

ClearQuest 联机教程

概述

开始本教程之前，请确认已经读过 Rational ClearQuest 的介绍。

在这个教程中，你将学习如何使用 ClearQuest Designer 和 ClearQuest client 开始执行基本的 ClearQuest 管理任务。完成本教程后，你就可以开始为用户自定义 ClearQuest client。

该教程有两部分，每部分都包括三节课程：

- 第一部分的第 1, 2 和第 3 课，介绍 ClearQuest Designer，你将学习如何创建一个 schema 和 user database，自定义 schema 并且对 user database 应用你的定制。你将创建 users 和 user groups。
- 第二部分的第 4, 5 和第 6 课，介绍 ClearQuest Client，你将学习如何创建公共查询和图表，创建从 ClearQuest 数据中得到的报告，配置 ClearQuest e-mail。

第一部分：运行 ClearQuest Designer

ClearQuest Designer 是一种为 ClearQuest client 定制行为和用户界面的工具。这部分提供三节课程来介绍如何使用 ClearQuest Designer。

- 课程 1，将学习如何登录到 ClearQuest Designer 并且创建一个新的 schema 和 databases。
- 课程 2，将学习如何定义一个 schema。
- 课程 3，将学习如何管理 users 和 user groups，创建 new users，将他们添加到 groups，使用 user 信息更新 database。

创建 Schemas 和 Databases

本课中，将学习如何做以下的事情：

- 登录到 ClearQuest Designer
- 创建一个新的 schema
- 创建一个新的 user database 并且关联至相应的 schema
- 创建一个测试 database

登录到 ClearQuest Designer

在这个练习中，你将登录到 ClearQuest Designer。

1. 从开始菜单中选择 ClearQuest Designer。
2. 在 ClearQuest 登录对话框中，User Name 输入 admin，Password 为空。
3. 单击 OK。
4. 出现 Open Schema 对话框，单击 Cancel。

一般情况下，你可以通过 Open Schema 对话框打开准备使用的 schema，但在接下来的练习中，你将创建一个新的 schema。

接下来是什么？

在接下来的练习中，你将创建一个新的 schema。

创建一个新的 schema

一个 ClearQuest schema 包含元数据 that defines the process for working with records within the ClearQuest client.

利用 ClearQuest Designer，可以以一个已经存在的 schema 为模版创建一个新的 schema。所有的 schema 都被保存在 schema 储存库中。

在这个练习中，你将创建一个以 TestStudio schema 为模版的新的 schema。如果 TestStudio schema 不能使用，请选择 Common schema 来代替。

1. 选择 File > New Schema 打开一个新的 Schema 对话框。

- 2.从 schema 列表中选择 TestStudio, version 1, 单击 Next。
- 3.在 Schema Name 栏中输入 your-username_schema。
- 4.在 Comment 文本框中输入 Tutorial schema, 单击 Finish。
- 5.出现 if you want to create a database to associate with your new schema 时, 单击 No。在下一个练习中你将创建 database。
- 6.出现 if you want to check out the schema for editing, 单击 No。

创建一个新的 User Database

The user database is the database ClearQuest client users use to work with change request records.

user database 是 ClearQuest client 用户使用的变更请求记录数据库。

在这个练习中, 你将创建一个新的 user database 并且将其关联至上一个练习中创建的 your-username_schema schema。

- 1.选择 Database > New Database 打开一个新的 Database 对话框。
- 2.按照下列要求完成新的 Database 对话框:
 - a.在 Logical Database Name 中输入 my_db。
逻辑数据库的名字字符长度在 1-5 之间。
 - b.在 Comment 字符框中, 输入 This database uses the tutorial schema, your-username_schema。
 - c.单击 Next。
- 3.从 Vendor choice 列表中选择 MS_ACCESS。
- 在 Physical Database Name 中输入 c:\my_db 作为数据库的完整路径名称。路径中指定的目录必须存在。例如, 如果你指定了 c:\temp\my_db, 那么 temp 目录必须存在。
如果数据库不存在, 则 ClearQuest 会在指定的目录中创建。
- 4.选择 Test Database 选项, 单击 Next。
- 5.在 Timeout (240 minutes) 和 Poll Interval (1 minute)中接受默认值, 单击 Next。
- 6.在 schema 列表中选择 your-username_schema, version 1, 单击 Finish。
- 7.出现数据库创建成功提示消息窗口时, 单击 OK。

