

POWERBUILDER

Client/Server

Internet/Intranet

应用开发用户手册 卷 I

晓通网络数据库研究所

第八章 PowerBuilder

DataWindow 对象的引用

在 PowerBuilder 应用中, 可以用 PowerScript 来对数据窗口对象中的对象以及数据窗口中的数据进行动态的引用。通过数据窗口对象中对象的引用, 可以使对象的属性在程序的运行过程中发生变化, 从而可以根据数据库中的不同数据和用户的不同反应给用户展现不同的数据窗口形态; 通过对数据窗口数据的访问, 可以进行各种各样特殊的数据处理, 例如可以设计一个多级检索程序, 用户每次都是根据上一次的检索结果, 进一步给出检索条件, 以致最终得到希望的结果。所有这些都是通过表达式来实现的。下面先讲数据窗口表达式的写法。

本章配有例子盘, 读者可以根据例子盘的实际运行效果进一步体会数据窗口表达式的作用。示例中用到的窗口、数据窗口、结构、用户对象、菜单等都放在 Ddws.pb1、Ddws_1.pb1、Ddws_2.pb1 和 Ddws_3.pb1 中。

8.1 数据窗口表达式

数据窗口表达式是增强数据窗口对象功能的强有力措施之一。在数据窗口画笔中使用表达式可静态地设置数据窗口对象中的对象及其属性; 而在程序中使用表达式可动态地创建数据窗口对象中的对象、动态地设置数据窗口对象的属性以及数据窗口对象中的对象的属性、动态地访问数据窗口控制中的数据。表达式的应用给开发人员提供了施展才能的空间, 往往可以达到出人意料的效果。

8.1.1 数据窗口表达式的构成

数据窗口表达式是列名、属性名、操作符和数据窗口函数等的集合。它可以使用在数据窗口画笔、报表画表、数据库画笔和事件处理程序中。

这一节将重点介绍在数据窗口画笔、报表画笔和数据库画笔中表达式的应用。下一节将介绍在程序中运用表达式访问数据窗口控制中的数据、设置或改变数据窗口对象本身的属性和数据窗口对象中对象的属性的方法。

8.1.1.1 表达式中操作符的使用

数据窗口支持以下类型的操作符：

- 算术运算符
- 关系运算符
- 逻辑运算符
- 拼接符

下面分别讲述这四种类型的操作符。

8.1.1.1.1 算术运算符（ Arithmetic Operators ）

算数运算符有以下几种：

操作符	含义
+	加
-	减
*	乘
/	除
^	乘方

其中，乘和除的结果是全精度（连小数点和正负号共 16-18 位）的，并且四舍五入。如下表的例子：

表达式	结果
20/3	6.666666666666667
-20/3	-6.666666666666667
truncate(20/3,4)	6.6666

8.1.1.1.2 关系操作符（ Relational Operators ）

关系操作符用于列与常量的比较，其结果为布尔值（ True 或 False ）。数据窗口支持的关系操作符如下表所示：

操作符	含义	举例	描述
=	等于	salary=100	判断“salary”列的值是否等于 100，是则返回 True，否则返回 False
>	大于	salary>100	判断“salary”列的值是否大于 100，是则返回 True，否则返回 False
<	小于	salary<100	判断“salary”列的值是否小于 100，是则返回 True，否则返回 False
<>	不等于	salary<>100	判断“salary”列的值是否不等于 100，是则返回 True，否则返回 False
>=	大于等于	salary>=100	判断“salary”列的值是否大于等于 100，是则返回 True，否则返回 False
<=	小于等于	salary<=100	判断“salary”列的值是否小于等于 100，是则返回 True，否则返回 False
not=	不等于	salary Not=100	判断“salary”列的值是否不等于 100，是则返回 True，否则返回 False
like	与特定示例匹配	name like 'C%'	判断“name”列的值是否以“C”开头，是则返回 True，否则返回 False
not like	like 的取反表示与特定示例不匹配	name not like 'C%'	判断“name”列的值是否不以“C”开头，是则返回 True，否则返回 False
in	在一系列的离散值范围内	salary in(100,200,300)	判断“salary”列的值是否是 100，200 或 300 中的一个，是则返回 True，否则返回 False
not in	in 的取反，表示不在一系列离散值范围内	salary not in(100,200,300)	判断“salary”列的值是否不是 100，200 或 300 中的一个，是则返回 True，否则返回 False

