



# Infogix ACR Solutions for Common Coding Challenges

---

Release 4.3

**Home Office**

1240 East Diehl Road, Suite 400

Naperville, IL 60563

Tel: +1.630.505.1800

[www.infogix.com](http://www.infogix.com)

Infogix, the Infogix checkmark logo, Unitech Systems, Inc., Infogix Assure, Infogix Insight, ACR, ACR/Detail, ACR/Summary, ACR/Workbench, ACR/Connector, ACR/Instream, ACR/File, Information Integrity, and Information With Confidence are registered trademarks of Infogix, Inc.

Infogix ACR, Infogix EM, Infogix ER, Infogix Perceive, Infogix Controls, Infogix Nexix, ACR/TransMatch, and The Information Integrity Experts are trademarks of Infogix, Inc.

Any other trademarks or registered trademarks are the property of their respective owners.

Copyright 2011 Infogix, Inc. All rights reserved.

Confidential—Limited distribution to authorized persons only, pursuant to the terms of Infogix, Inc. (FKA Unitech Systems, Inc.) license agreement. This document is protected as an unpublished work and constitutes a trade secret of Infogix, Inc.

This document and the information contained herein are the property of Infogix, Inc. Reproduction or use in whole or in part of this document and the information contained herein by anyone without prior written consent of Infogix, Inc. is prohibited.

Publication Number 1903

Publication Date 6/6/11

---

# Contents

<b>Chapter 1</b>	<b>Introduction</b>	
	Organization of Information .....	5
	Contacting Infogix Customer Support.....	5
<b>Chapter 2</b>	<b>Input Types</b>	
	Using an External Translation Table with DB2.....	7
	Extracting Data from Two DB2 Tables .....	8
	Selecting Data Records from a Delimited File .....	9
<b>Chapter 3</b>	<b>Extractions</b>	
	Specifying Sections of an Input Source .....	11
	Specifying Records to Extract.....	12
	Determining Whether to Reconcile Data .....	14
<b>Chapter 4</b>	<b>Controls Processing</b>	
	Extending the Number of Internal Items .....	17
	Storing History for a Certain Number of Runs .....	19
	Matching Reconciliation Transactions from Two Input Sources .....	20
	Calculating Report Date Against Current Date.....	23
<b>Chapter 5</b>	<b>Reporting</b>	
	Formatting Free-form Report Headers .....	25
	Specifying Order of Items in a Report.....	26
	Printing a Report When No Data Was Extracted .....	27
	Printing a Qualifier Level Report.....	28

## Contents

---

---

## Introduction

This guide is a quick reference for common business solutions.

This guide does not provide start-to-finish instructions for rules writers, but rather, tips and techniques for some common rules-writing scenarios. The intent is to simplify the rule-writing process in situations where the solution to a problem may not be obvious.

If you have an unusual rule-writing problem, check the information in this guide to see if it addresses your problem. Conversely, if you have found a resolution for an unusual rule-writing problem, please consider sharing that information with Infogix Support for inclusion in this guide. See “Contacting Infogix Customer Support” on page 5.

## Organization of Information

This guide contains the following chapters:

Chapter 1, “Introduction” describes the components of the guide and provides the information to contact Infogix Customer Support.

Chapter 2, “Input Types” provides solutions for problems that pertain to specific input types.

Chapter 3, “Extractions” provides solutions for problems related to data extraction.

Chapter 4, “Controls Processing” provides solutions for problems that pertain to controls processing.

Chapter 5, “Reporting” provides solutions for problems that pertain to reports.

## Contacting Infogix Customer Support

To contact Infogix Customer Support, use the following contact information:

U.S. and Canada: Call us at (630) 505-1890 or send a fax to (630) 505-1883. You can also send e-mail to [support@infogix.com](mailto:support@infogix.com).

Outside the U.S. and Canada: Infogix maintains offices around the world. Check our Web site at [www.infogix.com](http://www.infogix.com) for the closest Infogix office or e-mail us at [support@infogix.com](mailto:support@infogix.com).

## 1 ■ Introduction

---

### *Contacting Infogix Customer Support*

---

## Input Types

This chapter provides solutions for problems that pertain to specific input types. This chapter contains the following topics:

- "Using an External Translation Table with DB2"
- "Extracting Data from Two DB2 Tables"
- "Selecting Data Records from a Delimited File"

### Using an External Translation Table with DB2

External translation tables allow you to match data from input sources where the data is in different styles, for example, IL and Illinois. You can also use an external table to obtain additional data that is not available in the input source, for example, a department name, where the input source contains only the department number.