ClearQuest Designer 创建 my_db 数据库并且使用 your-username_schema schema 进行初始化。

创建一个测试数据库

在接下来的练习中, 你将创建一个测试用户数据库, 该数据库将在后面的定制 your-username_schema schema 教程中用到。通过使用测试数据库, 你可以了解 how your schema customizations perform in the ClearQuest client without affecting your user database.

- 1.选择 Database > New Database 打开一个新的 Database 对话框。
- 2.在 Logical Database Name 中输入 test。 (逻辑数据库的名字字符长度在 1-5 之间。)在 Comment 文本框中输入 This is the test database for the tutorial schema, your-username_schema。单击 Next。
- 3.从 Vendor 列表中选择 MS_ACCESS。
- 在 Physical Database Name 中为数据库输入完整路径名称。例如 c:\test。路径中指定的目录必须存在。例如, 如果你指定了 c:\temp\my_db, 那么 temp 目录必须存在。
如果数据库不存在, 则 ClearQuest 会在指定的目录中创建。
- 选择 Test Database 选项单击 Next。
- 4.在 Timeout 和 Poll Interval 中接受默认值, 单击 Next。
- 5.在 schema 列表中选择 your-username_schema, version 1, 单击 Finish。
- 6.出现数据库创建成功提示消息窗口时, 单击 OK。

ClearQuest Designer 创建了 test database 并且用 your-username_schema schema 进行初始化。
注释：你应该只是用 test databases 作为临时数据库来验证你的 schema 的变更。不能将重要的数据放至 test database。

定制一个 Schema

在该课程中，你将学习如何做以下的事情：

- 添加一个新的 field
- 定义状态转换
- 测试并保存 schema 的变更
- 对用户数据库应用 schema 的变更

添加一个新的 field

注释：如果使用的是 TeamTest 版本的 ClearQuest，你不能在 schema 中增加一个 field。你只能自定义已经存在的 fields。

在接下来的练习中，你将完成以下操作：

1.检出一个 schema。

ClearQuest 在 schema 储存库中存储着已有 schemas 的所有版本，你必须从 schema 储存库中检出 schema 的最新的版本。

2.创建 field。

在 ClearQuest 中，你可以对一个已有的或新的 schema 增加、删除或者修改 fields。

3.为 field 增加一个 hook（分支代码）。

Hooks are entry points for pieces of code that execute automatically at specified times (triggers), that extend ClearQuest functionality.

分支代码是在指定时间（触发器）自动执行的代码部分的进入点，它扩展了 ClearQuest 的功能。

4.在记录窗体中增加 field。

ClearQuest 通过使用窗体来联系一个记录类型并显示相关信息。ClearQuest 也允许用户使用窗体来提交新的相应的记录。

检出一个 schema

ClearQuest 在 schema 储存库中存储着已有 schemas 的所有版本，为了更改一个 schema，你必须从 schema 储存库中检出 schema 的最后的版本。

在这个练习中，你将检出 your-username_schema。该课程的所有练习都将使用这个被检出的 schema。

1.在 ClearQuest Designer 中，选择 File > Open Schema 打开 Open Schema 窗体。

2.选择 Checkout a schema to edit。

注释：在 Open Schema 窗体中如果选择了 View a version of an existing schema，你将不能修改这个 schema。

3.选择 your-username_schema, version 1，单击 Next。

4.在备注中输入 Adding defect record field to the record form and a new state and action。

5.单击 Finish。

ClearQuest Designer 窗口左侧显示 your-username_schema 的工作区间。请注意你所工作的 your-username_schema 的版本号为版本 2。当你打开一个 schema 进行编辑时，ClearQuest Designer 为这个 schema 自动创建一个新版本。

