

precisely

ACR/Detail

User Guide

For Windows, UNIX, and
Linux



Notices



Trademarks

Infogix, the Infogix logo, ACR, ACR/Detail, ACR/Summary, ACR/Workbench, ACR/Connector, Infogix Assure, Infogix Insight, ACR/Instream, ACR/File, Infogix ER, Infogix Perceive, Data3Sixty, and Data360 are registered trademarks of Precisely. Data3Sixty Analyze, Data3Sixty Govern, Data3Sixty DQ+, Data360 Analyze, Data360 Govern and Data360 DQ+ are trademarks of Precisely. Any other trademarks or registered trademarks are the property of their respective owners.

1700 District Ave Ste 300
Burlington MA 01803-5231
USA

www.precisely.com

Copyright 2010, 2022 Precisely

Contents

Preface

About This Guide	13
Overview of ACR/Detail	16
Ways to Use ACR/Detail	16
Configuration Modes	17
Input Source Types Supported	17
ACR/Detail Processing Flow	18
ACR/Detail Components	19
ACR/Detail Client File Types	21
Database and File Processing Diagram	23
Contacting Customer Support	23

Chapter 1 Preparing to Use ACR/Detail Client

Accessing and Exiting from ACR/Detail Client	25
Using the ACR/Detail Client Interface.....	26
Configuring ACR/Detail Client	34
UNIX or Linux Only: Modifications for RDBMS Access	36
Initializing the Definition and History Databases	37
Downloading, Uploading, and Importing Definitions	37
Setting the User Options	38
Analyzing Your Reconciliation Needs	38

Chapter 2 Tutorial

Preparing for the Tutorial	41
Determining Your Reconciliation Requirements	42
Creating the Control Entity and Job.....	45
Running the Job	56
Viewing the Output Reports and Handling Errors.....	57

Chapter 3 Setting Up a Job

What Is a Reconciliation Job? 59
What is a Control Entity? 60
Preparing to Set up a Reconciliation Job 64
Setting Up and Running a Job..... 64
Viewing Reports and Handling Errors 68
Working with Job Definitions 69

Chapter 4 Setting Up Basic Job Information

Completing the Basic (Job) Information Options 74
Name 74
Control Report Title..... 74
Store History 74
Set Return Code 75
Print Control Report..... 75
Print Recap Report 75
Print Extracted Data Detail Report 75
Print Sorted/Accumulated Data Detail Report 76
Print History Data Detail Report 76
Print Free-Form Report 77
Write Output File..... 77
Auto Cycle Indicator 77
Round Results 78
Reconciliation Level..... 78
Use 22-Character Numeric Format 78
Initialize Text Items to Spaces 78
Control Report XML Option 78
Key Sort Order..... 79
Adding Job Comments 79
Display Processing Status..... 79
Out-of-balance Key Tolerance 79

Chapter 5 Setting Up an Output File

Choice of Output File Formats..... 81
Specifying Where the File Will Be Generated 83

Setting Up an Output File 83
 Record Layout for Original Output File 84

Chapter 6 Setting Up Job Definitions

Understanding Internal Items and Extended Internal Items 88
 Defining Internal Items..... 90
 Defining Extended Internal Items 90
 Defining Extraction Variables 91
 Defining History Items 94
 Defining Calculated Items..... 95
 Defining Rules 97
 Defining Messages 106
 Using Batch Options to Control Item Numbers..... 107
 Using Key Masks 109
 Constraints for Single-precision and Double-precision Numbers 113

Chapter 7 Setting Up Input Sources

What Is an Input Source? 115
 What Is a Reconciliation Key? 116
 Preparing to Set Up an Input Source..... 117
 Procedure for Setting Up an Input Source..... 118
 History Database Record Layout..... 124
 Selection Fields and Selection Groups..... 126
 Selecting an Access Mode 130
 Understanding Detail Field Processing 137
 Reformat Field Processing 144
 Understanding Field Formats 146
 Setting Up an Input Source for Relational Data..... 152
 Date and Cycle Number Extraction and Processing 158
 Time Extraction and Processing 161
 Deleting File Definitions 162
 Extracting Directly from a Delimited Data File 162

Chapter 8 Setting the User Options

The User Options File (udsuf.dat) 173

	Setting User Options	173
	Setting User Options on UNIX or Linux	174
Chapter 9	Using Multi-Level and Suspense Reconciliation	
	Using Multi-Level Reconciliation.....	185
	Using Suspense Reconciliation	189
Chapter 10	Using Translation Tables	
	Internal Vs. External Translation Tables	193
	Processing of Internal and External Translation Tables.....	195
	Using External Translation Tables	196
	External Translation Table Trace Report	208
	Using Internal Translation Tables	215
	Using Dynamic Translation Tables	218
	Using Hash Translation for a Key Value	219
Chapter 11	Using History Analysis	
	Understanding History Analysis.....	221
	Creating and Generating History Analysis Report Definitions	222
	Creating and Running History Analysis Process Definitions	223
	Summary Level and Key Length Example	224
	History Analysis Report Samples	226
	Cycle Tables	227
Chapter 12	Using the Utilities	
	Running Utilities.....	230
	Database Initialization	230
	Database Update.....	230
	List Definitions	231
	Control Update Information Report Utility.....	231
	Build Batch Transactions	232
	Build Extraction Program.....	232
	Cross Reference File Definitions	232

List History	233
Initialize History	234
Copy History	234
Copy Selected Records	234
Unload History	234
Reload History	235
Delete History	235
Reorganize History	236
Extract History	236
Print Management Report	239
Initialize Translation Database	239
List (Dynamic) Translation Database	239
Reorganize (Dynamic) Translation Database	240
Print Suspense Aging Report	240
Update History	241
Create Comma Delimited Text File	241
Integrity Check	242
UNINTC and UNICTN	242

Chapter 13 Setting Up Free-Form Reports

What is a Free-Form Report?	243
Creating a Free-Form Report	243
Viewing the Internal Codes	251
Deleting a Free-Form Report	251
Sample Free-Form Report	252

Chapter 14 Setting Up User Reports

What is a User Report?	253
Setting Up a User Report	253
Deleting a User Report	255
User Report Sample	255

Chapter 15 Importing Definitions

Purposes of Importing	257
Procedure for Importing Definitions	257

Understanding @FILETMP Jobs 259
Using @@FILLER Records 259

Appendix A Using Batch Definition Transactions

What Is a Batch Definition Transaction?..... 261
Format of Batch Definition Transaction Records 262
Updating the Definition Database..... 264
Purging Definitions 265
..... 266

Appendix B Reports

Working with Reports in ACR/Detail Client..... 267
Reports Table 269
System Messages Report (SYSOUT.RPT) 271
Control Report 271
Recap Report 275
Extracted Data Detail Report 276
Sorted Data Detail Report 277
Accumulated Data Detail Report 278
History Data Detail Report..... 279
Trace Report..... 280
Database Initialization Report..... 285
Detail History Analysis Reports 286
Detail Reconciliation Summary Report..... 286
Database History and Statistics Report..... 287
Index of Definition Keys Report 287
Detailed Job Definitions Report..... 288
Detailed File Definitions Report 291
Detailed Table Definitions: Internal Translation Table 291
Detailed Table Definitions: External Translation Table..... 293
Detailed Exception Definitions (History Analysis) 294
Detailed Report Definitions - History Analysis Threshold and
Fluctuation Analysis) 295
Database History and Statistics Report..... 296
Index of Detail Histories Report..... 297
Summary of Detail Histories Report 297

History Data Detail Report.....	298
Control Update Information Report.....	299
Definition Database Update Report.....	299
Definition Database Cross Reference Report	301
Build ACRUPDT Report.....	302
Translation Database Listing Report	303
Suspense Aging Report.....	303
History Database Transfer Report	304
History Database Reload Report.....	305
Transaction Listing Report.....	306

Appendix C Using ACR/Workbench

Restrictions for Mainframe Users	307
ACR/Workbench Configuration Modes.....	308
Downloading and Uploading between z/OS and ACR/Workbench .	309
Importing Definitions and Input Sources from z/OS	311
ACR/Workbench for Detail Tutorial.....	313
Using QuickChange.....	314
Features Applicable Only to Jobs Uploaded to z/OS.....	316

Appendix D Maintaining the Definition Database

Create the .def File with the Purge Records.....	319
Edit and Run the Update Definition Database File or Script.....	320

Appendix E Using ACR/Detail for Windows in Batch

Conditionally Executing a Step Based on a Return Code	323
Before You Begin	324
Enable Proper Processing of Tab-Delimited Input Sources.....	324
Increase the Maximum file Size (Optional).....	325
Editing Batch Files.....	325
Batch File Quick-Reference with Page Numbers	326
udpcgen.bat - Build Extraction Program.....	327
udptest.bat - Execute Extraction Program Test	328
udpuser.bat - Test User Program.....	329
uds2000.bat - Run Reconciliation.....	331

uds6000.bat - History Analysis	334
udsdfb.bat - Build Batch Transactions	335
udsdf1.bat - List Definitions	335
udsdfha.bat - Suspense Aging.....	336
udshfc.bat - Copy History	336
udshfd.bat - Delete History	337
udshfe.bat - Extract History	337
udshfl.bat - List History	338
udshfr.bat - Reorganize History	338
udshft.bat - Transfer History (Selected Records)	339
udshftu.bat - Unload History	339
udshftr.bat - Reload History	340
udshfu.bat - Update History.....	341
udsinit.bat - Initialize Databases.....	342
udsmr01.bat - Management Reporting.....	342
udstfint.bat - Initialize Translation Database.....	343
udstfls.bat - List Translation Database	343
udsupdt.bat - Update Definition Database.....	344
udscsv.bat - Create Comma Delimited Text File.....	345
Migrating Test Definitions to Production	346

Appendix F Using ACR/Detail for UNIX or Linux in Batch

Conditionally Executing a Step Based on a Return Code	349
Before You Begin.....	351
Editing and Running the Scripts	352
Script Quick-Reference with Page Numbers	354
udpcgen.sh - Build Extraction Program	355
udpcomp.sh - Compile COBOL Source.....	356
udpctest.sh - Execute Extraction Program Test	356
udpuser.sh - Test User Program.....	357
uds2000.sh - Run Reconciliation.....	359
uds6000.sh - History Analysis	363
udsdfb.sh - Build Batch Transactions	364
udsdf1.sh - List Definitions.....	364
udsdfha.sh - Suspense Aging.....	365
udshfc.sh - Copy History	365

udshfd.sh - Delete History	366
udshfe.sh - Extract History	366
udshfl.sh - List History	367
udshfr.sh - Reorganize History	367
udshft.sh - Transfer History (Selected Records).....	368
udshftu.sh - Unload History	368
udshftr.sh - Reload History	369
udshfu.sh - Update History	370
udsinit.sh - Initialize Databases	370
udsmr01.sh - Management Reporting	371
udstfint.sh - Initialize Translation Database	372
udstfls.sh - List Translation Database.....	372
udsupdt.sh - Update Definition Database.....	373
udscsv.sh - Create Comma Delimited Text File	374
Migrating Test Definitions to Production	374

Appendix G Non-Biased Variance and Standard Deviation Computations

Field Descriptions and Formulas	377
Examples.....	379

Glossary	385
-----------------------	------------

Index	397
--------------------	------------

Preface

This preface provides an introduction to ACR/Detail® for Windows®, UNIX®, and Linux® and ACR/Detail Client. It contains the following sections:

- “About This Guide” on page 13
- “Overview of ACR/Detail” on page 16
- “Ways to Use ACR/Detail” on page 16
- “Configuration Modes” on page 17
- “Input Source Types Supported” on page 17
- “ACR/Detail Processing Flow” on page 18
- “ACR/Detail Components” on page 19
- “ACR/Detail Client File Types” on page 21
- “Database and File Processing Diagram” on page 23
- “Contacting Customer Support” on page 23

About This Guide

Purpose

This manual provides instructions on how to use ACR/Detail for the Windows, UNIX, and Linux platforms.

Audience

This manual will be useful to users of ACR/Detail for Windows or ACR/Detail for UNIX or Linux. If you have ACR/Workbench® for Detail, consult Appendix C, “Using ACR/Workbench” before reading the rest of the manual.

Other Documentation

For installation information, see the following manuals:

- *ACR/Summary and ACR/Detail Installation Guide for Windows* provides instructions for installing ACR/Summary and ACR/Detail Server, ACR/Detail Client, and ACR/Workbench for Detail.
- *ACR/Summary and ACR/Detail Installation Guide for UNIX and Linux* provides instructions on installing ACR/Detail for UNIX or Linux on the UNIX or Linux host, setting global environment variables, and configuring for access to ACR/Detail Client.

ACR/Detail Client Restrictions by Platform

Note: For ACR/Workbench for Detail restrictions, see [Appendix C](#), “Using ACR/Workbench” on page 307.

Restrictions for Windows

If you run definitions created through ACR/Detail Client on Windows, the following features will not apply:

- UNIX, Linux, and Local configuration modes

Restrictions for UNIX or Linux

If you run definitions created through ACR/Detail Client on UNIX or Linux, the following features will not apply:

- Windows and Local configuration modes
- E-mail notification
- Web Extraction
- Web Publishing

Components of this Guide

Following is a summary of the contents of each component of this guide:

- Remainder of this Preface—Provides overviews of ACR/Detail and ACR/Detail Client, including configuration, components, and processing.
- [Chapter 1, “Preparing to Use ACR/Detail Client”](#)—Provides instructions for preparing to use the product and analyzing your reconciliation needs.
- [Chapter 2, “Tutorial”](#)—Takes you through the process of setting up and running a reconciliation job.
- [Chapter 3, “Setting Up a Job”](#)—Provides the procedure for setting up and running a job.
- [Chapter 4, “Setting Up Basic Job Information”](#)—Provides an overview of the Basic Information options in Control Task (Job) View.
- [Chapter 5, “Setting Up an Output File”](#)—Explains a feature that can write results from a reconciliation job to a sequential file for use as input to other programs, such as PC applications, including spreadsheets.
- [Chapter 6, “Setting Up Job Definitions”](#)—Provides information on setting up job definitions and controlling item numbers.

- **Chapter 7, “Setting Up Input Sources”**—Provides procedures for defining input sources. Extraction variables are also explained.
- **Chapter 8, “Setting the User Options”**—Explains how to set system-wide defaults.
- **Chapter 9, “Using Multi-Level and Suspense Reconciliation”**—Documents two specialized types of reconciliation jobs.
- **Chapter 10, “Using Translation Tables”**—Describes how to set up and use translation tables.
- **Chapter 11, “Using History Analysis”**—Describes how to use history analysis.
- **Chapter 12, “Using the Utilities”**—Provides general instructions for accessing and running utilities and descriptions of each utility.
- **Chapter 13, “Setting Up Free-Form Reports”**—Explains how to create a Free-Form Report.
- **Chapter 14, “Setting Up User Reports”**—Explains how to create a User Report.
- **Chapter 15, “Importing Definitions”**—Explains how to import definitions from z/OS®, UNIX, and Linux.
- **Appendix A, “Using Batch Definition Transactions”**—Explains batch definition transaction records and provides layouts for each type of batch definition transaction.
- **Appendix B, “Reports”**—Provides information and samples of ACR/Detail reports.
- **Appendix C, “Using ACR/Workbench”**—Documents features specific to ACR/Workbench for Detail.
- **Appendix D, “Maintaining the Definition Database”**—Provides the procedure for purging unused definitions from the definition database.
- **Appendix E, “Using ACR/Detail for Windows in Batch”**—Provides information on using the ACR/Detail batch files.
- **Appendix F, “Using ACR/Detail for UNIX or Linux in Batch”**—Provides information on using the ACR/Detail scripts.
- **Appendix G, “Non-Biased Variance and Standard Deviation Computations”**—Explains and provides examples of the non-biased variance and standard deviation computation options available in history analysis.

This guide also contains a glossary and index.

Overview of ACR/Detail

ACR/Detail is a rule-based software solution that works in conjunction with your application system to reconcile detailed information. Its components are designed to integrate easily with your current applications with no programming changes.

ACR/Detail automates the process of reconciling the results of each processing job with the output of other jobs to ensure data entry and processing accuracy.

ACR/Detail can perform reconciliation at the level of detail you require. For example, you might want to control processing at the job level, step level, grand total level, subtotal level, and at multiple detail levels (for example, division/departments/employee). You can specify that detail reconciliation will be performed only when the application results are out of balance at a higher level.

ACR/Detail works as a stand-alone system or in conjunction with the Infogix ACR/Connector product. ACR/Detail is available for z/OS, Windows, UNIX, Linux, and IBM i.

Ways to Use ACR/Detail

Ways to Define ACR/Detail Jobs

ACR/Detail jobs can be defined in the following ways:

- Through ACR/Detail Client. ACR/Detail Client is a graphical interface residing on the PC that makes it easier to create, maintain, and run reconciliation and related processing.
- Through ACR/Workbench for Detail.
- Through the ACR/Detail for z/OS interface or the ACR/Detail for IBM i interface.
- After creating or editing the batch definition transaction records on another platform, you can run a batch file (on Windows) or a script (on UNIX or Linux) to update existing job, file, table, or history analysis definitions.

Ways to Run ACR/Detail Jobs

ACR/Detail jobs can be run in the following ways:

- Through ACR/Detail Client.
- By customizing ACR/Detail batch files (Windows) or scripts (UNIX or Linux). These can then be run individually or included in your batch job stream. For more information, see Appendix E, “Using ACR/Detail for Windows in Batch” and Appendix F, “Using ACR/Detail for UNIX or Linux in Batch”

Configuration Modes

The following configuration modes are available for ACR/Detail Client, depending on your licensing:

Windows Mode

To use Windows mode, ACR/Summary and ACR/Detail Server must be installed on a Windows-based server. Jobs are created, saved, and run on the server.

UNIX or Linux mode

UNIX or Linux mode provides a graphical interface for UNIX or Linux processing. ACR/Detail definitions are saved and run on a UNIX or Linux machine where ACR/Summary and ACR/Detail Server is installed. This mode uses true client-server technology, which uses TCP/IP for its connection.

Input Source Types Supported

The following types of input sources can be used in reconciliation jobs that are set up through ACR/Detail Client:

- Physical sequential files (ASCII text files)
- Data on lines or records of up to 10,000 bytes
- Data from any ODBC-compliant database for which you have the appropriate driver installed
- The current history database
- Data accessed via a user program

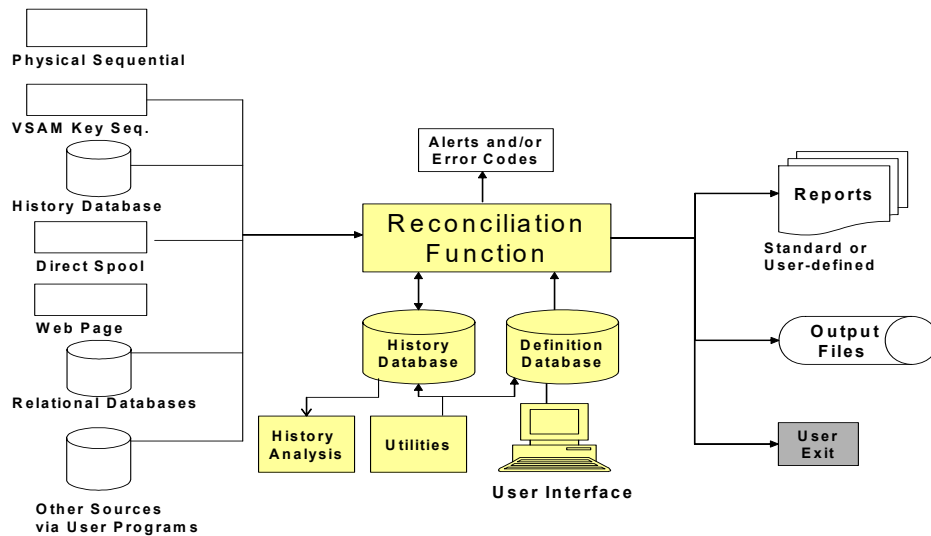
- Data from a Web page (applies in Windows mode)
- Data from Oracle® databases (applies only in UNIX or Linux mode)

ACR/Detail Processing Flow

The following figure shows an overview of ACR/Detail processing. The client interface refers to the graphical interface provided by ACR/Detail Client. The location of the components and files you will use to run ACR/Detail depends on the configuration mode you are using:

- When you are running in UNIX or Linux mode, only the client interface resides and runs on the PC. The other components and files used to run ACR/Detail reside on the UNIX or Linux box.
- When you are running in Windows mode, only the client interface resides and runs on the PC. The other components and files used to run ACR/Detail reside on the Windows server where ACR/Summary and ACR/Detail Server is installed.

The yellow shapes in the center represent the major system components, which are described in the next section.



ACR/Detail Components

User Interface

The graphical user interface provided by ACR/Detail Client makes it easier to create, maintain, and run reconciliation jobs. For an introduction to the interface, see “Using the ACR/Detail Client Interface” on page 26. To get some practice using the interface, see the “Tutorial” on page 41.

Definition and History Databases

Before you can use ACR/Detail Client, you need to initialize (create) the definition and history databases. Following are descriptions of these databases. For a diagram that shows database and file processing, see “Database and File Processing Diagram” on page 23.

Definition Database (Default Name `udsdf.dat`)

The definition database is a keyed physical file that stores the definitions (parameters) used in jobs that reconcile data and generate reports. Five types of definitions are used: job, file, table, history analysis process, and history analysis report definitions.

A definition database update applies the `.def` file containing the batch transactions to the definition database so that you can run the definitions. If you change any definitions and you do not have **Automatically update Definition Database on file saves** selected on the Add/Edit Configuration dialog box - File Information tab, you need to save the job and run the definition database update again.

A definition database update generates the following reports:

- A System Messages (SYSOUT) Report showing any errors that occurred and the step completion code, or the step completion code alone if there were no errors, and
- A Definition Database Update Report showing each transaction that was updated and any validation messages that were produced. See Appendix B, “Reports” for a sample.

History Database (Default Name udshf.dat)

This keyed physical file can be used to store extracted values so that they can be reused in a future run of the same job or in a different job.

If a job is set up to store history, when it runs, a record is created in the history database for each unique instance of the job's reconciliation key that is found. The fields on this record contain the values extracted for the key. If, for example, 20 unique instances of the job's reconciliation key are found, 20 records will be created in the history database.

History records are identified in the history database by a combination of reconciliation key and cycle (run) number. If a job runs later that has the same reconciliation key and cycle number, the original history record is overwritten.

Securing the Databases

Setting file permission appropriately is essential to securing your definition and history databases. At a minimum, we recommend that you protect your definition and history databases by allowing read-only access to the production databases for non-production jobs.

Assign one or more individuals (usually your change control administrators) to perform quality assurance and migration of definitions from test to production. Only these individuals should have read/write access to the production databases.

Reconciliation Function

Reconciliation is the process of comparing transaction-level data according to user-defined rules. An application system consists of one or more jobs in a processing stream. At any point along this stream, you can insert a reconciliation step that will check the data as it moves through the system. For most of your jobs, you can invoke the reconciliation function by including one additional step.

When you set up a reconciliation job, you can choose from a number of predefined or user-defined reports to let you know if the results are in or out of balance and to provide additional information to help you analyze the results and correct any problems.

History Analysis Function

The history analysis function enables you to extract, analyze, and report on selected data from the history database. It can perform the following tasks:

- Summarize historical data at multiple levels.
- Analyze values from corresponding periods.
- Locate missing or incomplete data.
- Highlight values falling outside an acceptable range.
- Discover unusual fluctuations in values.

Utilities

The utilities enable you to create, analyze, and maintain your ACR/Detail databases. For more information, see [Chapter 12, “Using the Utilities on page 229”](#).

ACR/Detail Client File Types

Control Entity File (.dob)

A .dob file (in addition to a .def file and .df2 file) is created when you save a control entity with at least one job. It is placed in the ACR/Detail default directory. It contains all definitions you created for the entity, including those created at the entity level and at the job level.

Batch Definition Transaction File (.def)

A .def file (in addition to a .dob file and .df2 file) is created when you save a control entity with at least one job. It is placed in the ACR/Detail default directory. A .def file contains all batch definition transaction records for the definitions you created and saved. The .def file is useful when you need to move the definitions between platforms.

.df2 File

A .df2 file (in addition to a .dob file and .def file) is created when you save a control entity with at least one job. It is placed in the ACR/Detail default directory. A .df2 file contains any information that is not stored as batch transactions such as key names or input source paths.

Database, User Options, and Control Files (.dat)

Definition Database (udsdf.dat)

The definition database (default name udsdf.dat) contains all of the job, file, table, and history analysis definitions that have been updated to the definition database.

History Database (udshf.dat)

The history database (default name udshf.dat) contains control values extracted as internal or extended internal items during reconciliation runs. These values can then be used in another run of the same job or a different job.

User Options File (udsuf.dat)

The User Options file (default name udsuf.dat) enables you to override the system defaults for formatting of dates, currency amounts, and negative numbers on reports. The file also controls some special processing options related to setting return codes and updating the history database. For more information, see [Chapter 8, “Setting the User Options on page 173”](#).

User Content File (udscf.dat)

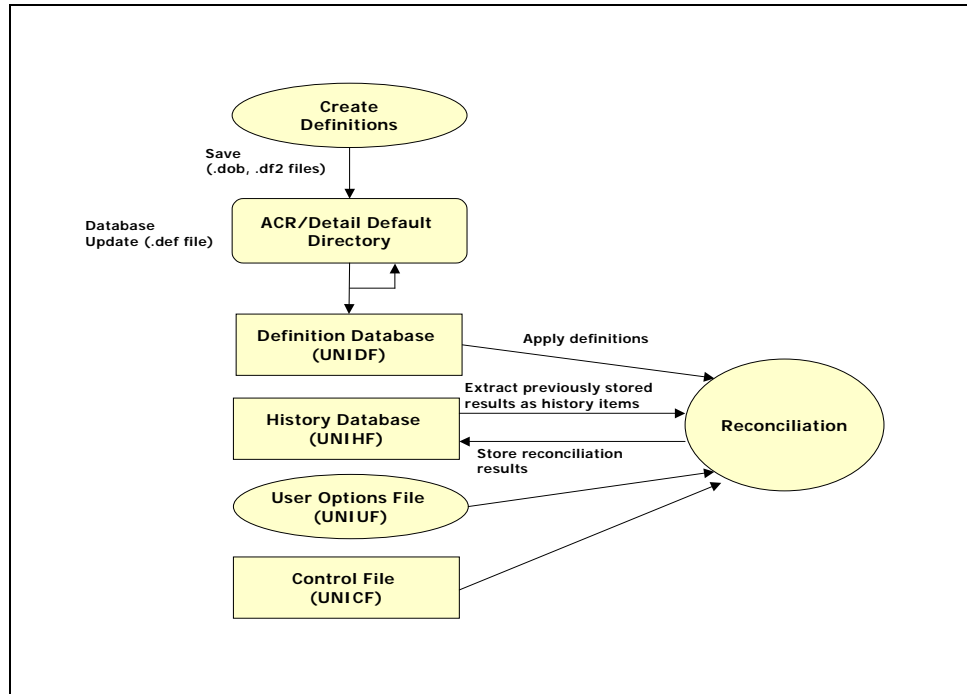
The User Content file (default name udscf.dat) resides on both the PC and the ACR/Summary and ACR/Detail Server and contains a record that begins with 'USERNAME='. You can edit this record to specify any name up to 40 characters in length on the right side of the equal sign (=). This name will then appear in the headings of all reports.

Licensing Control File (acrcntl.dat)

This file (default name acrcntl.dat) contains licensing information.

Database and File Processing Diagram

The following diagram shows how the files described in the previous section are processed in ACR/Workbench for Detail.



Contacting Customer Support

If you need assistance, contact Infogix Customer Support.

Support Phone: +1.630.505.1890

Support Email: support@infogix.com

Support Website: <http://support.infogix.com>

Fax Number: +1.630.505.1883

Visit our Website: www.infogix.com

Preface

Contacting Customer Support

Preparing to Use ACR/Detail Client

This chapter provides the procedures and interface information you will need to get ready to use ACR/Detail Client. It contains the following sections:

- “Accessing and Exiting from ACR/Detail Client” on page 25
- “Using the ACR/Detail Client Interface” on page 26
- “Configuring ACR/Detail Client” on page 34
- “UNIX or Linux Only: Modifications for RDBMS Access” on page 36
- “Initializing the Definition and History Databases” on page 37
- “Downloading, Uploading, and Importing Definitions” on page 37
- “Setting the User Options” on page 38
- “Analyzing Your Reconciliation Needs” on page 38

Accessing and Exiting from ACR/Detail Client

To access ACR/Detail Client, you can use any of these methods:

- Select **Start > Programs > Infogix > ACR/Detail (or ACR/Detail Client)**.
- Create a shortcut on your desktop to the executable file WDDDB32.exe in the installation folder. The default folder is C:\Infogix\Detail32.

You can exit an ACR/Detail Client session by using one of the following methods:

- Select **File > Exit**.
- Press **Ctrl + F4** from any view. If you have not saved your changes, you will be prompted to save them.

Using the ACR/Detail Client Interface

Interface Elements

The ACR/Detail Client interface provides the following elements to make it easier to use the product.

Title bar

The application title bar along the top of your screen displays the product name. A view title bar displays the name of the open view and the name of the object (control entity, control task, etc.) being displayed.

Views

These are windows that make it easy to accomplish a particular task. You can expand or collapse a view or section of a view using the commands on the View menu. If you select **Window > Cascade**, the open views will be arranged in an overlapping pattern with the currently active definition on top. For more information on views, see [“List of Views” on page 29](#).

When you start ACR/Detail Client, you will see a Control Entity view. All activities in ACR/Detail Client center around a control entity.

Folders

Each view has its own folders. You can click on 1) a folder, 2) an item you have defined in a folder (such as an internal item or a report option), or 3) the word **New** to open the corresponding dialog box or view. Help for each dialog box is available by pressing F1 or selecting the Help button. For more on working with folders, see [“Working with Folders” on page 28](#).

Menu bar

The menu bar provides access to ACR/Detail Client's pull-down menus. Each menu contains commands you can initiate, such as opening a file. Many of these commands can also be initiated by clicking on icons in the toolbar. To see the command associated with each icon, hover the mouse over it.

The following table provides a brief explanation of the menus.

Select this:	To do this:
File	Open and close files. Save your work. Import, upload, or download batch definition transactions. Print reports. Exit ACR/Detail Client.
Edit	Perform standard Windows cut and paste operations.
View	Display/hide the toolbar and status bar. Change the display font. Expand/collapse views.
<u>R</u> un	Run reconciliation. Run integrity check. Run database initialization. Run database update. Access the utilities. Run history analysis. Run quick change. Run product license report.
Reports	Open/close all on-line reports.
<u>O</u> ptions	Access the configuration options to change the current database files, report paths, and servers. Setup user options. Indicate how item numbers will display in batch records.
<u>W</u> indow	Perform standard Windows operations such as tile and cascade windows.
<u>H</u> elp	Access on-line help.

- **Toolbar icons**—These provide a fast alternative to the menu selections that are needed most often in a particular view.
- **Status bar**—Displays at the bottom of the view. This area shows messages describing the status of actions within ACR/Detail.

1 ■ Preparing to Use ACR/Detail Client

Using the ACR/Detail Client Interface

- **Dialog boxes**—These group the fields for a particular function. Most dialog boxes are available by clicking a folder option in the appropriate view. Others are accessed through menu options or toolbar icons. Navigation information for accessing each dialog box is included in the help for that dialog box. To obtain this information, search for the dialog box name in the Help. The dialog box help also provides the purpose of the dialog box, field definitions, and any other information that may be needed for completing the dialog box.

Working with Folders

Folders are of two types. One type of folder contains a list of options. The Basic (Job) Information folder in the Control Task (Job) View is an example of this type.

The other type of folder enables you to set up and edit items of a specific type. For example, the Input Sources folder enables you to set up input sources, and the Internal Items folder enables you to set up internal items. The word **New** indicates that you can define a new item for the folder.

To edit a folder option or create/edit a folder item, do one of the following:

- Click on the option, the item, or the word **New**.
- Use the arrow keys to select the option, the item, or the word **New** and press **Enter**.
- Use the arrow keys to select the option, the item, or the word **New**, click on **Edit** in the main menu, and choose **Open** or **New**.

List of Views

Following is a table showing the views. After you access the view, you can also get help by pressing the F1 key

Toolbar icons for each view are documented in the following sections.

View	Purpose	How to Access
Control Entity View	Allows you to create and save multiple reconciliation jobs (control tasks) that share some of the same components.	This is the default view when you open ACR/Detail. To open an existing entity: File > New
Control Task (Job) View	Create or modify a control task (reconciliation job).	File > New > Control Task (Job)
Input Source View	Define the file(s), reports and/or program(s) from which values will be extracted for use in your reconciliation job.	Control Task (Job) View > Input Sources > New
User Report View	Set up a (Reconciliation) User Report. This is a customized report limited that prints any or all of the data that can be included in the Control Report in the order that you specify.	Control Task (Job) View > Report Information > Reconciliation User Report
Free-Form Report View	Create a custom report that includes any of the control values obtained when the job is run and their description, There is no column limitation for this report.	Control Task (Job) View > Report Information > Free-Form Report
External Translation Table View	Set up a table that is built at run time, usually by referencing data in an existing file external to ACR/Detail. Unlike an internal translation table, an external translation table can translate the values to a different data type.	File > New > Table (External/Internal/Cycle) > External Translation Table
Internal Translation Table View	Set up a table to do simple one-to-one translations from one text value to another text value.	File > New > Table (External/Internal/Cycle) > Internal Translation Table

1 ■ Preparing to Use ACR/Detail Client

Using the ACR/Detail Client Interface






View	Purpose	How to Access
Cycle Table View	Set up a table to be used in reconciliation or in history analysis to verify whether the cycle numbers being processed are valid	File > New > Table (External/Internal/Cycle) > Cycle Table
History Analysis View	Create a set of definitions to extract, analyze, and report on selected history data.	File > New > History Analysis Document
Report View	Provides a set of icons for working with reports generated when a job or history analysis document is run.	Report > Open all reports










Toolbar Icons

The icons for each view are shown on the following tables. For more information on the command executed via an icon, look up Menus in the help system. The help for a menu will show any toolbar icons associated with a menu option and will explain what the option does.

Note: When some icons are clicked on, a dialog box will be displayed for entering options.

Icons for Control Task (Job) View












Icons for Control Task (Job) View		
Icon	Command	Menu to Access This Command
	New	File Menu
	Open	File Menu
	Save	File Menu
	Cut	Edit Menu
	Copy	Edit Menu

Icons for Control Task (Job) View		
Icon	Command	Menu to Access This Command
	Paste	Edit Menu
	Integrity Check	Run Menu
	Database Update	Run Menu
	Reconciliation	Run Menu
	Database Utilities	Run Menu
	Open Reports	Reports Menu
	Close Reports	Reports Menu
	Cascade Windows	Window Menu
	Print	File Menu

1 ■ Preparing to Use ACR/Detail Client

Using the ACR/Detail Client Interface






Icons for Input Source View

Icons for Input Source View		
Icon	Command	Menu to Access This Command
	Selection Record	Define Menu (Input Source View)
	Reformat Record	Define Menu (Input Source View)
	Key	Define Menu (Input Source View)
	Detail Record	Define Menu (Input Source View)
	Relative Record	Define Menu (Input Source View)
	Relative Record	Define Menu (Input Source View)
	Increase Indent	Define Menu (Input Source View)
	Decrease Indent	Define Menu (Input Source View)
	AND/OR Toggle	Define Menu (Input Source View)
	Find (available when cursor is in the Extract Window at the bottom of the view.)	View Menu
	Repeat Find (available when cursor is in the Extract Window at the bottom of the view.)	View Menu




Icons for Free-Form Report View

See “Icons for Free-Form Report View” on page 248.

Icons for External Translation Table View




Icons for External Translation Table View		
Icon	Command	Menu to Access This Command
	Selection (Criteria)	Define Menu (External Translation Table View)
	Reformat Record	Define Menu (External Translation Table View)
	Column Assignment	Define Menu (External Translation Table View)
	Process Control (Rule)	Define Menu (External Translation Table View)
	Relative Record	Define Menu (External Translation Table View)

Icons for Report View

Icons for Report View		
Icon	Command	Menu to Access This Command
	Zoom In	View Menu (after selecting Reports > Open all reports)
	Zoom Out	View Menu (after selecting Reports > Open all reports)
	Find	View Menu (after selecting Reports > Open all reports)

1 ■ Preparing to Use ACR/Detail Client

Configuring ACR/Detail Client

Icons for Report View		
Icon	Command	Menu to Access This Command
	Repeat Find	View Menu (after selecting Reports > Open all reports)
	Shade Bar	View Menu (after selecting Reports > Open all reports)
	Ruler	View Menu (after selecting Reports > Open all reports)

Working with Existing Job Definitions

For information on working with existing job definitions through the interface, see [“Working with Job Definitions” on page 69](#).

Configuring ACR/Detail Client

What Is a Configuration?

In ACR/Detail, a configuration is a group of settings that specify the following:

- A unique name. This enables you to create multiple configurations for the same mode (for example, multiple configurations with **Server Platform=Windows**).
- The server platform (configuration mode) where definitions will be stored and jobs will be run.
- A path for storing output reports.
- An ACR/Connector configuration name (if ACR/Connector is being used).
- The paths and file names for the definition and history databases used by this configuration.
- E-mail and Web publication settings, if applicable.
- The user options file to be used (Windows configurations only).

What Configuration Modes Are Available to You?

For a discussion of the configuration modes available, see “[Configuration Modes](#)” on page 17.

Setting Up A Configuration

You can create one or more configurations for configuration modes available to you through the Configurations dialog box (select **Options > Configurations**). As explained in the help, select **Add** or **Edit** to display the Add/Edit Configuration dialog box. The help for each dialog box tab will walk you through the process of setting up or changing a configuration.

After completing the Add/Edit Configuration dialog box, highlight the name of the configuration you added or edited and press **Set** to use it as your current configuration.

See the following sections for additional detail.

Setting Up Windows Configurations

You need to set up a separate configuration with the **Server platform** field set to Windows for each copy of ACR/Summary and ACR/Detail Server where you plan to run jobs on Windows from ACR/Detail Client.

For each Windows configuration, you will enter the IP address or DNS name of the Windows-based server and the port number specified in the ACR/Summary and ACR/Detail Server Properties dialog on the ACR/Summary and ACR/Detail Server. You will also specify the path name and file name of the definition and history databases on the server.

Complete the Web publication settings portion of the Add/Edit Configuration dialog box - Internet Information tab only if you are not using a UNIX or Linux configuration and you want to use Web publication as described in “[Distribution of Reports](#)” on page 268.

Note: In Windows mode, the E-mail notification settings at the top of the tab are grayed out. This is because, if you want to use e-mail notification in Windows mode, you must do the setup on ACR/Summary and ACR/Detail Server. For information on e-mail notification and configuration instructions, see [Message Processing](#) on page 106.

After completing the Add/Edit Configuration dialog box, highlight the name of the configuration you added or edited and press **Set** to use it as your current configuration.

1 ■ Preparing to Use ACR/Detail Client

UNIX or Linux Only: Modifications for RDBMS Access

Setting Up UNIX or Linux Configurations

You need to set up a separate configuration with the **Server platform** field set to UNIX for each UNIX or Linux box that you want to connect to.

For each UNIX or Linux configuration, you will enter the IP address or DNS name of the UNIX or Linux host and the port number of ACR/Summary and ACR/Detail Server.

You will also specify the path name and file name of the definition and history databases on UNIX or Linux.

UNIX or Linux Only: Modifications for RDBMS Access

Depending on your needs, the person who installed ACR/Detail for UNIX or Linux may already have made the following modifications to the product environment file, `acrprf`:

- Modified the Oracle section to enable direct access to Oracle tables as input sources.
- Modified the ODBC section to enable access to relational database tables via your ODBC drivers.

Note: When setting up a job where ODBC is used to retrieve data from a relational database table, you will need to verify with your UNIX or Linux database administrator that the UNIX or Linux data source for your ODBC session has been set up.

- Modified the DB2® section to enable access to DB2 tables through ODBC.

If the installer made the pertinent changes, you will have the required access. If not, you will need to use the administrative account to make the appropriate modifications in the administrative account's \$HOME/bin directory. For instructions, see the *ACR/Summary and ACR/Detail Installation Guide for UNIX and Linux*.

Initializing the Definition and History Databases

Before you can use ACR/Detail Client for the first time, you need to initialize (create) the definition and history databases.

- The definition database stores the definitions (parameters) used to reconcile data and generate reports.
- If the job is set up to store history, the history database stores the history of all the reconciliation keys that are processed (see [“What Is a Reconciliation Key?”](#) on page 116).

You can initialize both databases at once through the Database Initialization dialog box (select **Run > Database Initialization**).

Note: If you reinitialize, be careful to specify the correct database to initialize, because existing data will be erased.

Use the dialog box help to complete the fields and run the initialization. For more information on database initialization, see [“Database Initialization”](#) on page 230 and [“Initialize History”](#) on page 234.

Downloading, Uploading, and Importing Definitions

Note: Because downloading from and uploading to z/OS and importing from z/OS are primarily applicable to ACR/Workbench for Detail, instructions for this can be found in [“Using ACR/Workbench”](#) in the following sections:

- [Downloading and Uploading between z/OS and ACR/Workbench on page 309](#)
 - [Importing Definitions and Input Sources from z/OS on page 311](#)
-

Uploading to UNIX or Linux

With a UNIX configuration selected from the dropdown list at the right end of the toolbar, you can select **File > Upload** to display the File Upload dialog box.

Importing Definitions

For information on importing definitions, see [“Importing Definitions”](#) on page 257.

Setting the User Options

User options enable you to override a variety of defaults for system-wide ACR/Detail processing and reporting options in four major areas: miscellaneous, currency formats, negative signs, and date/time formats.

User options file settings can be overridden within specific job and file definitions.

For information on setting user options, see [Chapter 8, “Setting the User Options on page 173”](#).

Analyzing Your Reconciliation Needs

Before you can set up a reconciliation job, you need to plan exactly what the job will do. The analysis you need to perform is outlined below.

Determining the Input Source Information

An input source is an application report or file that contains values you want to extract for use in reconciliation. Before you can define input sources for a job, you need to know the following:

- The path and file name of each input source.
- The file organization type of each input source. Valid types include:
 - Physical Sequential—A physical sequential (ASCII) file.
 - Detail History Database—The history database will be used as input.
 - VSAM Key Sequenced—This is an ISAM file in Windows, UNIX, or Linux.
 - DB2 Table—Direct access to a DB2 table. If this file organization type is used, the job must be uploaded and run on z/OS.
 - Oracle—Direct access to an Oracle table. If this file organization type is used, the job can be run on UNIX or Linux only.
 - Direct Spool Dataset—If this file organization type is used, the job can be run on z/OS only.
 - User Program Accessed File—An external program will be called to access an input source type that ACR/Detail does not directly access.

- ODBC—Access to a relational database table via your site’s ODBC drivers. Definitions for this file type can be created and run on Windows, UNIX, or Linux.
- Web Page—The input source will be a Web page from the Internet.
- The values you want to reconcile from each input source. For example: Total charge for each telephone number from a Detail Charges Report and the sum of the charges for each telephone number from a Summary Charges Report.
- The key segments that will make up the reconciliation key (see “[What Is a Reconciliation Key?](#)” on page 116). The reconciliation key can consist of 40 positions: a maximum of 5 key segments, each of which can be up to 8 positions long. For example, Key 1 could be the area code 708 (3 positions). Key 2 could be the telephone number 8200252 (7 positions).

Planning for Internal and Extended Internal Items

You need to determine the internal and/or extended internal items that should be defined. For more information about these items, see “[Understanding Internal Items and Extended Internal Items](#)” on page 88. In your job, these items serve as storage places for the following types of data:

- Values extracted from the input source.
- Values derived from calculated items (see below).

Determining Calculated Items and History Items Needed

You may need to define calculated items and history items.

- A calculated item specifies mathematical manipulation of values that can include internal items, extended internal items, history items, other calculated items, or literals (numeric constants or text literals).
- A calculated item specifies mathematical manipulation of values from internal (or extended internal) items, history items, other calculated items, literals (numeric constants or text), or functions. Calculated items can be used in reconciliation rules in the same way as internal items, extended internal items, and history items. When a non-conditional calculated item is used in a reconciliation rule, the calculation is performed before the rule is evaluated. A conditional calculation is performed after rules are evaluated.

1 ■ Preparing to Use ACR/Detail Client

Analyzing Your Reconciliation Needs

- A history item specifies a value to be retrieved that was extracted in a previous run of the current reconciliation key or another specified key and stored in the history database. The retrieved value can be an internal or extended internal item, job run date, job run time, or cycle number.

Planning the Reconciliation Rules and Messages

Rules

Reconciliation rules specify the criteria to be applied to the values of internal items, extended internal items, calculated items, and/or history items to determine if an input source is in or out of balance.

Return Codes

A 3- or 4-digit return code can be assigned to a rule to indicate that the rule is out-of-balance on the Control Report. Use 4-digit codes between 0050 and 3999 to avoid duplicating system-generated codes. **Exception:** We recommend that you use 3 digits if you plan to use the \$? environment variable on UNIX or Linux. This variable is limited to 3 digits.

Messages

Each return code can be associated with a message that will appear in the Control Report to give information about the problem or to provide instructions for actions to be taken.

Planning for Reports

A Control Report will automatically be generated for every job run to report on the results of the job unless you specify otherwise. The Control Report shows whether the results of the reconciliation are in or out of balance.

When you set up the job, you will be able select whether you want to stop printing of the report or print the detail for only the key segments that are out of balance.

In addition to the Control Report, you can set up the User Report, Free-Form Report, and an output file containing the reconciliation results. For information on these reports, see [“Creating a Free-Form Report” on page 243](#), [“Setting Up User Reports” on page 253](#), and [“Write Output File” on page 77](#).

Tutorial

This chapter contains a tutorial that demonstrates the basic procedures for setting up and running a reconciliation job using ACR/Detail Client. It consists of the following sections:

- “Preparing for the Tutorial” on page 41
- “Determining Your Reconciliation Requirements” on page 42
- “Creating the Control Entity and Job” on page 45
- “Running the Job” on page 56
- “Viewing the Output Reports and Handling Errors” on page 57

Note: A tutorial for ACR/Workbench for Detail is provided in *ACR/Workbench for Detail Tutorial* on page 313

Preparing for the Tutorial

Before you begin this tutorial, you need to do the following:

1. Access ACR/Detail Client. For instructions, see “[Accessing and Exiting from ACR/Detail Client](#)” on page 25.
When you start ACR/Detail Client, the Control Entity View appears. This is the default view.
2. Familiarize yourself with the product interface. See “[Using the ACR/Detail Client Interface](#)” on page 26.
3. Establish configuration(s) for the tutorial. Note the following:
 - If you will be defining and running the job on a Windows server with ACR/Summary and ACR/Detail Server installed, establish a configuration with Server Platform set to Windows.
 - If you will be running the job on UNIX or Linux, ACR/Detail for UNIX or Linux must be installed on the UNIX or Linux box you will use.
See “[Configuring ACR/Detail Client](#)” on page 34 for more information.

2 ■ Tutorial

Determining Your Reconciliation Requirements

When you establish your configuration, ensure that the option to **Automatically update Definition Database on file saves** has been selected in the Add/Edit Configurations dialog box - File Information tab.

- If you will be running the tutorial using ACR/Workbench, you can use a Local configuration.
- 4. Make sure the configuration you set up in the preceding steps is selected in the list box to the right of the toolbar when you start the tutorial.
- 5. Ensure that you followed the instructions in “[Initializing the Definition and History Databases](#)” on page 37.
- 6. Verify that you have access to the sample reports that will be used as input sources in the tutorial. The default locations for these reports are as follows:
 - Windows mode: The reports should be in the SumDetServer\Samples directory that was created when ACR/Summary and ACR/Detail Server was installed.
 - UNIX or Linux mode: ../sumdet/samples/. (In your product installation subdirectory on UNIX or Linux.)
 - Local mode: c:\Infogix\Detail32\Samples is the default.

Determining Your Reconciliation Requirements

In this tutorial you will create a reconciliation job that uses two telephone company report files as input. The job will take values from each report and verify whether they are in balance based upon a rule that you will define. The reports are as follows:

The Summary Report (Dex1dd1.rpt) contains the total charge for each telephone number.

SUMMARY REPORT	
<u>PHONE NO.</u>	<u>TOTAL</u>
7088200252	0036.51
3123625227	0000.25
4126384664	0028.03

Determining Your Reconciliation Requirements

The Detail Report (Dex1dd2.rpt) is a detailed list of each telephone call and the associated charge.

DETAILED ACTIVITY REPORT	
<u>PHONE NO.</u>	<u>AMOUNT</u>
7088514698	0004.31
7088200252	0002.31
2013457623	0003.11
2157483927	0022.99
7088514698	0023.13
7088200252	0005.11
7088200252	0012.97
3123525227	0000.25
4126834644	0028.03
7088200252	0016.12

These reports will be the input source files for your job.

Determine your reconciliation requirements as follows:

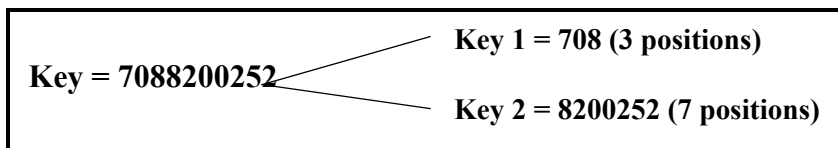
1. Decide what you want to reconcile.

For this tutorial, assume you want to verify that the company's billing is correct by verifying that the totals on the Summary Report equal the sum of each individual call for that phone number on the Detail Report.

2. Identify the key fields.

The job's reconciliation key is a hierarchical structure composed of up to 5 key segments, each of which can contain up to 8 positions. The maximum size of a reconciliation key is therefore 40 positions. Each key segment is created by defining a key field.

In our job, we will define two key segments for each input source file: the area code 708 and the telephone number 8200252, as shown below. For more information, see [“What Is a Reconciliation Key?”](#) on page 116.



3. Identify the detail fields. A detail field defines the value that is to be extracted from an input source each time the reconciliation key is found. For this tutorial, you will identify two detail fields:

2 ■ Tutorial

Determining Your Reconciliation Requirements

- For the Summary Report (Dex1dd1.rpt), one detail field will be needed to extract and accumulate the total charge for each telephone number. The detail field will specify the first position and length of the data to be extracted. The accumulated value will be stored as internal item 001.

Note: A longer value (a number with 16 to 30 digits or a text value with 9 to 80 characters) would need to be stored as an extended internal item. Because extended internal items are very similar to internal items, they are not included in the tutorial. For more information, see [Understanding Internal Items and Extended Internal Items on page 88](#).

- For the Detail Report (Dex1dd2.rpt), one detail field will be needed to extract and accumulate the detailed charge for each telephone call by telephone number. Again, the field will specify the first position and length for data to be extracted. The accumulated value will be stored as internal item 002.
4. Identify any rules that will be used to manipulate the internal items.
The only rule required for this job will be: SUMMARY AMOUNT (internal item 001) should be equal to DETAIL AMOUNT (internal item 002). When defining the rule, you will specify the return code that will be generated when the rule is out of balance.
 5. Determine any messages you need to define that will be included in the Control Report when a particular return code is generated.
Messages can help identify why a job is out of balance or provide instructions about the measures that should be taken.

You will create the following message to be printed when the rule for this job is out of balance:

EXAMPLE IS OUT OF BALANCE

Creating the Control Entity and Job

Before you begin, ensure that you have completed the instructions in “Preparing for the Tutorial” on page 41.

Complete the Basic (Entity) Information

The Control Entity View is the default view when you start ACR/Detail Client. You will use this view to define the control entity for the tutorial.

A control entity is a feature for easily creating and saving one or more reconciliation jobs that share the same key structure.

For this tutorial, you need to complete only selected options and fields. For the other options and fields, you will use the default values.

1. Click on the Description option to display the Control Entity Description dialog box and enter the description PHONE REPORTS. Click **OK**.
2. Identify the keys (key segments).
At the control entity level, you can optionally give a meaningful description to each key segment that will be part of the control entity’s key structure. These descriptions will make it easier to create your input source definitions.
Click on the key segments shown in the table to open the associated dialog boxes and name the key segment. Select **OK**.

Field	Enter or Select	Reason/Notes
Key 1	Area Code	Provide a meaningful name for the key segment.
Key 2	Telephone Number	Provide a meaningful name for the key segment.

Name and Save the Entity (.dob) File

Click the **Save** icon to save the entity file.

Name the entity PHONE. The file extension will default to .dob.

Begin Defining the Control Task (Job)

In ACR/Detail Client, a control task is the same as a reconciliation job. It defines the rules and processing options required for reconciliation. Each job uses a reconciliation key composed of one or more hierarchical key segments to extract data for reconciliation.

To begin defining the job, click **New** under Control Tasks (Jobs) to open the Control Task (Job) View.

Complete the Basic (Job) Information

The Basic Information folder is the first folder on the Control Task (Job) View.

For this tutorial, you need to complete only selected Basic Information options and fields. For the other options and fields, you will use the default values.

1. Click on the **Name** option to display the Control Task (Job) Name dialog box.
2. Complete the fields shown in the table. Select **OK** to close each dialog box

Field	Enter or Select	Reason/Notes
Control Task Name	JOBXYZ (before the slash). STEP10 (after the slash)	The job name and step name together provide a unique name for the job.
Control Report Title	PHONE EXAMPLE	Provides a meaningful name for the Control Report for this job.
Set Return Code	Yes (highest Reconciliation Return Code)	Sets the job completion code to the highest return code for the job.

3. Click the **Save** icon again to save the entity with the basic job definitions you have created.

Note: To identify the icons, hover the mouse over them or see [Toolbar Icons on page 30](#).

Define the First Input Source

For each input source you define for a control task, you must include Basic Information and at least one selection group.

Input sources are defined through the Input Source View. Your first input source will be the Summary Report, Dex1dd1.rpt. To begin defining it, click on **New** under Input Sources to open the Input Source View.

1. In the Basic Information section at the top of the view, click **Name** to display the Name dialog box. Complete only the fields shown in the following table:

Field	Enter or Select	Reason/Notes
File ID	SUMPHONE	Provides a meaningful name for the input source.
File Name	Enter the full path and file name of the report, Dex1dd1.rpt, or browse to select the file. When you select OK , the file will be displayed in the Extract Window at the bottom of the view.	If you are in UNIX or Linux mode, the file should be in ../sumdet/samples/ (your product installation subdirectory on UNIX or Linux). If you are in Windows mode, the file should be in the \\..SumDetServer\Samples folder that was created when ACR/Summary and ACR/Detail Server was installed. If you are in Local mode, the file should be in C:\Infogix\Detail32\Samples.

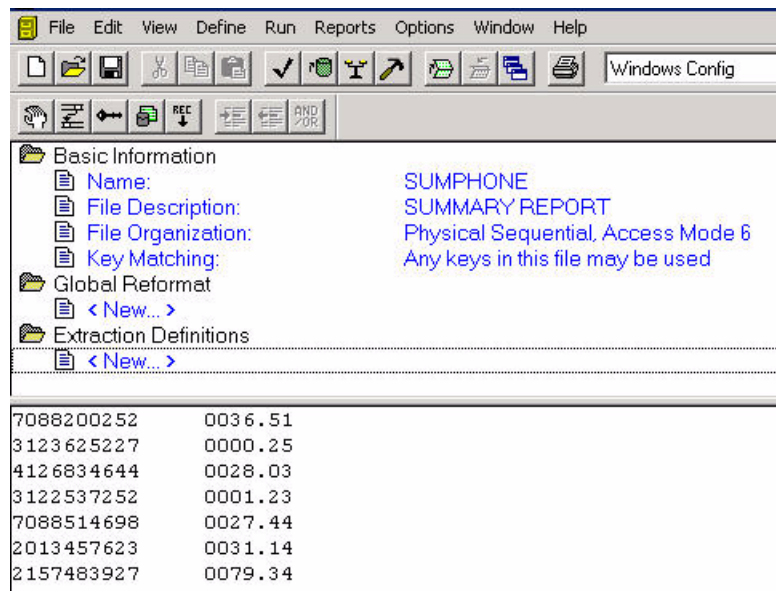
Note: If you were using a copybook, you would enter its name in the **Record Layout** field.

2. Add a description for the input source file. Click **File Description** to display the File Description dialog box, enter the description **SUMMARY REPORT**, click **OK**.

2 ■ Tutorial

Creating the Control Entity and Job

With the input source displayed, the view will look like this:



Now you are ready to complete your extraction definitions. Extraction definitions are made up of one or more selection groups.

A selection group consists of one or more consecutive selection fields that are evaluated together to determine whether the current record will be selected for further processing or bypassed. (In access mode 6, nested selection groups are supported). In addition, the group must have one or more detail fields. One or more reformat fields and relative record fields can optionally be used.

3. Define a selection field.

Begin by highlighting the **70882002** in the displayed report with your mouse.

Click the **Selection** icon on the toolbar to display the Selection Field dialog box.

By default, in the Selection Area, the **Format** is Text. Do not change this option. You want to have **Format** set to Text in order to scan any alphanumeric characters at the indicated positions. You can then set up the field to run a comparison to determine if all characters are numeric before picking them up. If you changed the **Format** field to Number, this would automatically indicate that any values occupying the specified positions on the record were numeric (even if they were not), text values would be ignored, and the comparison would not work.

The **Comparison Type** is Equal To and the **Compare to** field is set to Constants. This would result in selecting only the records that have the exact value you selected with the mouse. To indicate that any record should be selected that has numerics in the first five positions, change the **Comparison Type** to Numeric.

When you click **OK** to close the dialog box, the Input Source View is displayed showing your selection field definition under Extraction Definitions.

```
SELECT IF POSITION 1 FOR A LENGTH OF 8 IS NUMERIC
```

4. Recall from “[Determining Your Reconciliation Requirements](#)” on [page 42](#) that you need to define the following key segments from each report:

- Key 1 708 (the area code)
- Key 2 8200252 (the telephone number)

Define Key 1 by highlighting 708 on the report with your mouse.

Click the **Key** icon on the toolbar to open the Key Field dialog box.

The **Position** and **Length** fields will be pre-filled. In the **Key ID** field, select AREA CODE. This is the label for the job’s first key segment, which you defined in Control Entity View.

5. Define Key 2, the telephone number, by highlighting **8200252** on the report.

Click the **Key** icon on the toolbar to open the Key Field dialog box.

In the **Key ID** field, select TELEPHONE NUMBER. This is the label for the job’s second key segment, which you defined in Control Entity View.

6. Define the detail field and specify the internal item that will store the extracted value.

A detail field specifies the actual value you want to extract (and optionally accumulate) from each selected record. For the Summary Report, you will need to specify one detail field for the summary total charge for a telephone number.

Highlight **0036.51** on the report.

Click the **Detail** icon on the toolbar to open the Detail Field dialog box. The position and length fields will be pre-filled.

Note that you will be using the following defaults:

- For the **Format** field, use the default, Number.
- The default for the **Field Type** field is SUM. The values extracted will be summed by key (that is, by telephone number).

2 ■ Tutorial

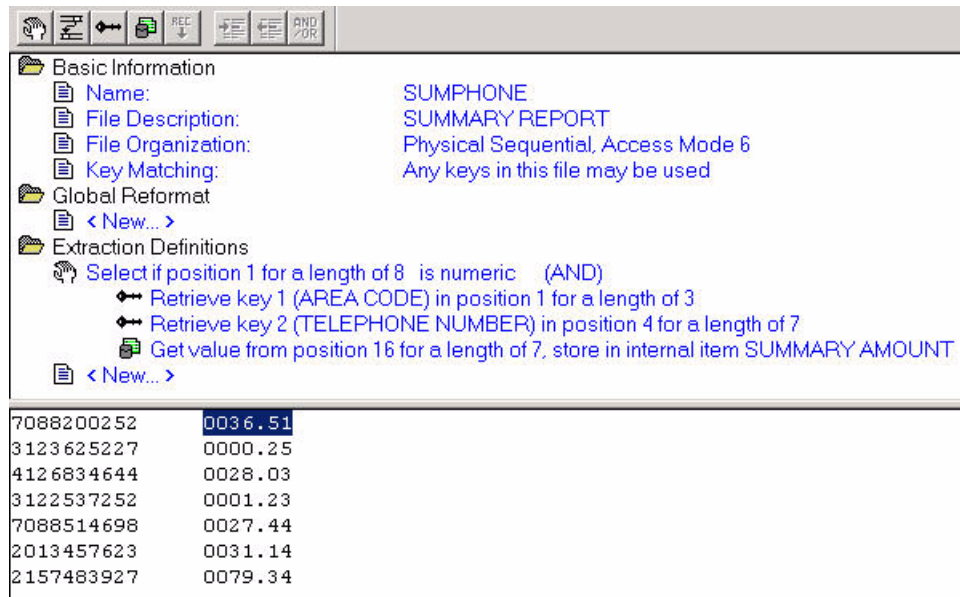
Creating the Control Entity and Job

- The default for the **Accumulate Option** field is Sum values with the same key; replace history. The values extracted will be summed by key and any values for the same key in the history database will be replaced.
 - The default for the **Target Area** is Internal Item. The summary total charge for each key (that is, for each telephone number) will be stored in an internal item for use in the reconciliation rules.
7. Click **Create/Change** to open the Create/Change Internal Items dialog box. Double click **New** to define a new internal item. When the Internal Item dialog box appears, complete only the fields shown in the following table:

Field	Enter or Select	Reason/Notes
Name	SUMMARY AMOUNT	Indicates that this internal item will store the summary charge for each key (that is, for each telephone number).
Description	AMOUNT FROM REPORT 1	Adds additional description for the item.
Format	Amount	Specifies the format of the extracted data.

Click **OK** to return to the previous dialog box. Click **Close**. Click **OK** to return to the Input Source View and view your extraction definitions for this input source.

Note: This and any other internal item you define for a job in this control entity will appear in the Common Internal Items folder in Control Entity View and can be selected for use in other jobs in this control entity.



8. Close the Input Source View.

Define the Second Input Source

Your second input source will be the Detail Report, Dex1dd2.rpt.

Click on **New** under Input Sources to open the Input Source View.

1. In the Basic Information section at the top of the view, click **Name** to display the Name dialog box. Complete only the fields shown in the following table.

2 ■ Tutorial

Creating the Control Entity and Job

Field	Enter or Select	Reason/Notes
File ID	DETPHONE	Provides a meaningful name for the input source.
File Name	Enter the full path and file name of the report, Dex1dd2.rpt, or browse to select the file. When you select OK , the file will be displayed in the Extract Window at the bottom of the view.	If you are in UNIX or Linux mode, the file should be in ../sumdet/samples/ (your product installation subdirectory on UNIX or Linux). If you are in Windows mode, the file should be in the \\..SumDetServer\Samples folder that was created when ACR/Summary and ACR/Detail Server was installed. If you are in Local mode, the file should be in C:\Infogix\Detail32\Samples.

2. Add a description for the input source file. Click **File Description** to display the File Description dialog box, enter the description **DETAIL REPORT**, click **OK**.

3. Define a selection field.

Starting with the first position of the second line, highlight **04216** in the displayed report with your mouse.

Click the **Selection** icon on the toolbar to display the Selection Field dialog box.

As you did for the selection field in the first input source, change the **Comparison Type** to **Numeric**.

When you click **OK**, the Input Source View is displayed showing your selection field definition under **Extraction Definitions**:

```
SELECT IF POSITION 1 FOR A LENGTH OF 5 IS NUMERIC
```

4. Again you will define the following key segments:

- Key1 708 (the area code)

- Key 2 8200252 (the telephone number)

Define Key 1 by highlighting **708** on the report with your mouse.

Click the **Key** icon on the toolbar to open the Key Field dialog box.

The **Position** and **Length** fields will be pre-filled. In the **Key ID** field, select **AREA CODE**.

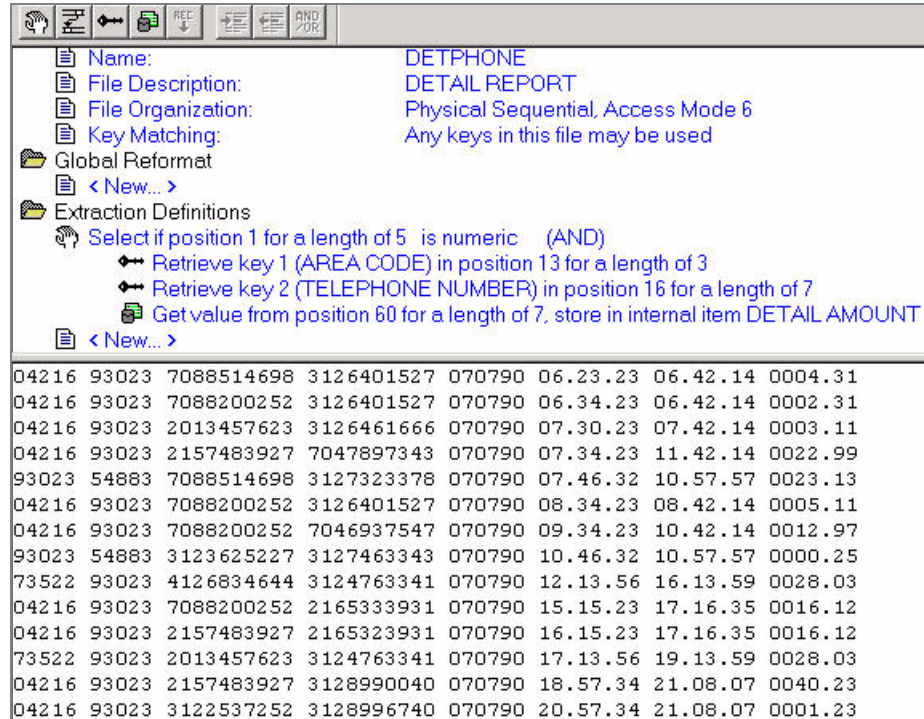
5. Define Key 2, the telephone number, by highlighting **8200252** on the report.
Click the **Key** icon on the toolbar to open the Key Field dialog box.
In the **Key ID** field, select TELEPHONE NUMBER.
6. Define the detail field and specify the internal item that will store the extracted value.
For the Detail Report input source, you will need one detail field to extract the detailed charge for each telephone call.
Highlight **0002.31** on the report.
Click the **Detail** icon on the toolbar to open the Detail Field dialog box.
Note that you will be using the following defaults just as you did for the first detail field you defined:
 - The **Position** and **Length** fields will be pre-filled. Use the pre-filled values.
 - The default for the **Format** field is Number. Use the default.
 - The default for the **Field Type** field is SUM. The detail charges extracted will be summed by key (that is, by telephone number).
 - The default for the **Accumulate Option** field is Sum values with the same key; replace history. The detail values extracted will be summed by key and any values for the same key in the history database will be replaced.
 - The default for the **Target Area** is Internal Item. The summary total charge for each key (that is, for each telephone number) will be stored in an internal item for use in the reconciliation rules.
7. Click **Create/Change** to open the Create/Change Internal Items dialog box. Double click **New** to define a new internal item. When the Internal Item dialog box appears, complete only the fields shown in the following table:

Field	Enter or Select	Reason/Notes
Name	DETAIL AMOUNT	Indicates that this internal item will store the accumulated detailed charges for each key (that is, for each telephone number).
Description	AMOUNT FROM REPORT 2	Adds additional description for the item.
Format	Amount	Specifies the format of the extracted data.

2 ■ Tutorial

Creating the Control Entity and Job

Click **OK** to return to the previous dialog box. Click **Close**. Click **OK** to return to the Input Source View and view your extraction definitions for this input source.



8. Close the Input Source View.

Define the Reconciliation Rule

You can now define the rule that will be applied to the internal items to determine if the values extracted from the reports are in or out of balance. Recall from “[Determining Your Reconciliation Requirements](#)” on page 42 that the rule will be: SUMMARY AMOUNT (internal item 001) should be equal to DETAIL AMOUNT (internal item 002).

1. Under the Rules folder, click **New** to display the Rule Type dialog box. You can then choose between the two types of rules:
For this tutorial, keep the default selection, Standard Rule, and click **OK**.

2. When the Standard Rule dialog box opens, on the General Information tab, complete only the fields shown in the following table:

Field	Enter or Select	Reason/Notes
Name	SUMMARY/ DETAIL CHECK	Give a meaningful name to the rule.
Return Code	1000	Indicates the code that you want to be set if the rule is out of balance.
FORMAT	Amount	You are comparing extracted amounts.

3. Select the Comparison tab. The **Available Items** field on the right side of the screen shows the internal items you defined for this job. First, complete the left-hand side of the comparison as follows.

Field	Enter or Select	Reason/Notes
Left-hand side		Make sure this field is selected.
Operator	EQ (equal to)	Make sure this field is selected.
Available Items	SUMMARY AMOUNT	Click this to highlight.
Add	Click this.	Enters SUMMARY AMOUNT in the Included Items field.

4. Complete the right-hand side of the comparison as follows:

Field	Enter or Select	Reason/Notes
Right-hand side	Select this.	
Available Items	DETAIL AMOUNT	Click this to highlight.
Add	Click this.	Enters DETAIL AMOUNT in the Included Items field.

5. Click on the General Information tab and verify that the Rule is defined as follows: (SUMMARY AMOUNT) = (DETAIL AMOUNT)

6. Click **OK**. The Message dialog box displays.

Define the Message

The Message dialog box enables you to define a message to be printed when this rule generates the out-of-balance return code (in this case 1000). In the **Message Text** field, enter the following message:

```
EXAMPLE IS OUT OF BALANCE
```

Click **OK** to return to the Control Task (Job) View. The return code and message text will be displayed:

```
1000          EXAMPLE IS OUT OF BALANCE
```

Running the Job

Update the Definition Database

Before you can run the job, your definitions must be updated to the definition database. Click on the **Database Update** icon to open the Update Definition Database dialog box and click **OK**.

When you update the database, an integrity check will be performed. This utility examines the validity of the current control entity and its control tasks (jobs) and identifies any inconsistencies in the Integrity Check Window.

If problems are detected, correct any errors and rerun the database update.

When the update completes successfully, you should see a step completion code of 0000 in the status bar.

Run Reconciliation

Click the **Reconcile** icon to display the Run Reconciliation dialog box.

The Current Databases section shows the definition and history databases for your selected configuration. The definitions for this job will come from the definition database shown here. The Runtime Parameters section shows the job ID, which is entered automatically.

1. In the **Cycle Number** field, optionally enter a meaningful number such as today's date. Keep the default entries for the other fields.
2. Click **OK**. When the job completes successfully, you should see a step completion code of 0000 in the status bar at the bottom of the screen.

Viewing the Output Reports and Handling Errors

Click the **Open Reports** icon to view the reports. You should see a SYSOUT report showing 7 keys for the job, 7 records selected from SUMPHONE and 7 records selected from DETPHONE.

Close the SYSOUT report to see the Control Report with the result IN BALANCE and Return Code 0000 at the bottom of the page. For more information, see [“Viewing Reports and Handling Errors”](#) on page 68.

2 ■ Tutorial

Viewing the Output Reports and Handling Errors

Setting Up a Job

This chapter provides summary instructions for setting up a typical reconciliation job. It includes the following sections:

- “What Is a Reconciliation Job?” on page 59
- “What is a Control Entity?” on page 60
- “Preparing to Set up a Reconciliation Job” on page 64
- “Setting Up and Running a Job” on page 64
- “Viewing Reports and Handling Errors” on page 68
- “Working with Job Definitions” on page 69

For information on special types of reconciliation, see Chapter 9, “Using Multi-Level and Suspense Reconciliation on page 185”.

What Is a Reconciliation Job?

The basis of ACR/Detail's functionality is the reconciliation job, which is also called a control task. A reconciliation job defines the rules and processing options required for reconciliation. Each job uses a reconciliation key to extract data for reconciliation.