ACR/Summary and ACR/Detail allow the creation of an external translation table for the following dataset formats: Physical Sequential, VSAM Key Sequenced, User Program Accessed File, Constant Translation Data, and Web Page.

#### Problem

The information that needs to be extracted requires translation. The data resides in a DB2 database.

#### Solution

Choose from the following possible solutions:

- Run an extract against the DB2 database and use the sequential result as the translation table. To run an extract, see your DB2 Administrator.
- In ACR/Detail or AC/Summary, write a user program to read the DB2 data and use the User Program option to read and translate the data. For information about user programs, see the *ACR User Program Guide*.
- Run an ACR/Detail job that extracts the DB2 data through an input source, and writes the data to a User Report or Freeform Output file (UNIDAT2), and then use the User Report or UNIDAT2 file as the translation table in a subsequent step. See the *ACR/Detail User Guide* for more information about User Reports or UNIDAT2 files.

## 2 ■ Input Types

### Extracting Data from Two DB2 Tables

---

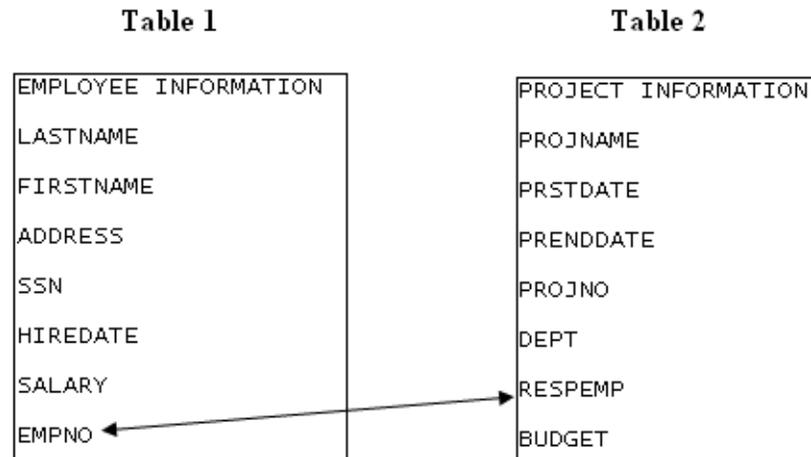
## Extracting Data from Two DB2 Tables

Typically, data is extracted from a single input source.

### Problem

You need to extract data from two DB2 tables.

*Example*



A company uses two DB2 tables to store human resource information. One table contains employee information and the second table contains project information. The tables have at least one column of data in common. A user wants to create a single report that displays columns from both tables.

From the first table, the report will include the following columns:

- LASTNAME
- FIRSTNAME
- HIREDATE
- SALARY

From the second table, the report will include the following columns:

- PROJNAME
- PRSTDATE
- PROJNO

The first table has a column, EMPNO, which contains the employee number. The second table has a column, RESPEMP, which contains the employee number for the person responsible for a particular project. This information is the key that ties the two tables together.

## Solution

To extract the data needed from both tables, join the tables.

On the File Organization dialog box, select the Use Freeform SQL Statements check box.

1. Set up a SELECT statement to show the columns to extract.

```
SELECT LASTNAME, FIRSTNAME, HIREDATE, SALARY, PROJNAME,
PRSTDATE, PROJNO
```

2. For z/OS, complete the following task. For Windows, complete this task after you set up the WHERE statement in step 4.

Assign these fields to keys or internal items.

For ACR/Detail, list the keys before the internal items.

3. Set up a FROM statement to show both tables.

```
FROM EMPLOYEE, PROJECT
```

4. Set up a WHERE statement to show the correlation of data in the two tables:

```
WHERE RESPEMP = EMPNO
```

## Selecting Data Records from a Delimited File

Infogix ACR user programs, such as Delimited Field Access (UUPDLIM), revise the way an input source is processed. UUPDLIM extracts each delimited field as its own record, making it possible to use a static position and length to extract data from a delimited file.

## Problem

You need to extract data from a delimited file, such as a trade file that contains data delimited with commas. You do not want to process the header records in the file. Or you may want to begin the data extraction with a particular record.

