

# Rational ClearQuest 使用说明书

写在前面的话:

对于测试人员来说,缺陷的提交和处理是日常最常见的工作。但是手工填写缺陷表格,对于缺陷的整理和统计很不方便,在这种情况下,就出现了各种缺陷管理工具。Rational 的 ClearQuest 就是其中比较强大的一种缺陷跟踪工具。ClearQuest 类似一个开发平台,可以很方便的配置各种需要的选项。

我这里单纯的把 CQ 用于缺陷管理,其实有些浪费了。在 Rational 的套件中, ClearQuest 应该是变更管理用的,缺陷应该从测试用例发现,而不是独立提交。我这里把软件开发过程割裂了开来, CQ 只当成单纯的缺陷提交管理工具。

下面的文档虽然是关于 CQ 的缺陷管理配置的,但是如果想把它用于其他的方面,也应该有些借鉴意义吧。

这篇东西可以说是翻译 CQ 的帮助,同时也是一些经验的总结,测试流程使用的是类似北航的 QAMonitor。(另外说些别的:Qamonitor 是个好东西,使用方便及了,但是他的问题有几种:1.使用的是 Excel97,很容易损坏,即使可以修复也会丢失记录,所以一定要每天做备份。2.使用过程中服务器端必须完全共享测试文件,可能引起病毒等问题。3.必须安装客户端,现在的趋势应该是使用 Web 方式的。4.客户端和其他的程序可能会有冲突,至少 QAMonitor 和 Rational Robot 的一个 dll 会冲突。5.无法自定义一些选项,流程过于僵化。虽然有这么多的问题,但在小型开发过程中, QAMonitor 绝对可以满足你的缺陷管理需求。)

我这里介绍的是 Rational 的 2002.05.10 版,最新的 IBM Rational 2003 版我没有看到,所以不知道和此版有多少的差别。

这篇东西写很久了,一年多前就写成了,其中也修改了几次,但可能还是会有问题。建议大家如果可能就买正版,没有技术支持使用这些很不方便。

说真的,这篇本身我并不想贴出,几个月前我发了《ClearQuest Web 端使用指南》,一点回应都没有,使我很怀疑多少人对此感兴趣。不过上个月一位朋友给我发了邮件,询问关于此篇的消息,哪怕只有一个人关心,为了这么一个人,我也要发出来给他看。

一、前言. 1

二、附属安装. 2

1.支持的环境. 2

2. 数据库. 2

三、Rational ClearQuest 配置. 4

1. Rational ClearQuest Maintenance Tool 设置. 4

2. Rational ClearQuest Designer 设置. 7

四、Rational ClearQuest Web 服务器配置. 25

1. 设置注册表存取权限. 26

2. NTFS 格式机器的读取权限设置. 27

3. 创建 Web 虚拟目录. 28

4. 编辑虚拟目录的属性. 30

5. 编辑 cache 和 images 目录的属性. 33

6. 其他设置. 34

7. IE 设置. 35

五、Rational ClearQuest 客户端配置. 37

1. 提交缺陷. 38

2. 修改记录. 40

3. 控制台操作. 41

六、附录. 50

1. 脚本片段. 50

2. 补充说明. 52

一、前言

Rational ClearQuest 功能十分的强大,可以和 Rational 的其他的的产品结合,比如 Rational ClearCase、Rational Rose 等。ClearQuest 主要用于变更管理和缺陷跟踪。这个文档主要是说明 ClearQuest 用于缺陷跟踪的时候,如何进行相应的配置工作。

在文档中,我尽量使用中文,但是一些英文单词的翻译可能不是很恰当。不过在翻译的过程中,我都附上了相应的中文或英文,应该对理解不造成障碍。其中 ClearQuest 简写为 CQ。

在文档中,标记为红颜色的都是值得特别注意的地方。

下面是部分中英文和缩写对照表:

CQ:ClearQuest。

计划: Schema

自定义: Custom

维护表: Schema Repository。

版本: Version

注释: comment

延迟: Timeout

间隔: Interval

字段: Fields

缺陷: Defect

工作台: Workspace

状态转移矩阵: State Transition Matrix

## 二、附属安装

### 1. 支持的环境

Rational ClearQuest 可以安装在 WinNT 4.0, Windows2000, Windows98, WindowsMe, WindowsXP 上。ClearQuest 的安装十分的简单, 点击 setup.exe 文件, 选择 Rational ClearQuest, 一路选择“下一步”就可以了。

不过要注意, 如果想配置 ClearQuest 的 Web 服务器, 在安装 CQ 的时候, 必须选择 Custom (自定义) 安装, 在 Choose Features 中, 把 Web Server Components 选中。

CQ 分为服务器端、客户端、Web 端三个部分。

在安装 Rational ClearQuest 的时候, 实际就是安装了 ClearQuest 的服务器端和客户端。

1. 服务器端: 通常情况下, 创建 (Create) 维护表 (Schema Repository) 的机器可以称之为服务器端。

服务器端创建的维护表需要连接数据库, 数据库可以选择 SQL Server、Oracle、Access2000 等 (支持其他类型的数据库, 但是我想应该很少用)。

数据库不一定非要安装在服务器端, 也可以使用专门的数据库服务器。

服务器端主要是创建维护表 (Schema Repository)。

如果想使用 Web 端功能, 在服务器端, 必须进行相应的配置工作 (Web 服务器只支持 WinNT4.0 和 Win2000Server)。

通常情况下, 服务器端也是缺陷记录数据库服务器。

2. 客户端: 客户端也需要安装 Rational ClearQuest。客户端是指连接服务器端创建的维护表, 使用 ClearQuest 工具提交缺陷的机器。

3. Web 端: Web 端不需要安装 ClearQuest, 只需要使用浏览器登陆服务器建立的相关网页。Web 端是只使用浏览器 (IE 或 Netscape) 提交缺陷的机器。但是注意, 如果想使用 Web 端, 必须配置 Web 服务器。

安装 CQ 大概需要 500M 的硬盘空间。如果在 CQ 服务器上配置了 Web 服务器, 可以使用 IE 进行缺陷跟踪。IE 客户端版本需要 5.01 以上。

### 2. 数据库

ClearQuest 支持 Microsoft SQL Server 7.0、Microsoft SQL Server2000、Oracle(8.0.5, 8.1.6, 8.1.7)、Access2000 等数据库, 这里主要介绍 SQL2000 和 Access2000 的配置方式。

#### 2.1 使用 Access 数据库

在使用 Access 数据库的时候, 一定要把存放 Access 文件的目录共享。比如 test 机器当服务器, 在建立 Access 文件的时候, 把 test.mdb 文件放到了“c:\共享”目录下, 其他机器在连接使用这个 Access 的时候, 使用“\\test\共享\test.mdb”连接此数据库文件。

不建议在网络环境中使用 Access 数据库。

#### 2.2 使用 Microsoft SQL Server 数据库

在 CQ 中使用 SQL 数据库, 必须在使用前建立数据库, 再在 CQ 中连接此数据库。下面就是说明建立数据库的方法。

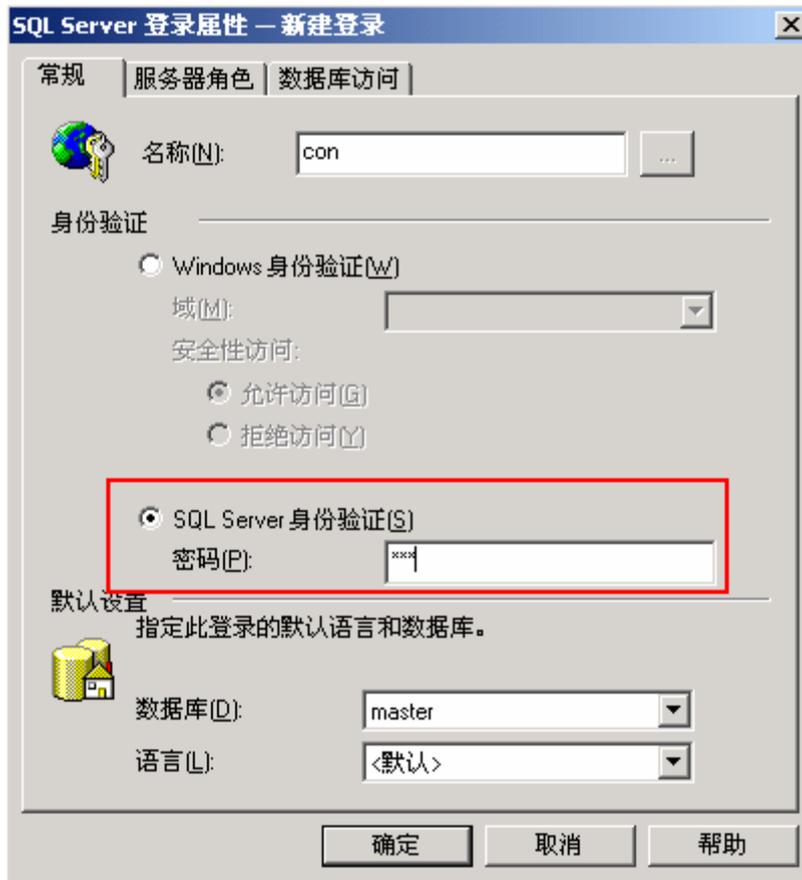
在安装 SQL 的时候, 选择混合模式进行安装。

在 ClearQuest 中使用 Microsoft SQL 数据库, 不允许使用 sa 进行口令验证, 所以必须建立新用户进行数据库的登陆验证。

举例: 在 Microsoft SQL Server 企业管理器中, 建立新数据库 connect (数据库名称首位必须用英文, 不允许在第一位使用数字) 给 ClearQuest Maintenance Tool 使用。

