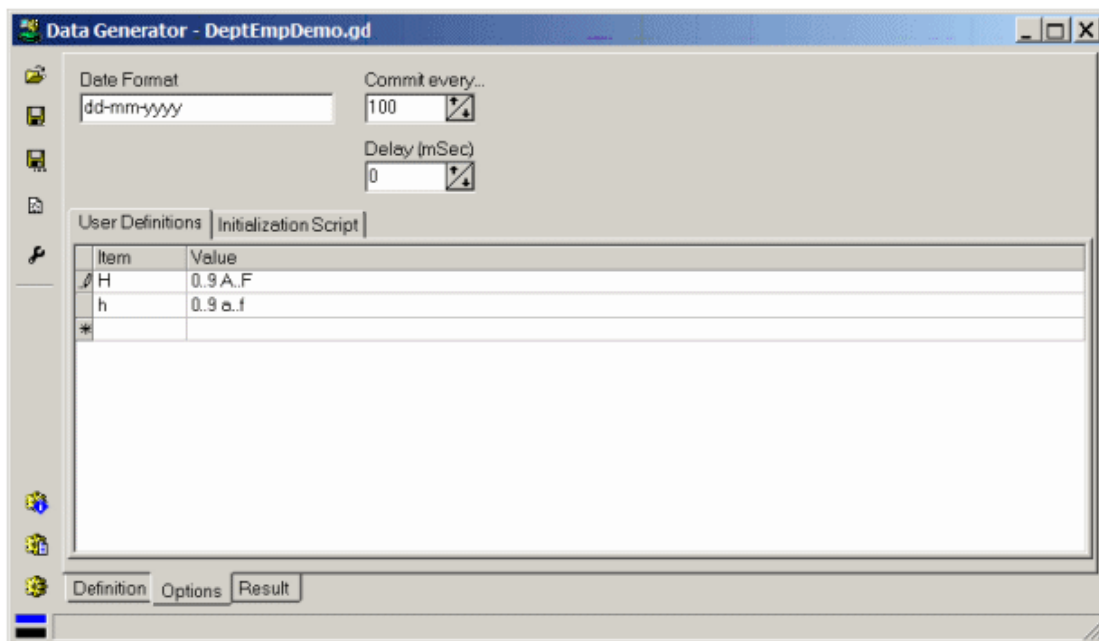
这些数据集在 DataGenerator\UserData 目录下以 elements.txt 和 components.txt 的方式存在。如果你想的话，可以增加你自己的集，只要增加一个以逗号分割的文件，第一行在方括号中写上描述。你可以像两个例子一样，通过指定 filename.description 从你的文件中使用数据。

上面提到的所有函数和数据可以添加到一起，例如：Random(10..99) + '-' + [A(4)]

+是可选的，但至少有一个空格分割。

选项（Options）

选项标签页允许你设置输入生成器定义的一些参数。你可以指定一个日期格式（同在 Random 函数的例子）。你也可以指定多少记录后提交事务（设置 0 为完成后提交事务一次）。延迟参数只有在一些实时测试中当你想让数据插入按照一个指定速度时有用。



"User Definitions"部分允许你定义或者覆盖在[data]定义的字符集。Item 总是一个字符，Value 是一个或者多个空格分割的字符范围像：A..Z a..z 0..9 #200..#220。例如：定义一个 16 进制字符集，你应该指定 H 作为 Item，并且 0..9 A..F 作为 Value。

"Initialization Script"部分允许你指定 SQL 语句，在数据被插入数据库之前执行。一般这是一个创建表，或者截去表，查询一个回滚段，等等语句，多个语句需要使用分号分割。

被用作所有数据生成器的默认常规选项可以按左侧的参数选择（Preferences）键。详见工具—数据生成器章节。如果相应的参数在数据生成器定义中设置将覆盖这些参数选择。

生成数据库（Generating Data）

在底部左侧有三个按键可用于生成实际数据：

开始一个测试运行（Start a test run）—这将生产数据并在结果（Result）标签页的表格

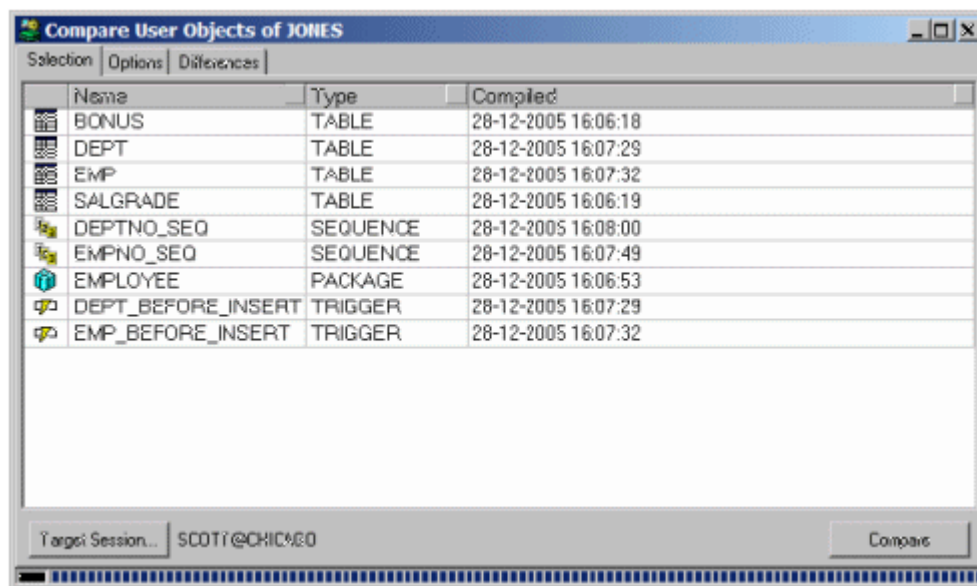
中显示结果。

创建数据为 SQL（Create data as SQL）—生产数据成为一个 SQL 脚本。当你没有使用文件函数从文件中增加数据时才可以使用。

创建数据到 DB（Create data in DB）—生产数据到 Oracle 数据库中。

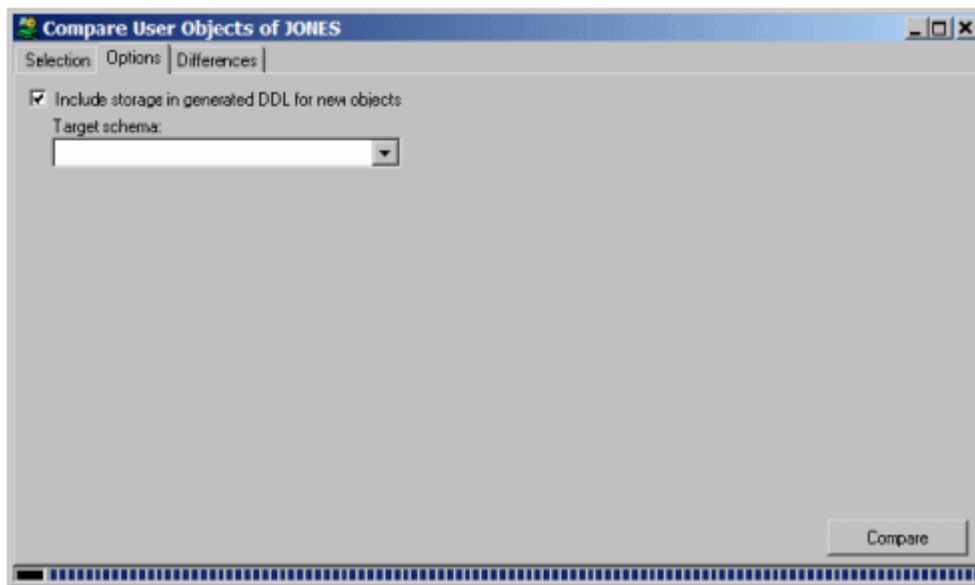
17.10 比较用户对象（Compare User Objects）

修改表定义，视图，程序单元，等等后，可能要传播这些变化到其他数据库用户。这可能是不同的开发环境或者一个测试环境。要比较你的开发用户和其他用户的对象，你可以使用工具菜单中的比较用户对象（Compare User Objects）功能，这将出现下面对话框：



在选择（Selection）标签页你可以选择你想比较的对象。选择后，你可以按目的会话（Target Session）键，选择你要比较的数据库和用户。比较（Compare）键将可用，你可以按此键开始比较操作。

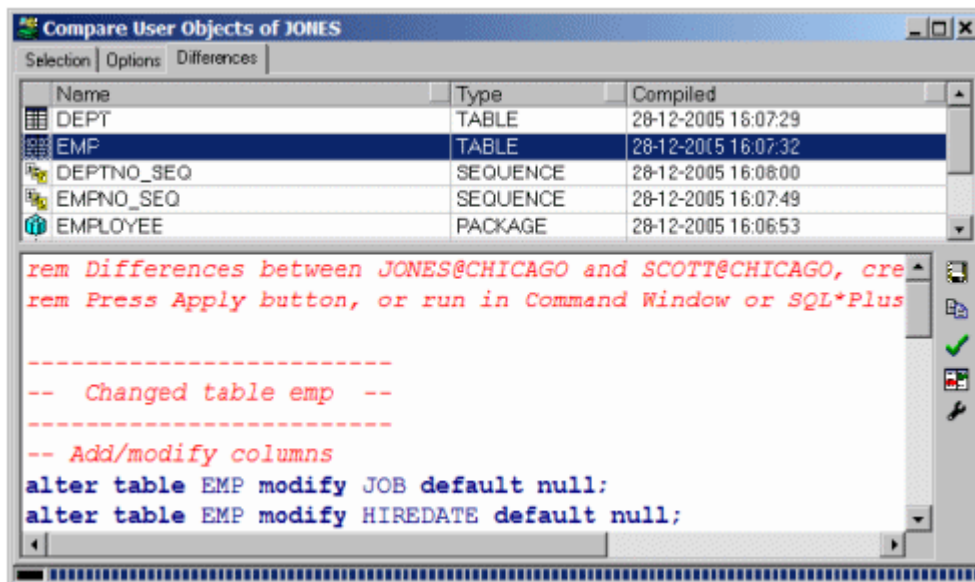
在选项（Options）标签页，你可以设置以下选项：



你可以选择包含存储... (Include storage...) 选项包含像新对象的表空间名和初始大小的存储信息。这些在数据库之间可能不同。所以，这个选项不一定总适合。

如果目的会话的对象不是已连接的目的用户拥有，可以设置目的方案 (Target schema)。

当比较操作完成后，对话框将切换到差异 (Differences) 标签页，显示所有差异对象列表。



这个列表按依赖排序。在目的用户的差异对象列表的下方，你可以看到为和当前用户的相应对象一致需要执行的 SQL。如果没有选择对象，显示全部对象的 SQL。如果一个或者多个对象被选择，只显示被选中对象的 SQL。在上面的例子中，目的方案中的 EMP 表的 JOB 和 HIREDATE 字段的默认值被清除。

显示差异 (Show Differences) 键将逐行可视化显示对象的新旧源代码差异。这对于查

看程序单元的变化可能有用，或者帮助你确定为什么其他对象类型生成一个特定的 DDL 语句。设置外部差异工具（Configure External Difference Tool）允许你设置使用的差异工具。默认使用 ExamDiff 工具，是一个 Pro 版本（见关于 ExamDiff 信息（About item of ExamDiff 扭 Info）菜单）。另见工具—差异（Tools - Differences）章节。

你现在可以按应用 SQL（Apply SQL）键在目的会话中执行 SQL。你也可以按保存 SQL（Save SQL）键保存 SQL 到一个文件中，或者你按拷贝 SQL（Copy SQL）键将 SQL 拷贝到剪贴板中。

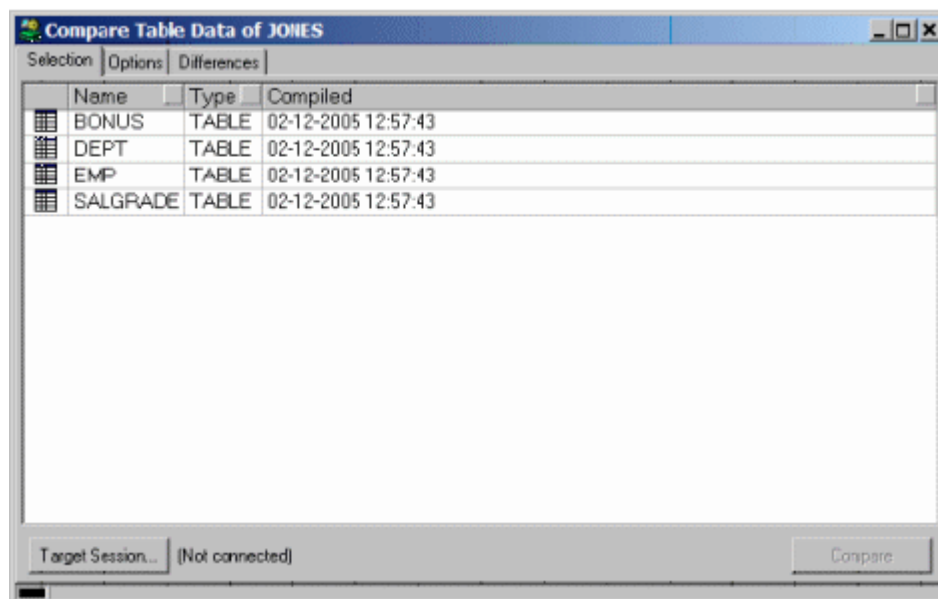
当对象被比较，下面属性被忽略：

- 存储（Storage）—像 the next extent 和 pct free of tables 和 indexes 不考虑。
- 系统生产限制名（Constraints with system generated names）—对于 2 个用户，这些现在有不同的名称，所以不被比较。如果表在目的会话中是新的，这些约束肯定被生成。
- 表创建属性（Table creation properties）—需要重建表的属性被忽略。
- 表数据（Table data）—要导出表数据，使用导出表功能（见导出表章节）。
- 序列值（Sequence values）— 一个序列的当前值是被考虑数据（considered data）。

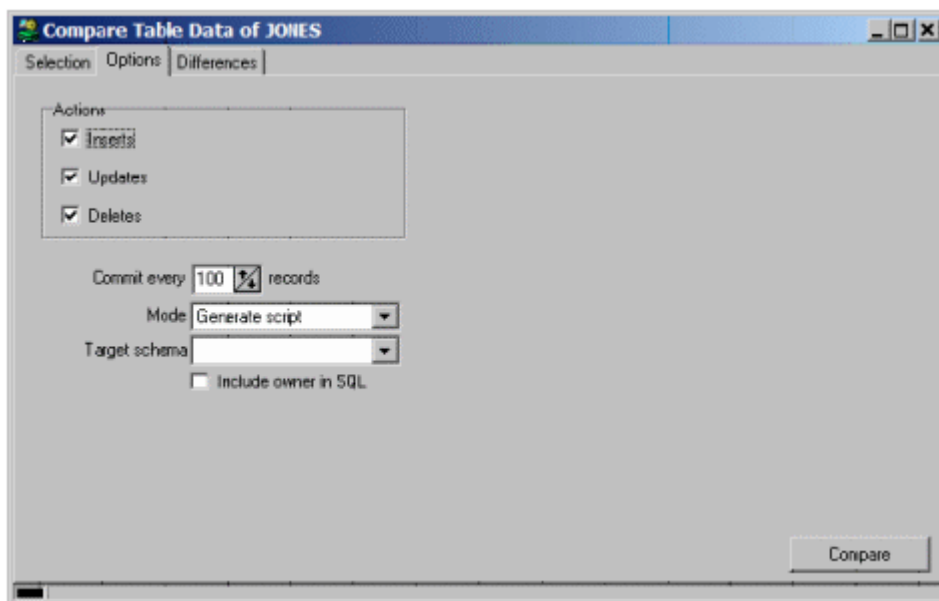
17.11 比较表数据（Compare Table Data）

在开发过程中，在一个或者多个表中插入，更新和删除记录后，传播这些变化到一个不同用户相应表可能是必要的。这可能是一个从开发到测试数据库的传播，或者到生产数据库，或者到其他项目成员。

要这样做，你可以用工具菜单下的比较表数据工具（Compare Table Data），这将调出下面对话框：



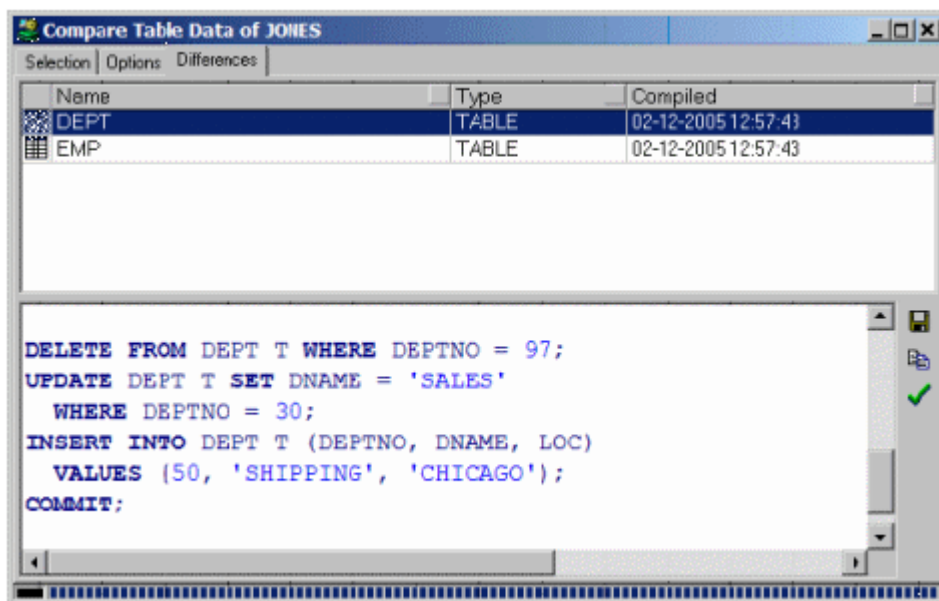
在选择（Selection）标签页中，你可以选择要比较的表。选择后，你可以按目的会话（Target Session）键选择你要比较的数据库和用户。比较（Compare）键将可用，你可以按此键开始比较操作。但是你也可以先到选项（Options）标签页：



在这页中指定比较处理的不同选项。

- **动作（Actions）** — 你可以选择在目的会话中执行的不同动作。例如，如果你省略了删除动作，只有插入和更新将被执行。
- **每 n 条记录后提交事务（Commit every N records）** — 控制多少条记录后提交一次事务。如果你指定 0，只有过程结束后一次提交事务。
- **模式（Mode）** — 选择生成脚本（Generate script）生成一个插入，更新，删除和提交的 SQL 脚本。你可以随后保存脚本并在命令窗口或者在 SQL*Plus 中执行。选项更新数据库（Update database）立即在目的会话应用更新。
- **目的方案（Target schema）** — 如果表的方案没有位于目的会话用户的方案中，你可以在此指定这个方案。目的用户必须在目的方案中对选中表有必须的权限。
- **在 SQL 中包含宿主（Include owner in SQL）** — 这个选项只在脚本模式下才有用，在生成的 SQL 脚本中，表前缀带表的宿主。这样做可以在以后连接一个不同用户运行脚本，并可以应用变更到正确的方案中。

选项适合的选项后，你可以按比较（Compare）键开始比较过程。比较完成后，差异将被显示：



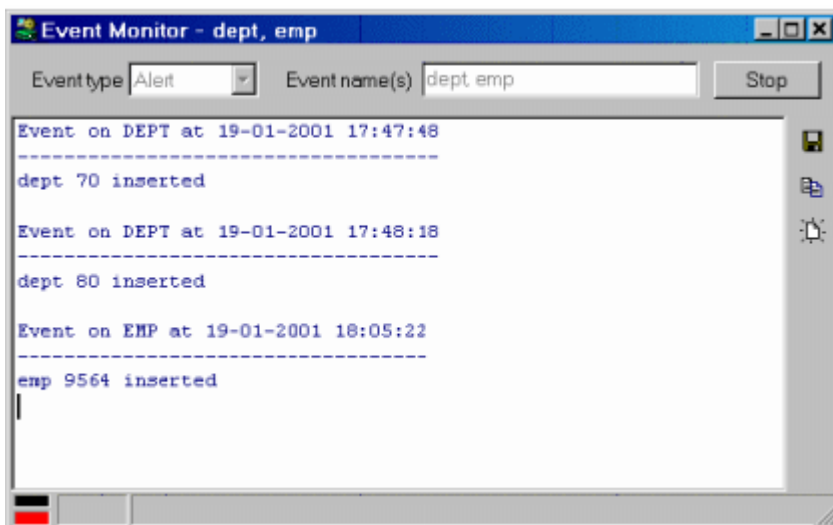
在更新模式下，这个标签页将只显示每个表插入，更新和删除记录的总数。

在脚本模式，你可以预览全部差异，按右侧的保存 SQL 到文件（Save SQL to file）键保存结果脚本。注意你可以在上面窗格中选中表来限制指定表的脚本。你也可以按相应键拷贝 SQL 到剪贴板或者应用选中的变化到目的会话中。

17.12 事件监视器（Event Monitor）

事件监视器允许你捕获和查看在其他会话中的程序单元产生的 dbms_pipe messages 或 dbms_alert signals。只要指定事件类型（Alert 或 Pipe），警告或者管道名，并按开始（Start）键。

下面的屏幕截图显示一个事件监视器等待一个警告名为'DEPT'或'EMP'：



每当这个事件发生，管道或警告名称和时间戳将被写到输出页，后跟事件信息：

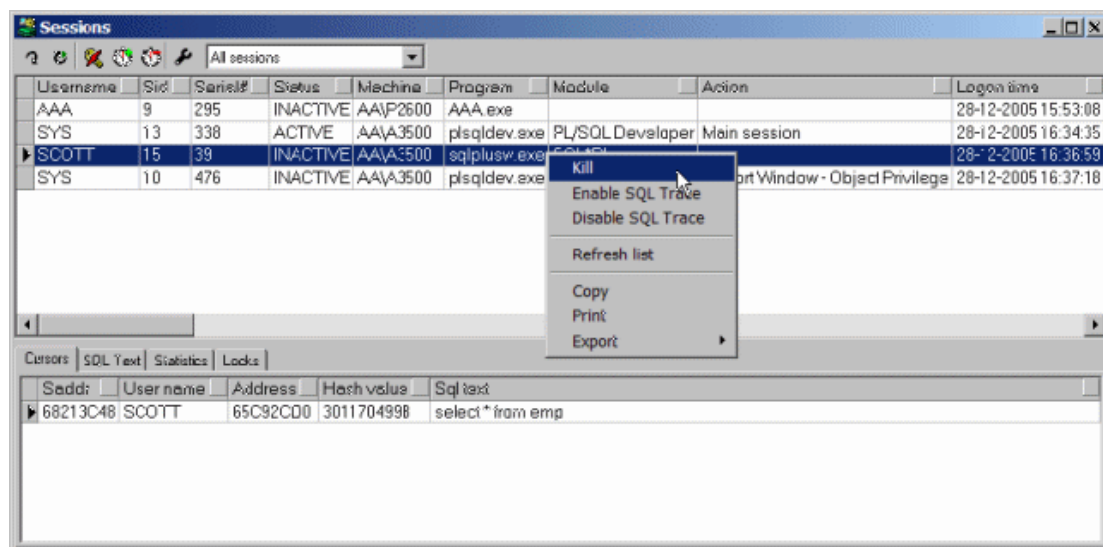
- 管道（Pipe）：显示全部信息项目

- 警告（Alert）：显示 dbms_alert.signal 调用的参数信息值。

对于警告事件，你可以指定多个名称，用冒号分割。对于管道事件，你只能指定一个管道名。

17.13 会话（Sessions）

会话工具显示你当前连接的数据库实例中的全部会话：



窗口的上半部分是会话的信息，如用户名，sid 和 serial number（标识会话），状态，等等。你可以点击一个列的标题按钮排序会话列表。例如：为快速查找所有活动会话，点击状态列的标题按钮。