## Solution

Use the Delimited Field Access user program (UUPDLIM) to extract data from delimited files. See the *ACR User Program Guide*. Specify the Sequence ID parameter (/s) to put the sequence numbers on the record (0000000100000001, 0000000100000002, and so on). The first eight characters of the sequence represent the record number; the second eight characters represent the field number on the record.

## 2 ■ Input Types

---

### *Selecting Data Records from a Delimited File*

Complete the following steps to extract only the required information:

1. To begin processing the data records, but not the header record, in a file with one header record, set up the following **SELECT** condition:  
`SELECT if position 1 for a length of 8 is not equal to 00000001`  
This parameter will not extract the data that appears in the header record.

Adjust the statement to accommodate a file with multiple header records or to skip some records and begin processing with a record that appears later in the file.

2. Select the fields you want to extract, based on their field number.  
`SELECT if position 1 for a length of 8 is not equal to 00000001`  
`SELECT if position 9 for a length of 8 is not equal to`  
`00000004`

The example above selects field 4, which is not in the header record.

---

## Extractions

This chapter provides solutions for problems related to data extraction. This chapter contains the following topics:

- "Specifying Sections of an Input Source"
- "Specifying Records to Extract"
- "Determining Whether to Reconcile Data"

### Specifying Sections of an Input Source

In ACR/Detail, access mode 6, Keys Precede Detail Values, evaluates every record in the input source by selection group, starting with the first group.

#### Problem

You have a large file that contains several concatenated reports. Each concatenated report has the word *Report* in positions 1 through 6. You need to extract data from a specific report, GL-135, but the selection criteria, CREDIT, exists on other reports, as well. As you add the values for the CREDIT selection criteria, you want to include only the CREDIT values from the relevant report. In other words, you want to skip the other reports and process only the relevant report.

#### Solution

Complete the steps below to set up selection criteria to show where to begin referencing the data.

1. Set up a literal in an extraction variable and set the value to Y when you reach the beginning of report GL-135. For the same extraction variable, set the literal to N to indicate the end of report GL-135.

This step ensures that only the desired report is processed.

```
SELECT if Pos 1 through 6 = "REPORT"
      SELECT if Pos 8 through 6 = GL-135
            Move literal "Y" to MyReportExtractionVariable
      SELECT if Pos 8 through 6 NOT = GL-135
            Move literal "N" to MyReportExtractionVariable
```

## 3 ■ Extractions

### *Specifying Records to Extract*

---

2. Add a SELECT statement for all criteria that pertain to the data on your report to make sure that you have the right report.

```
SELECT if MyReportExtractionVariable = "Y" and SELECT if Pos 5  
through 10 = "CREDIT"
```

3. Specify a detail field to extract the values from the correct report.

## Specifying Records to Extract

When you specify a driver file, the system determines what to extract from other files based on the keys in the driver file. However, sometimes the values that you want to extract are not associated with a key field.

For example, you have one file that contains only data that you need to reconcile. You have another file with data that you need to reconcile plus additional data that you do not want to extract at this time. The field value that determines whether to include the data is not a value that you want to use as a key for this job, so you cannot use the driver file option.

### Problem

You want to extract only the values that match the values from another input source. For example, you have a corporate report that contains information for every branch within the company. You need to extract only the information that is pertinent to your department's branches but you do not want to sort the data by branch number, so you are not including the branch number in the key.

### Solution

Complete the tasks below to specify the records to extract. You will need to set up an external transaction table and define the input source.

#### Set Up a Translation Table

1. Set up one file as an external translation table.
2. Create a column in the external translation table for the value you want to match, for example, the branch number, and for any other values that you want to extract.
3. Assign a literal value of YES to an extraction variable when the columns match, and assign the values that you want to extract to other extraction variables.

### Set Up an Input Source

1. Set up the other file as the input source.
2. Assign a literal value of NO to an extraction variable named BRANCH FOUND.
3. Set up a detail field extraction that will extract the item from the input source, translate the item, and place the item in an extraction variable named MY FIELD.
4. Set up a SELECT statement to check the extraction variable BRANCH FOUND, and create subsequent keys and detail fields to move the values to internal items if BRANCH FOUND = YES.

When you extract data from the input source, the system translates one value at a time, checking the BRANCH FOUND extraction variable to see if the branch is found in both files.

