

precisely

ACR/Summary

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	11
Overview of ACR/Summary	14
Ways to Use ACR/Summary	15
Configuration Modes	16
ACR/Summary Processing Flow	16
ACR/Summary Components	17
Balancing Function	20
ACR/Summary Client File Types	22
Database and File Processing Diagram	24
Contacting Customer Support	24

Chapter 1 Preparing to Use ACR/Summary Client

Accessing and Exiting from ACR/Summary Client	25
Using the ACR/Summary Client Interface	26
Configuring ACR/Summary Client	34
UNIX or Linux Only: Modifications for RDBMS Access	36
Initializing the Definition and History Databases	36
Downloading, Uploading, and Importing Definitions	37
Setting the User Options	37
Selecting an Application Interface Mode	38
Analyzing Your Balancing Needs.....	42

Chapter 2 Tutorial

Preparing for the Tutorial	45
Determining Your Balancing Requirements	46
Creating the Balancing Job	49
Running the Job	60
Viewing the Output Reports and Handling Errors.....	61

Chapter 3 Setting Up a Balancing Job

What is a Balancing Job? 63
Preparing to Set Up a Balancing Job 64
Setting Up and Running a Balancing Job 65
Viewing the Output Reports and Handling Errors 69
Working with Job Definitions 70

Chapter 4 Setting Up Basic Job Information

Basic (Job) Information Options 75
Name 76
Length of Descriptions 77
Report Headings 77
Store History 77
Print Control Reports 77
Extraction File Options 78
Access User Exit 83
Print Recap Report 83
Set Return Code 84
Cycle Processing 84
Automatic Rerun/Audit Trail Options 84
Round Results 84
Use 22-character Numeric Format 85
Control Report XML Option 85
Adding Job Comments 85

Chapter 5 Setting Up Job Definitions

About Internal Items and Extended Internal Items 88
Internal Items 90
Extended Internal Items 90
Extraction Variables 91
History Items 96
Calculated Items 97
Rules 99
Defining Messages 108
Alternate Control Tasks 112

	Recap Items	112
	Using Batch Options to Control Item Numbers.....	113
Chapter 6	Setting Up Input Sources	
	What Is an Input Source?	115
	Preparing to Set Up an Input Source.....	116
	General Procedure for Setting Up an Input source	117
	Selecting an Access Mode	120
	Understanding Field Formats	136
	Setting Up an Input Source for the Selected Access Mode.....	139
	Setting Up an Input Source for Relational Data.....	149
	Date and Cycle Number Extraction and Processing	156
	Time Extraction and Processing.....	161
	Deleting File Definitions.....	162
	Extracting Directly from a Delimited Data File	162
Chapter 7	Setting the User Options	
	The User Options File (uniuf.dat)	165
	Setting User Options	165
	Using Global Messages.....	166
	Setting User Options for UNIX or Linux.....	168
Chapter 8	Setting Up Free-Form Reports	
	What is a Free-Form Report?.....	187
	Creating a Free-Form Report	187
	Viewing the Internal Code Representations for Objects	193
	Report Sample.....	193
	Deleting a Free-Form Report.....	194
	Printing a Free-Form Report.....	194
Chapter 9	Setting Up User Reports	
	What is a User Report?	195
	Setting Up a User Report	195
	Deleting a User Report.....	197

	User Report Samples with Field Definitions	197
Chapter 10	Setting Up Cycle Processing	
	What is a Cycle ID?	199
	Reasons for Using Run Numbers Other than 000	200
	Cycle Processing Overview	201
	Using Similar Cycle Processing Options for Related Jobs	202
	Options File (UNIOF.DAT) Features for Cycle Processing	202
	Order of Precedence for Cycle ID Assignment	206
Chapter 11	Setting Up Cycle Tables	
	Purposes of Cycle Tables	209
	Procedure for Setting Up a Cycle Table.....	210
Chapter 12	Understanding Variable Cycle Processing	
	Review of Cycle IDs	211
	What Is Relative Cycle Processing?.....	211
	What Is Variable Cycle Processing?.....	213
	Choosing a Variable Cycle Processing Option	214
Chapter 13	Using Translation Tables	
	Translation Processing	221
	Internal Vs. External Translation Tables	222
	Using External Translation Tables	224
	External Translation Table Trace Report.....	233
	Using Internal Translation Tables.....	240
Chapter 14	Importing Definitions	
	Purposes of Importing	243
	Procedure for Importing Definitions	243
	Understanding @FILETMP Jobs.....	245
	Using @@FILLER Records.....	245