你也可以从工具条选中一个会话过滤器（Session Filter）限定到一个特定会话选集的会话列表，或者限制显示字段。一个过滤器也可以定义会话和列顺序。

如果你右键点击一个会话，这将调出一个带以下项目的弹出菜单：

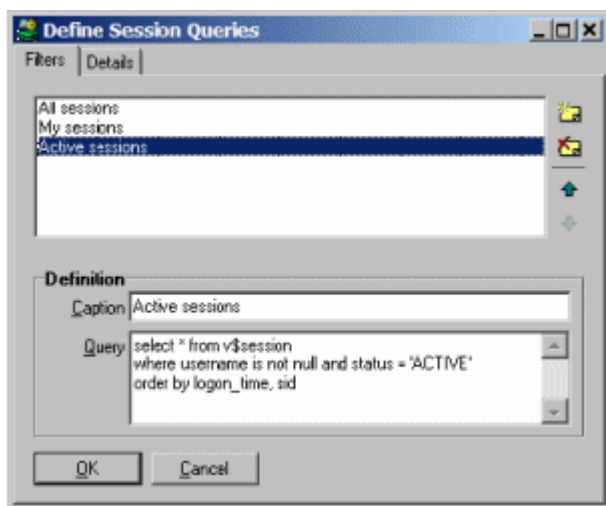
- 杀死（Kill）— 允许你杀死选中的会话。
- 启用 SQL 跟踪（Enable SQL Trace）— 启用选中会话的 SQL 跟踪。这个选项在 Oracle 服务器版本不支持情况下被禁用，或者你没有在 dbms_system package 上的权限。
- 禁用 SQL 跟踪（Disable SQL Trace）— 在选中的会话中禁用 SQL 跟踪。
- 刷新列表（Refresh list）— 刷新会话列表。
- 拷贝（Copy）— 拷贝会话列表到剪贴板中。
- 打印（Print）— 打印会话列表。
- 导出（Export）— 以 CSV, TSV, HTML 或 XML 格式导出会话列表。

刷新，杀死，启用 SQL 跟踪和禁用 SQL 跟踪功能也可以使用窗口上方的工具条。工具条中还包括一个自动刷新（Auto refresh）键。当按下，会话列表和会话信息将定期刷新。你

可以右键点击按键定义刷新周期。

窗口下半部分包含一些有关选中会话信息的标签页。在这些标签页中的信息可以刷新，拷贝或者打印，在列表上右键点击并从弹出菜单中选择相应菜单项。更多有关显示列表中的各列信息见"Oracle Server Reference"指南。

你可以创建，更新或者删除会话列表和会话明细标签页的会话查询。要做到这一点，在工具条中按定义会话查询（Define Session Queries），这将调出下面的对话框：



在上方你可以切换过滤器和明细查询。过滤器控制会话列表显示的内容，明细控制明细标签页显示的内容。

在右侧你看到四个按键，分别是创建一个新查询，删除一个查询，在列表中上下移动查询。当你创建或者修改一个查询时，你必须提供以下信息：

过滤器（Filters）

标题（Caption）显示在工具条中的过滤器选择列表中。

查询（Query）是一个查询语句，为主会话列表提取会话信息。你可以使用这个查询：

- 1.限制会话显示（如只有活动会话）。
- 2.定义会话的显示顺序。
- 3.定义显示的字段和顺序。

明细（Details）

标题（Caption）显示在明细标签页的标签上。列表的位置决定了在会话窗口中的标签位置。

查询（Query）是 SQL 查询语句，提取该会话信息。这个查询语句应该包含:sid 绑定变量，这样当查询被执行时将在查询中包含 sid。你可以使用 v\$session 视图中的任何字段，如:username, :sql_address, 或 :sql_hash_value。如果你的查询造成一个执行错误，在执行后将在相应的标签页中显示出来。

要串联（concatenate）一个会话明细查询的多行记录值，你可以在 SQL 文本中使用 `/* concatenate */` 提示。例如：

```
select sql_text from v$sqltext_with_newlines where address = hextoraw(:sql_address) and
hash_value = :sql_hash_value order by piece /* concatenate */
```

这个查询返回的所有 `sql_text` 字段将被连接，并且在会话明细表格中被显示为一个值。

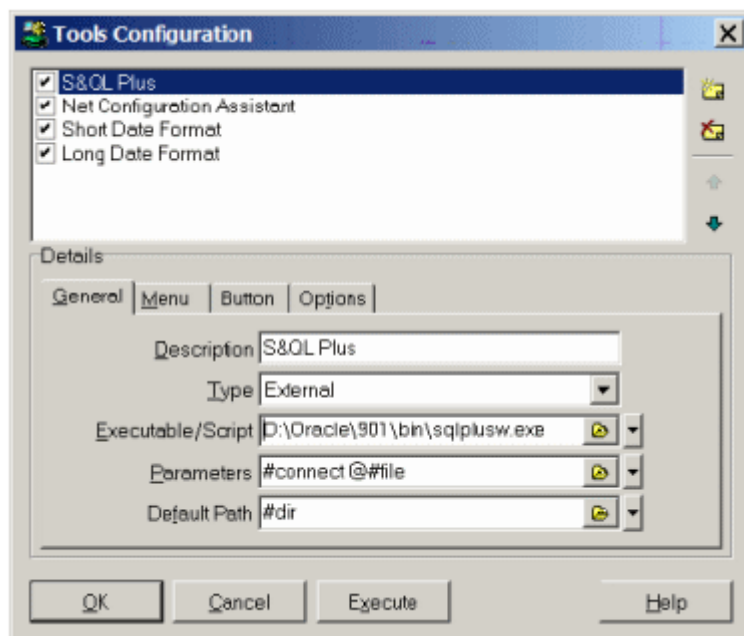
17.14 用户定义工具（User Defined Tools）

你可以定义自己的工具集成到 PL/SQL Developer 的集成开发环境中。这些工具可以添加到菜单项中以启动外部应用或者在当前数据库会话中执行 SQL 或者 PL/SQL 脚本。你可以配置参数，这样你可以传递一个文件名和连接串到你的应用程序。

一个好的例子应该是 SQL*Plus。你可以添加一个菜单启动 SQL*Plus，并使用相应的设置可以让 SQL*Plus 登录，甚至可以让它执行当前打开的文件。

另一个例子是让 NLS 设置一个特殊数值集的工具。你可以创建一个脚本执行正确的 "alter session" 语句，并添加这个脚本到菜单中，它可以在 SQL 窗口，测试窗口等会话中被调用。

要配置外部工具，你必须在工具菜单中选择配置工具（Configure Tools）菜单项。将出现下面的对话框：



右侧有四个按键允许你插入和删除项目，以及在列表中上下移动项目。如果你在按插入键时按下 **Ctrl** 键，将从当前选中项目拷贝一个新的项目。

执行（Execute）将可以用来执行选中的工具。当你按执行时按下 **Ctrl** 键，将弹出一个消息，包含有关要执行的（带替换参数）信息。

列表显示所有设置的工具，并在下半部分显示选中工具的设置。设置被分成四个部分：

- 常规（General）— 定义要执行的执行程序/脚本和它的参数。
- 菜单（Menu）— 定义相应菜单项应该出现的位置。
- 按钮（Button）— 定义一个按钮图形和在工具条中的描述。
- 选项（Options）— 额外的工具设置。

为了说明外部工具设置，我们将以添加一个 SQL*Plus 菜单到 PL/SQL Developer 中为例：

常规标签（The General Tab）

使用插入按钮创建一个新项目后的第一件事情是定义工具类型。如果你选择外部（External），工具将按一个外部程序被调用。这适用于 SQL*Plus。如果你选择会话（Session），当工具被调用时，这个工具将运行一个当前会话的 SQL 脚本。

下一步你将需要在常规标签页输入一个描述。这个描述是在菜单和列表中显示的名称。你可以在你想作为一个快捷键的字符前输入一个 &（S&QL*Plus 得到 SQL*Plus）如果你输入 '-' 作为一个描述，将显示一个分割行。

第三个重要的事情是执行的程序名或者脚本名。对于外部工具，你可以输入你希望的任何执行程序文件或者甚至是文档文件，这将启动关联的应用程序。浏览按钮打开一个文件对话框，允许你查找文件。大多数 Oracle 工具在 Oracle 的 bin 目录下，SQL*Plus 的程序是 <oracle_home>\bin\sqlplusw.exe。

这三步足以从一个菜单中让 SQL*Plus 工作了。你可以额外定义传入应用的参数和定义一个默认路径。你可以使用默认路径替代一个执行程序或者文件参数的完整路径。对于 SQL*Plus 你可以增加一个连接串（#connect）和一个连接到一个要执行的文件（#file）。

右侧带下肩头的小按钮允许你拾取一个你可以在任何输入域中插入的变量。这些变量都和当前连接和打开文件相关。它们将在工具执行时替换相应的信息。支持下面的参数：

变量	含义
#file	代表在当前窗口中的文件（不带路径）
#path	带路径的文件名
#dir	当前窗口文件的路径
#object	选中的浏览对象（如 SCOTT.EMP）
#otype	选中的浏览对象类型（如表）
#owner	对象宿主（SCOTT）
#oname	对象名称（EMP）
#connect	当前 PL/SQL Developer 连接的完整连接串

	(如 scott/tiger@demo)
#username	用户名
#password	密码
#database	数据库

结果是 SQL*Plus 不希望一个带有空格的完整路径作为文件参数，因此指定 #dir 作为默认目录，这样，只用将文件名传入到 SQL*Plus。

会话工具 (Session tools)

对于会话工具，你需要指定一个 SQL 脚本，包含多个 SQL 语句或者 PL/SQL 块。多个 SQL 语句可以用分号或者斜线隔开，PL/SQL 块必须以一个斜线分开。这和 SQL*Plus 语法一致。

例子 1—Germany.sql:

```
alter session set nls_language = 'german';
alter session set nls_territory = 'germany';
```

例子 2—SpecialRole.sql:

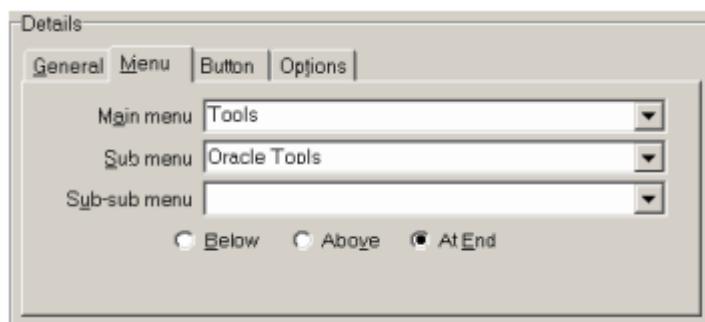
```
begin
    dbms_session.set_role(role_cmd => 'special_role identified by &Password');
end;
/
```

注意你可以在脚本中增加置换变量增加灵活性。当这个工具被调用时，你将被提示提供这些变量值。当上面的 SpecialRole 脚本被执行时，你将被提醒提供密码。

如果你想在脚本中使用任何特殊变量，你必须使用 &1, &2 等等。数字代表命令行参数的顺序。这些命令行置换参数不提示你提供数值。

菜单标签 (The Menu Tab)

默认情况下所有配置的工具将被添加到工具菜单底部。如果你想在其他地方创建他们，甚至在他们自己的主菜单中，你可用使用菜单标签页：



如果你想在指定的菜单下创建你的工具菜单项，你可以在 Main menu 编辑列表中

选中菜单，如果你想创建一个新的主菜单，你可以输入一个新的菜单名。

如果你输入了一个主菜单，你可以使用子菜单（Sub Menu）和子—子菜单（Sub-sub Menu）标明你想创建菜单项的具体位置。

你可以使用三个单选键（下，上和最后）标明新菜单相对于特定菜单的位置。如果你指定了一个新（子）菜单，你必须选中最后，因为下和上只有针对一个已存在的（子）菜单才有意义。

例子 1:

```
Main menu  Tools
Sub menu   Oracle tools
Sub-sub menu
Location At End
```

这将创建一个在 Tools 主菜单下的 Oracle tools 子菜单。如果你要添加多个 Oracle 相关工具，一个特殊子菜单可能是一个好主意。

例子 2:

```
Main menu  Oracle
Sub menu
Sub-sub menu
Location At End
```

这将创建一个 Oracle 主菜单。如果你经常性地使用一些工具并且想组织在一起，或许你只需要创建一个新的主菜单。

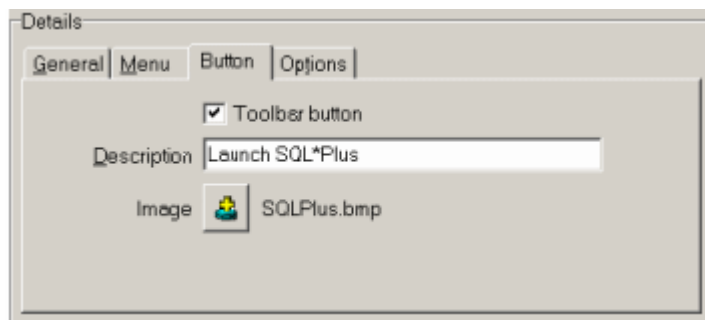
如果你想添加你的工具在工具菜单的第一项，你应该输入：

```
Main menu  Tools
Sub menu   Preferences...
Sub-sub menu
Location Above
```

这将在 Preferences 上方创建一个菜单。

按键标签（The Button Tab）

你可以定义一个外部工具是否和如何包含在工具条中：



- 工具条按钮 (Toolbar button)

当这个选项被启用，外部工具可以包含在工具条中。

- 描述 (Description)

这个描述将在你保持鼠标光标在工具条上时显示一个提示信息。如果你没有填写这个描述，将使用外部工具的描述（同在目录中显示）。

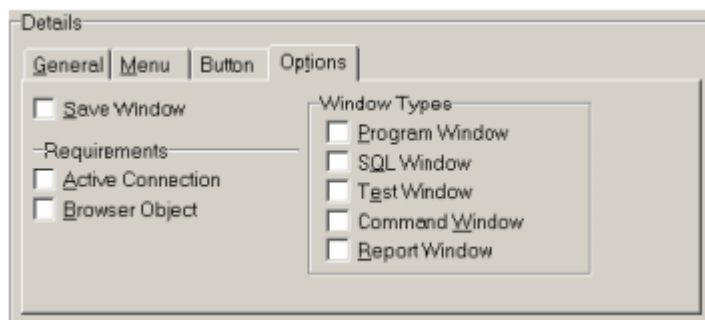
- 图象 (Image)

按图象 (image) 键为工具条按钮选择一个 Windows 位图文件 (*.bmp)。图象大小 20 x 20 比较适合。注意 PL/SQL Developer 总是从最初位置装载位图文件。所以，在没有修改相应的外部工具时，你不能移动或者重命名位图文件。

PL/SQL Developer 带有一些标准的位图文件，你可以从中选择。这些文件保存在 PL/SQL Developer 安装目录下的 Icons 子目录中。这是图象选择器的默认目录。

选项标签 (The Options Tab)

选项标签页允许你进行以下设置：



- 保存窗口 (Save Window)

当设置了这个选项，活动窗口将在工具执行前被保存。当工具获取一个参数作为文件名并且你想保证使用的是当前数据时你应该设置这个选项。

- 活动连接 (Active Connection)

如果工具只能在 PL/SQL Developer 连接时才能使用（如果你用参数传入一个连接串），你应该设置这个活动连接。如果 PL/SQL Developer 没有连接将禁用这个菜单。

- 浏览对象（Browser Object）

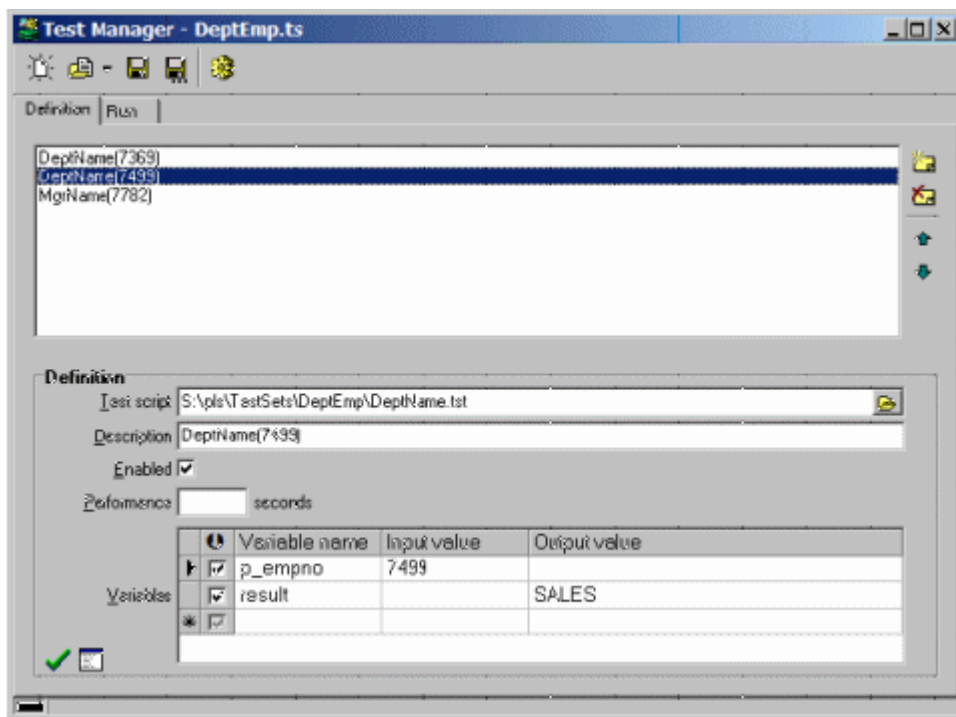
如果这个工具工作在浏览器选中对象上就设置这个选项。

- 窗口类型（Window Types）

如果工具要求一个特殊窗口，你可以在这里指定。如果一个或者多个被选中，菜单只有在活动窗口类型是指定的才可用。

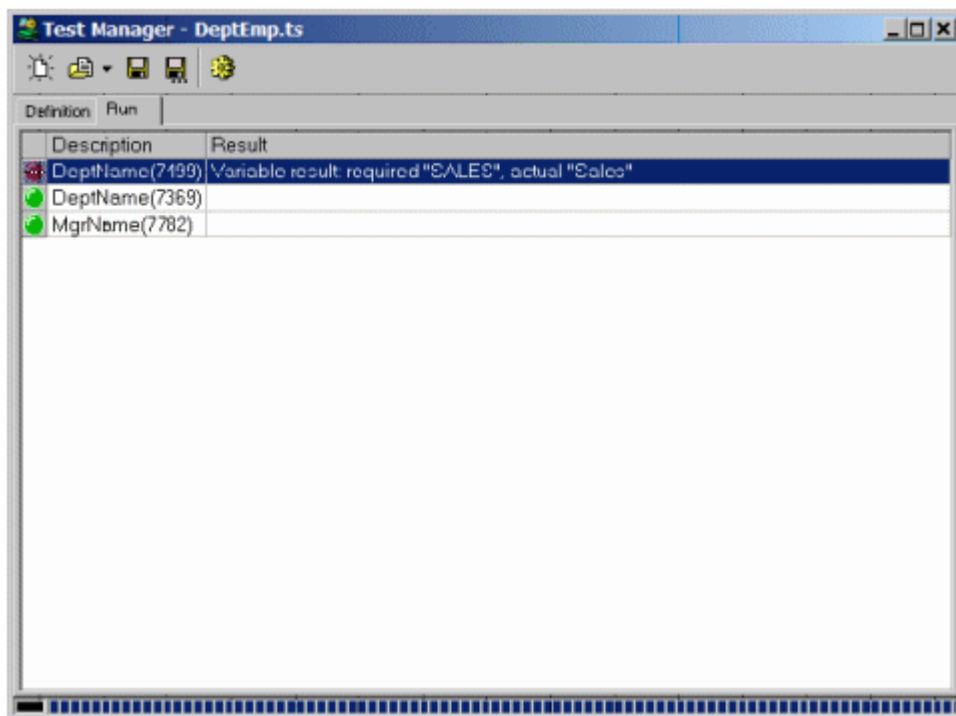
17.15 测试管理器（Test Manager）

要执行你的 Oracle 存储过程单元回归测试，你可以使用测试管理器。它允许你定义一个测试集，这是一个测试脚本带输入变量值和输出要求的收集器。它还允许你指定需要的性能。



对于这个样例，测试集包含 3 个测试脚本。对于选中的 DeptName.tst 脚本，p_empno 变量将获得 7499 值作为输入，执行后的结果变量值必须是 SALES。

运行一个测试集将快速揭示测试的程序单元功能是否仍然正确/或满足性能要求：



所有失败的测试脚本将放在上方并带一个红色标识器。在这个例子中，DeptName(7499)脚本失败，因为指定需要的结果是"SALES"而实际结果是"Sales"。

创建一个测试集（Creating a Test Set）

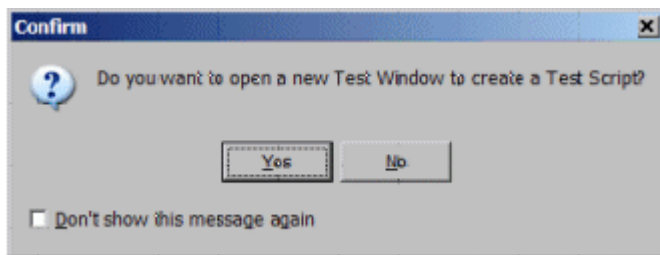
要创建一个测试集，从工具（Tools）菜单中选择测试管理器（Test Manager）菜单项。出现一个空的测试管理器，可以让你创建新的测试脚本或者添加测试脚本。在测试管理器上方，你看到一个测试脚本列表，在这里可以添加脚本，移除脚本或者修改脚本顺序。在下方你看到选中脚本的定义。

创建一个测试脚本前，按工具条中的保存测试集（Save Test Set）键保存测试集是一个好主意。所有新的测试脚本将作为测试集按默认创建在同一个目录中，每个测试集最好使用不同的目录，以使所有相关测试资源可以在一个独立的目录中管理。

添加一个测试脚本到测试集（Adding a Test Script to the Test Set）

有两种方法添加一个测试脚本到测试集：

1. 从对象浏览器中拖放一个过程或者函数到测试管理器中。一个带调用这个程序单元，和全部参数变量以及返回值的新测试窗口将被打开，使用一个描述性名称保存测试脚本，最好在测试集目录中。
2. 按测试脚本列表右侧的新建（New）键。将出现下面对话框：