If the branch is found in both files, the flag is set to YES and other data that you specified in external translation table columns is extracted into the extraction variables that you set up. See [step 3](#) under "[Set Up a Translation Table](#)." These extraction variables are then available in the input source to place into keys or internal items or for use in subsequent select statements and detail fields.

If the branch is not found, the flag value remains NO and the selection criteria fails. See [step 4](#), above. Processing continues without this item, skipping those records for which the branch number does not match between the two files.

## Determining Whether to Reconcile Data

In ACR/Detail, you can specify that the first file processed is the driver file. The system determines whether the keys from the driver file are present in the second file. If the keys from the first file are also present in the second file, the system processes only those keys. Keys in the second file that are not in the driver file are ignored.

### Problem

You need to reconcile two input sources using a 24-byte key, but only for accounts that reside in both input sources. The account number is in the first eight bytes of the key. Because you are matching only a portion of the key, the system cannot use the driver file feature to select the appropriate keys from the second file.

### Solution

Set up two numeric internal items to indicate which files contain the values. When you extract the value from the file, set the item that indicates the original input source for this value to +1.

Next, create reconciliation rules to look for non-zero values for both sources. If both values are non-zero, the account is present in both sources. The extracted values are passed via filter to the next level, which applies the true business rules that use the required 24-byte key.

Complete the steps below to reconcile the two input sources with a 24-byte key.

1. Create a base job that extracts the full 24-byte key but reconcile the job as a base level job with an 8-byte key.
2. Set the filter for a qualifier to ON, so that only the items that you note as out-of-balance are passed to a qualifier.  
See the “Setting Up Multi-Level Reconciliation” section in the *ACR/Detail User Guide*.
3. For the first input source, define an internal item that contains a tally to indicate whether the value is found in the first input source.
4. For the second input source, define an additional internal item with a separate tally to indicate whether the value is found in the second input source.

5. Create three reconciliation rules:
  - a. Create a standard rule to check the first input source internal item for a value greater than zero.
  - b. Create another standard rule to check the second input source internal item for a value greater than zero.
  - c. Create a conditional rule to check that both of the standard rules are true. If true, set a return code to flag which keys go to the RLQ.
6. Set up a qualifier job. This job has the same job/step name as the base job but has a value other than spaces in the qualifier field.
7. For the qualifier job, go into the reconciliation level and specify that the job uses the data filter from the base job, which you set up in [step 1](#), with the return code range appropriate for the value that you set up in [step 5c](#).

You can now reconcile with a 24-byte key because your qualifier job has only the data with matching accounts from both files, and all 24 bytes are available.

### 3 ■ Extractions

---

#### *Determining Whether to Reconcile Data*

---

## Controls Processing

This chapter provides solutions for problems that pertain to controls processing. This chapter contains the following topics:

- "Extending the Number of Internal Items"
- "Storing History for a Certain Number of Runs"
- "Matching Reconciliation Transactions from Two Input Sources"
- "Calculating Report Date Against Current Date"

### Extending the Number of Internal Items

ACR/Summary uses internal items to store control values extracted from a balancing job or control values that result from the manipulation of item values in a calculated item. For releases prior to 4.0, a job can have up to 100 internal items. Beginning with release 4.0, the system allows extended internal items, which can store lengthy extracted values. In addition to the 100 internal items, a job can have up to 100 extended internal items.

#### Problem

You need more than the 100 internal items allowed by the system.

#### Solution

Three possible solutions can effectively extend the number of internal items allowed.

##### Use Extended Internal Items

For releases 4.0 and above, a user can use extended internal items to hold smaller values as well as lengthy values. This increases the available number of internal items to 200.

##### Use Qualifiers

A qualifier represents multiple invocations of the same job/step combination or multiple processing of the same input file during a single balancing run. After the initial job/step runs, the system runs the qualifier or qualifiers. The addition of a qualifier can give you an additional set of internal items.

## 4 ■ Controls Processing

---

### *Extending the Number of Internal Items*

#### *Example*

If you add a qualifier of 01 to a job step called MYJOB/MYSTEP, you have two sets of rules in your definition database:

- MYJOB MYSTEP
- MYJOB MYSTEP 01

When you execute the job in batch mode, the system executes both the *blank* qualifier and the 01 qualifier. Thus, you have 100 internal items available for each set of rules.

For more information about qualifiers, see the *ACR/Summary User Guide*.