between	在特定范围内(包括边界值)	salary between 100 and 300	判断“salary”列的值是否在 100 至 300 (含 100 和 300) 范围内, 是则返回 True, 否则返回 False
not between	between 的取反, 表示不在特定范围内 (包括边界值)	salary not between 100 and 300	判断“salary”列的值是否不在 100 至 300 (含 100 和 300) 范围内, 是则返回 True, 否则返回 False

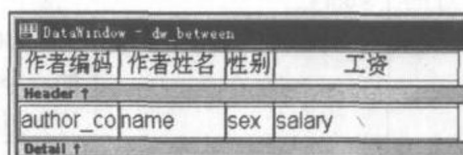
对于上表, 有以下几点说明:

- 说明一: “author_code”和“nationality”为表“auths”中的列名。
- 说明二: LIKE 与 NOT LIKE 操作符。使用 LIKE 或 NOT LIKE 时可用特殊字符去匹配示例中未知的字符(进行模糊匹配), 下表列出了可与 LIKE 或 NOT LIKE 一起使用的特殊匹配符:

特殊字符	含义	举例
%(百分号)	匹配任意一组字符	LIKE GOOD% 匹配以 GOOD 开头的所有字符串
_(下划线)	匹配任意一个字符	LIKE GOOD___ 匹配以 GOOD 开头的所有长度为 7 个字符的字符串

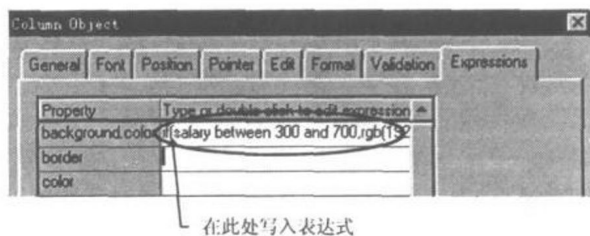
- 说明三: BETWEEN 与 NOT BETWEEN 操作符。用 BETWEEN 检测一个值是否在指定范围内; 用 NOT BETWEEN 检测一个值是否不在指定范围内(包括指定范围的边界值)。

对于下图所示的数据窗口, 下面的这个例子用表达式指定若“salary”列的值在 300 至 700 范围内, 其背景色 (Background.Color) 变为灰色; 否则变为白色。



- 1) 建立如上图所示的数据窗口, 选中要加表达式的“工资 (salary)”列并单击鼠标右键, 在弹出的菜单上选取 Properties... 项 (或直接双击“Salary”列), 弹出

“ Column Object ” 对话框, 如下图所示:

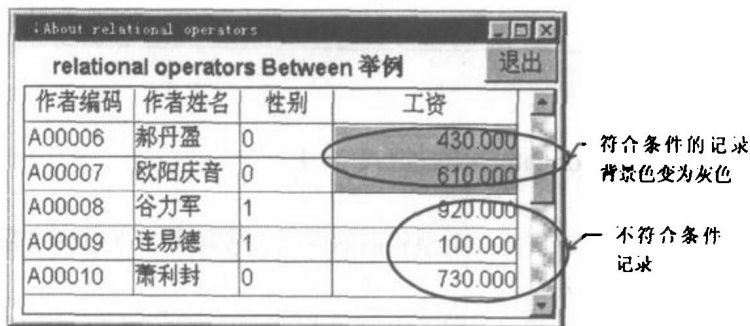


- 2) 在上图所示的对话框中选择  页, 显示 Expressions 对话框, 在 Background. Color 框中输入表达式:

`if(salary between 300 and 700,rgb(255,0,0),rgb(255,255,255))`

表达式的含义为: 若 “ salary ” 列的值在 300 至 700 范围内, 其背景色变为灰色 (rgb(255,0,0)); 否则变为白色 (rgb(255,255,255)) 。

- 3) 运行示例盘中的 Ddws. pb1, 选择 “关于数据窗口表达式” -> “关系运算符 Between 举例” 菜单项, 结果如下:



8.1.1.1.3 逻辑操作符 (Logical Operators)

逻辑操作符如下表所示:

操作符	含义	举例
not	逻辑非 若 “ A ” 为逻辑真 (True), “ NOT A ”	<code>not salary=1000</code> 若 “ salary ” 列的值等于 1000 , “ not salary=1000 ” 的值为假; 若 “ salary ” 列

	为逻辑假（ False ）； 反之“ A ”为逻辑假， “ NOT A ”为逻辑真。	的值不等于 1000 ， “ not salary=1000 ” 的值为真。
and	逻辑与 若“ A ”和“ B ”都为逻辑真，那么“ A AND B ”为逻辑真； 反之若“ A ”和“ B ”都为逻辑假或“ A ”和“ B ”中有一个为逻辑假，那么“ A AND B ”的值为逻辑假。	salary>1000 and sex='1' 若“ salary>10000 ”和“ sex='1' ”的值都为真，则“ salary>1000 and sex='1' ”为真；若“ salary>10000 ” sex='1' ”的值都为假或任意一个为假，则“ salary>1000 and sex='1' ”的值为假。
or	逻辑或 若“ A ”和“ B ”任意一个为真或都为真，那么“ A OR B ”为逻辑真； 反之“ A ”和“ B ”都为假，那么“ A OR B ”为逻辑假。	salary>1000 or sex='1' 若“ salary>10000 ”和“ sex='1' ”的值都为真或任意一个为真，“ salary>1000 or sex='1' ”的值为真；若“ salary>10000 ”和“ sex='1' ”的值都为假，则“ salary>1000 or sex='1' ”的值为假。 （其中“ salary ”和“ sex ”为表“ auths ”中的列名）

8.1.1.1.4 拼接符（ Concatenation oprator ）

拼接符为“+”，它可以将两个类型相同的内容连接起来，可拼接的类型有字符串和大文本（ BLOBS ）。例如：

字符串表达式	拼接后的结果
“over”+“stock”	若有两个字符串“ over ”和“ stock ”，那么拼接后的结果为“ overstock ”。
author_code+','+nationality	若列“ author_code ”的值为“王达琳”，列“ nationality ”的值为“汉”，则拼接后的结果为“王达琳,汉”（其中“ author_code ”和“ nationality ”为表“ auths ”中的列名）。

8.1.1.1.5 操作符的优先级

下表从前到后列出了操作符的优先级（若优先级相同按从左到右的优先级计算）：

操作符	用途
()	成组
+, -	正负号（一元加减）
^	乘方
*, /	乘, 除
+, -	加, 减, 字符连接
=, >, <, <=, >=, <>	关系操作符
not	逻辑非
and	逻辑与
or	逻辑或

8.1.1.2 数据窗口画笔函数

数据窗口画笔函数是 PowerBuilder 的 DataWindow Painter 提供的一系列内嵌函数，共有 100 余个，每个函数完成某种特定的功能。大部分数据窗口画笔函数与 PowerBuilder 提供的内嵌函数(400 余个)通用。它们的不同之处是，前者用在数据窗口画笔中或报表画笔中，而后者一般用在 PowerScript 语言中。这些函数根据用途可分为以下几种类型：

函数类型	用途
统计函数	用于统计
横向列表函数	得到有关横向列表的信息
数据类型检测和转换函数	改变数据的类型或格式
数据窗口信息函数	得到有关数据窗口信息
日期和时间函数	得到有关日期的信息
数字函数	对数字进行操纵
字符串函数	对字符串进行操纵
其它函数	其它工作