增加一个新的 Field

注释：如果使用的是 TeamTest 版本的 ClearQuest 并且没有一个完整的 ClearQuest 许可证，你就不能在 schema 中增加一个 field。你只能自定义已经存在的 fields 就像增加一个 field 分支代码。

在这个练习中，你将编辑 your-username_schema schema，通过创建一个新的 field，允许用户在缺陷记录中输入用户编号。

1.在工作区，展开 Record Types > Defect 并且双击 Fields 显示 Fields 网格。

Fields 网格显示了记录中的所有 fields。每个 field 列显示其属性，包括 field 名称，类型，缺省值，权限，有效性等等。

2.选择 Edit > Add Field，打开 Field Properties 对话框，选择 General 标签，在 Field Name 中输入 User_number。在 Type 中选择 INT，因为该字段包含整数。

缺省的数据类型是 SHORT_STRING。

DB Column Name 的默认值与 field 名称相同。

作为默认选项，选择项 Visible in query 已被选择。

3.In the Help Text tab, as Enter your user number type a number between 1 and 100. 在 Help Text 标签项，as Enter your user number type a number between 1 and 100。

该字段必须为 1-100 之间的整数。当用户在 ClearQuest 客户端通过快捷菜单使用帮助时将看到该帮助信息。

4.关闭 Field Properties 对话框。

请注意，这时类型为 INT 的 User_number field 显示在 Fields 网格中的底部。

创建一个 Field 分支代码

注释：如果使用的是 TeamTest 版本的 ClearQuest 并且没有一个完整的 ClearQuest 许可证，你就不能在 schema 中增加一个 field。你只能自定义已经存在的 fields。

你可以编辑 functionality of fields，操作和为记录指定的分支代码。当 ClearQuest 客户端用户执行了一个操作或一个 field 中输入了一个值，ClearQuest 运行指定的分支程序代码。

在这个练习中，你将为上一练习中生成的 User_number field 添加一个 validation hook，这个分支代码用来验证 User_number field 的值。

1.展开 Record Types > Defect 项，双击 Fields。

2.在 Fields 网格中，点击 User_number field 中的 Validation 单元格，然后点击下拉箭头。

选择 SCRIPTS > BASIC 打开脚本编辑器。脚本编辑器以灰色输出行的形式提供一个 Visual Basic 脚本框架。

脚本编辑器的顶部，Fields 选项显示 User_number，并且 Hook Types 选项显示

FIELD_VALIDATION。如果显示的不是这些，通过滚动选中这些选项。

3.Copy 下面的示例代码并粘贴到脚本编辑器中注释行的下面：

```
REM End If
Dim value_info
Set value_info=GetFieldValue(fieldname)
If Not IsNumeric(value_info.GetValue) Then
User_number_Validation="Must be an integer between 1 and 100"
ElseIf (value_info.Getvalue<1) or (value_info.Getvalue>100) Then
```

User_number_Validation="Must be between 1 and 100"

End If

这个例子中，当用户在 field 中输入一个数字，ClearQuest 客户端运行相应的验证分支代码。

如果验证分支代码返回一个非空字符串，用户将被提示该 field 中包含无效值。

4.选择 Hooks > Compile。编译脚本并检测确定没有语法错误。如果没有错误，在 Script Errors 框中将会显示 No error(s) found 。

5.关闭脚本编辑器。

注释：分支代码能够影响 field 的状态，因此在发布它们之前，你必须慎重的计划并测试这些代码。例如，ClearQuest 设置未初始化的整数为 0。练习中添加的分支代码检测这个 field 的值，如果发现小于 1 或大于 100，则返回错误信息。这就使得 User_number field 成为强制性的 field，因为 0 不能通过有效性检测。

然而，如果你看 User_number field 的属性（展开 Record Types > Defect > States and Actions 并双击 Behaviors），该 field 被定义为可选 field。因此，这个验证分支代码实际上改变了这个 field 的属性。

在记录窗体中添加 field

注释：如果使用的是 TeamTest 版本的 ClearQuest 并且没有一个完整的 ClearQuest 许可证，你就不能在 schema 中增加一个 field。你只能自定义已经存在的 fields。