#### **Use ACR/Detail**

You can use ACR/Detail to assign keys or to use static keys. For more information about using keys, see the *ACR/Detail User Guide*.

#### *Assign Keys*

Use this solution if you need more than 100 items in ACR/Summary to repeatedly extract a similar value for different areas. For example, you need to extract counts and amount total values for each company department. In ACR/Detail, each key has its own set of internal items. Therefore, each key can have its own unique count and amount total. You can use the different departments for the key fields, then extract the count and amount total one time for each key.

#### *Use a Static Key*

If you do not repeatedly extract a similar value for different areas, but you can extract with access mode 6 or 7, you can still use this solution in ACR/Detail. Set a key to a literal value and extract as you would in ACR/Summary. In ACR/Detail you can use up to 999 internal items for each reconciliation job.

## Storing History for a Certain Number of Runs

In a suspense processing job, where you can match off items over time, it is important to build in a mechanism to age off histories. This can be implemented by date, either internal item date or cycle number, but there are times when it is important to match off items for a specified number of runs.

### Problem

You need to store history to use for matching for 30 runs. Due to weekends and holidays, 30 runs span more than 30 days.

### Solution

Several solutions are possible:

- Use the Suspense Aging Report to see the actual age of the history items, and then manipulate the items in ACR/TransMatch to obtain the desired data.
- Periodically run a job to delete history for items older than 30 days or whatever time frame you want.
- Use the @DATE function to save the current date as an internal item for the original job. With each run, subtract the original date from the current date and delete the item if it is older than a certain number of days, for example, if it is older than 30 days.
- Create a run counter for a key, which allows you to use rules to delete or maintain history when normal history roll-off or roll-off by the number of days will not work.

Complete the following steps to set up a run counter and store history for 30 runs:

1. Set up an input source and when you initially extract a key, force a value of zero (0) into an internal item that is the count of days the item has been in history (I-001).
2. Create a calculated item that will be stored as the internal item that will add 1 to the original internal item that is the count of days the item had been in history (I-001 + 1).
3. Set up your history file as an input source.
4. Set up the rules to check the number of days and delete the key value if the value in the calculation equals 30, that is, the number of runs for which you want to store history.

# Matching Reconciliation Transactions from Two Input Sources

In some instances of reconciliation processing, multiple transactions can have the same key combinations. For example, ATM transactions posted to your account may include the account number, ATM terminal number, and date as the key. Multiple transactions on the same day would have the same key and you want to make sure that each transaction posts individually.

Typical ACR/Detail processing combines transactions with the same key. You want to verify that you have the correct number of transactions, so you do not want to combine the transactions when you match them to the general ledger.

## Problem

You need to match data between input sources where there are multiple occurrences of the same key data that needs to be matched individually.

## Solution

You need to write rules to determine if the key values are the same from one record to the next in each input file. If the key values are the same, increment a tally that starts with 1 to keep each occurrence separate. If the key values are different, reset the tally back to 1. This solution will match items in order of appearance, so if an item in the middle of a sequence is missing, all subsequent items will not match. This solution does not indicate which item is the actual item that did not match, but it narrows the field of mismatches and starts the errors with the missing item.

### Putting Items in Order

This example starts numbering each key with a 1, so that transaction 1 for your ATM key on your first input source can be matched to transaction 1 for your ATM key on your second input source. To begin the numbers with 1 for each input source, the data in the input sources must have the keys sorted together.

If the keys are not together in your input source, you will need to put them together on the file, so that each transaction within the keys can be numbered in sequential order, beginning with 1.

**Matching Reconciliation Transactions from Two Input Sources**

Complete the steps below to run an ACR/Detail job, for each input source, to put the keys in the correct order. If your keys are already together on your file, proceed to the "Matching the Items" section, below.

1. Add +1 to an extraction variable named MyInitialTallyEV.
2. Extract the ATM key fields and also use MyInitialTallyEV as a key. For this example, you need to extract the following key values:
  - Account Number
  - ATM Terminal Number
  - Date
  - MyInitialTallyEV
3. Extract your remaining record in internal or extended internal items, as needed, to maintain your input values. For example,
 

```
Move pos 1 to 80 to X-001
Move pos 81 to 160 to X-002
```
4. Create a free-form output file that has your initial record value items as the outputs, such as X-001, and then X-002. See "Setting Up an Output File" in the *ACR/Detail User Guide*. The free-form output file will then replace your input source as the input to the actual reconciliation step. The transactions are the same as the original transactions, except that they are in order by the keys that you specified, Account Number, ATM Terminal Number, and Date.