关于数据窗口画笔各类函数的详细内容，请参见函数手册。

8.1.2 表达式的使用范围

表达式可使用在数据窗口画笔、报表画笔和数据库画笔中。在不同画笔中的作用可概括为下表：

画笔	使用表达式的范围
数据窗口画笔和报表画笔	计算域 属性值的条件表达式 校验规则（ Validation rules ） 过滤器（ Filters ） 排序（ Sorting ） 统计图中的序列值和列值（ Series and values in graphs ） 横向制表中的列,行和单元格的值 （ Columns,rows and values in crosstabs ）
数据库画笔	校验规则（ Validation rules ）

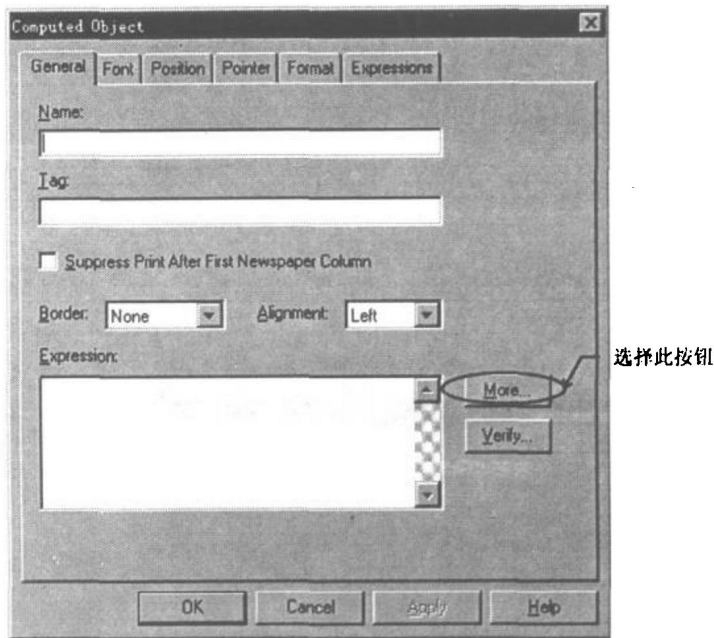
8.1.2.1 在数据窗口和报表画笔中使用表达式

8.1.2.1.1 计算域中的表达式

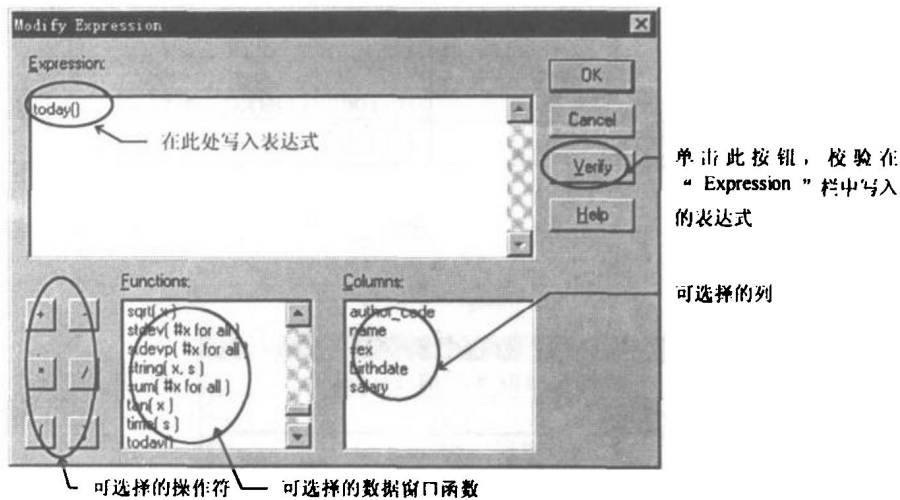
在计算域中使用表达式丰富了数据窗口的应用，例如可进行汇总统计、基于列的计算、显示系统信息等等。在计算域中，表达式可以赋任意值，表达式的数据类型就是计算域的数据类型。

下面讲述在计算域中加表达式的方法：

- 1) 选择数据窗口工作区 PainterBar 工具栏的  下拉列表中的  图标，然后在数据窗口工作区要加入计算域的带区域内单击鼠标左键，弹出“Computed Object”对话框，如下图所示：