点击 **Yes** 创建一个新的空测试脚本。你需要手工编写测试脚本并在测试集可使用之前保存它。

点击 **No** 如果你想添加一个已存在的测试脚本到测试集中。

测试脚本保存后，你可以在测试集中指定脚本的定义。

- **测试脚本 (Test script)** — 测试脚本的文件名。如果你已经创建并保存了一个新测试脚本，这将被自动填写。如果你想添加一个已存在的测试脚本，在这个输入域右侧按浏览 (**Browse**) 键。
- **描述 (Description)** — 一个脚本的功能描述。如果你多次使用不同的输入参数值执行一个脚本，你也应该在描述中包含，以使在列表中快速识别。
- **启用 (Enabled)** — 使用这个选项启用或者禁用测试集中的脚本。只有启用的脚本将被包含在测试运行中，禁用脚本将被跳过。
- **性能 (Performance)** — 需要的性能可以用最大秒数指定（允许小数）。如果没有填写这个输入域，性能将不被测试。
- **变量 (Variables)** — 对于测试脚本中的每一个变量，输入名称，输入值，和需要返回值。输入值将在执行前被传入测试脚本。需求返回值将在执行后被测试。如果你没有填写输出值，将不被测试。为测试空值，输入 "null"。

在左下侧你看到 2 个按键，允许你快速运行和测试选中的测试脚本，或者在测试窗口编辑打开。

复制一个测试脚本 (Duplicating a Test Script)

经常你想使用不同的输入值和输出参数值运行同一个测试脚本。为满足这种情况，你可以在测试集中选择一个已存在的测试脚本，按新建 (**New**) 键的同时按下 **Ctrl** 键。新建的测试项将继承选中的项目，因此你只需要修改描述和变量值。

运行一个测试集 (Running a Test Set)

要运行一个测试集，在工具条中按运行 (**Run**) 键。测试管理器将切换到运行标签页，每个启用的测试脚本将在一个分开的测试窗口中执行。失败的将用红色指示器提示，并将放置在列表上方。成功脚本将有一个绿色指示器，并将放置在列表下方。

要研究一个失败，右键点击脚本，从弹出菜单中选择调试 (**Debug**)，或者在脚本上双击。一个带测试脚本的测试窗口将被创建，并且输入变量值将被正确设置。现在你可以调试

测试脚本来确定产生错误的原因。

在测试运行结果上右键点击拷贝他们到剪贴板，打印他们，或者以多种格式导入他们。

18 编辑器 (*The Editor*)

每个允许你编辑一些 SQL 或者 PL/SQL 源代码的窗口使用同一个编辑器。这个编辑器有很多特殊的特性可以使一个编程人员的工作更轻松一些。

18.1 选择功能 (Selection functions)

编辑器允许你在一个选中的文字上执行不同功能。这些功能在编辑 (Edit) 菜单下的选择 (Selection) 菜单下，只有在你选择了文字才可应用。你可以通过键设置 (Key Configuration) 参数选择为这些功能赋予一个功能键。

- 缩进/撤销缩进 (Indent / Unindent)

字符数依据编辑器参数选择。如果一个文字选择存在，Tab 和 Shift-Tab 键总是赋予这两个功能。

- 小写/大写 (Lowercase / Uppercase)
- 注释/撤销注释 (Comment / Uncomment)

使用 /*和*/快速注释一个文字选择

- 应用语法大小写 (Apply Syntax Case)

如果你设置了编辑器主关键字大小写 (Editor Keyword Case) 参数为大写，小写或者首字母大写，则这个功能将应用这个风格到全部选择文字中的主关键字。

- 排序 (Sort)

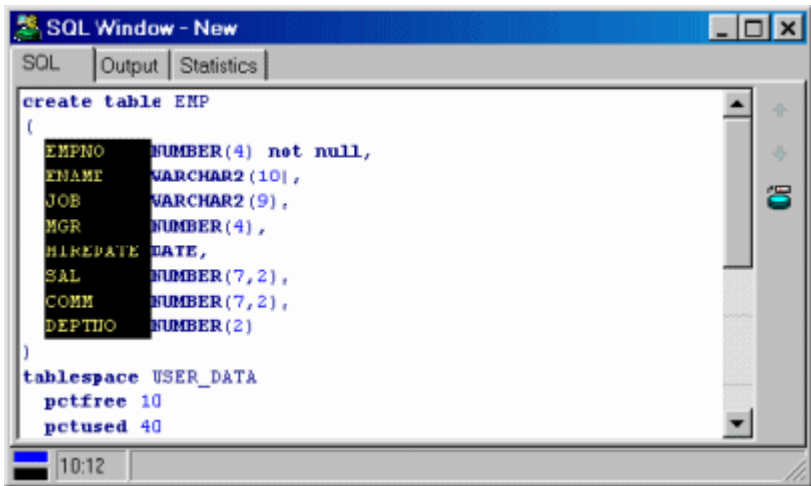
选择行将按升序排序。如果你再次调用排序功能，选择行将按降序排序。

- 颜色标记 (Color mark)

应用当前标记颜色到选择文字 (见颜色标记章节)。

18.2 列选择 (Column selection)

当你做一个选择时按下 Alt 键可以选择一个文字列：



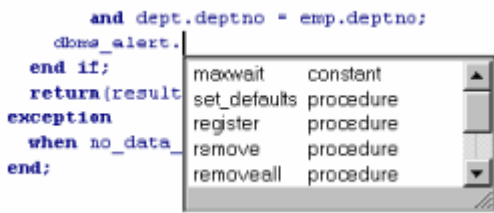
你可以照例拷贝，粘贴，剪切，删除，大写，小写，（撤销）缩进，和（撤销）注释所选列。如果你使用应用语法大小写功能，并且一个关键字跨越了列边界，它将不被处理。

18.3 导向图（Guidelines）

当编辑复杂代码带多个 if/then/else 语句和循环跨越 PL/SQL 代码的多个页，在编辑器中有一个垂直导向图将很有用。要做到这点，简单将光标放置在想得到的水平位置，从编辑（Edit）菜单中选择 Guideline 菜单项。这个菜单项承担一个开关，所以，你可以再次使用移除导向图。

18.4 代码助手（Code Assistant）

当你输入 SQL 或 PL/SQL 代码时，代码助手是提供帮助的强有力部件。例如，当你输入 dbms_alert.并犹豫了一下，编辑器将显示在 dbms_alert 包中的元素列表：



当你继续输入 dbms_alert 过程名时，列表中内容将减少。例如，如果你在 dbms_alert.后输入一个 r，只有 register, remove 和 removeall 过程将还出现在列表中。当你按回车时代码助手可见，当前选中项的名称将插入到代码中。你也可以使用箭头键在代码助手中查看和选择想得到的项目。按 Escapt 键将移除代码助手。

代码助手将提供以下对象类别的协助：

对象类别	元素
包	过程，函数，类型，变量，限制和异常

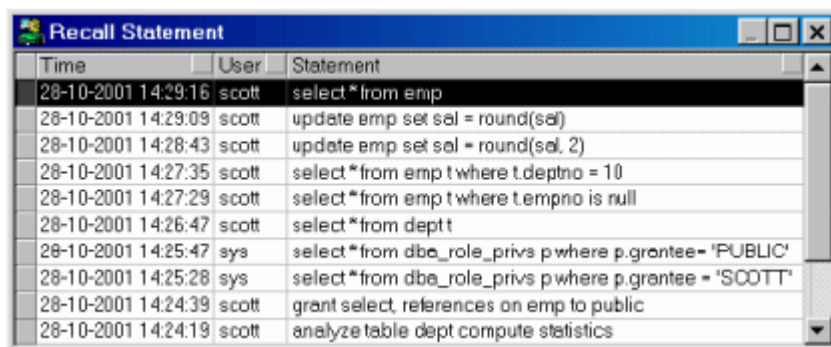
函数，过程	参数，使用命名插入(param =>)
表，视图	字段
序列	当前值和下一个值
用户	用户拥有的对象
游标变量	游标选择语句的字段
记录类型变量	记录类型的字段
Table%rowtype 变量	表字段
对象类型	属性（对于默认的构造器）
对象类型变量	属性和方法
对象类型字段	属性和方法
集合变量	集合方法(first, next, limit 等等)

代码助手还可以帮助你输入对象名（表，包等），关键字和在当前上下文中有意义的 PL/SQL 标识符。例如，如果你在 SQL 编辑器中输入 `pro` 后停顿，代码助手将显示关键字 `procedure` 和 `profile`，伴随着任何以 `pro` 开始的数据库对象名。上下文依赖当前连接的用户（决定可用的对象名），编辑器的类型（决定可用的关键字），你编辑的程序单元（决定可用的 PL/SQL 标识符）。

代码助手可以自动和/或者手工调用，在输入一个对象名后延迟多长时间后代码助手自动激活可以使用参数定义。此外，你可以定义是否想描述一个用户，如果这样，还包括你想包含的对象类型。上下文和代码助手被激活前必须输入的最少字符数都可以定义。全部这些参数选项详见用户界面—代码助手章节（User Interface - Code Assistant）。

18.5 重新调用语句（Recalling statements）

每当你在一个 SQL 窗口，命令窗口，或者报表窗口成功执行了一条语句，这个语句被保存在全局语句重调缓存中。你可以从编辑（Edit）菜单下选择重新调用语句（Recall Statement）在编辑器重新调用最近执行的语句，或者按 `Ctrl-E`。这将调出一个选择列表：



语句按执行顺序显示，最后执行的在第一个。在列表中你看到时间，用户，和（部分）语句文本。你现在可以选择一个语句，双击它（或按回车）将它插入到在调用列表时编辑器中的光标位置。

注意带密码的语句因安全原因不放入全局语句重调缓存中。

你可以按相应的列表标题按键在列表中按时间，用户或者语句文字排序。只显示当前连接用户的最近执行语句，右键点击列表，从弹出菜单中选择当前用户（Current user）菜单项。在同一个弹出菜单，你可以切换回全部用户，并且可以删除一个语句或者拷贝语句文字到剪贴板中。

要查找一个特殊语句，你可以使用相应的功能键调用查找功能（默认 Ctrl-F），或者在列表上右键点击，从弹出菜单中选择查找语句（Find statement）。查找功能将从当前选中语句开始，并且在未查找到一个匹配时从第一条继续查找。

因此，弹出菜单包含从列表中删除语句，拷贝到剪贴板，或导出整个列表到一个 CSV 文件的功能。

重新调用缓存可以保存 200 个语句，这将保存在你的 Windows profile 的应用数据目录。你可以通过参数选项修改这个语句限制和保存目录，见工具—重新调用语句章节（Tools - Recall Statement）。

18.6 特殊拷贝（Special Copy）

如果你正使用 PL/SQL Developer 编写 SQL 和 PL/SQL 代码随后想在其他工具中使用，如 3GL 程序语言，这可能需要你的代码在格式上有些不同。让我们假设你已经在 PL/SQL Developer 中编写和测试了一个 SQL 语句像这样：

```
select deptno, sum(sal) mgr_sal from emp
where job = 'MANAGER'
group by deptno
order by mgr_sal desc
```

如果假设你想使用这个语句在 Borland Delphi 中，你需要一个格式像这样：

```
SQL := 'select deptno, sum(sal) mgr_sal from emp' + #13#10 +
      'where job = "MANAGER"' + #13#10 +
      'group by deptno' + #13#10 +
      'order by mgr_sal desc';
```

为了这个目的，你可以使用编辑菜单或者在编辑器中选中后右键点击，从弹出菜单中选择特殊拷贝（Special Copy）功能。这个功能有一个子菜单，显示全部定义的特殊拷贝格式。选择一个格式后，转换的代码被保存在剪贴板中，因此，你可以在相应工具的编辑器中

粘贴。

特殊拷贝格式在 PL/SQL Developer 安装目录下的 SpecialCopy 子目录中定义。你可以修改预定义的拷贝格式，或者增加新的拷贝格式，只要增加一个带 .copy 扩展名的文件，文件中包含一个第一行的 PL/SQL 代码(<line_1>)，最后一行的 PL/SQL 代码(<line_N>)的变量和一个其他行(<line_*>)的变量。对于 Borland Delphi 的例子如下：

```
;PL/SQL Developer SpecialCopy definition for Borland Delphi

;<line_1> for first line

;<line_*> for all other lines

;<line_N> for last line

;

SQL := '<line_1>' + #13#10 +
        '<line_*>' + #13#10 +
        '<line_n>';
```

需要第一行之前赋值 SQL 变量，并且必须跟随一个 CR/LF 对。最后一行不需要 CR/LF 对，但是需要以分号结束。其他行只需要跟随 CR/LF。

你可以忽略<line_1> 和 <line_n>如果他们同<line_*>。

在一些语言中，特殊字符必须使用 escape 序列。例如，C++中\t 代表一个制表符(ASCII code 9)。为定义这些 escape 序列，使用#define 关键字：

```
#define char(9) = \t

#define \ = \\

String("<line_1>\n") +

String("<line_*>\n") +

String("<line_n>");
```

另外，你可以使用"#define compress"指出你想从结果中移除所有多余空字符（空格，制表符，和换行）。

注意.copy 文件名将被包含在菜单中，所以使用描述性的文件名。

18.7 上下文敏感帮助 (Context sensitive help)

大多数编程人员经常在他们编程过程中需要获取参考信息。要尽快找到这个参考信息，将光标放置在代码中的关键字上按 F1 将在 Oracle 的联机手册中查找这个关键字。

你可以阅读帮助系统(Help Systems)获得更多有关 PL/SQL Developer 帮助系统的信息。

18.8 数据库对象弹出菜单 (Database object popup menu)

当你的程序存取一个数据库对象，你经常需要从这个对象中获取一些信息，或者你想操作它。你也许想知道它的描述或者属性。对于一个表或者视图，你也许想查询或者编辑数据。对于一个函数或者过程，你可能想测试一个特定场景。

如果你在编辑器中右键点击一个对象名，你将看到和在对象浏览器中右键点击一样的弹出菜单。因此，从编辑器中立即就可以使用对象的所有功能。

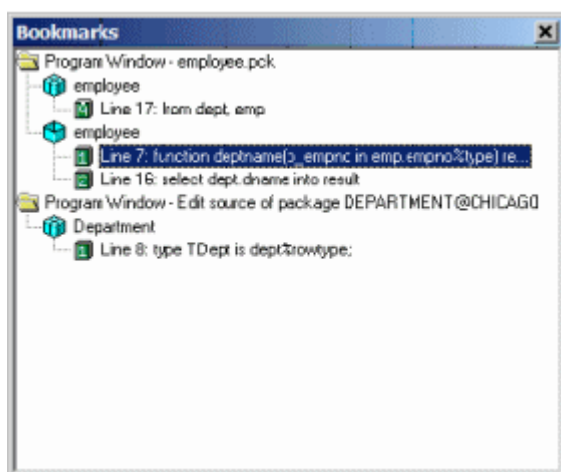
18.9 解释计划 (Explain Plan)

在编辑器中要调用一个 SQL 语句的解释计划，选择语句的文本，在上面右键点击，从弹出菜单中选择解释计划 (Explain Plan) 菜单项。如果没有文本被选择，整个源代码将被放到解释计划窗口。如果 SQL 语句包含 PL/SQL 变量，你必须在解释计划窗口中像在使用解释计划窗口 (Using the Explain Plan Window) 章节描述的改变语句。

18.10 书签 (Bookmarks)

要在一个编辑器中标记一个位置，你可以添加一个书签。只要在当前位置按 **Ctrl-Kn** (n 是 0 到 9 的数字) 增加一个书签。在装订线中将显示一个带相应数字的绿色书签。要在以后导航到这个位置，你可以按 **Ctrl-Qn** 做为选择，你可以使用编辑菜单中的设置书签 (Set bookmark) 和转到书签 (Go to bookmark) 菜单项。

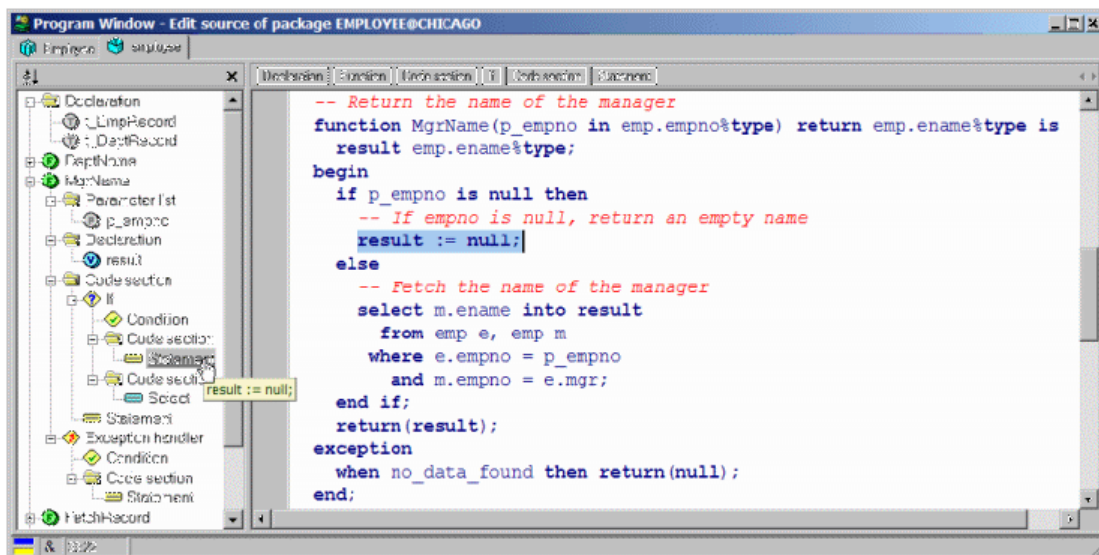
编辑菜单的书签列表 (Bookmark list) 菜单项将调出一个全局层级显示的全部编辑器书签：



在树型视图中你可以看到窗口，编辑器，和一个书签定位的行。在一个书签上点击将使窗口和编辑器在前端显示，并将转到书签。你可以停靠书签列表使它在不挡住任何其他窗口的情况下总是可用的。

18.11 代码目录（Code Contents）

代码目录的特性被限定到程序编辑器。它显示一个程序单元的完整结构。这对于大的可以超过上千行 PL/SQL 代码的包体和类型体非常有用。代码目录在程序窗口左侧的树中显示：



代码目录跟踪你在源代码中的当前位置。当你在源代码中移动时，被选择项目将提示你现在的精确位置。

如果你在代码目录中移动鼠标到一个项目上，编码器中相应代码将像上面图例显示的一样被高亮。

在代码目录中的一个项目上点击将立即导航到它的声明。在项目上双击将展开或折叠它。在项目上右键点击将显示一个弹出菜单允许你在编辑器中选择相应的代码，剪切或者拷贝它到剪贴板，从剪贴板中粘贴，转换为注释，展开所有子项目，为它创建一个测试脚本（如果它是一个共有的程序单元）。

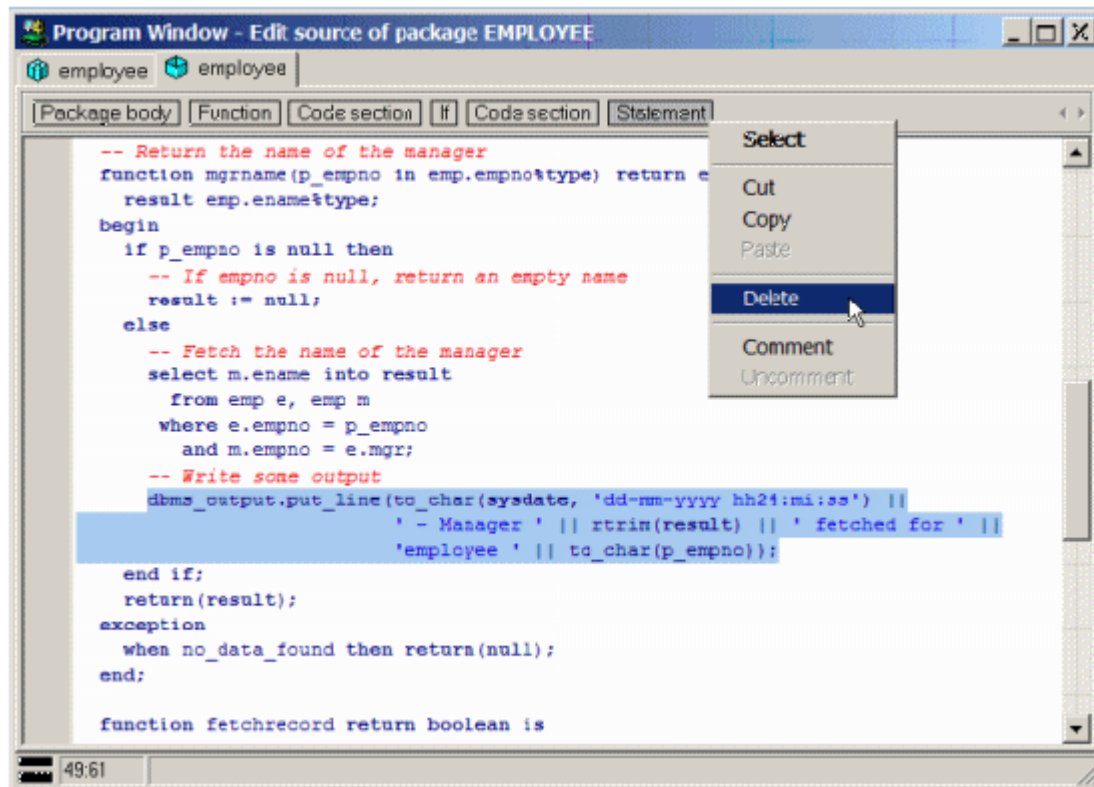
目录可以通过按目录窗格上方的排序目录（Sort contents）键按字母顺序排序。只有主目录将被排序：参数，局部变量等将不被排序。该键将保存按下，再次按它将回复目录到他们原始顺序。

你可以按右上角的关闭键关闭代码目录。你也可以在编辑器中右键点击，从弹出菜单中选择目录（Contents）菜单项使它不显示或者显示，或者在工具菜单中选择代码目录（Code Contents）菜单项。

默认情况下，显示代码目录。要修改这个默认，你可以在当前程序窗口关闭代码目录并在窗口（Window）菜单中选择保存布局（Save Layout）菜单项。

18.12 代码层级 (Code Hierarchy)

代码层级被限制在程序编辑器上，并且显示在光标位置的代码层级结构。他被放置在编辑器的上方：



在上面的例子中，光标位于一个 `dbms_output` 语句，语句在一个代码区内，在一个 `If` 语句内，在一个代码区内，在一个函数内，在一个包体内。如果你移动鼠标穿过代码层级的项目，相应的代码将被高亮，在一个项目上点击将选择相应的代码。

如果你在一个项目上右键点击（如上面的图例），一个弹出菜单将被显示允许你选择源代码，剪切或者拷贝源代码到剪贴板，在项目上粘贴剪贴板，删除源代码，或者将注释转换为代码或者相反。

18.13 超链接导航 (Hyperlink navigation)

在你的 PL/SQL 代码中你经常引用在别处定义的元素。例如你可以引用相同包体中的一个 PL/SQL 类型，或者在相同包中的声明中，或者在其他包中。这同样适用于函数或者过程调用，变量，视图等等。