1. 建立数据库: 打开“SQL Server 组->机器名 (或 IP 地址)->数据库”, 鼠标右键单击“数据库”, 在出现的菜单中选择“新建数据库”, 填写数据库名称 connect, 点击“确定”保存建立的数据库。

2. 配置数据库所有者: 在 CQ 中连接数据库的时候, 不支持使用 sa 或 Administrator 用户名和密码, 必须建立使用新的数据库所有者。这里设置 connect 数据库用户所有者为用户名 con, 密码 con 的用户。打开“SQL Server 组->机器名 (或 IP 地址)->安全性->登陆”, 点击鼠标右键, 出现的菜单中选择“新建登陆”, 出现如下图界面, 在名称中添入 con, 选择“SQL Server 身份验证”, 在密码中录入 con (密码可以为空, 但是必须选择“SQL Server 身份验证”选项)。



选择“数据库登陆”页，如下图所示，在许可中，选择所属的 connect 数据库；在数据库角色中，选中“db\_owner”，点击“确定”，输入确认密码，此 connect 数据库就建立完毕了。



3. 照上面说的方法，再建立一个名为 test 的数据库给 ClearQuest Designer 使用，用户名和密码是 test。

说明：一个用户可以有多个所属数据库，在上图的“数据库访问”页中选中相应的数据库许可就可以了，不要忘了选中“db\_owner”角色。

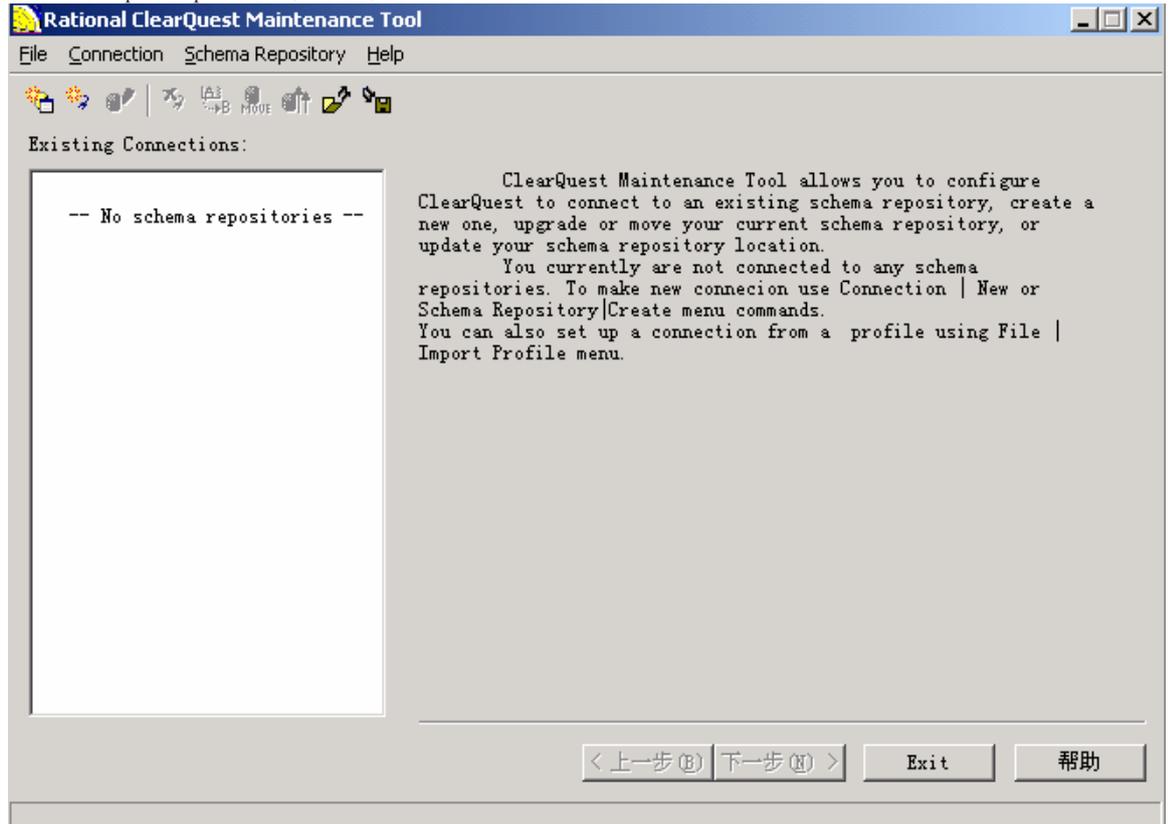
下面所有使用的数据库，都是 SQL2000 数据库，其他的数据库类型也是类似的原理创建和连接。

### 三、Rational ClearQuest 配置

在 CQ 使用过程中，首先要创建或连接 ClearQuest 维护库，再到 Designer 中配置 CQ 使用环境，最后使用 CQ 客户端或 Web 端进行缺陷的提交和管理。

#### 1. Rational ClearQuest Maintenance Tool 设置

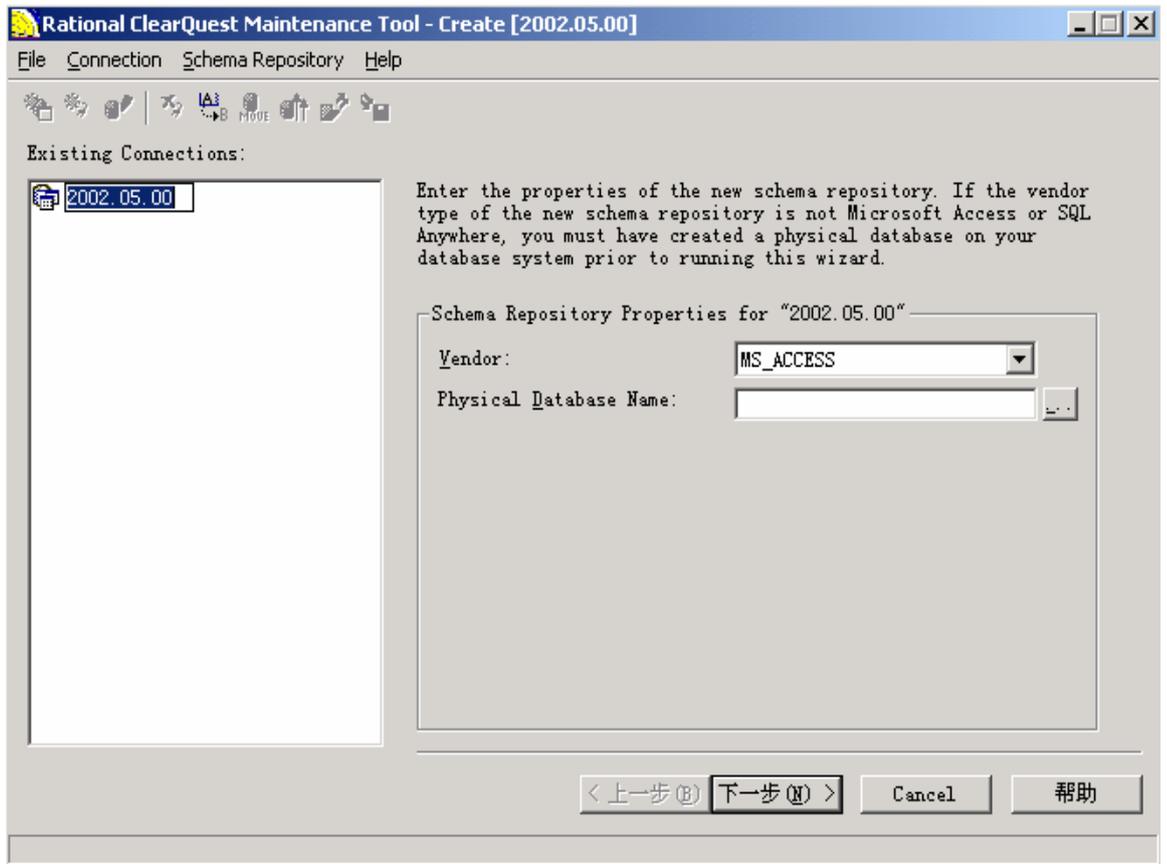
在使用 ClearQuest 前，必须设置维护表。从开始菜单中打开 Rational ClearQuest Maintenance Tool（安装目录中 cqdbsetup.exe），如下图所示：



