The slide features a blue header with the IBM logo and the text "IBM Software Group | Rational software". The main content area is white and contains the title "IBM® Rational® ClearQuest®" and the subtitle "ClearQuest parent child hierarchy". Below the subtitle is the "Rational software" logo. A horizontal bar with various icons is positioned below the logo. The footer is blue and includes the text "@business on demand.", "© 2008 IBM Corporation", and "Updated February 8, 2008".

IBM Software Group | Rational software

IBM® Rational® ClearQuest®

ClearQuest parent child hierarchy

Rational software

@business on demand.

© 2008 IBM Corporation
Updated February 8, 2008

This presentation explains parent child hierarchy (sometimes called parent child relationship). This presentation assumes that you are using at least ClearQuest 7.0.1 and you are familiar with schema design in the ClearQuest Designer.

Module objectives

- These topics are covered in this module:
 - ▶ Setting up a parent child hierarchy
 - ▶ Creating a back reference
 - ▶ Order of adding records
- Upon completion of this module, you will be able to:
 - ▶ Understand how to set up a parent child hierarchy
 - ▶ Understand that adding records can only be done in one direction



In ClearQuest, you may want to set up a relationship between two record types and display the relationship in both. This module will cover setting up a parent child hierarchy, creating a back reference, and a discussion on the order of adding records.

ClearQuest parent child – Add

- In the ClearQuest Designer, you can relate two record types by creating either a reference (one-to-one relationship) or reference_list field (one-to-many relationship)
- You will add the relationship between the records from one record type (known as the parent record)
- The second record type (known as the child record) will display the records it is related to but you cannot create a relationship between the records from the child record
- Map out your data before setting this relationship up so you can decide where to create the relationship and which record will be the parent



In the ClearQuest Designer, you can relate two record types by creating either a reference, a one-to-one relationship, or reference_list field – a one-to-many relationship. You will add the relationship between the records from one record type (known as the parent record).

The second record type, known as the child record, will display the records it is related to but you cannot create a relationship between the records from the child record. A helpful note: map out your data before setting this relationship up so you can decide where to create the relationship and which record will be the parent.

ClearQuest parent child - Create

Add a reference_list or reference type field to the parent record. In this example, defect is the parent record.

Defect Fields - child_records

This dialog box contains properties for the selected field .

General Help Text

Field Name: child_records

DB Column Name: child_records

Type: REFERENCE_LIST

Visible in Query

Owned By: None

Reference To: Customer

Back Reference

4

ClearQuest parent child hierarchy

4

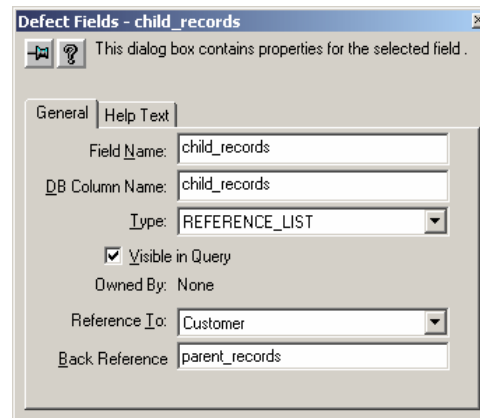
© 2008 IBM Corporation

In the example given here, the defect record has a new field called child_records that is a reference_list that points to the customer record.

You are not adding the customer record to the defect record, you are only setting up a field that shows the relationship between the defect and customer records.

ClearQuest parent child - Create

Enter a name for a back reference field. A back reference field is a read-only field that allows you to view the link from the perspective of the child record.



The screenshot shows a dialog box titled "Defect Fields - child_records". It has a "General" tab selected. The fields are as follows:

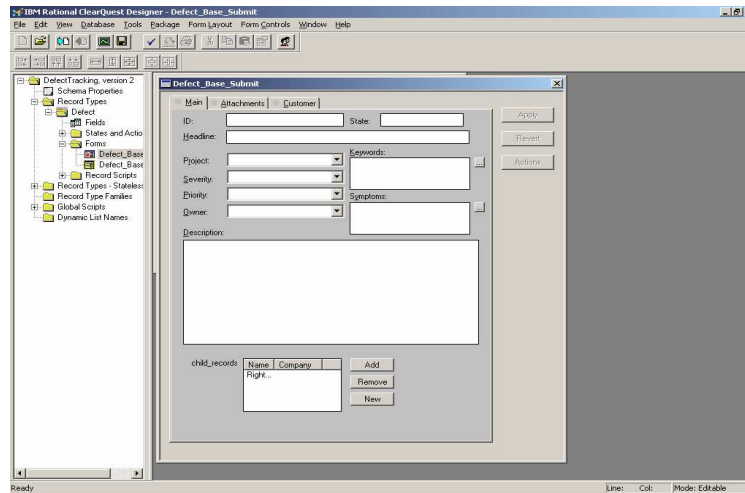
Field Name	Value
Field Name	child_records
DB Column Name	child_records
Type	REFERENCE_LIST
Visible in Query	<input checked="" type="checkbox"/>
Owned By	None
Reference To	Customer
Back Reference	parent_records

In this example, the back reference field is called parent_records. You name the field and it can be called anything although you probably want to name it something meaningful to you and your users.