为了在一个程序窗口中轻易找到一个声明，你可以移动鼠标指针到这个引用上，保持按下 `Ctrl` 键并按鼠标左键。保持按下 `Ctrl` 键，在鼠标指针下的标识符将变为浅蓝色并加下划线，类似熟悉的 HTML 文档中的超链接：


```

-- Created : 17-12-1999 17:19:26
-- Purpose : Lock an employee record
procedure lockrecord(p_empno in emp.empno%type) is
  dummy varchar2(1);
begin
  if p_empno != g_lastrecord.empno then
    select null into dummy from emp
    where empno = p_empno
    for update;
  end if;
end lockrecord;

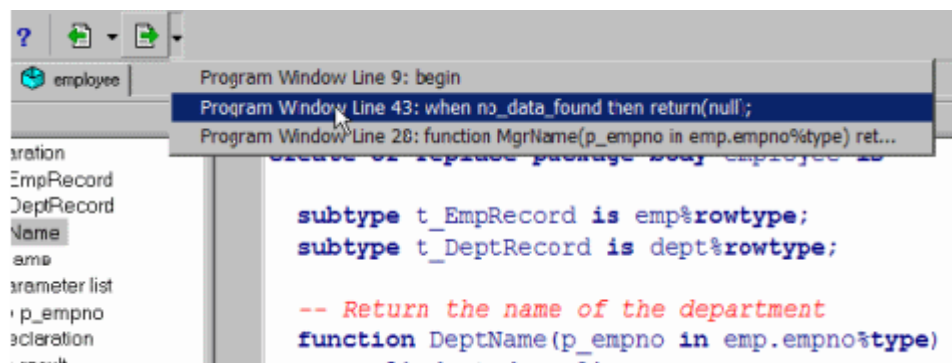
```

如果声明位于同一个源代码中，PL/SQL Developer 将导航到那里。然后，在相同程序窗口中全部其他程序单元将被查找。然后它将查找全部当前打开的其他窗口。最后，它将为包含声明的对象尝试打开一个新程序窗口或者 SQL 窗口，并导航到那里。

默认情况下，如果适当的，超链接导航将定位在一个包中的一个声明或者类型定义。要打开包或者类型体而不是声明，你可以在一个超链接上右键点击。

18.14 导航键 (Navigation buttons)

无论何时你从一个窗口中的编辑器中跳转到一个绝对位置，PL/SQL Developer 将为原位置产生一个书签。这些书签可以使用主工具条中的向后导航 (Navigate back) 和向前导航 (Navigate forward) 键访问：



当你在代码目录中导航，使用查找功能，点击一个编译错误，转到编辑器顶部或者底部，使用超链接导航，打开一个新窗口等等时书签将被添加。

18.15 重构 (Refactoring)

重构功能允许你快速重新组织你的 PL/SQL 代码。它工作在被选择代码上，或者一如果没有做选择—在当前语句上。在一个语句上右键点击或者在相应子菜单中选择提供的以下重构函数：

- 重命名项目 (Rename item)

重命名当前变量，参数，常量或者程序单元。声明和使用都将被重新命名。

- 提炼过程 (Extract procedure)

如果一个程序单元变得太大或者太复杂，你可以进行一个选择并且将它转换为

一个独立过程。在选择之中和之外使用的全部变量都将转换成参数。只在选择之中使用的全部变量将从当前程序单元转移到新的程序单元。这个选择将替代为一个新程序单元的调用。

- 提炼局部常量 (Extract local constant)

如果一个特定表达式应该转化为一个局部常量，你可以选择它并提供一个常量名。一个局部常量将被创建在当前子程序中，类型由表达式确定。所有在当前子程序出现的这个表达式将用该常量名替代。

- 提炼全局常量 (Extract global constant)

如果一个特定表达式应该转化为一个全局常量，你可以选择它并提供常量名。一个全局常量将被创建在当前包中，类型由表达式确定。所有在当前包中出现的这个表达式将被常量名替代。

- 替换赋值为设定初值 (Replace assignment with initialization)

如果一个局部变量赋值纯粹为了设置初值，你可以转移它到该变量的声明。你可以在一个语句或者选择多个语句上右键点击。

如果一个重构动作的结构不满意，你只用回退它。

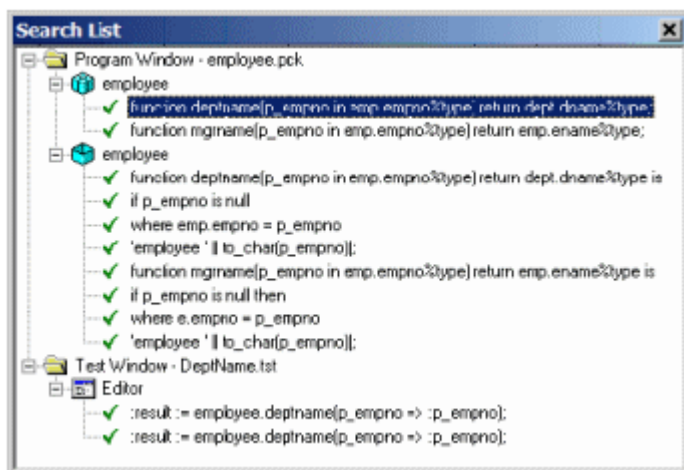
18.16 查找条 (Search Bar)

查找条可以从编辑菜单选择使用并可以显示为一个浮动工具，或者停靠在工作区的上方或者下方。在它的停靠位置，可以在多个编辑器和多个窗口中执行查找：



输入一个查找词后，你可以按查找键或者按回车。查找结构将在编辑器中高亮，你可以按下一个/上一个键导航查找结构。你可以按 Esc 将光标从查找条移到编辑器。按隐藏出现 (Hide occurrences) 键清除编辑器中的查找结果高亮。

查找列表键将调出一个层级显示查找结果：



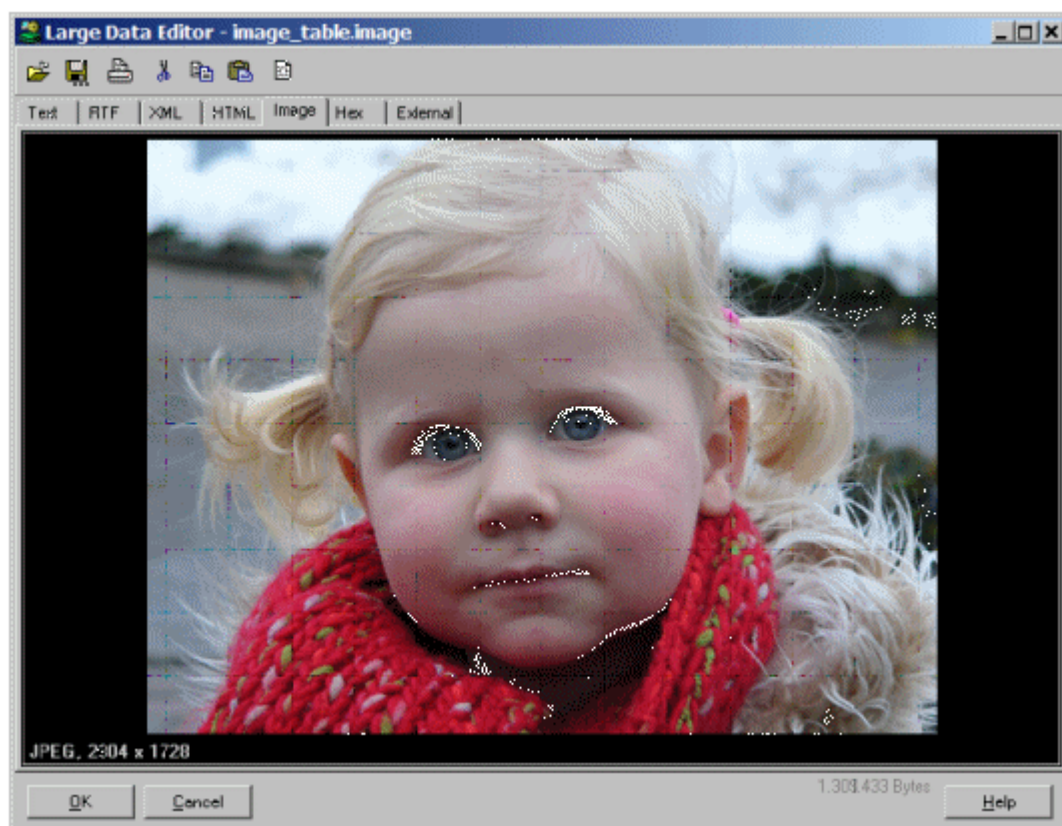
在这个树型视图中你可以看到窗口，编辑器，和查找结果所在行。在一个查找结果上点击将使窗口和编辑器在前端显示，并导航到查询结果和选中它。你可以停靠查找列表，使它在不会挡住其他任何窗口的情况下一直可以使用。

另外，你可以在查找条中启用查找全部编辑器或者只有当前编辑器，大小写敏感查找，全字匹配，和使用常规表达式选项。

19 大数据编辑器 (The Large Data Editor)

只要你需要查看或者编辑 long, long raw, CLOB, BLOB 或 BFILE 字段或者变量值，大数据编辑器将被调用。在编辑 varchar2 数据时也可以使用它。大数据编辑器将分析数据，并将尝试以正确格式显示它。它可以以 text, RTF, XML, HTML, Image (bmp, jpg, gif, tiff 和其他), 和十六进制数据等形式显示值。此外，它可以装载一个外部应用程序（如 MS Word 或者 Acrobat Reader）来查看或者编辑数据。

为一个图像调用大数据编辑器将出现下面屏幕的结果：



编辑器上方的工具条包括下面按钮：

- 打开（Open）—打开一个文件内容到编辑器中。
- 保存（Save）—保存编辑器的内容到一个文件。
- 打印（Print）—用当前选择的格式打印编辑器内容。
- 剪切（Cut）—用当前选择的格式将编辑器内容放到剪贴板，并随即清除（空）编辑器内容。
- 拷贝（Copy）—用当前选择格式将编辑器内容放到剪贴板。
- 粘贴（Paste）—用当前选择格式粘贴剪贴板数据到编辑器。
- 清除（Clear）—清除（空）编辑器内容。

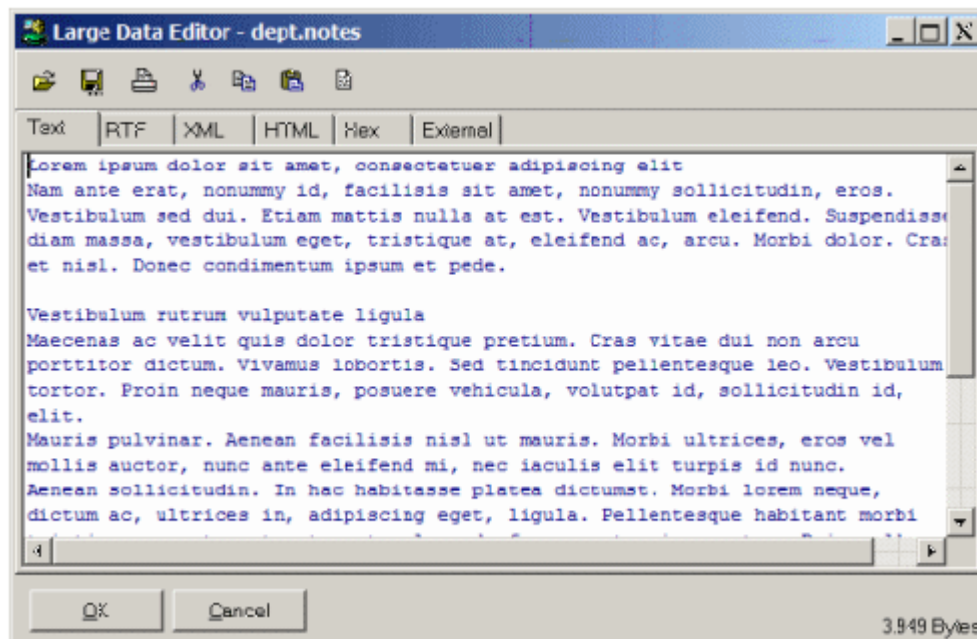
在工具条下，你可以选择一个标签页对应编辑器内容的正确格式。多种格式可能对应某些数据。例如，一个 HTML 文档可以用 HTML 格式或者 Text 格式查看。全部数据可以用十六进制格式查看。

当在一个特定格式改变编辑器内容时，你可以立即切换到一个不同的格式查看这些变化。例如，当查看一个 HTML 文档时，你可以切换到 Text 格式修改并回 HTML 格式查看这些变化。

按 OK 键后，你的改变将被传递到调用大数据编辑器的功能中。

19.1 编辑普通文本 (Editing plain text)

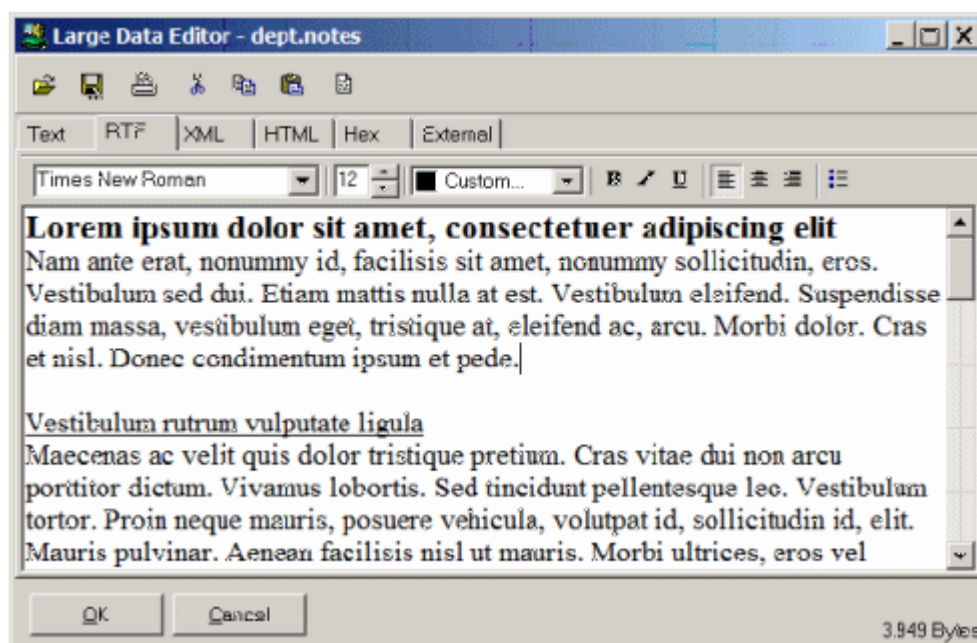
文本编辑器可以用来查看或者编辑任何类型的普通文本数据：



它也可以用来编辑 XML 和 HTML 文本。切换到 XML 或者 HTML 标签页将立即显示这些修改的结果。

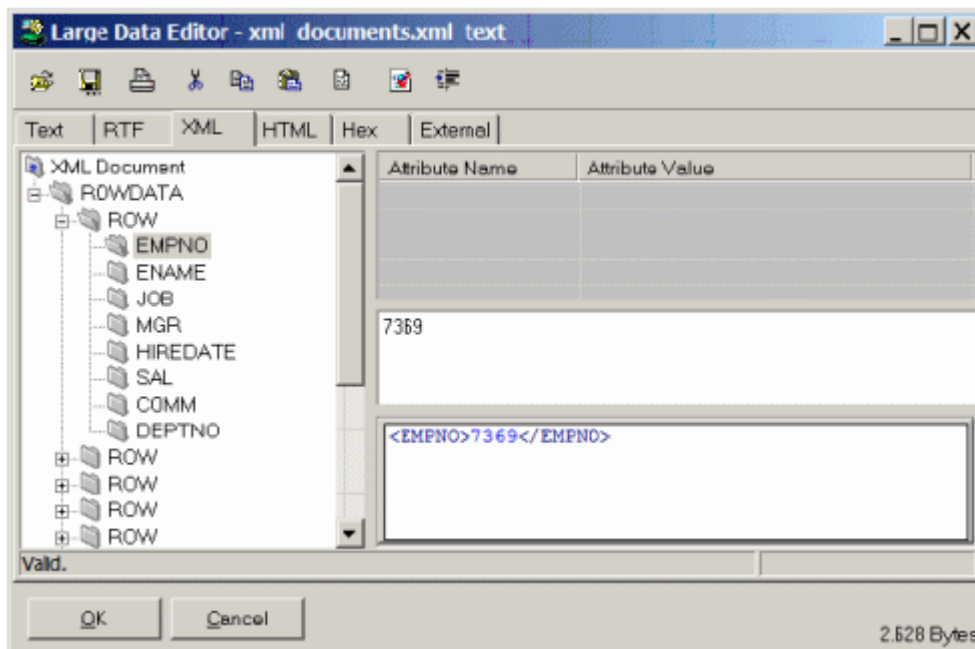
19.2 编辑 RTF (Editing RTF)

RTF 编辑器可以用来查看和编辑 RTF 格式文本。它允许你方便地修改字体类型，大小，颜色，字型和对齐：



19.3 编辑 XML (Editing XML)

当查看 XML 内容时，编辑器初始将切换到 XML 标签页：

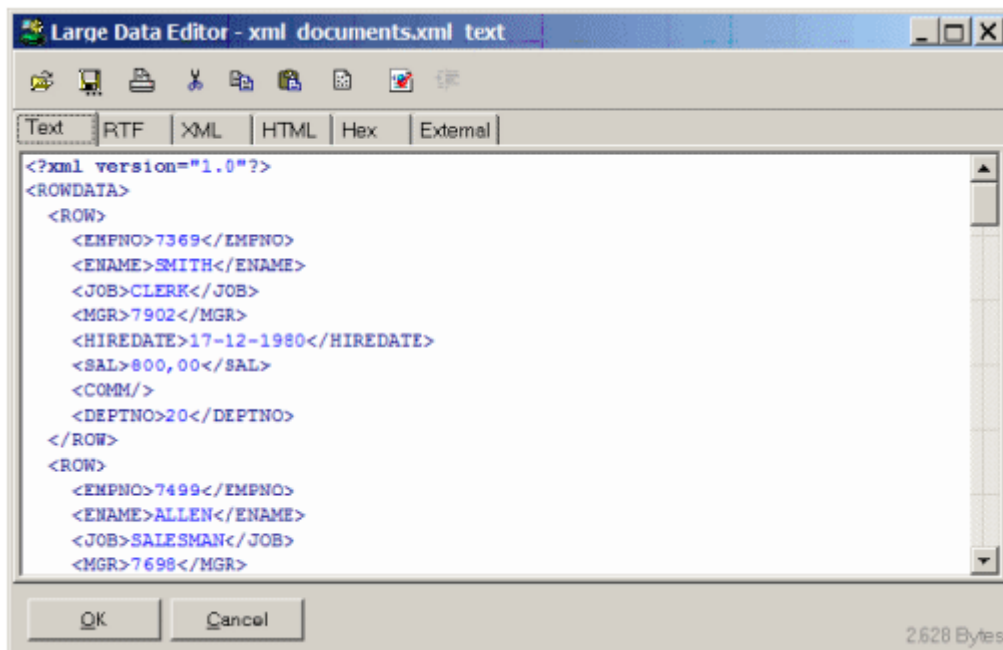


这里你可以在一个树型视图中看到 XML 文档的结构，和编辑单独的元素和属性。

按解析 XML (Parse XML) 键将解析 XML 文档，并将报告任何错误。这需要你有 Net8 8.1 或更新版本，并且用户能访问 SYS.XMLTYPE 类型或者 XMLPARSER 包。

你可以按工具条上的格式化 XML (Format XML) 键在该文档上应用标准格式规则。

如果你把对一个 XML 文档切换到普通文本格式标签页，则应用适当的语法高亮。



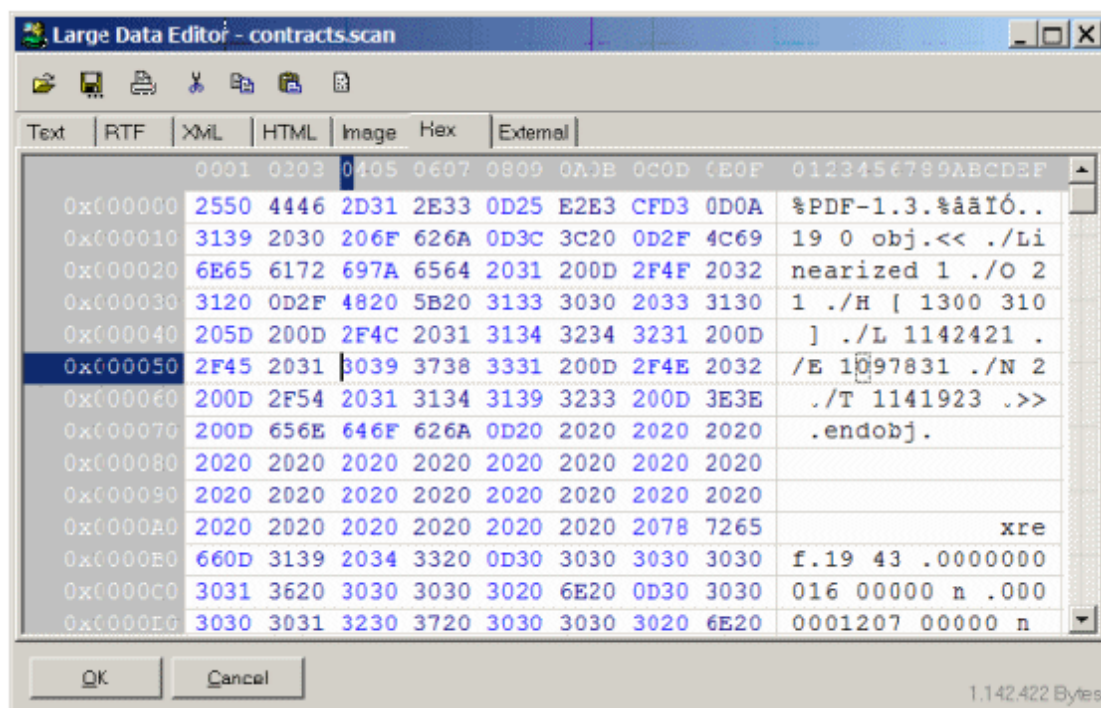
19.4 编辑图像 (Editing images)

图形编辑器将显示最流行格式的图像，如 JPG, BMP, GIF 等。如果图像比编辑器可以显示的大，图像将被调整大小适合显示。这不影响内容，只是一个显示效果。在图像下方可以看到图像格式和图像的像素大小。

修改一个图像，你既可以在工具条上按打开键从文件中载入它，或者从其他源中拷贝到剪贴板中再按粘贴键拷贝。注意当用剪贴板拷贝/粘贴图像时，将变成 BMP 格式！要确保原格式，你应该从文件中载入。

19.5 编辑十六进制数据 (Editing Hex data)