Chapter 15 Using History Analysis

What is History Analysis?	247
Creating History Analysis Definitions	248
Report Samples	250

Chapter 16 Using the Utilities

Running Utilities	254
Database Initialization	254
Database Update	254
List Definitions	255
Control Update Information Report Utility	255
Build Batch Transactions	256
Cross Reference Internal Items	256
Cross Reference File Definitions	257
List History	257
Initialize History	257
Copy History: Copy All Records	258
Copy History: Copy Selected Records	258
Copy History: Unload Records to a Work File	258
Copy History: Reload Records to a Work File	259
Delete History	259
Reorganize History	259
Extract History	260
Reprint Control Report	263
Print Management Report	263
Update History	263
Create Comma Delimited Text File	264
Integrity Check	264

Chapter 17 Modeling Job and File Definitions

What is Modeling?	267
Modeling Terminology	268
Modeling Job Definitions	268
Modeling File Definitions	272

Appendix A Using Batch Definition Transactions

What Is a Batch Definition Transaction?.....	283
Format of Batch Definition Transaction Records	284
Updating the Definition Database.....	286
Purging Definitions	287

Appendix B Reports

Working with Reports on ACR/Summary Client.....	289
Reports Table.....	291
System Messages Report - SYSOUT.RPT.....	292
Control Report - UNIACR.RPT	292
Recap Report - UNIRCR.RPT	296
User Report - UNIUSR.RPT	297
Free-Form Report - UNIUFR.RPT	297
Trace Report - UNIETR.RPT	298
History Analysis Report - UNISSR2.RPT.....	302
Database Initialization Report - UNIIR.RPT.....	302
Database History and Statistics Report - UNIDLR.RPT	303
Index of Definition Keys Report - UNIDLR.RPTT	303
Detailed Job Definitions - UNIDLR.RPT	305
Detailed File Definitions - UNIDLR.RPT	307
Detailed Table Definitions - Internal Translation Table - UNIDLR.RPT .. 309	
Detailed Table Definitions - External Translation Table - UNIDLR.RPT .. 310	
Detailed Spreadsheet (History Analysis Definitions) Report - UNIDLR.RPT	311
Print Control Update Information Report - UNIDLRCU.RPT	312
Build ACRUPDT Report - UNIDBR.RPT.....	313
Definition Database Update Report - UNIUPDT.....	313
Cross Reference Internal Items Report - UNIDXR.RPT	316
Cross Reference File Definitions Report - UNIDXR.RPT	317
Database History and Statistics Report - UNIHDLR.RPT	318
Index of ACR Histories - UNIHDLR.RPT.....	318
Summary of ACR Histories - UNIHDLR.RPT	319
History Data Detail Report - UNIHSTD.RPT.....	319

Transaction Listing Report - UNITR.RPT	320
History Database Copy Report - Summary of Histories Transferred - UNIHTR.RPT	321
History Database Copy Report - Summary of Histories Unloaded - UNIHTR.RPT	321
History Database Copy Report - Summary of Histories Reloaded - UNIHTR.RPT	322
History Database Reorg Report - UNIHRR.RPT	322
Print Management Report - UNIMR.RPT	323

Appendix C Using ACR/Summary for Windows in Batch

Conditionally Executing a Step Based on a Return Code	325
Before You Begin	326
Editing Batch Files	327
Batch File Quick-Reference with Page Numbers	328
Migrating Test Definitions to Production	345