Components of a Job

Each job requires both job and file definitions, as follows:

- Job definitions specify how values are processed. Job definitions include definitions for:
 - Internal Items
 - Extended Internal Items
 - Extraction Variables
 - History Items
 - Calculated Items
 - Rules
 - Messages
 - Reports

More details on each type of definition are provided in Chapter 6, “Setting Up Job Definitions on page 87”.

3 ■ Setting Up a Job

What is a Control Entity?

- File definitions specify how values will be extracted from one or more input sources (often reports from your application), and may specify how they will be reformatted or translated and how the extracted values will be stored. More details on file definitions will be provided in [“Setting Up Input Sources” on page 115](#).

Definitions for external translation tables, internal translation tables, cycle tables, and history analysis reports, which are specified at the control entity level, are not defined in the Control Task (Job) View, but in the Control Entity View (detailed in the next section). However, a job may utilize any of the tables or history analysis definitions defined for its control entity.

What is a Control Entity?

A control entity is a feature of the graphical interface that makes it easy to create and save multiple reconciliation jobs (control tasks) that share some of the same components.

The control entity does not exist in the z/OS version of ACR/Detail, so you may find it helpful to set up your jobs in ACR/Workbench and upload them to z/OS. In the graphical interface, you can refer to the internal items, extended internal items, extraction variables, history items and keys by name rather than just by number. These names will be visible in the Control Entity View and/or the Control Task (Job) View.

Control Entity View is the default view when you start ACR/Detail. You can also open this view by selecting **File > New**.

For instructions on creating a new control entity containing one job, see [“Setting Up and Running a Job” on page 64](#).

The following sections describe the folders in Control Entity View.

Basic Entity Information

The following control task (job) settings (for all jobs in the entity) are defined in the Basic Entity Information Folder in Control Entity View.

Description

This is an optional description of the entity.

Length of Descriptions

This setting specifies the maximum number of characters allowed for descriptions of internal items, extended internal items, history items, calculated items, and rules: (40 or 80) for a job in this entity. If you will be moving jobs to z/OS, use only 40 byte descriptions.

History Runs

This setting defines the maximum number of job runs whose results will be saved in the history database for a job in this entity.

Key 1, Key 2, Key 3, Key 4 Key 5: Reconciliation Key Segments

Any key segment defined in one control task (job) in an entity using a key field will appear in the Basic Entity Information folder in Control Entity View and will then be available to be selected and reused by any other job in the entity. While all jobs in the entity use the same sequence of key segments, each job need not use all of the segments that appear in Control Entity View. For example, job A might use Key 1, Key 2, and Key 3. Job B may use Key 1, Key 2, Key 3, and Key 4.

You can optionally click on any key segment (Key 1, Key 2, etc.) in Control Entity View and give it a meaningful name. This name will then appear in Control Task (Job) View instead of the key number. If you want to name the keys, you can do so before or after you define the keys using key fields in Input Source View.

Common Internal (and Extended Internal) Items

Common Internal Items

When you define common internal items, you are identifying values that can be shared, accumulated, and stored by any control tasks (jobs) in this control entity. As you set up these items, you can name them and then use those names when you construct your rules. The common internal item names also appear on the Control Report as labels for the values that were reconciled.

You can define internal items from the Control Entity View or define them when you set up your control tasks (jobs) (see “[Control Tasks \(Jobs\)](#)” on [page 62](#) below). Any internal items that you define at the control task (job) level will also automatically be classified as common internal items. If you copy and paste an internal item in Control Task (Job) View, the corresponding common internal item will also be built.

3 ■ Setting Up a Job

What is a Control Entity?

Common Extended Internal Items

Common extended internal items are like common internal items (described above) except that they can be used to store longer values (numbers with 16 to 30 digits or text values with 9 to 80 characters).

You will learn more about internal and extended internal items in [“Understanding Internal Items and Extended Internal Items”](#) on page 88.

Common Extraction Variables

Extraction variables are items you define to store regular or extended data from file definitions or external translation table definitions for further processing.

Typically, you will define extraction variables when you are creating file definitions at the input source level or when you are creating external translation table definitions in External Translation Table View. These definitions will automatically be classified as common extraction variables and will appear under Common Extraction Variables in the Control Entity View.

Alternatively, you can define common extraction variables directly from Control Entity View.

You will learn more about extraction variables in [“Defining Extraction Variables”](#) on page 91.

Control Tasks (Jobs)

This folder enables you to create a job within the entity. For instructions on creating a new control entity containing one job, see [“Setting Up and Running a Job”](#) on page 64.

Alternate Control Tasks

This setting applies only to jobs in this entity that are uploaded and run on z/OS. For more information, see [“Alternate Control Tasks \(Control Task \(Job\) View > Alternate Control Tasks > New\)”](#) on page 316.

Tables

You can use this folder to create definitions for external translation tables, internal translation tables, and cycle tables. Internal and external translation tables can be referenced in reconciliation jobs to translate data from an input source. Cycle tables are used to select cycles for history analysis. For more information, see [Chapter 10, “Using Translation Tables on page 193”](#) and [“Cycle Tables” on page 227](#).

Note: In addition to the table types created using the Tables folder, you can create and use dynamic translation tables to translate key segments of up to 80 bytes to 8 byte identifiers. For more information, see [Initializing and Referencing Dynamic Translation Tables on page 218](#).

History Analysis Process Definitions and Report Definitions

History analysis process definitions and history analysis report definitions are used to run the history analysis feature, which enables you to analyze historical data.

The history analysis process definitions identify the following:

- The history keys and cycles (job runs) to be processed.
- The history analysis report definitions to be processed using these keys and cycles.

Each set of history analysis report definitions defines one History Analysis Report. The report is cycle-oriented. A history key or an accumulation of history keys in the current cycle of data is compared to prior cycles of data and then displayed in the report.

For more information, see [“Using History Analysis” on page 221](#).

Saving the Contents of a Control Entity

When an ACR/Detail Client job is created and saved, *all definitions* within its entity (including the Basic Entity Information settings and all definitions created within the entity for jobs, files, tables, and history analysis) and are placed in the ACR/Detail Client default directory along with the following files:

- A .dob file containing all definitions.
- A .def file containing all batch transaction information.
- A .df2 file contains the control task name and associated input source paths.

3 ■ Setting Up a Job

Preparing to Set up a Reconciliation Job

After you update the definition database (from any view within this control entity), you can run any of the jobs and history analysis definitions defined in the entity.

Preparing to Set up a Reconciliation Job

Before you begin setting up a job, ensure that you have done the following, all of which were documented in [Chapter 1, “Preparing to Use ACR/Detail Client.”](#)

- Configured ACR/Detail Client.
- Initialized your definition and history databases.
- Familiarized yourself with interface elements such as views, folders, menus, toolbar icons, and dialog boxes. See [“Using the ACR/Detail Client Interface” on page 26,](#)
- Analyzed your reconciliation needs so that you know exactly what you want to accomplish.

Setting Up and Running a Job

This section provides an overview of the steps involved in setting up a control entity containing one reconciliation job through ACR/Detail Client.

Note: For information on working with existing job definitions, see [Working with Job Definitions on page 69.](#)

The procedure is as follows:

1. Open a new Control Entity View. This is the default view when you start ACR/Detail. You can also open this view by selecting **File > New**.
2. If your job requires tables, first set up your translation and/or cycle tables by double clicking **New** under Tables in Control Entity View. You may need the following table types:
 - **Translation table**—A table used to translate values from an input source for reconciliation purposes. Two types are available:
 - **Internal translation table**—A static table that you create and populate within ACR/Detail to do simple translations from one text value to another text value.

- **External translation table**—A table used to translate values from an input source for reconciliation purposes. Unlike an internal translation table, an external translation table is built at run time, usually by referencing tabular data external to ACR/Detail. External translation tables are used when an internal translation table (which performs only simple 1 to 1 translations from one text value to another text value) will not suffice.

For more information on translation tables, see [Chapter 10, “Using Translation Tables on page 193”](#).

- **Cycle table**—A table used to select cycle numbers for history analysis. For example, if you want to analyze data from the last ten Tuesdays, you would have created and named a cycle table that points to the cycles that represent the last ten Tuesdays. For more information, see [“Cycle Tables” on page 227](#).
3. Use the Basic Entity Information folder to set up the basic entity information options. These options were described in [“Basic Entity Information” on page 60](#).
 4. Click **File > New > Control Task (Job)** to open the Control Task (Job) View and start defining a job.
 5. Use the options in the Basic Information folder to set up the basic job information. For more information on this step, see [“Completing the Basic \(Job\) Information Options” on page 74](#).
 6. Use the Input Sources folder to open the Input Source View and set up your first input source.

An input source is a file (often a report from your application) from which values will be extracted to be used in the reconciliation job.

When you set up each input source, you define the following information:

- Location and name of the file or its record layout (or the URL, if it is a Web page).
- File organization type, such as physical sequential file, Detail History Database, etc.
- Access mode, which determines how records will be selected from the input source file. Most jobs use access mode 6 (keys precede detail values).
- Key matching requirements: Options for using the keys extracted from this input file in determining whether keys extracted from each of the job's other input files will be used in reconciliation. An option is available to match data based on a partial key.

3 ■ Setting Up a Job

Setting Up and Running a Job

- An optional return code to be set when no records are selected for extraction from this file.
- Optional comments on the file that will print on a List Definitions Report that includes detailed file information.
- Extraction definitions. These define how records will be selected from the input source and what data will be extracted.

If you have additional input sources, open another Input Source View for each.

For more information, see [Chapter 7, “Setting Up Input Sources on page 115”](#).

7. Create the job definitions as needed. There is a separate folder in Control Task (Job) View for each job definition type. Click **New** to create a new definition.

Internal items—These are storage places for the following types of data to be used in your reconciliation rules:

- Values (counts, amounts, dates, text items, or time items) that will be extracted from the input source.
- Values derived from calculated items.

Extended Internal items—These are like internal items but can be used to store longer values (numbers with up to 30 digits or text values with up to 80 characters).

Extraction variables—These are items you define to store regular or extended data from file definitions or external translation table definitions for further processing.

History items—These identify values extracted in a previous run of the current reconciliation key or another specified key and stored in the history database that will be used in the current run.

Calculated items—These specify computations that can use internal items, extended internal items, history items, and other calculated items.

Rules—Rules enable you to specify the criteria ACR/Detail will use to determine if your reports, files, and other information sources are in- or out of balance. If one or more rules are out of balance, the entire reconciliation job is considered to be out of balance. When a rule detects an out-of-balance condition, it generates the return code that you define.

Messages—These are lines of text that are included in the Control Report when a particular return code is generated.

8. Use the Report Information folder in Control Task (Job) View if you want to create the following customized reports:
 - Reconciliation User Report**—This report can show any of the data in the Control Report in the sequence you specify. For more information, see [“Setting Up Free-Form Reports” on page 243](#).
 - Free-Form Report**—A customizable report on the job results. For more information, see [Chapter 14, “Setting Up User Reports on page 253”](#).
 - Output File**—A Freeform Output File called UNIDAT2.DAT in line sequential format that contains results for specified internal and/or extended internal items or for a reconciliation key. For more information, see [“Setting Up an Output File” on page 81](#).
9. Click the **Database Update** icon to update the definition database.

Note: When you update the definition database, ACR/Detail automatically performs an integrity check. For more information on the integrity check feature, see [Integrity Check on page 242](#).

10. Save the job.
11. Prepare to run the job. From Control Task (Job) View, click the **Reconciliation** icon. If you do not have a control task open or selected and the control entity has multiple control tasks, select the task you want to run from the Control Task List dialog box. Click the **Database Update** icon. The Run Reconciliation dialog box will display. From this dialog box, you can:
 - Click **Add** or **Edit** to display the File ID and File Name dialog box. This enables you to:
 - Add, edit, or remove to add, edit, or remove an input source from this run of reconciliation processing.
 - Request a Trace Report by clicking **Produce Trace Report**. This report shows a formatted list of the selection groups as they were loaded in preparation for processing the input file. A Trace Report can be produced for an input source and/or for an external translation table. For a sample Trace Report for an input source, see [“Trace Report” on page 280](#). For a sample Trace Report for an external translation table, see [“External Translation Table Trace Report” on page 208](#).
 - Optionally click **Overrides** to specify run-time overrides for selected options from the Basic Job Information folder.
12. Click **Run** to run the job.

Viewing Reports and Handling Errors

Viewing Reconciliation Reports

Review any reconciliation reports you specified for the job (Control Report, User Report, Output File, Free-Form Report, Recap Report, Extracted Data Detail Report, Sorted or Accumulated Data Detail Report, History Data Detail Report, Trace Report), for processing errors, out-of-balance conditions, and other information. Most of the reports are described in Appendix B, “Reports”. For more information on the following reports, see these sections:

- “Setting Up Free-Form Reports” on page 243
- “Setting Up User Reports” on page 253

Handling Errors

The following are general guidelines for handling errors:

Processing errors—If the status line at the bottom of the display or the System Messages (SYSOUT) Report shows #U or #INS system messages or return codes, look up the messages and return codes in the messages and codes manual for further information.

Note: ACR/Detail return codes include the following:

- 0016, indicating a processing error.
 - 4xxx, indicating the type of processing error.
-

Out-of-balance situations—If the Control Report shows a 4-digit return code and an associated message indicating an out-of-balance condition:

- Determine if modifications need to be made to the job or file definitions.
- Ensure that you are using the right files as input sources and that the job ran in the correct sequence.
- Make any required changes.

If none of these problems occurred, you may have an actual out-of-balance condition, indicating one or more errors in the data that need to be corrected.

After making corrections, if the job needs to be rerun, it can be executed with the same cycle ID, which automatically overlays the information previously stored in the history database for that job.

Getting Help

If you encounter errors that you cannot resolve, see “Contacting Customer Support” on page 23.

Working with Job Definitions

This section covers creating new entities and jobs as well as creating, saving, opening, copying, pasting, and deleting definitions

Creating a New Control Entity and Job

When you start ACR/Detail Client, you will see the Control Entity view, which is the default view. You can also open the view for a new job by clicking the **New** icon on the toolbar.

From the Control Entity View, open the Control Task (Job) View to create a reconciliation job.

Saving Definitions

Click the **Save** icon to save all definitions in the control entity. This displays a dialog box that enables you to give a meaningful name to the entity. The default location for the file is the ACR/Detail installation folder, but you can change it. Click the **Database Update** icon if you want to move saved definitions to the definition database and run the job.

Note: When configuring your system, you can set the option to **Automatically update Definition Database on file saves** in the Add/Edit Configurations dialog box - File Information tab.

If you open an entity and modify any of its definitions, you should save the changes and run the database update again.

Opening an Existing Entity

To open an existing control entity, click the **Open** icon in the toolbar. The default file type is .dob, which displays existing entity files. To open another file type, change the setting in the **Files of Type** field.

Copying and Pasting Definitions Between Jobs

You can copy selected definitions (such as selection definitions, internal items, extended internal items, and history items) from one job to another using standard windows copy and paste functions.

Note: Although you can copy and paste input sources from one Control Entity to another, be aware that detail fields will be copied, but the assigned internal or extended internal items will not. To include these items in the new Control Entity, you must re-enter the related data in the Internal Item or Extended Internal Item dialog box.

The copy and paste procedure is as follows:

1. Select the definitions that you wish to copy.
2. Click the **Copy** icon.
3. Open the job to which you want to copy the definitions.
4. Open or create the input source to which you want to copy the definitions.
5. In Input Source View, do one of the following:
 - If there are no definitions entered for this job yet, click **New** under **Extraction Definitions**.
 - If this job already has some definitions entered, click on the definition above which you want to paste the definitions from the other job.
6. Click the **Paste** icon.
7. If you clicked on existing definitions, a dialog box may open with the message **Are you sure you want to delete the item(s)?** Do one of the following:
 - To overlay the highlighted definition with the pasted definitions, click **Yes**.
 - To insert the pasted definition(s) above the highlighted definition, click **No**.

The definitions from your original job will be pasted into the current job.

Deleting Job Definitions

To delete individual job definitions, such as messages, internal items, extended internal items, history items, calculated items, or rules, in the Control Task (Job) view, highlight the definitions and press the **Delete** key. The system will display either a simple confirmation prompt or the Deletion Notification dialog box, as explained below.

Confirmation Prompt for Unreferenced Items

If the item definitions you highlighted for deletion are not referenced in the User Report, Free-Form Report, or Free-Form Output File, and are also not referenced in any of the job's history items, calculated items, messages, or rules, you will receive the following prompt:

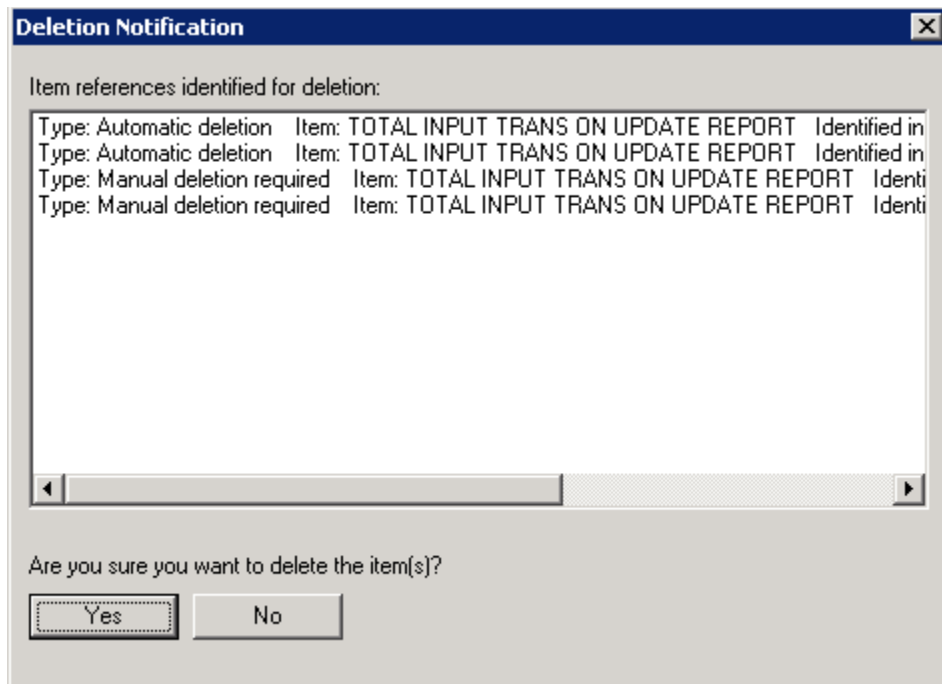
Are you sure you want to delete the item(s)?

To confirm the deletion, select **Yes**.

Deletion Notification dialog box for Referenced Items

If the item definitions you highlighted for deletion are referenced in the User Report, Free-Form Report, and/or or Free-Form Output File, or in history items, calculated items, rules, or messages, the Deletion Notification dialog box will appear.

An example of this dialog box is shown below:



3 ■ Setting Up a Job

Working with Job Definitions

This dialog box prompts you to confirm the item deletion at the bottom. Before confirming the deletion, read the information in the **Item references for deletion** field. This field lists referenced definitions that will be deleted automatically as well as those that you should delete manually. You can scroll to the right to see all of the information, which will be in one of the following formats:

Format 1: Automatic Deletion Information

"Type: Automatic deletion" item: XXX identified in report name"

Where:

XXX is the name of item to be deleted, and *report name* is the User Report, Free-Form Report, or Free-Form Output File

Lines in format 1 identify references to the item to be deleted that currently appear in the User Report or Free-Form Report. After you confirm the deletion, these item references will be deleted automatically.

Format 2: Manual Deletion Information

"Type: Manual deletion required" item: XXX identified in YYY"

Where:

XXX is the name of item to be deleted, and *YYY* is the name of item that references the item to be deleted

Lines in format 2 identify references to the item to be deleted that currently appear in the definitions for history items, calculated items, or rules. After you confirm the deletion, these item references must be deleted manually.

After reviewing the information, if you want to confirm the item deletion, first make a note of the item references that you must delete manually.

Deleting File Definitions

See ["Deleting File Definitions"](#) on page 162.

Setting Up Basic Job Information

As explained in “[Setting Up and Running a Job](#)” on page 64, you start creating your job by completing the Basic Entity Information options at the top of the Control Entity View. These settings are described in “[What is a Control Entity?](#)” on page 60.

You then open the Control Task (Job) View and complete the Basic Information folder for the job. This chapter provides an overview of these options. It contains the following sections:

- “[Completing the Basic \(Job\) Information Options](#)” on page 74
- “[Name](#)” on page 74
- “[Control Report Title](#)” on page 74
- “[Store History](#)” on page 74
- “[Set Return Code](#)” on page 75
- “[Print Control Report](#)” on page 75
- “[Print Recap Report](#)” on page 75
- “[Print Extracted Data Detail Report](#)” on page 75
- “[Print Sorted/Accumulated Data Detail Report](#)” on page 76
- “[Print History Data Detail Report](#)” on page 76
- “[Write Output File](#)” on page 77
- “[Auto Cycle Indicator](#)” on page 77
- “[Round Results](#)” on page 78
- “[Reconciliation Level](#)” on page 78
- “[Use 22-Character Numeric Format](#)” on page 78
- “[Initialize Text Items to Spaces](#)” on page 78
- “[Control Report XML Option](#)” on page 78
- “[Key Sort Order](#)” on page 79
- “[Adding Job Comments](#)” on page 79
- “[Display Processing Status](#)” on page 79
- “[Out-of-balance Key Tolerance](#)” on page 79

Completing the Basic (Job) Information Options

To specify an option, click on it to open and complete the corresponding dialog box.

Following is a summary description of each option and its dialog box. Click the **Help** button on the dialog box for field descriptions and additional information.

Name

The Control Task (Job) Name dialog box allows you to assign a meaningful name that is unique within the definition database to be used for this job in accordance with your naming conventions. In addition, if you are going to use multi-level reconciliation, you need to specify a level qualifier. For more information, see [“Using Multi-Level Reconciliation” on page 185](#).

Control Report Title

This field allows you to specify a descriptive title for the Control Report and, if printed, the Recap Report.

Store History

The Store History dialog box allows you to specify the following:

- Whether to store internal (or extended internal) item values obtained in this job in the history database so that they can be used in another run of the same job or in another job.
- If storing history, whether to allow prior cycle inserts.

The number of history cycles that will be stored per reconciliation key is specified through the History Runs option in Control Entity View.

Set Return Code

Use this dialog box to specify return code processing for the job and/or to specify a return code when no keys are selected. The return code is a 3- or 4-digit code that appears on the status bar when a job is run and in the Control Report. It indicates balancing rules that are out-of-balance or in error. If more than one rule detects an out-of-balance condition, the highest return code generated can be used as the return code for the job.

Use only 3 digits if you plan to use the \$? environment variable on UNIX or Linux, because this variable is limited to 3 digits.

Print Control Report

The Print Control Report dialog box enables you to set your preferences for printing the Control Report, which is the primary output of a reconciliation job. See [“Control Report” on page 271](#) for instructions on customization, report documentation, and a sample of the report.

Note: Be sure to read the information on customizing the report if you want the report to include extended internal items that have been added to a job created before Release 4.0.

You can also choose to print an out-of-balance summary whether or not you print the Control Report.

Print Recap Report

The Print Recap report dialog box enables you to set your preferences for printing the Recap Report. This report is a summary listing for all keys that were reconciled, showing the reconciliation results and the highest return code associated with that key. See Appendix B, “Reports” for a description and sample of the report.

Print Extracted Data Detail Report

The Print Extracted Data Detail Report dialog box enables you to set your preferences for printing this report, which shows the key and detail field values extracted from every input file record that met one or more selection criteria. This report is used for debugging your reconciliation job.

See Appendix B, “Reports” for a sample of this report.

Print Sorted/Accumulated Data Detail Report

The Print Sorted/Accumulated Data Detail Report dialog box enables you to set your preferences for printing the Sorted Data Detail Report or the Accumulated Data Detail Report.

These reports are used for debugging your reconciliation job.

The Sorted Data Detail Report contains the keys and detail field values extracted from every input file record meeting at least one selection criterion. Extracted data is shown after it has been sorted by reconciliation key.

The Accumulated Data Detail Report contains the extracted data shown on the Sorted Data Detail Report after it has been accumulated by reconciliation key.

See Appendix B, “Reports” for samples of these reports.

Print History Data Detail Report

The Print History Data Detail Report dialog box enables you to select your preference for printing the report.

The Print History Data Detail Report shows the results of accumulating the extracted data with the source histories. This provides you with a picture of what the history database will contain if the history updates are completed with no errors. This report is used for debugging your reconciliation job.

See Appendix B, “Reports” for a sample of this report.

Print Free-Form Report

The Print Free-Form Report dialog box enables you to do the following:

- Set your preferences for printing the Free-Form Report (as well as the report width and length), if you specify one by opening the Free-Form Report View in the Report Information section.
- Set your preferences for generating the Free-Form Report in XML format.

Free-Form Reporting allows you to completely customize a report showing reconciliation output for a job. If you want a Free-Form Report, you must create it as described in [Chapter 13, “Setting Up Free-Form Reports.”](#)

Write Output File

This sequential file can be output by the reconciliation process to be used as input to other programs, such as PC applications, including spreadsheets. For more information, see [“Setting Up an Output File” on page 81.](#)

Auto Cycle Indicator

This feature enables you to have the system automatically assign a cycle number to all of the reconciliation keys for this job. The cycle number is an 8-digit date or other ascending sequence number used to uniquely identify the set of data you are reconciling.

If you do not choose an option for the Automatic Cycle Format feature, the system defaults to None (no automatic assignment). In this case you must 1) specify that the cycle value will be extracted from the input source file using a key field or 2) assign the cycle number manually in the **Cycle Number** field on the Run Reconciliation dialog box when you run the job.

Round Results

The Round Results dialog box allows you to specify how the system handles decimals for the following types of values:

- Results from calculated items and rules.
- Values moved from extraction variables into internal/extended internal items or into other extraction variables.

Users have the ability to specify round or truncate criteria for Calculated Items and Balancing rules results both at the job level, and separate criteria individually during the setup of a Calculated Item or a Standard Rule.

Rounding options specified at the Calculated Item level or during the processing of the Standard Rules will override the rounding options specified at the Job level.

Reconciliation Level

This dialog box allows you to set up a multi-level reconciliation job. For more information, see [“Using Multi-Level Reconciliation” on page 185](#).

Use 22-Character Numeric Format

The Use 22-character Numeric Format dialog box allows you to determine whether to format counts and amounts in the Control Report, User Report, and Free-Form Report in 22-character format. Using 22-character format enables proper alignment of longer numbers that may include a currency symbol, commas, decimal point, sign, and up to 15 digits.

Initialize Text Items to Spaces

This dialog box enables you to control how empty internal items, extended internal items, and history items, which are formatted as text, will be initialized at the job level.

Control Report XML Option

This dialog box allows you to set a job-level option for generating the Control Report in XML format. This setting will override the corresponding option in your user options.

Key Sort Order

This option controls the sort order of the Freeform Output File (UNIDAT2) and several reconciliation reports. The reports include the Control Report, User Report, Free-Form Report, Sorted/Accumulated Data Detail Report, and History Data Detail Report.

Customers who may want to use this option include those who use dollar amounts as reconciliation keys and want the reports to show the highest dollar amount first.

Note: **Reconciliation Level Qualifier (RLQ) jobs:** The Key Sort Order settings from the base job will be applied to RLQ jobs. Likewise, any run-time overrides set for the Key Sort Order setting in the base job will be applied to RLQ jobs. Therefore, in an RLQ job, setting up a run-time override for the key sort order is not allowed. For more about RLQ jobs, see [Using Multi-Level Reconciliation on page 185](#).

Adding Job Comments

This option allows you to paste or type in up to 70,992 characters of your own comments regarding a job. To paste, use **CTRL + V**.

All of the comments will print when you generate a List Definitions Report that includes detailed job information.

Display Processing Status

This option allows you to monitor the progress of a job by displaying processing status information in the System Messages Report (SYSOUT).

Out-of-balance Key Tolerance

This option allows you to specify out-of-balance tally and accumulation information at the job level.

You can set a tolerance range for tallied or accumulated out-of-balance keys and choose to report the number of rules that are out of balance.

4 ■ Setting Up Basic Job Information

Out-of-balance Key Tolerance

Setting Up an Output File

This chapter provides information about the output file feature. This sequential file can be output by the reconciliation process to be used as input to other programs, such as PC applications, including spreadsheets. The chapter contains the following sections:

- “Choice of Output File Formats” on page 81
- “Specifying Where the File Will Be Generated” on page 83
- “Setting Up an Output File” on page 83
- “Record Layout for Original Output File” on page 84

Choice of Output File Formats

An output file can be generated in one of two formats:

Original Output File

Original Output File Features

The original output file (UNIDATA.DAT) is a preformatted hexadecimal file that includes the following reconciliation data:

- Reconciliation key
- Cycle number
- Run number
- Job name
- Step name
- Qualifier
- Return code
- Note area
- Internal item count
- Internal items
- Extended internal item count
- Extended internal items

5 ■ Setting Up an Output File

Choice of Output File Formats

Because most of the data is in packed format, you must run a utility before you can download the file to a PC application. See “[Create Comma Delimited Text File](#)” on page 241.

For the file layout, see “[Record Layout for Original Output File](#)” on page 84.

Freeform Output File

This format (UNIDAT2.DAT) offers enhanced features, as follows:

Additional Data for Freeform Format

This format lets you include all of the data you can include in the original format along with the following additional information:

- History items
- Calculated items
- Rule information, including the rule return code, message, and (for a standard rule) the variance .
- Job information, including the job ID, run date, and run time
- Reconciliation key information, including the overall return code, message, and cycle number

Other Enhanced Features for Freeform Format

In addition to the information listed above, you will have the ability to do the following:

- You can have the file generated in delimited format, and optionally suppress the delimiter after any field.
- You can specify a header for any item.
- For numeric items, you can include only part of the value. You can also have the data unpacked, include a decimal, reverse the sign, suppress the positive sign, suppress leading zeros, and replace leading zeros with blanks.
- For text items, you can include only part of the text and customize the justification.
- For dates, you can specify whether to include a date separator.

Specifying Where the File Will Be Generated

The output file is generated when you run reconciliation. You can specify the location where the file will be generated as shown below.

	Windows	UNIX or Linux
Original Output File	Location is specified with the dd_unidata statement.	Location is specified with the dd_unidata statement in the udptest.sh script in the uds2000.bat file.
Freeform Output File	Location is specified with the dd_unidat2 statement.	Location is specified with the dd_unidat2 statement in the udptest.sh script in the uds2000.bat file.

Setting Up an Output File

The procedure is as follows:

1. From the Control Task (Job) Window, select **Basic Information > Write Output File** and complete the Write Reconciliation Output File dialog box. If you specify the Original Output File as the format, skip to [step 8](#). Otherwise, complete the next steps.
2. Click **Output File** in the Report Information folder to open the Output File view.
3. If you want the option to specify a header for any item, select **Generate Header** and complete the Output File Header dialog box.
4. If you want the output file to be in delimited format, select **Generate Delimiter** and complete the Output File Delimiter dialog box.
5. In the Data Fields Folder, click **New** to display the Output File Data Field dialog box. Here you can specify and customize the first item or information type whose value should be included in the file.
6. Repeat [step 5](#) until you have included all of the data you want in the file.
7. Save your work.
8. When you run the job, the output file will be generated in the location you specified (see [“Specifying Where the File Will Be Generated” on page 83](#)).

5 ■ Setting Up an Output File

Record Layout for Original Output File

Record Layout for Original Output File

Record Layout Part 1: Positions 1-247

The following table shows the layout for the first 247 positions of the record.

Record Layout for Original Output File: Part 2

The following tables shows the starting points for regular and extended internal items, respectively. Alphanumeric data is left justified and numeric and packed data is right justified within an item's positions.

Regular Internal Items Table

A UNIDATA file can have up to 999 regular internal items. The starting points for the first 50 regular internal items are shown in the table below. To determine the starting position of an item whose number is greater than 50, use the following formula: (Item number x 8) + 240.

Int. Item No.	File Pos.	Int. Item No.	File Pos.	Int. Item No.	File Pos.	Int. Item No.	File Pos.	Int. Item No.	File Pos.
1	248	11	328	21	408	31	488	41	568
2	256	12	336	22	416	32	496	42	576
3	264	13	344	23	424	33	504	43	584
4	272	14	352	24	432	34	512	44	592
5	280	15	360	25	440	35	520	45	600
6	288	16	368	26	448	36	528	46	608
7	296	17	376	27	456	37	536	47	616
8	304	18	384	28	464	38	544	48	624
9	312	19	392	29	472	39	552	49	632
10	320	20	400	30	480	40	560	50	640

Extended Internal Items Table

A UNIDATA file can have up to 100 extended internal items. The starting points for the first 10 items are shown in the table below. To determine the starting position of an item whose number is greater than 10, use the following formula: (Extended Item number x 80) + 8160

Item No.	File Pos.
1	8240
2	8320
3	8400
4	8480
5	8560
6	8640
7	8720
8	8800
9	8880
10	8960

5 ■ Setting Up an Output File

Record Layout for Original Output File

Setting Up Job Definitions

Job definitions specify how values extracted from input sources are processed. This chapter provides information for setting up several types of job definitions.

Typically, you will complete the Basic Information folder options as described in [Chapter 4, “Setting Up Basic Job Information on page 73”](#) before you set up the job definitions described here.

Information on item numbering, key masking, and decimal specification, all of which are related to job definitions, is also provided here.

This chapter includes the following sections:

- [“Understanding Internal Items and Extended Internal Items” on page 88](#)
- [“Defining Internal Items” on page 90](#)
- [“Defining Extended Internal Items” on page 90](#)
- [“Defining Extraction Variables” on page 91](#)
- [“Defining History Items” on page 94](#)
- [“Defining Calculated Items” on page 95](#)
- [“Defining Rules” on page 97](#)
- [“Defining Messages” on page 106](#)
- [“Using Batch Options to Control Item Numbers” on page 107](#)
- [“Using Key Masks” on page 109](#)
- [“Constraints for Single-precision and Double-precision Numbers” on page 113](#)

Understanding Internal Items and Extended Internal Items

This section explains internal items and extended internal items.

Characteristics Shared by Regular and Extended Internal Items

Internal items (regular and extended) are used to store control values obtained when your file definitions are processed.

A control value can be either of the following:

- A value obtained during extraction (a count, amount, text item, date, or time).
- The result of a calculated item. Calculated item results are values obtained by adding, subtracting, multiplying or dividing the values of other items.

Both regular and extended internal items can be used in calculated items and rules and, if the job stores history, will be stored in the history database so that you can define them as history items for reuse in a later run of the same job or of another job.

Differences Between Regular and Extended Internal Items

Internal Items

- You can use up to 999 internal items per reconciliation job.
- In the interface, internal items are identified with an I. For example, I-001, I-002, etc.
- Internal items cannot be used to store lengthy extracted values. For example, only 15 bytes of numeric data or 8 bytes of text can be stored in an internal item.

Extended Internal Items

- You can use a maximum of 100 extended internal items per reconciliation job.

Note: You can use up to the maximum number of both internal and extended internal items in the same job.

- Extended internal items are identified with an X. For example, X-001, X-002, etc.

- Extended internal items enable you to store significantly longer control values than regular internal items. These longer values are called extended data. Extended data simplifies job creation when working with lengthy values by reducing the number and complexity of the definitions required. This is because extended data can be used just like regular data in calculated items, history items, and rules.

Restrictions for Extended Data

- Selection fields, and keys do not support extended data.
- Storage of extended internal items in the history database may impact performance.

Maximum Lengths for Regular and Extended Data

The following table shows the maximum lengths for regular vs. extended data in each supported format.

Format of Extracted Data ¹	Maximum Length	
	Regular Data	Extended Data
Numeric ²	A maximum of 15 digits can exist within the specified length.	A maximum of 30 digits can exist within the specified length.
Alphanumeric ³	8	80
Packed decimal ³	8	16
Zoned decimal (Overpunch) ²	A maximum of 15 digits can exist within the specified length.	A maximum of 30 digits can exist within the specified length.
Binary	8	8
Unsigned packed	8	16

1 - The data formats listed in column 1 are described in [Understanding Field Formats on page 146](#)

2 - For numeric and zoned decimal (overpunch) values, 80 positions can be identified for extraction, but only 15 or 30 digits can exist within these positions. Additional digits will not be evaluated.

3 - Packed decimal and alphanumeric are the only valid formats for values stored in extraction variables.

Defining Internal Items

Internal items and extended internal items were described in the preceding sections.

Internal items are created through the Internal Item dialog box. This dialog box can be accessed in three ways:

- You can create one or more internal items in Control Entity View by clicking **New** under the Common Internal Items folder. This will enable you to set up internal items before defining a control task. Any internal item created in this way can be used in any control task in the entity.
- You can create an internal item from the Control Task (Job) View by clicking **New** in the Internal Items folder and completing the dialog box.
- You can create or select an internal item from the Detail Field dialog box. If you set the **Target Area** to Internal Item, you can do either of the following:
 - Click **Create/Change** and click **New** to create a new internal item.
 - Select an existing internal item in the **Internal Item** field.

Defining Extended Internal Items

Internal items and extended internal items were described in the preceding sections.

Extended internal items are created through the Extended Internal Item dialog box. This dialog box can be accessed in three ways:

- You can create one or more internal items in Control Entity View by clicking **New** under the Common Extended Internal Items folder. This will enable you to set up extended internal items before defining a control task. Any extended internal item created in this way can be used in any control task in the entity.
- You can create an extended internal item from the Control Task (Job) View by clicking **New** in the Extended Internal Items folder and completing the dialog box.
- You can create or select an extended internal item from the Detail Field dialog box. If you set the **Target Area** to Extended Internal Item, you can do either of the following:
 - Click **Create/Change** and click **New** to create a new extended internal item.

- Select an existing extended internal item in the **Extended Internal Item** field.

Defining Extraction Variables

Extraction variables are items you define to store regular or extended data from file definitions or external translation table definitions for further processing. You can specify up to 999 extraction variables per control entity.

Specifying Extended Formatting for an Extraction Variable

An extraction variable should be in extended format (that is, it should be an extended extraction variable) if you plan to use it to store a packed decimal number that is 9 to 16 bytes long (16 to 30 digits) or text that is 9 to 80 characters long.

To indicate an extended extraction variable, set the Extended field to Y on every file definition and/or external translation table definition that references it. In addition, as explained in “[Setting Up Extraction Variables](#)” on page 93, you should indicate in the description whether the extraction variable is in extended format.

Consistency in Specifying Extended Format for an Extraction Variable

When setting the Extended indicator for an extraction variable, you must be consistent. For example, for extraction variable LAST NAME, you should not select the check box for **Use extended format** in one file definition, and later, when you reference LAST NAME in another file definition in the same control entity, leave the check box blank.

Warning: If you are not consistent in setting the **Use extended format** field when referencing the same extraction variable multiple times, when you run the job, a run-time (#U) error will be generated, and the value of the extraction variable will be set to spaces (if the format is text) or zeroes (if the format is packed decimal).

For more information about extended data, see “[Maximum Lengths for Regular and Extended Data](#)” on page 89.

6 ■ Setting Up Job Definitions

Defining Extraction Variables

Extraction Variable Definitions

An extraction variable definition consists of its name and description. The extraction variables that you define in extraction definitions at the control task (job) level or in external translation table definitions at the control entity level will automatically be classified as common extraction variables and will appear under Common Extraction Variables in the Control Entity view. If you copy and paste an extraction variable in Control Task (Job) View, a corresponding common extraction variable will be built.

Storing and Saving Extraction Variables

To store the value of an extraction variable in history, use a detail field to assign it to an internal or extended internal item.

The extraction variables defined when creating file definitions will be saved when you save the job.

The extraction variables defined when creating an external translation table will be saved as part of table definitions and also saved as common extraction variables in the control entity.

To store the value of an extraction variable in history, use a detail field to assign it to an internal or extended internal item.

Automatic Generation of Extraction Variable Definitions

When you open a job created before Release 4.1, definitions for the extraction variables specified in the file definitions will be generated automatically and added to the Job Window. Saving the job will move them to the definition database.

When you open an external translation table created before Release 4.1, definitions for the extraction variables specified in the table definitions will be generated automatically. Saving the table will move them to the definition database.

In either case the automatically generated extraction variable description will default to Vnnn, where V stands for extraction variable and nnn is the extraction variable number. You can change the automatically descriptions.

Setting Up Extraction Variables

Setting Up Extraction Variables from Control Entity View

Procedure:

1. Select **Common Extraction Variables > New**.
2. Complete the Common Extraction Variable dialog box.

Note: We recommend that you indicate in the **Name** field whether the extraction variable is in extended format. (You specify the format when you reference the variable in a file or extended translation table definition.) When setting the Extended indicator, you must be consistent. See [Consistency in Specifying Extended Format for an Extraction Variable](#) on page 91.

Setting Up Extraction Variables from the Control Task (Job) View

Do one of the following:

- **Typical method:** When you are referencing an extraction variable in a file definition, you can select an existing extraction variable from the list or select **Create/Change** to create a new extraction variable. then select **New** and complete the Extraction Variable dialog box. See the note in [step 2](#) in the previous section about completing the **Name** field.
- In the Control Task (Job) View, select **New** under Extraction variables and complete the Extraction Variable dialog box. See the note in [step 2](#) in the previous section about completing the **Name** field.

Setting Up Extraction Variables from the External Translation Table View

When you are referencing an extraction variable in translation table definition, you can select an existing extraction variable from the list or select **Create/Change** to create a new extraction variable. then select **New** and complete the Extraction Variable dialog box. See the note in [step 2](#) in "Setting Up Extraction Variables from Control Entity View" above about completing the **Name** field.

Defining History Items

What is a History Item?

A history item specifies a value to be retrieved that was extracted in a previous run of the current reconciliation key or another specified key and stored in the history database. The retrieved value can be an internal or extended internal item, job run date, job run time, or cycle number.

A maximum of 999 history items can be defined in a job.

A history item is defined through the History Item dialog box, which is accessed from the History Items folder in the Control Task (Job) View.

Note: When you define a history item, calculated item or rule, the system assigns it a consecutive number starting with 1. You can view and change these numbers through the Batch Options dialog box.

You can insert filler records if you need to work with non-consecutive item numbers.

For more information on batch options and filler records, see [Using @@FILLER Records on page 259](#).

Setting Up History Items

To set up a history item, click **New** in the History Items folder in Control Task (Job) View to display the History Item dialog box.

Note: If you are defining a history item stored for a reconciliation key other than the current key, or if you want to use key masking, select **Use Key/Mask and/or variable cycle processing**, click **Key/Mask** to display and complete the History Key/Mask dialog box. For more information on key masking, see [Using Key Masks on page 109](#).

Defining Calculated Items

Understanding Calculated Items

A calculated item specifies mathematical manipulation of values from internal (or extended internal) items, history items, other calculated items, literals (numeric constants or text), or functions. Calculated items can be used in reconciliation rules in the same way as internal items, extended internal items, and history items.

When a non-conditional calculated item is used in a reconciliation rule, the calculation is performed before the rule is evaluated. A conditional calculation is performed after rules are evaluated. See below for more on conditional calculated items.

Up to 999 calculated items can be specified per job ID.

Formula

The formula for computing the value of a calculated item is defined as the arithmetic expression:

Left-hand side Operator Right-hand side

The operator specifies how the left-hand side relates to the right-hand side.

Note: When you define a history item, calculated item or rule, the system assigns it a consecutive number starting with 1. You can view and change these numbers through the Batch Options dialog box.

You can insert filler records if you need to work with non-consecutive item numbers.

For more information on batch options and filler records, see [Using @@FILLER Records on page 259](#).

Conditional Calculated Items

Note: For more information about the tabs and fields in this section, see [“Setting Up Calculated Items”](#) below.

A calculated item can be conditional. To make it conditional, select the **Conditional Calculated Item** on the Calculated Items dialog box. This means it will be processed only if one of the following is true:

- The standard rule that references it in the Standard Rule dialog box - General Information tab's **Conditional Calculated item to process** field is processed and is in balance.

6 ■ Setting Up Job Definitions

Defining Calculated Items

- The conditional rule that references it in the Conditional Rule dialog box - General Information tab's **Conditional Calculated item to process** field is processed and the condition is met. See the section below for the messages that may be generated.

Messages Generated When a Calculated Item is Not Processed

When a conditional calculated item is not processed, one of the following messages will appear in the Control Report:

- **BYPASSED.** This indicates that 1) the standard rule that referenced the calculation was not in balance, or 2) the conditional rule that referenced the calculation was processed but the condition was not met.
- **NOT PROCESSED.** This indicates that the **Conditional Calculated Item** field on the Calculated Item dialog box - General Information tab was selected but the item was not referenced in a rule.
- **INACTIVE.** This indicates that the standard or conditional rule in which the item was referenced remained inactive.

Setting Up Calculated Items

A calculated item is set up through the Calculated Item dialog box, which is accessed from the Calculated Items folder in the Control Task (Job) View. The dialog box has two tabs:

- The General Information tab, which is used to identify and format the item and to set the **If item is not found** value. For more information on this indicator, see [“Rule Processing When Item Is Not Found” on page 104.](#)
- The Formula tab, which defines the computation.

Round Results

Select an option to compute the decimal value.

Rounding options specified at the calculated item level will override the rounding options specified at the Job level.

Valid options are **Default, Yes, No.**

Select **Default** to use the rounding options specified at the Job level. Use **Yes** to round the decimal value to the number of Decimals specified. Use **No** to truncate the decimal value at the specified decimal position.

The dialog box help provides field definitions and other information.

Defining Rules

What Are Rules?

Rules define the criteria to be applied to the values of internal items, extended internal items, calculated items, and/or history items to determine if values extracted from an input source are in or out of balance.

A single job may need to have multiple rules defined to determine if a variety of values are in balance. A maximum of 100 rules can be defined in a job.

Typically, you will want to have a rule set a return code for an out-of-balance condition and create a text message associated with the return code. Return codes and the messages you associate with them appear on the Control Report.

Two types of rules can be created:

- “Standard Rules” on page 97
- “Conditional Rules” on page 99

Standard Rules

A standard rule compares values from any combination of internal, extended internal, calculated, and history items to determine if they are in- or out-of-balance.

The comparison for a standard rule has the following format:

Left-hand side Operator Right-hand side

You can optionally use, on one side or the other (but not both) of the comparison, either a literal or a numeric constant. Standard rules can be either active or inactive.

Round Results

Select an option to compute the decimal value.

Rounding options specified at the rule level will override the rounding options specified at the Job level.

Valid options are **Default**, **Yes**, **No**.

Select **Default** to use the rounding options specified at the Job level. Use **Yes** to round the decimal value to the number of Decimals specified. Use **No** to truncate the decimal value at the specified decimal position.

6 ■ Setting Up Job Definitions

Defining Rules

Active Standard Rules

An active standard rule defines criteria for a comparison (using items, literals, and constants as described above) that is evaluated when reconciliation is performed. The status of an active standard rule can also be evaluated by conditional rules. If the result is out-of-balance, a standard rule can perform one of the following actions:

- Set the return code as specified in the Set Return Code option on the Basic Information screen.
- Set the return code of your choice as the completion code for the job if this rule is out-of-balance. A 3- or 4-digit return code can be assigned. Use 4-digit codes between 0050 and 3999 to avoid duplicating system-generated codes. **Exception:** We recommend that you use 3 digits if you plan to use the \$? environment variable on UNIX or Linux. This variable is limited to 3 digits. If more than one rule is out-of-balance, the highest code will be set.
- Trigger an abend of the reconciliation job.

Following are examples of active standard rules:

I-001 = X-002, set an abend code

I-001 > +100, set a return code of 3075

I-001 + C-001 < E-003, set a return code of 3200

If the rule is in-balance, activate a conditional calculated item.

Inactive Standard Rules

Inactive standard rules are not evaluated each time a job is run. They are used when control items must be compared under specific circumstances.

Inactive standard rules can only be evaluated within a conditional rule. The criteria of the condition is defined by the conditional rule.

Inactive standard rules can:

- Evaluate control items in a reconciliation job. Using an inactive standard rule in the left side of a conditional rule equation will evaluate the criteria defined by the inactive standard rule.
- Perform an action in a reconciliation job. Using an inactive standard rule in the right side of a conditional rule equation activates the rule, causing it to perform as an active standard rule.

Following are examples of inactive standard rules:

R001: I-002 > \$50,000

R002: I-011 > +90

Conditional Rules

What Is a Conditional Rule?

A conditional rule uses IF/THEN logic to evaluate the status of up to 10 other rules to determine if a condition is met. You can specify whether the condition should be true or false.

Valid Conditions

When evaluating a conditional rule:

- The valid condition for a standard rule is assumed to be **in-balance**.
- The valid condition for a conditional rule is assumed to be **condition met**.

The negate \neg [NOT] symbol (see "NOT" below) can be used to change the valid conditions.

AND and OR

The rules to be evaluated in the IF portion of the rule are specified on the Conditional Rule screen. Rules are linked by + [AND] and | [OR] symbols.

Each rule on the IF side is evaluated to determine whether the condition is met. For example, here is Rule 4 (R4), an active conditional rule:

Left-hand side	Right-hand side
IF R1 + R2 True	THEN process R3

The above rule is interpreted as:

If (Rule 1 is in balance) and (Rule 2 is in balance) are true, then process Rule 3.

NOT

The negate \neg [NOT] symbol can be used before a rule to change the valid conditions. If the negate symbol is used:

- The valid condition for a standard rule is assumed to be **out-of-balance**.
- The valid condition for a conditional rule is assumed to be **condition not met**.

Following is an example where the \neg [NOT] symbol is used with R2:

Left-hand side	Right-hand side
IF R1 + \neg R2 True	THEN process R3

6 ■ Setting Up Job Definitions

Defining Rules

Assuming that R2 is a standard rule, the above rule can be interpreted as:
If (Rule 1 is in balance) and (Rule 2 is **out-of-balance**) are true, then process Rule 3.

Active Conditional Rules

Active conditional rules are evaluated every time the job is run. The status of up to 10 other rules is evaluated to determine if a condition is met. You can specify whether the condition should be true or false.

The THEN (right-hand side) portion of the rule defines the action to be taken if the condition is met, which can be either of the following:

- Execute Process Rules. This activates up to 10 inactive standard rules.
- Set Return Code. This sets the return code shown in the **Return Code** field on the Conditional Rule screen.

Additionally, if the rule references a conditional calculated item, the conditional calculated item will be processed if the condition is met.

Inactive Conditional Rules

Inactive conditional rules are intended for use within active conditional rules as a means to associate subsets of rules within a rule. They take the place of the logical expressions (parentheses). That is, the rule number of the inactive conditional rule is used in place of the logical expression.

Inactive conditional rules are used when you need to:

- Check the status of more than two rules.
- Use both AND and OR logical expressions to describe a condition.
- Evaluate more than 10 rules in an active conditional rule.

In an inactive conditional rule, the status of a rule is evaluated to determine if a condition (which can be true or false) is met. The THEN (right-hand side) portion of the rule does not trigger an action. The THEN portion simply sets the status of the evaluation to condition met or condition not met.

How Conditional Rules Are Evaluated

This section provides a detailed explanation of how conditional rules are processed.

If True and If False Logic in Conditional Rules

The following tables show active conditional rules and what they mean depending on whether you specify that the condition described in the Evaluate field on the Conditional Rule screen must be true or false.

The rules used in the sample If statements in the tables are standard rules, so their status is described as in or out of balance. If these were inactive conditional rules, their status would be described as condition met or not met. If they were active conditional rules, their status would be action taken or action bypassed.

This table shows what will happen when the TRUE logic is applied:

If False	Then	Description	What Will Happen
R1	Process R3	If “rule 1 is in balance” is true, then process rule 3.	If Rule 1 is in balance, Rule 3 will be processed.
¬R1	Process R3	If “rule 1 is not in balance” is true, then process rule 3.	If Rule 1 is out-of-balance, Rule 3 will be processed.
R1 + R2	Process R3	If “rule 1 is in balance and rule 2 is in balance” is true, then process rule 3.	If Rule 1 AND Rule 2 are in balance, Rule 3 will be processed.
R1 + ¬R2	Process R3	If “rule 1 is in balance and rule 2 is not in balance” is true, then process rule 3.	If Rule 1 is in balance AND Rule 2 is out-of-balance, Rule 3 will be processed
R1 R2	Process R3	If “rule 1 is in balance or rule 2 is in balance” is true, then process rule 3.	If Rule 1 OR Rule 2 are in balance, Rule 3 will be processed.
R1 ¬R2	Set return code 3010	If “rule 1 is in balance or rule 2 is not in balance” is true, then set return code 3010.	If Rule 1 is in balance OR Rule 2 is out-of-balance, return code 3010 will be set.

6 ■ Setting Up Job Definitions

Defining Rules

This table shows what will happen when the FALSE logic is applied:

If False	Then	Description	What Will Happen
R1	Process R3	If “rule 1 is in balance” is false, process rule 3.	If rule 1 is out of balance, rule 3 will be processed.
$\neg R1$	Process R3	If “rule 1 is not in balance” is false, process rule 3.	If rule 1 is in balance, rule 3 will be processed.
$R1 + R2$	Process R3	If “rule 1 is in balance and rule 2 is in balance” is false, process rule 3.	If rule 1 OR rule 2 is out of balance, rule 3 will be processed.
$R1 + \neg R2$	Process R3	If “rule 1 is in balance and rule 2 is not in balance” is false, process rule 3.	If rule 1 is out of balance OR rule 2 is in balance, rule 3 will be processed.
$R1 R2$	Process R3	If “rule 1 is in balance or rule 2 is in balance” is false, process rule 3.	If rule 1 AND rule 2 are out of balance, rule 3 will be processed.
$R1 \neg R2$	Set return code 3010	If “rule 1 is in balance or rule 2 is not in balance” is false, set return code 3010.	If rule 1 is out of balance AND rule 2 is in balance, return code 3010 will be set.

If you look at the “What Will Happen” column in the table above, you will notice that AND and OR logical expressions are evaluated differently in these If False statements than in the If True statements shown in the previous table.

AND Logical Expressions

When an AND logical expression is evaluated within an If True statement, the expression will be true if *every* condition connected by the AND is true. But an If False statement will be false (and the actions you have specified for the rule will be performed) if *any* condition connected by the AND is false.

The following If True statement has the same effect as the “If $\neg R1 | \neg R2$ is false” statement:

If $R1 + R2$ is true, then set return code 3010.

It is easy to see in the above statement that if BOTH rules are IN BALANCE, the rule will set return code 3010.

Avoiding AND and OR Connectors in the Same IF Statement

IF statements that combine AND and OR connectors can be difficult to read and maintain. Here is an example of a potentially confusing AND/OR combination:

IF R1 | ¬R2 + R4 is false, then set return code 3010.

This rule will take action IF

(“Rule 1 is in balance” is false OR “Rule 2 is out of balance” is false)

AND “Rule 4 is in balance” is false.

Notice that the statement is evaluated from left to right. ACR/Detail automatically evaluates the first logical expression first: R1 | ¬ R2. Then, it evaluates the entire IF statement as an AND logical expression consisting of the true or false result of evaluating the first logical expression, AND R4. To keep your rules as clear and simple as possible, you can use an inactive conditional rule to replace either the AND condition or the OR condition if they are in the same IF statement.

OR Logical Expressions

Within an If True statement, an OR logical expression will be true if *any* condition connected by the OR is true. For example, the following rule contains an OR logical expression that connects two conditions:

If (R1 + R2) | R3 is true, then set return code 3010.

In the rule above, “Rule 1 is in balance AND Rule 2 is in balance” is one condition and/or “Rule 3 is in balance” is the other. If either or both of these conditions are true, the rule will take action and set return code 3010.

In contrast, when you use an OR logical expression within an If False statement, every condition connected by the OR must be false for the rule to take action. For example, the following rule will take action if Rule 1 is out of balance AND Rule 2 is in balance:

If R1 | ¬R2 is false, then set return code 3010.

Here is another true versus false example:

If ¬R1 | ¬R2 is true, then set return code 3010.

In the rule above, if EITHER or both Rule 1 and Rule 2 are OUT OF BALANCE, the rule will set return code 3010. But what will happen if we change true to false?

If ¬R1 | ¬R2 is false, then set return code 3010.

6 ■ Setting Up Job Definitions

Defining Rules

The rule will set return code 3010 only if BOTH rules are IN BALANCE.

Note: To avoid confusion as to how a lengthy IF statement will be evaluated, it is best to define an inactive conditional rule for each logical expression that you would normally enclose in parentheses. Then, combine these inactive rules together in a single, active conditional rule.

Rule Processing When Item Is Not Found

Each internal item, extended internal item, history item, and calculated item contains the indicator **If item Is not found**, which can be set to one of the following values:

- Set all rules involving this item out-of-balance.
- Skip all reconciliation rules involving this item.
- Set this item to zero/spaces and continue processing.

When items within a rule are processed, if no items are missing, the rule uses the value of each item to determine whether the rule is in- or out of balance.

When items within a rule are missing, the indicator value for each missing item is stored. The indicator with the highest order of precedence determines how to process the rule.

Order of Precedence

The order of precedence referenced above is as follows:

1. Set all rules involving this item to out of balance. The item is required for an in-balance condition.
2. Skip all reconciliation rules involving this item, including conditional rules that evaluate or process rules that involve this item.
3. Set this item to zero (if it is numeric) or spaces (if it is text) and continue processing. The item is not required for an in-balance condition.

Reporting Rule Status

The Control Report indicates the status of rules as follows:

- Active standard rules—These are reported as in balance or out-of-balance.
- Inactive standard rule—Reported as in balance or out-of-balance only if the rule was made active and therefore evaluated. (The status of an inactive standard rule that was not evaluated or made active by a conditional rule will not affect the in- or out-of-balance result of a job step.) You can set up the User Report and Free-Form Report to display the status of inactive standard rules if they were not made active.
- Active conditional rules—Reported as Action Taken (if the condition is met) or Action Bypassed (if the condition is not met).
- Inactive conditional rules—Reported as Condition Met or Condition Not Met.

Setting Up Rules

To create a standard rule, from the Control Task (Job) View, click **New** and select Standard Rule to display the Standard Rule dialog box. Standard rules must have already been defined before you can create a conditional rule, so if you plan to create both types of rules, create the standard rules first.

To create a conditional rule, click **New** and select Conditional Rule to display the Conditional Rule dialog box.

The dialog box help provides field definitions and other information. The help for the Condition tab includes the procedure for completing the tab.

Defining Messages

Messages are lines of instructional text, up to 80 characters long, that are associated with a return code. When you set up a balancing rule, you specify a return code to indicate an out of balance condition. If the rule issues the return code, or if the return code is 0000, the system prints the associated messages.

Messages serve the following purposes:

- Identify an out of balance condition and outline correction procedures
- Provide reminders or explanations, and include specific information, such as a contact name and phone number or an exact item value and description.

Message Processing

All messages print on the Control Report, and if specified, on the Free-Form and User Reports. Messages associated with a 0000 return code print whether the rule is in or out of balance. If the system cannot find any messages associated with the return code, the message NONE prints in the Messages section of the Control Report, and processing continues. If you included an item value or description within the message, the printed text is expanded to 112 characters.

For jobs run on z/OS, you can choose to write messages to the console, with or without a required response. For jobs run on other platforms, you can choose to write messages to the System Messages (SYSOUT) Report.

For jobs run on Windows, you can send messages to an email address, and attach one or more of the following reports: Control Report, SYS-OUT Report, Recap Report, User Report, and Free-Form Report. Specify email notification settings on the Server Properties dialog in ACR/Summary and ACR/Detail server.

Setting Up Messages

If you created a rule and defined a new return code for the rule, the Message dialog box opens automatically, enabling you to define a message. To edit an existing message or create a message at a later time, from the Control Task (Job) View, select the message to edit or select **New**.

The message dialog box enables you to add up to 80 characters of text. You can include the value for internal, extended internal, history, or calculated items, or the name and description for internal, extended internal, and history items. This option expands the printed text to 112 characters.

If you need more characters, create additional messages for the same return code.

Message Examples

The following message provides notification:

```
NOTE: NO INPUT FROM PAB20311 TODAY
```

The following message includes contact information for an out of balance condition:

```
OUT OF BALANCE BY OVER $100 -- PROCESSING HALTED. CALL BRENDA  
JONES, 527-8990
```

The following message includes the exact item value, which allows the user to determine the severity of an out of balance condition:

```
THE INVENTORY BUDGET IS $300K. CURRENT VALUE IS $377,102.00.  
WE ARE OVER BUDGET!
```

Using Batch Options to Control Item Numbers

By default, the history items, calculated items, and rules you create are assigned consecutive item numbers. The first rule you create will be Rule 1, the next will be Rule 2, etc. The item numbers are generated when you update the definition database.

Also by default, these item numbers are not displayed in ACR/Detail Client.

The Batch Options dialog box (**Options > Batch Options**) enables you to control the sequence or display/hide the numbers, as described in the following section.

Control the Sequence of the Item Numbers for History Items, Calculated Items, and Rules

Pattern Assignment

A different numbering pattern can be assigned to each of the above item types as follows:

- Enter only 1 number to increment all subsequent item numbers by the specified amount. Examples:

Enter 2 to generate 2, 4, 6, 8,...

Enter 4 to generate 4, 8, 12, 16, ...

- Enter multiple numbers in any ascending order to assign an item number sequence based on the difference between the last two numbers you enter. Examples:

Enter 1, 2, 5, 10 to generate 1, 2, 5, 10, 15, 20, ...

Enter 1, 3 to generate 1, 3, 5, 7, 9, ...

Item Number Generation Error

When entering a pattern, make sure it does not result in an attempt to assign a sequence number that is higher than the maximum supported.

For example, the maximum number of 999 history items can be specified in a job. Suppose you assign the sequence numbers 100, 200, 300, 400 for history items, and you create 10 history items. Your generated item numbers would be 100, 200, 300, 400, 500, 600, 700, 800, 900, 1000. The last item number, 1000, would be larger than the maximum number, 999, allowed. When you attempted to update the definition database, the following message would appear: “The number of history items in Control Task will not fit into the item number sequence specified. How do you wish to proceed.”

Select one of the following:

- **Ignore**—Update the definition database using the default sequence numbers 1, 2, 3, etc. instead of the sequence identified in Batch Options.
- **Change**—Display the Batch Options dialog. This will allow you to change your sequence numbers.
- **Cancel**—Cancel the request to update the definition database.

Display/Hide Item Numbers

The batch options also enable you to display or hide the item numbers generated for history items, calculated items, and rules.

Using Key Masks

What is a Key Mask?

Because the history database stores records based on a combination of reconciliation key and cycle (run) number, records for identical reconciliation keys having the same run number cannot be stored.

The key mask feature allows you to replace values at specific positions in an extracted reconciliation key with constants. By assigning an application-specific identifier to all of the keys extracted in a particular job, key masking enables you to store records from different applications, each of which would otherwise have an identical reconciliation key, separately in the same history database.

A key mask consists of 1 to 40 characters composed of asterisks and other characters (for example, ABC*D**EF*). When the job runs, the key mask is superimposed onto the original reconciliation keys extracted. Any trailing non-masked positions in the original reconciliation key are set to blanks.

Key Masking at the Entity Level

Use the following instructions to set up key masking in extraction definitions:

1. Add a common key segment to hold the key mask: In the Control Entity view, double click on the last available key under Basic Entity Information. The Key Name dialog box opens.
2. Enter a key name such as MASK in the **Key** field to indicate that this is the reconciliation key segment that will hold your key mask.
3. Enter the appropriate key mask in the **Mask** field.

6 ■ Setting Up Job Definitions

Using Key Masks

Example of Key Masking in Extraction Definitions

Suppose you have two applications, Payroll and Personnel. Job A, the reconciliation job for Payroll, and Job B, the reconciliation job for Personnel, have the same reconciliation key, Employee Number. Assume you want to be able to store values from each job separately in the history database. You could do this by defining a key mask to prefix all reconciliation keys extracted for each job with an application identifier (PAYROLL or PRSNNEL).

Example of a key mask for Job A, a Payroll reconciliation job:

Columns	0-----1-----2
Original Reconciliation Key	1234567890123
Reconciliation Key Mask	PAYROLL*****
Resulting Reconciliation Key	PAYROLL890123

Example of a key mask for Job B, a Personnel reconciliation job:

Columns	0-----1-----2
Original Reconciliation Key	1234567890123
Reconciliation Key Mask	PRSNNEL*****
Resulting Reconciliation Key	PRSNNEL890123

In this example, the values extracted for the reconciliation key PAYROLL890123 in Job A and the values extracted for the reconciliation key PRSNNEL890123 in Job B can be stored on separate records in the same history database for use in other runs of the same job or in other jobs.

Example of Key Masking in History Items

If you have used key masking as in the [“Example of Key Masking in Extraction Definitions”](#) on page 110, you will be able to define a history item to extract values from the history database based on the application (Payroll or Personnel) to be used in the current job.

The following key mask could be used to define a history item that extracts only records from the Payroll application:

```
PAYROLL*****
```

The following key mask could be used to define a history item that extracts only records from the Personnel application:

```
PRSNNEL*****
```

Procedure

To set up a key mask for a history item, do the following:

1. From the History Item dialog box, select **Use Key/Mask and/or variable cycle processing**, then click **Key/Mask** to display the History Key/Mask Information dialog box.
2. Enter the key mask in the **External History Key/Mask** field. Be sure to:
 - Enter the key mask in the correct key segment field. For example, if you have only two key segments in your reconciliation job, be sure to leave asterisks in key segment fields one, two, four, and five and add the key mask in the third key segment field.
 - Fill in any remaining space within a key segment field with spaces. For example, in the key mask example PRSNNEL, only 7 spaces of the 8-character key segment field will be filled. Therefore, you will need to add 1 space to the end of PRSNNEL to complete the key segment field.
3. Complete the **Extracted Value Number** field with the internal or extended internal item number to be retrieved.
4. Select an option in the Use Variable Cycle Processing section to indicate whether the use of this history item for reconciliation depends upon the cycle number and run time of an earlier history key. You can optionally enter a mask in the **Variable Cycle History Key/Mask** field.
5. Click **OK** to return to the History Item dialog box and complete the history item definition.

Example Combining Key Masking in Extraction with Key Masking in a History Item

The following example combines both types of key masking discussed in the previous sections "[Example of Key Masking in Extraction Definitions](#)" and "[Example of Key Masking in History Items](#)."

Problem

Suppose you have the following reconciliation jobs:

Job 1: Reconcile Personnel information.

6 ■ Setting Up Job Definitions

Using Key Masks

Personnel uses the employee ID as the reconciliation key in a job. The reconciliation job consists of validating Personnel information on a weekly basis between two processes: an edit process and an update process called Employees Terminated.

Job 2: Reconcile Payroll information.

Payroll also uses the employee ID as the reconciliation key. However, their job confirms that year-to-date accumulations are working correctly.

You want to ensure that the Payroll run does not send checks to terminated employees as defined on the Personnel system.

Currently, Personnel jobs and Payroll jobs are kept on different history databases. Job 1 (Personnel) has internal items 1 through 15 defined. Job 2 (Payroll) has internal items 1 through 10 defined. You need to combine the databases and allow the data to reside in one history database.

Resolution

To resolve this situation, you can set up a reconciliation key mask which will allow you to store history records from the Personnel and Payroll jobs separately on the same history database. Then, set up a history key mask to retrieve the Personnel information in the Payroll run.

To do this you will modify each job:

Job 1:

Modify the Basic Job Information to include a reconciliation key with the designated mask (PRSNNEL):

```
***** ***** ***** ***** PRSNNEL
```

This is the same as saying, “store the extracted keys on the history database with the literal PRSNNEL in the last key segment.” This is necessary to store Personnel histories separately from payroll histories since these will both be stored in the same database. Set up internal item I-016 with the description DATE TERMINATED so that it will be available for Payroll to use later.

Job 2:

Modify the Basic Job Information to include the following:

- A reconciliation key with the designated mask (PAYROLL):

```
***** ***** ***** ***** PAYROLL
```

- A history item that references the reconciliation key mask.

Add a history item definition E-011 using the following history key mask:

```
***** ***** ***** ***** PRSNNEL
```

using internal item I-016 from Job 1 as current.

- A reconciliation rule to test that E-01 (DATE TERMINATED from the PRSNNEL history) does not indicate that you are paying a terminated employee.

Constraints for Single-precision and Double-precision Numbers

If you use single-precision or double-precision numbers, you need to be aware of the following constraints:

- Accuracy is not guaranteed for single-precision numbers.
- When using double-precision numbers, accuracy can be guaranteed provided that the user-defined scale is less than the actual number of decimals stored in the database.

For example, the number 9987654.4321987654 could be stored as +9.98765443219876E+06. In this example, 8 decimals are stored instead of the 10 in the original number. Thus, a user-defined scale of 7 or less will be accurate.

6 ■ Setting Up Job Definitions

Constraints for Single-precision and Double-precision Numbers

Setting Up Input Sources

This chapter explains how to create file definitions for an input source. The file definitions specify the records to be scanned for extractable data, the values to be extracted, and how the values will be processed and stored. The values are ultimately stored in internal items or extended internal items to be used in the reconciliation rules for your job.

This chapter includes the following sections:

- “What Is an Input Source?” on page 115
- “What Is a Reconciliation Key?” on page 116
- “Preparing to Set Up an Input Source” on page 117
- “Procedure for Setting Up an Input Source” on page 118
- “History Database Record Layout” on page 124
- “Selection Fields and Selection Groups” on page 126
- “Selecting an Access Mode” on page 130
- “Understanding Detail Field Processing” on page 137
- “Reformat Field Processing” on page 144
- “Understanding Field Formats” on page 146
- “Setting Up an Input Source for Relational Data” on page 152
- “Date and Cycle Number Extraction and Processing” on page 158
- “Time Extraction and Processing” on page 161
- “Deleting File Definitions” on page 162
- “Extracting Directly from a Delimited Data File” on page 162
- “Matched File Processing” on page 163

What Is an Input Source?

An input source is a file (often a report from your application) or Web page from which values will be extracted to be used in a reconciliation job. Multiple input sources can be defined for a single job.

What Is a Reconciliation Key?

In ACR/Detail, the extraction and accumulation you specify in your file definitions are based on the job's reconciliation key. So it is important to understand what makes up a reconciliation key and how it is used.

A job's reconciliation key is a hierarchical structure composed of key segments (these are called the job's keys). The key segments, which are defined using key fields, set the criteria that will be used locate and extract data from the input source. Key 1, the first key segment defined, determines the highest level of the job's sort criteria. Key 2, the second key segment defined, determines the next level, and so forth.

The reconciliation key can consist of up to 5 key segments, each of which can contain up to 8 positions. This results in a maximum length for the reconciliation key of 40 bytes.

Because of its hierarchical structure, the reconciliation key provides ACR/Detail with its ability to extract, accumulate, reconcile, and report on data at multiple levels of detail.

Processing of a Reconciliation Key

When your reconciliation job runs, ACR/Detail scans the input source files for each occurrence of the reconciliation key and, based on the detail fields you define, extracts and optionally accumulates the values for that occurrence of the key. The results are ultimately stored in an internal or extended internal item to be used in the job's reconciliation rules.

Example

Suppose you want to reconcile amounts for all checks written in a certain month against the checks that cleared the bank that month.

For input to your reconciliation job, you have 1) Report A, which shows individual check numbers and amounts for each check, and 2) Report B, which shows the cleared items.

Using key fields, you could define a reconciliation key composed of 4 key segments to identify each separate check:

- Date (month)
- Routing number
- Account number
- Check number

You could define a detail field to extract the amount from all unique instances of this reconciliation key (all checks).

You could then create a reconciliation rule to compare the amount written from Report A with the amount cleared from Report B.