**Matching the Items**

After you have determined that the items are in order, set up extraction definitions for selection groups in the following order:

1. Set up the first selection group to see if values in extraction variables are equal to values of the key fields on the input source.

```
SELECT if pos 1 for 8 = MyLastAcctNumEV AND
SELECT if pos 9 for 8 = MyLastATMTermEV AND
SELECT if pos 18 for 8 = MyLastDateEV
  Move literal of "MATCH" to MyEVIndicator
  Add +1 to MyEVTally
```

If the values in the first selection group match, place a literal value of MATCH into an extraction variable, MyEVIndicator. This extraction variable is used to keep track of the match. Adding 1 to MyEVTally creates a unique key for transactions that would otherwise have the same key.

## 4 ■ Controls Processing

### *Matching Reconciliation Transactions from Two Input Sources*

---

2. Set up the second selection group to see if values in extraction variables are not equal to values of the key fields on the input source.

```
SELECT if pos 1 for 8 NOT = MyLastAcctNumEV OR
SELECT if pos 9 for 8 NOT = MyLastATMTermEV OR
SELECT if pos 18 for 8 NOT = MyLastDateEV
    Move literal of "NONMATCH" to MyEVIndicator
    Move pos 1 for 8 to MyLastAcctNumEV
    Move pos 9 for 8 to MyLastATMTermEV
    Move pos 18 for 8 to MyLastDateEV
    Move literal of +1 to MyEVTally
```

If the values in the first selection group do not match, place the key values into extraction variables and a literal value of NONMATCH into the extraction variable, MyEVIndicator. Moving 1 to MyEVTally creates a unique key and starts a new sequence for transactions that would otherwise have the same key. Identifying mismatches between input sources is easier because each key sequence starts at 1.

3. Set up a selection group to extract your data, using the MyEVTally item to pick up the keys and internal items.

```
SELECT if pos 1 for 3 = ATM
where ATM is your business selection criteria
    Move MyLastAcctNumEV to Key1
    Move MyLastATMTermEV to Key2
    Move MyLastATMTermEV to Key3
    Move MyEVTally to Key4
    Move pos 32 for 15 into Internal Item 1
```

where 32 for 15 is the data that you need to extract and Internal Item 1 is the location where you want to place the extracted data.

## Calculating Report Date Against Current Date

Infogix ACR stores dates as a Date Count. Sometimes, a customer needs to check dates on an input source or a report to determine if the input data is current.

### Problem

You want to determine the number of days between a date on a report and the current system date. For example, you want to make sure that a record is less than one year old or that a posted transaction date is less than one day old.

### Solution

Infogix ACR stores both the current date and the report date as date counts. Follow the steps below to calculate the report date against the current date.

1. To obtain the report date, create a detail field to extract the report date, and specify the date format, such as MMDDYY, in an internal item defined as a date format.

The target detail field must specify the field type and format and must contain the Last Date and the Last Date-Count Value.

2. To obtain the current date, set up a calculated item and extract the date using the following equation:

```
+99999999 >(into) + @DATE
```

3. Create another calculated item to subtract the internal item report date from the calculated current date.

```
CURRENT DATE -REPORT DATE
```

4. Define a rule, formatted as a count, to analyze the calculated item with the difference in the number of days.

## 4 ■ Controls Processing

---

### *Calculating Report Date Against Current Date*

---

## Reporting

This chapter provides solutions for problems that pertain to reports. This chapter contains the following topics:

- "Formatting Free-form Report Headers"
- "Specifying Order of Items in a Report"
- "Printing a Report When No Data Was Extracted"
- "Printing a Qualifier Level Report"

### Formatting Free-form Report Headers

The Free-form Report is a completely customizable report. The user decides the content as well as the formatting.

In ACR/Detail, content is typically printed in its entirety for each key, with one key per page. Multiple keys can be printed on a single page by removing the header break character.

The Free-form Report repeats every line of data for each key. When printing multiple keys on the same page, a user can choose to have data appear as a report or page heading.