要以十六进制格式查看或者编辑编辑器内容，你可以切换到十六进制 (Hex) 标签页：

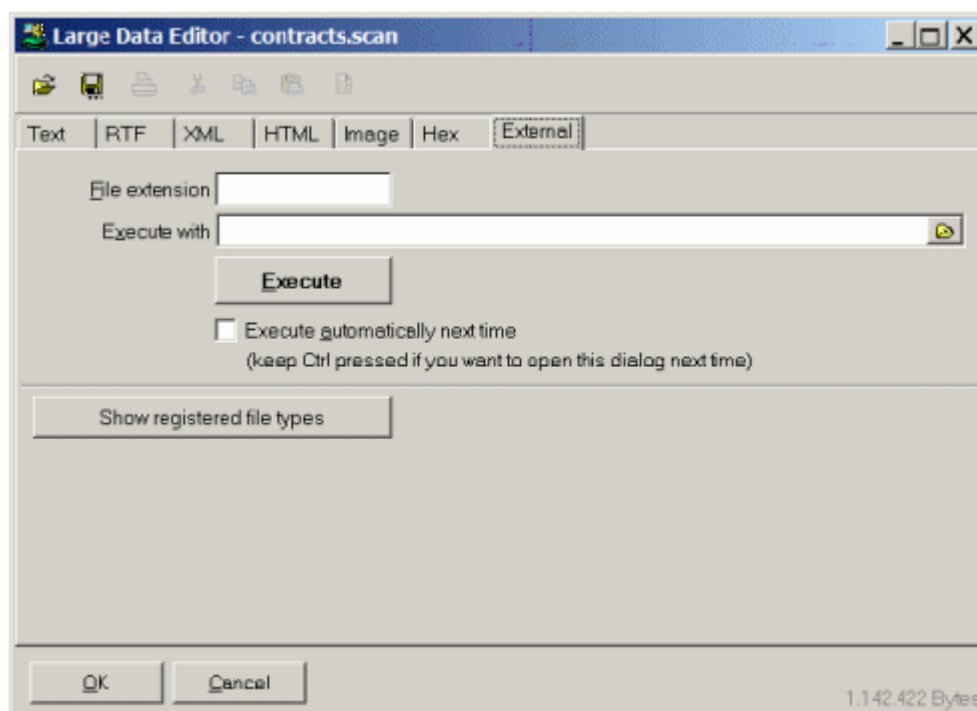


在左侧你看到 8 列十六进制数据字 (16bits)。右侧你看到十六进制数据的文本展现。你可以在十六进制和文本两种方式下编辑。

如果你在十六进制编辑器中右键点击，你可以选择一个不同的列布局。

19.6 调用一个外部阅读器或编辑器 (Invoking an external viewer or editor)

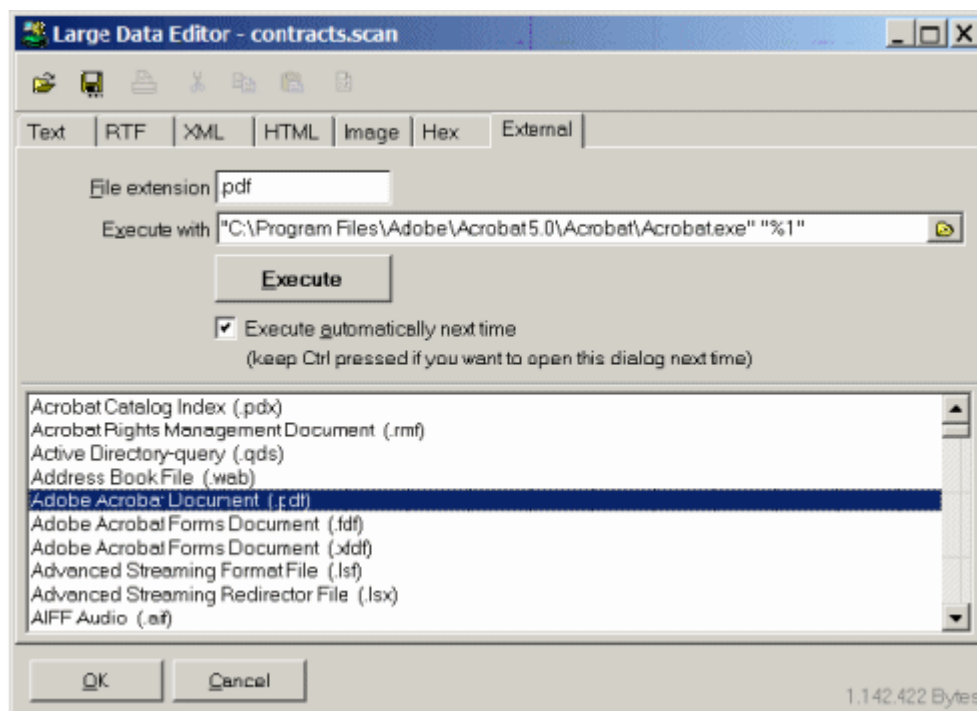
对于许多保存在数据库中的大对象，必须调用一个外部阅读器或者编辑器。例如，如果一个 BLOB 字段包含一个 Acrobat PDF 文件 (轻便的文档文件)的内容，这就必须调用 Acrobat Reader。当为这个内容调用大数据编辑器时，你最初将被带到 Text 或者 Hex 标签页。然后，你可以切换到外部 (External) 标签页：



这里你可以定义哪个程序应该被执行。首先你需要定义文件扩展名。在执行外部程序之前，一个带编辑器内容的临时文件将以这个扩展名保存。下一步你必须定义哪个程序将被执行。以后你可以按执行（Execute）键在外部程序中查看这个内容。如果你在外部程序修改了文件并保存，大数据编辑器将注意这个修改并将允许你导入这些修改到该编辑器中。

当查看字段数据时，表/字段和外部程序的关联为你下次使用而记住，如果你另外启用了下次自动执行（Execute automatically next time）选项，那么大数据编辑器将立即调用外部程序。要覆盖这个自动行为，在调用编辑器时按下 **Ctrl** 键。

大多数程序都注册了他们的文件类型，替代手工定义扩展名和程序，你可以简单地从一个列表中选择它。首先你必须按显示注册文件类型（Show registered file types）键，然后你可以查找正确的文件类型并双击它，对应一个 PDF 文件将是 Adobe Acrobat File:



如果你在文件类型列表上右键点击，你可以按扩展名或者描述（默认）排序。

20 查询设计器 (The Query Builder)

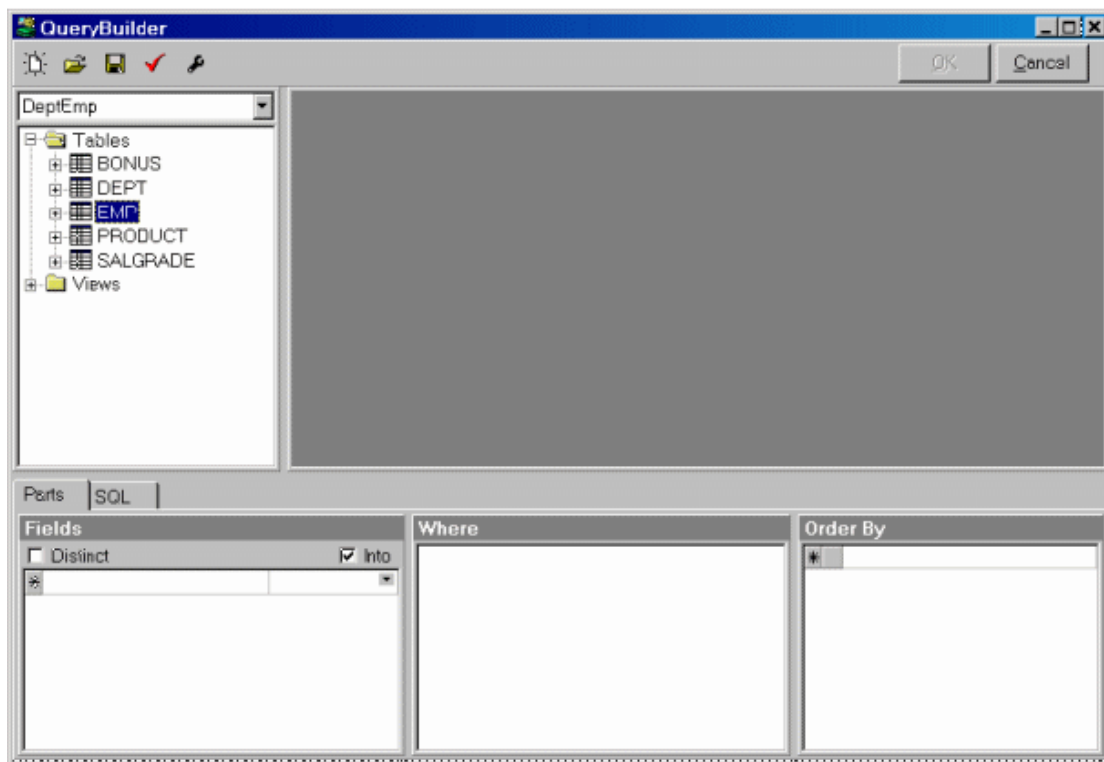
查询设计器允许你创建和修改你的 PL/SQL 和 SQL 源文件中的查询语句。它提供了一个简单的指向和点击界面来构造查询语句的字段列表，表列表，条件子句和排序子句。

20.1 创建一个新的查询语句 (Creating a new select statement)

对于这个例子我们将假设想创建一个在 dept 和 emp 表之间关联语句，并从两个表中显示一些字段：

```
select e.empno, e.ename, d.deptno, d.dname
from emp e, dept d
order by e.empno
```

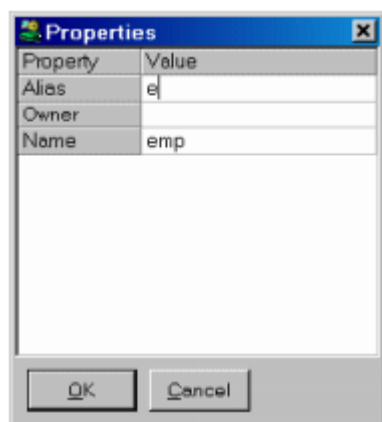
要创建一个新的查询语句，放置编辑光标在你想它被插入的文本位置，按工具条上的查询设计器 (Query Builder) 键（或者从工具菜单选择查询设计器）。这将调出一个空的查询设计器表单：



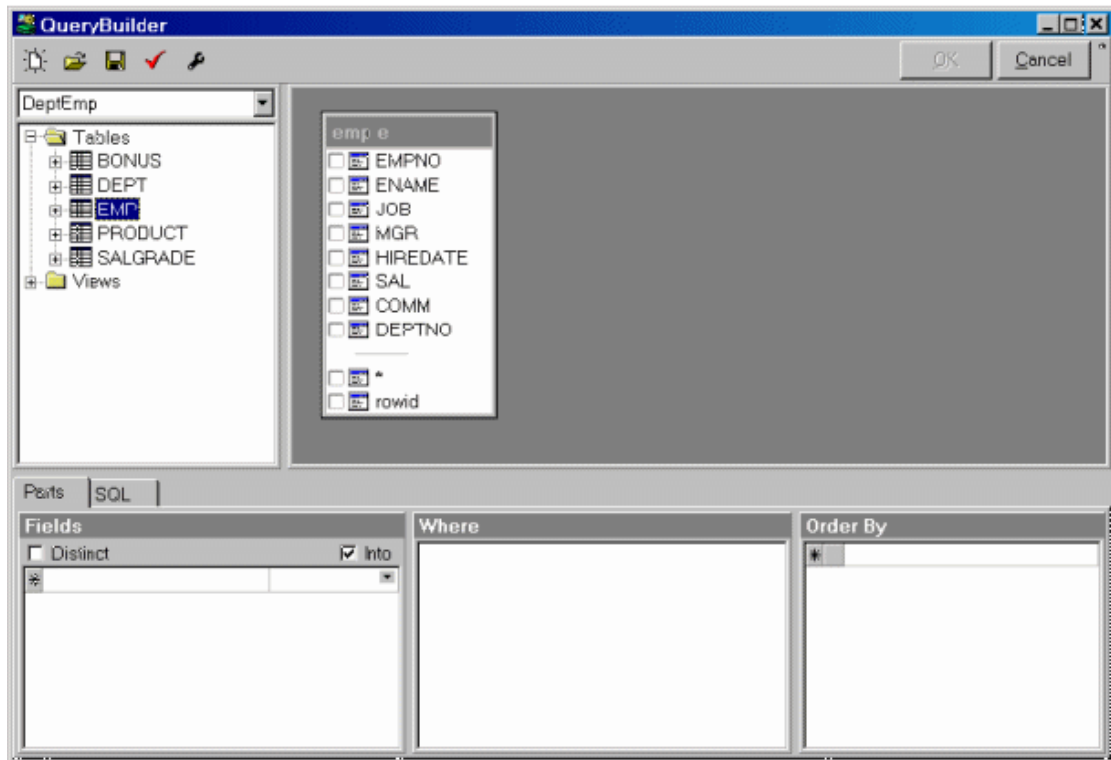
在左侧你看到一个只有表和视图的对象浏览器。右侧你看到当前是空的工作区，在下方你看到 3 窗格中有字段列表（和可选进入项目列表），条件子句和排序子句。底部包含一个标签页允许你切换到从当前查询定义中将要被创建的实际 SQL 文本。

在窗口的上方，你看到一个工具条，一个新建（New）键用来创建一个新查询，一个打开（Open）键用来从一个文件打开一个原保存的查询定义，一个保存（Save）键用来保存当前查询定义到一个文件，一个解析（Parse）键解析当前查询定义，和一个参数选择键。OK 键返回 PL/SQL 或者 SQL 编辑器并插入你已设计的 SQL 文本。

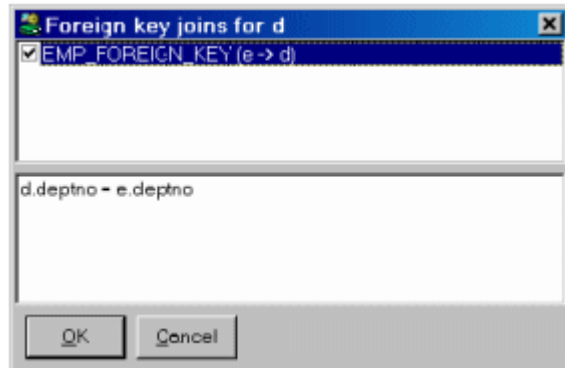
要包含表或者视图到查询语句，从对象浏览器拖动到工作区。每次你添加一个新表，将出现表的属性对话框：



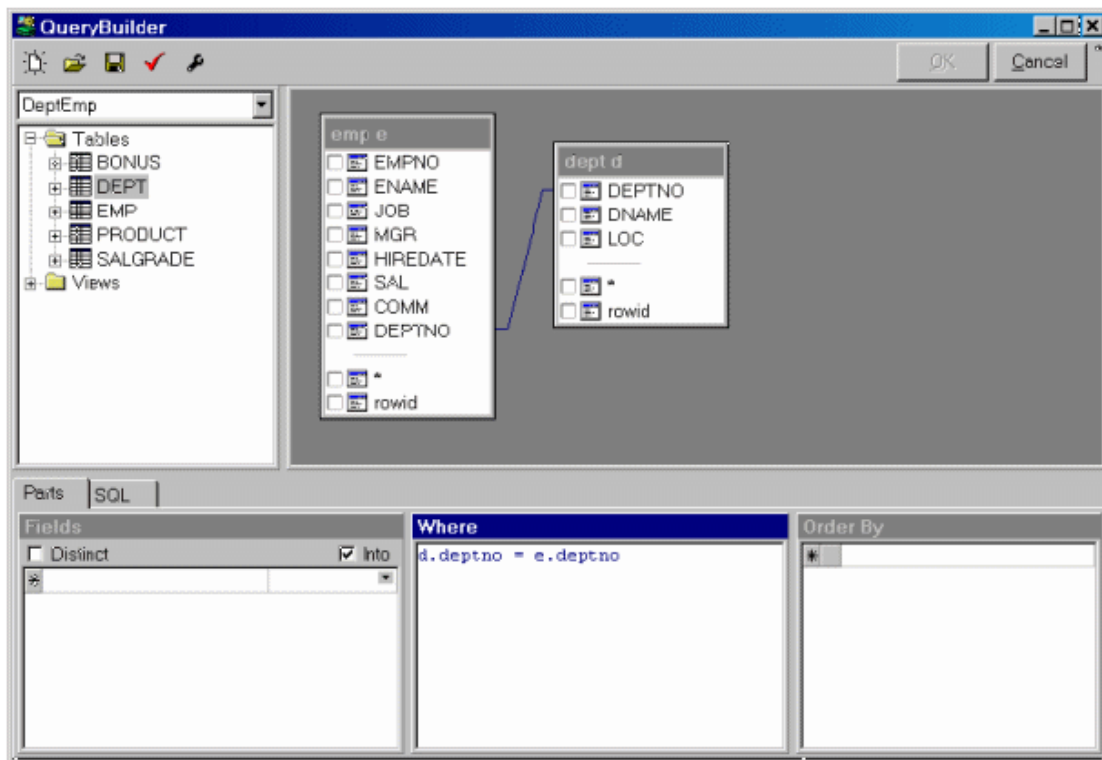
这里你可以输入一个表的别名。对应我们这个例子，我们想使用 e 代表 emp 表：



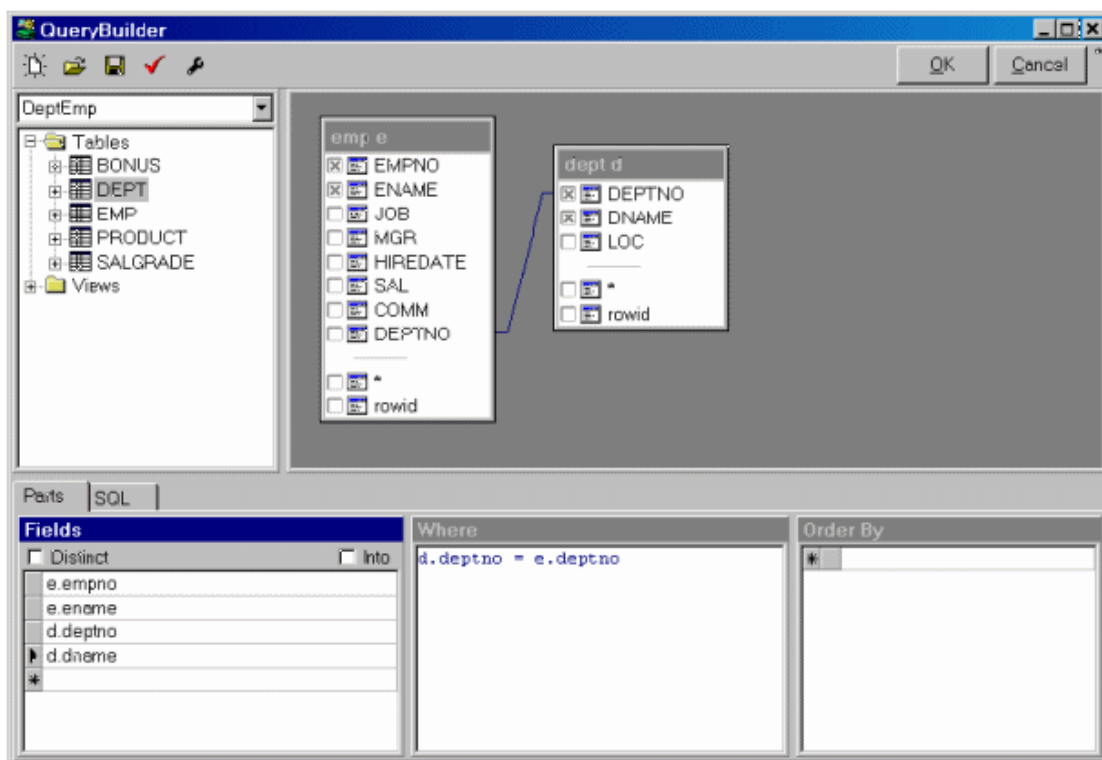
使用相同方法添加 dept 表后，你被询问是否想使用外键名为 EMP_FOREIGN_KEY 的字段关联 2 张表：



勾选这个选项后，你将在该表单的底部看到新的关联条件。
现在 Dept 表被添加，并且在工作区中关联条件被可视化：

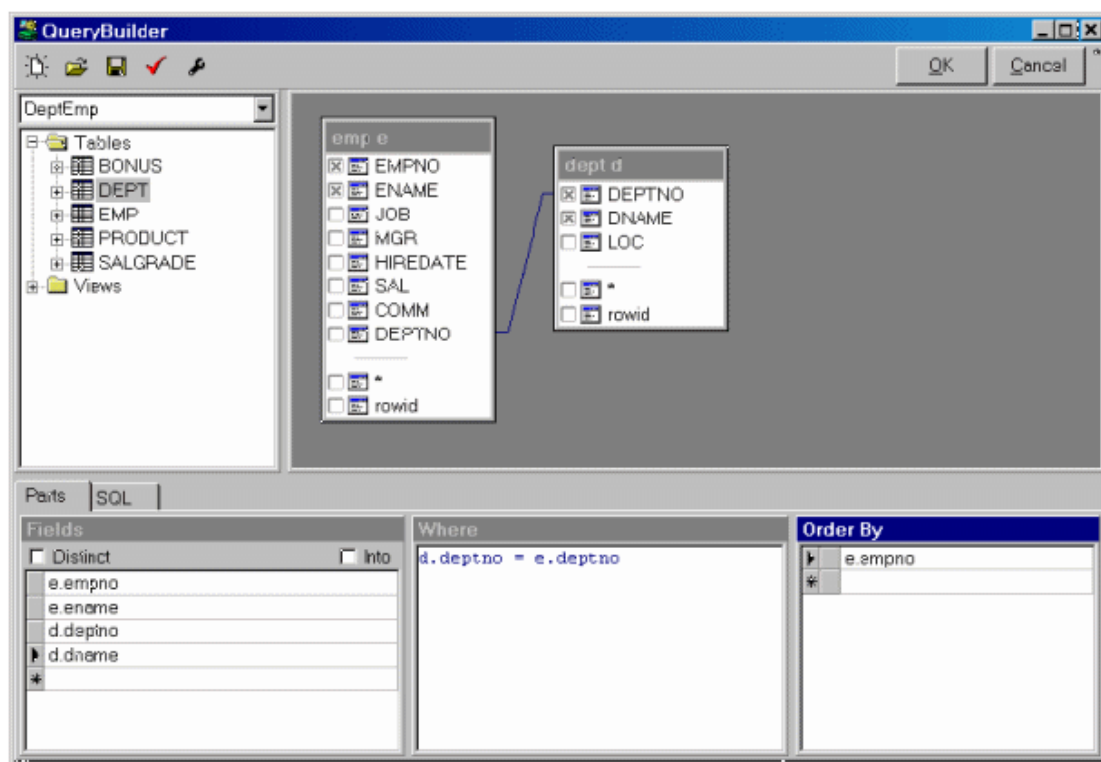


我们现在需要做的是使查询语句的字段列表中包含字段。只要点击字段名旁边的复选框。字段将被显示在字段窗格：



如果你想创建一个 PL/SQL select .. into 语句，你还要指定字段旁边的 Into 列表项目。选择列表将显示你插入这个语句的 PL/SQL 代码中的 PL/SQL 变量。在这里我们不想创建一个 Into 列表，所以我们不填写这些项目。

为创建排序子句，拖动 empno 字段到排序（Order By）窗格。



现在你可以按 OK 键插入这个 SQL 语句到编辑器中。

20.2 修改一个已存在的查询语句（Modifying an existing select statement）

要修改一个在你的 PL/SQL 或者 SQL 源代码中的查询语句，只要在查询语句上右键点击，从弹出菜单中选择查询设计器（Query Builder）菜单项。整个查询语句将在编辑器中被标识并且查询设计器将被显示。现在你可以像上一章节中描述的方法修改查询定义。