When you run the job, each report would be scanned to identify unique instances of the reconciliation key (individual checks) and, for each, the detail field would extract the amount. The reconciliation rule would compare the amounts from Report A with the amount from Report B for each check.

Preparing to Set Up an Input Source

Gathering Information About Your Input Sources

Before you can set up your input sources, you need to gather the basic information (the path and file name of each input source, its file organization type, etc.) described in [“Determining the Input Source Information” on page 38](#).

Preparing to Use Mark-and Capture

For non-relational database input sources, you can streamline the process of creating file definitions by using the mark-and-capture method. This is the process of selecting data with the mouse from the input file or its COBOL record layout, which is displayed in the Extract Window in Input Source View.

In order for the file or record layout to be displayed, it must be in a location that is accessible to ACR/Detail Client.

If the file is not currently accessible to ACR/Detail Client and you want to use mark-and-capture, you need to download it in text (ASCII/CR/LF) format.

For more information on mark-and-capture, see [step 10](#) in the [“Procedure for Setting Up an Input Source” on page 118](#).

Preparing to Use the Manual Method

If you cannot use the mark-and-capture method, review each input file and note the exact starting position, length, and format of each field that you wish to select from the input source.

7 ■ Setting Up Input Sources

Procedure for Setting Up an Input Source

Modifying Your Environment File

For Oracle input sources or to use ODBC to access a relational database table, make sure you have followed the instructions in “[UNIX or Linux Only: Modifications for RDBMS Access](#)” on page 36.

Procedure for Setting Up an Input Source

Following is the procedure for setting up an input source. The dialog box help provides field definitions and other details.

Note: If your input source is a relational database table (File Organization Type DB2, Oracle, or ODBC), see [Setting Up an Input Source for Relational Data](#) on page 152.

1. Complete the instructions in the previous section, “[Preparing to Set Up an Input Source](#)” on page 117. In particular, we recommend that you download input files as needed to use mark-and-capture (described in [step 10](#) below) to select data from them.
2. From Control Task (Job) View, select **Input Sources > New** to open the Input Source View. You will see the toolbar icons that display in Control Task (Job) View, plus an additional set of icons pertaining only to input sources. Scroll your mouse over the icons to see the labels that identify them.
3. In the Basic Information section of the view, complete the Name dialog box using the instructions in the dialog box help. Make special note of the following:

Complete the **File ID** field by providing a name for this input source. (The **Qualifier** is used only if the file is to be processed more than once during the job.) Complete a, b, or c below depending on the type of input source file you are using. Then go to [step 5](#).

- a. Complete one of the following fields:
 - Complete the **File Name** field if this input source is an ASCII text file accessible to ACR/Detail Client.
 - Complete the **Record Layout** field if the COBOL record layout of this input source file is accessible to ACR/Detail Client. and you want to use this instead of the file itself to create file definitions.

If the file is in ASCII format and is accessible to the ACR/Detail Client, it will display in the Extract Window.

- b. If your input source is a Web page, and the machine where ACR/Summary and ACR/Detail Server resides has access to the Internet, do the following:
 - Access the Internet and find the Web page you want to use.
 - In the Name dialog box, select **Specify web page information**.
 - In the **URL Address** field, enter or copy the complete URL address of the web page.
 - In the **Local Copy** field, enter path and filename for a file ACR/Detail will create to store a copy of the page.
 - Click **Retrieve** to retrieve the local copy of the page that will display in the Extract Window when you are setting up your file definitions. Unique, sequential numeric identifiers will appear in front of each text line and table cell. Optionally, you can use these identifiers to select the data you want to reconcile.

You can optionally reuse file definitions that are in the definition database for an existing File ID by entering that existing File ID in the **File ID** field and selecting the **Use existing input source definitions** check box. If you do this, you will not be able to create any additional file (extraction) definitions specific to this job.

4. Optionally complete the File Description dialog box to help you when reviewing your definitions. The description will not appear on reports.
5. Complete the File Organization dialog box, making special note of the following:
 - The fields will change based on your selection in the **File Organization/Type** field. After you have completed the field, the dialog box help will provide field documentation. The following types are available:

Physical sequential—A physical sequential file.

Note: Direct extraction of delimited data is supported for physical sequential input sources. See [Extracting Directly from a Delimited Data File](#) on page 162.

7 ■ Setting Up Input Sources

Procedure for Setting Up an Input Source

Detail history database—Specifies that you want to use all or selected records from the existing history database as input.

Note: When using the history database as an input source, the **Extract All** field on the dialog box determines if all history records should be extracted.

If you do not use this option, you will have to create the selection fields, key fields, and detail fields to extract any history records during reconciliation. For the record layout of the history database, see [History Database Record Layout on page 124](#)

VSAM key sequenced—This is an ISAM file in Windows, UNIX, or Linux.

Direct spool dataset—This file type is applicable only if you are running jobs on z/OS.

User Program Accessed File—Call an external COBOL program to access data that could not be accessed otherwise. For information on the user programs provided by Infogix, see the *ACR/Summary and ACR/Detail User Program Guide*.

Web page—Access data from a page on the Internet.

- Specify the access mode (6 or 7) to control how the file definitions will be processed. Access mode 6 is the typical mode, but in some cases access mode 7 is preferable. For more information, see “[Selection Fields and Selection Groups](#)” on page 126 and “[Selecting an Access Mode](#)” on page 130
- 6. Complete the Key Matching dialog box only if you want the keys extracted from this input file to be used in determining whether keys extracted from each of the job’s other input files will be used in reconciliation.
- 7. Complete the Set Return Code dialog box if you want to specify a return code when no records are selected for extraction from this file. The return code is a 3- or 4-digit code that appears on the status bar when a job is run and in the Control Report. (Use only 3 digits if you plan to use the \$? environment variable on UNIX or Linux, because this variable is limited to 3 digits.)
- 8. Complete the File Comments dialog box if you want to paste (**CTRL + V**) or type in up to 70,992 characters of your own comments regarding the input source file. All of the comments will print when you generate a List Definitions Report that includes detailed file information

9. Use the Global Reformat folder if you want to define one or more global reformat fields that will be applied to every record in the input source. Reformat fields define criteria to change the positions and/or formats of data from 1) the input file, 2) a literal, or 3) an extraction variable.

A global reformat field will reformat all matching records in your file before any other definitions are processed. Consequently, all later reference to positions must refer to the new reformatted positions.

Note: You can also create reformat fields within a selection group, but they are not global. See [step 11 on page 122](#).

10. Before you begin creating extraction definitions, familiarize yourself with the following procedures:

Mark-and-Capture Method

You can use mark-and-capture if the input source or its record layout appears in the Extract Window at the bottom of the Input Source View. The procedure is as follows:

- a. Using your mouse, select the data you want to extract from the file displayed in the Extract Window. This activates the toolbar icons for the various extraction definition types you can create.

NOTE: If the input source is lengthy, you may want to use the **Find** and/or **Repeat Find** icons to find the data you want. These icons are available when the cursor is in the Extract Window.

- b. After selecting the data, click the icon for the extraction definition type you want to create.
- c. When the corresponding dialog box displays, the **Key Value**, **Position**, and **Length** fields (if they exist on the dialog box) will be populated for you, but you can change them.

Note: For more information about how dates are extracted, processed, and stored, see [Date and Cycle Number Extraction and Processing on page 158](#).

- d. Repeat the previous 3 steps to create any additional required extraction definitions for this input source.

7 ■ Setting Up Input Sources

Procedure for Setting Up an Input Source

Manual Procedure

If the input source does not appear in the Extract Window, you need to use the manual method. The procedure is as follows:

- a. Make sure you know the exact starting position, length, and format of each field that you want to select from the input source.
- b. Click the icon for the extraction definition type you want to create.
- c. When the corresponding dialog box displays, use the information from [step a](#) above to complete the information.
- d. Repeat the previous 3 steps to manually create any additional required extraction definitions for this input source.

11. Complete the Extraction Definitions section of the view.

Extraction definitions, also called file definitions, are composed of one or more selection groups. The way selection groups work depends on whether you are in access mode 6 or 7.

Create the field types required for each selection group. Following is a typical procedure:

- a. Create one or more selection fields. Consecutive selection fields are part of the same selection group. In access mode 6, consecutive selection fields can be joined by AND or OR. For more information, see [“Selection Fields and Selection Groups”](#) on page 126.

To create a selection field definition, select the value in the Extract Window (if you are using mark-and-capture), then click the **Selection** icon and complete the dialog box.

- If you are selecting from the input area and are using mark-and-capture, the position and length will be pre-filled.
- If you are selecting from an extraction variable, you will need to select or create the extraction variable.

If the records containing the value you want to extract contain no unique value, see [step b](#).

- b. Create any needed relative records. These are used if you are selecting from the input area, and the current record does not contain a unique value. A relative record moves the selection forward a specified number of records from the last record selected, that is, the record selected in the selection field that precedes it.

To create a relative record definition, click the **Relative Record** icon and complete the dialog box.

- c. Create any needed reformat fields. Reformat fields defined within a selection group and/or between selection fields are not the same as reformat fields defined in the Global Reformat folder mentioned earlier in this procedure. These are applied only to records that satisfy the selection group's criteria.

To create a reformat field, click the **Reformat** icon and complete the dialog box.

- d. Create any key fields needed to specify a unique identifier for one part (segment) of the reconciliation key. The key identifiers for all key fields together make up the reconciliation key for the job.

To create a key field, select the field with your mouse (if you are using mark-and-capture), click the **Key** icon, and complete the dialog box.

Select **Use Translation** and select **Translate** if you want to pass values from a key field through one of the following:

- An internal or external translation table for data conversion.
- A dynamic translation table to translate the key.
- A hash translation for the key.

For more information see [“Using Translation Tables” on page 193](#).

- e. One or more detail fields are required. A detail field extracts data from the selected record, from a literal, or from an extraction variable (or counts the keys found), processes extracted values according to the **Field Type** and **Accumulate Option** specified. The result can be stored in an internal item or extraction variable, either of which can be regular or extended. Literals are still a maximum of 8 characters.

To create a detail field definition, select the field with your mouse (if you are using mark-and-capture), click the **Detail** icon and complete the dialog box.

Select **Use Translation Table** and select **Translate** if you want to pass values from a detail field through an internal or external table. For more information, see [“Understanding Detail Field Processing” on page 137](#).

12. Click the **Save** icon to save the control entity, including the job and file definitions you have created for this job.
13. Repeat [step 1](#) through [step 12](#) to create any additional input sources needed for the job.

7 ■ Setting Up Input Sources

History Database Record Layout

You can optionally generate a Trace Report when you run reconciliation. For more information, see [“Trace Report” on page 280](#).

When you have finished your job and file definitions, the values stored in internal items or extended internal items will be used along with any calculated items and history items in your reconciliation rules.

History Database Record Layout

When using the history database as an input source, the Extract All field on the dialog box determines if all history records should be extracted.

If you do not use this option, you will have to create the selection fields, key fields, and detail fields to extract any history records during reconciliation as you would for any other input source.

Following is the record layout of the history database.

Field#	Description	Position	Length	Format	Value
1	Reconciliation key 1	9-16	8	X(8)	
2	Reconciliation key 2	17-24	8	X(8)	
3	Reconciliation key 3	25-32	8	X(8)	
4	Reconciliation key 4	33-40	8	X(8)	
5	Reconciliation key 5	41-48	8	X(8)	
6	Reserved	49-104	56	X(56)	
7	XI Run Value Count	105-106	2	S(9) Comp-3	
8	Reserved	107-135	29	X(29)	
9	Cycle ID	136-146	11	9(11)	
10	Reserved	147-181	35	X(35)	
11	Note Data	182-261	80	X(80)	
12 ¹	Run Value Count	262-263	2	S(9) Comp-3	
13 ²	Detail Value Occurs 0-999 times	264-271	8	S(15) Comp-3	
14 ³	Detail XI Value - Occurs 0-100 times	8256- 8336	80	X(80)	

1. Run Value Count: Number of detail items for this key.

2. Detail Value: Format is X(8) for text detail values. If there are no XI (extended internal) items, field 13 is variable length, depending on the number of internal items. If there are extended internal items, field 13 is 999 detail values long, no matter how many field 13s there are.

3. Detail XI Value: Number of XI detail value items.

Selection Fields and Selection Groups

In ACR/Detail, for all input source types except relational database tables, file definitions are made up of selection groups, each of which includes one or more selection fields. The following sections describe selection fields and selection groups.

Two access modes, numbered 6 and 7 to correspond with similar access modes in ACR/Summary, are available in ACR/Detail. After you read this information, you will be need to select an access mode, as described in [“Selecting an Access Mode” on page 130](#).

Selection Fields

The following information applies to both access modes.

A selection field is a sophisticated feature for determining whether the current record will be selected for further processing or bypassed. Its features include:

- The evaluation can be performed using data from one of the following input types: 1) data from an input area (the current record) or 2) data from a saved extraction variable.
- You can choose from multiple selection types. For example, the evaluation can test the format of the value (numeric or alphabetic), or can test the value using comparisons such as equal to, less than or equal to, or within range. In addition, you can restrict processing to a specific portion of the file.
- Depending on your input type and selection type, you can evaluate against 1) one constant or one range, 2) multiple constants or multiple ranges), or 3) the value of an extraction variable.

Selection Groups

This information applies to both access mode 6 and access mode 7 except as noted.

One or multiple selection fields can be combined with other file definition types that are valid for access modes 6 and 7 to make up a selection group. A selection group consists of the following components:

- [“Group-Level Selection Group” on page 127](#).
- [“Subgroup-Level Selection Groups” on page 127](#).
- [“Relative Records, Reformat Records, Key Fields, and Detail Fields” on page 128](#).

You will learn about “[Working with Selection Groups](#)” on page 127.

Group-Level Selection Group

A selection group must include one selection field or a set of consecutive selection fields at the **group level**.

Group-level selection fields are always performed and are not indented in the display. Consecutive selection fields are joined by AND or OR as explained below.

Note: Joining of selection fields using the logical operator OR is supported only in access mode 6.

You can define one or multiple selection groups at the group level.

Subgroup-Level Selection Groups

Optionally, a selection group can contain one selection field or a set of consecutive selection fields at the **subgroup level (nested)**.

Note: Selection fields at the subgroup level (nested) are supported only in access mode 6.

Subgroup-level selection fields are indented in the display. A subgroup will be processed only if the record has passed the group-level evaluation directly above it. Only one level of nesting is supported.

You can define one or multiple selection groups at the subgroup level within a selection group.

Working with Selection Groups

Demoting or Promoting Between Group Level and Subgroup Level

To toggle a selection field between the selection group level and the selection subgroup level, click on the field that you want to demote or promote and then click the **Increase Indent** or **Decrease Indent** icon on the toolbar, respectively.

Use of AND/OR Logical Operators

Consecutive selection fields at the same level (group or subgroup) are joined for evaluation using the logical operator AND (the default) or OR. The logical operator that is currently being used to evaluate consecutive selection fields is displayed to the right of the first selection field in the group or subgroup. Use a sub-group if you need to use both AND and OR in your selection criteria.

7 ■ Setting Up Input Sources

Selection Fields and Selection Groups

Toggling Between AND and OR

To toggle between AND and OR, select the first selection field in the group or subgroup and click the **Toggle AND/OR** icon on the toolbar.

AND and OR work as you would expect:

- When AND is used, the criteria of all consecutive selection fields at the same level must be satisfied before the record will be selected for processing against the remaining file definitions in the same selection group or subgroup.
- If OR is used and the criteria of any one of the consecutive selection fields at the same level are satisfied, the record will be selected for processing against any remaining file definitions in the selection group or subgroup.

Relative Records, Reformat Records, Key Fields, and Detail Fields

In addition to selection fields, a selection group consists of the following:

- One or more detail fields.
- Optionally, one or more reformat fields, key fields, and relative record fields.

These file definition types can be defined at the group level, the subgroup level, or both. They will be processed if the record passes the preceding group or subgroup evaluation. Following are descriptions of these file definition types:

Relative Records

A relative record (also called advance record) can be used to locate a record in relation to the last record selected. You specify the number of records to move forward (advance) relative to the last record selected. When the system encounters a relative record, no other file definitions are processed until the new record is found. Then the following file definitions will be processed against this record.

Note: A relative record does not have to be part of a selection group. It can be defined before the first selection group. In this case, the system advances the specified number of records from the first record in the file.

If the relative record request follows a selection field with comparison type **Begin - Equal To**, the system processes the relative record only one time, when the **Begin** reference is found. If the relative record request follows other extraction field types, the relative record is processed whenever the selection criteria is met.

Reformat Records

A reformat record can be used to change the positions and/or format of data from 1) the input file, 2) a literal, or 3) an extraction variable. Regardless of where the data to be formatted comes from, the reformatting/moving is done in memory only and does not change the input source, extraction variable, or literal)

For more information, see [“Reformat Field Processing” on page 144](#).

Key Fields

A key field specifies a unique identifier for one part (segment) of the reconciliation key. The key identifiers for all key fields together make up the reconciliation key for the job.

For example, if you want to extract and reconcile data for each unique combination of area code and telephone number, you would define two key fields, one to identify each area code and the other to identify each telephone phone number for each area code.

If you need only one key segment to identify the records you want to scan in order to extract data for the job, one key field can define the entire reconciliation key.

A key field can be populated with data extracted from the input source, with text supplied in a literal, with data from a reformatted combination of the two, with a value from an extraction variable, or with a value from a function.

Detail Fields

A **detail field** extracts data from the selected record, from a literal, or from an extraction variable (or counts the number of records selected), processes extracted values according to the **Field Type** and **Accumulate Option** specified, and stores the result in an internal or extended internal item or in an extraction variable. For more information, see [“Understanding Detail Field Processing” on page 137](#).

Selecting an Access Mode

For all input source types except relational database tables, you complete the **Access Mode** field on the File Organization dialog box to specify how the file definitions will be processed.

Two access modes, numbered 6 and 7 to correspond with similar access modes in ACR/Summary, are available in ACR/Detail. Both access modes use selection groups, described in the preceding sections, to select records that will be scanned for extraction.

Choose Access Mode 8 (Use matched file processing), to read and compare the keys and/or the items from two input sources concurrently for improved performance. For more information, see “[Access Mode 8 \(Use Matched File Processing\)](#)” on page 170

The following sections provide the information you will need to select the appropriate mode for your input data and reconciliation needs:

- “[Similarities and Differences Between Access Modes 6 and 7](#)” on page 130. This section is a summary. The sections on each access mode will provide more information.
- “[Access Mode 6—Keys Precede Detail Values](#)” on page 131
- “[Access Mode 7—Keys Follow Detail Values](#)” on page 133

Similarities and Differences Between Access Modes 6 and 7

The similarities are as follows:

- Both access modes select or bypass records using selection fields.
- Both access modes use the same file/record types:
 - Selection fields. A selection field defines criteria used to evaluate a record for selection and further processing. For more information, see “[Selection Fields and Selection Groups](#)” on page 126.
 - For descriptions of the other valid file/record types, see “[Relative Records, Reformat Records, Key Fields, and Detail Fields](#)” on page 128.

The following table shows the differences between the two access modes:

Factor	Access Mode 6—Keys Precede Detail Values	Access Mode 7—Keys Follow Detail Values
Suitable input source structure	Keys precede detail values.	Keys follow detail values.
Key processing¹	<ol style="list-style-type: none"> 1. When the second reconciliation key is read, the first key is written out along with any detail values extracted since the top of the file. 2. When the next key is read, the previous key is written out along with any detail values extracted since the previous key. 3. Remaining keys are processed as in step 2. 	<ol style="list-style-type: none"> 1. When the first reconciliation key is read, it is written out along with any detail values extracted since the top of the file. 2. When the next key is read, it is written out along with any detail values extracted since the previous key. 3. Remaining keys are processed as in step 2.
Support for OR and subgroups	Yes.	No.
Selection group processing	Every record in the input source will typically be evaluated by each selection group in turn, always starting with the first group.	This is not always the case. See Access Mode 7 Selection Group Processing on page 133.

1 - If keys and detail fields reside on the same records, use access mode 6.

Access Mode 6—Keys Precede Detail Values

When to Use Access Mode 6

Access mode 6 is appropriate for the way the data is organized in most input sources, and it offers enhanced features to make it easy to set up your file definitions. Access mode 6 can be used when either of the following is true:

- The values to be defined as keys and the values to be extracted as detail fields reside on the same record.
- The records containing keys precede the records containing the values to be extracted as detail fields.

Otherwise, access mode 7 is preferable. For more information, see [“Access Mode 7—Keys Follow Detail Values”](#) on page 133.

7 ■ Setting Up Input Sources

Selecting an Access Mode

In access mode 6, every record in the input source will be evaluated by each selection group in turn, always starting with the first group.

Note: In both access mode 6 and access mode 7, you can use the following features to bypass records:

- Relative (advance) records.
- The Begin equal to and/or End equal to selection types (available in the Selection Field dialog box).

Records that pass the selection group evaluation can be processed by the other file definition types available for access modes 6 and 7 (see “[Relative Records, Reformat Records, Key Fields, and Detail Fields](#)” on page 128) to obtain data for use in reconciliation rules.

Access Mode 6 File Definitions Example

DEPT	CHICAGO	00001
AMT	125.00	
DATE	05/02/2007	
DEPT	NEW YORK	00002
AMT	155.00	
DATE	05/03/2007	

Suppose your input source is the one shown above, and you want to extract the city for each key. The key will be the department code (00001 and 00002). The key resides on the same record as the detail field to be extracted, making this data appropriate for access mode 6.

You could set up your definitions like this:

Definitions

Extraction Definitions
Select if position 1 for a length of 4 is equal to DEPT (AND)
Retrieve key 1 (DEPTCODE) in position 22 for a length of 5
Get value from position 11 for a length of 8, store in internal item CITY

Input source:

DEPT	CHICAGO	00001
AMT	125.00	
DATE	05/02/2011	
DEPT	NEW YORK	00002
AMT	155.00	
DATE	05/03/2011	

This example shows a simple selection group with one selection field (Select if position 1 etc.). Using the **is equal to** selection type, the specified positions of the input area are evaluated against one constant (DEPT). Only records that meet this criterion will be scanned for extraction. Data will be picked up based on the key DEPTCODE occupying the 5 positions beginning at position 22. The data to be extracted for each key is defined using the detail field (the Get value line in your definitions) and assigned to the internal item CITY.

When all file definitions have been processed, the next record will be read and evaluated against all of the same file definitions in sequence, starting at the top.

Note: Notice the AND in parentheses next to the selection statement. This is the default logical operator. It will not be used because there are no consecutive selection statements at the same level.

Access Mode 7—Keys Follow Detail Values

When to Use Access Mode 7

Access mode 7 is preferable when, in your input source file, both of the following are true:

- Values to be defined as keys and values to be extracted as detail fields do not reside on the same records.
- The records containing keys follow the records containing values to be extracted as detail fields.

Otherwise, access mode 6 is preferable.

Access Mode 7 Selection Group Processing

In access mode 7, processing of selection groups is as follows:

Note: During the processing described below, keys and extracted values will be written out as described in [Similarities and Differences Between Access Modes 6 and 7](#) on page 130.

1. The first record in the file (we will call it Record 1) will be evaluated against all selection groups. The last selection group that passes will be noted.

7 ■ Setting Up Input Sources

Selecting an Access Mode

2. The system then reads the next record (Record 2). This record will be evaluated starting with the next selection group after the one noted when the previous record was read. Evaluation of Record 2 will be as follows:
 - a. If the record passes all remaining groups through the last group defined, evaluation of the record will stop. The system will read the next record and evaluate it just as Record 1 was evaluated.
 - b. If the record fails any of the remaining selection groups through the last group defined, it will be evaluated by all selection groups.
 1. If the record passes at least one group, the last group that passes will be noted.
 2. If the record fails all groups, the last record that fails will be noted.

The system will then move to the next record and evaluate it just as Record 2 was evaluated.

Access Mode 7 Selection Group Processing Example

This example shows how selection groups would be processed in access mode 7 with various pass/fail scenarios.

Assume you have an input source consisting of 5 records. You have defined 4 selection groups. For each record, the following table shows which selection group will be processed first, and when the next record will be read.

Record	Selection Group Evaluated	Result	Selection Group Noted	Next Record is Read	Notes (Refer to the processing steps described in the preceding section.)
1	1	Pass			
	2	Pass			
	3	Fail			
	4	Fail	2	Y	See step 1.
2	3	Fail			
	4	Fail			
	1	Fail			
	2	Fail	2	Y	See step 2b2.
3	3	Pass			
	4	Pass		Y	See step 2a.
4	1	Pass			
	2	Fail			
	3	Pass			
	4	Fail	3	Y	See step 1.
5	4	Pass			See step 2a. Evaluation of all records in the input source is now complete.

7 ■ Setting Up Input Sources

Selecting an Access Mode

Access Mode 7 File Definitions Example

DATE	05/02/2007	
AMT	125.00	
DEPT	CHICAGO	00001
DATE	05/03/2007	
AMT	155.00	
DEPT	NEW YORK	00002

Suppose your input source is the one shown above, and you want to extract the city, date, and amount for each city.

The key resides on a separate record from the records where two of the detail fields (date and amount) reside. In addition, the records containing the key **follow** the records containing the date and amount, which will be extracted as detail fields. These factors make this data appropriate for access mode 7.

You could set up your input source definitions like this:

Definitions:

Select if position 1 for a length of 4 is equal to DATE (AND Get value from position 11 for a length of 10, store in internal item DATE Select the record that occurs 1 records from the current record (AND) Get value from position 11 for a length of 6, store in internal item AMOUNT Select the record that occurs 1 records from the current record (AND) Get value from position 11 for length of 7, store in internal item CITY Retrieve key 1 (CITY) in position 11 for a length of 7
--

Input source:

DATE	05/02/2011	
AMT	125.00	
DEPT	CHICAGO	00001
DATE	05/03/2011	
AMT	155.00	
DEPT	NEW YORK	00002

The first selection statement (Select if position 1 etc.) uses the selection type **is equal to**. The specified positions of the input area will be evaluated against the constant DATE. The first record will pass the evaluation, and the following detail field (Get value etc.) will extract the date.

The next definition (Select the record that occurs etc.) is a relative record that causes the system to move down one record. The following detail field (Get value etc.) will extract the amount.

The next definition (Select the record that occurs etc.) is also a relative record that causes the system to move down one record. The following detail field (Get value etc.) will extract the city.

When the system reads the key field (Retrieve key etc.), the key (CHICAGO) will be retrieved and written out along with the detail fields extracted since the top of the file: the date 05/02/2007, the amount \$125.00, and city name CHICAGO.

The last three records will be processed in the same way. The key (NEW YORK) will be retrieved and written out along with the detail fields extracted since the top of the file: the date 05/03/2007, the amount \$155.00, and city name NEW YORK).

Understanding Detail Field Processing

This section consists of the following subsections:

- "Stage 1 Processing" below.
- "Stage 2 and Stage 3 Processing" on page 140.

Sample Input Source for Detail Field Processing

The following sections use an example to illustrate the three processing stages for a detailed field. Following is the report that will be used as the input source for this example. The CC numbers (CC00001 and CC00002) are the reconciliation keys.

Credit Card Charges					
Report Created September 1, 1995					
CC00001	HARDWARE STORE	0100.00	08/03/06		
CC00001	GROCERY STORE	0110.00	08/05/06	_ _ _ _	Key Break 1
CC00002	DEPARTMENT STORE	0200.00	08/05/06	_ _ _ _	Key Break 2
CC00001	RESTAURANT	0080.00	08/06/06	_ _ _ _	Key Break 3
CC00002	RESTAURANT	0080.00	08/06/06		
CC00002	THEATER	0070.00	08/09/06	_ _ _ _	Key Break 4

Stage 1 Processing

All detail fields undergo stage 1 processing. Stage 1 processes values by key break based on the **Field Type** option you specify in the Detail Field dialog box.

7 ■ Setting Up Input Sources

Understanding Detail Field Processing

A key break occurs when the value of the data being used as the key changes. One record for each key break will be stored in a temporary file containing key and internal or extended internal items. The sample input source above shows four key breaks, which means that there will be four records in the temporary file.

For detail fields stored in extraction variables, stage 1 is the only processing stage: the results of stage 1 are stored in the extraction variable. For detail fields stored in internal or extended internal items, the values from stage 1 will be sorted by key and fed into stage 2.

Stage 1 Processing Options

The **Field Type** options are: 1) to save the first value, 2) to save the last value, or 3) to accumulate the values. Here are the options:

The first four options are to save the last value at the time of the key break. Each of these options corresponds to a different item type: last count, last amount, last date, last text:

- **Last Count.** The last count value extracted from the last detail record selected from the input file with the current key. This is the value associated with the last occurrence of the key within the consecutive list of keys, that is, the last value from stage 1 processing before the key break.
- **Last Amount.** The last amount value extracted from the last detail record selected from the input file with the current key.
- **Last Date.** The last date-count value for the date extracted from the last detail record selected from the input file with the current key. The Date field type is not valid if you chose Literal from the extract from field.
- **Last value.** The last text value extracted from the last detail record selected from the input file with the current key.

The middle options is Sum. This is the sum of all count or amount values for each key break.

The last four options are to save the first value found after a key break. Each of these options corresponds to a different item type: first count, first amount, first date, first text:

- **First Count.** The first count value extracted from the first detail record selected from the input file with the current key.
- **First Amount.** The first amount value extracted from the first detail record selected from the input file with the current key.

- **First Date.** The first date-count value for the date extracted from the first detail record selected from the input file with the current key. (The Date field type is not valid if you chose Literal from the extract from field.)
- **First Value.** The first text value extracted from the first detail record selected from the input file with the current key.

Note: Remember that dates are converted to date-counts.

If you specify First Value, then the value associated with the first occurrence of the key within the consecutive list of keys is used.

Temporary Records	Last Count/ Amt Values	First Count/Amt Values	Last Date Value*	First Date- count Value	Last Text Value	First Text Value
Key Break 1 CC00001	110.00	100.00	148140	148138	GROCERY STORE	HARDWARE STORE
Key Break 2 CC00002	200.00	200.00	148140	148140	DEPARTMENT STORE	DEPARTMENT STORE
Key Break 3 CC00001	80.00	80.00	148141	148141	RESTAURANT	RESTAURANT
Key Break 4 CC00002	70.00	80.00	148144	148144	THEATER	RESTAURANT

For detail fields being stored as extraction variables: Unlike values of internal or extended internal items, extraction variable values do not reset with key breaks. So if the detail field will be stored in an extraction variable and you want to sum all count or amount values extracted, you need to initialize the value of the extraction variable by assigning a literal value of zeroes (00000000) to it before loading a new value into it.

Based on our example, the results for this option would be as follows:

Temp Location	Count/Amount Value
Key Break 1—CC00001	0210.00
Key Break 2—CC00002	0200.00
Key Break 3—CC00001	0080.00
Key Break 4—CC00002	0150.00

Stage 2 and Stage 3 Processing

Stage 2 and stage 3 processing are determined by the option selected for the **Accumulate Option** field in the Detail Field dialog box, as follows:

- Stage 2 identifies like keys from stage 1 and accumulates them according to the option chosen for this field.
- Stage 3 stores history based on the option chosen for this field (if the job stores history).

Stage 2 and Stage 3 Processing Options

The options for this field are explained in the following sections. These sections include some examples of the results you would receive using each option based on the sample input data provided in “[Sample Input Source for Detail Field Processing](#)” on page 137.

- “Sum values with the same key” on page 140
- “Retain the first non-zero value for each key” on page 141
- “Retain the last value for each key” on page 142
- “Sum values with the same key; replace history” on page 142
- “Detail value must not equal history value” on page 142
- “Detail value must be greater than history value” on page 142
- “Detail value must be less than history value” on page 143
- “Retain minimum value; replace history” on page 143
- “Retain maximum value; replace history” on page 144
- “Retain the first value for each key (including zero values)” on page 144

Sum values with the same key

Add together any count or amount values from stage 1 processing that have the same key and add the resulting value to the value stored in history.

Note: In our example from stage 1, key break 1 stored values for the key CC00001, key break 2 stored values for key CC00002, key break 3 stored values for CC00001, and key break 4 stored values for CC00002. Even though there were four key breaks, there were only two keys represented.

The results based on our example will be as follows:

Stage 1 Processing Option	CC00001	CC00002
Last Count/Amount Value	190.00	270.00
Last Date	Not applicable	Not applicable
Last Value	Not applicable	Not applicable
Sum	290.00	350.00
First Count/Amount Value	180.00	280.00
First Date	Not applicable	Not applicable
First Value	Not applicable	Not applicable

Retain the first non-zero value for each key

Retain the first non-zero value for each key and store it in history. The results based on our example will be as follows:

Stage 1 Processing Option	CC00001	CC00002
Last Count/Amount Value	110.00	200.00
Last Date	148140	148140
Last Value	GROCERY STORE	DEPARTMENT STORE
Sum	210.00	200.00
First Count/Amount Value	100.00	200.00
First Date	148138	148140
First Value	HARDWARE STORE	DEPARTMENT STORE

7 ■ Setting Up Input Sources

Understanding Detail Field Processing

Retain the last value for each key

Retain the last value for each key and store it in history. The results based on our example will be as follows:.

Stage 1 Processing Option	CC00001	CC00002
Last Count/Amount Value	80.00	70.00
Last Date	148141	148144
Last Value	RESTAURANT	THEATER
Sum	80.00	150.00
First Count/Amount Value	80.00	80.00
First Date	148141	148141
First Value	RESTAURANT	RESTAURANT

Sum values with the same key; replace history

Add together count or amount values with the same key; replace history. This is similar to the first stage 2 option, Sum values with the same key, except that with this option history is replaced instead of accumulated with the value.

Detail value must not equal history value

If the last count, last amount, last value, or last date option is specified in stage 1, retain the last value for each key.

If the first count, first amount, first value, or first date option is specified in stage 1, retain the first value for each key.

If the Sum option is specified in Stage 1 processing, accumulate the values for each key.

Store the retained value in history as follows: If this value is not equal to the value currently stored in history, replace the value currently stored in history. If the values are equal, halt processing.

Detail value must be greater than history value

Retain the value from Stage 1 as follows:

If the last count, last amount, last value, or last date option is specified in stage 1, retain the last value for each key.

If the first count, first amount, first value, or first date option is specified in stage 1, retain the first value for each key.

If the Sum option is specified in Stage 1 processing, accumulate the values for each key.

Store the retained value in history as follows: If this value is greater than the value currently stored in history, replace the value currently stored in history. If the values are equal or if the history value is greater, halt processing.

Detail value must be less than history value

Retain the value from Stage 1 as follows:

If the last count, last amount, last value, or last date option is specified in Stage 1 (Field type field), retain the last value for each key.

If the first count, first amount, first value, or first date option is specified in Stage 1, retain the first value for each key.

If the Sum option is specified in Stage 1 processing, accumulate the values for each key.

Store the retained value in history as follows: If this value is less than the value currently stored in history, replace the value currently stored in history. If the values are equal or if the history value is less than the detail value, halt processing.

Retain minimum value; replace history

Retains the minimum value for each key from Stage 1 processing and replaces the value currently stored in history. The results based on our example will be as follows:.

Stage 1 Processing Option	CC00001	CC00002
Last Count/Amount	80.00	70.00
Last Date	148138	148140
First Count/Amount	80.00	80.00
First Date	148138	148140

7 ■ Setting Up Input Sources

Reformat Field Processing

Retain maximum value; replace history

Retains the maximum value for each key from Stage 1 processing and replaces history. The results based on our example will be as follows:

Stage 1 Processing Option	CC00001	CC00002
Last Count/Amount	110.00	200.00
Last Date	148141	148144
First Count/Amount	100.00	200.00
First Date	148141	148141

Retain the first value for each key (including zero values)

Retains the first value for each key (including zero values) from Stage 1 processing and replaces history. The results based on our example will be as follows:

Stage 1 Processing Option	CC00001	CC00002
Last Count/Amount Value	110.00	200.00
Last Date	148140	148140
Last Value	GROCERY STORE	DEPARTMENT STORE
Sum	210.00	200.00
First Count/Amount Value	100.00	200.00
First Date	148138	148140
First Value	HARDWARE STORE	DEPARTMENT STORE

Reformat Field Processing

Reformat fields are used to change the positions and/or formats of data from the input source, a literal, or an extraction variable for reconciliation purposes.

For example, there may be characters in your input file that you want to use as a key, but the characters do not appear together on the file. You can use reformat fields to specify the location from which the characters should be extracted and to output the characters so that they are next to each other to be extracted as the key.

When reformat field definitions are used, a copy of the data to be reformatted is made in memory and the reformatting is done to this copy. The data in the input source, literal, or extraction variable is not changed.

You can then overlay the copied data with data from an extraction definition, a literal, or an extraction variable. Any combination of data, literals, and extraction definitions can be used.

The reformatted data can then be placed in one of the following:

- An output area
- The note area of the output record. This is an 80-character line printed below each key on the Control Report and saved on the history database if the job stores history.

After reformatting, field positions specified in subsequent selection fields, key fields, or detail fields will refer to the output positions.

Example

You want to extract a social security number from your input source. In the input source, the number is formatted with hyphens as follows:

111-22-3333

You want to extract it without the hyphens. This would require 2 reformat records:

- One record to move the 22 to the left one position.
- One record to move the 3333 to the left two positions.

If you do this, the result will be 11122333333 (in memory) because the last two 3's from the original field were not overlaid. You could then use the first 9 positions (the reformatted social security number) in subsequent selection field or detail field definitions.

Global Reformat Fields

Global reformat fields are defined in the Global Reformat Folder in Input Source View. A global reformat is a reformat that is performed before any selection criteria are applied. It is identified by a selection group number of zero.

7 ■ Setting Up Input Sources

Understanding Field Formats

Reformat Fields in Extraction Definitions

Reformat fields defined within and/or between the selection groups in the Extraction Definitions folder are processed only against input records that satisfy the selection group's criteria. This applies even though reformats between selection groups are given their own distinct selection group number.

Reformat Data Types

Generally, you reformat data to the same type. For example, text to text or numeric to numeric. If you reformat text to numeric, be aware that after the data is copied to the new location, each position of the output area is checked, and non-numeric characters are replaced with a zero. For example, if you reformat 07/01/29 from text to numeric, you will get 07001029.

You may need to reformat a packed field to numeric because you cannot pick up part of a packed field. In this case you would use reformatting to unpack the data. After the data has been reformatted from packed to numeric, you can use a detail field to extract just the portion you need. Keep in mind that the sign of the packed field (positive or negative) will be removed after the reformat. If you want to keep the sign, use zoned decimal (overpunch) format.

Understanding Field Formats

Descriptions of Field Formats

When creating record/field definitions, you will be required to specify the format of the data. The options are as follows:

-
- Note:**
- The only valid formats for extraction variables are alphanumeric and packed.
 - For information on maximum lengths when extracting data in each format, see [Maximum Lengths for Regular and Extended Data](#) on page 89.
-

Numeric. All of the characters are numeric. Blanks will be ignored, alphanumeric characters will be stripped out, and the field will be padded with preceding zeros. For example, suppose the value has a

length of eight and you specify the format as numeric. The bytes are stored as follows:

Value	This will be interpreted as:
1234 (4 blanks after value)	000000000001234
1A2B3C4D	000000000001234
1234ABCD	000000000001234
1 2 3 4 (spaces between numbers)	000000000001234
12341111	00000012341111
(all blanks)	000000000000000

Character (Alphanumeric). Some or all of the characters are alphabetic. This format allows punctuation and numbers as well as alphabetic characters. The string below could be formatted as alphabetic.

JOHN DOE'S PHONE# IS (630)555-1212

7 ■ Setting Up Input Sources

Understanding Field Formats

Packed. The value is stored in packed decimal format, i.e. there are two digits for each byte, except for the last byte, which consists of the last digit (0-9) and a sign. For example, with a hex display on the mainframe, 12345 in packed format would look like this:

```
12345C
| | |
| | |
| | | --Byte 3 contains 5C. C indicates a positive number.
| | |
| | | --Byte 2 contains 34
| | |
| | | --Byte 1 contains 12
```

Zoned decimal. Each digit is represented by a single byte. The first four bits of each byte make up the sign (F), and the last byte, where the first four bits will contain a C for a positive number or a D for a negative number. The last four bits of each byte contain the value of the digit. For example, with a hex display on the mainframe, 12345 in zoned decimal format would look like this:

```
F1F2F3F4C5
| | | | |
| | | | | --Byte 5 contains C5. C indicates a positive number.
| | | | |
| | | | | --Byte 4 contains F4
| | | | |
| | | | | --Byte 3 contains F3
| | | | |
| | | | | --Byte 2 contains F2
| | | | |
| | | | | --Byte 1 contains F1
```

Note: When extracting data in packed format, position and length should not be specified because they will be ignored. To see how Windows-based ODBC column types are formatted in ACR/Detail, see “[Formats for Windows-Based Column Types](#)” in the next section.

Binary. Has a decimal equivalent consisting of the decimal digits 0 through 9 and a sign. Binary stores numbers in hexadecimal (base 16), represented by the characters 0-9 and A-E. For example, with a hex display on the mainframe, 12345 in binary format would look like this:

```
3039
| |
| | --Byte 2 contains 39
| |
| | --Byte 1 contains 30
```

Unsigned packed. Same as packed except that there is no sign in the last byte. Two digits are stored in every byte and the value is always positive.

For example, with a hex display on the mainframe, 12345 in unsigned packed format would look like this:

```
012345
| | |
| | | --Byte 3 contains 45
| | |
| | | --Byte 2 contains 23
| | |
| | | --Byte 1 contains 01
```

For information on maximum lengths when extracting data in each format, see [“Maximum Lengths for Regular and Extended Data”](#) on page 89.

Formats for Windows-Based Column Types

The following table shows how Windows-based column types are formatted in ACR/Detail. Any column types not specified in the table will be treated as text.

Note: When extracting data in packed format, position and length should not be specified because they will be ignored.

Windows-Based Column Types	ACR/Detail Format
Double	Packed
Float	Packed
C-Double	Packed
C-Float	Packed
Numeric	Packed
Decimal	Packed
Integer	Binary
Small Integer	Binary
Var Char	Text
Long Var char	Text

7 ■ Setting Up Input Sources

Understanding Field Formats

Windows-Based Column Types	ACR/Detail Format
Date	Numeric = Packed
Time	Numeric = Packed
Timestamp	Numeric = Packed

Extract file name

When Extract from is either **Input Area** or **Extraction Variable**, you have the option of capturing a portion, or the entire File Name of the Input Source, and assign it to an Internal Item, Extended Internal Item, or an Extraction Variable.

The available options are:

No: The name of the input source file is not extracted.

Full, up to first 80 characters: The first 80 characters of the input source path are extracted.

By Position/Length: Specify the starting position (Position) and the length (Length) of the input source path to extract.

By Level: Extracts the text between the **Level From** and **To** level separator characters. You can extract a value up to **20** levels down.

Use this functionality you can extract a portion, or the entire name of the Input Source, and assign it to an Internal Item, Extended Internal Item, or an Extraction Variable. This option is available when Extract from is Input Area or Extraction Variable for Access Modes 1-4, 6, and 7.

The ability to extract the file name is available only for non-RDBMS files. You can extract the value using levels (up-to 20 level deep), that match the criteria specified in the Level separators, or using a position and length.

The available options are:

No: The name of the input source file is not extracted.

Full, up to first 80 characters: The first 80 characters of the input source path are extracted.

By Position/Length: Specify the starting position (Position) and the length (Length) of the input source path to extract.

By Level: Extracts the text between the **Level From** and **To** level separator characters. You can extract a value up to 20 levels down.

Example:

```

--- 0---|--- 10---|--- 20---|--- 30---|--- 40---|--- 50---|--- 60---|---
123456789|123456789|123456789|123456789|123456789|123456789|123456789|12
\\hostname\c$\Infogix\Detail32\Samples\Mandays1.rpt

```

If the File Name of the Input Source is
\\hostname\c\$\Infogix\Detail32\Samples\Mandays1.rpt, and
Extract file name option is

Full, up to first 80 characters, the extracted value is:

\\hostname\c\$\Infogix\Detail32\Samples\Mandays1.rpt

By Position/Length, when Position is 41 for a Length of 20, the extracted value is: **Mandays1.rpt**

By Level, with a Level From of 1 To 4, and Level separator \, the extracted value is: **\\hostname\c\$**

By Level, with a Level From of 10 To 10, and Level separator \, the extracted value is: **Mandays1.rpt**

Relative position direction

Instead of specifying a fixed starting position to extract the value of a field, you can now pick up the value dynamically based upon the actual start position of the select field using the relative position direction indicator feature. Specify the direction of data selection relative to the last successful Select field. This feature is available for Selection, Reformat, Key, and Detail field types in Access Modes 6, 7, and 8.

Valid options are **As Is**, **Left**, or **Right**. This field is enabled only for non-DDF file, and when the **Select From** is "Input Area".

For example, a Position of 4, Length of 5, and a Relative position direction of Right will result in a rule that will select a value that is 5 characters long but starting from the 4th position to the Right of the last Selection field.

Setting Up an Input Source for Relational Data

There are two methods of setting up input sources for relational database tables (File Organization/Types: DB2 Table, Oracle, or ODBC).

Notes Regarding ODBC Extraction

Windows Only

If you will be extracting relational data using your own ODBC driver, ensure that you have established an ODBC connection as follows: From the Control Panel in Windows, select **Administrative Tools > Data Sources (ODBC)**. Click on the System DSN tab and set up the connection through the ODBC Data Source Administrator.

Extraction via your ODBC driver can be accomplished using either of the methods discussed in the next section, "[Relational Data Extraction Methods](#)." The procedure for each method explains how to set the datasource name on the File Organization dialog box when the **File/Organization Type** is ODBC.

UNIX or Linux Only

Verify with your UNIX or Linux database administrator that the UNIX or Linux data source for your ODBC session has been set up.

There are two methods of setting up input sources for relational database tables (File Organization/Types: DB2 Table, Oracle, or ODBC).

Relational Data Extraction Methods

The method you will use to extract relational data is determined by your entry for the **Use Freeform SQL Statements** field in the File Organization dialog box.

- If you select **Use Freeform SQL Statements**, you will be provided with dialog boxes for pasting in a tested, valid SQL query. See the section below.
- If you do not select **Use Freeform SQL Statements**, you will enter the extraction definitions and the query will be formatted for you. See "[Using the RDBMS Extraction Method](#)" on page 155.

Note: Extraction definitions for relational database tables cannot be ported across platforms.

Using Freeform SQL Statements

The freeform SQL statements method enables you use tested, valid SQL queries in ACR/Detail jobs. This method supports lengthy, complex queries, including table joins. Dialog boxes enable you to paste in the SELECT, FROM, and WHERE clauses.

Note: If a floating-point value is extracted by ACR/Detail using the Freeform SQL feature, the value is processed as an approximate numeric datatype. The value has an approximate representation of decimal value, rather than an exact representation. In general, a number is approximately represented as a fixed number of significant digits (significand) and scaled using an exponent. The base for the scaling is normally two, ten, or sixteen. In ACR/Detail, the significand is 15.

Example:

Using the Freeform SQL feature via the ODBC or Oracle Direct Access method, the floating-point value 678901234567890.987654999 becomes 678901234567891.000000000. After the floating-point value is extracted, processed and rounded by ACR/Detail the approximate representation of the floating-point value in decimal format is 9. 678901234567891.000000000. The significand in this case is 15.

Restrictions

- Queries that require a value from a WHERE clause to an internal item or extraction variable are not supported. Only a SELECT clause can return a value to an item.
- Queries that use an * in the SELECT clause are not supported. Column names must be specified so that the returned values can be mapped to internal items.
- The Freeform SQL dialog boxes are designed to process valid, tested SQL. They do not check the validity of SQL. The only dialog box edits are as follows:
 - In the Freeform SQL Select Clause dialog box: The statement must begin with SELECT. If column functions are included, an AS statement must be entered for each function. Fully qualified column names are limited to 160 bytes. The TOP keyword can be used if supported by your database. The percent keyword is not currently supported.
 - In the Freeform SQL From Clause dialog box, the statement must begin with FROM.

7 ■ Setting Up Input Sources

Setting Up an Input Source for Relational Data

- In the Freeform SQL Where Clause dialog box, the statement must begin with WHERE. You can include other clauses within the WHERE clause, but the first word must be WHERE.
- Only one SELECT clause, one FROM clause, and one WHERE clause can be defined.

Procedure for Using Freeform SQL Statements

1. From Control Task (Job) View, select **Input Sources > New**.
2. In the Basic Information section of the view, complete the Name dialog box.
3. Complete the File Organization dialog box. The File Organization/Type should be DB2 Table, Oracle, or ODBC. The fields that display will depend on the type you select.

If 1) you select DB2 Table, 2) the job will be run on the mainframe, and 3) you want the SQL to display in SYSOUT at run-time, select **Display SQL in SYSOUT**.)

Select the **Use Freeform SQL Statements** field. The RDBMS Tables folder will be replaced with the Freeform SQL folder in Input Source View.

If **File Organization/Type** = ODBC, the data source name can be either of the following:

- A file data source (file-based DSN).
- A machine data source (DSN). This can be a “user” data source (user-dedicated) or a “system” data source (available to all users on a specific computer or system of computers)

The method you use depends on the type of configuration you are using:

If you are using a Windows configuration, click the browse button, click the appropriate tab, and select the data source. Depending on the type of data source you select, additional information may be required. If so, complete the dialog boxes that will display and you will be returned to the File Organization dialog box.

If you are using a UNIX or Linux configuration, enter the name of the data source directly in the **Data Source Name** field (do not use the browse button).

4. Complete the Key Matching dialog box if appropriate.

5. In the Freeform SQL folder, click **New > Select Clause** to display the Freeform SQL Select Clause dialog box and paste in the SELECT clause from your valid SQL query.
6. Click **New > From Clause** to display the Freeform SQL From Clause dialog box and paste in the FROM clause from your query.
7. Click **New > Where Clause** to display the Freeform SQL WHERE clause dialog box. You can paste in the WHERE clause from your query as well as any clauses other than SELECT and FROM. To include an extraction variable in the WHERE statement: 1) Select it from the list at the bottom of the dialog box, 2) select the **Use Extended format** and/or **Use Numeric Type** fields as appropriate, and select **Add Extraction Variable to Where Clause**. For information on extraction variable codes, see “[Extraction Variable Codes for Relational Database Extraction](#)” on page 157.
8. In the Extraction Definitions folder, select **New** and specify one or more key fields and/or one or more detail fields.
On both the Key Field dialog box and the Detail Field dialog box, the Position and Length fields should not be completed unless you are extracting a portion of the column.
9. Click the **Save** icon to save the job with the new input source and its file definitions.

Using the RDBMS Extraction Method

This method builds the query dynamically based on your extraction definitions.

Before you begin, review each table you will use in your input source and note the exact starting position, length, and format of each field that you want to extract.

Procedure for the RDBMS Extraction Method

1. From Control Task (Job) View, select **Input Sources > New**.
2. In the Basic Information section of the view, complete the Name dialog box.
3. Complete the File Organization dialog box. The fields that display will depend on the type you select.
If 1) you select DB2 Table, 2) the job will be run on the mainframe, and 3) you want the SQL to display in SYSOUT at run-time, select **Display SQL in SYSOUT**.)
The **Use Freeform SQL Statements** field should not be selected.

7 ■ Setting Up Input Sources

Setting Up an Input Source for Relational Data

If **File Organization/Type** = ODBC, the data source name can be either of the following:

- A file data source (file-based DSN).
- A machine data source (DSN). This can be a “user” data source (user-dedicated) or a “system” data source (available to all users on a specific computer or system of computers)

The method you use depends on the type of configuration you are using:

If you are using a Windows configuration, click the browse button, click the appropriate tab, and select the data source. Depending on the type of data source you select, additional information may be required. If so, complete the dialog boxes that will display and you will be returned to the File Organization dialog box.

If you are using a UNIX or Linux configuration, enter the name of the data source directly in the **Data Source Name** field (do not use the browse button).

4. Complete the Key Matching dialog box if appropriate.
5. In the RDBMS Tables folder, click **New** and complete the information for the primary relational database table that contains the data to be extracted.
6. Click **New** again and complete the data for each additional table from which you want to extract data for reconciliation. The Browse button is available only if the File/Organization type specified is ODBC.
7. Complete the Extraction Definitions section of the view.
 - a. Click the **Selection** icon to display the WHERE clause dialog box. The WHERE clause provides criteria to select rows to retrieve from the table. Each WHERE clause, with its corresponding key and detail fields, makes up a selection group. If you create multiple selection groups, they must be concatenated into a single query. To enable this, in the **Literal Continuation** field, select Keep trailing spaces.

You can optionally include one or more extraction variables by entering the codes described in the next section. Extraction variables must be only on the left side of the comparison.
 - b. Click the **Key** and **Detail** icons to create any required key fields and/or detail fields.
 - c. Click the **Save** icon to save the job with the new input source and its file definitions.

Extraction Variable Codes for Relational Database Extraction

Whether you include extraction variable codes by selecting from a list using freeform SQL statements or you enter the codes manually using the RDBMS extraction method, the codes must conform to the guidelines and formats described below.

Note: ■ **If you are using freeform SQL statements**, extraction variables are entered in the Freeform SQL Where clause dialog box and can appear on either or both sides of the comparison. For example:

Field name1 or extraction variable = field name2 or
extraction variable

■ **If you are using the RDBMS extraction method**, any extraction variables are entered in the WHERE clause dialog box. Any extraction variables must be only on the left side of the comparison. For example:

Field name1 or extraction variable = field name2

Format for a regular extraction variable:

:CV-*nnn* (for a text extraction variable column) or :NV-*nnn* (for a numeric extraction variable)

where *nnn* is the number of the extraction variable (for example, 001).

:CV-*nnn* or :NV-*nnn* must be followed by a space unless it is the only thing in the field.

If you want to concatenate text type extraction variables, 2 regular extraction variables can be combined to make a longer text value. For example,

:CV-001 :CV-002 = 'upto16characters'
 ^ ^
 space space

Format for an extended extraction variable:

:CXV-*nnn* (for a text extraction variable column) or :NXV-*nnn* (for a numeric extraction variable).

Two text type extended extraction variables can be combined to make a text value up to 160 characters long. For example::

:CXV-001 :CXV-002 = 'upto160characters'
 ^ ^
 space space

7 ■ Setting Up Input Sources

Date and Cycle Number Extraction and Processing

Format for a field name

To specify a field name, just enter the correct field name. Case-sensitivity depends on your database. Spaces always count.

Date and Cycle Number Extraction and Processing

Specifying the Length and Format of the Date to Extract

Length Should Include Date Separators

If the date uses separators such as spaces, commas, slashes, or periods, include the separators in the length. The date separators will be stripped out when the value is stored. For example:

01/15/2005 is a length of 10.

Jan. 01, 2000 is a length of 13.

Date or Cycle Format

A date or cycle specified for extraction should be formatted as numeric unless it includes a 3-character month (mmm). A date or cycle that includes a 3-character month should be formatted as text or character.

Date Formats

Accept Date Number As Is (7)

This format is available in the **Cycle Format** field in the Key Field dialog box when you are specifying a date to be extracted for use in assigning the cycle number. It indicates that, in general, the extracted value will be used as the cycle number with no changes. The (7) shows the value 7 will represent this format in the batch transaction record.

If you extract fewer than 8 digits using this format, the system will fill the leftmost positions with zeros.

Note: This format cannot be used when extracting a value that includes a 3-character month.

Date Formats that Use DD, MM, C, CC, YY, JJ, and/or MMM

Note: Date formats use the following conventions:

dd represents the day number. Days less than 10 must have a leading 0. For example, Jan 1, 2006 cannot be extracted, but Jan 01, 2006 can be extracted.

mm represents the 2-digit month number.

c represents the 1-digit century, where 0 = 19 (as in 1999) and 1 = 20 as in 2010.

cc represents the 2-digit century (CC).

yy represents the 2-digit year.

jjj represents the 3-digit Julian day.

mmm represents the 3-character month (JAN, FEB, etc.).

The formats listed below are available for extracting dates regardless of whether the date will be used as a cycle number. These formats identify the format and sequence of the day, month, century, and year in the input source.

The date separators (/) shown are not relevant because they will be stripped out. The number or letter in parentheses following each format shows the value that will represent that format in the batch transaction record.

yy/mm/dd (1)

mm/dd/yy (2)

mm/dd/ccyy (3)

dd/mm/yy (4)

yy/jjj (5)

dd/mm/ccyy (6)

cy/mm/dd (8)

ccyy/mm/dd (9)

cc/yyjjj (A)

dd/mmm/yy (B)

dd/mmm/ccyy (C)

mmm/dd/yy (D)

mmm/dd/ccyy (E)

Users can extract the day or month elements when their values do not contain leading zeros. For example, if you want to extract an input date value of 3/1/2018 (March 1, 2018), specify the date format (MMDDCY) and the Date Separator character as / (forward slash).

7 ■ Setting Up Input Sources

Date and Cycle Number Extraction and Processing

The system processes the values as March 1, 2018 based on the date format and separator, eliminating the need to modify the input data to append a leading zero. Leave blank, if the input data has leading zeros.

Date separator is enabled only when Extract from is Input Area, Field Type is Last Date or First Date, and when translating Date values.

Note: On the Detail Field dialog box, date formats containing 3-character months cannot be extracted from a literal.

Calculation of Century Values

When the century is not extracted as part of the date, (in this case, the date format contains dd, mm, yy, jjj, and/or mmm but no c or cc), the century is calculated automatically. A century value of 20 is assigned for years less than or equal to 80. Otherwise a century value of 19 is assigned.

Storage of Extracted Dates and Cycle Numbers

Storage of Extracted Dates

After the century is calculated (if required), extracted dates that are not used as cycle numbers are stored internally as date count values.

Storage of Extracted Cycles

After the century is calculated (if required), extracted dates that are used as cycle numbers are stored internally in CCYYMMDD format.

Display and Printing of Dates

The default format for display and printing of dates is MM/DD/YY.

Note: Even if you select one of the formats that contain the century, such as DDMMCCYY, to format dates on reports, only the last 2 digits of the year will print unless you select the **Print century before year** field on the User Options dialog box - Date Formats tab.

You can control the following defaults through your user options as described in “[Setting User Options](#)” on page 173:

- Order of the date elements.
- Character to use as the date separator (instead of the default slash mark).
- Display of leading zeros in days and months.

- Whether the century prints before the year.
You may want to print the century to verify that the correct century was calculated (see “[Calculation of Century Values](#)” on page 160). Another way to verify that the correct century was calculated is to run the List History utility for detail values of the job ID in question and examine the date count values stored in internal or extended internal items. If a date count value is more than 145731 (December 31, 1999), the century value was assumed to be 20.

Time Extraction and Processing

The Add Time Elements feature supports time data extraction. The extracted time data is converted to centiseconds before it is stored in an internal item or an extended internal item.

Date & Time item entries are now allowed in the LHS and RHS of a calculated item and balancing rule.

Time Format and Notation

The supported time formats are: HH:SS, HH:MM:SS, and HH:MM:SS.NN. The default time format is HH:MM:SS. The supported time notations are: 12-hour format and the 24-hour format. The default time notation is 24-hour format.

Print Format Options

To print the time element values in reports, two new print formats: ‘Time’, and ‘Date & Time’ are added.

When the print format is ‘Time’, time element is printed as ‘-99999999 TTTTTTTTTTTTTT’ format where -99999999 is the relative day and TTTTTTTTTTTTTT is the formatted time.

When the print format is ‘Date & Time’, time element is printed as ‘DDDDDDDD TTTTTTTTTTTTTT’ where DDDDDDDD is the formatted date and TTTTTTTTTTTTTT is the formatted time.

The date output options and the time output options in the User Options File are used to format date and time.

Time Conversion Utility

Time Conversion Utility is a new online utility that converts the time values. The available options are:

7 ■ Setting Up Input Sources

Deleting File Definitions

- Convert days/time to centiseconds.
- Convert centiseconds to days/time.

Deleting File Definitions

To delete a file definition, in the Input Source view, highlight the definition and press the **Delete** key. Respond to the prompt to confirm the deletion. Multiple definitions can be deleted at a time.

Extracting Directly from a Delimited Data File

You can extract delimited values directly from input source records in delimited data files. This feature allows you to specify multiple delimiters, including the tab character, for keys, items, and translation tables.

Note: On Windows, you must change a Micro Focus COBOL default to enable proper processing of tab-delimited input sources. See [Enable Proper Processing of Tab-Delimited Input Sources](#) on page 324.

Procedure for Extracting Delimited Fields Directly

1. When setting up your input source file, on the File Organization dialog box, (or Table Dataset Organization dialog box for external translation tables) your **File Organization/Type** (or **Dataset Organization** for external translation tables) must be Physical Sequential. Select **Use delimited data fields** and complete the related fields that display at the bottom of the dialog box after you make this selection.
2. In your extraction definitions, (or Table Build Rules for external translation tables) complete the **Delimited data field number** field as follows: If you are extracting from the delimited area of the file, enter the field number (this number is displayed automatically if you used the mark-and-capture method). Otherwise leave the field blank.

Limitations

When extracting directly from delimited data files, the following limitations apply:

- Direct delimited data extraction is supported only for physical sequential files.

- Direct delimited data extraction is allowed only if the data is extracted directly from the input record. It does not apply when extracting from extraction variables or literals.
- Maximums for a delimited data file:
 - The maximum length allowed for the fixed data area is 1000.
 - The maximum number of delimited data fields allowed per record is 1,000.
- For tab-delimited files, the following limitations apply, but you can work around them:
 - The data shows in the Extract Window in the columnar (tabbed) view. This prevents automatic calculation of the number for the delimited field. When using tab-delimiters, please manually count the field number and enter it appropriately.
 - On Windows, you must change a Micro Focus COBOL default to enable proper processing of tab-delimited input sources. See [“Enable Proper Processing of Tab-Delimited Input Sources” on page 324](#).
- Delimited data cannot be reformatted to the note area.
- Partial extraction is allowed for all field formats. To extract the whole field, leave the position and length fields set to zeros or spaces.

Matched File Processing

Consider a scenario where you have two input files – one that contains 100K records and another one that contains only a smaller subset of those records. You have an use case to identify and report on the data in the smaller file that does not match with the data in the bigger file.

Defining your input sources to use access mode 8, will enable the application to read the two files simultaneously, and apply the matching criteria. This type of processing results in significant performance improvement as the number of system “writes” are reduced when a match is found.

For each input source, matched records or unmatched records will be extracted and reported based on the specified Matching Options. The matching criteria can be applied at two levels:

- Key Level

Item Level (when there is a match at the Key level)

When using match file processing , the following limitations apply:

7 ■ Setting Up Input Sources

Extracting Directly from a Delimited Data File

- Match file processing is supported for Physical Sequential, and VSAM Key Sequenced file types only.
- Data in the two input sources should contain unique keys, and be in an ascending Key order. Warning or an Error is generated if keys are not sorted, or duplicated.
- Stage 2 processing is not available in this mode
- For each job, only two input sources can be defined to use matched file processing. The first input source defined to use matched file processing is considered as Match File 1. The second input source specified to use matched file processing is considered as Match File 2.
- If a job contains more than two input sources, then only the last two input sources may be defined in access mode 8.
 - There may be a situation where you have to setup more than two input sources. The main purpose of setting up additional input sources may be to pick up data (e.g. cycle number) and pass it to the match files (using an extraction variable) for record selection. Additional input sources can be specified together with the two access mode 8 input sources. However these input sources have to be setup and processed before the two access mode 8 input source files. No key data is allowed to be extracted from those non-access mode 8 input sources.

Matched File Processing Example

Consider a scenario where you have two input files –

- File 1 contains all US State names and their Capitals
- File 2 contains a subset of the State names with some erroneous Capitals.

File 1 (Input Source 1)	
STATE	CAPITAL
Alabama	Montgomery
Arizona	Phoenix
Arkansas	Little Rock
California	Sacramento
Colorado	Denver
Delaware	Dover
Florida	<i>Tallahassee</i>
Georgia	Atlanta
Illinois	<i>Springfield</i>

File 2 (Input Source 2)	
STATE	CAPITAL
Alabama	Montgomery
Arizona	Phoenix
Arkansas	Little Rock
California	Sacramento
Colorado	Denver
Delaware	Dover
Florida	<i>Orlando</i>
Georgia	Atlanta
Illinois	<i>Chicago</i>

Indiana	Indianapolis
Iowa	Des Moines
Kentucky	Frankfort

Kentucky	Frankfort
New York	Albany
Texas	Austin

The matching criteria can be applied at two levels:

- Key Level
- Item Level (when there is a match at the Key level)

Valid matching options are:

Default: Data extracted depends on the Matching Option specified for the other input source.

Matched records: Following records are extracted:

- Key exists in both input sources.
- For the Key that exists in both input sources above, only if the values of the items match.

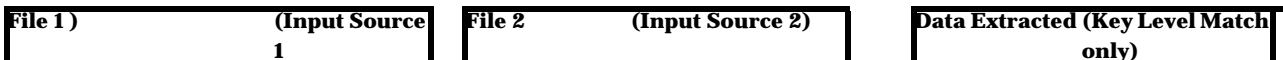
Unmatched records: Following records are extracted:

- Keys that are not found in the other input source.
- If the key exists in both input sources, but the values of the items do not match.

Data extraction depends on the combination of Matching Options specified for the two input sources. Following matching combinations are supported:

Extract Matched Data:

Match Option (Input Source 1)	Match option (Input Source 2)	Matching Output
Matched	Matched	Matched records from both input sources
Matched	Default	Matched records from both input sources
Default	Matched	Matched records from both input sources



7 ■ Setting Up Input Sources

Extracting Directly from a Delimited Data File

KEY	I-001
Alabama	Montgomery
Arizona	Phoenix
Arkansas	Little Rock
California	Sacramento
Colorado	Denver
Delaware	Dover
Florida	<i>Tallahassee</i>
Georgia	Atlanta
Illinois	<i>Springfield</i>
Indiana	Indianapolis
Iowa	Des Moines
Kentucky	Frankfort

KEY	I-002
Alabama	Montgomery
Arizona	Phoenix
Arkansas	Little Rock
California	Sacramento
Colorado	Denver
Delaware	Dover
Florida	<i>Orlando</i>
Georgia	Atlanta
Illinois	<i>Chicago</i>
Kentucky	Frankfort
New York	Albany
Texas	Austin

KEY	I-001	I-002
Alabama	Montgomery	Montgomery
Arizona	Phoenix	Phoenix
Arkansas	Little Rock	Little Rock
California	Sacramento	Sacramento
Colorado	Denver	Denver
Delaware	Dover	Dover
Florida	<i>Tallahassee</i>	<i>Orlando</i>
Georgia	Atlanta	Atlanta
Illinois	<i>Springfield</i>	<i>Chicago</i>
Kentucky	Frankfort	Frankfort

File 1 (Input Source 1)	
KEY	I-001
Alabama	Montgomery
Arizona	Phoenix
Arkansas	Little Rock
California	Sacramento
Colorado	Denver
Delaware	Dover
Florida	<i>Tallahassee</i>
Georgia	Atlanta
Illinois	<i>Springfield</i>
Indiana	Indianapolis
Iowa	Des Moines
Kentucky	Frankfort

File 2 (Input Source 2)	
KEY	I-002
Alabama	Montgomery
Arizona	Phoenix
Arkansas	Little Rock
California	Sacramento
Colorado	Denver
Delaware	Dover
Florida	<i>Orlando</i>
Georgia	Atlanta
Illinois	<i>Chicago</i>
Kentucky	Frankfort
New York	Albany
Texas	Austin

Data Extracted (I-001 to I-002 Level Match)		
KEY	I-001	I-002
Alabama	Montgomery	Montgomery
Arizona	Phoenix	Phoenix
Arkansas	Little Rock	Little Rock
California	Sacramento	Sacramento
Colorado	Denver	Denver
Delaware	Dover	Dover
Florida	Fallahassee	Orlando
Georgia	Atlanta	Atlanta
Illinois	Springfield	Chicago
Kentucky	Frankfort	Frankfort

Extract Unmatched Records from First Input Source :

Match Option (Input Source 1)	Match option (Input Source 2)	Matching Output
Unmatched	Default	Unmatched records from input source 1

- Records in input source 1 with a missing key from input source 2 are extracted.

Setting Up Input Sources ■ 7

Extracting Directly from a Delimited Data File

- Records where the key in both the input sources match, but whose items do not match are extracted

File 1)	(Input Source 1
KEY	I-001
Alabama	Montgomery
Arizona	Phoenix
Arkansas	Little Rock
California	Sacramento
Colorado	Denver
Delaware	Dover
Florida	<i>Tallahassee</i>
Georgia	Atlanta
Illinois	<i>Springfield</i>
Indiana	Indianapolis
Iowa	Des Moines
Kentucky	Frankfort

File 2	(Input Source 2)
KEY	I-002
Alabama	Montgomery
Arizona	Phoenix
Arkansas	Little Rock
California	Sacramento
Colorado	Denver
Delaware	Dover
Florida	<i>Orlando</i>
Georgia	Atlanta
Illinois	<i>Chicago</i>
Kentucky	Frankfort
New York	Albany
Texas	Austin

Data Extracted (Key Level Match only)		
KEY	I-001	I-002
Indiana	Indianapolis	
Iowa	Des Moines	

File 1 (Input Source 1)	
KEY	I-001
Alabama	Montgomery
Arizona	Phoenix
Arkansas	Little Rock
California	Sacramento
Colorado	Denver
Delaware	Dover
Florida	<i>Tallahassee</i>
Georgia	Atlanta
Illinois	<i>Springfield</i>
Indiana	Indianapolis
Iowa	Des Moines
Kentucky	Frankfort

File 2 (Input Source 2)	
KEY	I-002
Alabama	Montgomery
Arizona	Phoenix
Arkansas	Little Rock
California	Sacramento
Colorado	Denver
Delaware	Dover
Florida	<i>Orlando</i>
Georgia	Atlanta
Illinois	<i>Chicago</i>
Kentucky	Frankfort
New York	Albany
Texas	Austin

Data Extracted (I-001 to I-002 Level Match)		
KEY	I-001	I-002
Florida	<i>Tallahassee</i>	<i>Orlando</i>
Illinois	<i>Springfield</i>	<i>Chicago</i>

7 ■ Setting Up Input Sources

Extracting Directly from a Delimited Data File

Extract Unmatched Records from Second Input Source :

Match Option (Input Source 1)	Match option (Input Source 2)	Matching Output
Default	Unmatched	Unmatched records from input source 2

- Records in input source 2 with a missing key from input source 1 are extracted.
- Records where the key in both the input sources match, but whose items do not match are extracted

File 1)	(Input Source 1)
KEY	I-001
Alabama	Montgomery
Arizona	Phoenix
Arkansas	Little Rock
California	Sacramento
Colorado	Denver
Delaware	Dover
Florida	<i>Tallahassee</i>
Georgia	Atlanta
Illinois	<i>Springfield</i>
Indiana	Indianapolis
Iowa	Des Moines
Kentucky	Frankfort

File 2	(Input Source 2)
KEY	I-002
Alabama	Montgomery
Arizona	Phoenix
Arkansas	Little Rock
California	Sacramento
Colorado	Denver
Delaware	Dover
Florida	<i>Orlando</i>
Georgia	Atlanta
Illinois	<i>Chicago</i>
Kentucky	Frankfort
New York	Albany
Texas	Austin

Data Extracted (Key Level Match only)		
KEY	I-001	I-002
New York	Albany	
Texas	Austin	

File 1 (Input Source 1)	
KEY	I-001
Alabama	Montgomery
Arizona	Phoenix
Arkansas	Little Rock
California	Sacramento
Colorado	Denver
Delaware	Dover
Florida	<i>Tallahassee</i>
Georgia	Atlanta
Illinois	<i>Springfield</i>

File 2 (Input Source 2)	
KEY	I-002
Alabama	Montgomery
Arizona	Phoenix
Arkansas	Little Rock
California	Sacramento
Colorado	Denver
Delaware	Dover
Florida	<i>Orlando</i>
Georgia	Atlanta
Illinois	<i>Chicago</i>

Data Extracted (I-001 to I-002 Level Match)		
KEY	I-001	I-002
Florida	<i>Tallahassee</i>	<i>Orlando</i>
Illinois	<i>Springfield</i>	<i>Chicago</i>

Setting Up Input Sources ■ 7

Extracting Directly from a Delimited Data File

Indiana	Indianapolis
Iowa	Des Moines
Kentucky	Frankfort

Kentucky	Frankfort
New York	Albany
Texas	Austin

Extract Unmatched Records from both Input Sources

Match Option (Input Source 1)	Match option (Input Source 2)	Matching Output
Unmatched	Unmatched	Unmatched records from input source 2

- Records in input source 1 with a missing key from input source 2 are extracted.
- Records in input source 2 with a missing key from input source 1 are extracted
- Records where the key in both input sources match, but whose items do not match are extracted

File 1)	(Input Source 1
KEY	I-001
Alabama	Montgomery
Arizona	Phoenix
Arkansas	Little Rock
California	Sacramento
Colorado	Denver
Delaware	Dover
Florida	<i>Tallahassee</i>
Georgia	Atlanta
Illinois	<i>Springfield</i>
Indiana	Indianapolis
Iowa	Des Moines
Kentucky	Frankfort

File 2	(Input Source 2)
KEY	I-002
Alabama	Montgomery
Arizona	Phoenix
Arkansas	Little Rock
California	Sacramento
Colorado	Denver
Delaware	Dover
Florida	<i>Orlando</i>
Georgia	Atlanta
Illinois	<i>Chicago</i>
Kentucky	Frankfort
New York	Albany
Texas	Austin

Data Extracted (Key Level Match only)		
KEY	I-001	I-002
Indiana	Indianapolis	
Iowa	Des Moines	
New York		Albany
Texas		Austin

File 1 (Input Source 1)	
KEY	I-001
Alabama	Montgomery
Arizona	Phoenix

File 2 (Input Source 2)	
KEY	I-002
Alabama	Montgomery
Arizona	Phoenix

Data Extracted (I-001 to I-002 Level Match)		
KEY	I-001	I-002
Florida	<i>Tallahassee</i>	<i>Orlando</i>
Illinois	<i>Springfield</i>	<i>Chicago</i>

7 ■ Setting Up Input Sources

Extracting Directly from a Delimited Data File

Arkansas	Little Rock
California	Sacramento
Colorado	Denver
Delaware	Dover
Florida	<i>Tallahassee</i>
Georgia	Atlanta
Illinois	<i>Springfield</i>
Indiana	Indianapolis
Iowa	Des Moines
Kentucky	Frankfort

Arkansas	Little Rock
California	Sacramento
Colorado	Denver
Delaware	Dover
Florida	<i>Orlando</i>
Georgia	Atlanta
Illinois	<i>Chicago</i>
Kentucky	Frankfort
New York	Albany
Texas	Austin

Invalid Matching Combinations

Match Option (Input Source 1)	Match option (Input Source 2)	Matching Output
Matched	Unmatched	Invalid
Unmatched	Matched	Invalid
Default	Default	Invalid

- Records in input source 1 with a missing key from input source 2 are extracted.