- 2) 在“Computed Object”对话框中点取  按钮，弹出“Modify Expression”对话框，如下图所示：

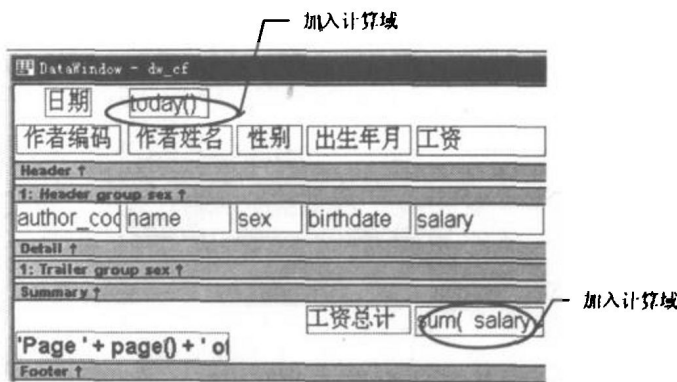


- 3) 在“Modify Expression”对话框中的“Expression”栏中编辑计算域表达式，编辑完后点取上图的 OK 按钮，返回到“Computed Object”对话框，定义的表达式显示在此对话框的“Expression”栏中，最后点取此对话框中的 OK 按钮，计算域中表达式的定义完成。

例一：

下面这个例子要在已建立的数据窗口中加入两个计算域：当前日期（today()）和工资总计（sum(salary for all)），分别用来显示当前日期和统计工资总和。步骤如下：

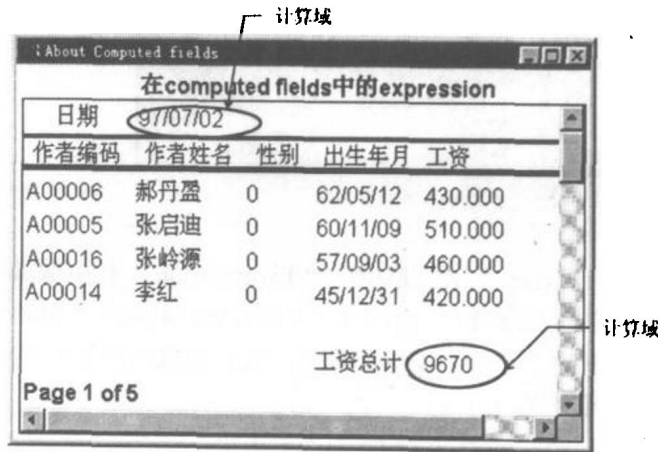
1) 建立如下图所示的数据窗口 dw_cf：



2) 加入两个计算域 today()、sum(salary for all)，如下表所示：

表达式	描述
Today()	用 Today() 函数显示日期
Sum(Salary for all)	用 Sum() 函数计算工资总额

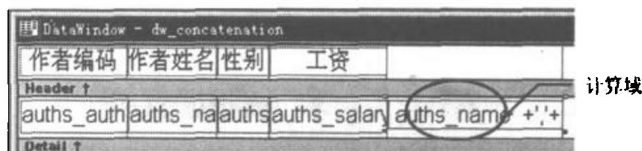
3) 运行示例盘中的 Ddws. pb1，选择“关于数据窗口表达式”->“计算域中表达式的应用”菜单项，结果如下：



例二:

此例是在已存在的数据窗口中加入一计算域, 用来将 “ auths_name ” 列和 “ nations_nationality ” 列拼接起来。

- 1) 在下图所示的数据窗口中, 加入一计算域, 其表达式为 “ auths_name + ',' + nations_nationality ”, 如下图:



- 2) 运行示例盘中的 Ddws. pbl, 选择 “关于数据窗口表达式” -> “连接符的使用” 菜单项, 结果如下:



8.1.2.1.2 属性值的条件表达式 (Conditional expressions for property value)