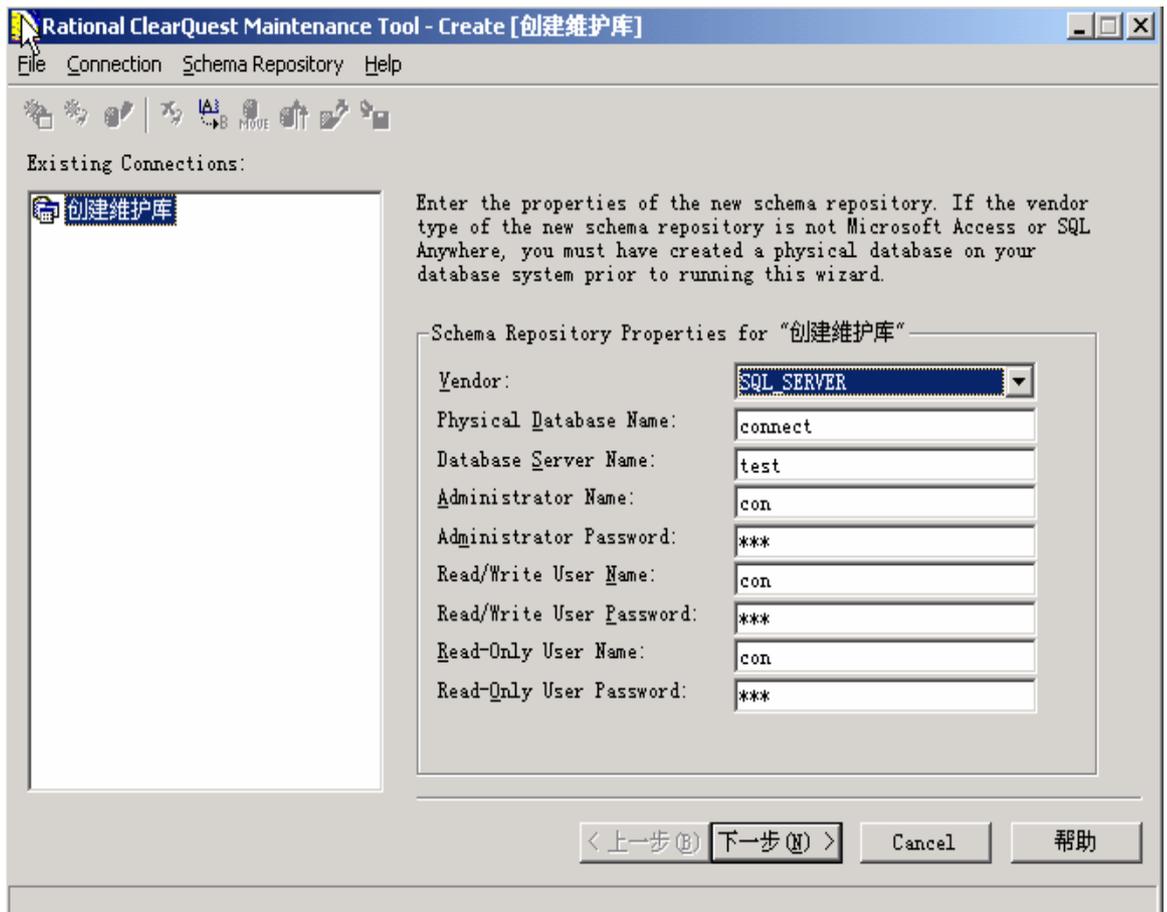
如果此机器作为服务器使用，那么就创建维护库；如果此机器做为 CQ 客户端使用，那么就连接维护库。

#### 1. 1 创建维护库

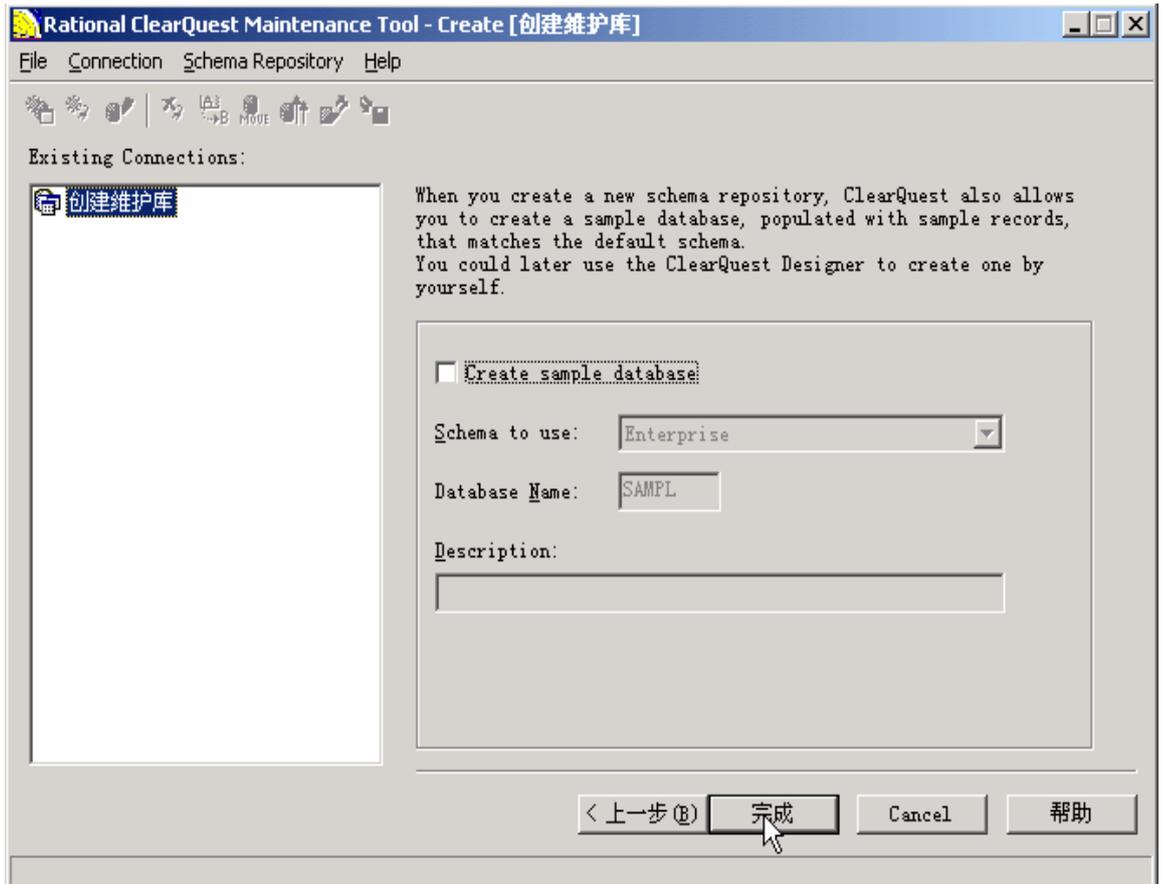
1. 点击菜单中“Schema Repository->Create”，或快捷菜单栏中第一个图标 Create，出现下图界面：



2. 在 Existing Connections 中录入可以识别的名称。Vendor 选择 SQL\_SERVER; Physical Database Name 填写创建的数据库名称 connect; Database Server Name 填写数据库服务器名称或服务器 IP 地址; 剩下的项目, 填写 connect 数据库所属用户名和密码, 这里都是 con, 填写完毕后, 点击“下一步”。



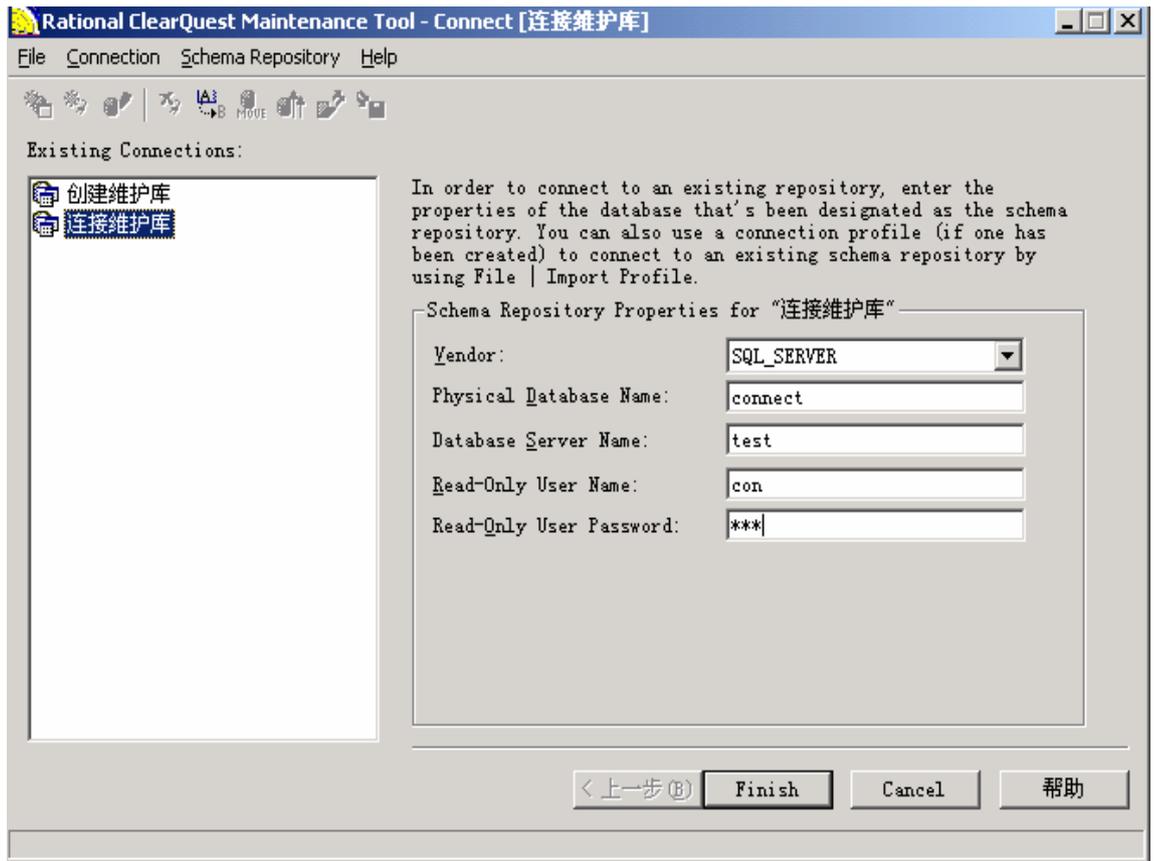
3. 在第二页中，不选择“Create sample database”，点击“完成”，就完成了维护库的创建。