Access Mode 8 (Use Matched File Processing)

Use Access Mode 8 to read data from two input sources (or from one input source) concurrently to identify:

- Key matches
- Key mismatches
- if values of internal/extended internal items match (when the Key match)
- if values of internal/extended internal items mismatch (when the Key match)

This mode is available only for **Physical Sequential** and **VSAM Key Sequenced** file types.

Procedure for Configuring Matched File Processing

File Organization

When setting up your input source file, on the File Organization dialog box, your **File Organization/Type** must be Physical Sequential or VSAM Key Sequenced type.

Select **Access Mode 8 (Use matched file processing)**

Select a **Matching Option**: Default, Matched records, Unmatched records

Select an appropriate **Action to take when keys are not unique or not in ascending order**: Issue warning and continue processing, Issue error and stop processing

Internal Item/Extended Internal Item Configuration

To define the matching criteria, define a new Internal Item or Extended Internal Item. Or, select an existing Internal Item/Extended Internal Item if it already exists.

Select **Match Item Type**: Select the target item type with which you wish to match the value of this Internal Item.

Select **None**, if you want to match the values only at the Key level.

Select **Internal Item**, if you want to compare the value of this internal item with another internal item.

Select **Extended Internal Item**, if you want to compare the value of this internal item with an extended internal item.

Select **Match Item Value**, the name of the item whose value you wish to match against.

Refer to context sensitive help for more details.

7 ■ Setting Up Input Sources

Extracting Directly from a Delimited Data File

Setting the User Options

This chapter explains the user options feature. User options, which are stored in the user options file `udsuf.dat`, enable you to override a variety of system defaults. User options include the following categories: miscellaneous, currency formats, date/time formats, negative signs, and XML options.

Some user options settings can be overridden within specific job and file definitions.

This chapter includes the following sections:

- “The User Options File (`udsuf.dat`)” on page 173
- “Setting User Options” on page 173
- “Setting User Options on UNIX or Linux” on page 174

The User Options File (`udsuf.dat`)

User options are stored in the user options file (`udsuf.dat`). This file:

- Is accessed for every reconciliation job that is run.
- Is a flat file that contains an 80-character record specifying user options and parameters. These will affect every report that is produced by the system.
- Is not required. System defaults are available if it does not exist.

Setting User Options

Setting User Options in Windows Mode

If you set user options in Windows mode, your settings will be stored in the user options file (`udsuf.dat`) on the Windows server. These options will apply to all ACR/Detail jobs run by all users on the Windows server.

8 ■ Setting the User Options

Setting User Options on UNIX or Linux

To set the user options for jobs running on the Windows server, do the following:

1. When setting up the configuration, you will be completing the Add/Edit Configurations dialog box - File Information tab. Specify the **User options file** you want to use on this tab. Valid options include the following:
 - a. Use user option file specified by server.
 - b. Use local user option file (will always be copied to server)
 - c. Use personal user option file on server (specify path)
2. After the configuration is set up, select it and then select **Options > User Options** and complete the tabs containing options you want to change. Your changes will be stored in the user options file that you selected in the previous step.

Setting User Options for UNIX or Linux Mode

To set user options for UNIX or Linux mode, see the next section.

Setting User Options on UNIX or Linux

Overview

If you want to customize the user options for running jobs from ACR/Detail Client in UNIX or Linux mode or for running scripts without ACR/Detail Client, you need to edit the user options (udsuf.dat) file using an editor such as vi. By default this file is located in the infogix/license/files directory on the UNIX or Linux host.

Note: ACR/Summary and ACR/Detail share a single udsuf.dat file on the UNIX or Linux host. Some options that are valid on ACR/Summary are not valid on ACR/Detail, as noted in [UNIXOPT Record Layout on page 175](#).

Modifications you make on the UNIX or Linux host to udsuf.dat options that are valid on both products will apply to all scripts and to all users of both products, if you have both.

The records on `udsuf.dat` that you may want to customize are UNIXOPT and UNIEXEO. See the following sections for more information and explanations of each option:

- “Execution Options Record: UNIXOPT” on page 175
- “Bypass Processing and Return Code for No Extraction: UNIEXEO” on page 184

Execution Options Record: UNIXOPT

UNIXOPT Record Layout

The Option Type, located in the first ten characters of the UNIXOPT record, is UNIXOPT. The rest of this record consists of fields for specifying user options.

Note: ACR/Summary and ACR/Detail share a single user options file. Options not applicable to this product are noted in the Description column.

The record layout is as follows:

UNIXOPT Record Layout					
Field	Description	Position	Length	Format	Value
1	Type	1-10	10	X(10)	UNIXOPT
2	Reserved (used by ACR/Summary only)	11-12	2	X(2)	Blanks
3 ¹	Currency Information Provided	13	1	X(1)	Y, N, blank = N
4	Thousands Place Marker	14	1	X(1)	
5	Decimal Place Marker	15	1	X(1)	
6	Currency Symbol	16-21	6	X(6)	
7	Second Currency Symbol	22-27	6	X(6)	
8	Currency Padding Length	28	1	9(1)	0-8
9	Currency Symbol Location	29	1	9(1)	2,4,6 (front, back, both)
10	Currency Symbol Length	30	1	9(1)	1-6
11	Negative Sign	31-36	6	X(6)	

8 ■ Setting the User Options

Setting User Options on UNIX or Linux

UNIXOPT Record Layout					
Field	Description	Position	Length	Format	Value
12	Second Negative Sign	37-42	6	X(6)	
13	Negative Sign Padding Length	43	1	9(1)	0-8
14	Negative Sign Location	44	1	9(1)	2,4,6 (front, back, both)
15	Negative Sign Length	45	1	9(1)	1-6
16 ²	Date Information Provided	46	1	X(1)	Y, N, blank= N
17	Date Output Format	47-49	3	X(3)	YMD, MDY, etc.
18	Date Separator	50	1	X(1)	
19	Date Zero Suppress Indicator	51	1	X(1)	Y, N
20	Century Indicator	52	1	X(1)	Y, N
21	Reserved (used by ACR/S only)	53-54	2	X(2)	Blanks
22	Print Commas for Counts	55	1	X(1)	Y, N, blank=Y
23	Print Commas for Amounts	56	1	X(1)	Y, N, blank=Y
24	22-Character Numeric	57	1	X(1)	Y, N, blank=N
25	Initialize text items to spaces	58	1	X(1)	Y, N, blank=N
26 ³	Control Report XML Option	59	1	X(1)	1 or blank, 2, 3
27 ⁴	User Report XML Option	60	1	X(1)	1 or blank, 2, 3
28 ⁵	XML DB Fail Option	61	1	X(1)	1 or blank, 2, 3
29 ⁶	XML DB Message Size (in 10,000s)	62-65	4	X(4)	Blank, 0 through 1677
30 ⁷	Free-Form Report XML Option	66	1	X(1)	1 or blank, 2, 3
31	Time Separator	67-67	1	X(1)	
32 ⁸	Time Output Format	68-68	1	X(1)	1, 2, 3, blank
33 ⁹	Time Output Notation	69-69	1	X(1)	1, 2, blank
34	Reserved (used by ACR/ Summary only)	70-78	9	X(9)	Blanks

UNIXOPT Record Layout

Field	Description	Position	Length	Format	Value
35	Send Alternate Job ID	79-79	1	X(1)	Y, N, blank
36	Reserved	80-80	1	X(1)	Blank

1 Currency Information Provided - Enter 'Y' in this field if you plan to specify any currency options in fields 4–10.
2 Date Information Provided - Enter 'Y' in this field if you plan to specify any date options in fields 17–20.
3 Control Report XML Option - 1 or blank - do not generate the report in XML format, 2 - generate, write to file, 3 - generate, autoloading to database.
4 User Report XML Option - 1 or blank - do not generate the report in XML format, 2 - generate, write to file, 3 -generate, autoloading to database.
5 XML DB Fail Option - 1 or blank - display warning message and continue processing the job, 2 - display warning message, write message to file, and continue, 3 - display error and stop processing.
6 XML DB Message Size - Maximum size for writing the XML to a database (in increments of 10,000 characters)
0000 or blank - use the default of 50 (500,000 characters)
0001 through 1677 - 10,000 to 16,777,000 characters)
7 Free-Form Report XML Option - 1 or blank - do not generate the report in XML format, 2 - generate, write to file, 3 -generate, autoloading to database.
8 Time Formats: 1- HH:MM 2 or blank - HH:MM:SS 3 - HH:MM:SS.NN
9 Time Notations: 1- 12-hour format (AM or PM will be printed) 2 or blank – 24-hour format

Currency Format Options

The currency format options allow you to customize the currency information appearing in your reports. Any options specified in the user options file will override the system defaults. By default, an amount will print with a dollar (\$) sign to the left of the values, commas mark the thousands position, and a period marks the decimal point. The currency format options are as follows:

CURRENCY INFORMATION PROVIDED

This is a 1-character indicator specifying whether currency information is provided. This option necessarily precedes all options governing the appearance of currency amounts. Y (yes) and N (no) responses are permitted. Leaving the field blank defaults to N. The Y option specifies that currency information is provided in the following field to override system defaults.

THOUSANDS PLACE MARKER

This is a 1-character symbol to use as the thousands place indicator. This is the character used to separate every set of 3 digits to the left of the decimal point. In the U.S. it should normally be denoted by a comma; in typical European usage, a period. Any character is allowed. The system default is a comma. This example shows a period used as the thousands place marker:

```
SF 1.000,00-
```

8 ■ Setting the User Options

Setting User Options on UNIX or Linux

DECIMAL PLACE MARKER

This is a 1-character symbol to denote the decimal place. In the U.S., a period is used; in typical European usage, a comma. Any character is allowed. The default is a period. This example shows a comma being used as the decimal place marker.

```
DM 1.000,00-
```

CURRENCY SYMBOL

This can be any string of up to 6-characters, that specifies the symbol to be printed for amount items. The system default is a dollar sign (\$). Enter NONE if you want the currency symbol to be one or more spaces. This example shows a British pound symbol being used as the currency symbol.

```
£1,000.00-
```

SECOND CURRENCY SYMBOL

Up to 6-characters that supply a second currency symbol to use. If both before and after are specified under the Currency Symbol Location option below, this second currency symbol will print to the right of the amount field. This example shows a second Swiss franc symbol being used as the second currency symbol.

```
SF -1.000,00 SF
```

CURRENCY PADDING LENGTH

This is a 1-character field specifying the number of spaces to appear between the currency symbol and the amount. Up to eight spaces may be specified. This example shows 3 spaces being used between the amount and the currency symbol.

```
1.000,00   FR
```

CURRENCY SYMBOL LOCATION

This is a 1-character code specifying whether the currency symbol is printed before, after, or both before and after the amount. Valid values are 2 (before), 4 (after), and 6 (both). If 6 is specified, the currency symbol is printed before the amount, and the second currency symbol is printed after the amount. The default location is before the amount. Examples:

```
SF 1.000,00   (Currency Symbol SF and 2)
1.000,00 SF   (Currency Symbol SF and 4)
<< 1.000,00 >> (Currency Symbols <<, >>, and 6)
```

CURRENCY SYMBOL LENGTH

This allows you to specify the length (1-6) of the currency symbol you entered in the Currency Symbol field. For example you would specify a length of 2 for SF. You would specify a length of 1 for \$. The default is 1.

Negative Sign Format Options

The negative sign format options allow you to print a negative sign other than a dash (-) on your reports. It also allows file extraction to recognize a symbol other than a dash (-) or CR.

NEGATIVE SIGN

This is a 1 to 6-character symbol used as a negative sign. Any character string is permitted. A maximum of six characters is allowed. The system default is the conventional negative sign (-). Example:

```
DM -1.0
```

SECOND NEGATIVE SIGN

This is a 1 to 6-character symbol to use as a second negative sign. If both before and after are specified under the Negative Sign Location option, this symbol is used after the amount. A maximum of six characters is allowed.

This example shows the conventional negative sign appearing both before and after the amount.

```
L -1.000,00 -
```

NEGATIVE SIGN PADDING LENGTH

This is a 1-character field specifying the number of spaces to be printed between the sign and the amount. Up to eight spaces are permitted.

In this example, 1 space has been specified.

```
- 1.000,00 SF
```

NEGATIVE SIGN LOCATION

This is a 1-character code specifying whether the negative sign is printed before, after, or both before and after the amount. Valid values are 2 (before), 4 (after), or 6 (both). If 6 is specified, the negative sign is printed before the amount, and the second negative sign is printed after the amount. The default location is after the amount. Examples:

```
-1.000,00 SF (2 specified)
1.000,00 - SF( 4 specified)
(1.000, 000) (6 specified, with left and right
              parentheses used as the negative signs)
```

NEGATIVE SIGN LENGTH

This is a 1-character field specifying the number of characters to print from the negative signs. a length of 1 to 6 characters may be specified.

8 ■ Setting the User Options

Setting User Options on UNIX or Linux

Date/Time Format Options

The date/time format options allow you to customize the date and time information appearing in your reports. This affects the headings on your reports as well as any items defined as date or time formats and not overridden at a lower level.

DATE INFORMATION PROVIDED

This is a 1-character indicator specifying whether date information is provided. This option necessarily precedes all other options governing the appearance of dates. Y (yes) and N (no) responses are allowed; leaving the field blank defaults the value to N. The Y option specifies that date information is provided in the following fields to override system defaults.

DATE OUTPUT FORMAT

This is a 3-character code specifying the order in which the year, month, and day should appear in the date. Any combination of Y, M, and D is permitted. The system default is YMD. This example denotes that the date output format is year followed by month followed by day, as in 06/10/31 for October 31, 2006.

DATE SEPARATOR

This is a 1-character symbol that will print between year, month, and day in the date output. Any character may be specified. The system default is a slash (/). For example, you could enter a hyphen if you want to use this character to separate the month, the day, and the year.

DATE ZERO SUPPRESS INDICATOR

This is a 1-character indicator specifying whether zeroes will be suppressed. Y (yes) and N (no) are permitted. If Y, then any month or year that is less than 10 will have the leading zero suppressed.

CENTURY INDICATOR

This is a 1-character indicator specifying whether the century should be printed. Y (yes) and N (no) are permitted. If Y, then the century will appear in front of the year. If N, blanks will print in the century positions.

You might want to choose Y, for example, if you do not want to see a date like 00/01/01 (yy/mm/dd) on your Control Report. By changing the century indicator to Y, you can make the date appear as 2000/01/01 (ccyy/mm/dd).

You can also use the century indicator as a way of verifying the century value of a date item. This is especially important if you are performing calculations with dates.

For example, if you pick up 79/12/31 as the date item and have the **Print century before year** option turned on, the date will print as 2079/12/31. If your date item was supposed to refer to 1979, the date count value stored in history is incorrect.

TIME SEPARATOR

This is a 1-character symbol that will print between hour, minutes, and seconds in the time output. Any character may be specified. The system default is a colon (:). For example, you could use a colon to separate the hour, minutes, and seconds; as in 10:09:45.

Print Commas for Counts

This option is a 1-character code that allows you to suppress printing of the thousands place indicator (the default character is a comma) in count fields on the Control Report, User Report, and Free-Form Report.

Valid values are **Y** (Yes) to print the indicator, **N** (No) to suppress printing of the indicator, or **Blank** to print the indicator (i.e. use the default of **Y** [Yes]).

Print Commas for Amounts

This option is a 1-character code that allows you to suppress printing of the thousands place indicator (the default character is a comma) in amount fields on the Control Report, User Report, and Free-Form Report.

Valid values are **Y** (Yes) to print the indicator, **N** (No) to suppress printing of the indicator, or **Blank** to print the indicator (i.e. use the default of **Y** [Yes]).

22-character Numeric

The 22-character Numeric option allows you to format counts and amounts in the Control Report, User Report, and Free-Form Report in 22-character format. Using 22-character format enables proper alignment of longer numbers that may include a currency symbol, commas, decimal point, sign, and up to 15 digits.

Valid values are **N** (No) to use the default 18-character formatting, **Y** (Yes) to use 22-character numeric formatting, or **Blank** to use the default of **N** (No).

Initialize Text Items to Spaces

The Initialize Text Items to Spaces option enables you to control how empty internal items, extended internal items, and history items, which are formatted as text, will be initialized.

8 ■ Setting the User Options

Setting User Options on UNIX or Linux

Valid values are **N** (No) to initialize these items to low-values, **Y** (Yes) to initialize to spaces, or **Blank** to use the default of **N** (No).

Shut off processing and set zero return code

Select if you want the following to occur: When a job starts, the return code will be set to 0000 and the processing will stop immediately without performing balancing. When you are ready to reactivate the product, deselect this option.

Override set zero return code option

Globally overrides the **No (always zero)** option if specified at the job level in the **Set Return Code** dialog box.

- **No override** - Job-level option will be used.
- **Set step return code** - Sets the step completion code equal to the highest return code identified when a balancing step is out of balance, or in error. Subsequent steps may also respond to the same return code.
- **Set abend return code** - Forces an abend if any rule is out of balance.

Override set step return code option

Globally overrides the **Yes (highest balancing Return Code)** option if specified at the job level in the **Set Return Code** dialog box.

- **No override** - Job-level option will be used.
- **Set zero return code** - Resets the step completion code to 0000 on the Control Report. The job stream continues to process and the out-of-balance condition is noted on the Control Report.
- **Set abend return code** - Forces an abend if any rule is out of balance.

Override set abend return code option

Globally overrides the **Abend (with highest balancing Return Code)** option if specified at the job level in the **Set Return Code** dialog box.

- **No override** - Job-level option will be used.
- **Set zero return code** - Resets the step completion code to 0000 on the Control Report. The job stream continues to process and the out-of-balance condition is noted on the Control Report.
- **Set step return code** - Sets the step completion code equal to the highest return code identified when a balancing step is out of balance or in error. Subsequent steps may also respond to the same return code.

XML Options

Note: The report must be created and flagged as **Print** in the Print Control Report dialog box in order for the XML to be generated.

Control Report XML Option

The Control Report XML option allows you to specify an option for generating the report in XML format.

Valid values are as follows:

- 1 or blank. Do not generate the report in XML format.
2. Generate, write to file.
3. Generate, autoload to database.

User Report XML Option

The User Report XML option allows you to specify an option for generating the report in XML format. Valid values are as follows:

- 1 or blank. Do not generate the report in XML format.
2. Generate, write to file.
3. Generate, autoload to database.

Free-Form Report XML Option

The Free-form Report XML option allows you to specify an option for generating the report in XML format. Valid values are as follows:

- 1 or blank. Do not generate the report in XML format.
2. Generate, write to file.
3. Generate, autoload to database.

XML DB Error Option

This option allows you to specify the action to take when you have specified that the XML should be written to a database and the write to database fails.

Valid values are as follows:

- 1 or blank. Display warning and continue.
2. Display warning message, write message to file and continue.
3. Display error and stop.

8 ■ Setting the User Options

Setting User Options on UNIX or Linux

XML DB Message Size

This option allows you to specify the maximum number of characters (in 10,000s) to write when you have specified that the XML should be written to a database (in increments of 10,000 characters). Valid options are as follows:

- 0000 or blank. Use the default of 50, which represents 500,000 characters.
- 0001 through 1677. Represents 10,000 to 16,770,000 characters.

Pass Alternate Control Task option

This option allows you to pass the Alternate Job ID instead of the Base Job ID to the XML file for the report message. The default value of the option is unchecked.

Bypass Processing and Return Code for No Extraction: UNIEXEO

To specify the global processing options for Bypass Processing and Set Return Code for No Extraction, you need to create a UNIEXEO record that conforms to the record layout shown below. You can create only one UNIEXEO record within the user options file.

UNIEXEO Record Layout					
Field	Description	Position	Length	Format	Value
1	Type of user options card	1-10	10	X(10)	UNIEXEO
2	Master on/off switch to shut off or turn on ACR/Detail reconciliation	11	1	X(1)	blank or 1=on (default), 0=off
3	Reserved for ACR/Summary	12-14	3	X(3)	Blanks
4 ¹	Set Return Code for No Extraction	15	1	X(1)	Y, N, Blank=N
5	Return Code	16	4	X(4)	0000-9999, blanks
6	Reserved	20-80	61	X(61)	Blanks

¹ Set Return Code for No Extraction: Y - set return code, N or blanks - do not set return code.

Using Multi-Level and Suspense Reconciliation

This chapter shows you how to run two specialized types of reconciliation. It includes the following sections:

- “Using Multi-Level Reconciliation” on page 185
- “Using Suspense Reconciliation” on page 189

Using Multi-Level Reconciliation

What is Multi-Level Reconciliation?

With multi-level reconciliation, you can create a **base-level** job to extract data at the lowest level of detail you need, using the most general level of reconciliation key. Then define additional **RLQ jobs** to reconcile the extracted data at up to 99 more detailed (subordinate) levels. For example, suppose you want to extract data at three levels. The first level might use the first 8 characters of the key, the second level might use 16 characters of the key, then the third might use all 40.

Subordinate jobs use the same job name and step name as the base level job, each with a unique **Level Qualifier** (also called the **Reconciliation-Level Qualifier** or **RLQ**), to reconcile data at a more detailed level. The job name, step name, and level qualifier for each job are specified in the Control Task (Job) Name dialog box.

For the base level job, the **Level Qualifier** field is blank. Sequential, numeric RLQs are assigned in the **Level Qualifier** field to the level qualifier jobs. Out-of-balance keys for each reconciliation level can be written to a filter file for use in a subsequent RLQ.

Other specifications for multi-level reconciliation can be set up via the **Level Information** field in the Reconciliation Level dialog box.

9 ■ Using Multi-Level and Suspense Reconciliation

Using Multi-Level Reconciliation

Example

A chain of department stores wants to reconcile data for each of its stores. If the results are out of balance, management wants to find out where the problem is. So when an out of balance situation is found at the store level, the same job definitions and rules need to be run at the department level. The reconciliation at the store level would be the base-level job and the reconciliation at the department level would be reconciliation level qualifier (RLQ) job 01.

Processing

Data is extracted from the input file using the file definitions in the base job and then the job definitions are applied. The RLQ jobs are run automatically if no return code range exists to check return codes from the prior level or if the return code value returned from the prior level falls within the return code range specified.

Reconciliation can also be run against a subset of keys stored in the filter file if this is specified (see “[Setting Up Multi-Level Reconciliation](#)” on [page 186](#)).

Reports can be produced at each reconciliation level. If the jobs store history, data can be stored on the history database at any reconciliation level.

Setting Up Multi-Level Reconciliation

The following instructions cover only the steps for creating a job that are specific to multi-level reconciliation. For a more complete procedure for creating a job, see “[Setting Up and Running a Job](#)” on [page 64](#).

1. Create the base level job and any RLQ jobs.

In the Control Task (Job) Name dialog box for each job, the base job and all RLQ jobs will have the same job name and step name.

For the base level job, the **Level Qualifier** field should be left blank. Assign sequential, numeric RLQs in the **Level Qualifier** field to the RLQ jobs.

2. Complete the Reconciliation Level dialog box for each job as follows:
 - a. Enter a Reconciliation Key Length. You can:
 - Leave the field blank (the default is 40).
 - Enter a value from 1 to 40 if you want to limit the reconciliation levels processed. For example, if your reconciliation key were:
1234567890123

and you want to use only the first portion of the reconciliation key, you would enter 8.
 - b. Select or deselect the **Create Data Filter** check box. Select if you want a filter file to be created that will store out-of-balance keys for use in the next RLQ that is run.
 - c. If this is a non-base level job:
 - Complete the **Return Code From** field, which determines when to execute the current RLQ. It specifies the beginning of the return code range values from the last RLQ which will initiate the current RLQ. For example, if you wanted the current RLQ to be executed when a rule in the base job is out of balance with a return code between 1000 and 3000, you would set the **Return Code From** field to 1000 and the **Return Code To** field to 3000.
 - Select or deselect the **Use Key Return Code** check box. If you select this box, the system will indicate the return code that must be used to determine whether the reconciliation level should be performed.
 - Select or deselect the **Use Base Job Definitions** check box. If you select this box, the system will use the job definitions (rules, messages, etc.) from the base level job in this RLQ job. Any definitions you enter for this RLQ job will override or be added to the base job definitions at run-time. Definitions for User Reports and Free-Form Reports from the base job will be used depending on how you set the following fields: 1) **Use Base and RLQ Definitions** field in the Use Base and RLQ Definitions dialog box, and 2) **Select a choice for restarting the Free-Form Report (RLQ)** field in the Print Free-Form Report dialog box.
 - Select or deselect the **Use Data Filter** check box. Selecting this box will cause the system to reconcile out-of-balance keys only when the RLQ job is run.
 - d. Complete all other job and file definitions as required.

9 ■ Using Multi-Level and Suspense Reconciliation

Using Multi-Level Reconciliation

- e. Update the definition database and run the job. See the next section, "Deleting Batch Definition Transaction Records from the Definition Database."

Deleting Batch Definition Transaction Records from the Definition Database

Updating the definition database does not remove batch definition transaction records for obsolete RLQs.

For example, assume you have the following base job:

```
JOB001  STEP10
```

In addition to the base job, you save the following RLQs to the database:

```
JOB001  STEP10  01
JOB001  STEP10  02
JOB001  STEP10  03
JOB001  STEP10  04
JOB001  STEP10  05
```

If you later delete the last 2 RLQs and then update the definition database, the transaction records for these RLQs will remain in the definition database until purged. For instructions on purging these records, see [Appendix D, “Maintaining the Definition Database” on page 319](#).

Using Suspense Reconciliation

Suspense reconciliation provides you with the ability to use the history database to track ongoing out-of-balance keys. This is useful when not all of the data that you need to reconcile is available at the same time, or if you need to hold data “in suspense” for a certain amount of time. Available data can be stored on the history database until all data is available for reconciliation, until the data to be reconciled is in balance, or until the data for a given period of time has been stored.

Aging information for data in suspense can be obtained by running the Suspense Aging Report. For an example, see [“Suspense Aging Report” on page 303](#).

Note: Cycle Numbers should not be specified for the history database. Suspense reconciliation does not support multiple cycles. When the cycle number is not defined, it will default to 00000001.

Preparing to Use Suspense Reconciliation

Before setting up a suspense reconciliation job, be sure that you are familiar with the job setup chapters in this manual:

- Chapter 3, “Setting Up a Job”
- Chapter 4, “Setting Up Basic Job Information”
- Chapter 6, “Setting Up Job Definitions”
- Chapter 7, “Setting Up Input Sources”

Job Settings for Suspense Reconciliation

When setting up a suspense reconciliation job, set the Print Report options and Store History option in the Basic Information folder in the Control Task (Job) View to No. You can then override the settings if necessary in the Actions sections of the Standard and Conditional Rule dialog boxes. This allows you to control, through rule action, which keys are written to history or to output. If No is specified for the Store History option and you do not override it in when setting up rules, no keys will be output.

Examples

There are two methods of extracting data for suspense reconciliation. You can use the history file as an input source, or you can obtain the same values using history items. The following are examples of each.

Example 1

This example extracts data from a report and from the existing history database as internal items during reconciliation.

This can be up with the following specifications:

File Organization Type: Detail History Database. This selection uses the existing history database as input during reconciliation. (In Input Source View, select File Organization to open the dialog box.)

Detail Field Accumulation Option: Accumulate & Replace. This option adds together any detail values that have the same reconciliation key, and replaces the value in history. Detail fields are specified in Input Source View. See Chapter 7, “Setting Up Input Sources” for more information on the Detail Field dialog box.

Standard or conditional rules can be used to perform rule actions after evaluating true and false conditions. But typically, conditional rules are used to make it easier to maintain the rules.

(Standard rules delete out-of-balance items, so to use standard rules you would have to write your rules backward. For example, I-001 not equal I-002 would be the rule, instead of what seems more logical: I-001 equal I-002.)

Conditional rules allow you to specify an action to be taken when True or when False. The conditional rules for Example 1 are as follows:

Rule #	Type	LHS	Operator	RHS	T/F	Rule Action
1	Inactive	I-001	EQ	I-002		
2	Conditional	Rule 1	Set Return Code		False	Add key to history database
3	Conditional	Rule 1	Set Return Code		True	Delete key from history database

- Notes:**
- Cycle numbers should not be specified for the history database. Suspense processing does not support multiple cycles. When the cycle number is not defined, it will default to 1.
 - In the Basic Information section of Control Task (Job) View, set the Print Report options and Store History Option to No, and then override the settings as applicable on the when you set up your rules. This allows you to control, through Rule Action, which keys are written to history, or to output. If No is specified for the Store History option and you do not override it when setting up the rules, no keys will be output. You want the unmatched suspense items to be written, so be sure to have an override on these items.

Example 2

This example extracts the report data as input, but items from history are not extracted and are processed as history items rather than internal items.

This can be set up with the following specifications:

The file organization type: The file organization type Detail History Database is not needed because items in history will be processed as history items.

Detail Field Accumulation Option: Accumulate & Replace (See Example 1.)

9 ■ Using Multi-Level and Suspense Reconciliation

Using Suspense Reconciliation

Conditional reconciliation rules: These are used to perform rule action information after evaluating true and false conditions. The conditional rules for Example 2 are:

Rule #	Type	LHS	Operator	RHS	T/F	Rule Action
1	Inactive	I-001*	EQ	E-001*		
2	Conditional	Rule 1	Set Return Code		False	Add key to history database
3	Conditional	Rule 1	Set Return Code		True	Delete key from history database

* I-001 represents a data value from the report file. E-001 represents a data value from the history file.

Using Translation Tables

This chapter provides instructions for external, internal, dynamic translation tables, and hash translation in the following sections:

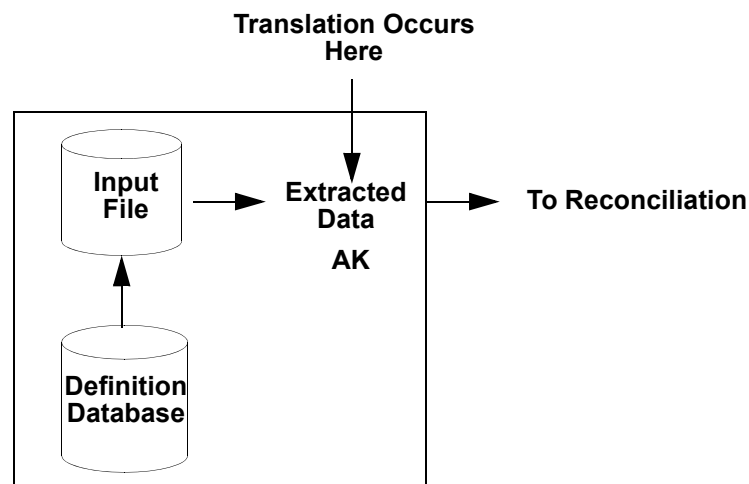
- “Internal Vs. External Translation Tables” on page 193
- “Processing of Internal and External Translation Tables” on page 195
- “Using External Translation Tables” on page 196
- “External Translation Table Trace Report” on page 208
- “Using Internal Translation Tables” on page 215
- “Using Dynamic Translation Tables” on page 218
- “Using Hash Translation for a Key Value” on page 219

Note: For information on cycle tables, see [Cycle Tables](#) on page 227.

Internal Vs. External Translation Tables

External and internal translation tables share the following characteristics:

- They can both be used to translate detail fields, key fields, or extraction variables for reconciliation purposes.
- With both external and internal translation tables, translation occurs after data to be translated is extracted from the input source. In the following graphic, a state abbreviation, AK, is extracted from a report file.



10 ■ Using Translation Tables

Internal Vs. External Translation Tables

Characteristics that are not shared by the two types of tables are described in the following sections.

What is an Internal Translation Table?

An internal translation table is made up of two columns (input and output). You create it by defining up to 999 input/output entries. This type of table is suitable for performing simple one-to-one, text-to-text translations within an ACR/Detail job.

For example, suppose you are reconciling values from two reports, and you decide to use a region number (01, 02, etc.) in your reconciliation.

However, the region number does not appear in one of the reports. Only the region name (Dallas, New York, etc.) appears. To address this situation, you could set up an internal translation table with the region names that need to be translated as input and the corresponding region numbers as output.

For more complex translations, use an external translation table, which is described in the next section.

What is an External Translation Table?

An external translation table is a more powerful table that can be used when an internal translation table will not suffice.

An external translation table is suitable for translation of dynamic data such as an employee list. This is because it is built at run time, usually by referencing a data source external to ACR/Detail. This data source can be a VSAM file, a physical sequential file, Web page, or another type of file accessible via a user access program. For more information, see the *ACR/Summary and ACR/Detail User Program Guide*.

Note: When there is no external data source, you can use constant translation data. See [Types of External Data Sources Supported on page 199](#).

Once the path and file name of the external data source have been defined in ACR/Detail, the column values in the external source can change without the need to modify the translation table definitions.

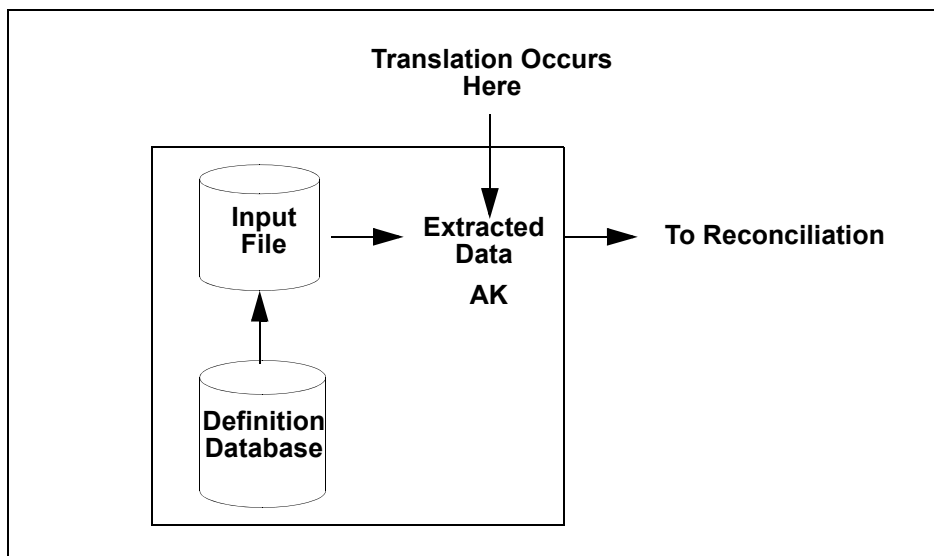
Following are some additional features of an external translation table:

- Data type translations from any format to any format are supported.
- Up to 999 columns can be used.
- Complex selection/reformatting/comparison rules can be used.

- A single external translation table can be used in multiple reconciliation jobs, even if they are not in the same entity.

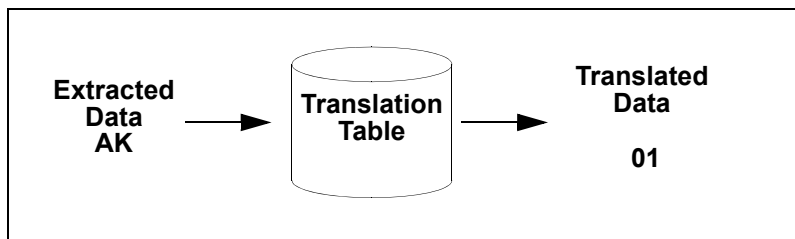
Processing of Internal and External Translation Tables

Regardless of the type of translation table you use, translation occurs after data to be translated is extracted from the input source. In the following graphic, a state abbreviation, AK, is extracted from a report file.



In the translation process, extracted data is compared to values in the translation table. If a match is found, the extracted value is translated. Then, the translated value either replaces the original value, or (for external tables only) is stored as an extraction variable. The translated value can then be used in the reconciliation process.

In the following graphic, the extracted state abbreviation AK is translated into the state code 01.



Using External Translation Tables

Creating and Generating External Translation Tables

To create an external translation table and a job that will use that table, use the following sequence of steps:

1. Create the translation table before you create the job.
2. In Control Entity View, in the Tables folder, select **New > External Translation Table** to display the External Translation Table View.
3. Complete the Basic Information folder:
 - a. Give the table a meaningful name and description. The description will be @ONEROW if you are using the VSAM ONEROW feature (See “Using VSAM ONEROW Processing” on page 202).
 - b. Identify the **Dataset Organization** of the external data source. The fields will change based on the source type you select. See “Types of External Data Sources Supported” on page 199.

If your file organization type is physical sequential, you can extract directly from a delimited data file. See “Extracting Directly from a Delimited Data File” on page 162.
 - c. Unless your dataset organization is Constant Translation Data, enter the path and file name for the external data source.
 - d. If appropriate, change the Build Table Data Option:
 - **On first lookup only**—The table will be built one time for each job run. This is the default.
 - **On each lookup**—The table will be rebuilt each time it is called. This is only appropriate when the building of the table depends on the input-output parameter values. Use this option when using the ONEROW feature referenced above.
 - e. Save the table.
4. Use the Input-Output Parameters section to specify each input-output parameter. When defining a parameter, you assign a name for the value to be translated, specify the format for the translated value on reports, specify where the value will be extracted from (and passed back to). Make special note of the following:
Use the **Name** field to name the parameter. This name will be used in assignment lookup rules or in the translation process.
Use the **Format** field to specify the format for the translated values.

The **Indicate what the parameter refers to** field enables you to do either or both of the following:

- Pass multiple values to the external table to be translated at once. For example, you could pass the department number and store number to get the department name and store name.
 - Combine multiple values and send them to the external table to determine the correct translated value. For example, you might need to pass both the department number and store number to the table to get the name of the department manager.
5. A column is a field on the translation table that will be used for the translation. Use the Table Columns folder to define each table column.
 6. Use the Table Lookup Rules folder to define processing that will take place when a value to be translated is encountered.

Selection lookup rules—Identify the values to be evaluated and state the comparison operator (equal to, not equal to, etc.)

Assignment lookup rules—Specify the value to assign when the Selection Rule is satisfied. For example, when an input-output parameter is equal to a particular column from the table, assign it the value of another column from the table.

Flow lookup rules—Specify the processing that should take place after a Selection Rule and an Assignment Rule are processed. By default, processing will terminate.

7. If your external data source is a VSAM Key Sequenced file, the Build Preparation Rules folder will display. A build preparation rule is used specify the source to be used in building the VSAM key.
8. Save the table.
9. Define the Table Build Rules. These specify the criteria for extracting data (usually from the external source file) and loading it to build the translation table. You can add a new rule in any of the following ways:
 - By using the toolbar icon for the rule type.
 - By selecting the rule type from the Define menu.
 - By clicking **New** under the Table Build Rules folder and selecting a rule type.

The build rule types, each of which has an associated dialog box, are as follows:

***Selection Criteria field**—Defines criteria for selecting records from the external data source.

10 ■ Using Translation Tables

Using External Translation Tables

Relative Record field—Identifies the relative position from a selected record. Use this rule type in conjunction with a selection criteria field.

***Reformat Record field**—Reformats a column value before using it in a column assignment.

***Column Assignment field**—Extracts column values from the input area (the external data source) from a literal, or from an extraction variable) and loads them into the translation table. This field is comparable to a field item or detail field in a reconciliation job.

Process Control field—Determines processing to be performed after a selection group has been satisfied.

Note: For the types marked with an asterisk (*), if the data source is an ASCII text file, a COBOL record layout, or a Web page that appears in the Extract Window, you can use the mark-and-capture method (select the data with the mouse) to easily create a field definition (in the dialog box, the position and length of the field will be entered automatically).

If the input source is lengthy, you may want to use the **Find** and/or **Repeat Find** icons to find the data you want. These icons are available when the cursor is in the Extract Window at the bottom of the view.

For displayed Web pages, in most cases you can select text or a column heading above the data you want and then use a relative record to locate the specific data to extract. To assist you in locating data, a unique, sequential, eight-digit identifier will be added before each text line. You can optionally use these identifiers to select the data you need.

When mark-and-capture is not available, you will need to enter the definitions in the dialog box manually.

10. If you selected Constant Translation Data in the **Dataset Organization** field in [step b on page 196](#), the Constant Table Data folder will display at the bottom of the view. You must use the Constant Table Data dialog box to enter each data record from which the table columns will be built.

Define as many columns as you want, as long as the total width is no more than 88 characters for an input record, 8 characters for an alphanumeric output column, or 15 characters for a numeric output column.

For example, suppose you want to create a table that converts cost center names (Dallas, Atlanta, etc.) into cost center numbers (100, 170,

etc.). You would type **Dallas 100** in the Data field to create the first data record. Dallas represents column 1 and 100 represents column 2. You would leave spaces between the cost center name and the cost center number based on the number of characters in the longest cost center name.

11. Save the table and update the definition database.
12. Create the job that will call the external translation table. When you create your extraction definitions, in the Key Field or Detail Field dialog box for the field to be translated, select **Use Translation** or **Use Translation Table**, select **Translate** and complete the Selection Translation Information dialog box by selecting the name of the table and the action to be taken if the value to be translated is not found in the table.
13. Save your work and update the definition database.
14. When you are ready to run the job, select the **Reconcile** icon. If you want to request an External Translation Table Trace Report, see [“External Translation Table Trace Report” on page 208](#).

Types of External Data Sources Supported

Physical Sequential

On Windows this is an ASCII text file.

VSAM Key Sequenced

Specifies that the source of the table is a VSAM file. You can optionally use VSAM ONEROW processing. See [“Using VSAM ONEROW Processing” on page 202](#).

User Program Accessed File

Specifies that a user program will be used to access external data that could not otherwise be accessed. You can use your own program or use one of the following user programs provided by Infogix (for more information, see the *ACR/Summary and ACR/Detail User Program Guide*):

UUPDLIM. Delimited Field Access Method can be used to extract data from files containing records of variable length fields, separated by one or more separator characters or delimiters.

UUPXML. XML File Access User Program can be used to extract data from XML files.

10 ■ Using Translation Tables

Using External Translation Tables

UUPEDIF. The EDI Access Method can be used to extract data from EDI files (in X12 format).

UUPVREC. The Variable Record Access Method can be used to extract data from variable length records consisting of a fixed “header” area and a variable number of “line” areas.

UUPFBIO. The Fixed Block Input/Output Access Method can be used to process and convert EBCDIC input sources with a fixed record length.

UUPVBIO. EBCDIC Variable Block Input Source Reader Program can be used to make mainframe EBCDIC variable block input files accessible on UNIX, Linux, and Windows.

UUPPDF. The PDF Extraction User Program can be used to extract data from PDF files. Within the same job, a PDF file can be used as an input source to the job, or as an input source to an external translation table associated with the job, but not both.

Constant Translation Data

If you use this **Dataset Organization** type, you will be defining the data for the translation table in the table definitions, rather than referencing an external file at run time (this is why the **File Name** field does not display when this dataset organization type is selected). Your table will be like an internal translation table. However, you will have the sophisticated functionality of an external translation table.

Web Page

After completing the fields, click **Retrieve** to establish an Internet connection, copy the Web page from the specified **URL Address**, convert this copy to ASCII text, and store it in the specified **Local Copy** path and file name.

External Translation Table Example

Before you start to build an external translation table, it may be helpful to have a simple example. The table will translate the 2-character abbreviation into a two-position numeric code so that the data on two input reports can be reconciled.

The Reports

Report 1 shows total quarterly sales figures by state, using the 2-position numeric code. Report 2 shows monthly sales figures by state, using a 2-character postal abbreviation.

Using the code as the key, we need to reconcile the quarterly figures on Report 1 to the monthly figures on Report 2.

XYZ COMPANY		QUARTERLY SALES FIGURES	
		Q3, 1994	
STATE		TOTAL	
01		175,000.00	
02		225,000.00	
03		120,000.00	
.	.		
.	.		

XYZ COMPANY		MONTHLY SALES FIGURES	
		Q3, 1994	
STATE		TOTAL	
AK		50,000.00	
AK		25,000.00	
AK		100,000.00	
AL		100,000.00	
AL		100,000.00	
AL		25,000.00.	
.	.		
.	.		

The Source Data

The source data contains the data needed to do the translation. The system will read this source data and build a translation table based on the definitions.

In this example, the source data is as follows:

01000001AK
02000002AL
03000003AZ
04000004CA
05000005CO
06000006DE
07000007GA

For our translation we will use the first two characters and the last two characters. The middle characters will not be used in the translation process.

10 ■ Using Translation Tables

Using External Translation Tables

The External Translation Table

The table to be built will consist of two columns: one for the numeric state code and one for the alphanumeric state abbreviation. During extraction, if the system comes across a key or detail value to be translated, it looks for the name of the translation table related to the value and then looks for a match in that table. If it finds a match, it performs the translation and returns the result to the extraction process.

Code	Abbreviation
01	AK
02	AL
03	AZ
04	CA
05	CO
06	DE

Using VSAM ONEROW Processing

If your source data is a keyed VSAM file, you can access the file randomly using a feature called ONEROW. This feature enables you to rebuild the table every time you translate an item using only one member.

The built table would look as follows if you processed the Alaska record from the "External Translation Table Example" above:

Code	Abbreviation
01	AK

Then, when you translated the Alabama record, the built table would look like this:

Code	Abbreviation
02	AL

The ONEROW feature can save processing time if the following are true:

- You have the key to the VSAM file.
- You expect all items to be found on the file.
- You will not process the same translation more than once.

Implementing ONEROW Processing

To implement ONEROW processing, enter @ONEROW in the Table Description dialog box and specify to build the table with each lookup.

Using Extraction Variables in External Translation Tables

The following table summarizes the possible uses of extraction variables in external translation table definitions:

Definition	Source	Target
Input-Output Parameter	Extraction Variable	Extraction Variable
Selection Record	Extraction Variable (LHS)	Extraction Variable (RHS)
Reformat Record	Extraction Variable	Extraction Variable
Column Assignment Record	Extraction Variable	Extraction Variable

When you build an external translation table, the source of your input-output parameters can be extracted data (a value that will be either a detail field or a key field) or an extraction variable (a temporary storage location for your extracted data). This feature allows you to use multiple values in determining your translated value or values. That is, it gives you n:n capabilities (n in, n out), instead of 1:1 (1 in, 1 out). In the example below, the source of the input-output parameter is extracted data at the detail level.

10 ■ Using Translation Tables

Using External Translation Tables

Example

Suppose you want to extract the posted credits of a financial corporation and sort the data by cost center number and branch number. Your input source has the values we are looking for, as well as the branch number. However, it does not use cost center numbers; it uses bank numbers as shown below.

TYPE	NUMBER	AMOUNT
PREVIOUS BALANCE	2	1,999,937.67
+CREDITS POSTED	21	5,090,173.68

In a master file you already have, however, you have the cost center numbers along with the corresponding bank numbers. This file will be the external tabular data source.

BNK#	BRCH#	COST CNT
003	00003	102
004	00004	103
005	00005	104
010	00006	105
082	00007	106
083	00008	107
085	00009	108
091	00001	109
092	00001	110
099	00001	111

In the job, you would define a detail field to extract the bank number and specify the translation table to be used to translate that value into the cost center number. The **Target Area** for the detail field will be an extraction variable (V-001). Later, you would use V-001 as Key 2, the cost center number, to produce the User Report.

You would create the job and translation table as follows:

1. Open a new job and complete the Basic Job Information.
2. In the Input Source View, specify the Basic Information and save your work.

3. From the Control Entity View, in the Tables folder select **New > External Translation Table** and create the table:
 - a. Enter the name of the table, VAR001, in the **Table Name** field.
 - b. Set the **Dataset Organization** to Physical Sequential and enter the dataset information. In the **File Name** file, enter the path and name of the master file containing the external tabular data. In this example, the file is physical sequential, so it will display in the Extract Window at the bottom of the view.
 - c. On the Build Table Data Option dialog box, specify On First lookup only because you only need to build the table once. If the values translated were each dependent upon the one that came before it, you would have chosen On Each lookup so that the system would rebuild the table each time it searched for another match.
 - d. Complete the Input-Output Parameter dialog box. The I/O parameter tells the system the source of the data and under what circumstances to pull data from a file and translate it. The **Indicate what the parameter refers to** field will be set to Extracted Data. The value on the way into translation is the bank number. On the way out it will be the cost center number. The cost center number will then be stored in an extraction variable and used as specified in your file definitions.
 - e. Define the table columns. The names will be as follows:
Column 1 - Bank Number
Column 2 - Cost Center Number
 - f. In the Table Build Rules folder, select the bank number from the Extract Window (if the data is displayed there), and click the **Column Assignment** icon. Assign position 1 for a length of 3 to column BANK NUMBER. The input source is an input record. The field type is text.
 - g. Select the cost center number from the Extract Window (if the data is displayed there), click the **Column Assignment** icon again, and assign position 31 for a length of 3 to column COST CENTER NUMBER. The input source is an input record. The field type is text.

When you run reconciliation and a field to be translated is extracted, the VAR01 table will be referenced. The table is built to have the number of columns with the specified lengths as defined in the table build rules.

10 ■ Using Translation Tables

Using External Translation Tables

- h. Use the Table Lookup Rules folder to create the following rules to select the row if there is a match on a bank number and assign the cost center to I/O PARM 1. This value will then be used by the file definitions.
 - Define the selection rule: Select: EXTRACTION VARIABLE 1 = BANK NUMBER
 - Define the assignment rule: Assign: EXTRACTION VARIABLE 1 = COST CENTER NUMBER

The I/O parameter stores the translated value in an extraction variable.

Key	Translated Value	Key
000	00001	100
001	00002	101
003	00003	102
004	00004	103
005	00005	104
010	00006	105

4. Return to the Control Task (Job) View and open the input source.
5. Create the extraction definitions. The input source and completed definitions will be shown below.

099 is the 3-digit bank number. You will pick it up and translate it to a cost center: V-001. V-001 is Key 2.

00001 is the branch number (Key 1).

	NUMBER	AMOUNT
PREVIOUS BALANCE	2	1,999,937.67
+CREDITS POSTED	21	5,090,173.68

Extraction Definitions			
	Select if Position 0004 for a Length of 01 is equal to -		
	Get value from position 0033 for a length of 03, store in extraction variable 001		
	Select if Position 0031 for a Length of 05 is numeric		
	Retrieve key 1 (BRANCH NUMBER) in position 0031 for a length of 05		
	Pick up key 2 (COST CENTER NUMBER) from extraction variable 001		
	Select if Position 0006 for a Length of 08 is equal to +CREDITS		
	Get value from position 0047 for a length of 12, store in internal item AMOUNT		

01-08-1996	099 XYZ BANK MINNESOTA FSB		
	00001 XYZ MINNNEAPOLIS		01-017
DEMAND DEPOSIT ACCOUNTING		ACTIVITY RECAP ONE	
TYPE	001 TOTALS CHECKING ACCOUNT		
***	*-----*		
	NUMBER		AMOUNT

Ln 1, Col -64199

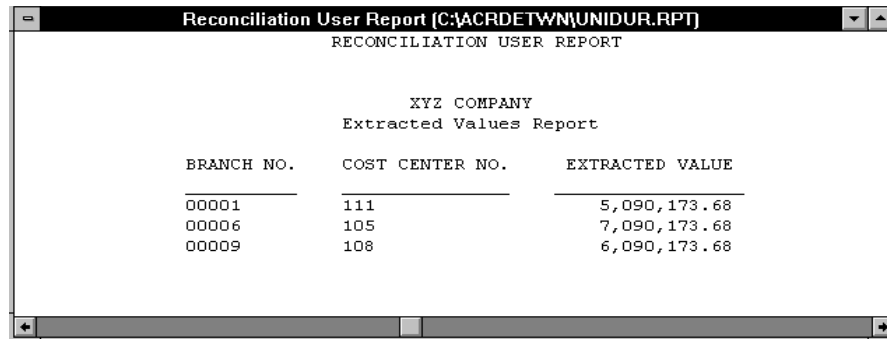
- a. Define a selection field. Select the dash in position 0004 for a length of 1.
- b. Define a detail field to extract the bank number. Select **Use Translation Table**, click **Translate**, and enter the table name: VAR001.

Translation can take place at the key level or detail level. In this job, translation takes place at the detail level. The translation table will not be built until the system encounters the detail field value that we have indicated.

6. Create two keys and one internal item as follows:
 - a. Define another selection field to select if position 0031 for a Length of 05 is numeric.
 - b. Define a key field to retrieve key 1 (BRANCH NUMBER) in position 0031 for a length of 05.
 - c. Define key 2 to pick up the translated extraction variable (COST CENTER NUMBER).
 - d. Define the next selection field. Select the word +CREDITS because it is unique to this record (report line).
 - e. Define a detail field to extract 5,090,173.68 on that line in the Amount column and store it in I-001.
7. Save your work, update the database, and run reconciliation.
8. The User Report will show the posted credits of the corporation, sorted by branch number and cost center number.

10 ■ Using Translation Tables

External Translation Table Trace Report



BRANCH NO.	COST CENTER NO.	EXTRACTED VALUE
00001	111	5,090,173.68
00006	105	7,090,173.68
00009	108	6,090,173.68

Report

You can generate a report on the external translation table processing, as described in the following section.

External Translation Table Trace Report

This report provides the following information for a requested input file ID:

- A formatted list of the table build rule definitions as they were loaded in preparation for building the external translation table.
- For each input record processed, the following information is provided:
 - The record count plus the first 132 characters of the input record
 - The processing of each table build rules against the record to build the external table. This includes the following:
 - The table build rule definition.
 - LH Value: The value as extracted from the input source.
 - RH Value: The value from the external table column to which the LH Value is being compared.
 - Action Taken: Whether the comparison passed or failed.
 - For rules that passed the comparison, the value assigned to replace the original value from the input source.
 - Details about each translation attempt.

The information below is divided into the following sections:

- “Generating the Report” on page 209
- “External Definition List Section” on page 210
- “Table Build Rule Processing Per Record Section” on page 211

■ “Formatting of Table Build Rules” on page 212

Note: You can generate another type of Trace Report that shows the processing of the file definitions against each record in the input source. For information on that report, see [Trace Report on page 280](#).

Generating the Report

This report will be generated when you run reconciliation if you request it as follows:

1. In the Run Reconciliation dialog box, under Input Sources, highlight the line that starts with the name of the table for which you want the report, and click **Edit**.
2. When the File ID and File Name dialog box appears, the check box for **External Translation Table** field at the bottom will automatically be selected, although it will be grayed out.
3. Select **Produce Trace Report**.
4. Select **OK** to save your settings and return to the Run Reconciliation dialog box.
5. Run reconciliation and generate the report.

10 ■ Using Translation Tables

External Translation Table Trace Report

External Definition List Section

The first report section, a sample of which is shown below, is a formatted list of the table build rule definitions.

Note: In the following report excerpts, extended values are printed as follows:

- Extended LH values (from the input source) that will not fit on a line are indicated with SEE VALUE BELOW.
- Extended RH values (from the external table column to which the LH Value is being compared) that will not fit on a line are indicated with SEE LITERAL.

The extended values are then printed on separate lines.

```
EXTERNAL DEFINITION LIST: LONG NUMBER
```

```
-----  
SELECT(0010/12/NUM/ NA ) INPUT (EQUAL ) LIT (000111222333444 )  
                                     (000222333444555 )  
                                     (000333444555666 )  
SELECT(0010/31/NUM/ NA ) INPUT (EQUAL ) LIT *SEE VALUE BELOW*  
  SELECT LITERAL: (111222333444555666777888999000 )  
  SELECT LITERAL: (222333444555666777888999000111 )  
  SELECT LITERAL: (333444555666777888999000111222 )  
ASGN(0010/31/CNT/ NA ) INPUT (001/NUM/XVAR)  
ASGN(0010/31/CNT/ NA ) INPUT (002/NUM/XVAR)  
SELECT(0010/12/NUM/ NA ) INPUT (GT ) LIT (000111222333444 )  
SELECT(0010/31/NUM/ NA ) INPUT (GT ) LIT *SEE VALUE BELOW*  
  SELECT LITERAL: (111222333444555666777888999000 )  
SELECT(0001/16/PAC/0001) XVAR (EQUAL ) XVAR(0002)  
SELECT(0001/16/PAC/0001) XVAR (GT ) LIT **SEE VALUE BELOW*  
  SELECT LITERAL: (111222333444555666777888999000 )  
ASGN(0010/06/CNT/ NA ) INPUT (001/NUM/COL )  
ASGN(0010/31/CNT/ NA ) INPUT (002/NUM/COL )
```


10 ■ Using Translation Tables

External Translation Table Trace Report

Formatting of Table Build Rules

Delimited Data Extraction Formats

Selection Criteria Field Format: Delimited Data Field Extraction

Format

```
SELECT(AAAA/BB/CCC/DDFN) I/DDF (FFFFFF)  
SELECT(AAAA/BB/CCC/DDFN) I/DDF (FFFFFF) LIT GGGGGGGGGGGGGGGG
```

Where:

AAAA/BB/CCC are the input field position/length.

DDFN is the delimited data field number.

I/DDF indicates that the input is from a delimited data field.

FFFFFF is the selection operator.

LIT indicates comparison to a literal.

GGGGGGGGGGGGGGGG is the literal. If non-extended, this can be up to 16 bytes long. If extended, this can be a number of up to 16 digits or a text value of up to 80 characters.

Column Assignment Field Format: Delimited Data Field Extraction

Format

```
ASGN(AAAA/BB/CCC/DDFN) I/DDF (FFFF/GG/HHH/DDFN) O/DDF
```

Where:

AAAA/BB/CCC/DDFN are the input field position/length/format. DDFN is the input delimited data field number.

I/DDF indicates that the input is from a delimited data field.

FFFF/GG/HHH/DDFN are the output field position/length/format. DDFN is the output delimited data field number.

O/DDF indicates that the output is assigned to a delimited data field.

Reformat Field Format: Delimited Data Field Extraction

Format

```
RFMT(AAAA/BB/CCC/DDFN) I/DDF (FFFF/GG/HHH/DDFN) O/DDF
```

Where:

AAAA/BB/CCC/DDFN are the input field position/length/format. DDFN is the input delimited data field number.

I/DDF indicates that the input is from a delimited data field.

FFFF/GG/HHH/DDFN are the output field position/length/format. DDFN is the output delimited data field number.

O/DDF indicates that the reformatted output is assigned to a delimited data field.

Non-delimited Extraction Formats

Selection Criteria Field Formats

Source Value is Compared to	Comparison Value is	Format
Input Record or Extraction Variable	Literal (Non-extended)	SELECT (AAAA/BB/CCC/NNNN) EEEEE (FFFFFF) LIT (GGGGGGGGGGGGGGGG)
Input Record or Extraction Variable	Literal (Extended)	SELECT (AAAA/BB/CCC/NNNN) EEEEE (FFFFFF) LIT *SEE VALUE BELOW* SELECT LITERAL: (GGGGGGGGGGGGGGGG)
Input Record	Extraction variable	(Extraction variable is non-extended) SELECT (AAAA/BB/CCC/NNNN) EEEEE (FFFFFF) EVAR(NNNN) (Extraction variable is extended) SELECT (AAAA/BB/CCC/NNNN) EEEEE (FFFFFF) XVAR(NNNN)

Where:

AAAA/BB/CCC/NNNN are the source field position/length/format/extraction variable number. NNNN will be NA when the source is not an extraction variable.

EEEEEE is the source of input, which can be one of the following: INPUT indicates the source is an input record. EVAR and XVAR indicate the source is a regular or extended extraction variable, respectively.

FFFFFF is the selection operator.

LIT indicates comparison to a literal.

EVAR or XVAR on the right side indicate comparison to a regular or extended extraction variable.

GGGGGGGGGGGGGGGGGGGG indicates that the comparison is against a literal. If non-extended, this can be up to 16 bytes long. If extended, *SEE VALUE BELOW* is printed and the literal, which can be a number of up to 30 digits or a text value of up to 80 characters, is printed on a separate line.

NNNN on the right side indicates that the comparison is against an extraction variable and provides the extraction variable number.

Reformat Record Format

Format

RFMT (AAAA/BB/CCC/DDDD) EEEEE (FFFF/GG/HHH) (JJJJ)

Where:

AAAA/BB/CCC/DDDD are the source field position/field length/field format.

DDDD is the source extraction variable number.

EEEEEE is the source of input, which can be one of the following: INPUT indicates the source is an input record. EVAR or XVAR indicate the source is a regular or extended extraction variable, respectively.

FFFF/GG/HHH are the target field position/field length/field format.

JJJJ is the target where the reformatted value is placed. If the target is an extraction variable, this will be the extraction variable number. If the target is the input field or a literal, NA will print.

10 ■ Using Translation Tables

External Translation Table Trace Report

Column Assignment Field Format

Format

ASGN(AAAA/BB/CCC/DDDD) EEEEE (FFF/GGG/HHHH) LIT JJJJJJJJJJJJJJ

Where:

AAAA/BB/CCC/NNNN are the source field position/length/format/extraction variable number. NNNN will be NA when the source is not an extraction variable.

EEEE is the input source type.

FFF is a column or extraction variable number.

GGG is the format.

HHHH is the literal EVAL or COL

JJJJJJJJJJJJJJ is the literal value to assign. If numeric, it can be up to 16 bytes. If alphanumeric, it can be up to 80 bytes.

Process Control Field Format

Format

BLDTBL (AAAA) STOP/WRITE

Where AAAA is the process control flow, which will be either STOP ("at end" processing) or WRITE (write a control key break).

Table Lookup Rule Processing for Each Record

This section shows the processing of translation attempts. Each attempt shows the processing of the table lookup rules to a value for which translation has been specified.

Following is an example of two translation attempts, ET000001 and ET000002:

```
ET000001 LOOKUP SEL001 (101 EQ 101) PASSED
ET000001 LOOKUP SEL002 (001 EQ 001) PASSED
ET000001 ASGN COLM(003) VAL(John Smith) TO XVAR 003
ET000001 ASGN COLM(004) VAL(000000000006955+) TO EVAR 004
ET000002 LOOKUP SEL001 (102 EQ 101) FAILED
ET000002 LOOKUP SEL001 (102 EQ 102) PASSED
ET000002 LOOKUP SEL002 (001 EQ 001) PASSED
ET000002 ASGN COLM(003) VAL(Rebecca Taft) TO XVAR 003
ET000002 ASGN COLM(004) VAL(000000000007930+) TO EVAR 004
```

Selection lookup rules compare values to be translated that are passed from input-output parameters to the values from external table columns or from other extraction variables using the specified selection type.

If the value passes the selection lookup rules, the associated assignment lookup rules are processed. Assignment lookup rules assign a value from a specified translation table column, extraction variable, or literal to the input-output parameter. The parameter then passes the translated value to the reconciliation process.

In attempt 1 (ET000001), the value to be translated was processed by the first 2 selection lookup rules and in both cases it passed. As a result, 2 assignments were made:

- John Smith was moved to extended extraction variable (XVAR) 003.
- 6955 was moved to extraction variable (EVAR) 004.

In attempt 2 (ET000002), the next value to be translated was processed by the first selection lookup rule and failed. The value is then processed against the next record from the external translation table.

The value passed the next 2 selection lookup rules, so the following assignments were made:

- Rebecca Taft is moved to extended extraction variable (XVAR) 003.
- 7930 was moved to extraction variable (EVAR) 004.

Using Internal Translation Tables

Set up an internal translation table as follows:

1. From the Control Entity View, in the **Tables** folder select **New > Internal Translation Table**. This displays the Internal Translation Table View.
2. Use the Translation Name dialog box to define the data format of the input value (From Column) and the output translated (To Column) value.
 - For example: If you wish to convert an abbreviation of State Name to its actual name, select Field Format "Text" for both From and To Columns.
 - If you wish to convert a Date value in DDMMYY format to a Date value in MMDDCCYY format, select Field Format and Date Format as Date and DDMMYY for From Column, and select Field and Date Format as Date and MMDDCCYY for To Column.
 - The Field Format, Date Format, Time Format specifications for the "To Column" should match the field type of the destination field. If the formats do not match, the program displays error and stops further processing. You need to resolve those mismatches before resubmitting your job. In the Table Name dialog box, assign a meaningful name to the table.
3. Use the Translation Table Entry dialog box to define each table entry, consisting of an input value and an output (translated) value.

10 ■ Using Translation Tables

Using Internal Translation Tables

- The Input Value is the value that will be extracted using a key field or detail field and passed to the table for translation.
 - The Output Value is the translated value for the input value. This value will be used in the key field or detail field definition.
4. Save the table and update the definition database.
 5. Incorporate the translation table into your reconciliation job as follows: On the Key Field or Detail Field dialog box, select **Use Translation** or **Use Translation Table**, click **Translate**, and on the Selection Translation Information dialog box, enter the table name and the action to be taken if the value to be translated is not found in the table.

To make Internal Translation table processing more efficient, instead of reloading an Internal Translation table multiple times, the whole table is loaded into memory when the table is referenced for the first time.

Example of an Internal Translation Table

Suppose you are reconciling values from two reports containing data submitted by many regions and you decide to use a region number (01, 02, etc.) as the field value. However, the region number does not appear on one of the reports—only the region name (Dallas, New York, etc.).

To address this situation, you would set up and name an internal translation table. It should contain a list of region names as table input, and region numbers as table output, as follows:

Table Input	Table Output
Dallas	01
New York	02
Chicago	03

You would use the Translation Table Entry dialog box to enter the values for each row. Save your work.

Next, you would create the job and, in the extraction definitions, define a detail field to extract the region name. On the Detail Field dialog box, select **Use Translation Table**, click **Translate**, and, on the Selection Translation Information dialog box, enter the table name and the action to be taken if the value to be translated is not found in the table. Save your work and update the definition database.

When the reconciliation function is executed, the region name will be extracted from the report. The translation table will be searched for a corresponding input entry. If an input entry is found, the table output is moved to the field value. If not, the action you specified on the Selection Translation Information dialog box will be taken.

Using Dynamic Translation Tables

The reconciliation function allows you to store a maximum of 5 key segments (defined using key fields). Each key segment is composed of a maximum of 8 characters. As a result, the reconciliation key length is limited to 40 bytes:

$$5 \times 8 = 40$$

A dynamic translation table (also called a dynamic translation database) can be used if you need to exceed this 40-byte limit. Using such a table, you can have a reconciliation key up to 400 bytes.

If you use a dynamic translation database, you can specify as many as 5 key segments of up to 80 bytes:

$$5 \times 80 = 400$$

When you run reconciliation, the long key segments to be translated will be written to the dynamic translation database and translated to the corresponding 8-byte identifier, which will be used to write to the history database.

To see the key before it was translated (Original Source Value) and after translation (Identifier), you can view the Control Report or run the List Translation Database utility (**Run > Database Utilities > List Translation Database**). For more information on the utility, see [“List \(Dynamic\) Translation Database” on page 239](#).

To reorganize a dynamic translation database, you can run the List Translation Database utility (**Run > Database Utilities > Reorganize Translation Database**). For more information on the utility, see [“Reorganize \(Dynamic\) Translation Database” on page 240](#)

Initializing and Referencing Dynamic Translation Tables

1. Run the Initialize Translation Database utility to establish your dynamic translation database (**Run > Database Utilities > Initialize Translation Database**).

Important: Because initialization erases any information that is currently in a database, be careful to specify the correct database.

2. Select **Options > Configurations** from the menu. Select the configuration you will use and click **Edit**. On the Add/Edit Configuration dialog box - File Information tab, enter the path and file name of the database that you initialized in the **Dynamic Translation** field.
3. When you create the job, when specifying key fields longer than 8 bytes, do the following:
 - Select the **Use Translation** check box.
 - Click the **Translate** button to display the Selection Translation Information dialog box.
 - At the bottom of the dialog box, select **Instead of using translation table, assign/retrieve a unique number from the dynamic table for the key**.
 - Complete the **Dynamic Table Name** field.

Using Hash Translation for a Key Value

Key fields for a reconciliation job can consist of up to 5 key segments, each of which can contain up to 8 positions. This results in a maximum length for a key field of 40 bytes. When a key field value exceeds the maximum 40 bytes allowed, hash translation for a key value can be used. Instead of storing the original key value in the History database, the hash value for the key is stored.

The hashing routine returns a hash value in a 4 byte binary format. Bytes 1-4 store the first 4 bytes of the original key and bytes 5-8 store the hash value returned from the hashing routine. For reporting purposes, the original key can be saved as an extended internal item.

It is important to note that spaces are a valid value for hashing. For example, a value of CHICAGO with three trailing spaces does not equal a value of CHICAGO with no spaces.

When browsing a history key with a hash value online, specify a range of keys that includes the first 4 bytes of the original key. For online Extraction Reports, the last 4 bytes of each 8 byte key, which is the hash value, is replaced by dots.

Referencing Hash Translation

1. Select **Control Tasks (Jobs) > Input Sources > Extraction Definition > Key Field**.
2. Select the **Use Translation** check box.
3. Click the Translate button to display the Selection Translation Information dialog box.
4. At the bottom of the dialog box, **select Use hash translation for the key**.