属性值的条件表达式的应用可以增强数据窗口的功能。

属性值的条件表达式是布尔(boolean)值, 即其值为 TRUE 或 FALSE。通过给数据窗口中的列加入属性值的条件表达式可以设置此列所具有的属性。方法如下:

- 1) 在数据窗口工作区中选中要设置属性的列, 单击鼠标右键, 在弹出的菜单中选 “ Properties ” 项, 弹出 “ Column Object ” 对话框;
- 2) 选择此对话框的 “ Expression ” 页 (此列所具有的属性页), 在要设置的属性框中

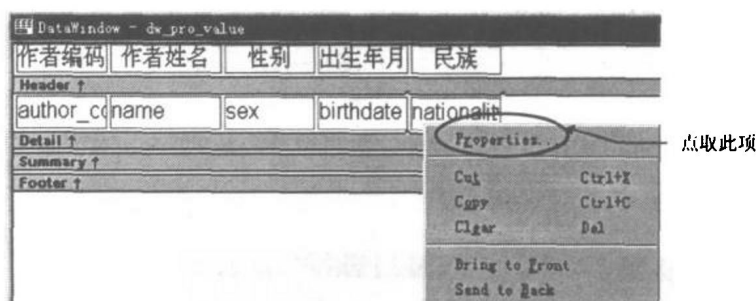
写入表达式;

- 3) 然后点取“Column Object”对话框的 O K 按钮, 属性值的条件表达式定义完毕。
PowerBuilder 将按照表达式设置列的属性。

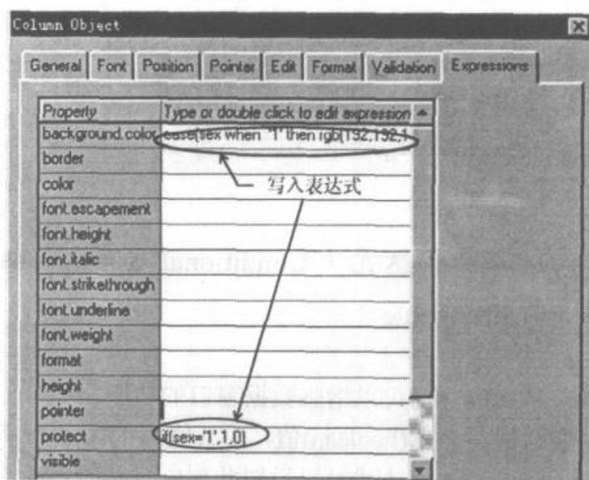
例一:

此例通过对属性“protect”的设置来实现数据窗口上按某种条件使某儿行的数据得到保护, 不允许修改。步骤如下:

- 1) 建立如下图所示的数据窗口 dw_pro_value :



- 2) 选中要设置属性的“nationality (民族)”列, 按下鼠标右键弹出菜单, 从中选择“Properties”项, “Column Object”对话框打开, 如下图所示:



- 3) 选取上图的  页, 列的属性设置对话框弹出 (如上图), 在“Protect”属性框中写入下面的表达式:

if(sex='1',1,0)

表达式的含义为：当性别为'1'(男)时，列“民族”被保护，不允许修改（即光标不能移动到性别为 '1'（男）时的对应的“民族”列中）。

为使不能修改的列明显的被表现出来，我们使其背景色为灰色，在“ Backgroundcolor ”属性框中写入表达式：

case(sex when '1' then rgb(192,192,192))

此表达式的含义为性别为'1'(男)时，列“民族”的背景色为灰色；

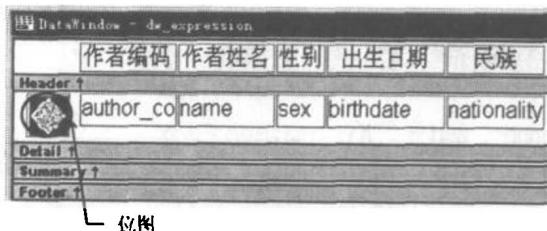
- 4) 运行示例盘中的 Ddws.pbl，选取“关于数据窗口表达式”->“属性值的条件表达式”菜单项，结果如下：