20.3 操作查询定义（Manipulating the query definition）

以下段落描述如何操作查询定义。

修改字段列表（Changing the field list）

要从工作区的一个表中添加一个字段到查询，你可以勾选字段旁边的复选框，或者你可以拖动字段到字段列表中。另一种方法是，如果字段列表获得焦点，你可以在字段上双击。

你也可以直接从对象浏览器拖动一个字段或者字段文件夹到字段列表中。字段文件夹位于表，限制和索引下。

你可以在字段列表中直接输入字段名和其他表达式。

要从字段列表中删除一个字段，右键点击它并从弹出菜单中选择删除选中（Delete selected）菜单项。如果你选中多个字段，你可以一次移除它们。

你可以通过选中字段和拖动到新位置移动字段到一个新的位置。

如果你想一个 `distinct` 结果集，在字段列表的上方勾选相应选项。

使用字段别名（Using field aliases）

要使用一个别名代表一个字段，在表达式输入别名，就像你手工输入查询语句时要做的一样。

修改条件子句（Changing the where clause）

条件子句是一个普通文本编辑器，在这里你可以输入条件。要包含一个字段，你可以从表拖动它到工作区，或者如果条件子句获得焦点时，你可以在字段上双击。

要增加一个基于外键限制条件的关联条件，在工作区中的表上右键点击并从弹出菜单选中外键（Foreign keys）菜单项。你不能使用这个功能移除一个已存在的关联条件，必须从编辑器中移除该文本才可以。

修改排序子句（Changing the order by clause）

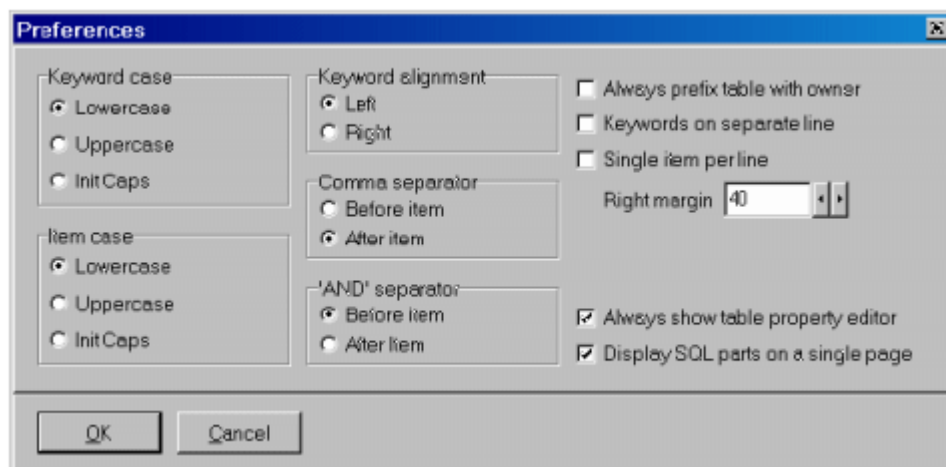
排序子句和字段列表的操作方法相同，除了你不能使用字段名旁的复选框。要修改项目升序和降序，你可以点击项目旁边的按键，它将显示相应的箭头。

使用一个表或者视图的同义词（Using a synonym for a table or view）

通过一个同义词使用一个表或者视图，在工作区中右键点击并从弹出菜单选择添加表（Add Table）菜单项。在名称属性中输入同义词，如果需要，提供一个别名，并按 OK 键。

20.4 查询设计器参数选择（Query Builder Preferences）

通过按工具条上的参数选择（Preferences）键，你可以修改查询设计器的行为。下面的表单将出现：



这些参数多数影响生成查询语句的布局：

- 关键字大小写（Keyword case）

确定 SQL 关键字 (select, from, into, where, order by) 如何插入到生成的 SQL 语句。

- 项目大小写 (Item case)

确定项目 (表名、字段名) 如何插入到生成的 SQL 语句。

- 关键字对齐 (Keyword alignment)

确定 SQL 关键字是左对齐或者右对齐。

- 逗号隔离 (Comma separator)

确定逗号放置在下一个项目之前, 还是当前项之后。这主要考虑是否每行一个项目 (Single item per line) 选项是否启用。

- AND 隔离 ('AND' separator)

确定在条件子句中将 'AND' 放置在下一个条件之前还是当前条件之后。

- 表前缀总带宿主 (Always prefix table with owner)

当启用, 表名前缀带宿主 (如 scott.emp 替代 emp)。当禁用时, 只有当表不是当前用户时, 表前缀带宿主。

- 关键字另起一行 (Keywords on separate line)

当启用, 一个 SQL 关键字另起一行, 并且所有项目以 2 个字符缩进放置在接下来的行。如果这个选项被禁用, 项目将在同一行中紧接着关键字。

- 每行一个项目 (Single item per line)

当启用, 每个项目将被放在一个新行中。当禁用, 项目将放在同一行中, 直到达到右边界。

- 右边界 (Right margin)

确定每行最大长度当每行一个项目 (Single item per line) 选项被禁用。

- 总显示表属性编辑器 (Always show table property editor)

当这个选项被启用, 在一个新表添加到查询定义后总显示表属性编辑器。当这个选项被禁用, 只有当新添加的表已经在查询中需要一个别名时才显示表属性编辑器。

- 在一个页中显示 SQL 部件 (Display SQL parts on a single page)

确定 3 个 SQL 部件 (字段列表, 条件子句, 排序子句) 是否在一个页上显示。如果这个选项被禁用, 每个部件将在 SQL 文本标签页后显示在独立的标签页中。

20.5 查询设计器插件 (Query Builder Plug-Ins)

如果一个查询设计器插件被安装, 则标准的查询设计器功能将调用插件。在查询设计器按键上右键点击允许你从安装的查询设计器列表选择一个查询设计器。当你再次使用查询设计器时, 最近使用的查询设计器将被调用。

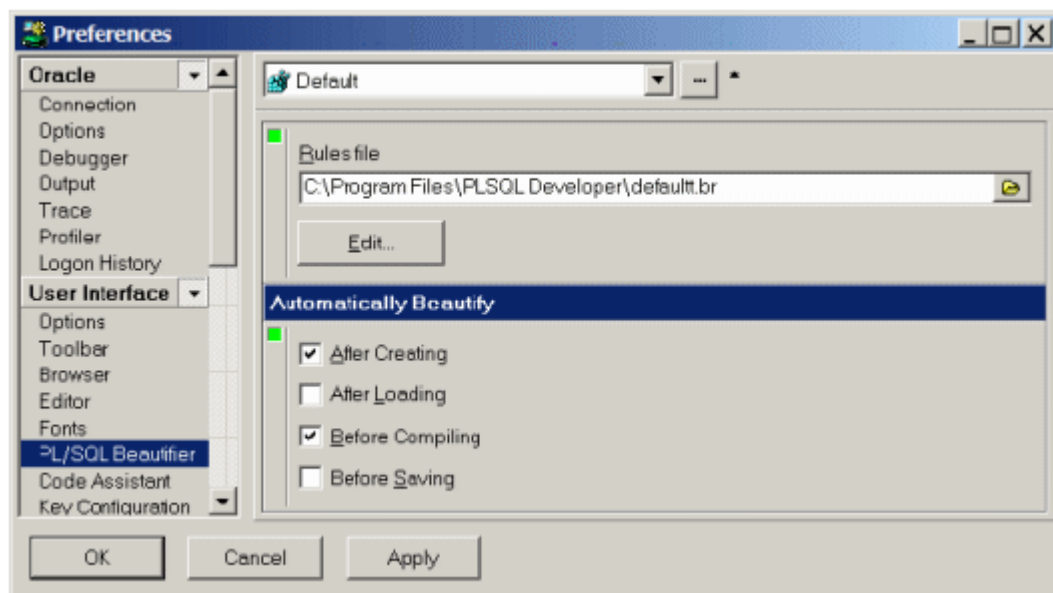
21 PL/SQL 美化器 (The PL/SQL Beautifier)

如果你在拥有许多开发者的项目团队中开发，维护和支持 PL/SQL 代码，所有项目成员使用相同的代码风格是有用的。这对质量和开发效率都将有益。要执行这个，PL/SQL Developer 包含一个 PL/SQL 美化器。

所有你必须做的是设置 PL/SQL 美化器选项，并通过 PL/SQL 美化器规则自行定义代码风格（你也可以简单地采用默认规则）。之后你的 PL/SQL 代码将在你工作时自动美化，或者通过激活美化功能明确地被美化。

21.1 定义选项 (Defining the options)

要定义 PL/SQL 美化器选项，从编辑菜单选择相应菜单项。将出现下面的参数选择对话框：



规则文件确定当你美化你的代码时将要遵守的规则。你可以不填写它采用默认的规则，它可以通过在 PL/SQL Developer 安装目录下的 default.br 文件被定义，更多有关这些规则的信息见下。

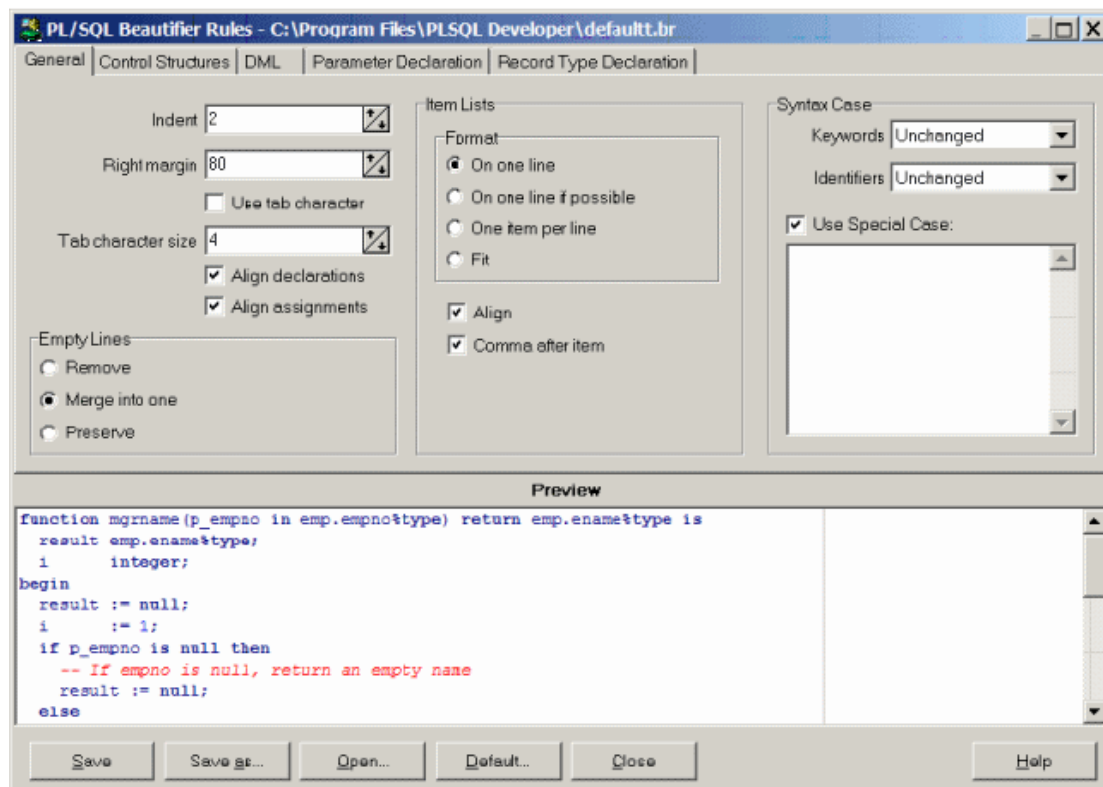
此外，你可以定义何时你的 PL/SQL 代码将被自动美化：

- **创建后 (After Creating)** — 每当一个新程序文件通过一个模版被创建，或者一个 DML 语句被从对象浏览器拖放到一个程序窗口中。
- **装载后 (After Loading)** — 一个程序文件被从文件系统中装载。这不影响从数据库中打开的代码。
- **编译前 (Before Compiling)** — 当你在程序窗口编译一个程序时，这保证所有在 Oracle 数据库中的 PL/SQL 代码符合规则。

- **保存前（Before Saving）** — 一个程序文件保存到文件系统前。

21.2 定义规则（Defining the rules）

如果你在选项对话框中按编辑（Edit）键，你可以查询或者编辑选中文件的规则：



调整这些规则后，你可以按保存（Save）键或者另存为（Save as）键保存这些变化。你也可以打开其他规则文件查看或者编辑，并且你可以按相应键恢复默认设置。如果你使用一个新名称保存文件，或者打开一个不同的文件，你必须在选项对话框中选择那个规则文件以使这些规则有效。

在常规标签页中你可以修改应用于你的 PL/SQL 代码的普通样式规则。你做的所有修改将立即反映在对话框底部的预览窗格中，大部分选项是自解释的，虽然下面选项需要一些解释：

- **缩进（Indent）** — 对于嵌套结构缩进的空格数，如 begin/end, if/then/else, loops 等。
- **右边界（Right margin）** — 每当代码需要转到下一行时，右边界将被当作一个准则。也有可能穿越右边界的情况，如当使用长字符串时。
- **使用跳格符（Use tab character）** — 当被启用，结果代码中将包含硬跳格符（字符 9）缩进代码，当被禁用，将使用空格。
- **跳格字符大小（Tab character size）** — 一个硬跳格符的字符数。这只有启用了使用跳格符选项时才有用，但是可以使用在含硬跳格符注释的准确对齐上。
- **空行（Empty Lines）** — 你可以定义空行应该被移除，美化器将使用自己的规则插入

空行，或者随后的一组空行应该合并成一个空行，或者保护全部空行。

- **项目列表 (Item Lists)** — 这些规则应用于所有没有归到其他标签页范畴的项目列表。
- **语法大小写 (Syntax Case)** — 确定 PL/SQL 关键字和标识符如何大小写。使用指定大小写 (Use Special Case) 允许你定义许多词，这些词将被包含在大小写描述中。例如，假设你在这个列表中包含 DBMS_Output，这将成为你在 PL/SQL 代码中的准确写法。

其他标签页包含指定你的 PL/SQL 代码的特殊规则。这些规则都是自解释的，并在预览窗格作示范。

21.3 使用美化器 (Using the beautifier)

如果你在选择对话框中没有选择任何美化事件，你可以通过选择编辑菜单中的 PL/SQL 美化 (PL/SQL Beautifier) 菜单项明确美化你的代码，按工具条上的相应键，或者使用一个分配给它的功能键。默认情况下，当前 PL/SQL 编辑器中的全部代码将被美化。如果在代码中有一个严重错误，它将不会被美化，你将得到一个错误消息。

你也可以选择一块完整的 PL/SQL 代码（如一个局部函数或者过程，或者一个 DML 语句），并激活美化器。

22 模版 (Templates)

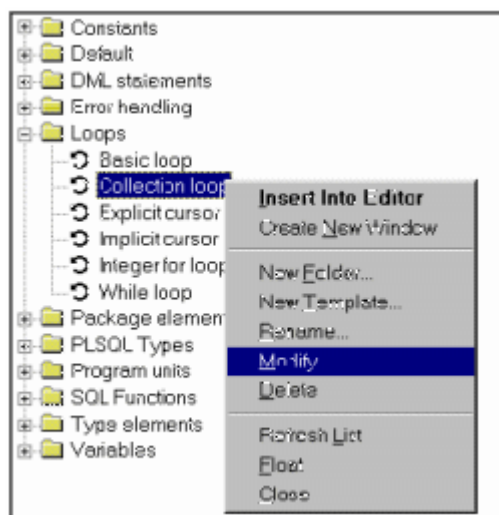
每当你创建一个新程序单元时，它的初始内容将基于一个包含预定义文本和变量的模版。当选择一个模版，你将被提示这些将被替换到模版文本的变量值。

模版也可以被用于插入部分 SQL 或者 PL/SQL 代码到已存在的源代码中。例如，你应该对一个 cursor-for-loop 创建一个模版，并在任何需要这样一个控制结构时使用它。

模版是普通文本文件，保存在 Template 目录并有一个.tpl 扩展名。对应所有程序单元类型的模版是预定义的，同普通的 SQL 和 PL/SQL 结构模版。你可以根据自己的特殊需要修改和创建模版。

22.1 模版窗口 (The Template Window)

模版窗口显示模版目录的层级结构。这便于组织你的模版。当你首次启动 PL/SQL Developer 时，模版窗口将停靠在对象浏览器下方。这样它总是可用并且不阻挡其他文档窗口视图：



有许多预定义的模版，并且它们被组织在下面模版目录中：

- 常量（Constants）—用于 PL/SQL 常量声明的模版。
- 默认（Default）—一个 SQL 窗口，测试窗口，命令窗口或者解释窗口中的默认内容模版。它还包含一个新视图对象模版。你不需要直接使用这个模版，但是你可以在这里维护它们。
- DML 语句（DML Statements）—你可以在 PL/SQL 中使用的 DML 语句模版。
- 循环（Loops）—各种类型的 PL/SQL 循环模版。
- 包元素（Package elements）—在包声明和包体中使用的元素模版。
- PLSQL 类型（PLSQL Types）—在 PL/SQL 中可以被定义的全部类型模版。
- 程序单元（Program Units）—这是一个包含新程序单元模版的特殊文件夹。每当你创建一个新程序单元，使用这个文件夹中和程序单元类型同名的模版。
- SQL 函数（SQL Functions）—全部 SQL 函数模版，像 `add_months`, `substr` 等。这些模版在当你不清楚一个指定函数的名称或者变量意思时有用。
- 类型元素（Type elements）—在类型声明或者类型体中可以使用元素模版。
- 变量（Variables）—PL/SQL 变量声明模版。

默认情况下，模版窗口停靠在对象浏览器下方，但是你可以通过右键点击它并从弹出菜单中选择浮动（Float）菜单项将它转成浮动窗口，要再次停靠窗口，遵循相同的程序。

你可以从弹出菜单中选择关闭（Close）菜单项关闭模版窗口。要再次使模版窗口可见，从工具菜单中选择模版（Templates）菜单项。

你可以为下次启动 PL/SQL Developer 保存这个布局，使用窗口（Window）菜单中的保存布局（Save Layout）菜单项，它将保存当前情况。这个功能在窗口布局（Window layout）章节中被详细描述。

22.2 使用一个模版 (Using a template)

要插入一个模版到一个已存在的源代码中，你只要在它上面双击。如果模版包含需要用户指定的变量，在文本被插入之前将出现一个输入表单。例如，假设你在函数实现 (Function implementation) 模版上双击，将出现下面的输入表单：

Template Wizard

Function Implementation

Purpose: Return the name of the manager of the employee

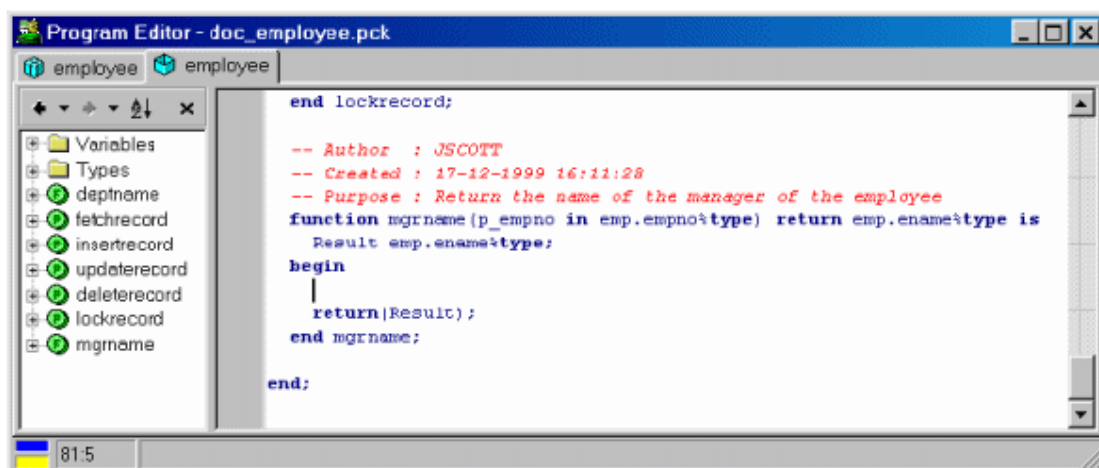
Name: mgrname

Parameters: p_empno in emp.empno%type

Return type: emp.ename%type

OK Cancel

填写表单后按 OK 键，模版文本将插入到源代码中的当前光标处：



你也可以从模版窗口中拖放一个模版到一个编辑器的指定位置。这个位置也将确定模版文本的缩进。类似地，你可以在编辑器的一个位置上右键点击，选择弹出菜单的模版 (Templates) 子菜单中的一个模版。

22.3 创建和修改模版 (Creating and modifying templates)

要修改一个已存在模版，在它上面右键点击并从弹出菜单中选择修改 (Modify) 菜单项。这将调出一个带模版文本的文本编辑器。这个文本包含模版的完整描述，包含变量，查询等等。模版的格式将在下面章节描述。

要创建一个新模版，在你想创建模版的文件夹上右键点击，从弹出菜单中选择新建模版 (New Template) 菜单项。你将先被提示输入模版名称，之后出现模版文本编辑器。你也

可以从弹出菜单中选择新建文件夹（New Folder）菜单项创建一个新文件夹。

模版文本（Template text）

模版描述包含文字文本和变量。文字文本将不经过任何修改被完整地拷贝到源文件中。考虑一个模版命名为 Commit，只包含文字文本 commit;当这个模版被双击，文本 commit;被插入到源代码的当前光标处。方括号代表在一个模版中的一个变量（见下），所以，如果你想使用一个方括号作为文字文本时，使用 2 个方括号替代，例如：[[option]]。

用户变量（User variables）

模版变量是在模版文本中的代码，它将被置换文本替代。这个置换文本可以是一个隐含值，像当前日期或者用户名，或者可以是用户指定的。模版被调用时，这些用户变量将呈现在输入表单中。以下例子为一个函数模版定义 Name 和 Type 变量：

```
create or replace function [Name] return [Type] is
begin
    return(result);

end [Name];
```

当这个模版被调用时，用户可以指定 Name 和 Type 变量值。像你看到的，Name 变量被使用了两次。可是用户将被提示输入一次，两个都使用同样值替换。

以下功能适用用户变量：

- 默认值（Default value）

一个变量的默认值只需放在变量名后。要定义 varchar2 作为函数默认类型，你可以用下面的声明：

```
[Type = varchar2]
```

- 受限制列表（Restricted list）

为限制一个变量的可能值，你可以定义它为一个值列表。要限制函数类型为 varchar2, number 和 date，你可以使用如下声明：

```
[Type = varchar2, *number, date]
```

数值前带星号是默认值。

- 建议列表（Suggestion list）

要提供一个用户使用一个建议值列表，同样允许其他值，在列表中使用...作为最后值：

```
[Type = varchar2, number, date, ...]
```

- 描述列表（Descriptive list）

代替在项目列表和置换文本中使用相同的文本，你也可以对每个项目使用一个描述。只要在描述后跟一个冒号 and 值：

[Level = Write No Database State:WNPS, Read No Database State:RNDS, ...]