Using History Analysis

This chapter describes the History Analysis function which allows you to analyze your history data and report the findings of the analysis. This chapter includes the following sections:

- “Understanding History Analysis” on page 221
- “Creating and Generating History Analysis Report Definitions” on page 222
- “Creating and Running History Analysis Process Definitions” on page 223
- “Summary Level and Key Length Example” on page 224
- “History Analysis Report Samples” on page 226
- “Cycle Tables” on page 227

Understanding History Analysis

History analysis extracts, analyzes, and reports on selected history data, producing a History Analysis Report. This report is cycle-oriented. A history key or an accumulation of history keys in the current cycle of data is compared to prior cycles of data and then displayed in the report.

Only history records whose cycle numbers match the entries in a cycle table are used in the history analysis processing. For more information, see “Cycle Tables” on page 227

In order to use history analysis, you need to create two kinds of definitions, History Analysis Report Definitions and History Analysis Process Definitions.

Creating and Generating History Analysis Report Definitions

History analysis report definitions define a History Analysis Report. To create a set of history analysis report definitions, do the following:

1. Click **New** under the History Analysis Report Definitions folder in the Control Entity view. The History Analysis Report Definition view opens.
2. Complete the Basic Report Information to provide formatting and printing options and determine how the history analysis will be processed. You will use this name in the Summary Levels dialog box when you create the history analysis process definitions.
3. Use the History Items folder to specify selected history items for a typical report where you do not want to process and display all of the 999 possible history items.
4. Use the Key Headings folder to provide heading text for the key portion of the report.
5. Use the Key Descriptions folder to specify:
 - Where to get the key values and where to print them in the Key portion of the History Analysis Report.
 - How many lines to skip after a key break.
6. Use the Columns folder to specify how and where to print the columns in the History Analysis Report. For a detailed explanation and examples of the Variance (non-Biased) and Standard Deviation (non-Biased) options in the **Column Type** field on the Columns dialog box, see [Appendix G, “Non-Biased Variance and Standard Deviation Computations”](#) on page 377.
7. Use the Analysis Rules folder to define one of the two types of analysis rules:
 - Threshold analysis rules specify the criteria used to detect data values that have fallen below a threshold value that you determine. The system scans column values (history items) from the oldest cycle to the current cycle. Column values that fall below the threshold value will be flagged as incorrect data with a user-defined exception code.

- Fluctuation analysis rules specify the criteria used to detect a data fluctuation situation. History items are scanned from the oldest cycle to the current cycle. History items whose data values lie outside specified upper and lower tolerance limits relative to a constant value or to another column, as determined by you, will be flagged as incorrect data with a user-defined exception code.
8. Save your work and update the definition database.

In order to generate the report, you need to create History Analysis Process Definitions that will be associated with it, as described in the next section.

Creating and Running History Analysis Process Definitions

History analysis process definitions define the history data to analyze and report in the History Analysis Report.

You can create and run a set of process definitions as follows:

1. Create a cycle table to specify the cycles that are to be used in the process definition. For instructions, see [“Cycle Tables” on page 227](#).
2. From the Control Entity view, click **New** under History Analysis Process Definitions to display the History Analysis Process Definition view.
3. Use the Basic Information folder enables you to specify the history items and cycles to analyze.
4. Use the Summary Levels folder to associate process definitions with report definitions. Make special note of the following:

Two aspects of Summary Levels must be considered: the Summary Level Number and the Key Length.

- The Key Length is that portion of the 40-character history key that identifies the information you wish to summarize. you define your key by giving a key length indicating how much of the history key you want to use at each summary level. if you choose less than the entire history key, your data will be collected and summarized for all history keys with the same key you have selected.
- When defining summary levels, you can define ten different levels of summarization and reporting. A Level 1 summarization produces one row of data per key for each history item defined. A Level 2 summarization has a Key Length less than or equal to the Key Length specified for your Level 1 summarization.

11 ■ Using History Analysis

Summary Level and Key Length Example

Your report will produce only one row of data per key. The row will contain the accumulation of all of the items defined on the database. The same summarization process will occur for each cycle that you request.

See the “Summary Level and Key Length Example” on page 224.

5. Select **Run > History Analysis** and complete the Run History Analysis dialog box. Make special note of the following:
 - Complete the **Cycle Number** field as follows:
 - If you specified Use Cycle Reference History Key or Use today’s Gregorian date as the cycle reference in this job, leave this field defaulted to zero. If the cycle number is not specified, history analysis will use the cycle processing options specified in the process definitions.
 - Otherwise specify the number to use as the current cycle for this execution of history analysis. Be sure that the number you enter is an entry in the cycle table defined.
 - If you want to limit the history keys that are processed, complete the **From** and **Through** fields base on the dialog box help.
6. When you have completed the dialog box, click **OK** to run the definitions. A sample report is shown in the next section. The report name will be uniexr.rpt.

Summary Level and Key Length Example

Suppose that you have the following data in your Detail History database (the History Keys are phone numbers: the area code, exchange and extension):

<u>History Key</u>	<u>Cycle</u>	<u>Item 1</u>	<u>Item 2</u>	<u>Item 3</u>
3124149617	080115	20	40	60
3124141628	080115	50	10	44
7085051800	080115	10	20	300
7085051801	080115	40	600	92
7085967164	080115	5	47	100

If you choose a Key Length of the entire History Key (the entire phone number) for Summary Level 1, the ACR/Detail report will contain the following data for cycle 080115:

Summary Level and Key Length Example

3124149617	Item 1 20
	Item 2 40
	Item 3 60
3124141628	Item 1 50
	Item 2 10
	Item 3 44
7085051800	Item 1 10
	Item 2 20
	Item 3 300
7085051801	Item 1 40
	Item 2 600
	Item 3 92
7085967164	Item 1 5
	Item 2 47
	Item 3 100

For Summary Level 2, since the Key Length must be less than or equal to what was used for Level 1, choose a Key Length of 6 (the area code and the exchange). All of the items will be added together (Item 1 + Item 2 + Item 3) and all of the totals with keys that match (those keys with the same first 6 characters), will be added together at this level. Your report will now contain this information for the 080115 cycle:

312414	224
708505	1062
708596	152

For Summary Level 3, specify a Key Length of 3 (just the area code). Now the totals you got from Level 2 will be added together for the keys that match (those keys with the same first 3 characters). Your report will now contain this information for the 080115 cycle:

312	224
708	1214

11 ■ Using History Analysis

History Analysis Report Samples

History Analysis Report Samples

Exception Message (Threshold Analysis) Report

This report type displays the data value and an exception code that you specify next to the data value.

DATE: yy/mm/dd TIME: 11:13:22		INFOGIX, INCORPORATED SAMPLE EXCEPTION REPORT			PAGE 1
RECONCILIATION KEY	ITEM DESCRIPTION	CURRENT CYCLE	PREVIOUS CYCLE (-1)	PREVIOUS CYCLE (-2)	AVERAGE OF ITEMS
---	-----	-----	-----	-----	-----
ABC	ITEM #1		>= AVERAGE	>= AVERAGE	
	ITEM #2		>= AVERAGE	>= AVERAGE	
	ITEM #3		>= AVERAGE	>= AVERAGE	
	ITEM #4		>= AVERAGE	>= AVERAGE	
GHI	ITEM #1	>= AVERAGE		>= AVERAGE	
	ITEM #2	>= AVERAGE		>= AVERAGE	
	ITEM #3	>= AVERAGE		>= AVERAGE	
	ITEM #4	>= AVERAGE		>= AVERAGE	
XYZ	ITEM #1	>= AVERAGE	>= AVERAGE		
	ITEM #2	>= AVERAGE	>= AVERAGE		
	ITEM #3	>= AVERAGE	>= AVERAGE		
	ITEM #4	>= AVERAGE	>= AVERAGE		

Data Values (Fluctuation Analysis) Report

This report type displays the data value and an exception code that you specify next to the data value.

DATE: yy/mm/dd TIME: 11:08:37		INFOGIX, INCORPORATED SAMPLE EXCEPTION REPORT			PAGE 1
RECONCILIATION KEY	ITEM DESCRIPTION	CURRENT CYCLE	PREVIOUS CYCLE (-1)	PREVIOUS CYCLE (-2)	AVERAGE OF ITEMS
---	-----	-----	-----	-----	-----
ABC	ITEM #1	3	27 @	15 @	15
	ITEM #2	6	30 @	18 @	18
	ITEM #3	9	33 @	21 @	21
	ITEM #4	12	36 @	24 @	24
GHI	ITEM #1	15 @	3	27 @	15
	ITEM #2	18 @	6	30 @	18
	ITEM #3	21 @	9	33 @	21
	ITEM #4	24 @	12	36 @	24
XYZ	ITEM #1	27 @	15 @	3	15
	ITEM #2	30 @	18 @	6	18
	ITEM #3	33 @	21 @	9	21
	ITEM #4	36 @	24 @	12	24

Cycle Tables

Cycle tables are used to specify the history records that can be included in a history analysis process. The system searches the cycle table to find a starting point matching the current cycle. That cycle number and all prior numbers up to the number of cycles specified in the basic process information are designated as valid cycles. These cycle numbers are then used to include or exclude history records for processing. Only history records whose cycle numbers match the valid cycles are used in the history analysis processing. A cycle table can have up to 1000 entries.

Example

Suppose your sales regions report to you once a week on Tuesdays and you would like to analyze the data that has been reported. You could set up and name a cycle table named TUESDAYS. Enter the Tuesday dates in ascending order as follows using July 4, 2008 as the beginning of the table.

Cycle Dates

20080704

20080711

20080718

20080725

20080801

etc.

When ACR/Detail searches the table to validate the cycle numbers to include in the analysis, it will only analyze Tuesday data.

Note: Cycle numbers must be entered in ascending order.

When you perform a batch update on the entries (add, delete and replace), no sorting is performed. You will receive an error message and will have to put the entries in order. To avoid this problem, make your entries in ascending order when you populate the table.

Creating a Cycle Table

1. In Control Entity View, in the Tables folder, select **New > Cycle Table** to display the Cycle Table View.
2. In the Basic Information folder, give the table a meaningful name. This name will be used in the Cycles dialog box when you set up your history analysis process definitions.
3. In the Cycle Table Entries folder, click **New** to display the Table Entry dialog box and enter the first cycle number you want to include in the table. Enter the cycle numbers in ascending order in the same format (for example, MMDDYY, YYJJJ) used in your job definitions. For cycle numbers representing dates later than December 31, 1999, use 20 in the first two century positions.
4. Click **New** again and complete the dialog box for each additional cycle number that you want to include in the table.
5. Save your work and update the definition database.

Using the Utilities

This chapter discusses the utilities available in ACR/Detail. It includes the following sections:

- “Running Utilities” on page 230
- “Database Initialization” on page 230
- “Database Update” on page 230
- “List Definitions” on page 231
- “Build Batch Transactions” on page 232
- “Build Extraction Program” on page 232
- “Cross Reference File Definitions” on page 232
- “List History” on page 233
- “Initialize History” on page 234
- “Copy History” on page 234
- “Copy Selected Records” on page 234
- “Unload History” on page 234
- “Reload History” on page 235
- “Delete History” on page 235
- “Reorganize History” on page 236
- “Extract History” on page 236
- “Print Management Report” on page 239
- “Initialize Translation Database” on page 239
- “List (Dynamic) Translation Database” on page 239
- “Reorganize (Dynamic) Translation Database” on page 240
- “Print Suspense Aging Report” on page 240
- “Update History” on page 241
- “Create Comma Delimited Text File” on page 241
- “Integrity Check” on page 242
- “UNINTC and UNICTN” on page 242

See Appendix B, “Reports” for samples of the output reports produced by each of the utilities discussed in this chapter.

Running Utilities

To run most utilities, select the **Utilities** icon or select **Run > Database Utilities**. When the Database Utilities dialog box displays, you select the utility you want and the associated dialog box will be displayed.

Note: The following utilities are exceptions:

- To run Database Initialization, select **Run > Database Initialization** and then complete the dialog box.
 - To run a Database Update, select the **Database Update** icon or select **Run > Database Update** and then complete the dialog box.
-

Database Initialization

This utility initializes (creates or re-creates) the definition and/or history database. Both databases must be initialized before you can begin using ACR/Detail for the first time.

Note: When reinitializing, be careful to specify the correct database to initialize, because existing data will be erased.

For an example of the report generated by this utility, see [“Database Initialization Report” on page 285](#).

To run Database Initialization, select **Run > Database Initialization** and then complete the dialog box.

Database Update

When an ACR/Detail Client job is created, saving the job and file definitions creates a .def file containing all batch transaction information. The Database Update utility applies the .def file containing the batch transactions to the definition database. This is necessary in order to use the definitions in ACR/Detail processing.

For an example and description of the report generated by this utility, see [“Definition Database Update Report” on page 299](#).

To run a Database Update, select the **Database Update** icon or select **Run > Database Update**.

List Definitions

This utility generates a report listing definitions from the definition database. This is useful in establishing an audit trail or for troubleshooting purposes. Definition types you can select include:

- All types
- Jobs
- Files
- Internal translation tables/cycle tables
- External translation tables
- (History Analysis) processes
- (History Analysis) reports
- Job and associated files. (This allows you list batch transaction definition records for all or selected job definitions and for their associated input source files at the same time.)

For examples and descriptions of the report sections you can generate using this utility, see the Definition Database Listing Reports beginning with “Database History and Statistics Report” on page 287.

Control Update Information Report Utility

This utility prints the most recent updates to definitions from the definition database, including the ID associated with the change and the date and time of the change. You can limit the listing based on definition type and date. Definition types you can select include:

- All types
- Jobs
- Files
- Internal translation tables/cycle tables
- External translation tables
- (History Analysis) processes
- (History Analysis) reports
- Job and associated files. (This allows you list batch transaction definition records for all or selected job definitions and for their associated input source files at the same time.)

For a sample report, see “Control Update Information Report” on page 299.

Build Batch Transactions

This utility builds batch definition transaction records from the records in the definition database. Batch transaction records are 80-character records that reflect the information stored on the definition database. This utility is useful for making mass definition changes, creating an audit trail, and transferring definitions between platforms.

Types of transactions you can select include:

- All types
- Jobs
- Files
- Internal translation tables/cycle tables
- (History Analysis) processes
- (History Analysis) reports
- Job and associated files. (This allows you build batch transaction definition records for all or selected job definitions and for their associated input source files at the same time.)

For an example and description of the report generated by this utility, see [“Build ACRUPDT Report” on page 302](#).

Build Extraction Program

This utility generates a VS COBOL II extraction program from file definitions for use with the Extraction Program Interface.

The generated programs are an integral part of Extraction Program Interface processing. For more information on the Extraction Program Interface, see [“Extraction Program Interface” on page 318](#).

Cross Reference File Definitions

This utility prints a report that lists all the file definitions used within the database and allows you to find job definitions that use the same file definitions. You can allocate a second file definition to search against two databases. The utility processes both databases in the same manner.

For a description and example of the report generated by this utility, see [“Definition Database Cross Reference Report” on page 301](#).

List History

This utility reports on the records in the history database. It can be useful in establishing an audit trail or for troubleshooting.

The following reports can be generated depending on your selections:

History Data Detail Report—Lists every history record and the values stored for that history.

Database History and Statistics Report—Lists detailed information about an individual history database such as the date and time of initialization, the date and time of the last update, the number of history keys, and the number of histories on file.

Note that the following values are always displayed with a value of zero on the report:

ESTIMATED NUMBER OF UNIQUE JOB IDS

ESTIMATED NUMBER OF HISTORY KEYS

AVERAGE NUMBER OF CYCLES PER KEY

AVERAGE NUMBER OF ITEMS PER KEY

Index of Detail Histories Report—Lists every history key in the history database and the number of histories (cycles) for each key.

Summary of Detail Histories Report—Lists information about each history key in the History database including cycle and run numbers, run date, run time, return code, and status of reruns and/or updates.

For example and descriptions of the reports generated by this utility, see the reports beginning with “[Database History and Statistics Report](#)” on [page 296](#).

Initialize History

This utility initializes (creates or re-creates) a history database.

Although you can initialize history through the Database Initialization dialog box, you will usually use the Initialize History utility when you need to create a second history database for use with the Copy History utilities.

Note: When reinitializing, be careful to specify the correct database to initialize, because existing data will be erased.

For a sample of the report produced by this utility, see “[Database Initialization Report](#)” on page 285.

Copy History

This utility copies all records from one history database to another. This utility can be useful for backup purposes or to create a copy of a production database to be used in a test environment.

For a sample of the report produced by this utility, see “[History Database Transfer Report](#)” on page 304.

Copy Selected Records

This utility copies selected records from an existing history database directly to another file. History records can be selected by history key, by relative date and by date and/or time.

For a sample of the report produced by this utility, see “[History Database Transfer Report](#)” on page 304.

Unload History

This utility unloads a history database to a flat file (or physical sequential file). History records can be selected by history key, by relative date and by date and/or time.

This utility might be used to:

- Backup/restore files
- Move a history database from one host to another

For a sample of the report produced by this utility, see “[History Database Unload Report](#)” on page 305.

Reload History

Warning: The Copy History reload option opens and closes the history database only once, which speeds up processing time significantly.

Because the history file will not be opened and closed for each record, to ensure the integrity of the file, you **MUST** run the reload at a time when no job or other process is using the file.

This utility can be used after you have used the Unload History utility. It reloads the previously unloaded history records to an empty database that you have already created. History records can be selected by history key, by relative date and by date and/or time.

For a sample of the report produced by this utility, see “Generated by the Copy History utility when you choose Reload records to a work file, his report lists each history record reloaded with information about the record.” on page 305.

Delete History

Use this utility to perform a physical delete of one or multiple history records based on a combination of key, cycle ID, job ID, and date criteria. Wildcards can be used to select multiple keys and multiple job IDs. The dialog box help provides detailed documentation.

Note: Be aware that when a physical delete is performed, the records cannot be recovered.

This utility can be used in various situations. For example, suppose Job B must run *after* Job A because Job B uses history items from Job A. If Job B runs first, it cannot get the correct information from Job A. Therefore you would delete the keys that were created in the incorrect run and rerun Jobs A and B.

Delete history could also be used when:

- A job ran and it was not scheduled to run.
- A job ran with the wrong cycle number.
- Old data exists on your history database (the key is no longer needed).

Reorganize History

This utility performs a physical reorganization of the contents of a history database in place. Detail history for any given history key is kept physically in order using relative cycle number. The utility rebuilds relative cycle numbers based on the actual cycle ID values found for any history key. Reorganizing history results in a more efficient file.

Extract History

This utility enables you to enter a number of user-defined criteria and extract records from the history database to a physical sequential file using one of two options:

- **Packed format.** Use this option to extract history records for 1) auditing or historical purposes or 2) to transfer the data to an application that requires comma-delimited format. For 2), after the extraction, you must run a second utility to convert the extracted data to comma-delimited format. See [“Create Comma Delimited Text File” on page 241](#).
- **Unpacked (alternate) format.** Use this option to create an extract file using unpacked data in order to easily move history data to another platform or product. You can optionally specify a platform identifier, a relative date indicator, and a cycle date format indicator. The file can be used with other Infogix, Inc. products to view the history records in a web browser.

Regardless of the option you use, the data extracted includes, for each extracted record, the following information:

- The version of the history extract program used to create the extract file
- The full history key
- The history cycle ID
- The reconciliation return code for the history
- The job ID of the reconciliation step that created the history
- The note data for the history
- A count of the number of items present
- The item values for the history

This utility generates the Transaction Listing Report. For more information, see [“Transaction Listing Report” on page 306](#).

The documentation for this utility includes the following sections:

- "Extract History File (Packed) Layout" below.
- "Extract History File (Alternate) Layout" on page 238

Extract History File (Packed) Layout

Fld #	Description	Position	Length	Format	Value
1	Database Version	1-4	4	X(4)	
2	Key	5-44	40	X(40)	
3	Reserved	45-84	40	X(40)	
4	Cycle and Run No.	85-95	11	9(11)	
5	Reserved	96-103	8	X(8)	
6	Job Name	104-111	8	X(8)	
7	Step Name	112-119	8	X(8)	
8	Qualifier	120-121	2	X(2)	
9	Return Code	122-125	4	X(4)	
10	Note Area	126-205	80	X(80)	
11	Reserved	206-243	38	X(40)	
12	Extended Value Count	244-245	2	S9(2)	Packed
12	Value Count	246-247	2	S9(2)	Packed
13 ¹	Run Value-X	248-8239	7992	1-999 X(8)	
14 ¹	Run Value-N	248-8239	7992	1-999 S9(15)	Packed
15	Extended Run Value-X	8240-16,240	8000	100 X(80)	
16	Extended Run Value-N	8240-16240	8000	100 S9(30)	Packed
1 - Positions 248-8239 can be populated by 1-999 8-byte alphanumeric fields or by 1-999 15-digit packed fields.					

12 ■ Using the Utilities

Extract History

Extract History File (Alternate) Layout

Fld #	Description	Position	Length	Format	Value
1	Job Name	1-8	8	X(8)	
2	Step Name	9-16	8	X(8)	
3	Qualifier	17-18	2	X(2)	
4	Key	19-58	40	X(40)	
5	Cycle Number	59-66	8	9(8)	
8	Run Date	67-74	8	9(8)	
7	Run Time	75-80	6	9(6)	
8	Return Code	81-84	4	X(4)	
9	Rerun Date	85-92	8	9(8)	
10	Rerun Time	93-98	6	9(6)	
11	Rerun Count	99-106	8	9(8)	
12	Run Value Count	107-109	3	9(3)	
13	Extended Run Value Count	110-112	3	9(3)	
14	Cycle Date	113-120	8	X(8)	
15	Format Indicator	121-18,102	1	999 X(1)	
16	Run Value	121-18,102	17	999 X(17)	
17	Extended Format Indicator	18,103-26,202	1	999 X(1)	
18	Extended Run Value-X	18,103-26,202	80	100 X(80)	
19	Extended Run Value-N	18,103-26,202	31	100 X(1) 100 9(30)	

Print Management Report

This utility allows you to process reconciliation information in the history database and produce management reports based on user-defined criteria. Managers can use this utility to analyze the functioning of the system. For example, you can run a report showing all the history keys that completed reconciliation with return codes greater than 0 (all out-of-balance jobs) for a particular time period.

For a sample of the report produced by this utility, see “[Detail Reconciliation Summary Report](#)” on page 286.

Initialize Translation Database

This utility initializes a dynamic translation database, which is required for the dynamic translation feature, which is documented in “[Using Dynamic Translation Tables](#)” on page 218. This database will be updated automatically when you run reconciliation.

Other utilities are available for dynamic translation tables:

- To list the contents of a dynamic translation database, use the List Translation Database utility (see below).
- To reorganize a dynamic translation database, see “[Reorganize \(Dynamic\) Translation Database](#)” on page 240.

List (Dynamic) Translation Database

This utility is used in conjunction with the dynamic translation feature, which is described in “[Using Dynamic Translation Tables](#)” on page 218.

This utility reports on the contents of the dynamic translation database. It lists each dynamic translation table along with the Identifier (the key after translation and the Original Source Value (the key before translation). Before you can run this utility, you must run the Initialize Translation Database utility (see “[Initialize Translation Database](#)” on page 239) to establish the dynamic translation database.

For a sample of the report, see “[Translation Database Listing Report](#)” on page 303.

Reorganize (Dynamic) Translation Database

This utility is used in conjunction with the dynamic translation feature, which is described in “[Using Translation Tables](#)” on page 193.

A dynamic translation database can store up to 99999999 keys per table ID. The keys are not automatically rolled off of the file in the way that history keys are rolled off of the history database. You can remove dynamic translation keys that are no longer in use by running the Reorganize Translation Database utility.

This utility reorganizes the information from the existing translation database into a new translation database, copying only used key numbers, as follows:

The utility searches the history database for run records that have a dynamic translation database entry. When it finds such a record, it searches the existing translation database for the associated entry, copies the entry to the new translation database, and updates the control record for the entry with the next available key number.

Print Suspense Aging Report

This utility produces a Suspense Aging Report showing aging information for up to two history items in the history database for suspense applications.

When you use suspense processing, you have a history database that consists of keys that are in “suspense.” The histories have an internal item that defines the date when the item went into suspense.

The Suspense Aging Report looks at the date information and summarizes the contents of the history (for all keys or for some level of the key if it contains a meaningful structure) in terms of the age of the specified items. The customizable criteria for the five aging columns is 0-30 days, 31-60 days, 61-90 days, 91-120 days, and >120 days.

An aging column for greater than a specific aging period can also be specified.

The day period is specified as greater than or equal to the From date and less than or equal to the To date. The month and year periods are specified as greater than the From date and less than or equal to the To date. This means that when you specify the aging period in days, you should identify the columns without overlapping criteria. For example, 0 - **30** days, **31** - 60

days, etc. However, when you specify the aging period in months or years, you should identify the columns using overlapping criteria. For example, 0 - 6 months, 6 - 12 months, etc.

For a sample of the report, see “[Suspense Aging Report](#)” on page 303.

Update History

This utility updates (replaces) one or more control values in the history database for a specified key and cycle ID. Up to 99 control values may be updated for each run of this utility. This utility can be used to correct out-of-balance situations when a job runs unsuccessfully and you later obtain the proper data to run your job.

Create Comma Delimited Text File

This utility creates a file with field values separated by commas. This utility can only be used after you have an extract file from the Extract History utility (see “[Extract History](#)” on page 236) or after you have created an output file using the Write Output File option found under Basic Job Information (see “[Write Output File](#)” on page 77).

The comma delimited file is in the following format:

- Full history key
- Cycle ID
- Return code
- Item count
- Items

All text values are surrounded by quotation marks and all dates appear either in the MM/DD/YY format if the century value is 19 or the MM/DD/CCYY format if the century value is not 19.

The comma delimited text file can be downloaded to a PC. In order to be opened in a spreadsheet application, the file must have a .csv extension.

Integrity Check

This utility examines the validity of the current control entity and its control tasks (jobs). An integrity check scrutinizes each item and its relationship to other items within the current control entity and identifies any inconsistencies in the Integrity Check Window.

An integrity check runs automatically when you do any of the following:

- Update the definition database.
- Run a job.
- Run a history analysis document.
- Select the **Integrity Check** icon.

Exceptions:

- If you create a control task named @FILETMP, an integrity check will not be performed on referenced translation tables. For more information, see “[Understanding @FILETMP Jobs](#)” on page 259.
- There is no integrity check for duplicate external translation table file names.
- The only integrity check that will be performed for conditional rules will take place on the left-hand side. If the LHS contains a conditional rule(s) with a higher rule number than the current conditional rule, an integrity check message will be issued. However, standard rules may be defined in any sequence and no integrity check will be issued.

UNINTC and UNICTN

The following utilities are available in Windows mode for jobs that will be uploaded to the mainframe:

- UNINTC, the Not (¬ ASCII 173) to Caret (^) utility
- UNICTN, the Caret (^) to Not (¬ ASCII 173) utility

For more information, see “[UNINTC - Not \(¬ ASCII 173\) to Caret \(^\) utility](#)” on page 317 and “[UNICTN - Caret \(^\) to Not \(¬ ASCII 173\) utility](#)” on page 317

Setting Up Free-Form Reports

This chapter explains how to create a Free-Form Report. It contains the following sections:

- “What is a Free-Form Report?” on page 243
- “Creating a Free-Form Report” on page 243
- “Viewing the Internal Codes” on page 251
- “Sample Free-Form Report” on page 251
- “Deleting a Free-Form Report” on page 252
- “Printing a Free-Form Report” on page 252

What is a Free-Form Report?

The Free-Form Report is a customizable report on the results of a reconciliation job. A maximum of 5 unique reports can be created for each job. You can also customize the sequence of the reports to print when you define more than one report. Depending on your selections, the report can include the run date and time, user name (company name from the Control File), reconciliation key, job ID, cycle number, return codes, messages, Control Report title, internal items, extended internal items, history items, calculated items, rules, descriptions of internal and history items, and free-form text. You determine where to place the information on the report.

If specified, this report is generated when you run reconciliation.

The file name for the report in your output will be UNIUFR.RPT.

Creating a Free-Form Report

Create the report as follows:

1. Optionally see “Setting the User Options” on page 173 to set several system-wide options for report formatting.
2. Plan the report. Decide what information to include and where to place it on the report, including both the headings and the data items. You may have a sample report that you can use as a template.

3. In the **Free-form Report** view, under the **Basic Information** section, specify a **Report ID** on the Free form Report Name dialog box and then the set the print specifications on the Print Free-Form Report dialog box. Do this before creating the report because it specifies whether the report will print, which keys will print, the report width, length and print sequence, and other important options.

If you are using multi-level reconciliation, specify the appropriate Print options separately for the base job and for each qualifier job.

The XML option enables you to set the job-level option for generating the report in XML format. Your setting here will override your setting in the user options.

4. From the **Report Information Definitions** folder, in the Free-form Report View, double-click **Free-Form Report** to display the Free-form Report View. Familiarize yourself with the following components of the view:

Menu Bar—The Edit Menu, View Menu, Draw Menu, Object Menu, and Report Menu contain options that can be useful when creating the report. Some of these menus will be referenced in the following instructions.

Toolbar—When you are in Free-Form Report View, you will see the toolbar icons that display in Control Task (Job) View and an additional set of icons pertaining only to Free-Form Reports. For an explanation of these icons, see [“Icons for Free-Form Report View” on page 248](#).

Workspace—The majority of the view is the workspace that you will use to design the report. Any items available for the Control Report are available for the Free-Form Report.

To hide/show the grid lines when creating your report, click **Grid Lines** in the **View** menu.

5. Set the font specifications. Use the Font option on the View Menu to specify the report font. For more information, see [“Changing the Font” on page 268](#).
6. Familiarize yourself with the use of report objects. For more information see [“Working with Report Objects” on page 246](#).
7. Optionally create a report header and a page header. See [“View the report by selecting the Open Reports icon. The file name for this report in your output will be UNIUFR.RPT.Creating Headers” on page 245](#).

8. Under the headers, create the other report components. Select each object you want to include by clicking the tool bar icon or by selecting it from the Draw Menu. Each object represents an item such as: cycle number, internal item, extended internal item, history item, rule, and text string.
9. Save your definitions and update the definition database.
10. Run the job to automatically create the report.
11. View the report by selecting the **Open Reports** icon. The file name for this report in your output will be UNIUFR.RPT.Creating Headers

You can optionally create two kind of headers for a Free-Form Report:

- Report header. This prints once at the top of the report.
- Page header. This prints under the report header on the first page and at the top of every subsequent page.

Creating a Report Header

To create a report header:

1. Place the objects you want to include in the report header in the rows at the top of the report.
2. To specify a row as a report header line, after clicking the Report Header Marker icon (\R), click at the bottom of the row. A solid line will display, as shown below.



Creating a Page Header

To create a page header:

1. Place the objects you want to include in the page header in the rows at the top of the report under the report header.
2. To specify a row as a page header line, after clicking the Page Header Marker icon (\P), click at the bottom of the row. A broken line will display as shown below.



Note: Page breaks, in contrast to page header lines, appear as dash-dot lines in the view.

13 ■ Setting Up Free-Form Reports

Creating a Free-Form Report

Example

Free-Form Report Window

In the following example, the headers are as follows:

- The report header consists of 3 blank lines followed by a line with the job name and step name.
- The page header consists of one line with the run date, Control Report title, and run time.

SCR1035 /STEP1 /		
mm/DD/YYYY	Job to Demonstrate Free-Form Report	HH.MM.SS
Entire Key		
QUANTITY		QUANTITY
COST		COST
TOTAL		TOTAL

Printed Report

The first page of the printed report will look like this:

JOB1035 /STEP1 /			
mm/dd/yy	Job to Demonstrate Free-Form Report		16.31.42
ITEM1			
QUANTITY		587	
COST		\$1.56	
TOTAL		\$915.24	

The second page of the printed report will look like this:

mm/dd/yy	Job to Demonstrate Free-Form Report		16.31.42
ITEM2			
QUANTITY		587	
COST		\$1.56	
TOTAL		\$915.72	

Working with Report Objects

Following is the procedure for working with report objects:

1. Click on the appropriate icon or select the object from the Draw menu. All Free-Form Report icons are shown in "Icons for Free-Form Report View" on page 248.
2. Position your cursor in the workspace location where you want the object to display.

3. Click and drag the object to the desired size. Some objects have a preset minimum size, however, you may increase the size of any object.
4. Right-click to select any of the following:

Properties—Sets object properties. (Or you can double-click on the object or select the object and select **Object > Properties** from the menu.) If the object has properties that you can change, the dialog box for specifying the properties displays. Otherwise you will get a message stating that the object cannot be edited.

Justify—The justification options control how your item will print: left-, center-, or right-justified, or as is. Regardless of the option you choose, the value that will appear on the report will be limited by the report object size and, where applicable, by the **Display partial text** feature.

The As is option specifies that no justification will be applied and leading and trailing spaces will be retained. Because the size of the report object does not change based on the size of the value, the placement and size you specify for the report object are important.

Example of As-is Versus Left/Right Justification

This example shows the results of as-is vs. left justification when the value is smaller than the number of positions specified

Assume that you are placing a text value on the report. The value consists of 2 leading spaces followed by 123 followed by 2 trailing spaces (7 characters). You enter 8 as the number of positions to reserve for the field on the report.

- If you use as-is justification, the value will print as 8 characters, consisting of 2 spaces followed by 123 followed by 3 spaces.
- If you use left justification, the value will print as 8 characters, consisting of 123 followed by 5 spaces.
- If you use right justification, the value will print as 8 characters, consisting of 5 spaces followed by 123.

Cut, Copy, Paste, Delete

Grid Lines—Allows you to toggle grid lines on and off.

Font—Provides access to the Font option on the View Menu. This displays the Font dialog box, which provides the following fields:

Font—Enables you to change the font on the display but not on the printed report.

Font style—Enables you to change the font style (bold, italics, etc.) on both the display and the printed report.

13 ■ Setting Up Free-Form Reports

Creating a Free-Form Report







Size—Enables you to change the font size on both the display and the printed report.












5. If necessary, select the object handles or click and drag to readjust the size or placement of the object.
6. If you want to delete the object, select it and click the **Delete** key.

Icons for Free-Form Report View

The following table provides a list of the objects and object customization commands available for Free-Form reports. These commands can be accessed either by clicking on the icon shown in left column or by selecting from the Draw Menu or the Object Menu, as indicated in the right column.

The commands accessed via the Draw menu are for selecting report objects and placing them in the report. The commands accessed from the Object Menu are for customizing or formatting objects that you have placed in the report as described in [step 4](#) through [step 6](#) in “[Working with Report Objects](#)” on page 246.

Icons for Free-Form Report View		
Icon	Command	Menu to Access This Command
	Select (Click to select an object. Click and drag to select multiple objects.)	Draw Menu
	Page Break (Adds a page break where you click). Appears as a dash-dot line in the display.	Draw Menu
	Report header marker. Makes the row a report header line (displays as a solid line at the bottom of the row). Each report header line will be printed once at the top of the report.	Draw Menu
	Page header marker. Makes the row a page header line (displays as a broken line at the bottom of the row). Each page header line will be printed under the report header on the first page and at the top of each subsequent page.	Draw Menu
	Run Date (of the report)	Draw Menu
	Run Time (of the report)	Draw Menu

Icons for Free-Form Report View		
Icon	Command	Menu to Access This Command
	User Name (company name from the Control File)	Draw Menu
	Control Report Title (specify the Control Report Title to be displayed on this report)	Draw Menu
	Job ID	Draw Menu
	Cycle Number	Draw Menu
	Reconciliation Key	Draw Menu
	Return Code	Draw Menu
	Message	Draw Menu
	String (enter a text string)	Draw Menu
	Internal Item Description	Draw Menu
	Internal Item	Draw Menu
	Extended Internal Item Description	Draw Menu
	Extended Internal Item	Draw Menu
	History Item Description	Draw Menu

13 ■ Setting Up Free-Form Reports

Creating a Free-Form Report

Icons for Free-Form Report View		
Icon	Command	Menu to Access This Command
	History Item	Draw Menu
	Calculated Item	Draw Menu
	Rule Variance	Draw Menu
	Page Number	Draw Menu
	Key Break	Draw Menu
	Key Break Value	Draw Menu
	Key Break Total	Draw Menu
	Properties	Object Menu (in Free-Form Report View)
	Left Justify	Object Menu (in Free-Form Report View)
	Center Justify	Object Menu (in Free-Form Report View)
	Right Justify	Object Menu (in Free-Form Report View)
	As is (with no justification). Leading and trailing spaces are not removed. For more information on this command, see Working with Report Objects on page 246.	Object Menu (in Free-Form Report View)

Viewing the Internal Codes

You can optionally view the internal code representation for each object in the report along with the positioning of each. To do so, open the Internal Representation dialog box (**Report > Internal Code**). The codes are documented in the dialog box help.

Sample Free-Form Report

```

LJOB/STEP: UGREPORT/SAMPLE /
CYCLE: 00000001
KEY: 001 FF01527
ROUTE CODE PHL:LAX
STAFF NUMBER 12
TOTAL TICKET PRICE $2,717.00
ISSUE DATE AND TIME (FROM C-00 03/01/21 17:23:32
TICKET NUMBER 000456234
ROUTE DESCRIPTION PHILADELPHIA LOS AN
KEY: 001 FF48379
ROUTE CODE DTW:MSY
STAFF NUMBER 22
TOTAL TICKET PRICE $2,804.50
ISSUE DATE AND TIME (FROM C-00 03/01/21 18:45:11
TICKET NUMBER 007624562
ROUTE DESCRIPTION DETROIT NEW OR
KEY: 001 FF91827
ROUTE CODE DCA:JFK
STAFF NUMBER 24
TOTAL TICKET PRICE $1,578.50
ISSUE DATE AND TIME (FROM C-00 03/01/23 14:14:10
TICKET NUMBER 000366744
ROUTE DESCRIPTION WASHINGTON, DCNEW YO

OFFICE TOTAL----> 001 $7,100.00
KEY: 002 FF48379
ROUTE CODE ORD:DTW
STAFF NUMBER 72
TOTAL TICKET PRICE $1,250.00
ISSUE DATE AND TIME (FROM C-00 03/01/15 08:40:34
TICKET NUMBER 004534899
ROUTE DESCRIPTION CHICAGO DETROI
KEY: 002 FF91827
ROUTE CODE JFK:DCA
STAFF NUMBER 13
TOTAL TICKET PRICE $1,333.00
ISSUE DATE AND TIME (FROM C-00 03/01/23 13:07:22
TICKET NUMBER 089347839
ROUTE DESCRIPTION NEW YORK WASHIN

OFFICE TOTAL----> 002 $2,583.00
KEY: 003 FF01527
ROUTE CODE DEN:LAX
STAFF NUMBER 11
TOTAL TICKET PRICE $4,736.00
ISSUE DATE AND TIME (FROM C-00 03/01/23 11:04:44
TICKET NUMBER 000875642
ROUTE DESCRIPTION DENVER LOS AN
KEY: 003 FF48379
ROUTE CODE JFK:ORD
STAFF NUMBER 53
TOTAL TICKET PRICE $794.00
ISSUE DATE AND TIME (FROM C-00 03/01/14 12:17:45
TICKET NUMBER 000023445
ROUTE DESCRIPTION NEW YORK CHICAG

1CYCLE: 00000001
OFFICE TOTAL----> 003 $5,530.00

```

Deleting a Free-Form Report

To delete a Free-Form Report, click to select the free-form report under the **Free-Form Reports** section in the Control Task (Job) view, then press the **Delete** key.

Printing a Free-Form Report

The Print Free-Form Report dialog box enables you to do the following:

- Set your preferences for printing the Free-Form Report (**Free-Form Report View > Basic Information > Print Free-Form Report**) including the report width, length and print sequence, if you specify one or more reports for a reconciliation job.
- Set your preferences for generating the Free-Form Report in XML format.

Setting Up User Reports

This chapter explains how to create a User Report. It contains the following sections:

- “What is a User Report?” on page 253
- “Setting Up a User Report” on page 253
- “Deleting a User Report” on page 255
- “User Report Sample” on page 255

What is a User Report?

The User Report is a customized report that enables you to format columns to show any of the data that can be included in the Control Report in the order that you specify. For a sample and field definitions, see “[User Report Sample](#)” on page 255. The file name for this report in your output will be UNIDUR.RPT.

Setting Up a User Report

Set up the report as follows:

1. In the Report Information section at the bottom of the Control Task (Job) View, double-click Reconciliation User Report to access the User Report View.
2. Complete the Basic Information folder.

The Auto Positioning of Columns dialog box enables you to select auto-positioning for columns, user-specified column positions, or fixed spacing.

The User Report XML Option dialog box enables you to set a job-level option for generating the User Report in XML format. This setting will override the corresponding option in your user options. If the report is for an RLQ (Reconciliation Level Qualifier) job, complete the Use Base and RLQ Definitions option to determine whether the system will maintain the headings and page numbering from the base job's User Report in this report.

14 ■ Setting Up User Reports

Setting Up a User Report

3. Optionally complete the Page Header Information folder. This allows you to define a one-line header containing your choice of information that will appear above the column headings on each page of the report. This information will pertain to the keys on the page. To create such a header, click **New** to define each item of information and its placement on the line.

If you use dynamic translation, you can choose Original Key to print the original key value before translation.

4. Complete the Column Information folder. Click **New** to define each column. Make special note of the following:
 - If you specified auto positioning of columns in the Basic Information folder, do not specify a column width or column position.
 - In the **Source Area** field, if you select Reconciliation Rule Value, remember that all standard rules, both active and inactive, will be evaluated when reconciliation is run. Inactive rules that have not been made active by a conditional rule will be evaluated but will not display on the Control Report. However, if selected in this field, the rule value, after it has been evaluated, will display on the User Report.

If you use dynamic translation, you can choose Original Key to print the original key value before translation.

- The **Accumulate this column value** field is not valid with source areas of key field, cycle ID, rule, or message.
 - The **Print column detail** field allows you to suppress or print the column data in the user report.
5. Optionally complete the Key Break Information folder. Click **New** to define each key break you want, if any.

Key breaks are used to specify the end of a series of identical keys in a report which the system will then accumulate and total. For example, if you want to accumulate and total the cost of phone calls by area code, you would enter a Key Size of three so that any change in the first three digits of the key (i.e. a new area code) would cause a key break.

Make a special note of the following option:

The **Print accumulated column totals** field allows you to suppress or print the accumulated column total in the key break line on the user report.

6. Save your work and update the definition database before running the job.

Deleting a User Report

To delete a User Report, you need to delete all entries in each folder by clicking on them and pressing the **Delete** key. Multiple items may be selected by clicking on each item with the **Shift** key depressed.

User Report Sample

ACR/D releasenumbr		RECONCILIATION USER REPORT					PAGE 1	
DATE: yy/mm/dd		INFOGIX, INCORPORATED					JOB ID: UGREPORT/SAMPLE /	
TIME: 11:58:10		TICKET REFUND REPORT						
20150103								
ROUTE CODE	STAFF	TICKET\$	ISSUE	TICKET#	ROUTE DESCRIPTION	FLIGHT#	REC TOT	
PHL:LAX	12	\$2717.00	2003/01/21 17:23:32	000456234	PHILADELPHIA LOS ANGELES	FF01527		
DTW:MSY	22	\$2804.50	2003/01/21 18:45:11	007624562	DETROIT NEW ORLEANS	FF48379		
DCA:JFK	24	\$1578.50	2003/01/23 14:14:10	000366744	WASHINGTON, DCNEW YORK	FF91827		
TOTAL BY OFFICE		\$7100.00					3	
ORD:DTW	72	\$1250.00	2003/01/15 08:40:34	004534899	CHICAGO DETROIT	FF48379		
JFK:DCA	13	\$1333.00	2003/01/23 13:07:22	089347839	NEW YORK WASHINGTON, DC	FF91827		
TOTAL BY OFFICE		\$2583.00					2	
DEN:LAX	11	\$4736.00	2003/01/23 11:04:44	000875642	DENVER LOS ANGELES	FF01527		
JFK:ORD	53	\$794.00	2003/01/14 12:17:45	000023445	NEW YORK CHICAGO	FF48379		
TOTAL BY OFFICE		\$5530.00					3	

14 ■ Setting Up User Reports

User Report Sample

Importing Definitions

This chapter provides documentation for importing definitions into ACR/Detail Client. It includes the following sections:

- “Purposes of Importing” on page 257
- “Procedure for Importing Definitions” on page 257
- “Understanding @FILETMP Jobs” on page 259
- “Using @@FILLER Records” on page 259

Purposes of Importing

Typically, definitions are imported for the following purposes:

- To move batch definition transactions created in a prior release into the current release.
- To move batch definition transactions from one platform to another.

Procedure for Importing Definitions

Steps for Importing

To import definitions, follow these steps:

1. If the file containing the definitions resides on UNIX, Linux, or z/OS, the file must be downloaded before importing.
 - Downloading instructions for UNIX or Linux are provided in “[Downloading, Uploading, and Importing Definitions](#)” on page 37.
 - Downloading instructions for z/OS are provided in “[Downloading and Uploading between z/OS and ACR/Workbench](#)” on page 309.
2. Choose **File** > **Import**. This opens the Open dialog box.
3. Select the .def (batch definition transactions) file containing the definitions you want to import, and click **OK**. This opens the Import Options dialog box.
4. Complete the dialog box based on the instructions in the dialog box help and the “[Considerations for Completing the Import Options Dialog Box](#)” on page 258. When you click **OK**, a new control entity containing the imported definitions will be displayed.

15 ■ Importing Definitions

Procedure for Importing Definitions

5. Click the **Save** icon to store the definitions.

Considerations for Completing the Import Options Dialog Box

The following information will assist you in completing the Import Options dialog box:

- If you choose the List Files option, only the file (input source) definitions from the specified .def file will be displayed. You then select the ones you want to import. After you click **OK**, the Control Task (Job) View will be displayed. The imported file IDs will be in the Input Sources folder. In the Basic Information folder, the job name will be @FILETMP. This is a reserved name for a control task that contains only file definitions. For more information see [“Understanding @FILETMP Jobs” on page 259](#).
- The **Name Creation** field has two options, Use Descriptions and Use Item Numbers. These options are applicable when moving definitions from z/OS. For more information, see [“Eliminating Discrepancies when Moving Definitions from z/OS” on page 311](#).
- The **Insert filler records when item numbers are skipped** option should be selected as follows:

Note: To see the item numbers (for history items, calculated items, and rules), select **Options > Batch Options >** to display the Batch Options dialog box and select the option Show item numbers in Control Task

- Select this option if the history items, calculated items, or rules to be imported have non-consecutive item numbers, and you want to include filler records (named @@FILLER) to maintain the gap between the numbers. Any sequence numbers you have set up through the Batch Options dialog box will be cleared. The filler records will use sequence numbers starting with 1. For more information, see [“Using @@FILLER Records” on page 259](#).
- Do not select this option if you want imported history items, calculated items, and rules that have non-consecutive item numbers to be renumbered consecutively starting with 1.

Understanding @FILETMP Jobs

@FILETMP is a reserved name for a control task that contains only file definitions. This name is assigned automatically when you perform a file import using the **List Files** option.

Using @@FILLER Records

What Are Filler Records?

Filler records (named @@FILLER in the Control Task [Job] View) are place holders for maintaining the gap between item numbers.

Note: To see the item numbers, select **Options > Batch Options >** to display the Batch Options dialog box and select the option Show item numbers in Control Task.

They are useful when you are working with history items, calculated items, and rules with non-consecutive numbers. Filler records can be inserted in two ways:

- When importing, you can select **Insert filler records when item numbers are skipped** on the Import Options dialog box as explained in [step 4 on page 257](#).

Note: If you want to use filler records and you have item number sequences in the Batch Options dialog box, the Import function will clear the sequence numbers and complete the import using filler records starting with number 1.

- From the Control Task (Job) View, you can highlight a row and select **Edit > Insert Filler**. Filler records are inserted in the row above the highlighted row.

Filler records have the following characteristics:

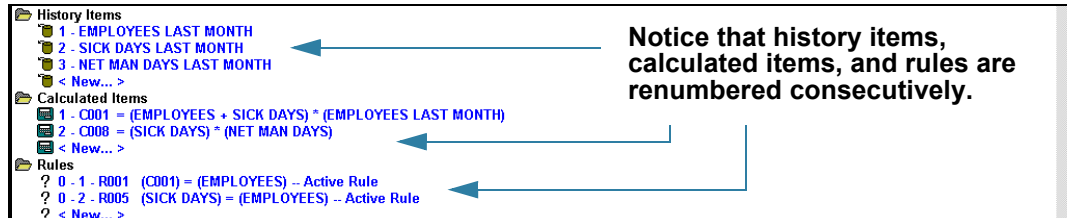
- They are only applicable to history items, calculated items, and rules.
- They can be cut, copied, pasted, or deleted, but cannot be edited.
- They are not checked by the Integrity Check utility.
- They will not be written to the batch definition transaction file (.def) and therefore will not be in the definition database.
- They are excluded from dialog boxes that display lists of history items, calculated items, or rules.

15 ■ Importing Definitions

Using @@FILLER Records

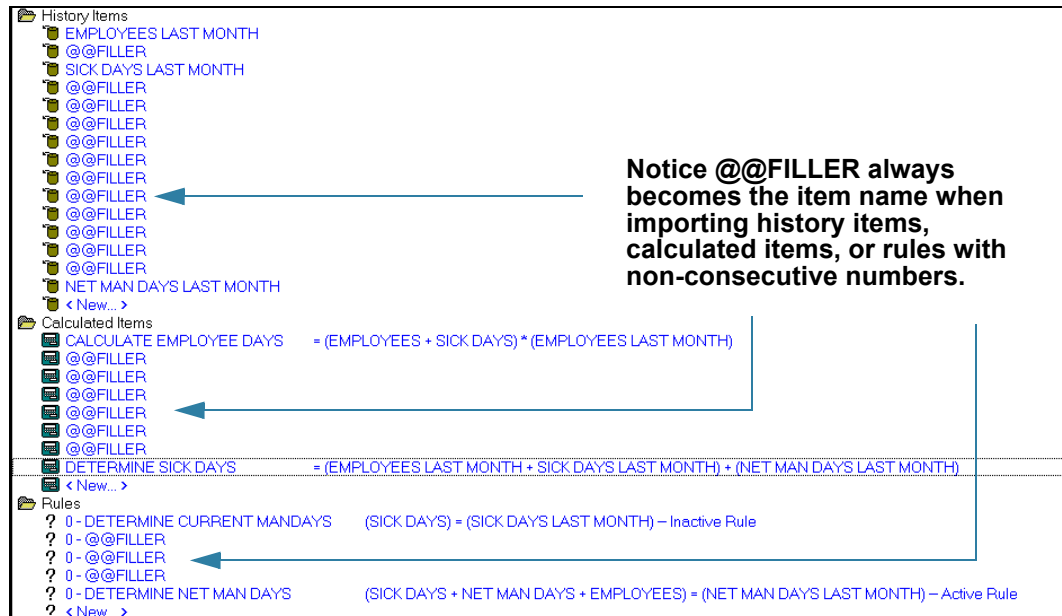
Examples of Items Imported with and without Insert Filler Records Option Selected

The following example shows items imported with the INSERT filler records when item numbers are skipped option **deselected**.



The screenshot displays a tree view of imported items. Under 'History Items', items 1, 2, and 3 are listed: '1 - EMPLOYEES LAST MONTH', '2 - SICK DAYS LAST MONTH', and '3 - NET MAN DAYS LAST MONTH'. Under 'Calculated Items', items 1 and 2 are listed: '1 - C001 = (EMPLOYEES + SICK DAYS) * (EMPLOYEES LAST MONTH)' and '2 - C008 = (SICK DAYS) * (NET MAN DAYS)'. Under 'Rules', items 1 and 2 are listed: '0 - 1 - R001 (C001) = (EMPLOYEES) -- Active Rule' and '0 - 2 - R005 (SICK DAYS) = (EMPLOYEES) -- Active Rule'. Blue arrows point from the text 'Notice that history items, calculated items, and rules are renumbered consecutively.' to the numbered items in each category.

The following example shows items imported with the Insert filler records when item numbers are skipped option **selected**.



The screenshot displays a tree view of imported items. Under 'History Items', the first item is 'EMPLOYEES LAST MONTH', followed by eight '@@FILLER' items, and then 'NET MAN DAYS LAST MONTH'. Under 'Calculated Items', the first item is 'CALCULATE EMPLOYEE DAYS = (EMPLOYEES + SICK DAYS) * (EMPLOYEES LAST MONTH)', followed by six '@@FILLER' items, and then 'DETERMINE SICK DAYS = (EMPLOYEES LAST MONTH + SICK DAYS LAST MONTH) + (NET MAN DAYS LAST MONTH)'. Under 'Rules', the first item is '0 - DETERMINE CURRENT MANDAYS (SICK DAYS) = (SICK DAYS LAST MONTH) - Inactive Rule', followed by three '@@FILLER' items, and then '0 - DETERMINE NET MAN DAYS (SICK DAYS + NET MAN DAYS + EMPLOYEES) = (NET MAN DAYS LAST MONTH) - Active Rule'. Blue arrows point from the text 'Notice @@FILLER always becomes the item name when importing history items, calculated items, or rules with non-consecutive numbers.' to the @@FILLER items in each category.

Using Batch Definition Transactions

This appendix describes the format and use of the batch definition transaction records. Typically this information is used to make mass changes to definitions that run in production. This appendix contains the following sections:

- “What Is a Batch Definition Transaction?” on page 261
- “Format of Batch Definition Transaction Records” on page 262
- “Updating the Definition Database” on page 264
- “Purging Definitions” on page 265

What Is a Batch Definition Transaction?

A batch definition transaction is a record stored in the definition database and used to apply changes to ACR/Detail definitions in batch.

ACR/Detail definitions include job definitions, file definitions, table definitions, history analysis process definitions, and user report definitions. Each type of definition is divided into definition information categories. A category contains multiple fields containing related definition information. For example, internal item information is one category of job definitions. Definition fields for internal items include information such as the internal item number and the item description. These are all fields containing information related to that category of information.

Each batch definition transaction record consists of 80 positions containing all of the information for a particular ACR/Detail definition along with a transaction type specifying how the transaction will be applied to the definition database. Each type of batch transaction record is identified by a transaction code. The records are created in a sequential file using the following methods:

- Any edit facility (such as Notepad) can be used to code information on batch definition transaction records.

A ■ Using Batch Definition Transactions

Format of Batch Definition Transaction Records

- The Build Batch Transactions utility (**Run > Database Utilities > Build Batch Transactions**) can be used to create batch definition transactions from job, file, table, history analysis process, and user report definitions that you have created through ACR/Detail Client. This enables you to create a template of the transaction format, which can then be copied and modified as needed.

Note: The positional nature of batch definition transaction records makes manually created records more prone to entry errors than records created through ACR/Detail Client.

Format of Batch Definition Transaction Records

Transaction Key

Each 80-character record begins with a 26-position transaction key that identifies each transaction and specifies how to process it.

Note: Records with an asterisk (*) in position 1 are treated as comments and are ignored during processing.

The transaction key consists of the following:

Transaction ID (18 Positions)

A definition identifying the specific transaction.

Job Definitions—the 18-character Job ID.

File Definitions—the 10-character file ID, the six-digit record sequence number, and two blanks.

Table Definitions—the 16-character table name and two blanks.

History Analysis Process Definitions—the 16-character process ID and two blanks.

User Report Definitions—the 10-character report ID and eight blanks.

Transaction Code

A code identifying the category of information on the transaction record. Codes beginning with:

- **J** are Job Definitions
- **F** are File Definitions
- **T** are Cycle Table or Internal Translation Table Definitions
- **X** are External Translation Table Definitions

A ■ Using Batch Definition Transactions

Format of Batch Definition Transaction Records

- **R** are History Analysis Report Definitions
- **E** are History Analysis Process Definitions
- **U** are User Report Definitions

The second and third characters of the code identify the definition information categories. For example, F4's are selection field definitions, F6's are key field definitions, J31's are history item description definitions, etc.

Transaction Number (3 Positions)

A number identifying the specific transaction within its category, such as the internal item number or selection field number.

Transaction Sequence (1 Position)

A number identifying the sequence of the records for a transaction. Sometimes a transaction requires 2 or more 80-character transaction records to contain all the definition information. The transaction sequence number is used to control the sequence of those records.

Transaction Type (1 Position)

A code specifying how the transaction is to be applied to the definition database when processed by the update utility. Valid transaction types are:

Transaction Type	Meaning
A	Add a new definition to the database.
D	Delete an existing definition from the database. When a transaction consists of more than one transaction record, only the first record (Transaction Sequence 1) is needed for delete processing. This record needs to contain only the transaction key (see also type * below).
R	Replace an existing record in the database.
*	Purge an entire set of definitions from the database. The Purge Transaction Type is valid only on a Basic Job Information, Basic File Information or Table Information transaction (i.e., on the first transaction record for each type of definition). When specified, it causes all definitions for that Definition ID to be deleted from the database. The transaction is ignored if no definitions exist for that Job ID key, File ID key, or Table Name in the database. For more information on purging, see “Processing Batch Definition Transaction Sets” below and Purging Definitions on page 265.

Processing Batch Definition Transaction Sets

A set of batch definition transactions is usually headed by a single purge transaction type, and then followed by a complete set of one or more Add transactions that contain your job, file, table, history analysis process, or user report definition parameters. For example, a set might contain a basic job information purge transaction (to delete all previous definitions for that job ID from the definition database), immediately followed by a basic job information Add transaction, and then followed by internal item Add, and so on. In other words, you purge all of the old definitions and add the new ones.

Batch definition transactions are automatically sorted by transaction key by the batch update utility before they are applied to the definition database. This means that the purge type is always applied first (i.e., the utility program will first purge the database of any previous definitions for that transaction ID, and then process all the other batch definition transactions that add to the database). This ensures that no “leftover” definitions will interfere with your new definitions, and that the set of Add transactions you process contains the complete set of definitions.

Alternatively, instead of purging a complete set of definitions and then adding the complete revised set to the database, you can Add, Replace and/or Delete individual definitions. The resultant executable definitions will be a combination of whatever definitions were originally in the database, with the individual Add, Replace and Delete transactions applied against them. The set of transactions you process is only a subset of the complete set. For this reason, this method is rarely used, and thus the transaction types **R** and **D** are rarely used.

Updating the Definition Database

Before the definitions can be used, the definition database must be updated with the batch definition transactions by running Update Definition Database using `udsuptd.bat` (Windows) or `udsuptd.sh` (UNIX or Linux). First, the batch definition transactions are sorted by Transaction key. Then, the `udsuptd.bat` batch file processes the batch definition transactions in two passes: first, the transactions are validated; then, if all the transactions are valid, they are processed or applied against the Definition database.

The transaction validation pass edits each individual transaction to ensure that valid interrelated information is entered in the proper columns. Each transaction is printed in the transaction validation section of the Definitions Update Report. For example, the field position in selection field

information must be numeric and between 1 to 9999. If it isn't, an error message will appear to the right and on the line below the transaction on the transaction validation section of the Definitions Update Report.

If all transactions are valid, transaction processing is performed. Using the transaction type from each transaction's key, the udsupdt.bat batch file attempts to Add, Delete, Replace, or Purge the specified transaction. If any errors occur (such as trying to Delete a definition that does not exist in the database), processing stops and no transactions are applied. Any error messages will be printed, along with the transactions in the transaction processing section of the Definitions Update Report.

Once applied to the definition database, the reconciliation function or history analysis function can be used to test the definitions, or you can use the List Definitions utility or batch file to get a formatted list of the definitions.

Purging Definitions

To purge definitions from the definition library or the definition database, you must submit a batch transaction with a single purge card and an asterisk (*). The purge cards are as follows:

Transaction Code	Transaction Type	Definition Type
J0	*	Basic Job Information Transaction Layout
F0	*	Basic File Information Transaction Layout
R0	*	History Analysis Process Information Transaction Layout
U0	*	Reconciliation User Report Basic Information Transaction Layout
T0	*	Cycle Table or Internal Translation Table Information Transaction Layout
X0	*	External Translation Table Information Transaction Layout

The following shows an example of deleting all file definitions for the file DD EMPSALES.

```
EMPSALES 000000 F0 0001*
```

The asterisk (*) indicates to select all file definitions for purging.

Reports

This appendix contains samples of multiple ACR/Detail reports and information on the report contents (or cross-references to this information). It contains the following sections:

- “Working with Reports in ACR/Detail Client” on page 267
- “Reports Table” on page 269

Working with Reports in ACR/Detail Client

Setting System-Level Options for Report Formatting

Several options for report formatting can be set in your user options. See “Setting the User Options” on page 173.

Opening, Cascading, and Closing Reports

To view all reports from the most recent run, click on the **Open Reports** icon. Select the **Cascade Windows** icon to display reports in a cascaded arrangement. Then click on the report you wish to view.

To close all open reports, click on the **Close Reports** toolbar icon.

Printing Reports

To print the active report, click on **Print** icon.

Printing the Century

If you want dates to show the century, select the **Print century before year** field in the User Options dialog box - Date Formats tab. (Select **Options > User Options** to display the dialog box.)

Display and Formatting of Reports

Icons and View Menu Options

When a report is open, icons display at the top of the report window to allow you to do the following: zoom in, zoom out, find, repeat find, toggle on/off the shade bars, toggle on/off the ruler.

In addition, you select **View** from the menu to see multiple options for changing the view, including the orientation (portrait or landscape).

B ■ Reports

Working with Reports in ACR/Detail Client

Changing the Font

Select **View > Font** to display the Font dialog box, enabling you to use the following fields to make font changes:

Field	Purpose
Font	Enables you to change the type of font.
Font style	Enables you to change the font style (bold, italics, etc.).
Size	Enables you to change the font size.

Changing the Margins, Colors, and Shade Bar

Select **Reports > Page Setup** to display the Page Setup dialog box and make changes as follows:

Dialog Box Tab	Purpose
Margins	Set the margins and show/hide the ruler.
Colors	Select the background color and text color.
Shade Bar	Set defaults for how the shade bar will display.

Distribution of Reports

Except in UNIX or Linux mode, the following features are available:

- The Control Report, System Messages [SYSOUT] Report, Recap Report, User Report, and Free-Form Report, or any combination, can be attached to an e-mail address specified in the Direct Message dialog box. E-mail notification must be set up as described in “[Message Processing](#)” on page 106.
- Reports from any ACR/Detail Client processing can be published to the Web using settings in the Add/Edit Configuration dialog box - Internet Information tab.

Viewing Reports on UNIX or Linux

You can view reports on the UNIX or Linux host using any editor. You may also want to view the reports on your screen by using a command such as the “more” command. You can also simply print the reports for viewing.

Reports Table

Report	Source	Page Number
System Messages Report	Multiple Processes	page 271
Control Report	Reconciliation	page 271
Recap Report	Reconciliation	page 275
Free-Form Report	Reconciliation	page 243
User Report	Reconciliation	page 255
Extracted Data Detail Report	Reconciliation	page 276
Sorted Data Detail Report	Reconciliation	page 277
Accumulated Data Detail Report	Reconciliation	page 278
History Data Detail Report	Reconciliation	page 279
Extraction Trace Report	Reconciliation	page 280
External Translation Table Trace Report	Reconciliation when using an External Translation Table	page 208
Database Initialization Report	Database Initialization	page 285
Detail History Analysis Report (Exception Message Report)	History Analysis	page 286
Detail Reconciliation Summary Reports	Print Management Report	page 286
Database History and Statistics Report	List Definitions Utility - All	page 287
Index of Definition Keys Report	List Definitions Utility - All	page 287
Detailed Job Definitions Report	List Definitions Utility - Job	page 288
Detailed File Definitions Report	List Definitions Utility - File	page 291
Detailed Table Definitions Report (Internal Translation Table)	List Definitions Utility - Internal Translation	page 291
Detailed Table Definitions Report (External Translation Table)	List Definitions Utility - External Translation	page 293
Detailed Exception Definitions (History Analysis)	List Definitions Utility	page 294
Detailed Report Definitions (History Analysis)	List Definitions Utility	page 295

B ■ Reports

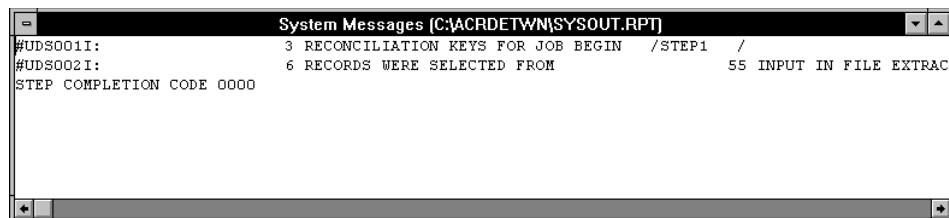
Reports Table

Report	Source	Page Number
Database History and Statistics Report	List History Utility - All	page 296
Index of Detail Histories Report	List History Utility - All	page 297
Summary of Detail Histories Report	List History Utility - All	page 297
History Data Detail Report	List History Utility - All or Detail	page 298
Control Update Information Report	Print Control Update Information Utility	page 299
Definition Database Update Report	Update from Batch Transactions Utility	page 299
Definition Database Cross Reference Report	Cross Reference File Definitions Utility	page 301
Build ACRUPDT Report	Build Batch Transactions Utility	page 302
Translation Database Listing Report	List Translation Database Utility	page 303
Suspense Aging Report	Print Suspense Aging Report Utility	page 303
History Database Transfer Report	Copy History Utility - Copy All Records, Copy Selected Records	page 304
History Database Unload Report	Copy History Utility - Reload Records from a Work File	page 305
History Database Reload Report	Copy History Utility - Unload Records to a Work File	page 305
Transaction Listing Report	Multiple	page 306

System Messages Report (SYSOUT.RPT)

This report shows the number of reconciliation keys used and the number of records selected, along with either a step completion code (the same one you saw on the left side of the status bar when reconciliation completed) or a step completion code and error message(s) describing any problems encountered during the reconciliation run.

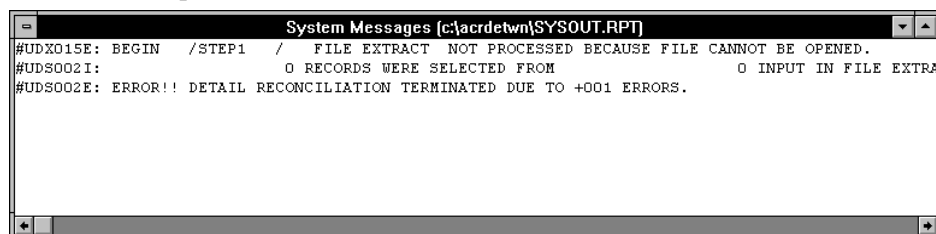
A SYSOUT report from a successful reconciliation run is shown below.



```

System Messages [C:\ACRDETWN\SYSOUT.RPT]
#UDS001I:      3 RECONCILIATION KEYS FOR JOB BEGIN  /STEP1  /
#UDS002I:      6 RECORDS WERE SELECTED FROM          55 INPUT IN FILE EXTRAC
STEP COMPLETION CODE 0000
  
```

A SYSOUT report from an unsuccessful reconciliation run is shown below.



```

System Messages [c:\acrdetwn\SYSOUT.RPT]
#UDX015E: BEGIN  /STEP1  /  FILE EXTRACT  NOT PROCESSED BECAUSE FILE CANNOT BE OPENED.
#UDS002I:      0 RECORDS WERE SELECTED FROM          0 INPUT IN FILE EXTRA
#UDS002E: ERROR!! DETAIL RECONCILIATION TERMINATED DUE TO +001 ERRORS.
  
```

Control Report

This report is the primary output of ACR/Detail. By default, it prints when you run a reconciliation job. It displays all the components of the reconciliation job and shows you the results of the reconciliation.

Customizing the Control Report to Your Needs

1. You can optionally set several system-wide options for report formatting. See [“Setting the User Options”](#) on page 173.
2. You can use the Print Control Report dialog box to turn off printing of the report, to specify that the report will print for all reconciliation keys or just the out-of-balance keys, or to print an out-of-balance summary at the end of the report.

B ■ Reports

Control Report

3. Generating in XML format: If you are not using user options or if you want to override the value in user options, you can optionally use the Control Report XML Option dialog box in the Basic Information section of Control Task (Job) View to set a job-level option that will override the XML option you set for the Control Report setting in your user options.
4. At run-time you can optionally select the **Overrides** button on the Run Reconciliation dialog box to display the Override Options dialog box and set selected run-time overrides.

Interpreting the Control Report

Following is an example of each report page along with information about its contents.

Control Report Page 1 of 3

ACR/D releasenumbr		INFOGIX, INCORPORATED		COPYRIGHT INFOGIX, INC. 2014	
DATE: yy/mm/dd		RECONCILIATION CONTROL REPORT		PAGE	
TIME: 09:40:49		PRINTING USER GUIDE SAMPLE REPORTS		DATA BASE VERSION: 0000002/000000	
ENTRY KEY= 001 FF01527				CYCLE#=00000003 RUN#=000	
				JOB ID=UGREPORT/SAMPLE /	
A. INTERNAL ITEMS:					
ITEM NO.	DESCRIPTION	LEFT	RIGHT		
I-001.	ROUTE CODE	'PHL:LAX '			
I-002.	STAFF NUMBER	12			
I-003.	TOTAL TICKET PRICE	\$2,717.00			
I-004.	ISSUE DATE	03/01/21			
I-005.	ISSUE TIME	17:23:32			
I-006.	ISSUE DATE AND TIME (FROM C-001)	03/01/21	17:23:32CALC'ED C-001		
I-007.	TICKET NUMBER	0000000000456234			
I-008.	CITY OF ORIGIN - AIRPORT CODE	'PHL '			
I-009.	CITY OF DESTINATION - AIRPORT CODE	'LAX '			
EXTENDED INTERNAL ITEMS:					
ITEM NO.	DESCRIPTION	VALUE			
X-001.	ROUTE DESCRIPTION	'PHILADELPHIA LOS ANGELES			
X-002.	CITY OF ORIGIN	'PHILADELPHIA			
X-003.	CITY OF DESTINATION	'LOS ANGELES			
X-004.	NOTES EXTRACTED FROM EXTENDED LITERAL	'REQUEST SUBMITTED; PENDING APPROVAL FROM THE CHICAGO HEADQUARTER			
X-005.	ROUTE & FLIGHT#	'PHL:LAX&FF01527			
B. HISTORY ITEMS:					
ITEM NO.	DESCRIPTION	INPUT	OUTPUT		
E-001.	FROM I-001: ROUTE(AIRPORT CODE)	'PHL:LAX '			
	----- HISTORY KEY -----	ITEM	CYCLE	-CYCLE#-	RUN#
001	FF01527	I-001	-01	00000002	000
		RUN DATE: 15/01/07	RUN TIME: 09:40:38		
E-002.	FROM I-002: STAFF NUMBER	12			
	----- HISTORY KEY -----	ITEM	CYCLE	-CYCLE#-	RUN#
001	FF01527	I-002	-01	00000002	000
		RUN DATE: 15/01/07	RUN TIME: 09:40:38		
E-003.	FROM I-003: TOTAL TICKET PRICE	\$2,717.00			
	----- HISTORY KEY -----	ITEM	CYCLE	-CYCLE#-	RUN#
001	FF01527	I-003	-01	00000002	000
		RUN DATE: 15/01/07	RUN TIME: 09:40:38		
OUT OF BALANCE (RETURN CODE = 1001)					
INFOGIX DATABASE: R99UC43.DET47W.DEFN		(UNIDF) / R99UC43.DET47W.HIST		(UNIHF)	
PROGRAM= UDS2000 LOAD LIBRARY NAME=UNI.TAPESHIP.R40V7M00.LOADLIB		JOB START DATE=15/01/07		JOB START TIME=09.40.49	

Report Header and Footer

The header displays the page number, date and time of the report, database version you are using, job information (job name, step name, and RLQ), and cycle ID (cycle number and run number) of the reconciliation step. The example shows 22-character format.