例二：

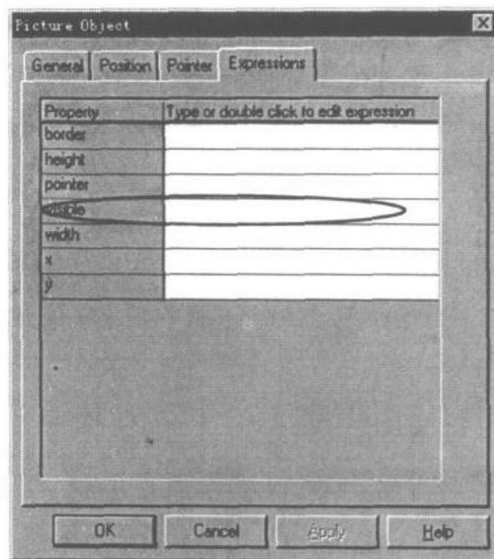
下面的这个例子是将数据窗口中满足某一条件的记录旁边显示一位图。步骤如下：

- 1) 打开要增加条件位图的数据窗口，在细日带中放置一图象(Picture)控制，并为这个位图指定一位图文件(a.bmp)，这时数据窗口如下图所示：



- 2) 选中图象控制，按下鼠标右键，弹出一菜单，从中选择“ Property ”选项，对话框

“ Picture Object ” 打开，如下图所示：

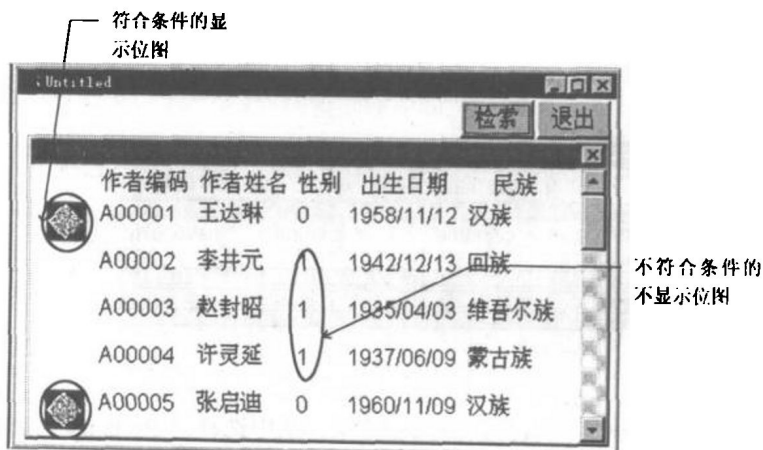


- 3) 选择上图中的 “ Expression ” 页，图象控制的属性列出。在上图标出的 “ Visible ” 属性框中输入以下表达式：

`if(sex='0',1,0)`

表达式的含义为：性别为 “ 0 ” （女）时，显示图象，否则不显示。

- 4) 运行示例盘中的 Ddws.pbl ， 选取 “关于数据窗口表达式” -> “在属性条件表达式中加位图” 菜单项，结果如下：



8.1.2.1.3 校验规则 (Validation rules for table columns)

中的表达式

表达式运用在校验规则中可使输入的数据进一步规范化。

校验规则中的表达式是布尔(boolean)表达式，用带有关系操作符和逻辑操作符的表达式与对应列进行比较，当校验规则中表达式的值为 TRUE 时，列中输入的数据合法；若为 FALSE，列中输入的数据非法。

可以在数据库画笔或数据窗口画笔中使用校验规则表达式。但校验规则表达式在两种画笔中使用的形式是不同的，如下表所示：

在数据库画笔中的表达式	在数据窗口画笔中的表达式	描述
@column>0	Long(gettext())>0 (在每一数据类型为数字型的列的“ Validation ”中加入此表达式)	校验表中数据类型为数字型的所有列的值是否大于零，若列的值不大于零，将有系统错误信息提示。
@salary>0	Long(gettext())>0 (在“ salary ”列的“ Validation ”中加入此表达式)	若输入的“ salary ”列的值不大于零时，将有一错误信息提示。 其中“ salary ”为表“ auths ”中的列。