现在，你刚才创建的 User_number field 对用户而言是不可见的，因为它没有出现在任何窗体中。在这个练习中，你将在提交窗体中添加 User_number field，从而使用户可以在提交缺陷报告时确定他们的用户编号。

1.工作区内，展开 Record Types > Defect > Forms 并双击 Defect_Base_Submit。

打开 Defect_Base_Submit 窗体的同时，Field List 和 Controls Pallet 同时也被打开。

2.如果需要的话，最大化 ClearQuest Designer 窗口，拉伸 Defect_Base_Submit 窗口以便在窗口的底部可以添加一个新的 field。

3.在 Field List 内，滚动并找到 User_number field。单击 User_number（单击名称而不是图标）并拖动至 Defect_Base_Submit 窗体的底部。

Field 名称和一个文本框出现在窗体中。ClearQuest 为 field 自动选择一个 Text Box 控件。

4.双击 text box 显示属性表。浏览控件提供的标签和控制功能，然后单击 OK 关闭属性表。

5.工作区内，右键单击 Defect_Base_Submit。确定 Submit Form 项在快捷菜单中已被选中。如果该项目没有被选中，单击并添加该选项。

6.关闭 Defect_Base_Submit 窗口。

注释：

- ClearQuest 使用窗体与记录类型进行联系，并使用该窗体显示相应记录类型的详细信息。
- 每一个记录类型可以包含一个或多个窗体：记录窗体和提交窗体。提交窗体不是必须的，但是每一个 schema 必须包含一个记录窗体。

添加状态和操作以及定义状态转变

ClearQuest 中，记录要经过操作来从一个状态转变为另一个状态。状态转换矩阵跟踪了一个记录的生命周期。

接下来的练习中，你将创建一个新的状态，新的操作并且定义一个状态转换。

ClearQuest Designer 状态转移矩阵确保用户按照 ClearQuest 管理员定义的矩阵来传递记录。

添加一个新的状态

ClearQuest 中，记录从一个状态转换为另一个是由于 ClearQuest 客户端用户执行了操作。例如，使用 your-username_schema schema，ClearQuest 客户端用户点击 New Defect 向数据库中提

交一个记录。这时该记录为提交状态，从这个状态，用户可以选择 **Actions > Assign** 来为一名工程师分配变更需求。

一个记录的可用状态和状态转换操作在 **ClearQuest Designer** 状态转换矩阵中显示。

在这个练习中，你将为 **your-username_schema schema** 增加一个叫做 **Reassigned** 的新的状态。该状态允许用户将缺陷重新指派给其他人。

1.在工作区内，展开 **Record Types > Defect > States and Actions** 并双击 **State Transition Matrix**。状态转换矩阵内，列值为源状态，行值为目的状态。交叉单元格内是将记录从源状态转换到目的状态的操作。

2.选择 **Edit > Add State** 显示 **Add State** 对话框。

3.在 **Name field** 输入 **Reassigned**，单击 **OK**。

Reassigned 状态在状态转换矩阵中的行和列同时出现。

添加一个新操作

ClearQuest 使用操作来编辑及转换记录状态。这个练习中，将添加名为 **Reassign** 的操作，该操作允许用户将上一练习中生成的记录转换为 **Reassigned** 状态。

1.工作区内，展开 **Record Types > Defect > States and Actions**，双击 **Actions** 打开 **Actions** 网格。

2.选择 **Edit > Add Action** 显示 **Defect Action** 对话框。

3.选择 **General** 标签，在 **Action Name** 内输入 **Reassign**。**ClearQuest** 将操作类型缺省设置为 **CHANGE_STATE**。

4.关闭 **Defect Action** 对话框。

Reassign 操作在操作网格的底部出现。

定义一个状态转换

前面的练习中，所创建的 **Reassign** 操作类型是 **CHANGE_STATE**。只有类型为 **SUBMIT**，**CHANGE_STATE**，或者 **DUPLICATE** 的操作才可以开始状态转换。其他类型的操作不能影响记录状态。