The footer displays the definition and history databases, the program that generated the report, and the job start date and time.

The remaining sections are identified by a letter and section name, as follows:

A. Internal Items and Extended Internal Items. For each regular or extended internal item, this section lists the item type (I for internal item, X for extended internal item) and item number, along with its description and value. If the value came from a calculated item, that is also noted.

B. History Items. This section lists each history item description and value along with the corresponding history key, item type and item number, relative cycle number, actual cycle number, run number, run date and time, value extracted, and any messages.

Note: Control values that do not fit on the line will be printed on the next line.

Control Report Page 2 of 3

ACR/D releasenumbr		INFOGIX, INCORPORATED		COPYRIGHT INFOGIX, INC. 2014	
DATE: yy/mm/dd		RECONCILIATION CONTROL REPORT		PAGE	
TIME: 09:40:49		PRINTING USER GUIDE SAMPLE REPORTS		DATA BASE VERSION: 0000002/000000	
ENTRY KEY= 001 FF01527				CYCLE#=00000003 RUN#=000	
				JOB ID=UGREPORT/SAMPLE /	
B. HISTORY ITEMS:					
ITEM NO.	DESCRIPTION	INPUT	OUTPUT		
E-004.	FROM I-007: TICKET NUMBER	00000000000456234			
	----- HISTORY KEY -----	ITEM	CYCLE	-CYCLE#-	RUN#
001	FF01527	I-007	-01	00000002	000
		RUN DATE:	15/01/07	RUN TIME:	09:40:38
E-005.	FROM I-006: (ISSUE DATE AND TIME)	03/01/21	17:23:32		
	----- HISTORY KEY -----	ITEM	CYCLE	-CYCLE#-	RUN#
001	FF01527	I-006	-01	00000002	000
		RUN DATE:	15/01/07	RUN TIME:	09:40:38
E-006.	CYCLE NUMBER AND RUN NUMBER(REL CYC -01)	0000000000000000002			
	----- HISTORY KEY -----	ITEM	CYCLE	-CYCLE#-	RUN#
001	FF01527	C-000	-01	00000002	000
		RUN DATE:	15/01/07	RUN TIME:	09:40:38
E-007.	RUN DATE (REL CYC -01)	15/01/07			
	----- HISTORY KEY -----	ITEM	CYCLE	-CYCLE#-	RUN#
001	FF01527	D-000	-01	00000002	000
		RUN DATE:	15/01/07	RUN TIME:	09:40:38
E-008.	RUN TIME (REL CYC -01)	00:15:40			
	----- HISTORY KEY -----	ITEM	CYCLE	-CYCLE#-	RUN#
001	FF01527	T-000	-01	00000002	000
		RUN DATE:	15/01/07	RUN TIME:	09:40:38
C. CALCULATED ITEMS:					
ITEM NO.	CALCULATION FORMULA	RESULT	CALC MESSAGE		
C-001.	(I-004 & I-005) + (0)	03/01/21	17:23:32		
C-002.	(I-003) * (.15)	\$407.55			
C-003.	(I-003) * (.09)	\$244.53			
C-004.	(99999999) INTO (@DAYOFWK)	3			
C-005.	(99999999) INTO (@TIMECSEC)	09:40:49			
C-006.	(99999999) INTO (@DTMCSEC)	15/01/07	09:40:49		
C-007.	(10) INTO (@CYCLERUN)	00000000000003010			
OUT OF BALANCE (RETURN CODE = 1001)					
INFOGIX DATABASE: R99UC43.DET47W.DEFN		(UNIDF) / R99UC43.DET47W.HIST		(UNIHF)	
PROGRAM= UDS2000 LOAD LIBRARY NAME=UNI.TAPESHIP.R40V7M00.LOADLIB		JOB START DATE=15/01/07		JOB START TIME=09.40.49	

B ■ Reports

Control Report

C. Calculated Items. This section lists each calculated item number, calculation formula and the result of the calculation.

Control Report Page 3 of 3

ACR/D releasenumbr		INFOGIX, INCORPORATED		COPYRIGHT INFOGIX, INC. 2014	
DATE: yy/mm/dd		RECONCILIATION CONTROL REPORT		PAGE	
OTIME: 09:40:49		PRINTING USER GUIDE SAMPLE REPORTS		DATA BASE VERSION: 000002/000000	
ENTRY KEY= 001 FF01527				CYCLE#=0000003 RUN#=000	
				JOB ID=UGREPORT/SAMPLE /	
				RETURN	
D. BALANCING RULES & RESULTS:					
RULE NO.	BALANCING RULE	RESULT	ERROR/TOLERANCE		
1.	(I-003) LE (2000.00)	**OUT OF BALANCE**	\$717.00		
2.	(I-007) LE (10000000)	IN BALANCE			
E. MESSAGES:					
RULE NO.	RETURN CODE	INSTRUCTIONS			
1.	1001	TICKET PRICE \$2,717.00 OUT OF RANGE!			
OUT OF BALANCE (RETURN CODE = 1001)					
INFOGIX DATABASE: R99UC43.DET47W.DEFN (UNIDF) / R99UC43.DET47W.HIST (UNIHF)					
PROGRAM= UDS2000 LOAD LIBRARY NAME=UNI.TAPESHIP.R40V7M00.LOADLIB JOB START DATE=15/01/07 JOB START TIME=09.40.49					

D. Balancing Rules & Results. This section lists each rule number, the comparison formula, and the result. If a rule is out of balance or is using a tolerance, the error/tolerance column shows the amount the rule is out of balance or out of the tolerance range.

To understand the kinds of rule status information that the report will include, see “Reporting Rule Status” on page 105.

E. Special Instructions. If one or more rules are out of balance, this section shows the text of any messages. If all rules are in balance, it shows None unless a message has been defined for the return code 0000 (in balance).

Control Report: Out-of-Balance Summary

The Out-of-Balance Summary prints at the end of the Control Report. There are two sections: Out-of-Balance Key Tolerance and Out-of-Balance Rules.

ACR/D releasenumbr		INFOGIX, INCORPORATED		COPYRIGHT INFOGIX, INC. 2013	
DATE: yy/mm/dd		OUT-OF-BALANCE SUMMARY		PAGE 1	
TIME:16:01.15		OOB KEY TOLERANCE		DATA BASE VERSION 0000215/0000077:ENTRY	
				CYCLE#=20130122 RUN#=000	
				JOB ID=OOBKEY1 /ACCMT /	
OUT-OF-BALANCE KEY TOLERANCE:					
KEYS	JOB TOTAL	OUT-OF-BALANCE TOTAL	TOLERANCE	ACTION TAKEN	
3		3	99.99%		
VALUE	\$13,703,580.36-	\$13,703,580.36-	60.00% SET RC 3999		
OUT-OF-BALANCE RULES:					
RULE NO.	# KEYS PASSED	# KEYS FAILED	OUT-OF-BALANCE VALUE		
0001	3	0	0		
0002	0	3	\$13,703.58036-		
0005	0	3	16,703.58-		
TOTAL NUMBER OF RULES: 3					
TOTAL NUMBER OF RULES PASSED: 1					
TOTAL NUMBER OF RULES FAILED: 2					

Recap Report

By default, this report prints when you run a reconciliation job. It provides a summary listing for all keys that were reconciled, or just the out-of-balance keys, depending on your specification in the Print Recap Report dialog box (you can also turn off printing of the report in that dialog box). The report shows the reconciliation results and any return codes. Following is an example and explanation of the columns.

ACR/D releasenumbr		INFOGIX, INCORPORATED		COPYRIGHT INFOGIX, INC. 2014	
DATE: yy/mm/dd		RECONCILIATION RECAP REPORT		PAGE 1	
TIME: 14:53:27		PRINTING USER GUIDE SAMPLE REPORTS		DATA BASE VERSION: 0000006/000000	
				JOB ID=UGREPORT/SAMPLE /	
KEY NO.	KEY	CYCLE #	RUN #	STATUS	CODE
1	001 FF01527	00000001	000	**OUT OF BALANCE**	1001
2	001 FF48379	00000001	000	**OUT OF BALANCE**	1001
3	001 FF91827	00000001	000	IN BALANCE	0000
4	002 FF48379	00000001	000	IN BALANCE	0000
5	002 FF91827	00000001	000	**OUT OF BALANCE**	1002
6	003 FF01527	00000001	000	**OUT OF BALANCE**	1001
7	003 FF48379	00000001	000	IN BALANCE	0000

KEY NO. This column numbers the keys in the reconciliation job.

KEY. This column lists keys used to pick up the data values from the input source. In the example above, the three keys listed represent companies ABC, DEF, and XYZ.

CYCLE #. This column shows the cycle numbers used for each key.

STATUS. This column shows if the key is in balance or out of balance.

CODE. The return code associated with the reconciliation of this key.

Extracted Data Detail Report

If specified in the Print Extracted Data Detail Report dialog box, the Extracted Data Detail Report prints when you run a reconciliation job. It shows the key and detail field values extracted from every input file record that met one or more selection criteria. This report is used in debugging.

ACR/D Releasename		INFOGIX, INCORPORATED				COPYRIGHT INFOGIX, INC. 2014			
DATE: yy/mm/dd		EXTRACTED DATA DETAIL				REPORT ID: DXR001		PAGE	
TIME: 14:53:27		PRINTING USER GUIDE SAMPLE REPORTS				DATA BASE VERSION: 0000006/*****			
<--- JOB ID --->	CYCLE RUN	----- KEYS -----				FROM RECORD	<--- JOB/STEP --->	<-FILE ID-->	
UGREPORT SAMPLE	00000001 000 001	FF48379				3	UGREPORT SAMPLE	UGRPTF1	
VALUES:	1:(ORD:BOS)	2:()	3:(11)	4:(96500)	5:(146834)	6:()	7:()	8:(4033000)	9:()
	6:()	7:(12345)	8:(ORD)	9:(BOS)	10:()	11:()	12:()	13:()	14:()
EXTENDED VALUES:	1:(CHICAGO BOSTON)								
	2:(CHICAGO)								
	3:(BOSTON)								
	4:(REQUEST SUBMITTED; PENDING APPROVAL FROM THE CHICAGO HEADQUARTER)								
	5:(ORD:BOS&FF48379)								
<--- JOB ID --->	CYCLE RUN	----- KEYS -----				FROM RECORD	<--- JOB/STEP --->	<-FILE ID-->	
UGREPORT SAMPLE	00000001 000 002	FF48379				4	UGREPORT SAMPLE	UGRPTF1	
VALUES:	1:(BOS:ORD)	2:()	3:(3)	4:(96550)	5:(146837)	6:()	7:()	8:(3247100)	9:()
	6:()	7:(15729999)	8:(BOS)	9:(ORD)	10:()	11:()	12:()	13:()	14:()
EXTENDED VALUES:	1:(BOSTON CHICAGO)								
	2:(BOSTON)								
	3:(CHICAGO)								
	4:(REQUEST SUBMITTED; PENDING APPROVAL FROM THE CHICAGO HEADQUARTER)								
	5:(BOS:ORD&FF48379)								
<--- JOB ID --->	CYCLE RUN	----- KEYS -----				FROM RECORD	<--- JOB/STEP --->	<-FILE ID-->	
UGREPORT SAMPLE	00000001 000 001	FF48379				5	UGREPORT SAMPLE	UGRPTF1	
VALUES:	1:(ORD:JFK)	2:()	3:(72)	4:(79450)	5:(146838)	6:()	7:()	8:(2900200)	9:()
	6:()	7:(4567822)	8:(ORD)	9:(JFK)	10:()	11:()	12:()	13:()	14:()
EXTENDED VALUES:	1:(CHICAGO NEW YORK)								
	2:(CHICAGO)								
	3:(NEW YORK)								
	5:(ORD:JFK&FF48379)								
<--- JOB ID --->	CYCLE RUN	----- KEYS -----				FROM RECORD	<--- JOB/STEP --->	<-FILE ID-->	
UGREPORT SAMPLE	00000001 000 002	FF91827				6	UGREPORT SAMPLE	UGRPTF1	
VALUES:	1:(JFK:MEM)	2:()	3:(22)	4:(110000)	5:(146838)	6:()	7:()	8:(4407300)	9:()
	6:()	7:(5432)	8:(JFK)	9:(MEM)	10:()	11:()	12:()	13:()	14:()
EXTENDED VALUES:	1:(NEW YORK MEMPHIS)								
	2:(NEW YORK)								
	3:(MEMPHIS)								
	4:(REQUEST SUBMITTED; PENDING APPROVAL FROM THE CHICAGO HEADQUARTER)								
	5:(JFK:MEM&FF91827)								
<--- JOB ID --->	CYCLE RUN	----- KEYS -----				FROM RECORD	<--- JOB/STEP --->	<-FILE ID-->	
UGREPORT SAMPLE	00000001 000 003	FF48379				7	UGREPORT SAMPLE	UGRPTF1	
VALUES:	1:(JFK:ORD)	2:()	3:(53)	4:(79400)	5:(146841)	6:()	7:()	8:(4426500)	9:()
	6:()	7:(23445)	8:(JFK)	9:(ORD)	10:()	11:()	12:()	13:()	14:()
EXTENDED VALUES:	1:(NEW YORK CHICAGO)								
	2:(NEW YORK)								
	3:(CHICAGO)								
	4:(REQUEST SUBMITTED; PENDING APPROVAL FROM THE CHICAGO HEADQUARTER)								
	5:(JFK:ORD&FF48379)								

Sorted Data Detail Report

If specified in the Print Sorted/Accumulated Data Detail Report dialog box, the Sorted Data Detail Report prints when you run a reconciliation job. It shows the keys and detail field values extracted from every input file record meeting at least one selection criterion. Extracted data is shown after it has been sorted by reconciliation key. This provides you with a picture of what the history database will contain if the history updates are completed with no errors. This report is used in debugging.

ACR/D Releasename	INFOGIX, INCORPORATED	COPYRIGHT INFOGIX, INC. 2014
DATE: yy/mm/dd	REPORT ID: DXR001	PAGE
TIME: 14:53:28	PRINTING USER GUIDE SAMPLE REPORTS	DATA BASE VERSION: 0000006/*****
<--- JOB ID --->	CYCLE RUN <----- KEYS ----->	FROM RECORD <--- JOB/STEP ---> <-FILE ID-->
UGREPORT SAMPLE 00000001 000 001	FF01527	12 UGREPORT SAMPLE UGRPTF1
VALUES: 1:(PHL:LAX) 2:() 3:(12) 4:(271700) 5:(146848) 6:() 7:(456234) 8:(PHL) 9:(LAX) 10:()		
EXTENDED VALUES:	1:(PHILADELPHIA LOS ANGELES) 2:(PHILADELPHIA) 3:(LOS ANGELES) 4:(REQUEST SUBMITTED; PENDING APPROVAL FROM THE CHICAGO HEADQUARTER) 5:(PHL:LAX&FF01527)	
<--- JOB ID --->	CYCLE RUN <----- KEYS ----->	FROM RECORD <--- JOB/STEP ---> <-FILE ID-->
UGREPORT SAMPLE 00000001 000 001	FF48379	3 UGREPORT SAMPLE UGRPTF1
VALUES: 1:(ORD:BOS) 2:() 3:(11) 4:(96500) 5:(146834) 6:() 7:(12345) 8:(ORD) 9:(BOS) 10:(4033000)		
EXTENDED VALUES:	1:(CHICAGO BOSTON) 2:(CHICAGO) 3:(BOSTON) 4:(REQUEST SUBMITTED; PENDING APPROVAL FROM THE CHICAGO HEADQUARTER) 5:(ORD:BOS&FF48379)	
<--- JOB ID --->	CYCLE RUN <----- KEYS ----->	FROM RECORD <--- JOB/STEP ---> <-FILE ID-->
UGREPORT SAMPLE 00000001 000 001	FF48379	5 UGREPORT SAMPLE UGRPTF1
VALUES: 1:(ORD:JFK) 2:() 3:(72) 4:(79450) 5:(146838) 6:() 7:(4567822) 8:(ORD) 9:(JFK) 10:(2900200)		
EXTENDED VALUES:	1:(CHICAGO NEW YORK) 2:(CHICAGO) 3:(NEW YORK) 4:(REQUEST SUBMITTED; PENDING APPROVAL FROM THE CHICAGO HEADQUARTER) 5:(ORD:JFK&FF48379)	
<--- JOB ID --->	CYCLE RUN <----- KEYS ----->	FROM RECORD <--- JOB/STEP ---> <-FILE ID-->
UGREPORT SAMPLE 00000001 000 001	FF48379	13 UGREPORT SAMPLE UGRPTF1
VALUES: 1:(DTW:MSY) 2:() 3:(22) 4:(104500) 5:(146848) 6:() 7:(7624562) 8:(DTW) 9:(MSY) 10:(6751100)		
EXTENDED VALUES:	1:(DETROIT NEW ORLEANS) 2:(DETROIT) 3:(NEW ORLEANS) 4:(REQUEST SUBMITTED; PENDING APPROVAL FROM THE CHICAGO HEADQUARTER) 5:(DTW:MSY&FF48379)	
<--- JOB ID --->	CYCLE RUN <----- KEYS ----->	FROM RECORD <--- JOB/STEP ---> <-FILE ID-->
UGREPORT SAMPLE 00000001 000 001	FF91827	8 UGREPORT SAMPLE UGRPTF1
VALUES: 1:(MEM:ORD) 2:() 3:(3) 4:(55100) 5:(146841) 6:() 7:(6729900) 8:(MEM) 9:(ORD) 10:(4801100)		
EXTENDED VALUES:	1:(MEMPHIS CHICAGO) 2:(MEMPHIS) 3:(CHICAGO) 4:(REQUEST SUBMITTED; PENDING APPROVAL FROM THE CHICAGO HEADQUARTER) 5:(MEM:ORD&FF91827)	

B ■ Reports

Accumulated Data Detail Report

Accumulated Data Detail Report

If specified in the Print Sorted/Accumulated Data Detail Report dialog box, the Accumulated Data Detail Report prints when you run a reconciliation job. It shows the extracted data shown on the Sorted Data Detail Report after it has been accumulated by reconciliation key. This report is used in debugging.

ACR/D Releasename	INFOGIX, INCORPORATED	COPYRIGHT INFOGIX, INC. 2014
DATE: yy/mm/dd	ACCUMULATED DATA DETAIL	REPORT ID: DXR001 PAGE
TIME: 15:15:34	PRINTING USER GUIDE SAMPLE REPORTS	DATA BASE VERSION: 000007/*****
<--- JOB ID --->	CYCLE RUN <----- KEYS ----->	FROM RECORD <--- JOB/STEP ---> <-FILE ID->
UGREPORT SAMPLE	00000001 000 001 FF01527	12 UGREPORT SAMPLE UGRPTF1
VALUES: 1:(PHL:LAX) 2:() 3:(12) 4:(271700) 5:(146848) 6:() 7:(456234) 8:(PHL) 9:(LAX) 10:()		
EXTENDED VALUES:	1:(PHILADELPHIA LOS ANGELES) 2:(PHILADELPHIA) 3:(LOS ANGELES) 4:(REQUEST SUBMITTED; PENDING APPROVAL FROM THE CHICAGO HEADQUARTER) 5:(PHL:LAX&FF01527)	
<--- JOB ID --->	CYCLE RUN <----- KEYS ----->	FROM RECORD <--- JOB/STEP ---> <-FILE ID->
UGREPORT SAMPLE	00000001 000 001 FF48379	3 UGREPORT SAMPLE UGRPTF1
VALUES: 1:(DTW:MSY) 2:() 3:(22) 4:(280450) 5:(146848) 6:() 7:(7624562) 8:(DTW) 9:(MSY) 10:()		
EXTENDED VALUES:	1:(DETROIT NEW ORLEANS) 2:(DETROIT) 3:(NEW ORLEANS) 4:(REQUEST SUBMITTED; PENDING APPROVAL FROM THE CHICAGO HEADQUARTER) 5:(DTW:MSY&FF48379)	
<--- JOB ID --->	CYCLE RUN <----- KEYS ----->	FROM RECORD <--- JOB/STEP ---> <-FILE ID->
UGREPORT SAMPLE	00000001 000 001 FF91827	8 UGREPORT SAMPLE UGRPTF1
VALUES: 1:(DCA:JFK) 2:() 3:(24) 4:(157850) 5:(146850) 6:() 7:(366744) 8:(DCA) 9:(JFK) 10:()		
EXTENDED VALUES:	1:(WASHINGTON, DCNEW YORK) 2:(WASHINGTON, DC) 3:(NEW YORK) 4:(REQUEST SUBMITTED; PENDING APPROVAL FROM THE CHICAGO HEADQUARTER) 5:(DCA:JFK&FF91827)	
<--- JOB ID --->	CYCLE RUN <----- KEYS ----->	FROM RECORD <--- JOB/STEP ---> <-FILE ID->
UGREPORT SAMPLE	00000001 000 002 FF48379	4 UGREPORT SAMPLE UGRPTF1
VALUES: 1:(ORD:DTW) 2:() 3:(72) 4:(125000) 5:(146842) 6:() 7:(4534899) 8:(ORD) 9:(DTW) 10:()		
EXTENDED VALUES:	1:(CHICAGO DETROIT) 2:(CHICAGO) 3:(DETROIT) 4:(REQUEST SUBMITTED; PENDING APPROVAL FROM THE CHICAGO HEADQUARTER) 5:(ORD:DTW&FF48379)	
<--- JOB ID --->	CYCLE RUN <----- KEYS ----->	FROM RECORD <--- JOB/STEP ---> <-FILE ID->
UGREPORT SAMPLE	00000001 000 002 FF91827	6 UGREPORT SAMPLE UGRPTF1
VALUES: 1:(JFK:DCA) 2:() 3:(13) 4:(133300) 5:(146850) 6:() 7:(89347839) 8:(JFK) 9:(DCA) 10:()		
EXTENDED VALUES:	1:(NEW YORK WASHINGTON, DC) 2:(NEW YORK) 3:(WASHINGTON, DC) 4:(REQUEST SUBMITTED; PENDING APPROVAL FROM THE CHICAGO HEADQUARTER) 5:(JFK:DCA&FF91827)	

History Data Detail Report

If specified in the Print History Data Detail Report dialog box, this report is generated when you run a reconciliation job. It shows the results of accumulating the extracted data with the source histories. This provides you with a picture of what the history database will contain if the history updates are completed with no errors. This report is used for debugging.

ACR/D Releasename		INFOGIX, INCORPORATED										COPYRIGHT INFOGIX, INC. 2008		
DATE: yy/mm/dd		HISTORY DATA DETAIL										REPORT ID: DXR001	PAGE 1	
TIME: 13:33:50		DETAIL RECONCILIATION SYSTEM										DATA BASE VERSION: 0000048/*****		
<--- JOB ID --->	<--- CYCLE RUN --->	KEYS ----->										FROM RECORD	<--- JOB/STEP -->	<--- FILE ID -->
MYJOB MYSTEP	00000004 000 101 002												MYJOB MYSTEP	
VALUES:	1:(55500) 2:(23450) 3:(32050) 4:(60095) 5:(28045)												MYJOB MYSTEP	
	6:(19034) 7:() 8:() 9:() 10:()												MYJOB MYSTEP	
	1:(JEWELRY												MYJOB MYSTEP	
<--- JOB ID --->	<--- CYCLE RUN --->	KEYS ----->										FROM RECORD	<--- JOB/STEP -->	<--- FILE ID -->
MYJOB MYSTEP	00000004 000 101 003												MYJOB MYSTEP	
VALUES:	1:(33345) 2:(100) 3:(2345) 4:(90010) 5:(87665)												MYJOB MYSTEP	
	6:(20075) 7:() 8:() 9:() 10:()												MYJOB MYSTEP	
	1:(CLOTHING (errors)												MYJOB MYSTEP	
<--- JOB ID --->	<--- CYCLE RUN --->	KEYS ----->										FROM RECORD	<--- JOB/STEP -->	<--- FILE ID -->
MYJOB MYSTEP	00000004 000 101 101												MYJOB MYSTEP	
VALUES:	1:(12345) 2:(10000) 3:(2345) 4:(10010) 5:(7665)												MYJOB MYSTEP	
	6:(10134) 7:() 8:() 9:() 10:()												MYJOB MYSTEP	
	1:(SPORTING GOODS												MYJOB MYSTEP	
<--- JOB ID --->	<--- CYCLE RUN --->	KEYS ----->										FROM RECORD	<--- JOB/STEP -->	<--- FILE ID -->
MYJOB MYSTEP	00000004 000 102 001												MYJOB MYSTEP	
VALUES:	1:(21338) 2:(17921) 3:(3417) 4:(76520) 5:(73103)												MYJOB MYSTEP	
	6:(98963) 7:() 8:() 9:() 10:()												MYJOB MYSTEP	
	1:(SPORTING GOODS												MYJOB MYSTEP	

Trace Report

The Trace Report (also called Extraction Trace Report) is useful in debugging because it provides a formatted list of the selection groups, details on processing of each record read, and notification of key related processes.

The following information is provided here:

- “Requesting the Report” on page 280
- “Report Sections with Examples” on page 281
- “Formatting of File Definitions” on page 283

Note: You can run another type of Trace Report that shows the processing of the table definitions against extracted values to build an external translation table. For information on that report, see [External Translation Table Trace Report on page 208](#).

Requesting the Report

This report prints when you run reconciliation if you request it as follows: In the Run Reconciliation dialog box, highlight each input source for which you want a report, and click **Edit**. When the File ID and File Name dialog box appears, select **Produce Trace Report**.

Note: You can run another type of Trace Report that shows the processing of the table definitions against extracted values to build an external translation table. For information on that report, see [External Translation Table Trace Report on page 208](#).

Report Sections with Examples

File Definition List

The first report section, a sample of which is shown below, is a formatted list of the selection groups as they were loaded in preparation for processing the input file.

Note: In this example, the source data is compared against a range of extended values. Because the SELECT FROM VALUE and SELECT TO VALUE are long, they are shown on separate lines after the message SEE EXTENDED SELECT VALUE BELOW.

FILE DEFINITION LIST: UGRPTF1					
SEQ. NO	SELECTION/EXTRACTION INFORMATION				LV OF
001001	SELECT(0001/02/TXT) DDF0000 TEST(EQUAL) VAL1(FF) VAL2()				L1 AND
001001	KEY (0081/03/TXT) DDF0000 KEY (K-01) XLATE(/)				
001001	KEY (0001/08/TXT) DDF0000 KEY (K-02) XLATE(/)				
001001	DTL (0097/08/TXT) DDF0000 TEXT INTO I-001 XLATE(/)				
001001	DTL (0090/03/NUM) DDF0000 COUNT INTO I-002 XLATE(/)				
001001	DTL (0042/08/NUM) DDF0000 ACCUM INTO I-003 XLATE(/)				
001001	DTL (0052/08/NUM) DDF0000 DATE INTO I-004 XLATE(/)				
001001	DTL (0061/08/NUM) DDF0000 TIME INTO I-005 XLATE(/)				
001001	DTL (0070/10/NUM) DDF0000 COUNT INTO I-007 XLATE(/)				
001001	DTL (0097/08/TXT) DDF0000 TEXT INTO V-001 XLATE(/)				
001001	DTL (V001/01/03) DDF0000 TEXT INTO I-008 XLATE(/)				
001001	DTL (V001/05/03) DDF0000 TEXT INTO I-009 XLATE(/)				
001001	DTL (0011/30/TXT) DDF0000 TEXT INTO X-001 XLATE(/)				
001001	DTL (0011/14/TXT) DDF0000 TEXT INTO X-002 XLATE(/)				
001001	DTL (0025/15/TXT) DDF0000 TEXT INTO X-003 XLATE(/)				
002001	RFT (0001/01/LIT) DDF0000 TO (0104/01/TXT) DDF0000 LIT(&)				
002001	RFT (0001/07/TXT) DDF0000 TO (0105/07/TXT) DDF0000 LIT()				
002001	DTL (0097/15/TXT) DDF0000 TEXT INTO X-005 XLATE(/)				
003001	SELECT(0011/08/TXT) DDF0000 TEST(NOT EQ.) VAL1(CHICAGO) VAL2()				L1 AND
003001	DTL (0001/80/TXT) DDF0000 TEXT INTO X-004 LIT(/)				

Processing for Each Record

The next report section shows the following for each record read:

- The record count and the first 132 characters of the record.
- For selection rules, the Source Value column shows the extracted value processed by the definition, and the Action Taken column shows the result of the evaluation (PASSED or FAILED).
- For selection definitions that passed, the processing of the associated key and detail field definitions.
- Notification of key related processes, including completion of a key, key breaks, partial keys dropped, and keys dropped because they have no associated detail data

B ■ Reports

Trace Report

Following is an example showing the processing that occurred when a record passed the evaluation:

```
INPUT RECORD: 00000001
0      1      2      3      4      5      6      7      8      9      A      B      C      D
---5---0---5---0---5---0---5---0---5---0---5---0---5---0---5---0---5---0---5---0---5---0---5---0---5---0---
FLIGHT  ORIGIN  DESTINATION  PRICE  DATE  TIME  TICKET#  OFFICE  STAFF#  ROUTE
-----
RECORD1ST
SEQ # NF.SELECTION/EXTRACTION INFORMATION                                     LV OP SOURCE VALUE  ACTION
-----
001001***SELECT(0001/02/TXT) DDF0000 TEST(EQUAL ) VAL1(FF ) VAL2( )          L1 ANDFL           FAILED
003001 SELECT(0011/08/TXT) DDF0000 TEST(NOT EQ.) VAL1(CHICAGO ) VAL2( )       L1 ANDORIGIN       PASSED
003001 DTL (0001/80/TXT) DDF0000 TEXT INTO X-004 LIT( / )                    ***SEE BELOW***
SOURCE VALUE: REQUEST SUBMITTED; PENDING APPROVAL FROM THE CHICAGO HEADQUARTER
INPUT RECORD: 00000002
0      1      2      3      4      5      6      7      8      9      A      B      C      D
---5---0---5---0---5---0---5---0---5---0---5---0---5---0---5---0---5---0---5---0---5---0---5---0---5---0---
-----
RECORD1ST
SEQ # NF.SELECTION/EXTRACTION INFORMATION                                     LV OP SOURCE VALUE  ACTION
-----
001001***SELECT(0001/02/TXT) DDF0000 TEST(EQUAL ) VAL1(FF ) VAL2( )          L1 AND--           FAILED
003001 SELECT(0011/08/TXT) DDF0000 TEST(NOT EQ.) VAL1(CHICAGO ) VAL2( )       L1 AND-----       PASSED
003001 DTL (0001/80/TXT) DDF0000 TEXT INTO X-004 LIT( / )                    ***SEE BELOW***
SOURCE VALUE: REQUEST SUBMITTED; PENDING APPROVAL FROM THE CHICAGO HEADQUARTER
```

The formatting of each definition type is shown in the next section.

Formatting of File Definitions

Selection Field Formats

Source	Target (to compare against)	Format
Input record or Extraction Variable	Constant or Range (Non-extended)	(Format when source is input record) SELECT (AAA/BB/CCC) DDFNNNN TEST(DDDD) VAL1(EEEE) VAL2(FFFF) (Format when source is extraction variable) SELECT (VNNN/B1/BB) DDFNNNN TEST(DDDD) VAL1(EEEE) VAL2(FFFF)
Input record or Extraction Variable	Constant or Range (Extended)	(Format when source is input record) SELECT (AAA/BB/CCC) DDFNNNN TEST(DDDD) ***SEE EXTENDED SELECT VALUE BELOW*** SELECT FROM VALUE (EEEE) SELECT TO VALUE (FFFF) (Format when source is extraction variable) SELECT (VNNN/B1/BB) DDFNNNN TEST(DDDD) ***SEE EXTENDED SELECT VALUE BELOW*** SELECT FROM VALUE (EEEE) SELECT TO VALUE (FFFF)
Input record	Extraction Variable	SELECT (AAA/BB/CCC) DDFNNNN TEST(DDDD) V-NNN
Extraction Variable	Input Record	SELECT(VNNN/B1/BB) DDFNNNN TEST(DDDD) AAA/BB/CCC

Where:

AAA is the field position of the source data from an input source file.

VNNN (with no hyphen) indicates that the source is an extraction variable. NNN is the extraction variable number.

B1 is the starting position of the source data

BB is the length of the source data.

CCC is the format of the source data (not applicable to extraction variables).

DDFNNNN is the 4-digit delimited data field number. When the source is not a delimited file, NNNN is 0000.

TEST is the hard coded literal 'TEST'.

DDDD (up to 8 Ds) is the selection operator.

EEEE is the constant or the start of the range to which the source value is being compared. If AAA is numeric, EEEE can be up to 16 bytes. If AAA is alphanumeric, EEEE can be up to 80 bytes.

FFFF is the end of the range to which the source value is being compared. This will be blanks if a range comparison is not specified. If AAA is numeric, EEEE can be up to 16 bytes. If AAA is alphanumeric, EEEE can be up to 80 bytes.

V-NNN (with a hyphen) indicates that the target value (to compare against) is an extraction variable. NNN is the extraction variable number.

B ■ Reports

Trace Report

Key Field Formats

Format Description	Format
Source is an input record, literal, or function	KEY (AAAA/BB/CCC) DDFNNNN KEY (K-DD) XLATE(FFFF/G)
Source is an extraction variable	KEY (VAAA/BB/CC) DDFNNNN KEY (K-DD) XLATE(FFFF/G)
Key is a cycle number	KEY (AAAA/BB/CCC) DDFNNNN CYC (DDDDDDDD) XLATE (FFFF/G)

Where:

AAAA is the field position of the source data from an input source file or literal.

VAAA is the field position of the source data in an extraction variable.

BB/CCC are the length/format of the source data.

DDFNNNN is the delimited data field number. When the source is not a delimited file, NNNN is 0000.

DD is the key number.

DDDDDDDD is the cycle number.

XLATE indicates that a translation table is defined for this field.

FFFF (up to 16 Fs) is the literal when the source is a literal.

G is blank if no translation is being used or Y if translation is being used.

Detail Field Formats

Source	Target	Format
Input record	Internal item	DTL (AAAA/BB/CCC) DDFNNNN DDDDD INTO I-NNN XLATE(FFFF/G)
Literal	Internal item	DTL (AAAA/BB/CCC) DDFNNNN DDDDD INTO I-NNN LIT(FFFF/G)
Extraction Variable	Internal item	DTL (VAAA/BB/CC) DDFNNNN DDDDD INTO I-NNN XLATE(FFFF/G)
Input record	Extraction variable	DTL (AAAA/BB/CCC) DDFNNNN DDDDD INTO V-NNN XLATE(FFFF/G)
Literal	Extraction variable	DTL (AAAA/BB/CCC) DDFNNNN DDDDD INTO V-NNN LIT(FFFF/G)
Extraction variable	Extraction variable	DTL (VAAA/BB/CC) DDFNNNN DDDDD INTO V-NNN XLATE(FFFF/G)

Where:

AAAA is the field position of the source data from an input record or literal.

VAAA is the field position of the source data in an extraction variable

BB/CCC are the length/format of the source data.

DDFNNNN is the delimited data field number. When the source is not a delimited file, NNNN is 0000.

NNN is the number of the internal item or extraction variable.

DDDDD is the type of detail field.

XLATE indicates that a translation table is being used.

LIT indicates that the source is a literal.

FFFF (up to 16 Fs) is the literal.

G is blank if no translation is being used or Y if translation is being used.

Reformat Field Formats

Source	Reformatted Area is	Format	
Input record or Literal	Output area	RFT (AAAA/BB/CCC) DDFNNNN TO (DDDD/EE/FFFF) DDFNNNN	LIT(GGGG)
Extraction variable	Output area	RFT (VAAA/BB/CC) DDFNNNN TO (DDDD/EE/FFFF) DDFNNNN	LIT(GGGG)
Input record or Literal	Note area	RFT (AAAA/BB/CCC) DDFNNNN TO (DDDD/EE/FFFF) DDF0000	NOTE LIT(GGGG)
Extraction variable	Note area	RFT (VAAA/BB/CC) DDFNNNN TO (DDDD/EE/FFFF) DDF0000	NOTE LIT(GGGG)

Where:

- AAAA is the beginning field position of the source data from an input record or literal.
- VAAA is the beginning field position of the source data in an extraction variable.
- BB is one of the following: 1) the length of the source data from an input record or literal, or 2) the beginning position of the source data from an extraction variable.
- CCC is the format of the source data from an input record or the hard coded 'LIT' if the source is a literal.
- CC is the length when the source data is from an extraction variable.
- DDFNNNN is the delimited data field number. When the source is not a delimited file, NNNN is 0000.
- DDDD is the beginning field position of the reformatted data in the output area or the note area.
- EE/FFFF are the length and format of reformatted field in the output area or the note area.
- NOTE indicates that the output area is the note area.
- GGGG (up to 16 Gs) is the literal if the source is a literal. Otherwise it is blanks.

Database Initialization Report

The Database Initialization Report lists the initialization transactions and any validation messages and presents a summary of the transactions.

```

ACR/D releasenumber                INFOGIX, INCORPORATED
DATE: yy/mm/dd                      DATA BASE INITIALIZATION REPORT
TIME: 15:36:06                       PAGE 001

-----INPUT TRANSACTION LISTING-----
TRANS                               TRANSACTION                               VALIDATION MESSAGES
NO.
-----1-----2-----3-----4-----5-----6-----7-----8-----
DFDSN  \progra-1\unitec-1\detail32\udsdf.dat
HFDSN  \progra-1\unitec-1\detail32\udshf.dat

-----DATA BASE INITIALIZATION SUMMARY-----
DEFINITION DSN      =  \progra-1\unitec-1\detail32\udsdf.dat
HISTORY DSN        =  \progra-1\unitec-1\detail32\udshf.dat

*****END OF REPORT*****

```

B ■ Reports

Detail History Analysis Reports

Detail History Analysis Reports

For more information on history analysis, see “Using History Analysis” on page 221. For and sample reports, see “History Analysis Report Samples” on page 226.

Detail Reconciliation Summary Report

This report is produced by the Print Management Report Utility. It generates management reports based on user-defined criteria.

Page 1 of 2

ACR/D releasenumbr	INFOGIX, INCORPORATED	PAGE 001
DATE: yy/mm/dd	INFOGIX DATA BASE UTILITY	
TIME: 11:16:07	DETAIL RECONCILIATION SUMMARY	DATA BASE VERSION: *****/0000008
SELECTION CRITERIA:		
PARAMETER	FROM	TO
-----	-----	-----
RETURN CODES	ALL	
HISTORY KEY 1 (FROM)	ALL	
HISTORY KEY 2 (TO)		
ABSOLUTE TIME (YY:JJJ:HH:MM:SS)	ALL	
RELATIVE TIME (DD:HH:MM)	ALL	
ABSOLUTE CYCLE-ID	ALL	
RELATIVE CYCLES	ALL	
CYCLE REFERENCE JOB ID	NONE	
DATA BASE:	HISTORY = \progra-1\unitec-1\detail32\udshf.dat	

Page 2 of 2

ACR/D releasenumbr	INFOGIX, INCORPORATED	PAGE 002							
DATE: yy/mm/dd	INFOGIX DATA BASE UTILITY								
TIME: 11:16:07	DETAIL RECONCILIATION SUMMARY	DATA BASE VERSION: *****/0000008							
HISTORY KEY	DEFINITION JOB ID	CYCLE	RUN	RETURN CODE	RUN DATE	RUN TIME	RERUNS	ADJ	RESULTS
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----
ABC	SAMPLE STEP1	00920612	000	0000	97/06/12	16:05:48	2		IN BALANCE
		00920611	000	0000	97/06/12	16:05:19	1		IN BALANCE
		00920610	000	0000	97/06/12	16:04:48	1		IN BALANCE
GHI	SAMPLE STEP1	00920612	000	0000	97/06/12	16:05:49	2		IN BALANCE
		00920611	000	0000	97/06/12	16:05:20	1		IN BALANCE
		00920610	000	0000	97/06/12	16:04:48	1		IN BALANCE
XYZ	SAMPLE STEP1	00920612	000	0000	97/06/12	16:05:49	2		IN BALANCE
		00920611	000	0000	97/06/12	16:05:20	1		IN BALANCE
		00920610	000	0000	97/06/12	16:04:49	1		IN BALANCE

Database History and Statistics Report

Generated by the List Definitions utility, this report shows detailed information about the database including date and time of initialization and last update and number of records by type.

ACR/D releasenumber DATE: yy/mm/dd TIME: 10:19:41	INFOGIX, INCORPORATED INFOGIX DATA BASE UTILITY DEFINITION DATA BASE LISTING DSN="C:\Infogix\Detail132\udsf.dat	COPYRIGHT INFOGIX, INC. PAGE 001 DATA BASE VERSION: 0000020/*****
DATA BASE HISTORY AND STATISTICS		
DATE AND TIME OF INITIALIZATION:	mm/dd/yyyy	AT 15:36:06
DATE AND TIME OF LAST UPDATE:	mm/dd/yyyy	AT 10:11:15
NUMBER OF BASIC INFORMATION RECORDS:		6
NUMBER OF INTERNAL ITEM RECORDS:		1
NUMBER OF EXTENDED INTERNAL ITEM RECORDS:		1
NUMBER OF BALANCING RULE RECORDS:		2
NUMBER OF SPECIAL INSTRUCTION RECORDS:		2
NUMBER OF FILE ITEM RECORDS:		2
NUMBER OF JOB COMMENT RECORDS:		3
NUMBER OF FILE COMMENT RECORDS:		2
NUMBER OF DETAIL USER REPORT BASIC RECORDS:		1
NUMBER OF DETAIL USER REPORT COLUMN RECORDS:		4
NUMBER OF DETAIL USER REPORT KEY BREAK RECORDS:		1
NUMBER OF FILE INFORMATION RECORDS:		2
NUMBER OF SELECTION FIELD RECORDS:		1
NUMBER OF KEY FIELD RECORDS:		2
NUMBER OF DETAIL FIELD RECORDS:		4
AGGREGATE RUN HISTORIES:		2

Index of Definition Keys Report

Generated by the List Definitions utility, this report lists definition keys by type.

ACR/D releasenumber DATE: yy/mm/dd TIME: 15:25:41	INFOGIX, INCORPORATED INFOGIX DATA BASE UTILITY DEFINITION DATA BASE LISTING DSN=R99UC43.DET47W.DEFN INDEX OF DEFINITION KEYS	COPYRIGHT INFOGIX, INC. 2014 PAGE 00 DATA BASE VERSION: 0000007/*****									
<table border="1"> <thead> <tr> <th>DEFINITION KEY</th> <th>TYPE</th> <th>COMMENTS</th> </tr> </thead> <tbody> <tr> <td>UGREPORTSAMPLE</td> <td>JOB ID</td> <td></td> </tr> <tr> <td>UGRPTF1 000000</td> <td>FILE-ID</td> <td></td> </tr> </tbody> </table>			DEFINITION KEY	TYPE	COMMENTS	UGREPORTSAMPLE	JOB ID		UGRPTF1 000000	FILE-ID	
DEFINITION KEY	TYPE	COMMENTS									
UGREPORTSAMPLE	JOB ID										
UGRPTF1 000000	FILE-ID										

B ■ Reports

Detailed Job Definitions Report

Detailed Job Definitions Report

Detailed Job Definitions Report - Sections A Through C

Generated by the List Definitions utility, this section of the Definition Database Listing is generated if you select Jobs or All as the type of definitions to list. This report lists the basic job information settings and item and report definitions for each job in the definition database.

ACR/D releasenumbr	INFOGIX, INCORPORATED	COPYRIGHT INFOGIX, INC.
DATE: yy/mm/dd	INFOGIX DATA BASE UTILITY	PAGE 003
TIME: 11:16:11	DEFINITION DATA BASE LISTING	DATA BASE VERSION: 000020/*****
	DSN=USR1.DET.SUSP.DEFN	
	DETAILED DEFINITIONS	
JOB ID: SAMLE JOB		
A. ALTERNATE JOB ID KEYS:		
NONE		
B. BASIC INFORMATION:		
JOB COMMENT INFORMATION:		
001 This is a job written to show reporting options. This was created durin		
002 g the 4.2 release for the purpose of reviewing the documentation. These		
003 are job level comments.		
PRODUCT:	ACR/D	
SYSTEM NAME:		
NUMBER OF HISTORIES RETAINED:	1	
LAST UPDATE BY:	SAMPUSR	DATE: yy/mm/dd TIME: 15:11:54
RUN OPTIONS:		
STORE RUN HISTORY:	YES	
REWRITE HISTORY:	NO	
PRINT NEW HISTORY:	NO	
PRINT ACR:	YES	
PRINT DRS RECAP:	NO	
PRINT EXTRACTED DATA:	NO	
SORTED OR ACCUMULATED DATA:	NO	
SET RETURN CODE:	NO	
SET RETURN CODE (NO EXTRACT):	NO	RETURN CODE: 4095
WRITE OUTPUT FILE:	NO	
TRUNCATE/ROUND DECIMALS:	ROUND	
RECON KEYS SORT ORDER:	ASCENDING	
DISPLAY PROCESSING STATUS	YES	
TALLY OOB KEYS	YES	RC: 1111 TOLERANCE IND: PCT. VALUE: 50.00
ACCUMULATE OOB VALUE:	YES	RC: 2222 TOLERANCE IND: PCT. VALUE: 60.00 ITEM: X-001
TALLY OOB RULES:	NO	
AUTO-CYCLE IND:	NONE	
REPORTING OPTIONS:		
PRINT 22 CHARACTER NUMERIC:	NONE	
INITIALIZE TEXT ITEM TO SPACE:	NONE	
CONTROL REPORT XML EXPORT OPT:DEFAULT TO USER OPTIONS		
RECONCILIATION LEVEL KEY LENGTH: 40	RTN CD RANGE FROM: TO	USE BASE DEFNS: USE FILTER: N CREATE FILTER: Y USE KEY RC: N
RECONCILIATION KEY MASK:	N/A	
C. INTERNAL ITEMS:		
ITEM	DESCRIPTION	L/R/C FORMAT DEC CI# REQD COM
I-001	AMOUNT VALUE	LEFT AMOUNT 2 NO
I-002	COUNT VALUE	LEFT COUNT 0 NO
I-003	DATE VALUE	LEFT DATE 0 NO

Detailed Job Definitions Report - Sections D Through L

ACR/D releasenumbr		INFOGIX, INCORPORATED		PAGE 002	
DATE: yy/mm/dd		INFOGIX DATA BASE UTILITY		DEFINITION DATA BASE LISTING	
TIME: 11:16:04		DSN=USRI.DET.SUSP.DEFN		DATA BASE VERSION: 0000032/*****	
		DETAILED DEFINITIONS			
EXTENDED INTERNAL ITEMS:					
ITEM	DESCRIPTION	FORMAT	DEC	CI#	REQD COM
X-001	DEPARTMENT NAME	TEXT	0		NO
D. EXTRACTION VARIABLES:					
ITEM	DESCRIPTION	DEC			
V-001	NO EXTRACT DEC PROCESSING, DEC=5	5			
V-002	EXTRACT DEC PROCESSING, DEC=6	6			
E. HISTORY ITEMS:					
----- HISTORY KEY -----		ADD MOST		----- VARIABLE CYCLE -----	
ITEM	KEY1 KEY2 KEY3 KEY4 KEY5	ITEM	CYC	RECENT	REQD IND
E-001	*****	I-001	-01	NO	NO
DESC: PRIOR MONTH GROSS PROFIT		FMT: AMT DEC: 2 I/O COL:		XP: 0	
F. CALCULATED ITEMS:					
ITEM	CALCULATION FORMULA	FORMAT	DEC	REQD	COM COND
C-001	ITEMS I-001 + ITEMS I-002	COUNT	0	NO	N
C-002	1 INTO @CYCLE	DATE	0	NO	N
G. BALANCING RULES:					
RULE	TYPE	BALANCING FORMULA	CODE	ACTION	FORMAT TOL.
R-001	ACT	ITEMS I-001 LE ITEMS I-002	1111	NONE	CNT 0 NONE
R-002	ACT	ITEMS I-002 LE ITEMS I-001	3333	NONE	CNT 0 NONE
H. SPECIAL INSTRUCTIONS:					
LINE	TYPE	INSTRUCTION TEXT/NOTIFY ADDRESS	CODE		
S-001	DIRECT	I-1 NOT LESS THAN I-2	1111		
S-002	DIRECT	I-2 NOT LESS THAN I-1	3333		
I. WRITE TO FREEFORM OUTPUT FILE:					
GENERATE HEADER:		YES			
GENERATE DELIMITER:		YES			
USE TAB AS DELIMITER:		NO			
DELIMITER CHARACTER:		&&			
HEADING WIDTH:		STANDARD ITEM SIZE			
FIELD	ITEM	OUTPUT	DATE	UNPACK	DECIMAL
NUMBER	ITEM	TYPE	NUMBER	POS	LEN
001	KEY		001	01	008
HEADING: CITY					
002	JOB/ID		000	00	000
HEADING: JOB ID					
005	KEY/RC		000	00	000
HEADING: KEY RC					
006	KEY/MSG		000	01	020
HEADING:					
007	INTERNAL		002	01	008
HEADING: CNT/LEAD0/SIGN					
J. FILE ITEMS:					
ITEM	DD NAME	NUMBER	KEYS REQUIRED		
1	CHARGE1		N/A		
K. FREEFORM REPORT BASIC INFORMATION:					
LOGICAL LINE WIDTH:		100			
PRINT RECONCILIATION KEYS:		Y (ALL)			
XML OPTION:		(1) DO NOT GENERATE			
USE BASE AND RLQ DEFINITIONS:		USE BASE AND RLQ DEF			
ALWAYS PRINT REPORT HEADINGS:		NO			
PAGE LENGTH:		0055			
PRINT WITH ASA CHARACTERS:		YES			
DYNAMIC TRANSLATION OVERRIDE:		OVERWRITE LINE 1 WITH ORIGINAL KEY 1 VALUE			
KEY BREAK	SIZE	PAGE BREAK	KEY BREAK	SIZE	PAGE BREAK
01	24	N	02	16	N
05	00	y	03	08	y
L. FREEFORM REPORT FORMAT ITEMS:					
FORMAT DATA					
<pre> "\N"[L010] "My Key Is"[L008] R001"\N"[L010]"Item 1 is"[L022] I001 "\N"[L010] "Item 2 is"[L020] I002 "\N"[L010] "item 3 is"[L020] I003 "\N"[L012] CY [L005] RC000"\N"[L080] S"\N"[L010] \$PAGE "\N"[L011] "Text Field"[L005] X001(001.010) {L001}" "{R005} X001(001.010) {L001}" "{C005} X001(001.010) {L001}" "{A005} X001(001.010)"\N"[L002] I002 {L001}" "{R002} I002 {L001}" "{C002} I002 {L001}" "{A002} I002 {L010} "\\" </pre>					

B ■ Reports

Detailed Job Definitions Report

Detailed Job Definitions Report - User Report Sections

This job generates a user report, so a User Report Definitions section appears at the end of the job definitions as shown below.

ACR/D releasenumbr		INFOGIX, INCORPORATED				COPYRIGHT INFOGIX, INC.				
DATE: yy/mm/dd		INFOGIX DATA BASE UTILITY				PAGE 003				
TIME: 11:16:11		DEFINITION DATA BASE LISTING				DATA BASE VERSION: 0000020/*****				
		DSN="C:\Infogix\Detail32\udsdf.dat								
		DETAILED USER REPORT DEFINITIONS								
JOB ID: SAMPLE JOB										
A. BASIC INFORMATION:										
REPORT TITLE:		SAMPLE REPORT								
PAGE WIDTH:		AUTO								
PAGE LENGTH:		AUTO								
REPORT OPTIONS:										
PRINT KEYS OPTION:		(Y) ALL								
ACCUM KEYS OPTION:		(N) NONE								
PRINT/TRUNCATE OPTION:		(P) PRINT								
AUTO POSITION OPTION:		(Y) YES								
PAGE ITEM TYPE:		(S) SPCL INST								
PAGE ITEM NUMBER:		N/A								
CASCADE KEYS OPTION:		(Y) YES								
ALWAYS PRINT HEADINGS OPTION:		(Y) YES								
XML OPTION:		(9) DEFAULT UOPT								
DEFINITIONS TO USE FOR RLQ:		(N) BASE AND RLQ								
B. PAGE HEADERS:										
COLUMN	COLUMN	ITEM	TYPE	ITEM#	WIDTH	POS.	LITERAL		HEADING	
									OPTION	
1	(I)	INTRNL	ITEM	2	10	10			CENTER	
2	(L)	LITERAL			20	25	Left-Justified Literal		LEFT	
3	(I)	INTRNL	ITEM	8	10	50			CENTER	
C. COLUMNS / ROW 1:										
COLUMN	COLUMN	ITEM	TYPE	ITEM#	WIDTH	POS.	(1/2)	HEADINGS	TOT	HEADING
									COL	OPTION
1	(K)	KEY	FIELD	1	AUTO	AUTO	(1)KEY IS		NO	CENTER
							(2)			
2	(I)	INTRNL	ITEM	1	5	AUTO	(1)AMOUNT ITEM		NO	CENTER
							(2)			
3	(I)	INTRNL	ITEM	2	AUTO	AUTO	(1)COUNT ITEM		NO	CENTER
							(2)			
4	(X)	EXTEND	ITEM	1	10	AUTO	(1)MY TEXT ITEM		NO	CENTER
							(2)			
D. KEY BREAK INFORMATION:										
KEY	KEY					LIT	LIT	SPACE	SPACE	
BREAK#	SIZE	KEY	BREAK	LITERAL		POS	LEN	BEFORE	AFTER	
1	3	ADD	THEM	UP	PLEASE	AUTO	AUTO	NEW	PAGE	1 LINE

Detailed File Definitions Report

Generated by the List Definitions utility, this section of the Definition Database Listing is generated if you select Files or All as the type of definitions to list. It provides detailed information on each file ID processed.

ACR/D releasenumber		INFOGIX, INCORPORATED				COPYRIGHT INFOGIX, INC. 2015													
DATE: yy/mm/dd		INFOGIX DATA BASE UTILITY				PAGE 001													
TIME: 10:02:30		DEFINITION DATA BASE LISTING				DATA BASE VERSION: 000020/*****													
		DSN=USR1.DET.SUSP.DEFN				DETAILED FILE DEFINITIONS													
FILE ID: UGRPTFL																			
FILE DESCRIPTION: FLYER																			
A. FILE INFORMATION:																			
ACCESS MODE:		ACCESS MODE 6																	
DATA SET ORGANIZATION:		SEQUENTIAL FILE/REPORT				RESET EXT VAR		DON'T DISPLAY SQL											
DD NAME FOR HARD COPY:																			
FILE MODEL - FILE ID:		BASE MODEL: NO																	
SET RETURN CODE (NO EXTRACT):		NO RETURN CODE:																	
FILE TRANSLATION TABLE:																			
LAST UPDATE BY:		R99UC47		DATE: 15/06/05		TIME: 13.51.56													
DDF INFORMATION:		USE DDF		FIXED DATA		DELIM 1		DELIM 2		DELIM 3		DELIM 4		DELIM 5		ENCLS			
REFERENCES:		NO																	
		JOB		STEP		QUAL													
		UGRREPORT SAMPLE																	
REC-ID: 001001	SELECTION TYPE	SEQ	POS	LEN	FORMAT	TYPE	EXT	INP	LVL	LOG	REL	SEL	EXT	VALUE	DDF				
	= TO VALUE 1	001	1	2	ALPHA		NO	INP	1	AND				FF					
	COMMENT- Includes FF fights only																		
	COMMENT-																		
KEY INFO:		001		81		3 ALPHA													
		COMMENT- Key 1 is office number																	
		COMMENT-																	
		002		1		8 ALPHA													
		COMMENT- Key 2 is flight number																	
		COMMENT-																	
DETAILS:		001		97		8 ALPHA		TEXT		NO		I-001		L DTL					
		002		90		3 NUMERIC		COUN		NO		I-002		L DTL					
		003		42		8 NUMERIC		SUM		NO		I-003		REPLACE					
		004		52		8 NUMERIC		DATE		NO		I-004		L DTL					
		005		61		8 NUMERIC		TIME		NO		I-005		L DTL		TM FMT:HHMMSS			
		006		70		10 NUMERIC		COUN		NO		I-007		L DTL					
		007		97		8 ALPHA		TEXT		NO		EV		001		L DTL			
		008		1		3 ALPHA		TEXT		NO		EV		1		I-008		L DTL	
		009		5		3 ALPHA		TEXT		NO		EV		1		I-009		L DTL	
		010		81		3 ALPHA		TEXT		NO						I-010		L DTL	
		011		81		3 NUMERIC		TEXT		NO						I-011		L DTL	
		COMMENT- Get office city																	
		COMMENT-																	
		012		1		1 NUMERIC		SUM		NO		L				I-012		L DTL	
		COMMENT- Tally counter for records																	
		COMMENT-																	

Detailed Table Definitions: Internal Translation Table

Generated by the List Definitions utility, this section of the Definition Database Listing is generated if you select **Internal/Cycle Tables** in the List Definitions dialog box. This example shows a report on an internal translation table.

Detailed Table Definitions: External Translation Table

Generated by the List Definitions utility, this section of the Definition Database Listing is generated if you select External Tables in the List Definitions dialog box. The example shows a report on an external translation table:

```

ACR/D releasenumber
DATE: yy/mm/dd
TIME: 13:36:15
                                XYZ COMPANY
                                INFOGIX DATA BASE UTILITY
                                DEFINITION DATA BASE LISTING
                                DSN=\progra-1\unitec-1\detail32\udsdf.dat
                                DETAILED TABLE DEFINITIONS
                                COPYRIGHT INFOGIX, INC
                                PAGE 001
                                DATA BASE VERSION: 0000078/*****

TABLE-ID: SAMPLE
A. TABLE INFORMATION:
TABLE DESCRIPTION:          TRANSLATE LONG NUMBER TO SHORT NUMBER
BUILD OPTION:              FIRST LOOKUP
STORAGE OPTION:           ACCESS MEMORY
DATA SET ORGANIZATION:    SEQUENTIAL
INPUT FILE DDNAME:       LONGNUM
LAST UPDATE BY:          SAMPUSR
DDF INFORMATION:         USE DDF  FIXED DATA  DELIM 1  DELIM 2  DELIM 3  DELIM 4  DELIM 5  ENCLS
                                DATE: yy/mm/dd    TIME: 15:11:54
                                NO
B. I/O PARAMETERS:
PARAMETER                  PARAMETER NAME          REFERENCE  FORMAT  DEC  LEN          PARAMETER DESCRIPTION
-----
1 LONG NUMBER TO SHORT NUMBER  FEXT          COUNT      0      0  30-DIGIT NUMBER TO 8-DIGIT NUMBER
C. TABLE COLUMNS:
COLUMN                     COLUMN NAME          FORMAT  DEC  LEN  ORDER          COLUMN DESCRIPTION
-----
1 SHORT NUMBER             COUNT              0      0      1  8-DIGIT
2 LONG NUMBER              COUNT              0      0      0  30-DIGIT
D. LOOKUP RULES:
RULE  RULE TYPE  LH OPERAND  RH OPERAND  SELECTION TYPE  LITERAL/ACTION  TERM IND  NEXT TABLE
-----
1 SELECTION: P001      C002
2 ASSIGN: C001        P001
                EQUAL TO
F. TABLE BUILD RULES:
SELECTION CRITERIA:  RULE  POS  LEN  DDF  FORMAT  LH OPR  SELECTION TYP  RH OPR  EXT BEGIN  LOCATE  LITERAL
-----
1 10 12  NUMERIC  INP  EQUAL TO  LIT  NO NO NO  111222333444
2 10 31  NUMERIC  INP  EQUAL TO  LIT  YES NO NO  222333444555
3 10 31  NUMERIC  COUNT  XW001  NO NO NO  333444555666
4 10 31  NUMERIC  COUNT  XW002  NO NO NO  LITERAL
5 10 12  NUMERIC  INP  GRT THAN EQ  LIT  NO NO NO  111222333444
6 10 31  NUMERIC  INP  GREATER THAN  LIT  YES NO NO  ***SEE BELOW***
7 1 16  PACKED  XW001  EQUAL TO  XW002  YES NO NO  ***SEE BELOW***
8 1 16  PACKED  XW001  GREATER THAN  LIT  YES NO NO  ***SEE BELOW***
9 10 6  NUMERIC  COUNT  C001
10 10 31  NUMERIC  COUNT  C002

```

B ■ Reports

Detailed Exception Definitions (History Analysis)

Detailed Exception Definitions (History Analysis)

Generated by the List Definitions utility, this section of the Definition Database Listing is generated if the List Type selected is (History Analysis) Processes.

ACR/D releasenumbr DATE: yy/mm/dd TIME: 10:19:41	INFOGIX, INCORPORATED INFOGIX DATA BASE UTILITY DEFINITION DATA BASE LISTING DSN=USERIDA.TEST.DETDEFN	COPYRIGHT INFOGIX, INC. PAGE 008 DATA BASE VERSION: 0000018/*****																
DETAILED EXCEPTION DEFINITIONS																		
EXCEPTION RPT ID: SAMPLERREPORT LAST UPDATE BY: SAMPUSR DATE: yy/mm/dd TIME: 15:11:54																		
1. EXCEPTION REPORT INFORMATION.																		
MAX CYCLES	MAX ITEMS	MISSING CYCLE	DIVISION OPT	SET BY	RTN ZERO	CODE	CYCLE	TABLE	NAME	TYPE	----- HISTORY KEY	----- KEY1	----- KEY2	----- KEY3	----- KEY4	----- KEY5	CYCLE OPTION	RELATIVE CYCLE
3	5	YES	YES	NO				SAMPLECYCLES									GREGORIAN DATE	3
2. EXCEPTION REPORT KEY INFORMATION.																		
KEY LEN	KEY SEQ	-----REPORT IDS -----																
3	01	SAMPLERPT1																
3	01	SAMPLERPT2																

Detailed Report Definitions - History Analysis Threshold and Fluctuation Analysis

Generated by the List Definitions utility, this section of the Definition Database Listing is generated if the List Type selected is (History Analysis) Reports. The examples below show details regarding the two types of reports you can generate.

Detailed Definitions Report (History Analysis - Threshold Analysis)

ACR/D releasenumbr DATE: yy/mm/dd TIME: 10:19:41	INFOGIX, INCORPORATED INFOGIX DATA BASE UTILITY DEFINITION DATA BASE LISTING DSN=USERIDA.TEST.DETDEFN	COPYRIGHT INFOGIX, INC. PAGE 009 DATA BASE VERSION: 0000018/*****												
DETAILED REPORT DEFINITIONS														
REPORT ID: SAMPLERPT1														
REPORT TITLE: SAMPLE EXCEPTION REPORT														
COLUMNS PER PAGE: 6														
REPORT TYPE: THRESHOLD ANALYSIS														
REPORT PRINT OPTION: YES														
TRUNCATE/ROUND DECIMALS: TRUNCATE														
REPORT MISSING CYCLES: NO														
LAST UPDATE BY: SAMPUSR														
DATE: yy/mm/dd														
TIME: 15:11:54														
A. ITEM INFORMATION:														
FORMAT	DEC	RULE	DESCRIPTION											
COUNT	000		ITEM #1											
COUNT	000		ITEM #2											
COUNT	000		ITEM #3											
COUNT	000		ITEM #4											
B. HEADING INFORMATION:														
LINE	FORMAT	POS	LEN	TYPE	TEXT									
1		1	18		RECONCILIATION KEY									
1		25	20		ITEM DESCRIPTION									
C. HISTORY KEY INFORMATION:														
----- FIELD ----- KEY -----														
LINE	FORMAT	POS	LEN	TYPE	FORMAT	POS	LEN	TYPE	SKIP					
1		1	20		1	3			1 LINE					
1		25	20		1	20			NO LINES					
D. COLUMN INFORMATION:														
PRINT	BEG	END	TYPE	FORMAT	DEC	SEQ	CONSTANT	FMT	PCT	PCT	DEC	HEAD1	HEAD2	CYCLE FORMAT
YES	0	0		C		001	0							CCYYMMDD
YES	0	0		C		002	0							MMDDYY
YES	1	2	PCT	C		003	0	Y				CNT (DEC-0)	PERCENT	
YES	1	2	PCT	C		004	0	N		5		CNT (DEC-4)	PERCENT	
YES	1	2	PCT	C		005	0	Y				AMT (DEC-0)	PREVIOUS	
YES	1	2	PCT	C		006	0	N		2		AMT (DEC-2)	PERCENT	

B ■ Reports

Database History and Statistics Report

Detailed Definitions Report (History Analysis - Fluctuation Analysis)

ACR/D releasenumbr		INFOGIX, INCORPORATED		COPYRIGHT INFOGIX, INC.											
DATE: yy/mm/dd		INFOGIX DATA BASE UTILITY		PAGE 011											
TIME: 10:19:41		DEFINITION DATA BASE LISTING		DATA BASE VERSION: 0000018/*****											
DSN=\progra-1\unitec-1\detail32\udsf.dat															
DETAILED REPORT DEFINITIONS															
REPORT ID: SAMPLERPT2															
REPORT TITLE: SAMPLE SUMMARY LEVEL REPORT															
COLUMNS PER PAGE: 6															
REPORT TYPE: FLUCTUATION ANALYSIS															
REPORT PRINT OPTION: YES															
TRUNCATE/ROUND DECIMALS: NO															
REPORT MISSING CYCLES: NO															
A. ITEM INFORMATION:															
FORMAT	DEC	RULE	DESCRIPTION												
COUNT	000		ITEM #1												
COUNT	000		ITEM #2												
COUNT	000		ITEM #3												
COUNT	000		ITEM #4												
B. HEADING INFORMATION:															
LINE	FORMAT	POS	LEN	TYPE	TEXT										
1		1	18		RECONCILIATION KEY										
C. HISTORY KEY INFORMATION:															
----- FIELD ----- KEY -----															
LINE	FORMAT	POS	LEN	TYPE	FORMAT	POS	LEN	TYPE	SKIP						
1		1	20			1	20		1 LINE						
D. COLUMN INFORMATION:															
PRINT	BEG	END	TYPE	FORMAT	DEC	SEQ	CONSTANT	FMT	PCT	PCT	DEC	HEAD1	HEAD2	CYCLE	FORMAT
YES	0	0		C		001	0								CCYYMMDD
YES	0	0		C		002	0								MMDDYY
YES	1	2	PCT	C		003	0	Y				CNT(DEC-0)	PERCENT		
YES	1	2	PCT	C		004	0	N			5	CNT(DEC-4)	PERCENT		
YES	1	2	PCT	C		005	0	Y				AMT(DEC-0)	PREVIOUS		
YES	1	2	PCT	C		006	0	N			2	AMT(DEC-2)	PERCENT		

Database History and Statistics Report

Generated by the List History utility, this section of the List Definitions Report is included depending on your specification on the List History dialog box.

ACR/D releasenumbr		INFOGIX, INCORPORATED		PAGE 008	
DATE: yy/mm/dd		INFOGIX DATA BASE UTILITY		DATA BASE VERSION: *****/0000002	
TIME: 16:11:53		HISTORY DATA BASE LISTING			
DSN=\progra-1\unitec-1\detail32\udshf.dat					
DATA BASE HISTORY AND STATISTICS					

DATE AND TIME OF INITIALIZATION:		yy/mm/dd	AT 15:50:53		
DATE AND TIME OF LAST UPDATE:		yy/mm/dd	AT 16:11:31		
ESTIMATED NUMBER OF UNIQUE JOB IDS:		0			
ESTIMATED NUMBER OF HISTORY KEYS:		0			
AVERAGE NUMBER OF CYCLES PER KEY:		0			
AVERAGE NUMBER OF ITEMS PER KEY:		0			
NUMBER OF HISTORY KEYS:		290			
NUMBER OF DETAIL HISTORIES:		290			

History Data Detail Report

Generated by the List History utility, this section of the History Data Base Listing Report is produced when Detail is included in the Report Type. It shows every history record and the values stored for it.

ACR/D Releasenumbr	INFOGIX, INCORPORATED	COPYRIGHT INFOGIX, INC. 2008
DATE: yy/mm/dd	HISTORY DATA DETAIL	REPORT ID: UT0502 PAGE 1
TIME: 14:02:06	HISTORY DATA BASE LISTING	DATA BASE VERSION: 0000001/*****
<--- JOB ID --->	CYCLE RUN <----- KEYS ----->	FROM RECORD <--- JOB/STEP ---> <-FILE ID->
MYJOB MYSTEP	00000003 000 101 002	1 MYJOB MYSTEP UNIHF
VALUES: 1:(RETURN CODE: 0222 PROG NAME: UDS2000 REC STAT:	
6:(2:(23450) 3:(32050) 4:(60095) 5:(28045)	
EXTENDED VALUES:	1:(JEWELRY)
<--- JOB ID --->	CYCLE RUN <----- KEYS ----->	FROM RECORD <--- JOB/STEP ---> <-FILE ID->
MYJOB MYSTEP	00000003 000 101 003	2 MYJOB MYSTEP UNIHF
VALUES: 1:(RETURN CODE: 0222 PROG NAME: UDS2000 REC STAT:	
6:(2:(33345) 2:(100) 3:(2345) 4:(90010) 5:(87665)	
EXTENDED VALUES:	1:(CLOTHING (errors))
<--- JOB ID --->	CYCLE RUN <----- KEYS ----->	FROM RECORD <--- JOB/STEP ---> <-FILE ID->
MYJOB MYSTEP	00000003 000 101 101	3 MYJOB MYSTEP UNIHF
VALUES: 1:(RETURN CODE: 0222 PROG NAME: UDS2000 REC STAT:	
6:(2:(12345) 2:(10000) 3:(2345) 4:(10010) 5:(7665)	
EXTENDED VALUES:	1:(SPORTING GOODS)
<--- JOB ID --->	CYCLE RUN <----- KEYS ----->	FROM RECORD <--- JOB/STEP ---> <-FILE ID->
MYJOB MYSTEP	00000003 000 102 001	4 MYJOB MYSTEP UNIHF
VALUES: 1:(RETURN CODE: 0222 PROG NAME: UDS2000 REC STAT:	
6:(2:(21338) 2:(17921) 3:(3417) 4:(76520) 5:(73103)	
EXTENDED VALUES:	1:(SPORTING GOODS)
<--- JOB ID --->	CYCLE RUN <----- KEYS ----->	FROM RECORD <--- JOB/STEP ---> <-FILE ID->
MYJOB MYSTEP	00000003 000 102 002	5 MYJOB MYSTEP UNIHF
VALUES: 1:(RETURN CODE: 0222 PROG NAME: UDS2000 REC STAT:	
6:(2:(89051) 2:(62493) 3:(26558) 4:(64268) 5:(37710)	
EXTENDED VALUES:	1:(JEWELRY)
<--- JOB ID --->	CYCLE RUN <----- KEYS ----->	FROM RECORD <--- JOB/STEP ---> <-FILE ID->
MYJOB MYSTEP	00000003 000 103 001	6 MYJOB MYSTEP UNIHF
VALUES: 1:(RETURN CODE: 0222 PROG NAME: UDS2000 REC STAT:	
6:(2:(78962) 2:(10000) 3:(68962) 4:(110010) 5:(41048)	
EXTENDED VALUES:	1:(SPORTING GOODS)
<--- JOB ID --->	CYCLE RUN <----- KEYS ----->	FROM RECORD <--- JOB/STEP ---> <-FILE ID->
MYJOB MYSTEP	00000003 000 103 002	7 MYJOB MYSTEP UNIHF
VALUES: 1:(RETURN CODE: 0222 PROG NAME: UDS2000 REC STAT:	
6:(2:(58702) 2:(50843) 3:(7859) 4:(12595) 5:(4736)	
EXTENDED VALUES:	1:(JEWELRY)

Control Update Information Report

Generated by the Print Control Update Information Report utility, this report lists the most recent updates to definitions from the definition database, including the ID associated with the change and the date and time of the change in yy/mm/dd format.