## 1. 2 连接维护库

如果想使用 CQ 的客户端，必须在 Maintenance Tool 中连接服务器创建的维护库。下面讲如何去连接服务器的维护库。

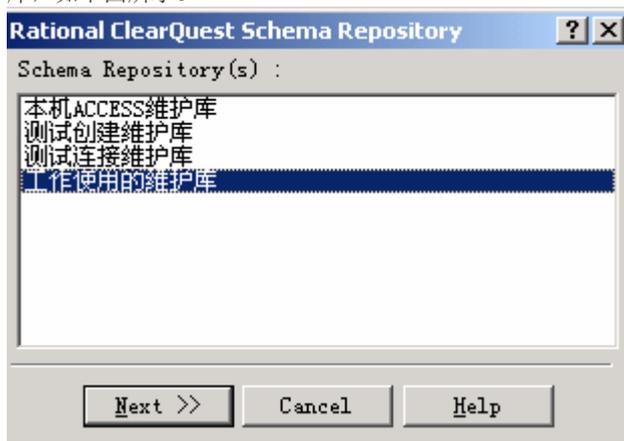
1. 点击菜单中“Connection->New”或快捷菜单栏中第二个图标 New Connction。
2. 在 Existing Connections 中录入可以识别的名称；Vendor 选择 SQL\_SERVER；Physical Database Name 填写连接的数据库名称 connect；Database Server Name 填写数据库服务器名称或服务器 IP 地址；Read-Only User Name 填写 connect 数据库所属用户名 con；Read-Only User Password 填写密码 con。填写完毕后，点击“Finish”完成维护连接。



### 1. 3 其他功能

在 Maintenance Tool 中，除了对于维护库的创建或连接外，也可以编辑、删除、修改、更新、升级维护库，在菜单中都有相应的操作，这里就不多介绍了。

一台机器，可以创建多个维护库（对应不同的数据库）；与之相应的，一台机器，也可以同时连接多个机器的维护库。在使用 CQ 客户端或 Designer 的时候，可以选择使用哪个机器的维护库，如下图所示。

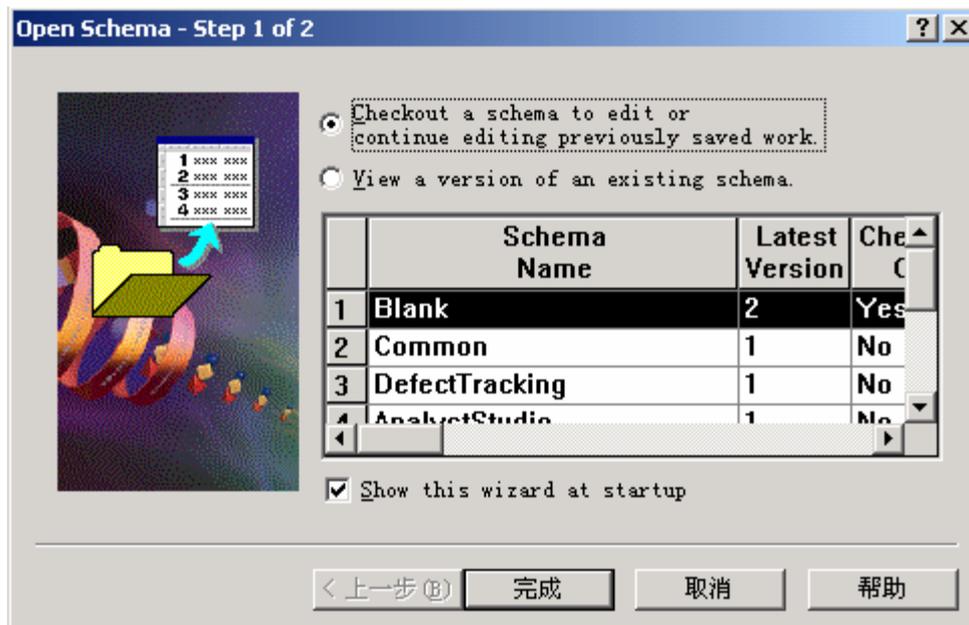


## 2. Rational ClearQuest Designer 设置

CQ Designer 主要进行缺陷提交页面和选项的配置，也是 CQ 中最重要的部分。这里的功能和配置很多，可以使用 Basic 或 Perl 进行编程设置。这里我也只研究明白了一小部分，欢迎大家以后能多多的研究补充。

### 2. 1 进入 CQ Designer

从菜单中打开 Rational ClearQuest Designer（安装目录中的 cqdesign.exe），出现的界面中，选择“测试连接维护库”进入刚才设定的维护库。在登陆界面中，User Name 输入 admin（缺省的超级管理员），Password 为空（可以参照 2.4 修改密码），点击“ok”进入 Designer（如下图）。



如果要修改已有的计划（Schema），可以选择需要修改的计划名称（Schema Name），点击“完成”进入修改页面。

我们这里要创建新的计划，所以点击“取消”关闭界面。

## 2. 2 建立新计划

在使用过程中，新的计划必须从现有的计划中继承，原计划中的各种配置新计划都可以继承下来。建议配置一个标准的计划，其他的计划都从此计划中继承。

每次修改计划，都需要把计划 check out，修改后，保存计划，再把计划 check in，计划的 Version（版本）加 1。

下面是计划的建立步骤：

1. 从菜单中选择“File-> New Schema”
2. 在 schema list 窗体中选择 DefectTracking, version 1，然后选择“下一步”。
3. 在 Schema Name 填写 test，comment（注释）中填写“测试计划”，点击“完成”
4. 当问你是否创建 database，点“否”。（也可以点击“是”，按照下面建立的数据库方式进行）
5. 当问你是否 check out database，点“否”。
6. Test 计划建立完毕。

## 2. 3 建立数据库

2.2 是计划建立的过程，但是此计划并没有数据库的支持，本小节就说明如何去建立数据库并和计划相连。

1. 菜单中选择“Database->New Database”，打开新建 Database 对话框。
2. Logical Database Name 添写 test (数据库名称 1-4 个字节长)，在 Comment（备注）中填写“测试数据库”，点击“下一步”。
3. Vendor 选择 SQL\_SERVER；Physical Database Name 填写数据库名称 test；Database Server Name 填写数据库服务器名称或服务器 IP 地址；剩下的项目，填写 test 数据库所属用户名和密码，这里都是 test。填写完毕后，选择“Production Database”，点击“下一步”。
4. 把 Timeout（延迟）和 Poll Interval（间隔）都设置为 0. 点击“下一步”。
5. 从计划列表中选择数据库需要连接的计划，这里是 test 计划，点击“完成”。
6. 显示“Database was created successfully”。
7. 询问是否 check out，选择“否”。

数据库连接计划完毕。

## 2. 4 添加用户和组

在使用 CQ 的过程中，用户和组是很关键的两个因素。下面介绍如何去添加用户和组。

在 CQ Designer 中，从菜单中选择“Tools->User Administration”，出现 User Administration 界面。

在操作完毕后，必须点击右侧 OK 按钮进行保存，否则所有改动无效。

#### 2. 4. 1 添加组

在用户管理（User Administration）界面中，点击右侧按钮 Group Action，在出现的菜单中选择 Add Group，出现 Add Group 界面。