在这个练习中，你将为 **Reassign** 操作通过指定源状态及目的状态来定义一个状态转换。其结果将是，**ClearQuest** 客户端用户能够通过选择 **Actions > Reassign**，将所有记录为 **Opened** 和 **Resolved** 的状态转换为 **Reassigned** 状态。

1.展开 **Record Types > Defect > States and Actions** 双击 **Actions**。

2.右键单击行标签 **Reassign**，选择 **Action Properties**，出现 **Action Properties** 对话框。

3.在 **State** 标签内，选择 **Opened** 和 **Resolved** 作为源状态。选择 **Reassigned** 作为目的状态。

4.关闭 **Action Properties** 对话框，应用该状态转换。

5.双击工作区内的 **State Transition Matrix**，检查 **ClearQuest** 是否应用了 **Reassigned** 状态的转换。

状态转换矩阵显示无论是从 **Opened** 还是 **Resolved** 状态，**Reassign** 操作将转换记录为 **Reassigned** 状态。

6.一旦记录处于 **Reassigned** 状态，用户应当能够打开记录。为了达到该目的，双击工作区内的 **Actions** 打开 **Actions** 网格。右键单击 **Open** 操作并选择 **Action Properties**。

在状态标签内，选择 **Reassigned** 作为源状态，**Opened** 作为目的状态。

7.关闭对话框。

重新观察状态转换矩阵，可以看到 **Open** 操作可以将记录从 **Reassigned** 状态转换为 **Opened** 状态。

接下来的练习中，你将从 **ClearQuest** 客户端测试先前的工作，看 **your-username_schema** 是否象定制好的那样工作。

检查你的设置

在检入 **schema** 并使得这些变化对于用户可以使用之前，你需要测试并确认 **schema** 的变化。在这个练习中，你将使用在 **Lesson 1** 中创建的测试数据库来检查基于 **your-username_schema** **schema** 所作的工作。这是在前进中检测你的工作的捷径；你可以在 **ClearQuest** 客户端很安全的测试 **schema** 中最近的大部分的变化，这样做并不会影响到你的 **my_db** 用户数据库。

首先，设定你的测试数据库

- 1.在 **ClearQuest Designer** 中，选择 **Database > Set Test Database**。
- 2.在 **Test Database** 对话框内，选择先前创建的测试数据库 **test**。
- 3.用户名输入 **admin**，口令为空。单击 **OK**。

当你使用 **ClearQuest** 客户端测试你的 **schema** 时，**ClearQuest** 使用这些信息进行登录。

现在，测试你的 **schema**

- 1.在 **ClearQuest Designer** 中，选择 **File > Test Work**。

ClearQuest Designer 保存 **schema** 并对其确认，在 **ClearQuest Designer** 窗口底部的确认面板显示发现的错误信息。如果有错误发生，你可以通过右键点击该错误信息并选择 **What's This?** 来得到关于该错误的详细描述。

ClearQuest 客户端被自动打开，因此你可以测试在 **your-username_schema** 中已经作的工作。

- 1.在 **ClearQuest** 客户端，单击 **New Defect** 来提交一个缺陷记录。在记录窗体的底部可以看到名为 **User_number** 的新的 **field**。
- 2.在 **User_number field** 内，输入 **101** 来测试你的验证分支代码。单击 **OK**。弹出消息框 **These fields have invalid values: User_number**。
- 3.右键单击 **User_number field** 并选择 **Error Message** 来显示引起错误的原因。此时，系统显示 **Must be between 1 and 100**。

记录提交后，你还不能测试新的 **Reassigned** 状态，因为还没有任何查询允许你检索记录。接下来，你将学习如何在 **ClearQuest** 客户端创建公共查询。

- 4.退出 **ClearQuest** 客户端。

注释：

- 当你检入 **schema** 之前的任何时候，都可以通过选择 **File > Save Work** 来保存工作。（这个选项只有当你还没有执行 **File > Test Work** 时是可用的。）在常规基础上保存工作是一个好的方法，从 **ClearQuest Designer** 注销后，当重新登录时可以继续执行你的工作。
- 可以通过选择 **File > Validate** 在任何时候来验证你的 **schema**。
- 状态和错误信息显示在位于窗口底部的验证面板内。