- 复选框 (Check box)

对于有明确选项的变量，你可以使用一个复选框。在值间放一个斜线代表未选 (左) 和勾选 (右) 复选框。下面变量将在复选框未选时插入 `for each row`，在复选框勾选时插入 `for each statement`：

[Statement level? = for each row / for each statement]

如果你想在置换文本中使用特殊字符 (像逗号，括号等)，你可以将文本放置在双引号之间。

下面是一个触发器模版的例子，只使用用户变量：

```
create or replace trigger [Name]
  [Fires = before, after, instead of] [Event = insert, update, delete, ...]
  on [Table or view]
  [Statement level? = for each row/for each statement]

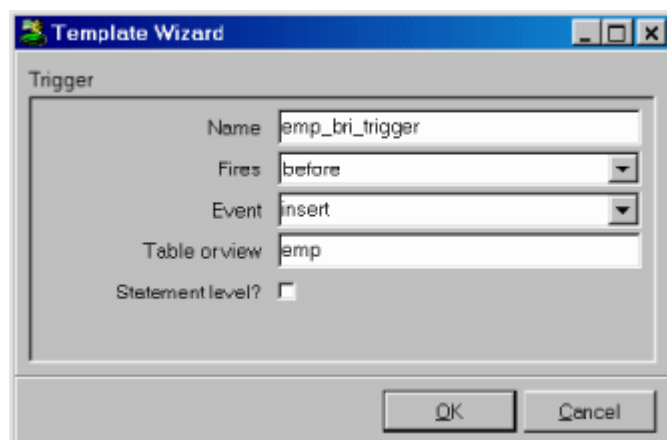
declare
  -- local variables here

begin
  ;

end [Name];
```

注意你可以创建多个程序单元在一个模版中，使用一个斜线(/)隔开它们。

当这个模版被使用时，将以下面方式提示用户提供变量值：



隐含变量 (Implicit variables)

除用户变量外，你还可以使用隐含变量。这些变量的置换文本不是由用户指定，而是被系统（日期，用户名），或者模版开发者（查询，文本）定义。

系统变量 (System variables)

下面 4 个系统变量被定义：

- \$OSUSER—操作系统用户名。
- \$DBUSER—当前登录数据库的用户名。
- \$DATE—当前日期。
- \$TIME—当前时间。

下面样例模版的前 2 行插入 Windows 用户和当前日期/时间到一个源文件中：

```
-- Author  : $OSUSER
-- Created : $DATE $TIME

-- Purpose : [Purpose]

procedure [Name] is

begin

    ;

end [Name];
```

注意系统变量在模版文本中不使用方括号，而是之前使用一个\$符号。

如果你想在结果文本中使用一个系统变量原名而不是置换值，你可以在名称前放第二个\$符号。例如：

```
-- $$Date$ $Revision$
```

结果文本将是\$Date\$ \$Revision\$。

光标位置 (Cursor position)

光标位置变量确定当模版文本插入到一个编辑器中后文本光标将被放置在何处。只要将[#]放在希望的位置：

```
loop
    [#]
end loop;
```

查询变量 (Query variables)

你可以在一个模版中使用查询组装选择列表。下面范例定义查询 seq_query，将被随后用作 Sequence 变量的一个建议列表：

```
[$QUERY seq_query =
```

```

select lower(object_name) from user_objects

where object_type = 'SEQUENCE'

order by object_name]

select [Sequence=$seq_query,...].nextval into [Variable name] from dual;

```

Sequence 变量是一个建议列表，因为查询后跟着 "...", 意味着用户可以手工输入其他值。查询结果可以简单地看作逗号分割的值列表。

包含的和排除的文本 (Included and excluded text)

你可以从一个模版中包含或者排除文本，依赖于其他变量的值。在下面的范例中，如果用户输入了 Search 条件，则添加 where:

```

select [Item list]

into [Variable list]

from [Table list]

[+Search condition=where] [Search condition];

```

于是，用户不用在查询条件中输入 where。要排除一个文字片段，使用-变量名替换+变量名语法。

文本变量 (Text variables)

你可以在一个模版中定义文本变量并在模版的其他部分引用这些变量。这对于你想选择性地包含大块文本相当有用。在下面范例中，如果用户指明，将包含一个 exception 块:

```

[$TEXT exception_block=
exception

when no_data_found then ...

when too_many_rows then ...

when others then ...

end;]

select [Item list]

from [Table list]

into [Variable list]

where [Search condition];

[Exception block =/$exception_block]

```

模版图标 (Template icons)

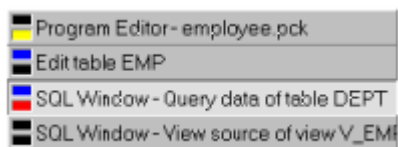
在模版窗口中的树型视图为每个模版显示一个图标。模版窗口首先查找和模版同名的位图。对应模版 `Cursor.tpl`，如果在相同目录下有 `Cursor.bmp` 文件将被显示。如果这个位图不存在，它将在当前目录下的目录中查找 `Cursor.bmp`。如果这个文件还不存在，它将在当前目录和目录下的目录中查找一个 `default.bmp` 文件。

模版的图标应该为 16 色 16x16 像素大小的位图。

23 窗口列表 (Window List)

窗口列表允许直接和方便地在多个文档窗口间导航。这个特性在你喜欢工作于最大化文档窗口时非常有用，如果你有大量打开的窗口，这样一个列表中会使导航容易一些。

默认情况下，窗口列表停靠在对象浏览器和模版窗口下方。它只在一个或者多个窗口被打开时可见，并显示这些窗口的名称和状态：



如果你点击其中一个项目，相应窗口将被激活。如果你在窗口列表中的一个项目上右键点击，将弹出下面的显示：



从这个弹出菜单你可用激活，关闭，和调整窗口到一个正常或者最小状态。你也可以在列表中的一个项目上 **shift** 点击关闭窗口。

此外，你可以重命名一个窗口。如果你刚创建了一个新的窗口并想提供一个有意义的名字时有用，例如，`SQL Window -New`。

新建 (New) 菜单项允许你创建一个新的程序窗口，SQL 窗口，测试窗口，命令窗口或者解释计划窗口。

要执行一个特定窗口，你可以选择执行 (Execute) 菜单项。

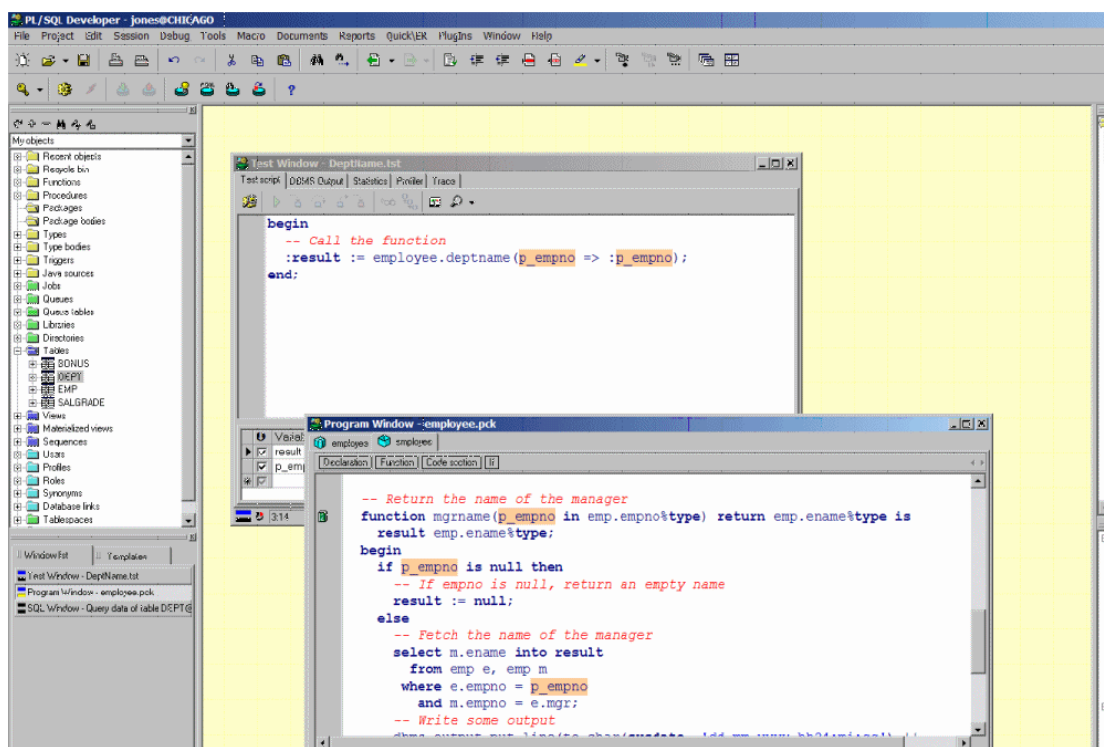
通过选择浮动 (Float) 菜单项，你可以将窗口列表转成浮动窗口。要再次停靠，，遵循相同的程序。要关闭窗口列表，选择关闭 (Close) 菜单项。要重新使窗口列表可见，选择窗口 (Window) 菜单中的列表 (List) 菜单项。你可以为下次启动 PL/SQL Developer 保存这个布局，使用窗口 (Window) 菜单中的保存布局 (Save Layout) 菜单项，它将保存当

前情况。这个功能在窗口布局（Window layout）章节中被详细描述。

24 停靠和浮动工具（Dockable and Floating Tools）

下面工具可以被停靠或者被展现在一个浮动窗口中：

- 对象浏览器
- 窗口列表
- 模版列表
- 书签列表
- 查找条
- 查找结果



要停靠一个工具，只要拖动它到工作区域的上下左右。一旦你接近工作区边缘，一个框将被显示，标识停靠位置。如果其他工具已经停靠在这个位置，你可以拖动该新工具在其他工具的上下左右，或者在其中。如果你拖动一个工具到另一个工具中，它们将变成同一个位置的不用标签页（见上面屏幕截图中的窗口列表和模版列表）。

要使一个工具浮动，只要拖动它从停靠位置到一个浮动位置。

25 授权（Authorization）

安装 PL/SQL Developer 后所有用户可以使用全部的 PL/SQL Developer 功能，连接数据库的 Oracle 用户限定在授予的系统权限和对象权限范围内。例如，Oracle 用户没有创建用

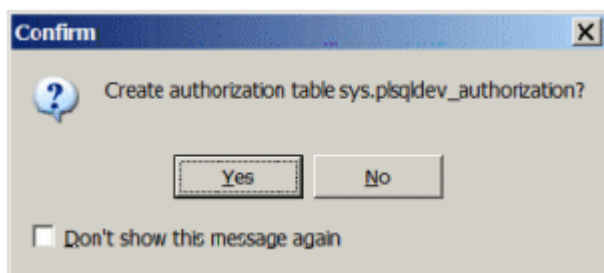
户的权限，PL/SQL Developer 用户可以启动创建用户功能，但是将最终得到一个从 Oracle 报出"ORA-01031, insufficient privileges"错误信息。

你可以明确授予全部相关的 PL/SQL Developer 功能给特殊的 Oracle 用户和角色。在一个数据库开发环境中，你将允许所有开发者拥有全部功能，然而在一个生产数据库中你一般应该对大部分用户禁用全部修改数据库或者耗费大量资源和影响性能的功能。

通过授予 PL/SQL Developer 权限给角色，你可以定制此授权给指定组。你可以使用已经存在的隐含映射到一个用户组的角色（如 DBA 和 RESOURCE）或者你可以创建 PL/SQL Developer 用户组的角色。

25.1 启用授权（Enabling authorization）

授权信息被存储在 sys 方案中的 psqldev_authorization 表。只要这个表在数据库中不存在，全部用户被授予使用所有 PL/SQL Developer 功能。如果你连接到一个还没有启用授权的数据库并且从文件（File）菜单中选择授权（Authorization）菜单项，你将得到下面询问：

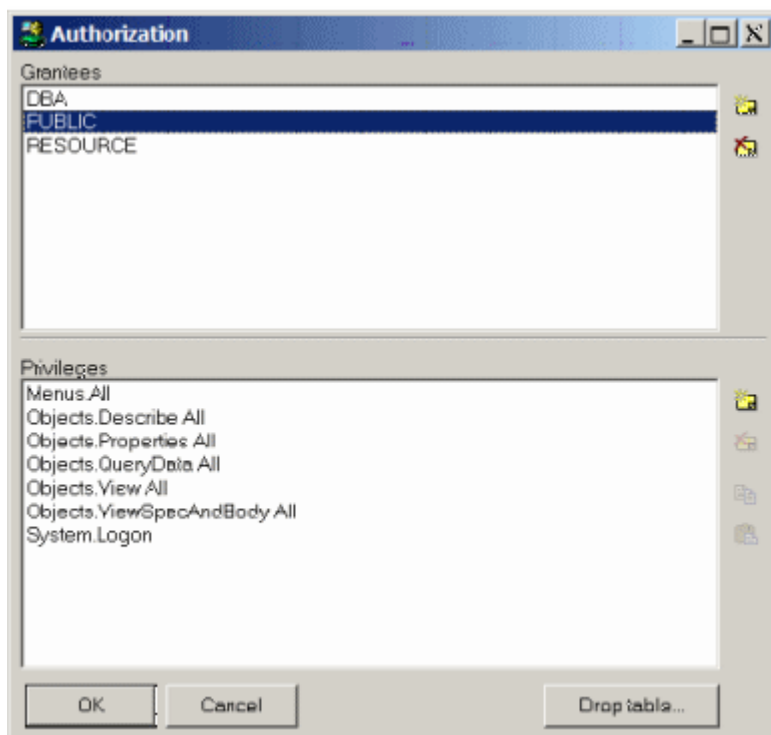


如果你选择 Yes，一个空授权表被创建。注意你必须以一个 DBA 身份连接才能创建这个表。一旦你在这个表中创建一个或者多个权限，授权被启用，并且只有拥有权限的用户可以在这个数据库中使用 PL/SQL Developer。

默认情况下，只有表的拥有者（sys）或者其他 DBA 可以修改表或者更新内容。全部用户有选择这个表的权限。

25.2 定义授权（Defining authorization）

要定义授权，你必须以一个 DBA 身份连接并从文件（File）菜单启动授权功能。将出现下面的对话框：



在顶部你将看到被授予一个或者多个 PL/SQL Developer 权限的全部用户和角色。在底部你看到当前选择的被授予者的实际权限。

要添加一个被授予者，按被授予者列表右侧的新建被授予者（New grantee）键。这键调出一个所有可能的被授予者列表（用户和角色）。你可以选择一个或者多个（Ctrl-点击或者 Shift-点击）被授予者并按 OK 增加他们。在此刻这些新被授予者还没有任何权限。

要移除一个被授予者和他的权限，按移除被授予者（Remove grantee）键。这将不影响实际数据库中的用户或者角色，而是仅仅从授权表中移除它的权限。

要授予权限，从列表中选择被授予者并按权限列表右侧的授予权限（Grant privilege）键。一个全部 PL/SQL Developer 权限的列表被显示，在列表中你可以选择一个或者多个权限。有 3 类权限：

- **系统权限（System privileges）**—PL/SQL Developer 系统功能（如登录）。
- **对象权限（Object privileges）**—数据库对象类型的特殊权限（如删除表）。
- **菜单权限（Menu privileges）**—使用 PL/SQL Developer 菜单的权限（如工具—会话）。

注意：如果一个用户没有系统登录或者全部权限，他或者她不能在当前数据库实例中使用 PL/SQL Developer。

你可以选择个别权限或者从一个权限层级集选择。例如：

- **All**—授予全部 PL/SQL Developer 权限。
- **Objects.All**—授予全部和数据库对象相关的 PL/SQL Developer 权限。
- **Objects.Rename All**—授予全部和重命名数据库对象的 PL/SQL Developer 权限（重

命名表，视图，序列和同义词）。

- **Objects.Rename Table**—授予重命名一个表的 PL/SQL Developer 权限。

注意对于数据库对象，Oracle 用户或者角色还必须有必要的执行操作的系统权限。

Granting Objects.Drop User 还需要删除用户的系统权限。

还要注意如果一个用户有 Drop User 系统权限，他还是可以从一个 SQL 或者 PL/SQL 脚本中执行删除用户命令。

25.3 禁用授权 (Disabling authorization)

要永久禁用授权，你可以删除 sys.plsqldev_authorization 表或者删除全部记录。要临时禁用授权，你既可以在表中废除权限或者重命名它（使 PL/SQL Developer 用户不可见）。今后要再次启用它，你可以再授予选择权限。

26 Oracle 文件系统 (Oracle File System (OFS))

OFS 允许你存储所有你的文件（源代码，脚本，报表等等）到 Oracle 数据库中。这样的好处是对象和操作这些对象的文件均保存在同一个数据库中，并且可以被所有可以存取这个数据的用户存取。而且，能够保证数据库对象和文件保持一致，和用一致的方法备份和恢复。

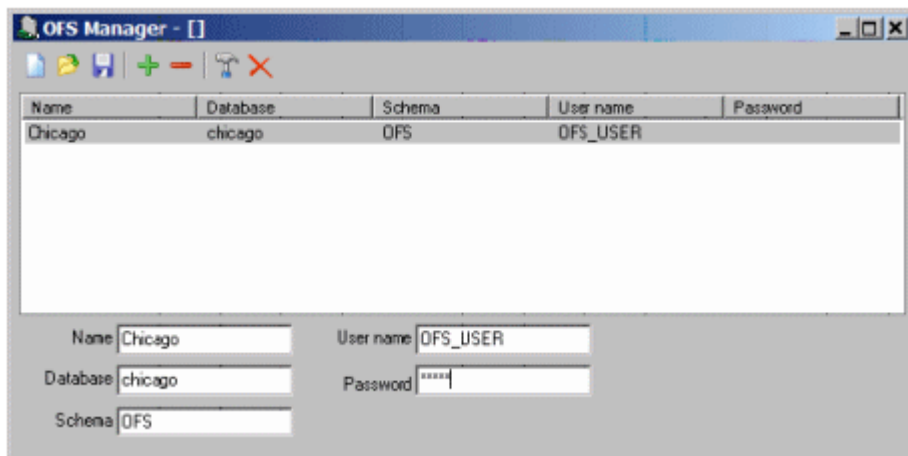
使用一个标准的带一些扩展名的文件打开/保存对话框，文件可以用通常的方法被保存和打开。

26.1 OFS 管理器 (OFS Manager)

任何人在 PL/SQL Developer 中使用 OFS 之前，你必须使用 OFS 管理器定义一个带一个或者多个 OFS 位置 (OFS Locations) 的 OFS 位置目录 (OFS Location Directory)。对于每个位置，你需要安装 OFS 数据库对象。

创建一个 OFS 位置目录 (Creating an OFS Location Directory)

要启动 OFS 管理器，转到 PL/SQL Developer 的参数选项 (Preferences)，选择目录页 (Directories page)。这个页底部你可以发现 OFS 管理器按键。启动它后你得到以下屏幕：



这是一个空的 OFS 位置目录, 你可以添加位置, 并保存到一个 OFS 位置目录文件(.ldf)。这个文件被用于 PL/SQL Developer 浏览目录并连接到位置和浏览文件。

添加 OFS 位置 (Adding OFS Locations)

要添加一个 OFS 位置, 按添加位置 (+) 键, 现在你可以输入下面信息:

- 名称 (Name) — 将被显示在文件选择器中的位置列表中。
- 数据库 (Database) — OFS 文件将被保存的数据库。
- 方案 (Schema) — 保持 OFS 数据库对象的方案 (表, 包等等)。这个用户需要 resource 角色权限并且为了创建它的对象还要拥有查询重写 (query rewrite) 系统权限。
- 用户名 (User name) — 用于存取 OFS 的 Oracle 用户。如果你没有指定一个 OFS 用户名, 则用户存取 OFS 时将被要求指定一个用户名和密码, 或者 OFS 将被作为 PL/SQL Developer 当前 Oracle 用户存取。
- 密码 (Password) — 用于存取 OFS 的 Oracle 用户密码。只有用户名输入了才有用。

创建了一个或者多个位置后, 你可以通过保存 (Save) 键保存 OFS 位置目录。如果你使用 OFS.ldf 保存在 PL/SQL Developer 目录, 则 PL/SQL Developer 将自动提取这个目录。否则, PL/SQL Developer 用户将必须在参数选项中指定正确的 OFS 位置目录文件。

安装 OFS 数据库对象 (Installing the OFS database objects)

添加一个位置到目录后, 你需要安装它的数据库对象。如果 OFS 方案和 (可选) OFS 用户在数据库中不存在, 你首先要定义它们:

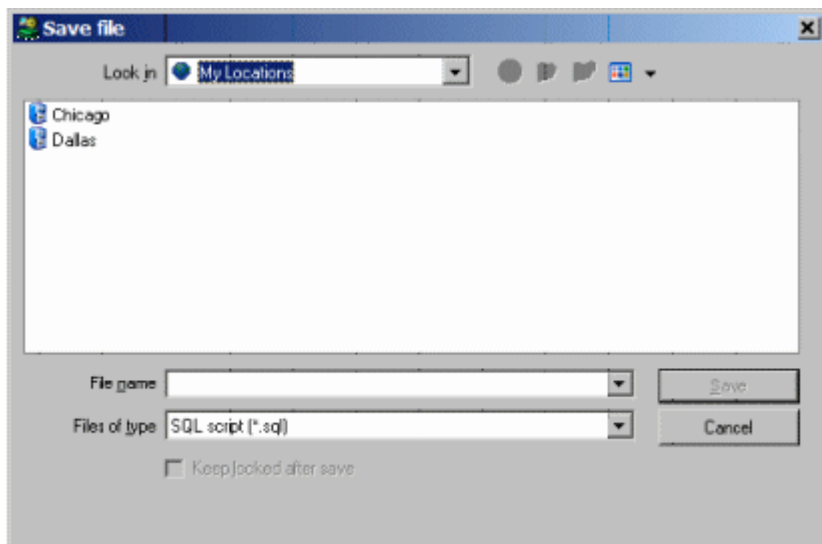
```
create user OFS identified by <password>;
grant connect, resource, query rewrite to OFS;
create user OFS_USER identified by <password>;
grant connect to OFS_USER;
```

下一步按安装数据库对象 (Install database objects) 键安装 OFS 数据库对象。提供方案

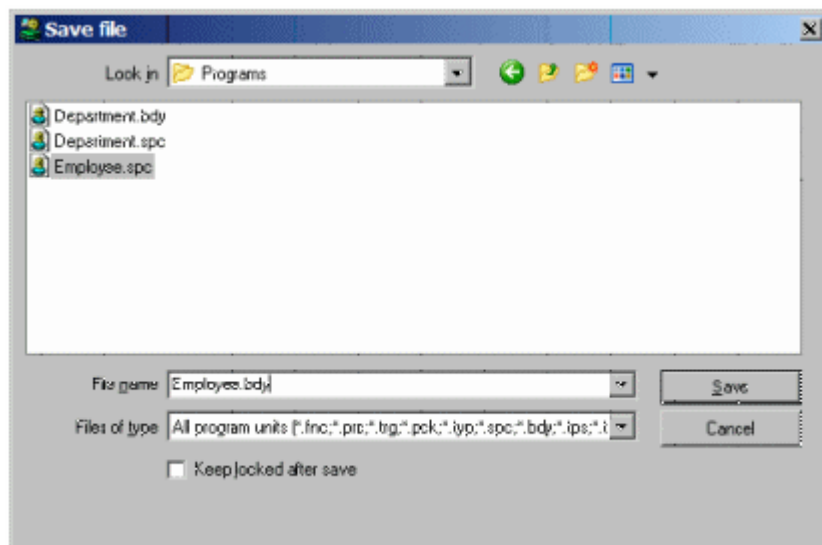
用户密码后，数据库对象将被创建，该位置已可以使用。

26.2 OFS 用法 (OFS Usage)

创建一个 OFS 位置目录，安装位置，并使 PL/SQL Developer 可用位置文件后，你可以从文件目录中使用 OFS 保存为 (OFS Save As) 菜单项保存一个文件到 OFS 中，将出现以下对话框：