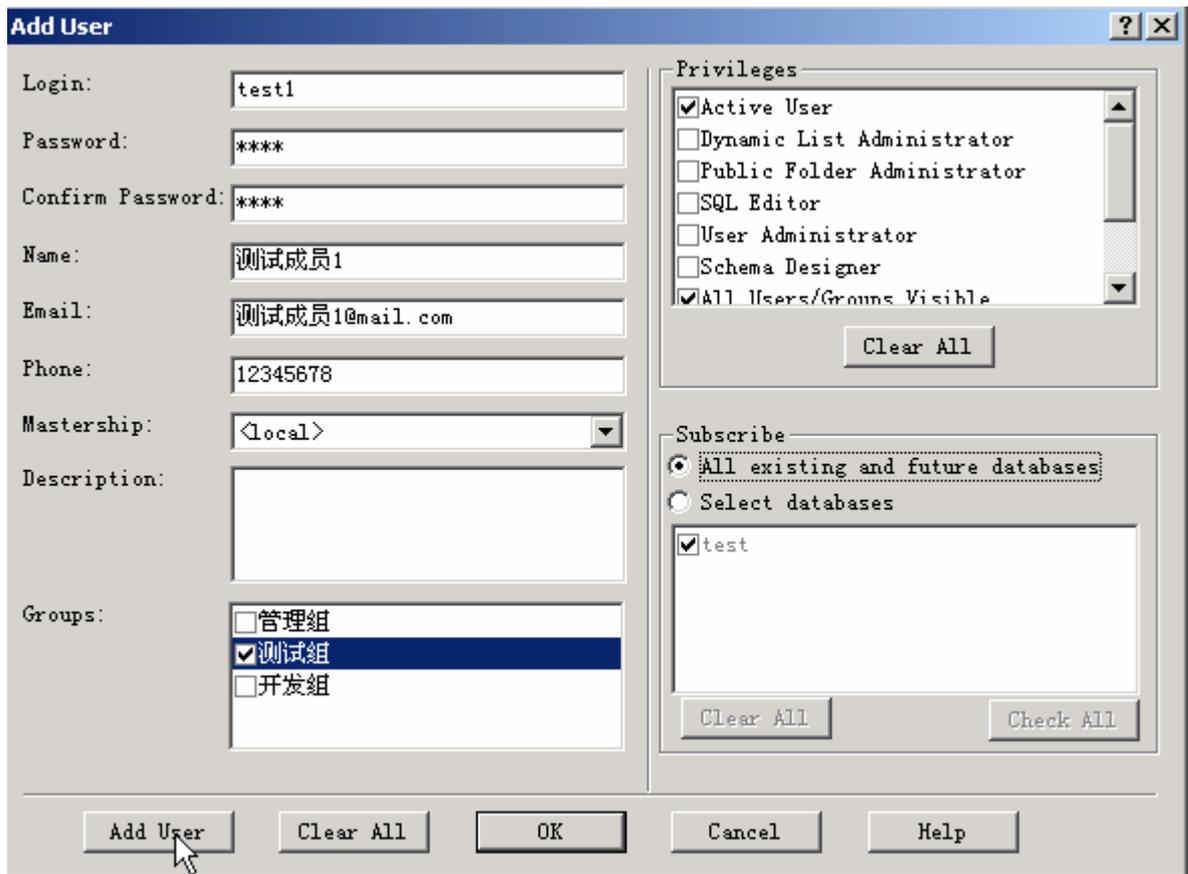
在 name 中填写需要建立的组名（建立“管理组”）；Active 选择是否新建组当前可用；MembershipGroups 中选择新建组的子组；MembershipUsers 中选择新建组包含的用户；Subscribe 中，选择新建组的数据库权限（选择“All existing and future databases”可以有任意数据库的权限，选择“Select databases”可以选择具有特定数据库的操作权限）。

如果需要添加新用户，点击下面的按钮“New User”（参照 2.4.2）；如果需要添加别的组，点击“Apply”按钮保存现在的组并添加新的组；全部组添加完毕后，点击“OK”结束新组的添加。

在这里，除了管理组，还按照上面的方法建立“测试组”和“开发组”。

#### 2. 4. 2 添加用户

在用户管理（User Administration）界面中，点击右侧按钮 User Action，在出现的菜单中选择 Add User，出现 Add User 界面。



在 Login 中添加登陆用户名 test1；按需要添加其他的内容（只有 Login 是必添项）；Groups 中，选择用户属于哪个组（这里我们选择测试组）；Privileges 中选择赋予此用户的权限，通常缺省权限就可以了；Subscribe 中选择此用户具有哪些数据库的操作权限。

如果添加多个用户，点击 Add User，如果全部用户填写完毕，点击 OK。

按照上面的方法，新建测试组用户 test2，开发组用户 pg1、pg2，管理组用户 manage1、manage2 备用。

注意：一个用户可以属于多个组。

#### 2. 4. 3 其他功能

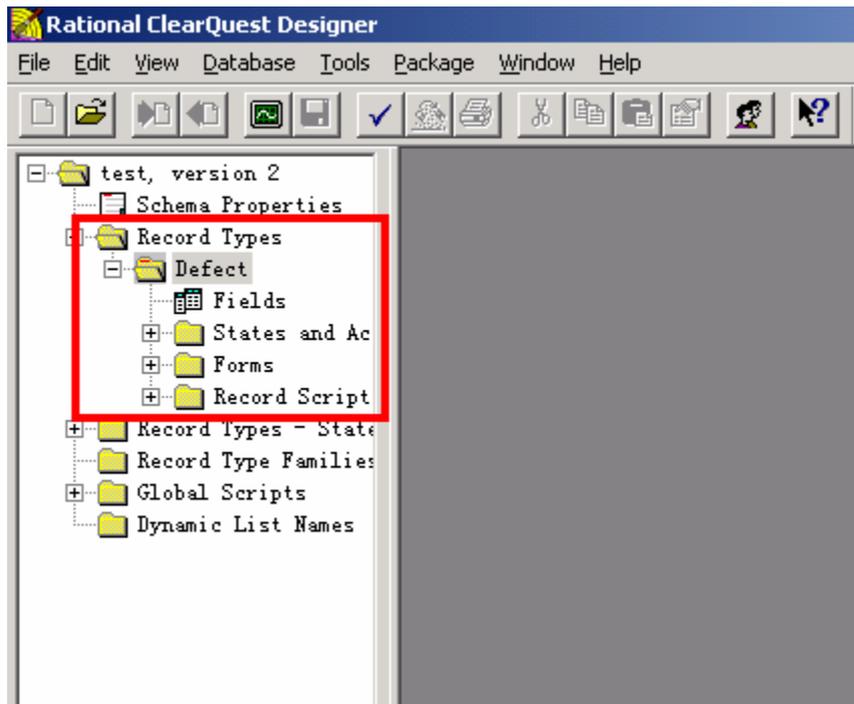
更新数据库：添加用户或组，以及修改了用户或组的数据库权限后，必须更新数据库。点击右侧按钮“DB Action”，在出现的菜单中选择 Upgrade，出现 Upgrade 界面，选择需要升级的数据库，点击 OK 更新数据库。

用户和组的修改等功能，比较简单，都可以从右侧按钮中选择，就不介绍了。

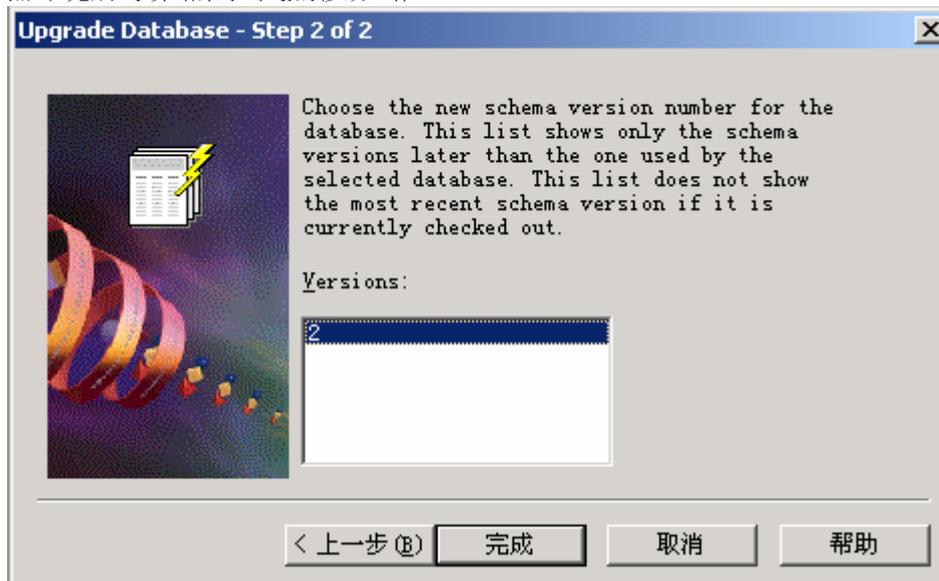
#### 2. 5 计划设计

打开计划：在 Rational ClearQuest Designer 界面中，选择菜单“File->Open Schema”，选择需要设定的计划 test，点击“下一步”，在 Comment 中添加本次操作的注释（可以不填写），点击“完成”打开 test 计划。

我们现在所设计的计划是给缺陷（Defect）管理使用，所以计划中主要设定部分都在 Record Types->Defect 中，下面所说的各种配置，都在 Defect 中进行。



计划配置完成后，需要从菜单中选择 File->Save Work 保存计划，并且 File->Check In。  
计划修改后，必须从菜单中选择 Database->Upgrade Database，出现的提示中选择“是”，在出现的数据库列表中，选中修改的计划所对应的数据库，点击“下一步”，出现的界面（如下图）中选择需要更新的版本，点击“完成”才算结束了计划的修改工作。