Appendix D Using ACR/Summary for UNIX or Linux in Batch

Conditionally Executing a Step Based on a Return Code	347
Before You Begin Your Edits	349
Editing the Scripts	350
Script Quick-Reference with Page Numbers	352
Migrating Test Definitions to Production	368

Appendix E Non-Biased Variance and Standard Deviation Computations

Field Descriptions and Formulas	371
Examples	373

Appendix F Using ACR/Workbench

Restrictions for Mainframe Users	379
ACR/Workbench for Summary Configuration Modes	380
Downloading and Uploading Between z/OS and ACR/Workbench .	381
Importing Definitions and Input Sources from z/OS	382

■ Contents

ACR/Workbench for Summary Tutorial.....	385
Features Applicable only to ACR/Workbench for Summary Jobs Uploaded to z/OS	386
Glossary	389
Index	403

Preface

This preface provides an introduction to ACR/Summary and ACR/Summary Client. It contains the following sections:

- “About This Guide” on page 11
- “Overview of ACR/Summary” on page 14
- “Ways to Use ACR/Summary” on page 15
- “Configuration Modes” on page 16
- “ACR/Summary Processing Flow” on page 16
- “ACR/Summary Components” on page 17
- “Balancing Function” on page 20
- “ACR/Summary Client File Types” on page 22
- “Database and File Processing Diagram” on page 24
- “Contacting Customer Support” on page 24

About This Guide

Purpose

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

Audience

This manual will be useful to users of ACR/Summary for Windows, ACR/Summary for UNIX, and ACR/Summary for Linux. If you have ACR/Workbench for Summary, consult this appendix before reading the rest of the manual.

Installation Documentation

Installation information can be found in the following manuals:

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

ACR/Summary Client Restrictions by Platform

Note: For ACR/Workbench restrictions, see “Restrictions for Mainframe Users” on page 379.

Restrictions for UNIX or Linux

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

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

Restrictions for Windows

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

- UNIX, Linux, and Local configuration modes

Organization of This Guide

Much of the Preface and all of the chapters in this manual focus on using ACR/Summary Client.

Information on using ACR/Workbench for Summary is provided in Appendix F, “Using ACR/Workbench” on page 379. Because the graphical interfaces and most functionality for ACR/Workbench for Summary and ACR/Summary Client are identical, this appendix documents only functionality that is specific to ACR/Workbench for Summary.

The following appendices provide information for using ACR/Summary in a batch environment:

Appendix A, “Using Batch Definition Transactions”

Appendix C, “Using ACR/Summary for Windows in Batch”

Appendix D, “Using ACR/Summary for UNIX or Linux in Batch”

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/Summary and ACR/Summary Client, including configuration, functionality, files and databases.

Chapter 1, “Preparing to Use ACR/Summary Client”—Provides essential instructions for using the product and analyzing your balancing needs.

Chapter 2, “Tutorial”—Takes you through the process of setting up and running a balancing job.

Chapter 3, “Setting Up a Balancing Job”—Provides the procedure for setting up and running a typical balancing job.

Chapter 4, “Setting Up Basic Job Information”—Provides an overview of the Basic Information options at the top of the Control Task (Job) View.

Chapter 5, “Setting Up Job Definitions”—Provides information on setting up job definitions and controlling item numbers.

Chapter 6, “Setting Up Input Sources”—Explains how to set up input sources for balancing purposes. When you set up an input source, you typically define the input source file, specify how it will be accessed, and provide definitions that will enable ACR/Summary to locate the pertinent records and extract the required values.

Chapter 7, “Setting the User Options” Discusses some system-wide defaults and how to change them.

Chapter 8, “Setting Up Free-Form Reports”—Explains how to set up this custom report that can include any of the control values obtained from a balancing run.

Chapter 9, “Setting Up User Reports”—Explains how to define a custom report that prints any of the data that can be included in the Control Report in the order that you specify.