FROM DATE: yy/mm/dd	TO DATE: yy/mm/dd	LAST UPDATE BY	DATE	TIME
ABC2A1L STEP1	ALTERNATE JOB (ABC2 STEP1)			
ABC2 STEP1	JOB	UserID	yy/mm/dd	12:56:33
XSELECT1TEXTFQ	JOB			
XSELECT2TEXTFNG	JOB			
XSELECT3TEXTGE	JOB			
XSELECT4NUMNOTEQ	JOB			
XSELECT5NUMNOTEQ	USER REPORT	UserID	yy/mm/dd	12:56:33
XSELECT5NUMOORNG	JOB			
XSEL1F1 000000	FILE			
XSEL2F1 000000	FILE			
XSEL3F1 000000	FILE			
XSEL4F1 000000	FILE			
XSEL5F1 000000	FILE			
ABC	EXT TRAN TABLE			
CYCLETABLE1	CYCLE TABLE			
SAMPLE	EXT TRAN TABLE			
SAMPLE1DUMMY	INT TRAN TABLE	UserID	yy/mm/dd	15:14:45
SAMPLE1DUMMY	CYCLE TABLE	UserID	yy/mm/dd	16:48:40
SAMPLE2DUMMY	CYCLE TABLE	UserID	yy/mm/dd	15:35:08
SAMPLE	EXT TRAN TABLE			
TS97IT2	INT TRAN TABLE			
HA1 SAMPLE	HA PROCESS	UserID	yy/mm/dd	10:36:02
HA1 STEP1	HA PROCESS			
VOLANALFD1	HA PROCESS			
SAMPOREPORT	HA REPORT	UserID	yy/mm/dd	10:36:02
VOLANALAR1	HA REPORT			

Definition Database Update Report

This report is generated when you run a Database Update. It consists of 2 sections. Each section contains a numbered list of the transactions processed, sorted by transaction key. The first section shows any validation messages generated for each transaction. The second section lists the processing that took place related to each transaction, including any records purged, added, deleted, or replaced.

Definition Database Update Report

ACR/D Releasenameumber		INFOGIX, INCORPORATED				COPYRIGHT INFOGIX, INC. 2008	
DATE: yy/mm/dd		DEFINITION DATA BASE UPDATE REPORT				PAGE 001	
TIME: 13:45:58		DSN=\progra-1\unitec-1\detail32\udsdf.dat				DATA BASE VERSION: 0000048/*****	
----- TRANSACTION VALIDATION -----							
TRANS NO.	TRANSACTION						VALIDATION MESSAGES
1	2	3	4	5	6	7	8
00001	TSTJB	000000	F0	0001*			
00002	TSTJB	000000	F0	0001A6PSA00000	0000000	Y	
00003	TSTJB	000000	F0	0002A			
00004	TSTJB	001001	F4	0011A0000062T	=STORE: 000 I11	Y	
00005	TSTJB	001001	F4	0012A00004			
00006	TSTJB	001001	F6	0011A0000032K	001 000 00012		
00007	TSTJB	002001	F4	0011A0000052T	=DEPT: 000 I11	Y	
00008	TSTJB	002001	F4	0012A00006			
00009	TSTJB	002001	F6	0011A0000032K	002 000 00012		
00010	TSTJB	002001	F7	0011A0000232T	001 L 000 Y00020		
00011	TSTJB	003001	F4	0011A0000082T	=STARTING 000 I11	Y	
00012	TSTJB	003001	F4	0012A00016			
00013	TSTJB	003001	F7	0011A0000091+	001 R 000 N00053		
00014	TSTJB	004001	F4	0011A0000072T	=ENDING 000 I11	Y	
00015	TSTJB	004001	F4	0012A00016			
00016	TSTJB	004001	F7	0011A0000071+	002 R 000 N00054		
00017	TSTJB	005001	F4	0011A0000072T	=COST OF 000 I11	Y	
00018	TSTJB	005001	F4	0012A00016			
00019	TSTJB	005001	F7	0011A0000071+	003 R 000 N00054		
00020	TSTJB	006001	F4	0011A0000062T	=GROSS 000 I11	Y	
00021	TSTJB	006001	F4	0012A00016			
00022	TSTJB	006001	F7	0011A0000061+	005 R 000 N00063		
00023	TSTJB	007001	F4	0011A0000062T	= SALES 000 I11	Y	
00024	TSTJB	007001	F4	0012A00015			
00025	TSTJB	007001	F7	0011A0000071+	004 R 000 N00054		
00026	TSTJB	008001	F4	0011A0000052T	=PRIOR 000 I11	Y	
00027	TSTJB	008001	F4	0012A00016			
00028	TSTJB	008001	F7	0011A0000081+	006 R 000 N00054		

ACR/D Releasenameumber		INFOGIX, INCORPORATED				COPYRIGHT INFOGIX, INC. 2008		
DATE: yy/mm/dd		DEFINITION DATA BASE UPDATE REPORT				PAGE 002		
TIME: 13:45:58		DSN=\progra-1\unitec-1\detail32\udsdf.dat				DATA BASE VERSION: 0000048/*****		
----- TRANSACTION PROCESSING -----								
TRANS NO.	TRANSACTION						RECORD KEY	ACTION
1	2	3	4	5	6	7	8	
00001	MYFILE	000000	F0	0001*				
						MYFILE	000000 F0 000 PURGED	
						MYFILE	001001 F4 001 PURGED	
						MYFILE	001001 F6 001 PURGED	
						MYFILE	002001 F4 001 PURGED	
						MYFILE	002001 F6 001 PURGED	
						MYFILE	002001 F7 001 PURGED	
						MYFILE	003001 F4 001 PURGED	
						MYFILE	003001 F7 001 PURGED	
						MYFILE	004001 F4 001 PURGED	
						MYFILE	004001 F7 001 PURGED	
						MYFILE	005001 F4 001 PURGED	
						MYFILE	005001 F7 001 PURGED	
						MYFILE	006001 F4 001 PURGED	
						MYFILE	006001 F7 001 PURGED	
						MYFILE	007001 F4 001 PURGED	
						MYFILE	007001 F7 001 PURGED	
						MYFILE	008001 F4 001 PURGED	
						MYFILE	008001 F7 001 PURGED	
00002	MYFILE	000000	F0	0001A6PSA00000	0000000	Y		
00003	MYFILE	000000	F0	0002A				
00005	MYFILE	001001	F4	0012A000004				
00006	MYFILE	001001	F6	0011A0000032K	001 000 00012			
00008	MYFILE	002001	F4	0012A00006				
00009	MYFILE	002001	F6	0011A0000032K	002 000 00012			
00010	MYFILE	002001	F7	0011A0000232T	001 L 000 Y00020			
00012	MYFILE	003001	F4	0012A00016				
00013	MYFILE	003001	F7	0011A0000091+	001 R 000 N00053			
00015	MYFILE	004001	F4	0012A00016				
00016	MYFILE	004001	F7	0011A0000071+	002 R 000 N00054			
00018	MYFILE	005001	F4	0012A00016				
00019	MYFILE	005001	F7	0011A0000071+	003 R 000 N00054			
00021	MYFILE	006001	F4	0012A00016				
00022	MYFILE	006001	F7	0011A0000061+	005 R 000 N00063			
00024	MYFILE	007001	F4	0012A00015				
00025	MYFILE	007001	F7	0011A0000071+	004 R 000 N00054			
00027	MYFILE	008001	F4	0012A00016				
00028	MYFILE	008001	F7	0011A0000081+	006 R 000 N00054			

Definition Database Cross Reference Report

This report lists all the file definitions used within the first database, and if specified, the second database. The report allows you to find job descriptions that use the same file definitions.

The report displays the following information:

- File ID
- File description
- Database name
- Job names that reference the file ID

The report uses an indicator (#) to show which file ID is referenced by an alternate job.

Example

0 FILE ID	DESCRIPTION / REFERENCES	TYPE: PS
OFILE1A	DESCRIPTION: FILE 1 A JOB1 FILEREF DATABASE: USER01.SUM.DEFN JOB1R2 FILEREF /JOB1 FILEREF #	TYPE: PS
OFILE1A	DESCRIPTION: FILE 1 A JOB1 FILEREF DATABASE: USER02.SUM.DEFN	TYPE: PS
OFILE2A	DESCRIPTION: FILE 2 A JOB2 FILEREF DATABASE: USER01.SUM.DEFN	TYPE: PS
# - ALTERNATE JOB/PRIMARY JOB		
*****END OF REPORT*****		

B ■ Reports

Build ACRUPDT Report

Build ACRUPDT Report

This report is generated by running the Build Batch Transactions utility. It shows each transaction included in the database update. For an explanation of the record structure, see “Format of Batch Definition Transaction Records” on page 262.

ACR/D releasenumbr		INFOGIX, INCORPORATED			COPYRIGHT INFOGIX, INC.				
DATE: yy/mm/dd		INFOGIX DATA BASE UTILITY			PAGE 001				
TIME: 12:47:20		BUILD ACRUPDT REPORT			DATA BASE VERSION: 0000002/*****				
		DSN=\progra-1\unitec-1\detail32\udsdf.dat							
TRANS NO.	RECORD TYPE	T R A N S A C T I O N							
		-1-	-2-	-3-	-4-	-5-	-6-	-7-	-8-
00001	JOB ID	SAMPLE STEPI	J0 0001*						
00002		SAMPLE STEPI	J0 0001A	FIELD EXPANSION TESTING			005YYNNNNNNNDS		
00003		SAMPLE STEPI	J0 0003A				0		
00004		SAMPLE STEPI	J0 0009A	Y					
00005		SAMPLE STEPI	J2 001LAREGULAR DATE (PICK FROM MMDDCCYY)						
00006		SAMPLE STEPI	J2 0012A				LD0000N		
00007		SAMPLE STEPI	J2 002LAREGULAR DATE (PICK FROM MMDDCCYY)						
00008		SAMPLE STEPI	J2 0022A				LD0000N		
00009		SAMPLE STEPI	J2 999LAINI-999						
00010		SAMPLE STEPI	J2 9992A				LT0000N		
00011		SAMPLE STEPI	J22001LAEEXTENDED COUNT						
00012		SAMPLE STEPI	J220012A				C0000N		
00013		SAMPLE STEPI	J22002LAEEXTENDED AMOUNT						
00014		SAMPLE STEPI	J220022A				A2000N		
00015		SAMPLE STEPI	J22003LAEEXTENDED DATE (PICK FROM MMDDCCYY)						
00016		SAMPLE STEPI	J220032A				D0000N		
00017		SAMPLE STEPI	J22004LAEEXTENDED DATE (PICK FROM MMDDCCYY)						
00018		SAMPLE STEPI	J220042A				D0000N		
00019		SAMPLE STEPI	J22005LAEEXTENDED TEXT						
00020		SAMPLE STEPI	J220052A				T0000N		
00021		SAMPLE STEPI	J22006LAEEXTENDED COUNT (=1)						
00022		SAMPLE STEPI	J220062A				C0000N		
00023		SAMPLE STEPI	J22007LAEEXTENDED AMOUNT (=1.23)						
00024		SAMPLE STEPI	J220072A				A1000N		
00025		SAMPLE STEPI	J3 0011A*****+00N001YNN						
00026		SAMPLE STEPI	J3 0012A						
00027		SAMPLE STEPI	J3 0013A000000000						
00028		SAMPLE STEPI	J3 0021A*****+00N002YNN						
00029		SAMPLE STEPI	J3 0022A						
00030		SAMPLE STEPI	J3 0023A000000000						
00031		SAMPLE STEPI	J3 0031A*****+00N001YNY						
00032		SAMPLE STEPI	J3 0032A						
00033		SAMPLE STEPI	J3 0033A000000000						
00034		SAMPLE STEPI	J3 0041A*****+00N002YNY						
00035		SAMPLE STEPI	J3 0042A						
00036		SAMPLE STEPI	J3 0043A000000000						
00037		SAMPLE STEPI	J3 0051A*****+00N003YNY						
00038		SAMPLE STEPI	J3 0052A						
00039		SAMPLE STEPI	J3 0053A000000000						
00040		SAMPLE STEPI	J3 0061A*****+00N004YNY						
00041		SAMPLE STEPI	J3 0062A						
00042		SAMPLE STEPI	J3 0063A000000000						
00043		SAMPLE STEPI	J3 0071A*****+00N005YNY						
00044		SAMPLE STEPI	J3 0072A						
00045		SAMPLE STEPI	J3 0073A000000000						
00046		SAMPLE STEPI	J31001LAREGULAR DATE (PICK FROM MMDDCCYY)				D0L		
00047		SAMPLE STEPI	J31002LAREGULAR DATE (PICK FROM MMDDCCYY)				C0L		
00048		SAMPLE STEPI	J31003LAEEXTENDED COUNT				C0L		
00049		SAMPLE STEPI	J31004LAEEXTENDED AMOUNT				A2L		

Translation Database Listing Report

This report is produced by the List Translation Database utility. It reports on the contents of the dynamic translation database. It lists each dynamic translation table along with the Identifier (the key after translation and the Original Source Value (the key before translation).

Translation Database Listing Report 1 of 2

ACR/D releasenumber	INFOGIX, INCORPORATED	PAGE	1
DATE: yy/mm/dd	INFOGIX DATA BASE UTILITY		
TIME: 08:34:07	UNITF DATA BASE LISTING		

UNITF TRANSLATION TABLE CONTROL RECORD			

TRANSLATION TABLE NAME: SAMPLETABLE			
NEXT AVAILABLE TRAN ID: 00000004			
CREATION DATE: 08/02/08			
CREATION TIME: 08:31:41			
LAST UPDATE DATE: 08/02/08			
LAST UPDATE TIME: 08:31:41			

TABLE NAME	IDENTIFIER	ORIGINAL SOURCE VALUE	

SAMPLETABLE	00000001	ABC	
SAMPLETABLE	00000002	GHI	
SAMPLETABLE	00000003	XYZ	

Translation Database Listing Report 2 of 2

ACR/D releasenumber	INFOGIX, INCORPORATED	PAGE	2
DATE: yy/mm/dd	INFOGIX DATA BASE UTILITY		
TIME: 08:34:07	UNITF DATA BASE LISTING		

UNITF DATA BASE SUMMARY INFORMATION			

TOTAL TRANSLATION TABLES: 001			
TOTAL TRANSLATION IDENTIFIERS: 00000003			
UNITF DATA BASE CREATE DATE: 08/02/08			
UNITF DATA BASE CREATE TIME: 08:24:53			

Suspense Aging Report

Generated by the Print Suspense Aging utility, this report summarizes the contents of history (for all keys or for some level of the key) in the suspense database in terms of the age of specified items.

B ■ Reports

History Database Transfer Report

Suspense Aging Report Page 1 of 2

ACR/D Releasenumbr	INFOGIX, INCORPORATED	COPYRIGHT INFOGIX, INC.
DATE: yy/mm/dd	ACR/DETAIL HISTORY AGING REPORT(132)	PAGE: 1
TIME: 10:11:29	SUSPENSE AGING REPORT	

SUSPENSE AGING REPORT CRITERIA:

```

HISTORY FILE ..... "C:\ACR\Detail-Summary\data\Udshf.dat"
FROM KEY ..... START OF FILE
TO KEY ..... END OF FILE
KEY LENGTH ..... 40
REPORT TYPE ..... STANDARD
AGING DATE ..... 03/05/2010
AGING ITEM ..... 3 INTERNAL
REPORT ITEM #1 ..... 1 INTERNAL
REPORT ITEM #2 ..... 2 INTERNAL
ROW # 1 ..... 02/03/2010 - 03/05/2010
ROW # 2 ..... 01/04/2010 - 02/02/2010
ROW # 3 ..... 12/05/2009 - 01/03/2010
ROW # 4 ..... 11/05/2009 - 12/04/2009
ROW # 5 ..... BEFORE OR = 11/04/2009
    
```

Suspense Aging Report Page 2 of 2

ACR/DETAIL HISTORY FILE: "C:\ACR\Detail-Summary\data\Udshf.dat"				AS OF: 03/05/2010	
HISTORY KEY: 000	00000				
DESCRIPTION	TIME	# OF ITEMS	PCT	TOTAL	PCT
0- 30		0	0.00%	\$0.00	0.00%
31- 60		0	0.00%	\$0.00	0.00%
61- 90		0	0.00%	\$0.00	0.00%
91-120		0	0.00%	\$0.00	0.00%
0- 0		1	100.00%	\$72,721.00	100.00%
TOTAL		1	100.00%	\$72,721.00	100.00%
=====					
0- 30		0	0.00%	0	0.00%
31- 60		0	0.00%	0	0.00%
61- 90		0	0.00%	0	0.00%
91-120		0	0.00%	0	0.00%
0- 0		0	0.00%	0	0.00%
TOTAL		0	0.00%	0	0.00%
=====					
AGING TOTALS					
0- 30		0		\$0.00	
31- 60		0		\$0.00	
61- 90		0		\$0.00	
91-120		0		\$0.00	
0- 0		1		\$72,721.00	
TOTAL		1		\$72,721.00	

History Database Transfer Report

Generated by the Copy History utility when you choose Copy selected records, this report lists each history record transferred with information about the record.

ACR/D releasenumbr	INFOGIX, INCORPORATED		PAGE 001							
DATE: yy/mm/dd	INFOGIX DATA BASE UTILITY									
TIME: 15:39:22	HISTORY DATA BASE TRANSFER REPORT		DATA BASE VERSION: *****/0000001							
	SUMMARY OF HISTORIES TRANSFERRED									
FROM: /home/run/0716hf.dat	TO: /home/run/hist2.dat									
HISTORY KEY	CYCLE#	RETURN RUN#	CODE	RUN DATE	RUN TIME	TYPE	RERUN DATE	RERUN TIME	DATABASE VERSION DEFN /HISTORY	PROGRAM NAME
123-45-6789	00940819	000	0000	97/07/19	13:05:09				0000019 0000008	UDS2000
123-57-3591	00940819	000	0000	97/07/19	13:05:14				0000019 0000008	UDS2000
223-22-3619	00940819	000	0000	97/07/19	13:05:15				0000019 0000008	UDS2000
323-16-9876	00940819	000	0000	97/07/19	13:05:15				0000019 0000008	UDS2000
323-23-9920	00940819	000	0000	97/07/19	13:05:15				0000019 0000008	UDS2000
323-66-1111	00940819	000	0000	97/07/19	13:05:16				0000019 0000008	UDS2000
423-16-7864	00940819	000	0000	97/07/19	13:05:16				0000019 0000008	UDS2000
522-34-1643	00940819	000	0000	97/07/19	13:05:16				0000019 0000008	UDS2000
523-29-0046	00940819	000	0000	97/07/19	13:05:17				0000019 0000008	UDS2000
623-00-3591	00940819	000	0000	97/07/19	13:05:17				0000019 0000008	UDS2000
666-26-1235	00940819	000	0000	97/07/19	13:05:17				0000019 0000008	UDS2000
723-25-3618	00940819	000	0000	97/07/19	13:05:17				0000019 0000008	UDS2000
723-57-0092	00940819	000	0000	97/07/19	13:05:18				0000019 0000008	UDS2000
923-44-1234	00940819	000	0000	97/07/19	13:05:18				0000019 0000008	UDS2000

History Database Unload Report

Generated by the Copy History utility when you choose Unload records to a work file, this report lists each history record unloaded with information about the record.

ACR/D releasenumbr		INFOGIX, INCORPORATED						COPYRIGHT INFOGIX, INC.				
DATE: mm/dd/yy		INFOGIX DATA BASE UTILITY						PAGE 001				
TIME: 15.29.32		HISTORY DATA BASE UNLOAD REPORT						DATA BASE VERSION: *****/0000011				
		SUMMARY OF HISTORIES UNLOADED										
FROM: USERID.TEST2.DETHIST2		TO: USERID.TEST3.DETHIST3										
HISTORY KEY	CYCLE#	RUN#	CODE	RETURN RUN DATE	RUN TIME	TYPE	RERUN DATE	RERUN TIME	DATABASE DEFN	VERSION /HISTORY	PROGRAM NAME	
NAOG		20080113	000	0000	mm/dd/yy	16:34:57	mm/dd/yy	16:35:04	0000006	0000002	UDS2000	
NAOG	453	20080113	000	0000	mm/dd/yy	16:35:05	mm/dd/yy	16:35:09	0000006	0000002	UDS2000	
NAOG	453	20080113	000	0000	mm/dd/yy	16:35:09			0000006	0000002	UDS2000	
NAOG	453	20080113	000	0000	mm/dd/yy	16:35:09			0000006	0000002	UDS2000	
NAOG	453	20080113	000	0000	mm/dd/yy	16:35:09			0000006	0000002	UDS2000	
NAOG	453	20080113	000	0000	mm/dd/yy	16:35:10			0000006	0000002	UDS2000	
NAOG	453	20080113	000	3002	mm/dd/yy	16:35:10			0000006	0000002	UDS2000	
NAOG	809	20080113	000	0000	mm/dd/yy	16:35:05			0000006	0000002	UDS2000	
NAOG	971	20080113	000	0000	mm/dd/yy	16:35:05			0000006	0000002	UDS2000	
NAPC		20080113	000	0000	mm/dd/yy	16:35:00	mm/dd/yy	16:35:05	0000006	0000002	UDS2000	
NAPC	629	20080113	000	0000	mm/dd/yy	16:35:05			0000006	0000002	UDS2000	
NAPC	958	20080113	000	0000	mm/dd/yy	16:35:05	mm/dd/yy	16:35:10	0000006	0000002	UDS2000	
NAPC	958	20080113	000	0000	mm/dd/yy	16:35:10			0000006	0000002	UDS2000	
NAPC	958	20080113	000	0000	mm/dd/yy	16:35:10			0000006	0000002	UDS2000	
NAPC	958	20080113	000	0000	mm/dd/yy	16:35:10			0000006	0000002	UDS2000	
NAPC	958	20080113	000	0000	mm/dd/yy	16:35:10			0000006	0000002	UDS2000	
NAPC	958	20080113	000	0000	mm/dd/yy	16:35:10			0000006	0000002	UDS2000	
NAPC	958	20080113	000	3002	mm/dd/yy	16:35:10			0000006	0000002	UDS2000	
NAPN		20080113	000	0000	mm/dd/yy	16:35:00			0000006	0000002	UDS2000	
NAPS		20080113	000	0000	mm/dd/yy	16:35:00			0000006	0000002	UDS2000	

History Database Reload Report

Generated by the Copy History utility when you choose Reload records to a work file, his report lists each history record reloaded with information about the record.

ACR/D releasenumbr		INFOGIX, INCORPORATED						PAGE 001				
DATE: yy/mm/dd		INFOGIX DATA BASE UTILITY						DATA BASE VERSION: *****/0000001				
TIME: 15:35:16		HISTORY DATA BASE RELOAD REPORT										
		SUMMARY OF HISTORIES RELOADED										
FROM: /home/run/unload.dat		TO: /home/run/hist1.dat										
HISTORY KEY	CYCLE#	RUN#	CODE	RETURN RUN DATE	RUN TIME	TYPE	RERUN DATE	RERUN TIME	DATABASE DEFN	VERSION /HISTORY	PROGRAM NAME	
NAOG		00960113	000	0000	97/07/16	16:34:57	97/07/16	16:35:04	0000006	0000002	UDS2000	
NAOG	453	00960113	000	0000	97/07/16	16:35:05	97/07/16	16:35:09	0000006	0000002	UDS2000	
NAOG	453	00960113	000	0000	97/07/16	16:35:09			0000006	0000002	UDS2000	
NAOG	453	00960113	000	0000	97/07/16	16:35:09			0000006	0000002	UDS2000	
NAOG	453	00960113	000	0000	97/07/16	16:35:09			0000006	0000002	UDS2000	
NAOG	453	00960113	000	0000	97/07/16	16:35:10			0000006	0000002	UDS2000	
NAOG	453	00960113	000	3002	97/07/16	16:35:10			0000006	0000002	UDS2000	
NAOG	809	00960113	000	0000	97/07/16	16:35:05			0000006	0000002	UDS2000	
NAOG	971	00960113	000	0000	97/07/16	16:35:05			0000006	0000002	UDS2000	
NAPC		00960113	000	0000	97/07/16	16:35:00	97/07/16	16:35:05	0000006	0000002	UDS2000	
NAPC	629	00960113	000	0000	97/07/16	16:35:05			0000006	0000002	UDS2000	
NAPC	958	00960113	000	0000	97/07/16	16:35:05	97/07/16	16:35:10	0000006	0000002	UDS2000	
NAPC	958	00960113	000	0000	97/07/16	16:35:10			0000006	0000002	UDS2000	
NAPC	958	00960113	000	0000	97/07/16	16:35:10			0000006	0000002	UDS2000	
NAPC	958	00960113	000	0000	97/07/16	16:35:10			0000006	0000002	UDS2000	
NAPC	958	00960113	000	0000	97/07/16	16:35:10			0000006	0000002	UDS2000	
NAPC	958	00960113	000	0000	97/07/16	16:35:10			0000006	0000002	UDS2000	
NAPC	958	00960113	000	3002	97/07/16	16:35:10			0000006	0000002	UDS2000	
NAPN		00960113	000	0000	97/07/16	16:35:00			0000006	0000002	UDS2000	
NAPS		00960113	000	0000	97/07/16	16:35:00			0000006	0000002	UDS2000	

Transaction Listing Report

Multiple utilities and other processes generate the Transaction Listing Report. This report shows the process that was executed and any validation messages. The specific content depends on the type of processing that generated the report.

This report is generated when you run any of a variety of processes. These include:

- Copy history utilities
- Delete History
- Extract History
- Update History
- Create Comma Delimited Text File

Following is an example.

ACR/D Releasename	INFOGLX, INCORPORATED	COPYRIGHT INFOGLX, INC	PAGE 001
DATE: yy/mm/dd	INFOGLX DATA BASE UTILITY		
TIME: 12:52:54	UTILITY TRANSACTION LISTING		

TRANS			
NO.	T R A N S A C T I O N		VALIDATION MESSAGES
-----1-----2-----3-----4-----5-----6-----7-----8-----			
00001 COPY HF			#UTL0001: PROCESSING COMPLETED WITHOUT ERRORS

Using ACR/Workbench

This appendix documents ACR/Workbench for Detail.

ACR/Workbench for Detail provides the same user interface and all of the functionality of ACR/Detail Client. In addition, ACR/Workbench enables you to create, store, and run jobs on your PC in Local mode. This can be useful, for example, when you want to download input sources from z/OS, create and test reconciliation jobs and their associated tables and history analysis documents on the PC using a graphical interface, and then upload the definitions and run them on z/OS.

Because the user interfaces and most functionality of ACR/Workbench for Detail and ACR/Detail Client are identical, this appendix documents only functionality that is specific to ACR/Workbench. For all other functionality, see the remainder of this manual.

This appendix contains the following sections:

- “Restrictions for Mainframe Users” on page 307
- “ACR/Workbench Configuration Modes” on page 308
- “Downloading and Uploading between z/OS and ACR/Workbench” on page 309
- “Importing Definitions and Input Sources from z/OS” on page 311
- “ACR/Workbench for Detail Tutorial” on page 313
- “Using QuickChange” on page 314
- “Features Applicable Only to Jobs Uploaded to z/OS” on page 316

Restrictions for Mainframe Users

For ACR/Workbench jobs uploaded and run on the mainframe, the following features are not supported:

- E-mail notification
- Web extraction
- Web publishing

ACR/Workbench Configuration Modes

ACR/Workbench can be used in Local mode, Windows mode, UNIX mode, or Linux mode, depending on your licensing.

The Windows, UNIX, or Linux configuration modes, which are also available in ACR/Detail Client, are documented in “[Configuring ACR/Detail Client](#)” on page 34.

Local Mode

In Local mode, all of the ACR/Detail components and files that will be used reside and run on the PC. In this mode you can set up, test, and run reconciliation jobs on the PC.

One Local configuration (both the server platform and the configuration name are Local) is set up by default when you open ACR/Workbench for the first time. This configuration contains default path names for storing definitions and reports, enabling you to run reconciliation jobs directly on the PC. Jobs defined using this configuration can be uploaded to z/OS, UNIX, or Linux without making any changes to the default configuration. You can edit this configuration and/or establish multiple configurations with the server platform set to Local, each with a unique name.

Complete the Add/Edit Configuration dialog box - Internet Information tab only if you are not using a UNIX or Linux configuration and you want to use e-mail notification as described in “[Message Processing](#)” on page 106 and/or Web publication as described in “[Distribution of Reports](#)” on page 268.

Setting User Options in Local Mode

If you set user options in Local mode, they will be stored in the user options file (uniuf.dat) on the PC. These options will apply only to jobs run in Local mode.

To set the user options for Local mode, select a local configuration and then select **Options > User Options** and complete the tabs containing options you want to change.

Downloading and Uploading between z/OS and ACR/Workbench

Note: In UNIX or Linux mode, you can download and upload between ACR/Workbench and UNIX or Linux just as you can in ACR/Detail Client. For more information, see [Downloading, Uploading, and Importing Definitions](#) on page 37.

Downloading Definitions and Input Sources from z/OS

Types of Files to Download

- ACR/Detail definitions. You may want to download existing ACR/Detail definitions from the mainframe in order to modify and test them using ACR/Workbench.

If you download definitions, you must import them before using them in ACR/Workbench. For information on importing definitions, see [Chapter 15, “Importing Definitions”](#) on page 257.

Files having the .def extension that have been downloaded from z/OS or for use in ACR/Workbench may contain definitions for multiple jobs along with their file definitions (definitions for their input sources) and any associated tables and history analysis definition files.

- Input sources. For non-relational database input sources, you can streamline the process of creating file definitions in ACR/Workbench by using the mark-and-capture method, which is described in [“Preparing to Use Mark-and Capture”](#) on page 117. If the input source file is on the mainframe, you must download it in text (ASCII/CR/LF) format to a location that is accessible to ACR/Workbench.

Download Procedure

In ACR/Workbench, select **File > Download** while using a Local configuration to display the File Download dialog box. See the dialog box help for more information.

Note: If the file on the host is too large to work with on the PC, you may have to create a representative sample of the file. You can then download the sample and create your extraction definitions for use with the entire file on the host.

UNICTN - Caret to Not Utility

If you are using a 3270 emulator transfer program that fails to convert caret symbols (^) to the not symbols (¬ ASCII 173), you need to run the UNICTN utility on a downloaded file before importing it. For instructions, see “UNICTN - Caret (^) to Not (¬ ASCII 173) utility” on page 317.

Uploading Definitions to z/OS

After completing and saving your definitions and updating the definition database using ACR/Workbench, you can upload them to z/OS to run the job. Before uploading, see the next section.

UNINTC - Not to Caret Utility

If you have a 3270 emulator transfer program that fails to convert the not symbols (¬ ASCII 173) in a file to caret symbols (^), you need to run the UNINTC utility on the file before you upload it.

Upload Procedure

To upload definitions and run them on the mainframe, you should be signed on to z/OS and be at the READY prompt. The PATH statement utilized by your PC must contain an entry that points to the directory where your 3270-emulation SEND and RECEIVE commands are located.

In ACR/Workbench, select **File > Upload** while using a Local configuration to display the File Upload dialog box with the fields required for uploading to z/OS.

Avoiding Problems When Uploading Jobs That Do Not Store History to z/OS

If you intend to upload your definitions to z/OS and you have multiple jobs storing to the same history database, carefully consider before combining 1) Do not add or replace history records as your selection for the **Store new history** field in the Override Options dialog box or 2) the Do not store history option in the Store History dialog box) with Perform Reconciliation in Rewrite History mode as your selection for the **Rewrite History** field on the Override Options dialog box. For more information, see the help for the Store history dialog box and the Override Options dialog box.

Importing Definitions and Input Sources from z/OS

General instructions for importing are provided in “[Importing Definitions](#)” on page 257. When importing from z/OS, follow the instructions there. In addition, be aware of the information regarding discrepancies provided in the following section.

Eliminating Discrepancies when Moving Definitions from z/OS

Name Creation Field Options

The **Name Creation** field in the Import Options dialog box enables you to select whether to use names or item numbers to identify imported history items, calculated items, and rules.

Note: To see the item numbers, select **Options > Batch Options >** to display the Batch Options dialog box and select the option Show item numbers in Control Task.

The two options for this field are as follows:

- **Use Descriptions.** If you choose this option, item names will be used. The first 40 characters from the description field on the batch transaction record will be placed in the item name field on ACR/Workbench. If this name is not unique, it will be appended with a number to make it unique. The next 40 characters will be placed in the item description field. This option sets the option in the Length of Descriptions dialog box to 40. When the List files option is selected, this option is disabled and Use Item Numbers is selected.
- **Use Item Numbers.** If you choose this option, item numbers (such as I001) will be used in the item name field on ACR/Workbench. The entire 80-character description from the batch transaction record will be placed in the item description field. This option sets the option in the Length of Descriptions dialog box to 80.

Differences between ACR/Workbench and ACR/Detail for the Mainframe

ACR/Workbench for Detail differs from ACR/Detail for the mainframe in the way internal items, extended internal items, history items, calculated items, and rules are identified.

When you create these items in ACR/Workbench, you assign a meaningful name of up to 40 characters and an optional description of up to 40 characters (unless Use 80 character descriptions is specified as described below).

ACR/Detail for z/OS does not assign meaningful names to items. Items are identified by an item number and a description of up to 80 characters.

Typical Import Processing of Item Names and Descriptions

When you import items from ACR/Detail for z/OS to ACR/Workbench for Detail, the 80 character z/OS item description is typically split into 2 fields:

- The first 40 characters are used to populate the name field.
- The last 40 characters are used to populate the description field.

If the first 40 characters of the item description for 2 items being imported are the same, the characters that populate the name field will be appended with a number when they are imported to ACR/Workbench.

Importing with the Use Item Numbers Option

If you do not want the 80 character z/OS description to be split into a 40 character name and 40 character description when importing to ACR/Workbench, select the Use Item Numbers option for the **Name Creation** field on the Import Options dialog box. When the item is imported into ACR/Workbench, the item name field will be populated with an item number such as I001. The item description will be 80 characters long and will be populated with the complete item description from z/OS.

The Length of Descriptions dialog box option will be automatically set to Use 80 character descriptions and cannot be changed.

Creating Batch Transaction Records in ACR/Workbench

When ACR/Workbench creates batch transaction records, the information in the name and description fields for internal items, extended internal items, history items, calculated items, and rules will be processed based on the option selected in the Length of Descriptions dialog box, as follows:

If Use 40 character descriptions is specified, the item name and description fields will be concatenated to create an 80-character description.

If Use 80 character descriptions is specified, the item description can be up to 80 characters long in ACR/Workbench, and the entire description will be written to batch. The value entered in the name field will not be written to batch.

ACR/Workbench for Detail Tutorial

This tutorial shows you how to set up and run an ACR/Detail reconciliation job in Local mode using ACR/Workbench for Detail.

Preparing for the Tutorial

Before you begin this tutorial, you need to:

1. Access ACR/Workbench for Detail. Select **Start > Programs > Infogix > ACR/Workbench for Detail**. (The release number will be part of the program name beginning with Release 4.0.) You will see the Control Task (Job) View, where you can set up your reconciliation job for the tutorial.
2. Familiarize yourself with the product interface, which is the same as that of ACR/Detail Client. See [“Using the ACR/Detail Client Interface” on page 26](#).
3. Use the default Local configuration for ACR/Workbench. Make sure the Local configuration is selected in the list box to the right of the toolbar when you start the tutorial.
4. Ensure that you followed the instructions in [“Initializing the Definition and History Databases” on page 37](#).
5. Verify that you have access to the sample reports that will be used as input sources in the tutorial. The report names are Dex1dd1.rpt and Dex1dd2.rpt. The default location for these reports in Local mode is: C:\Infogix\Detail32\samples.

Determining Your Reconciliation Requirements

Refer to “Determining Your Reconciliation Requirements” on page 42 for instructions.

Creating and Running the Job

The remainder of the tutorial is almost identical to the tutorial in “Tutorial” on page 41:

Note: Skip all sections for UNIX or Linux.

- In the “Tutorial” on page 41, begin with “Complete the Basic (Entity) Information” on page 45 and follow all instructions through step 3 on page 46.
- Follow the instructions in “Define the First Input Source” on page 47 except that when you complete path and file name in the **File Name** field in the Name dialog box, the file Dex1dd1.rpt will be in the Sample folder on the local PC.
- Follow the instructions in “Define the Second Input Source” on page 51 except that when you complete path and file name in the **File Name** field in the Name dialog box, the file Dex1dd2.rpt will be in the Sample folder on the local PC.
- Follow all of the remaining instructions for the tutorial, beginning with “Define the Reconciliation Rule” on page 54 and ending with “Viewing the Output Reports and Handling Errors” on page 57.

Using QuickChange

Description

QuickChange is a feature of ACR/Workbench for Detail that enables you to view information in your history database and update the value of history items for a specific key (in Windows mode only).

Using QuickChange

1. Access QuickChange from ACR/Workbench for Detail by selecting **Run > QuickChange**.
2. Select **File > Open** to open the Select Databases dialog box and specify the following:

- The full path and filename of the history database you want to view and definition databases for the history you want to view.
 - The full path and filename of the associated definition database.
3. Select **View > Display Filter** to access the Display Filter Criteria dialog box and select filter criteria (keys, run dates, cycles, return codes) and filtering options.
 4. In the left pane, double-click on a history key to display information on each cycle in the right pane. The column definitions are shown below.

Column Heading	Description
Cycle	Cycle number of the job using this key.
Return Code	Highest return code set for this cycle.
Run Date	Date of the first run of the cycle in YY/MM/DD format.
Run Time	Time of the first run of the cycle in HH:MM:SS format
Rerun Date	Date of the last rerun of the cycle in DD/MM/YY format.
Rerun Time	The time of the last rerun in HH:MM:SS format
Rerun Count	Number of times this cycle has been rerun, if any.
Definition Database Version (Def DB Ver)	Number of times the definition database has been updated by this cycle (Initialization, Update Definitions, etc.). This number increments by one number with every update.
History Database Version (Hist DB Ver)	Number of times the history database has been updated by this cycle (Initialization, Reconciliation, Update History, etc.). However, this number may not necessarily display a higher version number until a reconciliation cycle has been run.
Job ID	Job name and step name assigned to the job to which this cycle's key belongs.

5. In the left pane, double-click on a cycle to display information on each history item in the right pane. The column definitions are shown below.

Column Heading	Description
Item Number	The item number assigned to the item in the originating job.
Description	The description given to the item in the originating job.
Value	The currently recorded value for the item. For information on how this value relates to specific values in your input source, examine the Field Type and Accumulate Option choices set for the item on the Detail Field dialog box in the originating job.

Features Applicable Only to Jobs Uploaded to z/OS

The following features are available only for ACR/Workbench for Detail jobs that will be uploaded and run on z/OS.

- **Input Sources (Input Source View > File Organization dialog box > File/Organization Type)**
 - **DB2 Table**—Direct access to a DB2 table, as opposed to using ODBC.
 - **Direct Spool Dataset**—Access reports directly from the mainframe spool.
- **Alternate Control Tasks (Control Task (Job) View > Alternate Control Tasks > New)**

This is an alias for the (primary) job ID defined for the job. It is mainly used to submit JCL to run with production job names in a test environment.

■ UNINTC - Not (¬ ASCII 173) to Caret (^) utility

UNINTC is usually used before uploading files to the mainframe. It converts the not symbols (¬ ASCII 173) in a file to caret symbols (^) if your 3270 emulator does not perform this conversion.

The steps to use this utility are:

- a. Create the batch definition transactions on ACR/Workbench in Local mode.
- b. At the DOS prompt on the PC, run UNINTC against these transactions. The format of the command is UNINTC filename, where filename is the file to be converted.
- c. Upload to the TRANLIB on the mainframe.
- d. Update the definition database.

■ UNICTN - Caret (^) to Not (¬ ASCII 173) utility

UNICTN is usually used after downloading files from the mainframe. It converts the caret symbols (^) in a file to not symbols (¬ ASCII 173) if your 3270 emulator does not perform this conversion.

The steps to use this utility are:

- a. Download the batch definition transactions to the ACR/Workbench.
- b. At the DOS prompt on the PC, run UNICTN against the batch definition transactions. The format of the command is UNICTN filename, where filename is the file to be converted.
- c. Import the batch definition transactions into ACR/Workbench.

■ Extraction Program Interface. For more information, see the next section.

Extraction Program Interface

The ACR/Detail Extraction Program Interface is typically used by programmers to quickly process data files containing millions of records. The program built on ACR/Workbench is typically uploaded to z/OS, where it is compiled and run.

Note: You can compile and run the program on ACR/Workbench if you have a COBOL compiler.

This section contains only information specific to ACR/Workbench for Detail. For complete information, see the *ACR/Detail User Guide for z/OS*.

Build Extraction Program utility

The Build Extraction Program utility can be used through ACR/Workbench for Detail (or through the batch file `udpcgen.skl` or `udpcgen.bat` on Windows or `udpcgen.sh` on UNIX or Linux) to generate an Enterprise COBOL program from the file definitions for use with the Extraction Program Interface. The file name for this program is created by appending `.cbl` to the file definition ID. This COBOL file is typically uploaded to z/OS and compiled there.

Implementation

1. Develop and test definitions using ACR/Workbench or using batch or `.skl` files (Windows) or scripts (UNIX or Linux).
This step includes entering the job and file definitions, saving the definitions, and running reconciliation. Use the same file(s) that the application program will process to establish the extraction criteria. You may use the debugging facilities (Trace Report, Extracted Reports, etc.) in ACR/Workbench to develop satisfactory definitions.
2. Create extraction programs for file definitions using the Build Extraction Program Utility on ACR/Workbench.
3. Unless you are going to compile and run the program on ACR/Workbench (see note above), upload the generated program to z/OS, compile it there, and test it using `UDPTEST`.

Maintaining the Definition Database

Unused definitions take up space and slow down processing. To maximize processing efficiency, you need to remove unused definitions from the definition database. This appendix provides the procedure for purging such definitions.

This appendix includes the following sections:

- “Create the .def File with the Purge Records” on page 319
- “Edit and Run the Update Definition Database File or Script” on page 320

Create the .def File with the Purge Records

Use a text editor such as Microsoft® Notepad to create a definition (.def) file containing a purge record for each definition type you want to delete. A purge record consists of the transaction key for the job, file, or table definition type, with the purge code (an asterisk) in position 26. The transaction layout is job name, step name, RLQ, record type, sequence number, and asterisk. See the examples in the next section.

A definition type of J0 deletes a job, F0 deletes a file, etc. For more information on transaction keys, see “Transaction Key” on page 262.

Include the * (purge code) in position 26 to delete all definitions for the specified job ID from the definition database.

Examples of Purge Records

Sample purge record to delete all basic job (J0) definitions for a job:

```
-----1-----2-----+
JOB001  STEP10      J0 0001*
```

In the example above, no qualifier is specified. If JOB001 STEP10 is a base job with 17 reconciliation level qualifiers (RLQs), only the base job will be deleted.

D ■ Maintaining the Definition Database

Edit and Run the Update Definition Database File or Script

Sample purge records to delete basic job definitions for two reconciliation level qualifiers (RLQs):

```
-----1-----2-----+  
JOB001  STEP10  17J0 0001*  
JOB001  STEP10  18J0 0001*
```

The purge records shown above will purge RLQ 17 and RLQ 18 of the job/step JOB001 STEP10. The base job and any other RLQs will remain in the definition database.

Sample purge record for a basic file information (FO) definition:

```
-----1-----2-----+  
FILEDD01 000000  F0 0001*
```

Sample purge record for a translation table (T1) definition:

```
-----1-----2-----+  
TRANSLATIONNAME  T1 0011*
```

When you have created purge records for all the batch definition transaction records you need to purge, name the file and save it in text format with the extension .def.

Edit and Run the Update Definition Database File or Script

Windows Procedure

1. Locate or create UDSUPDT.BAT (your Update Definition Database batch file) in your ACR/Detail installation folder.
2. Copy the file to another folder and rename it.
3. Edit the SET UNIINP statement in the copied/renamed file to point to the purge file you created based on the instructions in the preceding section. For example, if the fully qualified path name of the purge file is C:\Infogix\Detail32\Purge\Purge.def. the edited SET UNIINP statement would be:

```
SET UNIINP="C:\Infogix\Detail32\Purge\Purge.def"
```

4. Execute the edited/renamed Update Definition Database batch file. Batch definition transaction records with the same job, file, or table ID as the purge card will be removed from the database.

UNIX or Linux Procedure

1. Locate install directory /sumdet/scripts/udsupdt.sh.
2. Copy the file to another directory.
3. Edit the file. Replace the question marks (?) with the correct information. The result should be an update script that has your purge definition file pointed to by dd_UNIINP.
4. Run udsupdt.sh.

D ■ Maintaining the Definition Database

Edit and Run the Update Definition Database File or Script

Using ACR/Detail for Windows in Batch

When you run ACR/Detail for Windows, batch files are executed. The job and file definitions used by the batch files are typically created through ACR/Detail Client or ACR/Workbench.

This appendix assists you in customizing the ACR/Detail batch files for your batch environment. In addition, it provides instructions for migrating your jobs from test to production.

This appendix contains the following sections:

- “Conditionally Executing a Step Based on a Return Code” on page 323
- “Before You Begin” on page 324
- “Enable Proper Processing of Tab-Delimited Input Sources” on page 324
- “Increase the Maximum file Size (Optional)” on page 325
- “Editing Batch Files” on page 325
- “Batch File Quick-Reference with Page Numbers” on page 326
- “Migrating Test Definitions to Production” on page 346

Conditionally Executing a Step Based on a Return Code

In a production batch stream, you may wish to conditionally execute a step based on the return code of an ACR/Detail step that precedes it. You can check the return code in your batch stream by evaluating `ERRORLEVEL`. For example:

```
@ECHO OFF
REM SAMPLE BATCH FILE
REM Run ACR/Detail
CALL UDS2000
REM Check return code
if ERRORLEVEL 0 goto OK
REM Not OK, so run summary job
CALL UAC2000
:OK
```

Before You Begin

This section gives you important information that you should know before attempting to set up and execute any of the batch files.

Location of Batch Files

Batch files are located in the \$BASE/bat subdirectory, where \$BASE is the directory where ACR/Summary and ACR/Detail Server was installed. `c:\Infogix\SumDetServer\bat` is the default.

Ensure that you leave the batch file in its original location and copy it to another folder, where you can modify it as needed.

The modified batch file should specify a path so that it can find the ACR/Detail COBOL modules.

Enable Proper Processing of Tab-Delimited Input Sources

As explained in “[Extracting Directly from a Delimited Data File](#)” on [page 162](#), you can extract delimited values directly from input source records without a user program. However, if the file is tab-delimited, Micro Focus COBOL, on which ACR/Detail’s batch executables for Windows are compiled, converts the tabs to spaces when the file is read, eliminating the delimiters, unless the default processing is changed.

To change the Micro Focus default processing, do the following:

1. Create the external file handler configuration file `extfh.cfg`, in the ACR/Detail installation folder `C:\Infogix\Detail32` and add the following contents:

```
[XFH-DEFAULT]
EXPANDTAB=OFF
```

2. Add the following statement to the `uds2000` execution `.skl .bat` file:

```
SET EXTFH=C:\Infogix\Detail32\extfh.cfg
```

Note: You can also add statements to the `extfh.cfg` file if you need to increase the maximum file size, as described in the following section.

Increase the Maximum file Size (Optional)

ACR/Summary and ACR/Detail ship with environment variable FILEMAXSIZE=4. This will be the setting unless the person who installed the product has overridden it as described below. This setting will be appropriate for users who always use a file input size of 4GB or less and whose definition and history databases are both less than 1GB.

If you use input files greater than 4GB or if either of your databases is greater than 1GB, you will have to override this value by setting FILEMAXSIZE=8 (no limit) in the external file handler configuration file.

To override the environment variable FILEMAXSIZE, do the following:

1. Create the file extfh.cfg in the ACR/Detail installation folder C:\Infogix\Detail32, and add the following contents:

```
[XFH-DEFAULT]
FILEMAXSIZE=8
IDXFORMAT=8
FILEPOINTERSIZE=8
```

2. Add the following statement to the uds2000 execution .skl .bat file:

```
SET EXTFH=C:\Infogix\Detail32\extfh.cfg
```

Note: You can also add statements to the extfh.cfg file to enable proper processing of tab-delimited input sources, as described in the previous section.

Editing Batch Files

Question Marks Indicate Required Edits

Replace question marks in the file with appropriate path, file name, path and file name, or parameter values.

Instructions for Editing Batch Files

In addition to the following general procedure, this chapter contains a section with documentation specific to each batch file. Please refer to that section when editing and running the batch file.

1. Go to Infogix's \$BASE\bat directory. c:\Infogix\SumDetServer\bat is the default.

2. Make a copy of the batch file for modification. Do not modify the original batch file, which should be treated as a template. See “Location of Batch Files” on page 324.
3. Set the environment variables as indicated by the comments in the batch file and the Required Environment Variables Table provided in the batch file-specific documentation.

Following is an example of some environment variables requiring definition in a batch file.

```
::-----  
-----  
::Set the product and cobol variables  
::-----  
-----  
SET ACRDWB=?????????  
SET RPT=?????????  
SET COBCONFIG=%ACRDWB%\cob.cfg
```

4. Set any required run-time parameters based on the Run-time Parameter (PARM Options) Table provided in the batch file-specific documentation.

Following is an example of a run-time parameter that needs to be set:

```
::-----  
:: Set run time parameters  
::-----  
SET PARM='?????????'
```

5. Set any jobname and stepname values required by the batch file. For example:

```
ECHO JOBNAME=????????? >>%JOBVAR%  
ECHO STEPNAME=????????? >>%JOBVAR%
```

A few batch files, such as uds2000.bat (Run Reconciliation), require additional edits, which are described in the batch file-specific documentation.

Batch File Quick-Reference with Page Numbers

The batch files are documented in the following sections:

- “udpcgen.bat - Build Extraction Program” on page 327
- “udptest.bat - Execute Extraction Program Test” on page 328
- “udpuser.bat - Test User Program” on page 329
- “uds2000.bat - Run Reconciliation” on page 331
- “uds6000.bat - History Analysis” on page 334
- “udsdfb.bat - Build Batch Transactions” on page 335
- “udsdf.l.bat - List Definitions” on page 335

- “udsdhfa.bat - Suspense Aging” on page 336
- “udshfc.bat - Copy History” on page 336
- “udshfd.bat - Delete History” on page 337
- “udshfe.bat - Extract History” on page 337
- “udshfl.bat - List History” on page 338
- “udshfr.bat - Reorganize History” on page 338
- “udshft.bat - Transfer History (Selected Records)” on page 339
- “udshftu.bat - Unload History” on page 339
- “udshfe.bat - Extract History” on page 337
- “udshfl.bat - List History” on page 338
- “udshfr.bat - Reorganize History” on page 338
- “udshft.bat - Transfer History (Selected Records)” on page 339
- “udshftu.bat - Unload History” on page 339
- “udshftr.bat - Reload History” on page 340
- “udshfu.bat - Update History” on page 341
- “udsinit.bat - Initialize Databases” on page 342
- “udsmr01.bat - Management Reporting” on page 342
- “udstfint.bat - Initialize Translation Database” on page 343
- “udstfls.bat - List Translation Database” on page 343
- “udsupdt.bat - Update Definition Database” on page 344
- “udscsv.bat - Create Comma Delimited Text File” on page 345
- “Migrating Test Definitions to Production” on page 346

udpcgen.bat - Build Extraction Program

This batch file is used to generate COBOL extraction programs that correspond to file definitions. The generated programs are an integral part of program interface processing. For more information see “[Extraction Program Interface](#)” on page 318.

To use the batch file, see “[Editing Batch Files](#)” on page 325 and refer to the following information.

E ■ Using ACR/Detail for Windows in Batch

udptest.bat - Execute Extraction Program Test

Required Environment Variables Table

Variable	Definition
ACRDWB	Path where UNIRUN.exe resides (\$BASE\Cobol)
RPT	Path where the report files will be generated
UNIDF	Path and file name of the definition database.
UNICBL	Path and file name of the program ID.

Run-time Parameter Table (PARM options)

Option	Use
FID - File Definition ID	Name of the file that contains your file definitions.
PID - Program ID	Name that you assign to the compiled program. The resulting file name will be followed by a .cbl extension. If this option is left blank, it will default to the FID name that you specified.

udptest.bat - Execute Extraction Program Test

This batch file is used to test generated extraction programs in program interface mode, prior to modifying application programs with application program interface calls. It can process input files with either physical sequential or user program dataset organizations. User Program dataset organizations call external COBOL programs which can be written to access input sources that ACR/Detail does not directly access.

See “[Extraction Program Interface](#)” on page 318 for instructions on creating the programs to be tested by this batch file.

Required Environment Variables Table

Variable	Definition
ACRDWB	Path where UNIRUN.exe resides (\$BASE\Cobol)
RPT	Path where the report files will be generated
UNIDF	Path and file name of the definition database.
UNIDF	Path and file name of the definition database.
UNIHF	Path and file name of the history database.
UNIDATA	Required if you specify the Write Reconciliation Output File option in your Basic Control Task information.

Run-time Parameter Table (PARM options)

Option	Use
-- blank --	Will default the cycle number to 1.
Cycle Number Override	Sets the cycle number according to your specification. Example: CYC#00960101

udpuser.bat - Test User Program

This batch file is used to test user programs in program interface mode prior to modifying application programs with API calls. It can process input files with either physical sequential or user program dataset organizations. User program dataset organizations call external COBOL programs which provide access to input sources that ACR/Detail does not directly access.

E ■ Using ACR/Detail for Windows in Batch

udpuser.bat - Test User Program

Required Environment Variables Table

Variable	Definition
ACRDWB	Path where UNIRUN.exe resides (\$BASE\Cobol)
RPT	Path where the report files will be generated
UNIDF	Path and file name of the definition database.
UNIHF	Path and file name of the history database.

Run-time Parameter Table (PARM options)

Option	Use
Parm ID	The name (8 characters or less) of the parameter that you wish to run.
EBCDIC Parm Option	An option indicating if the parm is in EBCDIC or not.
Parm Data	The contents (36 characters or less) of the parm itself.

Additional Instructions

Set the user application name by replacing the USERPGM question marks with the program ID of your user application:

```
USERPGM=???????????
```

uds2000.bat - Run Reconciliation

This batch file is used to run reconciliation, which extracts the specified data from the application input sources, processes your rules, and reports on the results.

Required Environment Variables Table

Variable	Definition
ACRDWB	Path where UNIRUN.exe resides (\$BASE\Cobol)
RPT	Path where the report files will be generated
UNIDF	Path and file name of the definition database.
UNIHF	Path and file name of the history database.
UNITF	Path and file name of the dynamic translation file.
DWBIOBUFF	This variable is commented out in the template batch file. When ACR/Detail needs to read records that are greater than 256 bytes, uncomment and set to 1.

E ■ Using ACR/Detail for Windows in Batch

uds2000.bat - Run Reconciliation

Run-time Parameter Table (PARM options)

Examples:

ROPT=Y.....

JNAM=CUST

STEP=STEP1

JQ=02

CYC#=20060815

Keyword	Possible Values
'JNAM=jjjjjjj'	Jobname = X(8) (max)
'STEP=sssssss'	Stepname = X(8) (max)
'CYC#=ccccccc'	Cycle Number = 9(8) (Must be numeric and > 0)
'ROPT=abcdefgh' (Processing Options)	a ¹ = Store History Option (Y,N or X or (.)) b ¹ = Print Reconciliation Control Report Option (Y,N,X,Z or (.)) c ¹ = Print Reconciliation Recap Report Option (Y,N,X,Z or (.)) d ¹ = Set Return Code Option (Y,N,X or (.)) e ¹ = Print Extracted Report Option (Y,N or (.)) f ¹ = Rewrite History Option (Y,N or (.)) g ¹ = Print Accumulated/Sorted Report (Y,N,X or (.)) h ¹ = Print History Option (Y,N or (.))
'ROPT2=abcdefgh'	a ¹ = Write Output File (Y,N,X,Z or (.)) b = Reserved (.) c ² = Key Sort Order (A, D, N, or (.)) d = Display Processing Status (Y, N, or (.)) e-h = Reserved (.)

1 - Y = Applies to all keys. N = Applies to no keys. X = Applies to out-of-balance keys only.
Z = Applies to in-balance keys only. (.) = Do not override. Use the information in the definition database.

2 - A = Sort in ascending order. D = Sort in descending order. N = Do not sort.
(.) = Do not override (use the information in the definition database).

Re: Reconciliation Level Qualifier (RLQ) jobs: The Key Sort Order settings from the base job will be applied to RLQ jobs. Likewise, any run-time overrides set for the Key Sort Order setting in the base job will be applied to RLQ jobs. Therefore, in an RLQ job, setting up a run-time override for the key sort order is not allowed.

E ■ Using ACR/Detail for Windows in Batch

uds2000.bat - Run Reconciliation

Keyword	Possible Values
'SYSOUT'	SYSOUT (A SYSOUT report will be created even if no #U messages are generated. The report will contain an informational header message: "#USY0001: INFOGIX JOB STARTED.")
'BYPASS'	BYPASS (The USD2000 program will bypass reconciliation processing and return a step completion code of 0000.)
'NEGATE'	NEGATE (Reverse numeric additions to the history by subtracting the values extracted from the input(s))
'NOEXTR'	NOEXTR (The data is already extracted into the workfile (UNIWRK). Used for backout and multiple history updates. If this parameter is used without the NOSORT parameter, then extraction writes directly to work file 2.
'NOSORT'	NOSORT (NOSORT and NOEXTR options can save time if the work files are available to be added to history. If you specify this parameter, sorting and the second round of accumulations are omitted. This speeds up processing when items are already in order.)
'NOHIST'	NOHIST (Helps in batching multiple jobs for application to the history file in one job stream)
'CRXML=n'	Control Report XML Option where n is one of the following: 1 - do not generate the Control Report in XML format 2 - generate, write to file 3 - generate, autoload to database
'URXML=n'	User Report XML Option where n is one of the following: 1 - do not generate the User Report in XML format 2 - generate, write to file 3 - generate, autoload to database
'FRXML=n'	Free-Form Report XML Option where n is one of the following: 1 - do not generate the Free-Form Report in XML format 2 - generate, write to file 3 - generate, autoload to database
'DEBUG=abcdefgh'	a ³ = XML Writer Debug Option (Y, N, blank or (.)) b ³ = Visibility API Debug Option (Y, N, blank or (.)) c ³ = Infogix Assure History Debug Option (Y, N, blank or (.)) e-h = Reserved (.)
3 - Y = Debug on, N = blank, or (.) = Debug off.	

Additional Instructions

Specify the Input Sources

For input source types other than relational database tables, specify the path and file name for each input source.

If your input is coming from a relational database, it is not necessary to set an input source. Simply comment out the SET statement from the Input Sources section of the batch file using a double colon (::).

Concatenation of Input Sources

When you specify the paths and file names for the input sources, you can optionally concatenate as many as 35 files in a single INPUTSOURCEDD. (You cannot, however, concatenate files that make up an external translation table.) The path names must be separated by semicolons. The maximum length of all concatenated pathnames is 8192 characters.

E-mail Notification

If you are using e-mail notification, uncomment the SET SMTPHOST and SMTPPORT parameters.

uds6000.bat - History Analysis

This batch file extracts, analyzes, and reports on selected history data. A history key or an accumulation of history keys in the current cycle of data is compared to prior cycles of data and then displayed in the report.

Required Environment Variables Table

Variable	Definition
ACRDWB	Path where UNIRUN.exe resides (\$BASE\Cobol)
RPT	Path where the report files will be generated
UNIDF	Path and file name of the definition database.
UNIHF	Path and file name of the history database.

udsdfb.bat - Build Batch Transactions

This batch file builds batch transaction cards directly from the definition database. For more about batch transactions records, see [“Using Batch Definition Transactions” on page 261](#).

Any or all of the records on the definition database may be selected. The batch file creates all the transaction records in the definition database or those associated with the selected definition type.

Required Environment Variables Table

Variable	Definition
ACRDWB	Path where UNIRUN.exe resides (\$BASE\Cobol)
RPT	Path where the report files will be generated
UNIDF	Path and file name of the definition database.
UNIUPDT	Path and file name of the batch definitions file.

udsdfb.bat - List Definitions

This batch file creates a formatted list of definitions from the definition database for audit trail or troubleshooting purposes. The batch file lists all the transaction records in the definition database or those associated with the selected definition type.

Required Environment Variables Table

Variable	Definition
ACRDWB	Path where UNIRUN.exe resides (\$BASE\Cobol)
RPT	Path where the report files will be generated
UNIDF	Path and file name of the definition database.

udsdhfa.bat - Suspense Aging

This batch file reports on aging information for values contained in the history database for suspense applications.

You have a history database that consists of history records that are in suspense. These have an internal item which defines the date on which the item went into suspense. The aging report summarizes the contents of the database (for all keys or for some level of the key if it contains a meaningful structure) in terms of the age of specified items. Each aging column can be specified in ranges of days, months, or years in any combination. An aging column for greater than a specific aging period can also be specified.

Required Environment Variables Table

Variable	Definition
ACRDWB	Path where UNIRUN.exe resides (\$BASE\Cobol)
RPT	Path where the report files will be generated
UNIDF	Path and file name of the definition database.
UNIHF	Path and file name of the history database.

udshfc.bat - Copy History

This batch file copies all records from one history database to another. The source history database and the target history database are specified by you. This batch file can be useful for backup purposes or to create a copy of a production database to be used in a test environment.

Required Environment Variables Table

Variable	Definition
ACRDWB	Path where UNIRUN.exe resides (\$BASE\Cobol)
RPT	Path where the report files will be generated
UNIHF	Path and file name of the history database.
UNIHF2	Path and file name of the new history database.

udshfd.bat - Delete History

Use this utility to perform a physical delete of one or multiple history records based on a combination of key, cycle ID, job ID, and date criteria. Wildcards can be used to select multiple keys and multiple job IDs. For more information, see [“Delete History” on page 235](#).

Required Environment Variables Table

Variable	Definition
ACRDWB	Path where UNIRUN.exe resides (\$BASE\Cobol)
RPT	Path where the report files will be generated
UNIHF	Path and file name of the history database.

udshfe.bat - Extract History

This utility allows you to create a sequential file (known as an extract file) of history records from the history database. For more information, including the record layout, see [“Extract History” on page 236](#).

Required Environment Variables Table

Variable	Definition
ACRDWB	Path where UNIRUN.exe resides (\$BASE\Cobol)
RPT	Path where the report files will be generated
UNIHF	Path and file name of the history database.
UNIDHFE	Path and file name of the extract file.

udshfl.bat - List History

This batch file prints a formatted summary of the records in the history database for purposes such as audit trail and troubleshooting. The history database to list is specified as an environment variable.

Required Environment Variables Table

Variable	Definition
ACRDWB	Path where UNIRUN.exe resides (\$BASE\Cobol)
RPT	Path where the report files will be generated
UNIHF	Path and file name of the history database.

udshfr.bat - Reorganize History

This batch file reorganizes the contents of a history database in place. Detail history for any given history key is kept physically in order using relative cycle number. During the reorganization process, all logically deleted histories are permanently erased. The batch file rebuilds relative cycle numbers based on the actual cycle ID values found for any history key. Reorganizing history results in a more efficient file.

Required Environment Variables Table

Variable	Definition
ACRDWB	Path where UNIRUN.exe resides (\$BASE\Cobol)
UNIDF	Path and file name of the definition database.
UNIHF	Path and file name of the history database.
UNIHFw	Path and file name of a flat (physical sequential) file to which the records will be unloaded.

udshftr.bat - Reload History

This utility can be used after you have used the Unload History utility (see “[udshftu.bat - Unload History](#)” on page 339). It reloads the previously unloaded history records to an empty database that you have already created. History records can be selected by history key and by date and/or time.

Refer to “[Editing Batch Files](#)” on page 325 and to the following tables when editing this batch file.

udsinit.bat - Initialize Databases

This batch file initializes the definition and/or history database of your choice, erasing the contents. Be careful to specify the correct database(s) so that you do not accidentally erase data from files.

You can also use this batch file to create a new definition and/or history database by specifying a new file name to initialize. This might come in handy when using the Copy History or Transfer History batch files.

Required Environment Variables Table

Variable	Definition
ACRDWB	Path where UNIRUN.exe resides (\$BASE\Cobol)
RPT	Path where the report files will be generated
UNIDF	Path and file name of the definition database.
UNIHF	Path and file name of the history database.

Run-time Parameter Table (PARM options)

Option	Use
-- blank --	Initialize both the definition and history databases.
DF	Initialize the definition database only.
HF	Initialize the history database only.

udsmr01.bat - Management Reporting

This batch file allows you to process reconciliation information in the history database and produce management reports based on user-defined criteria. You can use this utility to analyze the functioning of the system. For example, you can run a report showing all the history keys that completed reconciliation with return codes greater than 0 (all out-of-balance jobs) for a particular time period.

You can use one or more of the following criteria:

- Return code
- History key
- Absolute time

- Relative time
- Absolute cycles
- Relative cycles

Required Environment Variables Table

Variable	Definition
ACRDWB	Path where UNIRUN.exe resides (\$BASE\Cobol)
RPT	Path where the report files will be generated
UNIHF	Path and file name of the history database.

udstfint.bat - Initialize Translation Database

The reconciliation function allows you to store 40 bytes of key information. The dynamic translation feature allows you to store keys that are larger than 40 bytes. For more information, see [“Using Dynamic Translation Tables” on page 218](#)

Required Environment Variables Table

Variable	Definition
ACRDWB	Path where UNIRUN.exe resides (\$BASE\Cobol)
RPT	Path where the report files will be generated
UNITF	Path and file name of the translation file.

udstfls.bat - List Translation Database

Before you can run this batch file, you must have established a translation database by running the Initialize Translation Database batch file.

This batch file lists the contents of the dynamic translation database. It allows you to see how the key looked before it was translated (Original Source Value) and how it looks after it was translated (Identifier).

Required Environment Variables Table

Variable	Definition
ACRDWB	Path where UNIRUN.exe resides (\$BASE\Cobol)
RPT	Path where the report files will be generated
UNITF	Path and file name of the translation file.

udsupdt.bat - Update Definition Database

This batch file applies the batch transaction records to the definition database so that they can be used in ACR/Detail processing.

For more information on batch definition transactions, see [“Using Batch Definition Transactions” on page 261](#).

The batch file first sorts the batch definition transactions by transaction key and then processes them in two passes:

- First, the transactions are validated to ensure that valid interrelated information is entered in the proper columns.
- Then, if all the transactions are valid, they are processed or applied against the definition database. Using the transaction type from each transaction’s key, the batch file attempts to add, delete, replace or purge the specified transaction. If any errors occur, processing stops and no transactions are applied.

After running the update, you can use the Run Reconciliation or Run History Analysis batch file to test the definitions, or you can use the List Definitions batch file to get a formatted list of the definitions.

Required Environment Variables Table

Variable	Definition
ACRDWB	Path where UNIRUN.exe resides (\$BASE\Cobol)
RPT	Path where the report files will be generated
UNIDF	Path and file name of the definition database.
UNIINP	Path and file name of the batch definitions file.

udscsv.bat - Create Comma Delimited Text File

This batch file creates a file with field values separated by commas. Once created, the comma delimited text file can be downloaded to a PC. However, to be opened in a spreadsheet application, the PC comma delimited file must have a .csv extension. For more information, see [“Create Comma Delimited Text File”](#) on page 241.

Required Environment Variables Table

Variable	Definition
ACRDWB	Path where UNIRUN.exe resides (\$BASE\Cobol)
RPT	Path where the report files will be generated
UNIDF	Path and file name of the definition database.
UNICSV	Path and file name of the output comma delimited file.
UNIDATA	Path and file name of the extract file.

Migrating Test Definitions to Production

Overview

What are Definitions?

Definitions are data reconciliation controls including:

- Basic information
- Job, file, table, and process specifications
- Input sources
- Internal items
- Extended internal items
- History items
- Calculated items
- Rules
- Messages
- Report information

These controls have been developed by users according to their needs. These definitions should not be used in a production environment until they have been tested and proven to reconcile the data as desired.

Testing Definitions

Once you have set up definitions, you must ensure that they reconcile the data as you intended by either testing the definitions manually or running a test job and comparing the results with what you expected to find.

Migrating Test Definitions

When you are satisfied with your test definitions, it is time to migrate them into production. This process is accomplished in two stages:

- Building batch definition transactions
- Updating the production definition database.

The remaining sections in this chapter give you instructions for each of these stages.

Building Batch Definition Transactions (udsdfb.bat)

Use the instructions in “[udsdfb.bat - Build Batch Transactions](#)” on page 335 to edit and run the udsdfb.bat batch file, which will build batch transaction cards directly from the test definition database.

Updating the Production Definition Database

Use the instructions in “[udsuptd.bat - Update Definition Database](#)” on page 344 to edit and run the udsuptd.bat batch file. This batch file applies the batch transaction records to the definition database so that they can be used in ACR/Detail processing.

IMPORTANT: When you executed the udsdfb.bat batch file in the section above, a file was created and a file name was chosen.

As you execute the udsuptd.bat batch file to update production, the file name used for building batch definitions should be used as input to the UNIINP variable.

E ■ Using ACR/Detail for Windows in Batch

Migrating Test Definitions to Production

Using ACR/Detail for UNIX or Linux in Batch

When you run ACR/Detail for UNIX or Linux, scripts are executed. The job and file definitions used by the scripts are typically created through ACR/Detail Client or ACR/Workbench.

This appendix assists you in customizing the ACR/Detail scripts for your batch environment. In addition, it provides instructions for migrating your jobs from test to production.

This appendix contains the following sections:

- “Conditionally Executing a Step Based on a Return Code” on page 349
- “Before You Begin” on page 351
- “Editing and Running the Scripts” on page 352
- “Script Quick-Reference with Page Numbers” on page 354
- “Migrating Test Definitions to Production” on page 374

Conditionally Executing a Step Based on a Return Code

In a production batch stream, you may wish to conditionally execute a step based on the step completion code of an ACR/Detail step that precedes it. You can do this by editing the script. For example, following is the last part of `uds2000.sh`, the script for running reconciliation.

F ■ Using ACR/Detail for UNIX or Linux in Batch

Conditionally Executing a Step Based on a Return Code

The code in bold does the following:

- Checks for a non-zero step completion code.
- If a non-zero step completion code is found, exits at the end of the script and generates the exit status code (UNIX or Linux environment variable \$?) based on the step completion code.

```
#-----  
# run UDS2000  
#-----  
RUNCMD=${RUNCMD-cobrun} ; export RUNCMD  
CC=`$RUNCMD UDS2000 $PARM | grep "STEP COMPLETION" | awk '{print  
$4}'`  
#-----  
# cleanup transient files  
#-----  
for fname in $dd_JOBVAR $dd_UNIINP $dd_UNIWRK $dd_UNIWRK1 $dd_UNIWRK2  
$dd_UNIWRK3 $dd_UNIWRK4 $dd_UNISRT $dd_UNITRACE  
do  
  if [ -f $fname ]  
  then  
    rm $fname  
  fi  
done  
if [ $CC -gt 0 ]  
then exit $CC  
fi
```

After the script runs, one of the following will happen:

- If the job ended with a 0 step completion code, the next script will start. The 0 step completion code will be displayed on the screen unless you are redirecting the command to a sysout.rpt. UNIX or Linux will store a 0 exit status code.
- If the job exited with a non-zero return step completion code, this step completion code will be displayed on the screen unless you are redirecting the command to a sysout.rpt. UNIX or Linux will store the exit status code based on the following two-step calculation:

$$\begin{aligned} \text{Step completion code} / 256 &= x.y \\ 256 * y &= \text{exit status code} \end{aligned}$$

where x is the whole number and y is the remainder

For example, suppose the step completion code is 3000. The exit status code will be calculated as 184. Following are the calculations:

$$\begin{aligned} 3000 / 256 &= 11.71875 \\ 256 * .71875 &= 184 \end{aligned}$$

Before You Begin

This section gives you important information that you should know before attempting to set up and execute any of the scripts.