## 2. 5. 1 字段 (Fields) 设置

点击左侧树中 Defect 下 Fields，在右侧显示出字段表。

|  | Field Name        | Type             | Default Value | Permission | Value Change |
|--|-------------------|------------------|---------------|------------|--------------|
|  | State             | STATE            |               |            |              |
|  | version           | INT              |               |            |              |
|  | lock_version      | INT              |               |            |              |
|  | locked_by         | INT              |               |            |              |
|  | history           | JOURNAL          |               |            |              |
|  | is_duplicate      | INT              |               |            |              |
|  | unduplicate_state | SHORT_STRING     |               |            |              |
|  | Headline          | SHORT_STRING     |               |            |              |
|  | Description       | MULTILINE_STRING |               |            |              |
|  | Priority          | SHORT_STRING     |               |            |              |
|  | Severity          | SHORT_STRING     |               |            |              |
|  | Submitter         | REFERENCE        | BASIC,PERL    |            |              |
|  | Submit_Date       | DATE_TIME        | BASIC,PERL    |            |              |
|  | Owner             | REFERENCE        |               |            |              |
|  | old_id            | SHORT_STRING     |               |            |              |
|  | Keywords          | MULTILINE_STRING |               |            |              |
|  | Symptoms          | MULTILINE_STRING |               |            |              |
|  | Note_Entry        | MULTILINE_STRING |               |            | BASIC,PERL   |
|  | Notes_Log         | MULTILINE_STRING |               |            |              |

字段表中，灰颜色的部分是系统自动设置的不可修改字段；黑颜色的部分是可以修改的字段。

Type 列后面的内容都是可以进行编程控制的。

字段表中部分英文对应的常用汉语解释：

id: 编号。系统自动给每个缺陷一个编号，规则是数据库名称+8位数字，这个编号就是id。

State: 状态。每个缺陷都有相应的状态，State 字段记录缺陷相关的状态。状态详情查看 2.6.1。

Headline: 简述。简单描述缺陷的概要情况。

Description: 详细描述。详细说明缺陷发生的位置、操作等情况。

Priority: 优先级。表示修复缺陷的重要程度和应该何时修复。

Severity: 严重性。表示软件缺陷的恶劣程度。

Submitter: 提交人。缺陷是谁提交的，谁就是此缺陷的提交人。

Submit\_Date: 提交日期。错误提交的日期。

Owner: 负责人。指应该处理此条缺陷的人。一般情况下，负责人都是开发组开发此相关模块的程序员。

Symptoms: 错误分类。提交的缺陷记录分类。

Note\_Entry: 解决方案。关于缺陷处理的说明。

Notes\_Log: 验证结果。解决方案的集合。

Attachments: 附件。缺陷的补充说明。习惯上，附件中是一些缺陷状态的抓图。

下面简单的说明一下新字段的添加和修改。

例如：新添加一个“可重现性”字段。

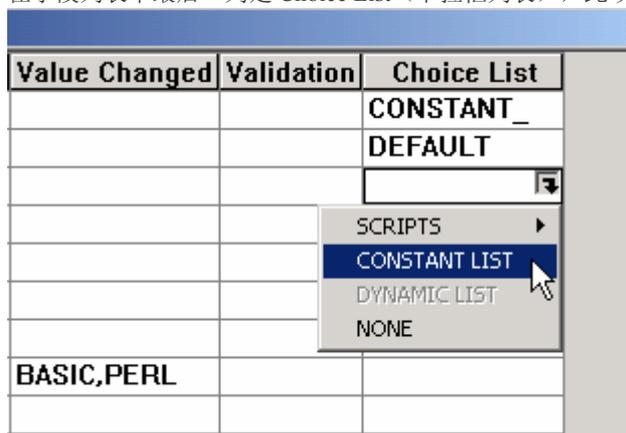
1. 在字段表中，点击右键，出现的菜单中，选择 Add Field，出现字段添加界面。



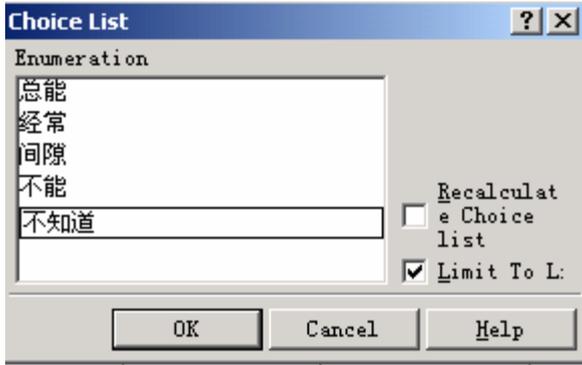
2. 在 Field Name 中填写需要添加的字段名称“可重现性”；DB Column Name 中填写数据库字段名称 kcxx；Type 中选择 SHORT\_STRING 类型；其他的缺省就可以了。填写完毕后，点击右上角的×关闭字段添加窗口，字段就建立成功了。  
按照上面的方法，添加如下字段备用。

| 字段名称 (Field Name) | DB Column Name | 数据类型 (Type)  |
|-------------------|----------------|--------------|
| 可重现性              | Kcxx           | SHORT_STRING |
| 模块                | Mk             | SHORT_STRING |
| 系统                | Xt             | SHORT_STRING |
| 提交时间              | Tjsj           | SHORT_STRING |
| 版本                | Bb             | SHORT_STRING |

在字段列表中最后一列是 Choice List (下拉框列表)，此项十分常用。下面说明如何配置 Choice List。



选择需要使用下拉列表功能的字段，这里我们选择“可重现性”，选中可重现性字段的 Choice List，会出现一个?，点击?，出现一个菜单，从菜单中选择 CONSRANR LIST，出现下图界面。输入下拉框选项“总能”、“经常”、“间隙”、“不能”、“不知道”。选择 Limit To List (选择此条件，在使用下拉框的时候，只能录入下拉框列表内的内容，不允许输入其他的内容)，点击 OK，完成 Chioce List 的设置。



下表中说明了其他使用 Choice List 的字段。

| Choice List 字段名 | Choice List 内容            | 是否选择 Limit To List |
|-----------------|---------------------------|--------------------|
| Priority (优先级)  | 应该修复、必须修复、考虑修复、立即修复       | 是                  |
| Severity (严重性)  | 致命错误、严重错误、一般错误、轻微错误、建议    | 是                  |
| Symptoms (错误分类) | 界面缺陷、数据错误、文档问题、缺少功能、..... | 否                  |
| 可重现性            | 总能、经常、间隙、不能、不知道           | 是                  |
| 模块              | 按软件需求添加                   | 是                  |
| 系统              | 按照需要添加                    | 否                  |
| 版本              | 按需要添加                     | 否                  |

附录：一些项目的解释。

**严重性：**表示软件缺陷的恶劣程度。致命错误一般指导致系统崩溃的错误；严重错误包括数据丢失、数据损坏、功能未实现等；一般错误包括操作性错误、所谓结果等；轻微错误包括错别字、UI 布局等；建议是指软件中值得改良的地方。以上的说明只是建议，具体的分类测试人员可以自行把握判断。

**优先级：**表示修复缺陷的重要程度和应该何时修复。立即修复，说明此缺陷阻止进一步测试，需要立即修复，否则会导致测试的停滞；必须修复，说明此缺陷在产品发布前必须修复，否则会影响软件的发布和使用；应该修复，说明如果时间允许应该修复此缺陷；考虑修复，说明此缺陷即使不修复修复，但是也能发布。

### 2. 5. 2 字段代码控制

在字段列表中，可以对字段进行编程控制。字段的列名是字段的属性，同时也可以作为字段的函数名使用。Default Value 可以控制字段的初始值；Permission 控制字段的读取属性；Value Changed 可以控制字段在内容值变化时的行为；Validation 可以控制字段的取值范围。

|   | Default Value | Permission | Value Changed |
|---|---------------|------------|---------------|
|   |               |            |               |
| ▶ | BASIC         |            |               |
|   | BASIC,PERL    | SCRIPTS ▶  | BASIC         |
|   | BASIC,PERL    | NONE       | PERL          |
|   |               |            |               |
|   |               |            |               |
|   |               |            | BASIC,PERL    |

选择需要使用下拉列表功能的字段，鼠标点击使用的属性控制字段，会出现一个?，点击?，出现一个菜单，从菜单中选择编程语言，就会出现 Script Editor 界面。下图是代码片段。在使用过程中，CQ 基本给出了代码属性的说明和示例，可以按照示例进行简单的编程。如果想更深一步，可以去看 VB 相关的资料（CQ 中使用的 SQA Basic 和 VB 十分类似）或者去看开始菜单中查询 Rational ClearQuest API Reference。

```

Field: 提交时间 Hook FIELD_VALUE_CHANGED

Sub tjsj_DefaultValue(fieldname)
  fieldname As String
  record type name is Defect
  field name is 提交时间
  REM Set the initial value of the field here. Example:
  REM SetFieldValue fieldname, 12345

  SetFieldValue "提交时间", time
End Sub