Chapter 10, “Setting Up Cycle Processing”—Explains cycle IDs and how to use cycle processing. In addition, it provides information on the cycle override and history insert features.

Chapter 11, “Setting Up Cycle Tables”—Explains how to set up a cycle table for use in balancing or in history analysis.

Chapter 12, “Understanding Variable Cycle Processing”—Reviews cycle IDs and relative cycle processing and then explains how variable cycle processing works.

Chapter 13, “Using Translation Tables”—Describes how to set up and use translation tables

Chapter 14, “Importing Definitions”—Explains how to import definitions from z/OS, UNIX, and Linux.

Chapter 15, “Using History Analysis”—Describes how to set up history analysis definitions.

Chapter 16, “Using the Utilities”—Provides general instructions for accessing and running utilities and describes each utility.

Chapter 17, “Modeling Job and File Definitions”—Provides procedures for job and file modeling.

Appendix A, “Using Batch Definition Transactions”—Explains batch definition transaction records and provides layouts for each type of batch definition transaction.

Appendix B, “Reports”—Describes each report and provides an example.

Appendix C, “Using ACR/Summary for Windows in Batch”—Documents ACR/Summary batch files and describes how to migrate your jobs from test to production.

Appendix D, “Using ACR/Summary for UNIX or Linux in Batch”—Documents ACR/Summary scripts and describes how to migrate your jobs from test to production.

Appendix E, “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.

Appendix F, “Using ACR/Workbench”—Documents features that are only applicable to ACR/Workbench for Summary.

This guide also includes a glossary and an index.

Overview of ACR/Summary

ACR/Summary is rule-based software that verifies the accuracy of key business information including totals, subtotals, record counts, dates and text across applications, files, databases, and reports. ACR/Summary is available for z/OS, Windows, UNIX, Linux, and the IBM i.

Ways to Use ACR/Summary

Ways to Define ACR/Summary Jobs

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

- Through ACR/Summary Client. ACR/Summary Client is a graphical interface residing on the PC that makes it easier to create, maintain, and run ACR/Summary balancing jobs. ACR/Summary Client makes it easier to set up ACR/Summary definitions for 1) balancing jobs, 2) their associated tables, and 3) history analysis reporting.
- By manually creating or editing batch definition transaction records. You can then run a batch file (on Windows) or a script (on UNIX or Linux) to update existing job, file, table, or history analysis definitions. See Appendix A, “Using Batch Definition Transactions” for the layouts of the batch definition transaction records if you want to create or edit them manually.

Ways to Run ACR/Summary Jobs

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

- Through ACR/Summary Client.
- By customizing ACR/Summary batch files (Windows) or scripts (UNIX or Linux). These can then be run individually or included in your batch job stream. The batch files and scripts are documented in the following appendices:
 - Appendix C, “Using ACR/Summary for Windows in Batch”
 - Appendix D, “Using ACR/Summary for UNIX or Linux in Batch”