Location of the Environment File (acrprf)

Before you run a script, you must ensure that the environment file, `acrprf`, is accessible.

`acrprf` was created in the administrative account's `$HOME/bin` directory and was copied to `installdirectory/sumdet/sv/bin` directory.

In order for `acrprf` to be accessible, one of the following must be true:

- The file must be in the same directory as the script.
- The administrative account's `bin` directory must be in the executor's (of the scripts) path.
- The line `./acrprf` in the script must be modified to incorporate the absolute pathname of the administrative account's `bin` directory.

Changing User Options

If you want to customize the user options, you can edit the user options (`UNIUF.dat`) file using an editor such as `vi`. By default this file is located in the `installdirectory/license/files` directory on the UNIX or Linux host. The modifications that are made to these records will apply to all scripts and to all users of ACR/Detail on this UNIX or Linux host. For more information, see [“Setting User Options on UNIX or Linux” on page 174](#).

Increase the Maximum File Size (Optional)

ACR/Summary and ACR/Detail ship with environment variable `FILEMAXSIZE=4`. This setting will be appropriate for users who always use a file input size of 4GB or less and whose definition and history databases are both less than 1GB.

If you use input files greater than 4GB or if either of your databases is greater than 1GB, you will have to override this value by setting `FILEMAXSIZE=8` (no limit) in the `extfh` configuration file.

To override the environment variable `FILEMAXSIZE`, create the file `$COBDIR/etc/extfh.cfg` with the following contents:

```
[XFH-DEFAULT]
FILEMAXSIZE=8
IDXFORMAT=8
FILEPOINTERSIZE=8
```

Editing and Running the Scripts

Conventions for Editing Scripts

Network Locations

For files on a local area network, you can enter either the mapped drive letter or the UNC name. For example:

```
J:\LANDATA\MYFILE.DAT
or
\\SERVER1\LANDATA\MYFILE.DAT
```

Question Marks Indicate Required Edits

Replace question marks in the script with appropriate path, file name, path and file name, or parameter values.

Instructions for Editing Scripts

In addition to the following general procedure, this chapter contains a section with documentation specific to each script. Please refer to that section when editing and running the script.

1. Go to the administrative account's scripts subdirectory.
2. Make a copy of the script to modify or use as a template. (Do not change the original scripts in `installdir/sumdet/scripts`.)

3. Set the environment variables as indicated by the comments in the script and the Required Environment Variables Table provided in the script-specific documentation.

Following is an example of some environment variables requiring definition in a script.

```
#-----  
#Set the product and cobol variables  
#-----  
ACRWRK=????????? ; export ACRWRK  
dd_UNIHf=????????? ; export dd_UNIHf
```

4. Set any required run-time parameters based on the Run-time Parameter (PARM Options) Table provided in the script-specific documentation.

Following is an example of a run-time parameter that needs to be set:

```
#-----  
# Set run time parameter for database initialization.  
#-----  
PARM=?? ; export PARM #run parameter
```

5. Set any jobname and stepname values required by the script. For example:

```
echo JOBNAME=????????? >>$dd_JOBVAR # must be the JOBNAME  
echo STEPNAME=????????? >>$dd_JOBVAR # must be the STEPNAME
```

A few scripts, such as uds2000.sh (Run Reconciliation), require additional edits, which are described in the script-specific documentation.

Script Quick-Reference with Page Numbers

The scripts are documented in the following sections:

- “udpcgen.sh - Build Extraction Program” on page 355
- “udpcomp.sh - Compile COBOL Source” on page 356
- “udptest.sh - Execute Extraction Program Test” on page 356
- “udpuser.sh - Test User Program” on page 357
- “uds2000.sh - Run Reconciliation” on page 359
- “uds6000.sh - History Analysis” on page 363
- “udsdfb.sh - Build Batch Transactions” on page 364
- “udsdfc.sh - List Definitions” on page 364
- “udshfa.sh - Suspense Aging” on page 365 “udshfr.sh - Reorganize History” on page 367
- “udshfc.sh - Copy History” on page 365
- “udshfd.sh - Delete History” on page 366
- “udshfe.sh - Extract History” on page 366
- “udshfl.sh - List History” on page 367
- “udshft.sh - Transfer History (Selected Records)” on page 368
- “udshftu.sh - Unload History” on page 368
- “udshftr.sh - Reload History” on page 369
- “udshfu.sh - Update History” on page 370
- “udsinit.sh - Initialize Databases” on page 370
- “udsmr01.sh - Management Reporting” on page 371
- “udstfint.sh - Initialize Translation Database” on page 372
- “udstfls.sh - List Translation Database” on page 372
- “udsuptd.sh - Update Definition Database” on page 373
- “udscsv.sh - Create Comma Delimited Text File” on page 374

udpcgen.sh - Build Extraction Program

This script is used to generate COBOL extraction programs that correspond to file definitions. The generated programs are an integral part of program interface processing. For more information see “[Extraction Program Interface](#)” on page 318. The programs are compiled using `udpcomp.sh`. See “[udpcomp.sh - Compile COBOL Source](#)” on page 356.

To use the script, see “[Editing and Running the Scripts](#)” on page 352 and refer to the following information.

Required Environment Variables Table

Variable	Definition
ACRWRK	Default working directory. All report files will be generated in the default working directory unless you modify the paths for individual reports.
dd_UNIDF	Path and file name of the definition database.
dd_UNICBL	Path and file name of the program ID.
dd_UNICBLS	Path and file name of the file definition ID.

Run-time Parameter Table (PARM options)

Option	Use
FID - File Definition ID	The name of the file that contains your file definitions.
PID - Program ID	The name that you assign to the compiled program. The resulting file name will be followed by a <code>.cbl</code> extension. If this option is left blank, it will default to the FID name that you specified.

udpcomp.sh - Compile COBOL Source

This script is used to compile programs generated from ACR/Summary and ACR/Detail Server when using the extraction program interface. For more information, see “[udpcgen.sh - Build Extraction Program](#)” on page 355 and “[Extraction Program Interface](#)” on page 318.

This script assumes that you have a compatible Micro Focus compiler.

The script can be run using one of the following formats:

- `udpcomp.sh` in interactive mode. You will be prompted for the directory and source file.
- `udpcomp.sh /pathname/filename`. This is non-interactive mode. You will receive no prompts.

udptest.sh - Execute Extraction Program Test

This script is used to test generated extraction programs in program interface mode, prior to modifying application programs with application program interface calls. It can process input files with either physical sequential or user program dataset organizations. User Program dataset organizations call external COBOL programs which can be written to access input sources that ACR/Detail does not directly access.

See “[Extraction Program Interface](#)” on page 318 for instructions on creating the programs to be tested by this script.

Required Environment Variables Table

Variable	Definition
ACRWRK	Default working directory. All report files will be generated in the default working directory unless you modify the paths for individual reports.
dd_UNIDF	Path and file name of the definition database.
dd_UNIHF	Path and file name of the history database.
dd_UNIDATA	Required if you specify the Write Reconciliation Output File option in your Basic Control Task information.
DWBIOBUFF	This variable is commented out in the template batch file. When ACR/Detail needs to read records that are greater than 256 bytes, uncomment and set to 1.

Run-time Parameter Table (PARM options)

Option	Use
-- blank --	Will default the cycle number to 1.
Cycle Number Override	Sets the cycle number according to your specification. Example: CYC00960101

udpuser.sh - Test User Program

This script is used to test user programs in program interface mode, prior to modifying application programs with API calls. It can process input files with either physical sequential or user program dataset organizations. User program dataset organizations call external COBOL programs which provide access to input sources that ACR/Detail does not directly access.

F ■ Using ACR/Detail for UNIX or Linux in Batch

udpuser.sh - Test User Program

Required Environment Variables Table

Variable	Definition
ACRWRK	Default working directory. All report files will be generated in the default working directory unless you modify the paths for individual reports.
dd_UNIDF	Path and file name of the definition database.
dd_UNIHF	Path and file name of the history database.

Run-time Parameter Table (PARM options)

Option	Use
Parm ID	The name (8 characters or less) of the parameter that you wish to run.
EBCDIC Parm Option	An option indicating if the parm is in EBCDIC or not.
Parm Data	The contents (36 characters or less) of the parm itself.

Additional Instructions

Set the user application name by replacing the RUNCMD question marks with the program ID of your user application:

```
$RUNCMD ??????????
```

uds2000.sh - Run Reconciliation

This batch file is used to run reconciliation, which extracts the specified data from the application input sources, processes your rules, and reports on the results.

Required Environment Variables Table

Variable	Definition
ACRWRK	Default working directory. All report files will be generated in the default working directory unless you modify the paths for individual reports.
dd_UNIDF	Path and file name of the definition database.
dd_UNIHF	Path and file name of the history database.
DWBIOBUFF	This variable is commented out in the template batch file. When ACR/Detail needs to read records that are greater than 256 bytes, uncomment and set to 1.

NOTE: When using external translation tables, you must set the file IDs and specify the paths of the files the table will use.

F ■ Using ACR/Detail for UNIX or Linux in Batch

uds2000.sh - Run Reconciliation

Run-time Parameter Table (PARM options)

Examples:

PARM=ROPT==N.....

PARM=CYC#00890815,NOEXTR,NOSORT

Keyword	Possible Values
'JNAM==jjjjjjj'	Jobname = X(8) (max)
'STEP==sssssss'	Stepname = X(8) (max)
'CYC#ccccccc'	Cycle Number = 9(8) (Must be numeric and > 0)
'ROPT==abcdefgh' (Processing Options)	<p>a¹ = Store History Option (Y,N or X or (.))</p> <p>b¹ = Print Reconciliation Control Report Option (Y,N,X,Z or (.))</p> <p>c¹ = Print Reconciliation Recap Report Option (Y,N,X,Z or (.))</p> <p>d¹ = Set Return Code Option (Y,N,X or (.))</p> <p>e¹ = Print Extracted Report Option (Y,N or (.))</p> <p>f¹ = Rewrite History Option (Y,N or (.))</p> <p>g¹ = Print Accumulated/Sorted Report (Y,N,X or (.))</p> <p>h¹ = Print History Option (Y,N or (.))</p>
'ROPT2=abcdefgh'	<p>a¹ = Write Output File (Y,N,X,Z or (.))</p> <p>b = Reserved (.)</p> <p>c² = Key Sort Order (A, D, N, or (.))</p> <p>d-h = Reserved (.)</p>
<p>1 - Y = Applies to all keys. N = Applies to no keys. X = Applies to out-of-balance keys only. Z = Applies to in-balance keys only. (.) = Do not override. Use the information in the definition database.</p> <p>2 - A = Sort in ascending order. D = Sort in descending order. N = Do not sort. (.) = Do not override (use the information in the definition database). (.) = Do not override (use the information in the definition database).</p> <p>Re: Reconciliation Level Qualifier (RLQ) jobs: The Key Sort Order settings from the base job will be applied to RLQ jobs. Likewise, any run-time overrides set for the Key Sort Order setting in the base job will be applied to RLQ jobs. Therefore, in an RLQ job, setting up a run-time override for the key sort order is not allowed.</p>	
'SYSOUT'	SYSOUT (A SYSOUT report will be created even if no #U messages are generated. The report will contain an informational header message: "#USY0001: INFOGIX JOB STARTED.")
'BYPASS'	BYPASS (The USD2000 program will bypass reconciliation processing and return a step completion code of 0000.)
'NEGATE'	NEGATE (Reverse numeric additions to the history by subtracting the values extracted from the input(s))

Keyword	Possible Values
'NOEXTR'	NOEXTR (The data is already extracted into the workfile (UNIWRK). Used for backout and multiple history updates. If this parameter is used without the NOSORT parameter, then extraction writes directly to work file 2.
'NOSORT'	NOSORT (NOSORT and NOEXTR options can save time if the work files are available to be added to history. If you specify this parameter, sorting and the second round of accumulations are omitted. This speeds up processing when items are already in order.)
'NOHIST'	NOHIST (Helps in batching multiple jobs for application to the history file in one job stream)
'CRXML=n'	Control Report XML Option where n is one of the following: 1 - do not generate the Control Report in XML format 2 - generate, write to file 3 - generate, autoload to database
'URXML=n'	User Report XML Option where n is one of the following: 1 - do not generate the User Report in XML format 2 - generate, write to file 3 - generate, autoload to database
'FRXML=n'	Free-Form Report XML Option where n is one of the following: 1 - do not generate the Free-Form Report in XML format 2 - generate, write to file 3 - generate, autoload to database
'DEBUG=abcdefgh'	a ³ = XML Writer Debug Option (Y, N, blank or (.)) b ³ = Visibility API Debug Option (Y, N, blank or (.)) c-h = Reserved (.)
3 - Y = Debug on, N = blank, or (.) = Debug off.	

Additional Instructions

Specify the Input Sources

For input source types other than relational database tables, specify the path and file name for each input source.

F ■ Using ACR/Detail for UNIX or Linux in Batch

uds2000.sh - Run Reconciliation

If your input is coming from a relational database, it is not necessary to set an input source. Simply comment out the statement from the Input Sources section of the batch file using a #.

Concatenation of Input Sources

When you specify the paths and file names for the input sources, you can optionally concatenate as many as 35 files in a single dd for an input source. (You cannot, however, concatenate files that make up an external translation table.) The path names must be separated by colons. In general, the maximum length of all concatenated pathnames is 8192 characters. For more information on concatenation, see the comments in the script.

For record lengths greater than 256, uncomment the DWBIOBUFF parameter.

uds6000.sh - History Analysis

This script extracts, analyzes, and reports on selected history data. A history key or an accumulation of history keys in the current cycle of data is compared to prior cycles of data and then displayed in the report.

Required Environment Variables Table

Variable	Definition
ACRWRK	Default working directory. All report files will be generated in the default working directory unless you modify the paths for individual reports.
dd_UNIDF	Path and file name of the definition database.
dd_UNIHF	Path and file name of the history database.

udsdfb.sh - Build Batch Transactions

This script builds batch transaction cards directly from the definition database. For more about batch transactions records, see “[Using Batch Definition Transactions](#)” on page 261.

Any or all of the records on the definition database may be selected. The script creates all the transaction records in the database or those associated with the selected definition type.

Required Environment Variables Table

Variable	Definition
ACRWRK	Default working directory. All report files will be generated in the default working directory unless you modify the paths for individual reports.
dd_UNIDF	Path and file name of the definition database.

udsdfb.sh - List Definitions

This script creates a formatted list of definitions from the definition database for audit trail or troubleshooting purposes. The script lists all the transaction records in the database or those associated with the selected definition type.

Required Environment Variables Table

Variable	Definition
ACRWRK	Default working directory. All report files will be generated in the default working directory unless you modify the paths for individual reports.
dd_UNIDF	Path and file name of the definition database.

udsdhfa.sh - Suspense Aging

This script reports on aging information for values contained in the history database for suspense applications.

You have a history database that consists of history records that are in suspense. These have an internal item which defines the date on which the item went into suspense. The aging report summarizes the contents of the database (for all keys or for some level of the key if it contains a meaningful structure) in terms of the age of specified items. Each aging column can be specified in ranges of days, months, or years in any combination. An aging column for greater than a specific aging period can also be specified.

Required Environment Variables Table

Variable	Definition
ACRWRK	Default working directory. All report files will be generated in the default working directory unless you modify the paths for individual reports.
dd_UNIDF	Path and file name of the definition database.

udshfc.sh - Copy History

This script copies all records from one history database to another. The source history database and the target history database are specified by you. This script can be useful for backup purposes or to create a copy of a production database to be used in a test environment.

Required Environment Variables Table

Variable	Definition
ACRWRK	Default working directory. All report files will be generated in the default working directory unless you modify the paths for individual reports.
dd_UNIHF	Path and file name of the history database.
dd_UNIHF2	Path and file name of the new history database.

udshfd.sh - Delete History

Use this utility to perform a physical delete of one or multiple history records based on a combination of key, cycle ID, job ID, and date criteria. Wildcards can be used to select multiple keys and multiple job IDs. For more information, see [“Delete History” on page 235](#).

Required Environment Variables Table

Variable	Definition
ACRWRK	Default working directory. All report files will be generated in the default working directory unless you modify the paths for individual reports.
dd_UNIHF	Path and file name of the history database.

udshfe.sh - Extract History

This utility allows you to create a sequential file (known as an extract file) of history records from the history database. For more information, including the record layout, see [“Extract History” on page 236](#).

Required Environment Variables Table

Variable	Definition
ACRWRK	Default working directory. All report files will be generated in the default working directory unless you modify the paths for individual reports.
dd_UNIHF	Path and file name of the history database.

udshfl.sh - List History

This script prints a formatted summary of the records in the history database for purposes such as audit trail and troubleshooting. The history database to list is specified as an environment variable.

Required Environment Variables Table

Variable	Definition
ACRWRK	Default working directory. All report files will be generated in the default working directory unless you modify the paths for individual reports.
dd_UNIHF	Path and file name of the history database.

udshfr.sh - Reorganize History

This script reorganizes the contents of a history database in place. Detail history for any given history key is kept physically in order using relative cycle number. During the reorganization process, all logically deleted histories are permanently erased. The script rebuilds relative cycle numbers based on the actual cycle ID values found for any history key. Reorganizing history results in a more efficient file.

Required Environment Variables Table

Variable	Definition
ACRWRK	Default working directory. All report files will be generated in the default working directory unless you modify the paths for individual reports.
dd_UNIDF	Path and file name of the definition database.
dd_UNIHF	Path and file name of the history database.

udshft.sh - Transfer History (Selected Records)

This script transfers selected records from an existing history database directly to another file. History records can be selected by history key and by date and/or time.

Refer to “[Editing and Running the Scripts](#)” on page 352 and to the following tables when editing this script.

Required Environment Variables Table

Variable	Definition
ACRWRK	Default working directory. All report files will be generated in the default working directory unless you modify the paths for individual reports.
dd_UNIHF2	Path and file name of the output history database.
dd_UNIDF	Path and file name of the definition database.
dd_UNIHF	Path and file name of the history database.
dd_UNIHFw	Path and file name of the work file.

udshftu.sh - Unload History

This script unloads a history database to a flat file (or physical sequential file). History records can be selected by history key and by date and/or time.

This utility might be used to:

- Backup/restore files
- Move a history database from one host to another

After you have used this utility, if you want to reload the unloaded records to another database, see “[udshftr.sh - Reload History](#)” on page 369.

Refer to “[Editing and Running the Scripts](#)” on page 352 and to the following tables when editing this script.

Required Environment Variables Table

Variable	Definition
ACRWRK	Default working directory.
dd_UNIDF	Path and file name of the definition database.
dd_UNIHF	Path and file name of the history database.
dd_UNIHF2	Path and file name of a flat (physical sequential) file to which the records will be unloaded.

udshftr.sh - Reload History

This utility can be used after you have used the Unload History utility (see [“udshftu.sh - Unload History” on page 368](#)). It reloads the previously unloaded history records to an empty database that you have already created. History records can be selected by history key and by date and/or time.

Refer to [“Editing and Running the Scripts” on page 352](#) and to the following tables when editing this script.

Required Environment Variables Table

Variable	Definition
ACRWRK	Default working directory. All report files will be generated in the default working directory unless you modify the paths for individual reports.
dd_UNIHF2	Path and file name of the output history database.
dd_UNIDF	Path and file name of the definition database.
dd_UNIHF2	The work file from which the records will be reloaded.

udshfu.sh - Update History

This script updates (replaces) one or more control values in the history database for a specified key and cycle ID. Up to 99 control values may be updated for each run of this script. This utility can be used to correct out-of-balance situations when a job runs unsuccessfully and you later obtain the proper data to run your job.

Each control value to be updated must be in one of two formats:

- A numeric value
- Up to 8 characters of text

This script can be used to correct out-of-balance situations when a job runs unsuccessfully and you later obtain the proper data to run your job.

Required Environment Variables Table

Variable	Definition
ACRWRK	Default working directory. All report files will be generated in the default working directory unless you modify the paths for individual reports.
dd_UNIHF	Path and file name of the history database.

udsinit.sh - Initialize Databases

This script initializes the definition and/or history database of your choice, erasing the contents. Be careful to specify the correct database(s) so that you do not accidentally erase data that you need. You can also use this script to create a new definition and/or history database by specifying a new file name to initialize. This might come in handy when using the Copy History or Transfer History scripts.

Required Environment Variables Tables

Variable	Definition
ACRWRK	Default working directory. All report files will be generated in the default working directory unless you modify the paths for individual reports.
dd_UNIDF	Path and file name of the definition database.
dd_UNIHF	Path and file name of the history database.

Run-time Parameter Table (PARM options)

Option	Use
-- blank --	Initialize both the definition and history databases.
DF	Initialize the definition database only.
HF	Initialize the history database only.

udsmr01.sh - Management Reporting

This script allows you to process reconciliation information in the history database and produce management reports based on user-defined criteria. Managers can use this utility to analyze the functioning of the system. For example, you can run a report showing all the history keys that completed reconciliation with return codes greater than 0 (all out-of-balance jobs) for a particular time period.

You can use one or more of the following criteria:

- Return code
- History key
- Absolute time
- Relative time
- Absolute cycles
- Relative cycles

Required Environment Variables Table

Variable	Definition
ACRWRK	Default working directory. All report files will be generated in the default working directory unless you modify the paths for individual reports.
dd_UNIHF	Path and file name of the history database.

udstfint.sh - Initialize Translation Database

The reconciliation function allows you to store 40 bytes of key information. The dynamic translation feature allows you to store keys that are larger than 40 bytes. For more information, see [“Using Dynamic Translation Tables” on page 218](#)

Required Environment Variables Table

Variable	Definition
ACRWRK	Default working directory. All report files will be generated in the default working directory unless you modify the paths for individual reports.
dd_UNITF	Path and file name of the translation file.

udstfls.sh - List Translation Database

Before you can run this script, you must have established a translation database by running the Initialize Translation Database script.

This script lists the contents of the dynamic translation database. It allows you to see how the key looked before it was translated (Original Source Value) and how it looks after it was translated (Identifier).

Required Environment Variables Table

Variable	Definition
ACRWRK	Default working directory. All report files will be generated in the default working directory unless you modify the paths for individual reports.
dd_UNITF	Path and file name of the translation file.

udsupdt.sh - Update Definition Database

This script applies the batch transaction records to the definition database so that they can be used in ACR/Detail processing.

For more information on batch definition transactions, see [“Using Batch Definition Transactions”](#) on page 261.

The script first sorts the batch definition transactions by transaction key and then processes them in two passes:

- First, the transactions are validated to ensure that valid interrelated information is entered in the proper columns.
- Then, if all the transactions are valid, they are processed or applied against the definition database. Using the transaction type from each transaction’s key, the script attempts to add, delete, replace or purge the specified transaction. If any errors occur, processing stops and no transactions are applied.

After running the update, you can use the Run Reconciliation or Run History Analysis script to test the definitions, or you can use the List Definitions script to get a formatted list of the definitions.

Required Environment Variables Table

Variable	Definition
ACRWRK	Default working directory. All report files will be generated in the default working directory unless you modify the paths for individual reports.
dd_UNIDF	Path and file name of the definition database.
dd_UNIINP	Path and file name of the batch definitions file.

udscsv.sh - Create Comma Delimited Text File

This script creates a file with field values separated by commas. Once created, the comma delimited text file can be downloaded to a PC. However, to be opened in a spreadsheet application, the PC comma delimited file must have a .csv extension. For more information, see [“Create Comma Delimited Text File” on page 241](#).

Required Environment Variables Table

Variable	Definition
ACRWRK	Default working directory. All report files will be generated in the default working directory unless you modify the paths for individual reports.
dd_UNIDF	Path and file name of the definition database.
dd_UNICSV	Path and file name of the output comma delimited file.
dd_UNIDATA	Path and file name of the extract file.

Migrating Test Definitions to Production

Overview

What are Definitions?

Definitions are data reconciliation controls including:

- Basic information
- Job, file, table, and process specifications

- Input sources
- Internal items
- Extended internal items
- History items
- Calculated items
- Rules
- Messages
- Report information

These controls have been developed by users according to their needs. These definitions should not be used in a production environment until they have been tested and proven to reconcile the data as desired.

Testing Definitions

Once you have set up definitions, you must ensure that they reconcile the data as you intended by either testing the definitions manually or running a test job and comparing the results with what you expected to find.

Migrating Test Definitions

When you are satisfied with your test definitions, it is time to migrate them into production. This process is accomplished in two stages:

- Building batch definition transactions
- Updating the production definition database.

The remaining sections in this chapter give you instructions for each of these stages.

Building Batch Definition Transactions (udsdfb.sh)

Use the instructions in [“udsdfb.sh - Build Batch Transactions”](#) on page 364 to edit and run the udsdfb.sh script, which will build batch transaction cards directly from the test definition database.

Updating the Production Definition Database

Use the instructions in “[udsupdt.sh - Update Definition Database](#)” on [page 373](#) to edit and run the `udsupdt.sh` script. This script applies the batch transaction records to the definition database so that they can be used in ACR/Detail processing.

IMPORTANT: When you executed the `udsdfb.sh` script in the section above, the variable `dd_UNIUPDT` specified a default output file called `UNIUPDT.dat`.

As you execute the `udsupdt.sh` script to update production, `dd_UNIINP` must be set to use the `UNIUPDT.dat` file.

Non-Biased Variance and Standard Deviation Computations

The history analysis feature includes two computation types available in the **Column Type** field on the Columns dialog box that can be used to calculate a “non-biased” or “moving range” variance or standard deviation across a range of columns or rows. These computation types are appropriate for calculations based on a sample rather than on an entire population. These computation types provide additional options for the field **Select a computation to perform across range**.

This appendix contains the following sections:

- “Field Descriptions and Formulas” on page 377
- “Examples” on page 379

Field Descriptions and Formulas

The following field definitions will help you understand the differences between **Variance** and **Standard Deviation** computation types and **Variance (Non-Biased)** and **Standard Deviation (Non-Biased)** computation types. For additional information, see “Examples” on page 379, after reading the field descriptions.

Note: In the formulas shown here:

- n represents the number of values included in the computation.
 - x represents each value used in the computation.
 - The symbol \sum represents summation.
-

Variance Field

Choose the **Variance** computation type to calculate the variance if the columns (or rows) you are using in your computation make up the entire population of values. For example, use this computation type if your population consists of 10 cycles and you are using the values of all 10 cycles to calculate the variance. This computation type uses the following formula:

G ■ Non-Biased Variance and Standard Deviation Computations

Field Descriptions and Formulas

$$\text{Variance} = \frac{n \sum x^2 - (\sum x)^2}{n^2}$$

Variance (non-Biased) Field

Choose the **Variance (non-biased)** computation type to calculate the variance if the columns (or rows) you are using in your computation are a *sample of the entire population* of values. For example, use this computation type if your population consists of 100 cycles and you are using a sample made up of 10 of these cycles to calculate the variance. This computation type uses the following formula:

$$\text{Variance (Non - Biased)} = \frac{n \sum x^2 - (\sum x)^2}{n(n - 1)}$$

Standard Deviation Field

Choose the **Standard Deviation** field to calculate the standard deviation if the columns (or rows) you are using in your computation make up the *entire population* of values. For example, use this field if your entire population consists of 10 cycles and you are using the values of all 10 cycles to calculate the standard deviation. This computation type uses the following formula:

$$\text{Standard Deviation} = \sqrt{\frac{n \sum x^2 - (\sum x)^2}{n^2}}$$

Standard Deviation (non-Biased) Field

Choose the **Standard Deviation (non-biased)** computation type to calculate the standard deviation if the columns (or rows) you are using in your computation are a *sample of the entire population* of values. For

example, use this computation type if your population includes 100 cycles and you are using a sample consisting of 10 of these cycles to calculate the variance. This computation type uses the following formula:

$$\text{Standard Deviation (Non - Biased)} = \sqrt{\frac{n \sum x^2 - (\sum x)^2}{n(n-1)}}$$

Examples

Assume you are interested in using the History Analysis feature to create a report for 10 cycles having the following values:

1345	1370
1301	1318
1368	1350
1322	1303
1310	1299

We will use these values to show how you can appropriately use the computation types described previously.

Using the Variance Computation Type to Compute the Variance of A Population

If the 10 cycles you are using in your computation make up the entire population, calculate the variance using the **Variance** computation type, which uses the following formula:

$$\text{Variance} = \frac{n \sum x^2 - (\sum x)^2}{n^2}$$

G ■ Non-Biased Variance and Standard Deviation Computations

Examples

The variance will be 678.84, as demonstrated in the following table:

x	x^2	$\sum x^2$	$n\sum x^2$	$(\sum x)^2$	$n\sum x^2 - (\sum x)^2$	$\frac{n\sum x^2 - (\sum x)^2}{n^2}$
1345	1809025					
1301	1692601					
1368	1871424					
1322	1747684					
1310	1716100					
1370	1876900					
1318	1737124					
1350	1822500					
1303	1697809					
1299	1687401					
		17658568	176585680	176517796	67884	678.84

Using the Variance (Non-Biased) Computation Type to Compute the Variance of a Sample

If the 10 cycles you are using in your computation are a sample of the entire population, calculate the variance using the **Variance (Non-Biased)** computation type, which uses the following formula:

$$\text{Variance (Non - Biased)} = \frac{n\sum x^2 - (\sum x)^2}{n(n - 1)}$$

G ■ Non-Biased Variance and Standard Deviation Computations

Examples

The variance will be 754.27, as demonstrated in the following table:

x	x ²	$\sum x^2$	$n \sum x^2$	$(\sum x)^2$	$n \sum x^2 - (\sum x)^2$	$\frac{n \sum x^2 - (\sum x)^2}{n(n-1)}$
1345	1809025					
1301	1692601					
1368	1871424					
1322	1747684					
1310	1716100					
1370	1876900					
1318	1737124					
1350	1822500					
1303	1697809					
1299	1687401					
		17658568	176585680	176517796	67884	754.27

G ■ Non-Biased Variance and Standard Deviation Computations

Examples

Using the Standard Deviation Computation Type to Compute the Standard Deviation of a Population

If the 10 cycles you are using in your computation make up the entire population, calculate the standard deviation using the **Standard Deviation** computation type, which uses the following formula:

$$\text{Standard Deviation} = \sqrt{\frac{n \sum x^2 - (\sum x)^2}{n^2}}$$

The standard deviation will be 26.054, as demonstrated in the following table:

x	x ²	$\sum x^2$	$n \sum x^2$	$(\sum x)^2$	$n \sum x^2 - (\sum x)^2$	$\sqrt{\frac{n \sum x^2 - (\sum x)^2}{n^2}}$
1345	1809025					
1301	1692601					
1368	1871424					
1322	1747684					
1310	1716100					
1370	1876900					
1318	1737124					
1350	1822500					
1303	1697809					
1299	1687401					
		17658568	176585680	176517796	67884	26.054

Using the Standard Deviation (Non-Biased) Computation Type to Compute the Standard Deviation of a Sample

If the 10 cycles you are using in your computation are a sample of the entire population, calculate the standard deviation using the **Standard Deviation (Non-Biased)** computation type, which uses the following formula:

$$\text{Standard Deviation (Non - Biased)} = \sqrt{\frac{n \sum x^2 - (\sum x)^2}{n(n - 1)}}$$

The standard deviation will be 27.463, as demonstrated in the following table:

x	x ²	$\sum x^2$	$n \sum x^2$	$(\sum x)^2$	$n \sum x^2 - (\sum x)^2$	$\frac{n \sum x^2 - (\sum x)^2}{n(n - 1)}$
1345	1809025					
1301	1692601					
1368	1871424					
1322	1747684					
1310	1716100					
1370	1876900					
1318	1737124					
1350	1822500					
1303	1697809					
1299	1687401					
		17658568	176585680	176517796	67884	27.463

G ■ Non-Biased Variance and Standard Deviation Computations

Examples

Glossary

A

access mode

This file organization method determines how the input source file is read and processed.

■ **Access Mode 6 (Keys Precede Detail Values)** processes groups from top to bottom for each record. This is the standard mode.

■ **Access Mode 7 (Keys Follow Detail Values)** keeps track of the first selection criteria with a *not found* condition and starts with that selection group's criteria for the next record. Each group is evaluated because the system wraps around the bottom to the top of your criteria, as necessary.

Accumulated Data Detail Report

This detail report shows the work record data after all input files are processed and after Stage 2 accumulation is complete. The report is typically used for testing/debugging.

accumulation

Accumulation is the ACR/Detail process that mathematically adds values in counts or amounts to obtain a total for a key break. See also stage 1 accumulation, stage 2 accumulation, and stage 3 accumulation.

ACR/Connector

The ACR/Connector product extends the capabilities of ACR/Summary and ACR/Detail across multiple platforms, processors, and geographic locations. Its cross-platform features include history item retrieval and synchronization of jobs and other processes.

ACR/Summary and ACR/Detail server

This server is the interface between the Workbench/client and the batch portion of the application, which resides on the platform. The main function of this server is to execute the job and return the results to the Workbench/client. The server can run as a service or a daemon.

active rule

See standard rule and conditional rule.

alternate control task (job) ID

See alternate job ID.

alternate job ID

An alternate job ID is a job/step without any associated rules that is used to execute a different job ID. An alternate job ID is used primarily for testing and applies only to jobs run on z/OS or in batch.

analysis rule

See fluctuation analysis rule and threshold analysis rule.

application program interface

An application program interface (API), is the tool that an application program uses to initialize the extraction program interface, send data to be extracted, and invoke the reconciliation of all extracted data.

assignment rule

This type of lookup rule specifies the values to assign when the selection rule is satisfied. For example, if an untranslated value equals a value in the translation table, translate that value to the value of another column from the table.

auto cycle indicator

This option, under Basic Information in the Control Task (Job) View, lets you choose if and how to automatically format the cycle number.

B

balancing rule

Balancing rules consist of standard and conditional rules. These rules determine if your reports, files, and other information sources are in or out of balance.

batch definition transaction record

This is an 80-character record, stored in the transaction library, and used to apply changes to definitions in batch. Each record contains information for a particular definition and a transaction type code that specifies how to apply the transaction to the definition database.

build rules

See table build rules.

C

calculated item

A calculated item specifies mathematical manipulation of values that can include internal items, extended internal items, history items, other calculated items, or literals. Calculated items can be used in rules, on Free-form Reports, on User Reports, or in the free-form output file.

column assignment

This table build rule extracts column values from the source data and loads them into the translation table.

comma delimited text file

This is a text file that contains sequential fields separated by commas and text values surrounded by quotation marks. The file can be read by most PC applications, such as spreadsheets.

common internal item

Common internal items are internal items defined at the control entity level. These items can be shared by multiple control tasks.

completion code

A completion code is a system code that indicates whether the job completed or why the job did not complete. See also return code.

conditional rule

A conditional rule uses IF/THEN logic to determine if a condition is true or false based on the combined status of up to 10 other rules. An active conditional rule is evaluated, and if the condition is met, takes one of the following actions: activates a calculation, sets a return code, or activates other rules. An inactive conditional rule takes no action.

configuration

Configuration is part of the initial setup. Configuration identifies a server platform or mode, the definition and history databases, a path for storing reports, and any other platform-specific information. You can specify multiple configurations and switch between them as needed.

control entity

A control entity is a group of common keys, internal items, and related control tasks. A control entity also provides alternate control tasks and table information. A control entity allows multiple control tasks to refer to the same items in history.

Control Report

This report is the primary output of a job run. It shows both the components and the results of the run. By default, the system automatically generates this report.

control task (job)

A control task contains the specifications entered for a job. See also job definition.

control task (job) ID

See job ID.

control task (job) name

See job name.

control value

A control value is a count, amount, text item, date, or time obtained during an extraction, or a value for an internal item, calculated item, history item, or rule input by a user for the job.

cycle ID

A cycle ID, which consists of an 8-digit cycle number and a 3-digit run number, uniquely identifies each run of a job. If numbers are not set up to increment for each cycle, the system assigns the default, 000, as the run number. Cycle IDs must be numeric and greater than zero, and they are typically processed in ascending order.

cycle number

The cycle number, which is part of the cycle ID, is an 8-digit ascending number that identifies the processing cycle for a job. Usually, cycle numbers are in *ccymmdd* format.

cycle table

A cycle table is used to verify cycle numbers. In history analysis, a cycle table can verify the cycle number of the record that has been retrieved for analysis.

D**data extraction**

This is the first phase of the reconciliation process. Data extraction uses file definitions to select, reformat, extract, and translate key and detail field values from the input sources and then sort, merge, and accumulate the values by reconciliation key.

data filter

The data filter option lets you create a temporary file in which to store out-of-balance keys for use in a subsequent qualifier.

database

See definition database or history database.

database initialization

Database initialization is the process that creates or deletes and re-creates the definition or history database.

database utilities

Database utilities are programs that are used to view, list, or maintain the definition, history, and dynamic translation databases.

DDNAME

The data definition name (DDNAME) identifies a file in the JCL. The DDNAME is equal to the file name in the file ID.

definition

A definition is a set of parameters that determines how a process is performed. Definition types include job, file, table, and history analysis.

definition database

This database is a keyed file that stores the job, file, table, and history analysis definitions.

detail field

This file definition type, for access modes 6 and 7, specifies the value to extract, whether to total or tally the extracted value, and whether to store the result in an internal item, extended internal item, or extraction variable.

detail value

A detail value is a count, amount, date, or text string extracted from an input source and used in reconciliation. Key fields or detail fields are used to extract detail values.

DSN

The dataset name (DSN or DSName) represents the catalogued name of a particular computer file.

dynamic translation table

Dynamic translation makes use of a translation database to turn strings of up to 80-bytes into standard eight-byte identifiers. A dynamic translation table allows you to store keys greater than 40-bytes. See also internal translation table, external translation table, and cycle table.

E**error message**

If the user interface cannot perform a requested function, the system displays a message that tells you what the problem is, and often, what to do about it. System messages typically begin with #U and are documented in the *Messages and Codes* guide.

exception code

This code prints next to items on the History Analysis Report that are outside of the tolerance for the analysis rule.

exception reporting

See history analysis.

extended data

Extended data is the data from extended internal items. Like data from regular internal items, extended data can be used in calculated items, history items, rules, reports, and output files.

extended internal item

An extended internal item is a storage place for a control value (count, amount, text item, date, or time) extracted or accumulated when a job is run, or a control value that results from the manipulation of item values in a calculated item. An extended internal item allows extraction and storage of a number with up to 30 digits or a text value with up to 80 characters. See also internal item.

external translation table

An external translation table is used to translate and match values across input sources. For example, you might want to translate a state name into a 2-character state abbreviation. An external translation table is built at run time, usually by referencing an external data source.

extract definition

See file definition.

extract file

An extract file lets you easily pass information between platforms or products. You can use the Extract History Utility to create a history extract file to easily move history data.

Extracted Data Detail Report

This detail report shows the work record data as it is extracted from each selected record. The report is typically used for testing/debugging.

extraction variable

An extraction variable is a storage place where regular or extended data from file definitions or external translation table definitions is held for further processing.

F

field format

Field format refers to the six choices to display selection, extraction, or output field information. Field formats include number, text, packed, signed numeric, binary, and unsigned packed.

file definition

A file definition specifies how to locate and extract values from records in the selected input files.

file ID

A file ID is a unique identifier for the input source. It consists of an 8-character file name and an optional 2-character file name qualifier that is usually used for repetitive processing of the same file during a job.

file interface mode

This is the typical mode to extract control values directly from your business application. No source code changes are required.

file item

A file item is job definition information that associates a job ID with a file ID. File items are automatically generated when you create an input source.

file name

This is the 8-character prefix of the file ID. See also DDNAME and file ID.

file name qualifier

This is an optional 2-character suffix to the file name. See also file ID.

file organization

File organization refers to the type of input file processed for the job.

filler record

Filler records are placeholders to maintain the gap between item numbers. Filler records are useful when you import history items, calculated items, and rules with non-consecutive numbers. In the Control Task view, filler records appear as @@FILLER.

flow rule

This type of lookup rule specifies the system response after a value is translated. By default, processing stops.

fluctuation analysis rule

A fluctuation analysis rule specifies the criteria to detect a data fluctuation situation. The definition specifies a range of valid data values. If values fall outside this range, an exception code or message is printed on a History Analysis Report.

FNQ

See file name qualifier.

free-form output file

This file, unidat2.dat, is an output file that can contain the data in the original output file, unidata.dat, as well as history items, calculated items, rule information, job information, and reconciliation key information. See also output file.

Free-Form Report

A Free-Form Report is a fully customizable report that can include any of the control values obtained from a job run.

G

global reformat

Global reformat reformats the data values in two input sources to the same format. This feature ensures that the data in different input files can be matched with a common reconciliation key. The reformat is performed before any selection criteria is applied.

H

hash translation

Hash translation allows a key field to be stored in and retrieved from the History database as a hashed value. The first 4 bytes are the first 4 bytes of the original key and the last 4 bytes is the hash value. The original key can be stored as an internal item.

history analysis

A history analysis job produces exception reports for data that lies outside a specified range. A history analysis job is useful for identifying trends in your balancing processes.

history analysis definition

See history analysis process definition and history analysis report definition.

history analysis ID

This is a unique name, up to 16-characters long, for the set of history analysis process definitions.

history analysis process definition

This information tells the system which cycles of which history keys to process for a History Analysis Report.

History Analysis Report

This report shows the results of a history analysis job.

history analysis report definition

This information tells the system how the report is set up, which history items to use, and specifies the rules to analyze the data.

history analysis rule

See fluctuation analysis rule and threshold analysis rule.

History Data Detail Report

This report shows history records and their values for a range of keys.

history database

This database is a keyed sequenced file that stores the results of previous runs for use in another run of the same job or in another job. Each history record contains a job ID, cycle ID, and the control values extracted for each internal or extended internal item.

history insert

History insert refers to the insertion of history records regardless of the cycle ID.

history item

This job definition type specifies that the current job will use a value that was extracted from a previous run of the current job or another job and stored in the history database.

history key

A history key is a 40-character identifier of detail values from history. The key has five segments, each eight characters in length.

history key mask

The history key mask enables a user to control which information to retrieve from the history database for use in history items. A user can *mask* the current reconciliation key by overlaying any of the five key fields with constant values. The system then retrieves the records from the history database that match the *masked* key. You can use this feature in conjunction with the reconciliation key mask feature.

I

import

You use the import method to open batch definition transaction records from z/OS in the client or ACR/Workbench.

inactive rule

See standard rule and conditional rule.

input area

When you define selection criteria, this option indicates that you will extract data from a position within the current record.

input source

An input source is typically a file or report from your application that contains the data to extract for use in a job.

integrity check

This feature examines the validity of the current job. It scrutinizes each item and its relationship to other items and identifies any inconsistencies. The system performs an integrity check if you run a job, run a history analysis, or update the definition database. You can also run an integrity check manually.

internal item

An internal item is a storage place for a control value (count, amount, text item, date, or time) extracted or accumulated when a job is run, or a control value that results from the manipulation of item values in a calculated item. An internal item allows extraction and storage of a number with up to 15 digits or a text value with up to 8 characters. See also extended internal item.

internal translation table

An internal translation table is used to translate values from an input source. The internal table consists of two columns, input and output. You define up to 999 input/output entries. This type of table is suitable for performing simple, static, one-to-one, text-to-text translations within a job.

J

job definition

A job definition defines the rules and processing options required to execute a job run.

job ID

A job ID is a compound data element that identifies a job and its definitions. It consists of an 8-character job name, an 8-character step name, and an optional 2-character qualifier.

job name

A job name is an 8-character name that, together with the step name, identifies the job. This is the first portion of the job ID. See also job ID.

job step

A job step represents one step in a multi-step job. A job step can execute an application, utility, or ACR program, which in turn invokes a job run.

K

key

See reconciliation key.

key break

A key break refers to the location in a file where a different reconciliation key is encountered. For each key break, the system extracts one value and stores the value in a temporary location based on field type.

key field

This file definition type, for access modes 6 and 7, specifies how to load one of up to five 8-character fields that together constitute the reconciliation key. Key fields can be loaded with data extracted from an input file, from text supplied in a literal, from a reformatted combination of the two, or from an extraction variable.

key mask

See history key mask or reconciliation key mask.

key segment

A key segment is an eight-character section of a reconciliation key. A reconciliation key can have up to five key segments.

L

level qualifier

See reconciliation level qualifier.

Linux mode

This mode allows you to directly access the ACR/Summary and ACR/Detail server on the Linux platform to define databases, run jobs or utilities, and view the results.

local mode

This feature of ACR/Workbench enables you to create, store, and run jobs on your PC. This can be useful if you want to download input sources from z/OS, create and test balancing jobs and their associated tables and history analysis documents on the PC through a graphical interface, and then upload the definitions and run them on z/OS.

lookup rules

Lookup rules tell the system what to do when it encounters a value to translate. There are three types of lookup rules: selection, assignment, and flow.

M**message**

A message is user-defined information associated with a return code set by a balancing rule. A message usually identifies why a step is not in balance and outlines the correction procedures. Messages print on the Control Report, and optionally, on the User Report, Free-Form Report, and z/OS console.

multi-level reconciliation

This type of reconciliation processing enables you to extract data at the maximum level of detail, and then reconcile the extracted data at a variety of key levels. A file definition specifies the most detailed level of key to use.

N**note area**

This is an 80-character line printed below each key on the Control Report and saved on the history database if the job stores history.

O**out-of-balance condition**

An out-of-balance condition exists when not all of the stipulations in your rules have been met.

out-of-balance message

See message.

output file

This file, unidata.dat, is output by the reconciliation process for use as input to other programs. This file contains the reconciliation key, cycle and run number, job, step, and qualifier name, return code, note area, internal and extended internal item count, as well as internal and extended internal items. See also free-form output file.

P**process control**

This table build rule specifies the action to take after a selection group has been satisfied.

process definition

See history analysis process definition

processing error

See error message.

process ID

See history analysis ID.

Q**qualifier**

In the job ID, qualifier is the optional 2-character field that identifies multiple invocations of a job with the same job/step name. In the file ID, qualifier is the optional 2-character identifier of the input source to include in the reconciliation run. See also reconciliation level qualifier.

R**Recap Report**

This user-defined report summarizes the results of multiple jobs.

reconciliation

Reconciliation consists of two phases: data extraction and reconciliation. It compares transaction-level data according to user-defined rules and produces reports. The reports show whether the results are in or out of balance and provide additional information to help you analyze the results and make needed corrections.

reconciliation key

A reconciliation key can consist of up to five key segments, each of which can contain up to eight positions. The segments define the criteria to locate and extract data from the input source. The first key segment determines the highest level of sort criteria.

reconciliation key mask

The reconciliation key mask is a job-level or job-step-level option that enables a user to place a constant value in any of the five key segments that make up the reconciliation key. This key mask allows the storage and retrieval of data from multiple jobs or job steps on the same history database.

reconciliation level qualifier

This two-digit level qualifier is used to indicate levels for a multi-level reconciliation job. The job has a base-level job with up to 99 subordinate-level qualifier jobs that use the same name and step name as the base job. Each qualifier has a unique level qualifier, the RLQ, to reconcile data at a more detailed level.

reconciliation rule

See balancing rule.

record layout

A record layout is a COBOL copybook that allows you to automatically calculate the position and length of the field definitions for a file to be used as an input source.

reformat field

See reformat record.

reformat record

Reformat record is both a file definition type and a table build rule. As a file definition type for access modes 6 and 7, it specifies how to rearrange or combine detail values from the input records, a literal, or an extraction variable and outputs the reformatted values to the output area or to an extraction variable. As a table build rule, it uses the same procedure to reformat a column value before using the value in a column assignment.

relative cycle

A relative cycle points to a history record of a previously run job relative to the current run. Relative cycles are numbered -998 to +000. A relative cycle of +000 refers to the current cycle, -001 refers to the prior cycle, and so forth.

relative record

Relative record is both a file definition type and a table build rule. As a file definition type, for access modes 6 and 7, it locates a record by specifying a number of records to move in relation to the last record selected. The move can be backward (toward the beginning of the file) or forward (toward the end of the file). As a table build rule, used in conjunction with selection criteria, it uses the same procedure to identify the relative position from a selected record.

Reorganize History Utility

This utility performs a physical reorganization of the history database contents. The utility maintains the physical order of the detail history for any given history key by rebuilding relative cycle numbers based on the actual cycle ID values for any history key.

report definition

See history analysis report definition.

return code

A return code is a user-defined four-digit code that indicates the result of a job run. Return codes and their associated messages appear on the Control Report, and optionally, on the User Report, Free-form Report, and z/OS console.

RLQ

See reconciliation level qualifier.

rule

A rule is a set of criteria to apply to the values of internal items, extended internal items, calculated items, or history items to determine if values extracted from an input source are in or out of balance. A single job may need to have multiple rules to determine if a variety of values are in balance. If one or more rules are out of balance, the entire job is considered out of balance. See also standard rule and conditional rule.

rule action

A rule action is the step that the system will take if a rule is out of balance.

rule type

See conditional rule and standard rule.

run

A run is a single execution of a job or process.

run number

The run number, which is part of the cycle ID, is a 3-digit ascending number that enables unique identification of multiple runs of a job with the same cycle number.

S**selection criteria**

See selection group.

selection field

Selection field is both a file definition type and a table build rule. As a file definition type for access modes 6 and 7, it locates records in the input area that contain a specified value. The specified value can be a text literal, numeric literal, or extraction variable. As a table build rule, it uses the same procedure to define criteria for selecting records from the source data.

selection group

A selection group consists of one or more consecutive selection fields that, when evaluated together, determine whether to extract data to process from the current record.

selection rule

This type of lookup rule compares the untranslated values from input/output parameters to the values in a translation table. When the selection rule is satisfied, the data from the selected table row becomes available for assignment and flow rules.

Sorted Data Detail Report

This detail report shows the work record data after it is extracted from the selected records and after stage 1 accumulation is complete. This report is typically used for testing/debugging.

stage 1 accumulation

Stage 1 accumulation is the accumulation of detail fields when detail values for the same internal item number are encountered consecutively for the same reconciliation key. The values are accumulated according to the access mode and field type specifications. See also accumulation.

stage 2 accumulation

Stage 2 accumulation occurs after all input files are processed. Any work records with identical reconciliation keys are merged into a single work record, with the detail values accumulated according to the accumulation option specified in the detail field information. See also accumulation.

stage 3 accumulation

Stage 3 accumulation is the process of accumulating values from work records and history into new history values. This type of accumulation occurs if your job is set up to store new history. See also accumulation.

standard rule

A standard rule compares values from any combination of internal, extended internal, calculated, and history items to determine if they are in or out of balance. It uses a rule equation in the left-side/operator/right-side format. An active standard rule sets a return code, performs a calculation, or abends the job. An inactive standard rule is evaluated only within a conditional rule.

status bar

This display area at the bottom of the client window shows the actions of menu items as you use the arrow keys to navigate through the menus, and messages regarding the actions of the toolbar buttons.

step name

A step name is an 8-character name that, together with the job name, identifies the job. This portion of the name enables you to assign a distinct name to each step of a multiple step job. See also job ID.

summary level information

Summary level information associates a set of history analysis process definitions with up to nine different history analysis report definitions.

Summary of Detail Histories Report

This detail report shows information on each history item in the history database including the run date, run time, and status.

Suspense Aging Report

This report shows aging information for data in suspense, which is stored in the history database.

System Messages Report

The system generates this report, SYSOUT, if you run a job, an integrity check, a database initialization or update, or a utility. This report shows the step completion code and, if errors exist, the corresponding return codes.

T

table build rules

Table build rules are selection criteria that tell the system how to extract parts of the source data to build a translation table.

table definition

A table definition identifies the table name and the type of information on the transaction record. The definition includes the specifications needed to access the data stored in a table. Table definitions types include external translation tables, internal translation tables, dynamic translation tables, and cycle tables.

table ID

See table name.

table name

A table name is a unique 16-character ID that identifies the translation or cycle table.

threshold analysis rule

A threshold analysis rule specifies the criteria to detect data values that have fallen below a specified threshold. Values that fall below the threshold are flagged as incorrect with a user-defined exception code.

Trace Report

This report has two uses. For an input source, in access mode 6 or 7, or for an external translation table, the report shows each record in an input file along with each extraction definition processed against that record. For an external translation table, the report also shows the lookup processing of the table definitions.

transaction code

This code identifies the transaction type of a batch transaction record.

transaction record

See batch definition transaction record.

translation table

See external translation table and internal translation table.

U

UNICTN utility

This utility converts caret symbols to not symbols.

UNIDATA

See free-form output file and output file.

UNINTC utility

This utility converts not symbols to caret symbols.

UNIX mode

This mode allows you to directly access the ACR/Summary and ACR/Detail server on the UNIX platform to define databases, run jobs or utilities, and view the results.

Update History Utility

This utility can change one or more control values in the history database. Use this utility to correct an out-of-balance condition.

user exit

A user exit program is COBOL source code, delivered with the product, that you can modify to perform special processing.

user options

User options are local configuration settings such as currency, negative signs, date and time, and report formats.

user program

A user program is an external COBOL program that enables access to an input source type that ACR/Detail cannot access directly.

User Report

This user-defined report can contain any or all of the data from a Control Report, in a specified sequence.

utilities

See database utilities.

V

validation error

The system displays this error if it receives input that it cannot recognize.

variable cycle processing

Variable cycle processing enables you to control the retrieval of history items based on a key mask or allows the run date, time, and cycle ID to be greater than the current run.

W**Windows mode**

This mode allows you to directly access the ACR/Summary and ACR/Detail server on the Windows platform to define databases, run jobs or utilities, and view the results.

Index

Symbols

@@FILLER 247
@FILETMP 230, 247

Numerics

22-character Numeric Format dialog box 78

A

access mode

- access mode 6 131
- access mode 7 133
- differences between 6 and 7 130
- example of access mode 6 132
- example of access mode 7 136
- selecting 130
- similarities between 6 and 7 130

Accessing ACR/Detail Client 25

accumulate out-of-balance values 79

Accumulated Data Detail Report 360

ACR/Detail

- components 19
- major components 18
- overview 16

ACR/Detail Client

- accessing 25
- configuration 34
- input types supported 17

ACR/Workbench for Detail 387

- configuration modes 388
- features for jobs uploaded to z/OS 396
- tutorial 393

advance record 128

alternate control task 396

analyzing your reconciliation needs 38

Auto Cycle Indicator dialog box 77

B

base job for multi-level reconciliation 173

basic job information

options 74

setting up 73

batch definition transaction

overview 249

record 16

batch definition transactions

sets 252

transaction code 250

transaction id 250

transaction key 250

transaction number 251

transaction sequence 251

batch file

see also Windows, running in batch

list of 406

locating and copying 404

maximum file size 405

preparing to use 404

tab-delimited files 404

udpcgen.bat 407

udptest.bat 408

udpuser.bat 409

uds2000.bat 411

uds6000.bat 414

udscsv.bat 425

udsdfb.bat 414

udsdfi.bat 415

udsdhfa.bat 415

udshfc.bat 416

udshfd.bat 416

udshfe.bat 417

udshfl.bat 417

udshfr.bat 418, 420

udshft.bat 418

udshfu.bat 419, 421

udsinit.bat 422

udsmr01.bat 422

udstfint.bat 423

udstfls.bat 423

udsupdt.bat 424

batch options 107

Build ACRUPDT Report 382
Build Batch Transactions utility 220
Build Definitions
 batch file 414
 script 443
Build Extraction Program utility 220
 batch file 407
 script 435

C

calculated item 39, 95, 96
 see also analyzing your reconciliation needs
 conditional 95
Caret to Not Utility 390, 397
century values
 calculation 158
 verifying 158
Comma Delimited Text File utility 229
commas
 see print commas 169
comments, job 79
conditional calculated item 95
configuration 17, 34
 see also Local mode, UNIX mode, Linux mode, and Windows mode
 setting up 35
Control Entity View 69
Control Report 40, 353
 out-of-balance summary 356
Control Report title dialog box 74
Control Report XML option dialog box 78
Control Report XML Option, UNIX or Linux 170
Control Task (Job) Name dialog box 74
Control Task (Job) View, accessing 69
Control Update Information Report 379
Control Update Information Report utility 219
Copy History
 batch file 416
 script 444
Copy History utility 222
Create Comma Delimited Text File 229

 batch file 425
 script 453
Cross Reference File Definitions utility 220
currency format options, UNIX or Linux 165
customer support 23
cycle number
 formats 156
cycle numbers
 display and printing 158
 extraction 156
 storage of 158

D

Database History and Statistics Report 367
Database Initialization Report 365
Database Initialization utility 218
Database Update utility 218
databases
 see also definition database and history database
 securing 20
date
 formats 156
date format options
 UNIX or Linux 168
dates
 display and printing 158
 extraction 156
 storage of 158
DB2
 access 36
DB2 table
 access 150
defaults for ACR/Detail
 see also user options
 overriding 38
definition database 19
 deleting unused records 399
 initializing 37
 maintaining 399
 update batch file 424
 update script 452
 updating 69
 updating in batch 252
Definition Database Cross Reference Report 381

- definition database file
 - default file name (udsdf.dat) 22
 - Definition Database Listing 367
 - Definition Database Update Report 379
 - definitions
 - see also job definitions, file definitions, table definitions, and history analysis definitions
 - downloading 117, 389
 - importing 37, 245
 - opening existing 69
 - saving 69
 - definitions referenced by deleted items
 - deleting 71
 - Delete History
 - batch file 416
 - script 445
 - Delete History utility 223
 - detail field 123, 129
 - processing 137
 - Detailed Exception Definitions Report 374
 - Detailed File Definitions Report 371
 - Detailed Job Definitions Report 368
 - Detailed Table Definitions Report 373
 - Detailed Table Definitions Reports 372
 - dialog boxes 28
 - downloading
 - definitions 117, 389
 - from z/OS 389
 - input sources 389
 - dynamic translation table 205
- E**
- environment file
 - global, for Oracle 36
 - environment variables
 - see also environment file
 - setting in scripts 433
 - setting in Windows batch files 406
 - Exception Message Report 214
 - extended internal item 90
 - extended internal items 88
 - external translation table 182
 - characteristics of 181
 - constant translation data 188
 - creating 184
 - example 188
 - view 33
 - VSAM Key Sequenced file as source 187
 - VSAM ONEROW processing 190
 - Web page as source 188
 - External Translation Table Trace Report 196
 - External Translation Tables
 - report sample 373
 - Extract History
 - batch file 417
 - script 445
 - Extract History utility 224
 - Extracted Data Detail Report 358
 - Extraction Program Interface 398
 - implementation 398
 - Extraction Program Test
 - batch file 408
 - extraction variables 91, 92, 191
 - extended 91
 - setting up 93
- F**
- field formats for file definitions 146
 - file
 - processing in ACR/Detail Client 23
 - types in ACR/Detail Client 21
 - file definitions
 - deleting 159
 - filler record 247
 - Fluctuation Analysis Report 214, 375
 - folders 26, 28
 - freeform output file 82
 - Free-Form Report 231
 - As is option 235
 - creating 231
 - creating headers 233
 - fonts 235
 - internal codes 239
 - justification options 235
 - sample 234
 - working with objects 234
 - Free-Form Report XML Option, UNIX or Linux 170

freeform SQL statements 151

H

hash translation 206

history analysis

overview 209

process definitions 211

report definitions 210

standard deviation 457

variance 457

History Data Detail Report 360, 378

history database 19, 20

default file name (udshf.dat) 22

initializing 37

record layout 125

History Database Listing Report 376

History Database Reload Report 385

History Database Transfer Report 384

History Database Unload Report 385

history item 94

setting up 94

I

icons

see toolbar icons

Import Options dialog box 246, 391

importing

definitions 37

from z/OS 391

importing definitions 245

Index of Definition Keys Report 367

Index of Detail Histories Report 377

Initialize Databases

batch file 422

script 449

Initialize History utility 222

initialize text items to spaces dialog box 78

initialize text items to spaces, UNIX or Linux 169

Initialize Translation Database

batch file 423

script 451

Initialize Translation Database utility 227

input source 115

see also analyzing your reconciliation

needs

planning for 38

setting up 118

for relational data 150

setting up for relational data 150

view 32

input types supported in ACR/Detail Client 17

installation manuals 13

integrity check 230

internal item 90

see also analyzing your reconciliation

needs

planning for 39

internal items 88

internal translation table 182

characteristics of 181

creating 203

item number 107

see also filler record

display or hide 108

generation error 108

moving definitions to and from z/OS 391, 392

pattern assignment 108

J

job

see also UNIX, running in batch and Linux, running in batch

see also Windows, running in batch

different methods of defining 16

running

through ACR/Detail Client 17

through ACR/Detail Workbench 17

setting up 59

Job Comments dialog box 79

job definitions

copying and pasting 70

extended internal items 88

internal items 88

setting up 87

job progress messages 79

K

key field 123, 129
 Key Masking 109
 Key Sort Order dialog box 79

L

Length of Descriptions dialog box 391, 392

Linux

see also Linux mode
 DB2 access 36
 ODBC access 36
 Oracle access 36
 restrictions 14
 running in batch 429

- changing user options 431
- conditionally executing a step 429
- list of scripts 434
- location of environment file 431
- maximum file size 432
- migrating test definitions to production 453
- preparing to use scripts 431
- setting environment variables 433
- specifying run-time parameters 433

 scripts 17, 429
 setting user options

- Control Report XML Option 170
- date format 168
- Free-Form Report XML Option 170
- initialize text items to spaces 169
- use 22-character numeric format 169
- User Report XML Option 170
- XML DB Error Option 171
- XML DB Message Size 171

 uploading from 37
 user options

- currency format 165
- negative sign 167
- print commas 169

 Linux mode 17

- setting up a Linux configuration 36

 Linux user options 162

- bypass processing 171
- set return code when no records are

extracted 171

udsuf.dat 162
 UNIEXEO 162, 171
 UNIXOPT 162

List Definitions

batch file 415
 script 443

List Definitions utility 219

List History

batch file 417
 script 446

List History utility 221

List Translation Database

batch file 423
 script 451

List Translation Database utility 227

Local mode 388

M**Management Reporting**

batch file 422
 script 450

mark-and-capture method 117, 121

Menu 26

edit 27
 file 27
 help 27
 options 27
 reports 27
 run 27
 view 27
 window 27

message 40

see also analyzing your reconciliation needs

examples 107
 inserting item description 106
 inserting item value 106
 overview 106
 processing 106
 setting up 106

migrating test definitions to production

UNIX or Linux 453
 Windows 426

multi-level reconciliation 173

N

negative sign format options, UNIX or Linux 167

Not to Caret Utility 390, 397

O

ODBC

access 36, 150

ONEROW processing, VSAM 190

Oracle

access 36, 150

Order of precedence 104

Out-of-Balance Summary Report 356

output file 81

original 81, 82

original, record layout 84

set up 83

specifying the location 83

P

print commas option, UNIX or Linux 169

Print Control Report dialog box 75

Print Extracted Data Detail Report dialog box 75

Print Free-Form Report dialog box 77

Print History Data Report dialog box 76

Print Management Report utility 227

Print Recap Report dialog box 75

Print Sorted/Accumulated Data Detail Report dialog box 76

Print Suspense Aging Report utility 228

Process Definitions 211

processing flow diagram for ACR/Detail 18

Q

qualifier 173

QuickChange 394

R

Recap report 357

Reconciliation

suspense 176

reconciliation

multi-level 173

Reconciliation Level dialog box 78

reconciliation level qualifier 173

Reconciliation User Report

see User Report

reformat field 123

reformat record 129

relational data 150

relative record 128

defining in access modes 6 and 7 122

Reload History

batch file 420

script 448

Reload history utility 223

Reorganize History

batch file 418

script 446

Reorganize History utility 224

Reorganize Translation Database utility 228

Report Definitions for History Analysis Report 210

reports 349

see also analyzing your reconciliation needs

Accumulated Data Detail Report 360

Build ACRUPDT Report 382

Control Report 353

Database History and Statistics Report 367

Database Initialization Report 365

debugging 358, 359, 360, 361

Definition Database Cross Reference

Report 381

Definition Database Listings 367

Definition Database Update Report 379

Detailed Exception Definitions Report 374

Detailed File Definitions Report 371

Detailed Job Definitions Report 368

Detailed Table Definitions Reports 372

e-mail distribution 350

Extended Translation Table Trace Report 184

External Translation Table Trace Report 196

- Extracted Data Detail Report 358
- Fluctuation Analysis 375
- History Analysis Reports 214
- History Data Detail Report 360, 378
- History Database Listing 376
- History Database Reload Report 385
- History Database Transfer Report 384
- History Database Unload Report 385
- Index of Definition Keys Report 367
- Index of Detail Histories 377
- recap 357
- Sorted Data Detail Report 359
- Summary of Detail Histories 377
- Suspense Aging Report 383
- System Messages Report 353
- table of all reports 351
- Threshold Analysis 375
- Trace Report 361
- Transaction Database Listing Report 383
- Transaction Listing Report 386
- viewing 33
- viewing on UNIX or Linux 350
- Web distribution 350
- working with 349
- restrictions
 - for jobs uploaded to the mainframe 387
 - for Windows, UNIX, and Linux 14
- return code
 - see also analyzing your reconciliation needs
 - definition 40
- RLQ job for multi-level reconciliation 173
- Round Results dialog box 78
- rule 40, 105
 - see also analyzing your reconciliation needs
 - conditional 99
 - active 100
 - evaluation of 100
 - inactive 100
 - definition 96
 - processing when item is not found 104
 - reporting status 105
 - standard 97
 - active 97
 - inactive 98
- Run History Analysis
 - batch file 414
 - script 442
- Run Reconciliation
 - batch file 411
 - script 439
- run-time parameters
 - specifying in Windows batch files 406
 - specifying scripts 433
- S**
- script 17
 - see also UNIX, running in batch and Linux, running in batch
 - list of 434
 - preparing to use 431
 - udpcgen.sh 435
 - udpctest.sh 436
 - udpuser.sh 437
 - uds2000.sh 439
 - uds6000.sh 442
 - udscsv.sh 453
 - udsdfb.sh 443
 - udsdfc.sh 443
 - udsdfhfa.sh 444
 - udshfc.sh 444
 - udshfd.sh 445
 - udshfe.sh 445
 - udshfl.sh 446
 - udshfr.sh 446, 448
 - udshft.sh 447
 - udshfu.sh 447, 449
 - udsinit.sh 449
 - udsmr01.sh 450
 - udstfint.sh 451
 - udstfls.sh 451
 - udsupdt.sh 452
- selection field 126
 - defining in access modes 6 and 7 122
 - selection group 126

Set Return Code dialog box (in Basic (job) Information) 75
single-precision and double-precision numbers
 constraints 113
Sorted Data Detail Report 359
status bar 27
Store History dialog box 74
Summary of Detail Histories 377
Suspense Aging
 batch file 415
 script 444
Suspense Aging Report 383
Suspense Aging Report utility 228
Suspense Reconciliation 176
SYSOUT Report 353
System Messages Report 353

T

tab-delimited files in Windows 404
table
 external translation
 using extraction variables 191
tally out-of-balance keys 79
Test Extraction Program
 script 436
Test User Program
 batch file 409
 script 437
Threshold Analysis Report 375
Title Bar 26
tolerance
 out-of-balance amounts 79
 out-of-balance keys 79
toolbar icons 27, 29, 30
 control task (job) view 30
 external translation table view 33
 Free-Form Report view 236
 input source view 32
 report view 33
Trace Report 361
Transaction Database Listing Report 383
Transaction Listing Report 386

Transfer History
 batch file 418
 script 447
Transfer history utility 222
translation processing 183
translation table, see also internal, external, and dynamic translation table
 see also
tutorial
 ACR/Detail Client 41
 ACR/Workbench for Detail 393
 Local mode 393

U

udpcgen.bat 407
udpcgen.sh 435
udptest.bat 408
udptest.sh 436
udpuser.bat 409
udpuser.sh 437
uds2000.bat 411
uds2000.sh 439
uds6000.bat 414
uds6000.sh 442
udscf.dat 22
udscsv.bat 425
udscsv.sh 453
udsdf.dat 22
udsdfb.bat 414
udsdfb.sh 443
udsdf.bat 415
udsdf.sh 443
udsdhfa.bat 415
udsdhfa.sh 444
udsfls.bat 423
udsfls.sh 451
udshf.dat 22
udshfc.bat 416
udshfc.sh 444
udshfd.bat 416
udshfd.sh 445
udshfe.bat 417
udshfe.sh 445
udshfl.bat 417
udshfl.sh 446
udshfr.bat 418, 420

- udshfr.sh 446, 448
- udshft.bat 418
- udshft.sh 447
- udshfu.bat 419, 421
- udshfu.sh 447, 449
- udsinit.bat 422
- udsinit.sh 449
- udsmr01.bat 422
- udsmr01.sh 450
- udstfint.bat 423
- udstfint.sh 451
- udsuf.dat 22, 161
- udsupdt.bat 424
- udsupdt.sh 452
- UNICTN
 - see Caret to Not Utility
- unidf.dat 19
- unihf.dat 20
- UNINTC
 - see Not to Caret Utility
- UNIX
 - see also UNIX mode
 - DB2 access 36
 - ODBC access 36
 - Oracle access 36
 - restrictions 14
 - running in batch 429
 - changing user options 431
 - conditionally executing a step 429
 - list of scripts 434
 - location of environment file 431
 - maximum file size 432
 - migrating test definitions to production 453
 - preparing to use scripts 431
 - setting environment variables 433
 - specifying run-time parameters 433
 - scripts 17, 429
 - setting user options
 - Control Report XML Option 170
 - date format 168
 - Free-Form Report XML Option 170
 - initialize text items to spaces 169
 - use 22-character numeric format 169
 - User Report XML Option 170
 - XML DB Error Option 171
 - XML DB Message Size 171
 - uploading from 37
 - user options
 - currency format 165
 - negative sign 167
 - print commas 169
- UNIX mode 17
 - setting up a UNIX configuration 36
- UNIX user options 162
 - bypass processing 171
 - set return code when no records are extracted 171
- udsuf.dat 162
- UNIEXEO 162, 171
- UNIXOPT 162
- UNIXOPT 163
- Unload History
 - batch file 419
 - script 447
- Unload history utility 222
- Update Definition Database
 - batch file 424
 - script 452
- Update History
 - batch file 421
 - script 449
- Update History utility 229
- uploading
 - to UNIX or Linux 37
 - to z/OS 390
- use 22-character numeric format, UNIX or Linux 169
- user control file (udscf.dat) 22
- user interface 19, 26
 - see also views, folders, menus, toolbar icons, status bar, and dialog boxes
- user options 38, 161
 - see also UNIX user options, Linux user options, UNIXOPT, udsuf.dat, and UNIEXEO
- file 22, 161
- setting in Local mode 388

Index

User Report 40, 241
 deleting 239, 243
 sample 243
 setting up 241
User Report XML Option, UNIX or Linux
170
Utilities
 reorganize history 224
utilities 217

V

views 26, 29
 control task (job) 30
 external translation table 33
 Free-Form Report 236
 input source 32
 report view 33
VSAM ONEROW processing 190

W

Web page
 input source 120
Windows
 see also Windows mode
 batch files 17, 403
 restrictions 14
 running in batch 403
 conditionally executing a step 403
 list of batch files 406
 locating batch files 404
 maximum file size 405
 migrating test definitions to production
 426
 preparing to use batch files 404
 setting environment variables 406
 specifying run-time parameters 406
 tab-delimited files 404
Windows mode 17
 setting up a Windows configuration 35
 setting user options 161

X

XML DB Error Option, UNIX or Linux 171
XML DB Message Size, UNIX or Linux 171

Z

z/OS
 ACR/Workbench features for jobs
 uploaded to 396
 downloading from 389
 importing from 391
 uploading definitions to 390