```

在最后的附录中，我会附上计划中使用的程序控制代码。可以参照我写的代码进行简单的变化。在代码写完后，需要选择菜单中的 Hooks->Compile 进行编译，编译通过后，字段的代码控制就算完成了。

### 2. 5. 3 缺陷窗体设计

在缺陷提交的过程中，窗体是很重要的一个方面。计划的各种设置最后都体现的窗体界面中。下面就说如何去配置缺陷的窗体。缺陷的窗体分为两个部分，分别是提交窗体和处理窗体。处理窗体比提交窗体多了两个页面，其他内容都是一样的。

从工作台（Workspace）中选择 Record Types->Defect->Forms，出现两个选项，第一个选项就是提交窗体（Defect\_Base\_Submit），第二个是处理窗体（Defect\_Base），现在分别说明设置方法。

#### 一、提交窗体

点击 Defect\_Base\_Submit，右侧工作区出现窗体设置界面。



可以从字段列表中鼠标拖动字段到窗体界面中，会自动出现标题和字段对应的属性内容。双击字段内容，会出现字段属性修改窗口，可以自己修改字段的各种属性。



选择菜单中的 Edit->Add Tab 可以添加页面；选择 Edit->Delete Tab 可以删除页面；选中字段后；点击键盘上的 Delete 按键可以删除字段。

在我们所使用的页面上，我重要做的修改是：

1. 把各个字段和页面的标签 (Label) 都修改成了中文。
2. 删除了 Project 字段，换成了模块字段。
3. 删除了 Keywords 字段。
4. 删除了 Customer 页。
5. 添加了高级页，里面设置了一些常用的备用选项。比如系统、可重现性、提交日期、提交时间、提交人、版本等等。高级页面可以自己添加需要的字段。

全部修改完毕后，如下图所示。注意：带下拉框的地方，尽力把下拉框设计的大一些，方便选择，缺省的下拉框长度实在是太小了，很不方便使用。

Defect\_Base\_Submit

属性 | 附件 | 高级

ID:  状态:

简述:

模块:  错误分类:

严重性:

优先级:

负责人:

详细描述:

Defect\_Base\_Submit

属性 | 附件 | 高级

附件

Defect\_Base\_Submit

属性 | 附件 | 高级

系统: [下拉菜单]

可重现性: [下拉菜单]

提交日期: [文本框]

提交时间: [文本框]

提交人: [文本框]

版本: [下拉菜单]

注意：提交日期对应的 Field\_Name 是 Submit\_Date，提交人对应的 Field\_Name 是 Submitter.fullname。

## 二、处理窗体

处理窗体比提交窗体多了两个页面，分别是解决方案页和历史页。

把 Resolution 页和 Customer 页删除，并添加高级页。进行相应的汉化工作。就完成了处理窗体的设置。

下面的是解决方案（Notes）页和历史（History）页的设置完毕图片：

Defect\_Base

属性 | 解决方案 | 附件 | 历史 | 高级

解决方案: [大文本框]

验证结果: [带阴影边框的文本框]



## 2. 6 缺陷变更管理

每一个缺陷都有一个状态（State），经过处理（Actions）后，状态（State）就会发生相应的改变。

下面的内容中，使用[]的就是状态（State），使用<>的就是动作（Action）。

**CQ 是变更管理工具，所以变更设置是 CQ 中最重要的部分。**

CQ 变更设计主要在 Record Types->Defect->States and Actions 中进行。它包括三个部分：State Transition Matrix（状态转移矩阵）、Actions（变更操作的动作）、Behaviors（变更过程中各个字段的属性权限）。在这个变更设计中，我使用了三个组：测试组、开发组、管理组。

测试组负责提交和确认错误的修改；开发组负责修改错误；管理组负责对不可确认的错误进行判断。

### 2. 6. 1 State Transition Matrix（状态转移矩阵）设置

前面说了很多的状态（State），这里就是关于状态设置的。

选择 Record Types->Defect->States and Actions-> State Transition Matrix，打开状态转移矩阵设置页面。

每个缺陷都有自己现有的 State（状态），缺陷状态可以通过 Action 改变。

比如测试人员提交的新缺陷，状态是[等待处理]，[等待处理]的缺陷经过开发人员修改后，变为[已经修改]，测试人员验证[已经修改]的缺陷，发现缺陷被修复，就 Action 缺陷为[关闭]，完成了一个缺陷的变更过程。

通常情况下，缺陷都是从[等待处理]开始，到[关闭]结束。

为了看着直观方便，所以我设计变更过程的时候，把大部分的 Action 和 State 起了一样的名字。

总共录入修改以下状态：

1. 等待处理——测试人员经过提交操作，缺陷为[等待处理]
2. 已经修改——开发人员修复了缺陷，经过<已经修改>Action
3. 暂不修改——错误由于某种原因暂时不处理。，经过<暂不修改>Action
4. 系统限制——确实是问题，但是由于开发工具的限制等原因暂时无法处理错误，开发人员经过<系统显示>Action 标记缺陷为[系统限制]。
5. 间接修复——在修改其他地方的时候把此缺陷记录修复，或者新版本自动修复了此问题缺陷。
6. 无法再现——测试人员提交的错误无法在开发人员那里重现。
7. 使用错误——测试人员由于操作等原因理解误会了某些地方。开发人员可以用使用错误来处理这些不是问题的记录。
8. 等待分配——测试人员无法确认的问题，经过<提交管理员>Action 使缺陷状态变为[等待分配]，等待分配的问题是交给管理组人员处理的。
9. 再次出现——缺陷没有被修改。
10. 关闭——缺陷已修改。或缺陷不再处理。

具体的 Action 和 State 变化关系，参照 2.6.2 进行。

### 2. 6. 2 Actions 设置

Action 是变更中很重要的一个环节。每种成员对应着不同的 Action，通过 Action，处理各种缺陷。

选择 Record Types->Defect->States and Actions->Actions，打开 Action 设置页面。

通过鼠标右键菜单，可以添加（Add Action）、删除（Delete Action）、重命名（Rename Action）、设置（Action Properties）Action。

在这里，我们保留 Import、Init\_Note\_Entry、Send\_Email\_Notif 三项，其他的项目，按照下表进行设置。

其中 Access Control 选项，可以选择哪个组有此操作权限。可以设定为所有组（All Users），也可以选择部分组（User Groups），或者进行编码（SCRIPTS）控制。

Action Name Type Access Control

提交 SUBMIT All Users

提交管理者 CHANGE\_STATE 测试组、开发组  
 再次出现 CHANGE\_STATE 测试组、开发组  
 关闭 CHANGE\_STATE 测试组、开发组  
 暂不修改 CHANGE\_STATE 开发组  
 无法再现 CHANGE\_STATE 开发组  
 间接修复 CHANGE\_STATE 开发组  
 使用错误 CHANGE\_STATE 开发组  
 已经修改 CHANGE\_STATE 开发组  
 系统限制 CHANGE\_STATE 开发组  
 修改 MODIFY 编码控制  
 删除 DELETE 管理组

修改的编码，请参照附录。  
 设置完毕后，如下图所示：

| Action Name      | Type         | Access Control |
|------------------|--------------|----------------|
| 提交               | SUBMIT       | All Users      |
| 提交管理者            | CHANGE_STATE | User Groups    |
| 再次出现             | CHANGE_STATE | User Groups    |
| 关闭               | CHANGE_STATE | User Groups    |
| 暂不修改             | CHANGE_STATE | User Groups    |
| 无法再现             | CHANGE_STATE | User Groups    |
| 间接修复             | CHANGE_STATE | User Groups    |
| 使用错误             | CHANGE_STATE | User Groups    |
| 已经修改             | CHANGE_STATE | User Groups    |
| 系统限制             | CHANGE_STATE | User Groups    |
| 修改               | MODIFY       | BASIC          |
| 删除               | DELETE       | User Groups    |
| Import           | IMPORT       | All Users      |
| Init_Note_Entry  | BASE         | All Users      |
| Send_Email_Notif | BASE         | All Users      |

在 Actions 页后面的列中，可以通过编程对 Action 进行限定。如果在这里对 Action 进行了编程限制，Behaviors 中相关的状态使用 USE\_HOOK。

这里的编程，自己研究，我写了一些代码，但是在 Behaviors 中，编码控制实在有些烦琐，这里我在使用过程中就去掉了那些编码。可以在附录中找到与此相关的一些代码片段。修改的 Access Control 的编码主要是限制非缺陷提交人只能浏览记录，不可以修改记录。

比如经过<无法再现>Action 操作后，缺陷的状态变为[无法再现]。一定要记住，虽然两者名称，但本质是不同的，一个是 State，一个是 Action。

在状态转移矩阵中，右键点击状态名称，会出现列表，里面可以添加（Add State）、删除（Delete State）、重命名（Rename State）、设置缺省值（Properties）State。

状态转移矩阵中，状态从列到行进行变化，比如[等待处理]列状态经过<已经修改>Action 后，变为[已经修改]行状态。

下面是说明如何设计 State Transition Matrix 中的 Action 和 State 管理。

鼠标右键点击需要设置的 Action 状态，从列表中选择 Action Properties，打开 Action 设置窗体。选中窗体的 State 页，在此设计 Action 和 State 的交互关系。