既然在数据库画笔和数据窗口画笔中都可为列定义校验规则表达式，那么如果在数据库画笔和数据窗口画笔中都定义了校验规则表达式，究竟以哪一个校验规则为准呢？

如果在数据库画笔中为列定义了校验规则表达式，则 PowerBuilder 自动将数据库画笔中定义的校验规则表达式带入数据窗口中，并转换成数据窗口中的校验规则表达式形式。如果开发人员随后又修改了数据窗口中的校验规则表达式，则数据库中的校验规则表达式与数据窗口中的校验规则表达式就会不一致。在数据库画笔中操纵数据将会以数据库中的校验规则表达式为准，在数据窗口中操纵数据将会以数据窗口中的校验规则表达式为准。

如果在数据库画笔中未定义列的校验规则表达式，而在数据窗口画笔中做了定义，那么将以数据窗口画笔中定义的校验规则为准。在数据窗口中定义的校验规则仅在数据

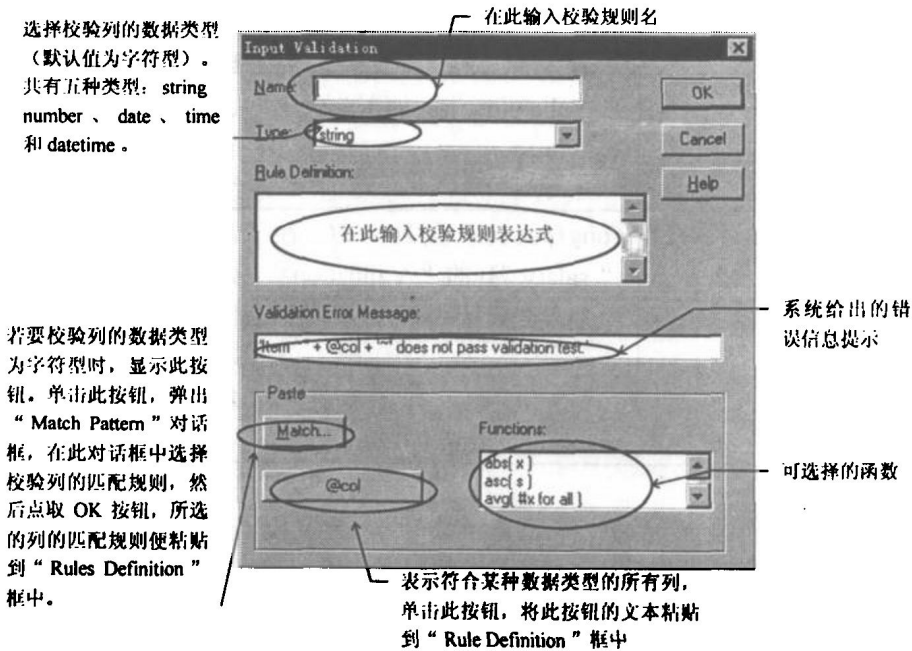
窗口环境下起作用。

此外，若要保持两种画笔中校验规则表达式的一致性，可使用 Powersoft 提供的 Advanced Developer's ToolKit 工具包来完成。

在数据库画笔中可以为列定义校验规则表达式用以校验此列输入的数据。定义方法有两种：

1. 为表中同一数据类型的所有列定义校验规则表达式

- 1) 在 PowerBar 工具栏中点取  画笔，打开数据库画笔工作区；
- 2) 在数据库画笔工作区中，打开对应表后，选择数据库画笔工作区 Design -> Validation Maintenance 菜单项，弹出“Validation Rules”对话框；
- 3) 在“Validation Rules”对话框中点取 NEW 按钮，弹出“Input Validation”对话框，如下图所示：



- 4) 在此对话框中输入校验规则表达式，然后点取 OK 按钮，则为表中同一数据类型的所有列的校验规则表达式定义完毕。

2. 为表中指定列定义校验规则表达式