Many people are confused about where the back reference field is stored. Think of it as belonging to the record just above it on the form. The back reference field is stored in the child record.

ClearQuest parent child - Create

Drag the child_records field onto the defect forms.

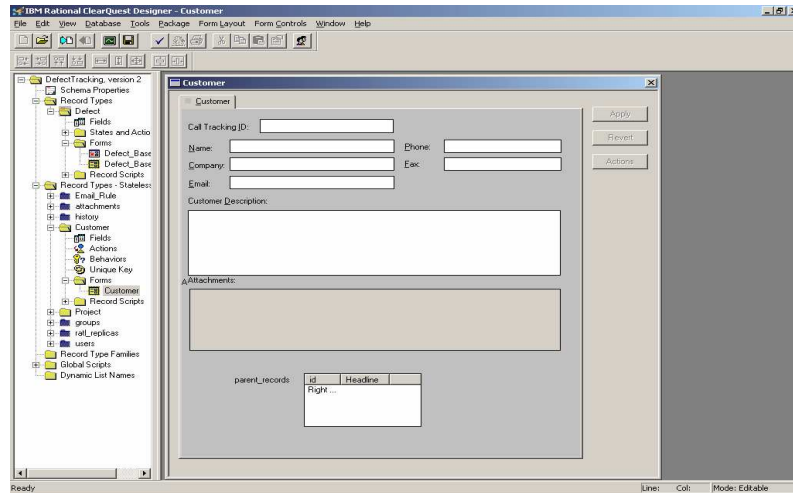


Note that when you drag the child_records field onto this form, it has the Add, Remove and New buttons that are typical of a list view control.

If you were to add the control before choosing the field instead of dragging the field onto the form, you would select the parent/child control in the Control Palette.

ClearQuest parent child - Create

On the customer record, drag the back reference field, parent_records, onto the form.



When you drag the back reference field onto the Customer form, note that you do not see the Add, Remove and New buttons. That is because the back-reference field only shows you a relationship that was established in the parent record. In this example, the parent record is the defect record.

ClearQuest parent child relationship

- You can only create the relationship to child records from the parent. In the example shown on the previous slide, you can only associate the customer record with the defect record from the defect record
- Before you establish this relationship, give some thought to which record will be the parent record



When you relate two record types in this way, you are indicating that one record type is the main record and you want to associate the second record type to the main record. ClearQuest allows you to associate the child record from the main record but does not allow you to add the parent record from the child record.

Summary

- Parent child hierarchy shows only the relationship between two record types
- You associate the two record types from one record known as the parent record
- You cannot then associate the two records from the parent



In summary, you should now know what a parent child hierarchy is, and how to set one up. Remember, parent/child hierarchy shows only the relationship between two record types. You know that the association is created from one record type but the association can be shown in both record types.

Feedback

Your feedback is valuable

You can help improve the quality of IBM Education Assistant content to better meet your needs by providing feedback.

- Did you find this module useful?
- Did it help you solve a problem or answer a question?
- Do you have suggestions for improvements?

Click to send e-mail feedback:

mailto:iea@us.ibm.com?subject=Feedback_about_RCQ_Module4_ParentChildHierarchy.ppt

This module is also available in PDF format at: ../RCQ_Module4_ParentChildHierarchy.pdf



You can help improve the quality of IBM Education Assistant content by providing feedback.

Trademarks, copyrights, and disclaimers

The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both:

ClearQuest IBM Rational

Rational is a trademark of International Business Machines Corporation and Rational Software Corporation in the United States, other countries, or both.

Product data has been reviewed for accuracy as of the date of initial publication. Product data is subject to change without notice. This document could include technical inaccuracies or typographical errors. IBM may make improvements or changes in the products or programs described herein at any time without notice. Any statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business. Any reference to an IBM Program Product in this document is not intended to state or imply that only that program product may be used. Any functionally equivalent program, that does not infringe IBM's intellectual property rights, may be used instead.

Information is provided "AS IS" without warranty of any kind. THE INFORMATION PROVIDED IN THIS DOCUMENT IS DISTRIBUTED "AS IS" WITHOUT ANY WARRANTY, EITHER EXPRESS OR IMPLIED. IBM EXPRESSLY DISCLAIMS ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. IBM shall have no responsibility to update this information. IBM products are warranted, if at all, according to the terms and conditions of the agreements (for example, IBM Customer Agreement, Statement of Limited Warranty, International Program License Agreement, etc.) under which they are provided. Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products.

IBM makes no representations or warranties, express or implied, regarding non-IBM products and services.

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents or copyrights. Inquiries regarding patent or copyright licenses should be made, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. The actual throughput or performance that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput or performance improvements equivalent to the ratios stated here.

© Copyright International Business Machines Corporation 2008. All rights reserved.

Note to U.S. Government Users - Documentation related to restricted rights-Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract and IBM Corp.