State 页中的 Source（源状态），相当于 State Transition Matrix（状态转移矩阵）中的列，Destination State（目的状态）相当于状态转移矩阵的行。也就是说，经过当前的 Action，可以把 Source 中的状态，转变为 Destination State 中的状态。可以选择多个 Source，但是只能有一个 Destination State。

下面图表是各个 Action 的设置情况：

Action Name ; Sourc ; Destination State

提交 ; 无 ; 等待处理

提交管理者 ; 暂不修改、系统限制、无法再现、间接修复、使用错误 ; 等待分配

再次出现 ; 等待分配、已经修改、间接修复、无法再现、使用错误 ; 再次出现

关闭 ; 等待分配、已经修改、间接修复、无法再现、使用错误 ; 关闭

暂不修改 ; 等待处理、再次出现 ; 暂不修改

无法再现 ; 等待处理、再次出现 ; 无法再现

间接修复 ; 等待处理、再次出现 ; 间接修复

使用错误 ; 等待处理、再次出现 ; 使用错误

已经修改 ; 等待处理、再次出现 ; 已经修改

系统限制 ; 等待处理、再次出现 ; 系统限制

注意，以上的 Action 是可以按照软件的需要随时修改的，上面的表格只是一种建议。

### 2. 6. 3 Behaviors 设置

选择 Record Types->Defect->States and Actions->Behaviors，打开 Behaviors 设置页面。

这里是整个 CQ 中最麻烦的地方。主要是设置 CQ 中的字段在各种缺陷下的权限。

Behaviors 设置中，行是各个字段，列是缺陷状态。表中内容说明字段在状态下的权限。在表中点击鼠标右键就会出现权限菜单。

字段权限包括：

1. Read Only: 字段只能读取，不能修改。

2. Mandatory: 字段必需填写。

3. Optional: 字段可添可不添。

4. Use Hook: 字段权限通过编程进行控制。

这里的内容，我就不一一列举了，根据需要自己修改。这里牵扯了我最多的精力，每次总会在这里因为字段的限制出问题，现在字段权限设置我仍然不是很满意。

注意：这里字段对应的状态是指 Action 以后的状态，而不是缺陷当前的状态。比如原本是[等待处理]状态的缺陷，经过<已经处理>Action 后，这个时候限制缺陷的 Behaviors 不是[等待处理]，而是[已经处理]，这里要特别注意。

### 2. 7 其他设置

#### 2. 7. 1 File 菜单

Delete Schema: 删除计划。

Delete Schema Version: 删除计划的版本。这个可以选择删除任意的计划版本。

Undo Check Out: 不保留当前版本，恢复 Check In 状态。

Test Work: 如果使用了 Test Database，就可以进行 Test Work，Test Database 和 Product Database 是不同的数据库。

Validate: 校验当前计划是否有错误。这个应该在 Save Work 前进行。

#### 2. 7. 2 Edit 菜单

Add(Edit、Delete) Record Type/Family: 创建（修改、删除）记录类型。我们一般所用的是 Defect 类型。

一般的添加、修改、删除操作，都可以在 Edit 菜单中找到，而且一些功能，只能在这里完成。

#### 2. 7. 3 View 菜单

Schema Summary: 从这里，可以看到计划的版本、连接的数据库等记录。

Database Summary: 查看使用的数据库各种信息。

#### 2. 7. 4 Database

Delete Databases: 删除数据库。

Undelete Databases: 恢复原先删除的数据库。

View Database Properties: 查看数据库属性。查看计划连接的数据库的详细情况。

Move User Database: 把现在使用的数据库转移到其他的地方。比如 test 计划现在使用的数据库是 test 机器的 test 数据库。那么你可以通过 Move User Database 把 test 计划连接的数据库设置到 test8 机器上的 test 数据库。说明的是, 在移动数据库的同时, 相关的缺陷记录会一起移动。实际相当于一个数据库 Copy 的过程。

Update User Database: 更新使用的数据库。这个实际是修改数据库连接属性。Update 和 Move 不同的地方在于, Update 只连接, 不转移记录。

Set Test Database: 设置 test 数据库。这个不是我们所使用的数据库。在我们创建数据库连接的时候, 选择的是 Product 库而不是 Test 库, Test 库使用中会有一些问题, 在最后的附录中会给予说明。

Database Timeout: Timeout (延迟) 和 Poll Interval (间隔) 设置。一般设置为 0 就可以了。

#### 四、Rational ClearQuest Web 服务器配置

提交缺陷, 除了使用 CQ 客户端外, 也可以通过浏览器进行提交。

这里的浏览器要求 IE5.01 版本以上, Netscape 4.72 版本以上。

如果想要使用 CQ 的 Web 端功能, CQ 服务器在安装 CQ 的过程中, 一定要选择 Custom (自定义安装), 在 Choose Features 中, 把 Web Server Components 选中。

下面说明 CQ Web 服务器的配置。建议使用 Win2000 Server 系统。下面配置使用 Win2000 Server (非域中机器) 进行说明的。如果进行其他系统的配置, 参照 ClearQuest 安装目录下 books 目录里面的 InstallationGuideClearQuest.pdf。

CQ Web 使用的是 ASP。

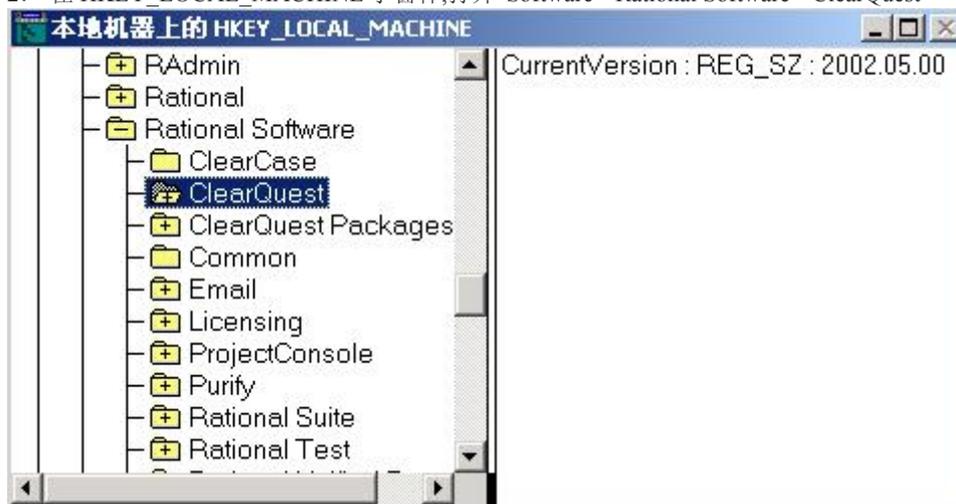
下面的设置, 都是使用的 Administrators 用户组中的用户登陆的系统。

##### 1. 设置注册表存取权限

在 IE 使用的过程中, 缺省的 Web 登陆用户是 anonymous (匿名用户), 但是在系统中, anonymous 用户并没有太多的存取权限, 下面就是设置 anonymous 用户的存取权限。

1. 点击“开始->运行”, 输入“regedt32”, 点击确定打开“注册表编辑器”。

2. 在 HKEY\_LOCAL\_MACHINE 子窗体, 打开“Software->Rational Software->ClearQuest”



3. 点击注册表编辑器的菜单“安全->权限”, 出现 ClearQuest 权限设置窗口。

4. 点击添加, 打开“选择用户或者组”界面。

5. 选择 IUSR\_机器名用户, 这里我选择的是 IUSR\_TEST

6. 点击添加, 再点击确定, 选择 IUSR\_TEST 用户并关闭“选择用户或者组”界面。

7. 在名称中选中刚才添加的 Internet 来宾账号, 在权限中, 允许完全控制。



8. 选择“允许将来自父系的可继承权限传播给对象”，点击“确定”完成权限设置。

9. 在 HKEY\_USERS 子窗体,打开“.Default->Software->Rational Software->ClearQuest”

10. 重复第 3 步到第 8 步, 设置 HKEY\_USERS。

2. NTFS 格式机器的读取权限设置

非 NTFS 格式的机器可以不用看这一小节。

在 NTFS 格式的机器上, 每个目录都有相应的读取权限, 所以下面的就是设置 internet 匿名用户也有 Web 目录的读取权限。

在 CQ 的安装目录下, 有一个 WWW 目录, 鼠标右键点击 WWW 目录, 出现 WWW 目录属性设置界面, 选择其中的安全页, 点击“添加”按钮, 选择 IUSR\_机器名的用户添加到用户列表中, 如下图所示, 点击“确定”完成 WWW 目录权限设置。



我个人不很喜欢 NTFS 格式，也许安全性高了些，但使用不是很方便，所以我本身用的 TEST 机器是 FAT32 格式，FAT32 格式的机器不用进行目录权限设置。上面的抓图使用的是 TEST8 机器，和前面的有差别，这里要特别注意。

如果是 NTFS 格式的机器，安装的时候，可能没有安装 cache 目录。手工在 WWW 目录中添加 cache 目录，并按照上面的方法，给予 cache 目录以 IUSR\_机器名的存取权限。