#### Problem

The heading repeats for each line item in a Free-form Report generated within ACR/Detail. You want the report to print with the title printed only once, at the top of the report. You want to print multiple keys on a single page but you want the page number and date to appear at the top of the page only, and not repeat with each key throughout the page.

#### Solution

Follow the steps that are appropriate for your version of ACR/Detail.

##### Graphical User Interface

For Windows, UNIX, or the ACR/Detail Workbench, complete these steps to insert report and page headers.

1. Use the Report Header Marker to insert a break after the report title and after each report header. Lines not marked as headers, including blank lines, will repeat for each key.
  - a. Click the report header icon (\R).



## Solution

Review the input source to determine the order of the items and then determine the order in which you want the items to appear in your report.

- If the input file lists the items in the desired order, choose unsorted for the key sort order.
- If the input file does not list the items in the desired order, choose ascending order and use a number as a literal in the first part of the key to manually order the items.

## Printing a Report When No Data Was Extracted

In releases prior to Infogix ACR 4.2, the system did not print a report unless an extraction occurred.

## Problem

In ACR/Detail 4.0 or earlier, you want to generate a report even though the rules did not result in a data extraction.

## Solution

Complete the steps below to print a report even if no valid data is extracted by creating a key for every run.

1. Create a dummy key and move +0 to an extraction variable.  
Move +0 to MyEmptyFileExtractionVariable
2. Create a SELECT statement that checks to see if valid data exists. If data is selected, pick up the keys and items and add +1 to the extraction variable.  
Select Pos 1 for 8 = "myData"  
Move +1 to MyEmptyFileExtractionVariable
3. If nothing is selected, the empty file extraction variable remains at zero. Then move a literal of DUMMY to the key.  
Select if MyEmptyFileExtractionVariable = +0  
Move literal "DUMMY" to Key1  
Move +1 to Internal Item 001
4. Add a rule that sets the override to avoid writing the DUMMY key to the history file.

## Printing a Qualifier Level Report

At the reconciliation level, you can create a data filter that stores out-of-balance keys for use in a subsequent qualifier report. In a multi-level job, it is important to make sure that the keys passed to the qualifier include only the keys you want on the report by setting return codes. This ensures that reports from subsequent levels are correct.

### Problem

In ACR/Detail, you want to create a data filter and print a qualifier level report whether or not keys exist at the qualifier level.

### Solution

Code the extraction definitions to extract a dummy key with dummy internal items that will meet the criteria to be put into the data filter, even if no other keys meet that criteria.

Complete the steps below to print a qualifier report even if no valid key is extracted.

1. Set up a base job and set the filter for a qualifier to ON, so that only the items that you note as out-of-balance are passed to a qualifier. See the “Setting Up Multi-Level Reconciliation” section in the *ACR/Detail User Guide*.
2. On the base level job, create a dummy key and move a literal of DUMMY to the key.  

```
Move literal "DUMMY" to Key1  
Move +1 to Internal Item 001
```
3. Add a rule that sets the dummy key value out of balance. Specify a return code that will pass the filter.  

```
If Internal Item 001 = +1  
    Set return code to 222
```
4. Add your regular business rules to determine if your key values are accurate. Specify a return code to pass any desired key to the qualifier reporting at that level.
5. Set up a qualifier job. This job has the same job/step name as the base job but has a value other than spaces in the qualifier field.
6. For the qualifier job, go into the reconciliation level and specify that the job uses the data filter from the base job, which you set up in step 1, with the return code range appropriate for the rules that you set up in steps 3 and 4.

At this point, you have the keys, if any, that you want to include in the qualifier report as well as the dummy key.

7. On the qualifier job, set up an active rule that sets the dummy key out of balance.

```
If Internal Item 001 = +1  
    Set return code to 222
```

8. Set the qualifier report to print only the keys that are in balance. This step eliminates the dummy key from printing on the qualifier report. Be sure to set the report to always print with headings, so that the report will print even if there are no keys.

In the instructions above, step 2 creates a key that will always pass a value to the qualifier, which ensures that the qualifier job will run. Because the qualifier job always runs, a qualifier report is always printed. These instructions avoid printing the dummy key by setting the dummy key out of balance and setting the report to print only keys that are in balance.

If you have other balancing rules and want to print only the out-of-balance keys, reverse the logic: mark the dummy key as in balance and choose to print only out-of-balance keys on the report.

## 5 ■ Reporting

---

### *Printing a Qualifier Level Report*