你首先需要打开一个你可以保存文件或者创建目录的位置：



要从 OFS 中打开一个文件，你可以使用文件菜单中的 OFS 打开 (OFS Open) 菜单项，或者使用重新打开 (Reopen) 菜单。OFS 文件的路径名是 OFS:\Location\Path (如 OFS:\Chicago\Programs\Employee.bdy)。一个已打开的 OFS 文件可以使用标准的保存功能保存。你可以使用保存为 (Save As) 功能保存 OFS 文件到标准的系统文件。OFS 保存为 (OFS Save As) 功能可以使用不同名称或者不同目录保存到 OFS 中。

当保存或者打开一个文件，你可以选择保持锁定（Keep locked）选项锁定它。被你锁定的一个文件不能被其他用户锁定或者覆盖。

在文件选择器中右键点击一个文件，你可以选择剪切，拷贝，粘贴，删除或者重命名该文件。从这个弹出菜单中你也可以查看和修改对象属性。你可以修改的属性包括只读状态，锁定状态和压缩状态。只有在 Oracle10g 或者更高版本上文件才可以压缩。因为使用的 utl_compress 包从 10g 后才可用。

27 帮助系统 (Help systems)

你可以配置将 Oracle 服务器软件提供的帮助文件和手册集成到 PL/SQL Developer 中。这章描述如何配置和使用这些帮助系统。

27.1 MS 帮助文件 (MS Help files)

升级到 Oracle7.2 版本后，提供的联机手册用 MS 帮助格式。这些帮助文件可以在编辑器中提供上下文敏感帮助，可在一个词上右键点击，从帮助菜单选择 SQL 帮助或者 PL/SQL 帮助菜单项，或者按关联到这两个菜单项的一个功能键。

PL/SQL Developer 在以下两个目录中搜索帮助文件：

1. PL/SQL Developer 所在目录。
2. SQL*Plus 所在目录。

SQL 帮助文件名被假设为 sqlhelp.hlp，PL/SQL 帮助文件是 plshelp.hlp。如果这些帮助文件不可用，但是，如果你有其他帮助文件想用于 SQL 或者 PL/SQL 的帮助，你可以将这些文件放到 PL/SQL Developer 目录并将它们的文件名修改为上面提到的文件名。

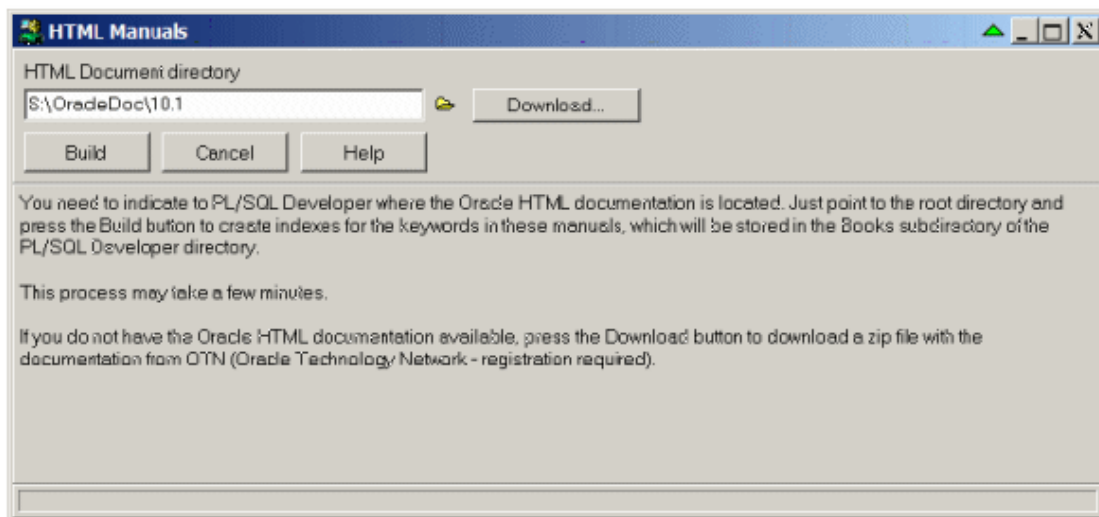
27.2 HTML 手册 (HTML Manuals)

从 Oracle7.3 版本开始联机手册提供 Acrobat Reader 和 HTML 格式。你可以设置 PL/SQL Developer 使用 HTML 手册，可在一个词上右键点击，从帮助菜单选择 HTML 手册，或者按 F1。此外，你可以在 Oracle 异常错误信息中按帮助（Help）键，这将自动带你到 Oracle 错误消息手册中的相应段落。在一个程序窗口双击一个编译错误也将显示这个信息。

注意：一些 Oracle 服务版本包含 1MB 以上的 HTML 联机帮助文件。装载这些文件耗费很长时间，建议使用一个 Oracle 服务器版本的联机帮助文档按章节分成较小的文件。Oracle 在 8.1.5 文档中再次减少了文件的大小。

设置 HTML 手册 (Configuring the HTML Manuals)

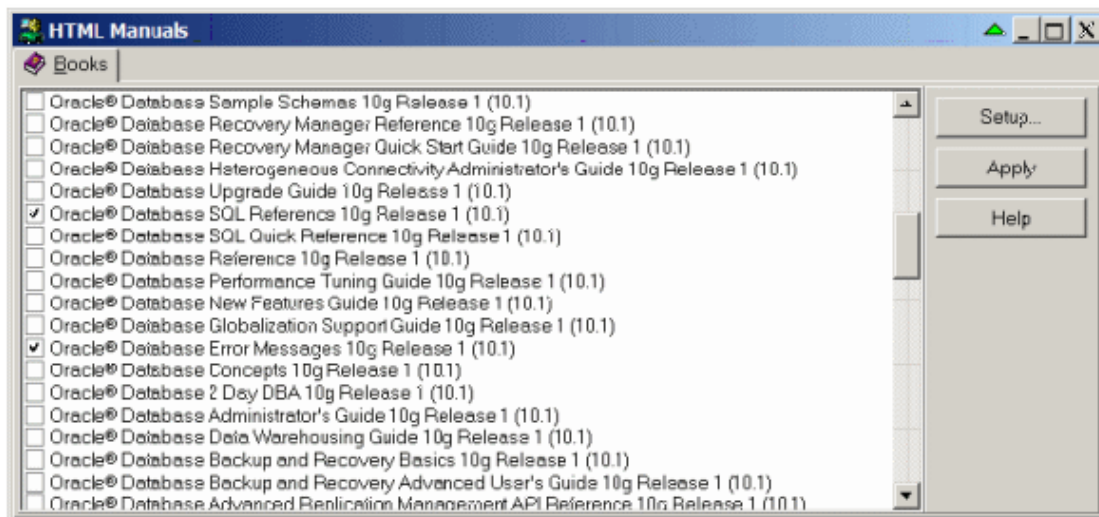
你第一次按上面描述的方法调用 HTML 手册时，你需要为 PL/SQL Developer 指定你的 HTML 文件位置：



只要指定根目录并按 **Build** 键创建这些手册的关键字索引，这将被保存在 **PL/SQL Developer** 目录下的 **Books** 子目录中。这个过程将持续一些时间。

如果你没有可用的 **HTML** 文档，你可以按 **Download** 键，这将带你到一个本地 **html** 页，从这里你可以选择适当的 **Oracle** 服务版本并下载一个 **HTML** 文档的 **zip** 文件。这个下载需要一个 **Oracle Technology Network** 账户。如果你没有一个 **OTN** 账户，你可以先免费创建一个。

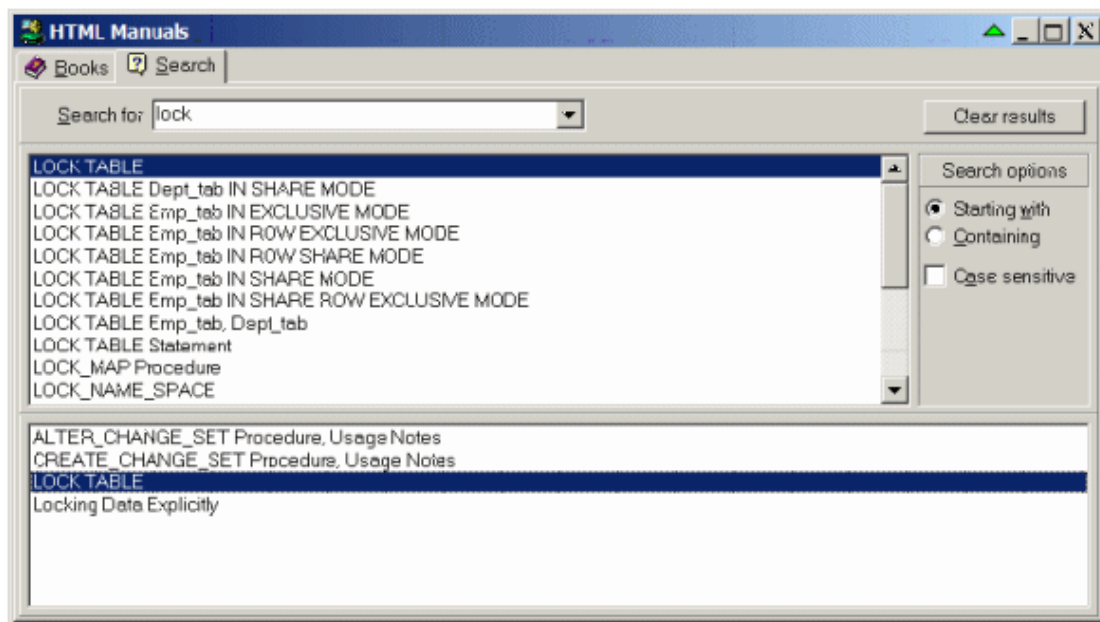
当这些索引构造完成后，你将得到一个已发现的手册列表。你现在可以选择你想使用上下文敏感帮助的书。默认这个选择将包含错误信息，**SQL** 参考和 **PL/SQL** 参考手册，这些在 **PL/SQL** 开发中都包含相关信息。如果你想在 **Oracle** 异常或者编译错误情况下自动得到可用的原因和动作，错误信息手册是需要的。当前的 **Oracle7** 手册没有允许这个特性的适用索引，而 **Oracle8** 版本没有任何问题。



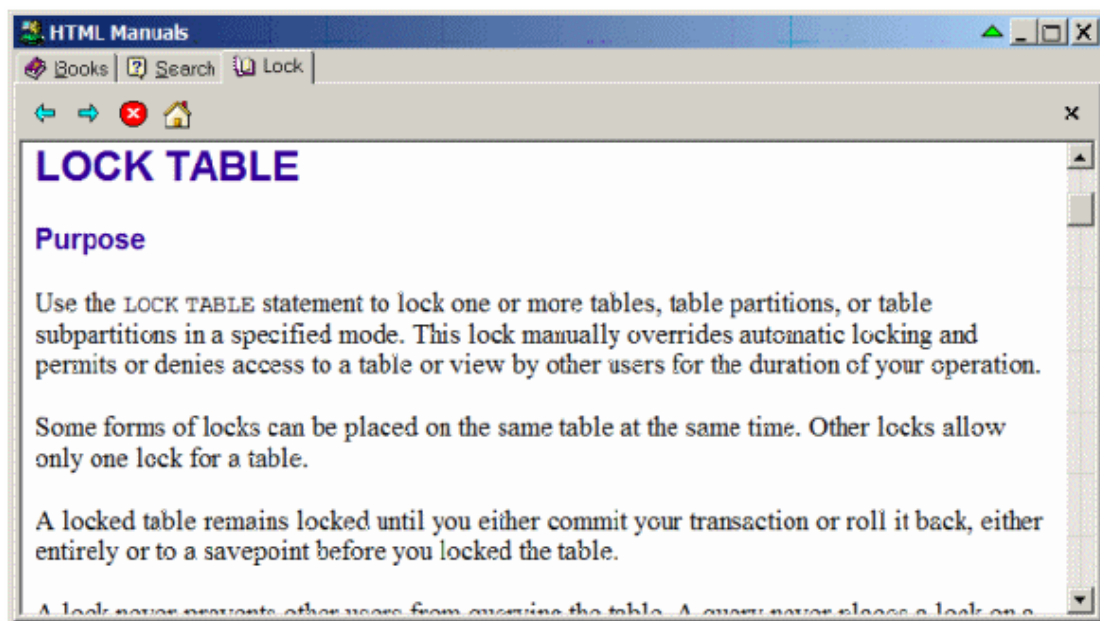
按 **Apply** 键使选择有效。之后你被转到查找页，这在下面章节讨论。通过按 **Setup** 键你可以扫描不同目录的 **HTML** 手册。

使用 **HTML** 手册 (Using the **HTML** manuals)

调用 HTML 手册后，你可以在查找页输入你想查找的关键字。两个列表显示匹配这个查找的关键字和这个关键字出现在手册中的主题：



在关键字列表右侧的 Search 选择窗格中，你可以改进搜索条件。下一步你可以在搜索页底部主题列表中的一个项目上双击显示在 HTML 手册中相应的段落：



搜索结果页显示一个 HTML 页，你可以像通常情况一样通过超链接导航。在页上方的键可以用来向前和向后定位你之前访问页，取消页积累，或者转到导航前的原始页。

你可以回到搜索页搜索不同关键字或者转到 Books 页选择不同书。每个搜索创建一个新的结果标签页，因此你可以保持不同搜索结果可用。要关闭一个结果页，按该标签页右上方的关闭键。要关闭所有页，在搜索页上按清空结果（Clear results）键。

默认情况下，HTML 窗口将总在 PL/SQL Developer 集成开发环境的其他窗口前端，所

以信息保持可见。要临时隐藏该窗口，你可以使用窗口右上角的绿色卷起键。

28 定制 (Customization)

PL/SQL Developer 的有些方面可以被定制以符合你个人的需要。这些方面在以下段落描述。

28.1 参数选择 (Preferences)

在工具菜单中，有一个参数选择菜单项允许你设置各种 PL/SQL Developer 参数。这些参数在参数选择 (Preferences) 章节中详细描述。

28.2 窗口布局 (Window layout)

有可以专门制作你的需要的各种各样窗口布局设置。从窗口菜单中选择保存布局 (Save Layout) 菜单项保存这些设置。布局设置包括：

- 应用窗口的大小和位置。
- 对象浏览器的出现和大小。
- 程序窗口代码目录的出现和大小。
- 窗口列表和模版窗口的出现，状态（浮动或者停靠）和位置。
- 程序窗口，测试窗口，SQL 窗口，解释计划窗口，命令窗口，报表窗口，编译无效对象窗口，查找数据库对象窗口，导出用户对象窗口和比较用户对象窗口的大小。

当你保存布局，最后打开的窗口大小被保存。如果有些类型的窗口不存在，那个窗口类型的大小设置不被改变。

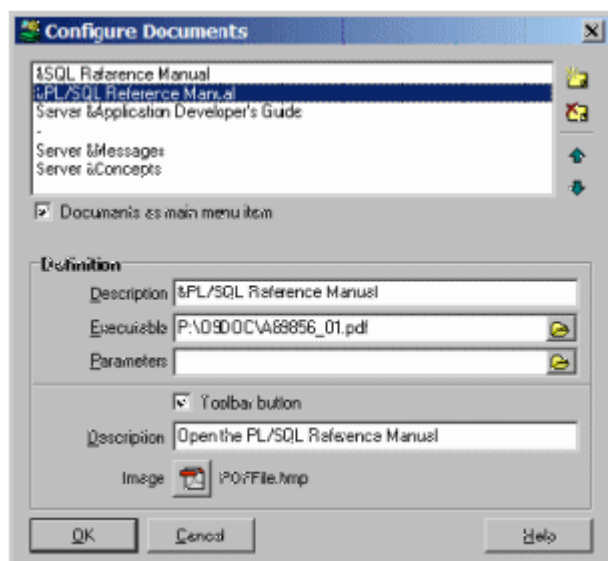
28.3 联机文档 (On-line documentation)

大多数 Oracle 的文档作为联机文档被提供。通过时间积累，这些文档有几种格式。迄今为止，有 MS 帮助文件，Oracle 书文件，Adobe PDF 文件和 HTML 文件。将来，Oracle 也许选择其他格式。

此外，你也许有一些公司标准和项目文档使用一个文字处理器格式并被经常使用。

所有这些联机文档都可以通过包含它们到文档菜单中集成到 PL/SQL Developer 的集成环境中。这样就可以通过一个简单的鼠标按键来使用它们。

要设置联机文档，在工具菜单中选择设置文档 (Configure Documents) 菜单项。下面对话框允许你为每个文档定义一个描述，和如果查看它：



在右侧你可以看到四个键，创建一个新文档，删除一个文档，和上下移动一个文档。当你创建或者修改一个文档时，你必须提供以下信息：

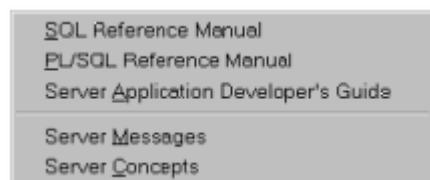
描述（Description），显示在菜单上。为一个文档菜单项创建一个菜单项快捷键，在描述中的字符前加一个&符号。如果你想使用&在描述中，使用两个&。如果你在描述中只输入一个'-'，它将变为一个菜单分隔符。使用这个方法可以将菜单逻辑分组。

执行程序（Executable），必须是你想查看文档的完整路径程序。如果文档已注册了一个文件类型（像一个 MS Help 文件或者 HTML 文件），你可以在这里输入文档的路径。

参数（Parameters），如果它是一个程序，可以用来传入信息。它至少包括文档的路径，但是也可以用来向程序传递选项。

当勾选了 **Documents as main menu item** 选项，Documents 菜单被放置在主菜单中。如果没有勾选，它将放置在工具菜单下。

在上面的例子中，PL/SQL Reference Manuel 是一个 PDF 文件（P:\O9DOC\A89856_01.pdf），是一个注册的文件类型，所以在 Executable 中可以只输入文件名。字母'P'作为菜单的快捷键。文档菜单看上去像这样：



当启用工具条工具（**Toolbar button**）选项，文档也可以被包含在工具条上。相应的描述将在你保持鼠标光标在该工具条按键上时作为提示显示。如果你没有填写描述，该文档的主描述将被使用（像在主菜单中的显示）。按图像键为工具条按键选择一个 Windows 位图文件（*.bmp）。这个图像适合的大小是 20 x 20 像素。注意 PL/SQL Developer 将总在原位置加载位图文件，所以在没有修改相应文档情况下不能移动或者重命名这个文件。PL/SQL

Developer 带有很多标准的位图文件，你可以从中选择。这些文件位于 PL/SQL Developer 安装目录下的 Icons 子目录中。这是图像选择器的默认目录。

28.4 命令行参数 (Command-line parameters)

下面参数可以被用在 PL/SQL Developer 命令行。默认快捷键不包含任何参数，只是从 PL/SQL Developer 安装目录中启动 plsqldev.exe。

一个默认参数集可以通过 PL/SQL Developer 安装目录中的 params.ini 文件定义。你可以在一个文本编辑器中编辑该文件，如 notepad。该文件包含每个参数的一个解释。如果一个参数被定义在 params.ini 中并在命令行上，则命令行优先使用。

userid

每次你启动 PL/SQL Developer，它将提示你提供至少是密码。要避免它，你可以提供一个 userid 变量带熟悉的 username/password@database 格式：

```
plsqldev.exe userid=scott/tiger@chicago
```

注意你也可以使用下面的注册键来提供一个默认登录：

```
HKEY_CURRENT_USER\Software\Allround Automations\PL/SQL Developer\Logon
```

在这里你可以添加一个用户名，密码和数据库。

最后一个方法的优点是允许你双击一个 PL/SQL Developer 注册文件后自动完成登录：

nologon

这个变量禁止 PL/SQL Developer 启动后显示登录对话框。这个变量没有任何自变量：

```
plsqldev.exe nologon
```

oraclehome

你可以在命令行中指定 Oracle Home 名，从而覆盖 Primary Oracle Home（默认使用）和 Oracle Home 参数选项：

```
plsqldev.exe oraclehome=ora817
```

注意这是 Oracle Home 名，像在注册表中的 ORACLE_HOME_NAME 指定的一样。这个名称也被 Oracle Home Selector 使用。它不是一个注册部分名或者一个直接名。

dontadjustpath

PL/SQL Developer 将不临时修改 Oracle Net 的 PATH 初始值：

```
plsqldev.exe dontadjustpath
```

nosplash

这个参数禁止 PL/SQL Developer 被启动时显示闪屏。这个参数不需要自变量：

```
plsqldev.exe nosplash
```

noplugins

指定这个变量将禁止任何插件被装载：

```
plsqldev.exe noplugins
```

library

指定报表的风格库位置：

```
plsqldev.exe library=p:\standard.lib
```

prefpath

指定个人参数选项集的路径。例如：

```
plsqldev.exe prefp=\\u:\userdata
```

详见参数选择集（Preference sets）章节。

commandfile

在命令窗口中执行指定命令文件。例如：

```
plsqldev.exe userid=scott/tiger commandfile="u:\sql scripts\demo_build.sql"
```

这个命令将首先联机数据库和随后执行 demo_build.sql。

viewobject

查看指定数据库对象。它还需要你指定 userid 参数。例如：

```
plsqldev.exe userid=scott/tiger@chicago viewobject=emp
```

这个命令将首先联机数据库，随后显示 Emp 表的表定义。注意对象说明也可以包含宿主（如 SCOTT.EMP）。

editobject

编辑指定数据库对象。它还需要你指定 userid 参数。例如：

```
plsqldev.exe userid=scott/tiger@chicago editobject=emp
```

这个命令首先联机数据库，随后显示 Emp 表的表定义。注意对象说明也可以包含宿主（如 SCOTT.EMP）。

registry

要使用 IDE 设置的不同集，你可以使用这个参数。例如：

```
plsqldev.exe registry=home
```

这个命令将导致所有 IDE 设置（窗口大小和位置，文件历史，登录历史等等）从一个 home 名装载和存储。

28.5 SQL, PL/SQL, Command, Java 和 XML 关键字 (SQL, PL/SQL,

Command, Java and XML keywords)

在 SQL, PL/SQL 和命令行编辑器中高亮的关键字可以定义在关键字文件中。在 PL/SQL Developer 安装目录中, 你可以发现文本文件 sql.kwf, plsql.kwf, command.kwf, java.kwf 和 xml.kwf。你可以简单地使用一个文本编辑器在相应部分修改, 添加或者移除关键字, 如记事本。

28.6 插件 (Plug-Ins)

你可以编写自己的插件来扩展 PL/SQL Developer 的功能。一个插件的功能可以添加到 PL/SQL Developer 菜单中, 并可以执行任何类型的动作: 存取数据库, 对象浏览器, 当前窗口和编辑器等。

一个插件是一个 DLL, 暴露一个指定接口到 PL/SQL Developer。要创建一个插件, 你可以使用任何可以创建 DLL 的编程语言。当一个插件 DLL 放置到 PL/SQL Developer 的 Plug-In 目录时, 它将自动被拾取。这使得发布你的插件很容易。

你可以为自己或者公共使用构造插件。Allround Automations网站上的一些插件是可用的 (<http://www.allroundautomations.com/plsqldev.html>)。这里的第三方插件也是可以使用的。

在 PL/SQL Developer 根目录下的 PluginDoc 子目录中的插件文档和例子是可用的。