Configuration Modes

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

Windows Mode

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

Input Source Types Supported

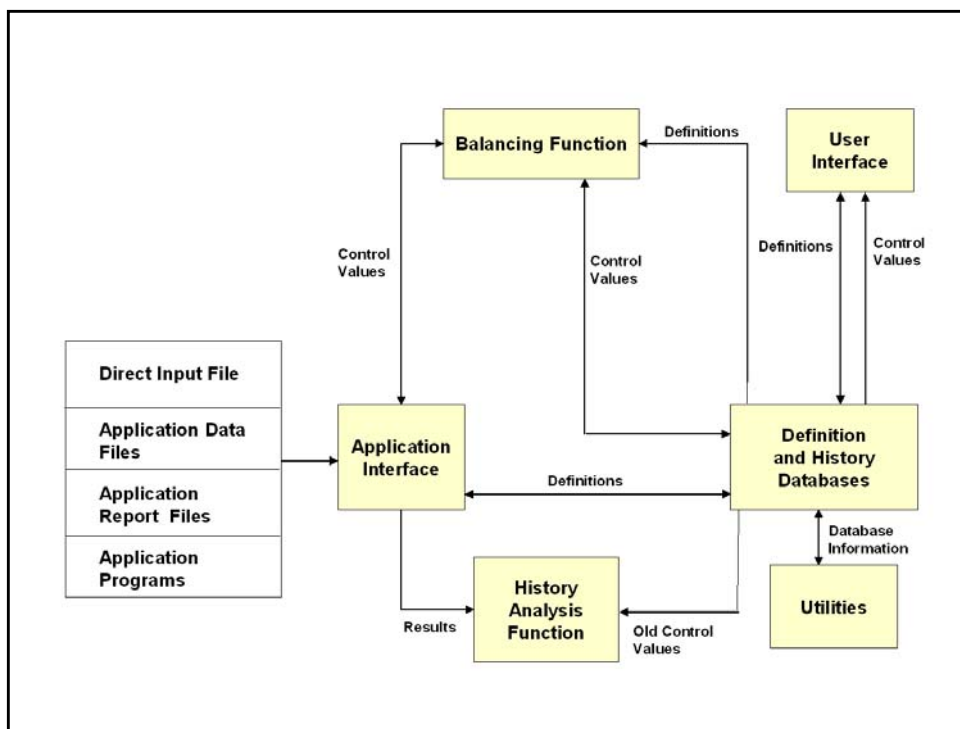
The following types of input sources can be used in balancing jobs that are set up through ACR/Summary Client:

- 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
- Data from Oracle databases (applies only in UNIX or Linux mode)

ACR/Summary Processing Flow

The following figure shows an overview of ACR/Summary processing. The user interface refers to the graphical interface provided by ACR/Summary Client. The location of the components and files you will use to run ACR/Summary 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/Summary 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/Summary reside on the Windows server where ACR/Summary and ACR/Detail Server is installed.



The major components of ACR/Summary shown above are described in the next section.

ACR/Summary Components

User Interface

The graphical interface provided by ACR/Summary Client makes it easier to create, maintain, and run ACR/Summary balancing jobs. For an introduction to the interface, see “Using the ACR/Summary Client Interface” on page 26. To get some practice using the interface, see the “Tutorial” on page 45.

Definition and History Databases

Before you can use ACR/Summary Client, you need to initialize (create) the definition and history databases. Following are descriptions of these databases. For a diagram that shows database processing, see “Database and File Processing Diagram” on page 24.

Definition Database (Default Name `unidf.dat`)

What is the Definition Database?

The definition database is a keyed physical file that stores the balancing specifications (parameters) used in ACR/Summary to balance data and generate reports.

You can specify the following types of definitions:

- **Job definitions** control the balancing process, specify the rules for balancing the extracted values, and define the reports that will run when the job runs.
- **File definitions** specify the input sources (files) to be used in the job and how the values to be used in balancing will be located and extracted from the input sources.
- **Table definitions** include the following table types:
 - **Cycle table**—A means of validating cycle numbers. A table is populated with valid cycle numbers against which the current cycle number is compared. Processing continues only if a match is found.
 - **Internal translation table**—A table that you create and populate within ACR/Summary to do simple one to one translations from one text value to another text value. For example, you might want to translate a 2-character state code to a 2-character state abbreviation.
 - **External translation table**—A table built at run time (usually by referencing data external to ACR/Summary) and used to translate values from an input source for balancing purposes.
- **History analysis definitions** enable you to define a report on selected data from the history database. The History Analysis Report can help in identifying trends in your balancing processes.

Updating the Definition Database

When you save a job, a table, or history analysis definitions created through ACR/Summary Client, you specify a name and location where the files containing your definitions will be stored. You then need to run a database update in order to add the definitions to the definition database before you can use them in a job run.

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 a job, a table, or history analysis definitions and modify them, save the changes and run the database update again.

When you update the definition database, an integrity check is performed (unless the feature has been turned off through the Options menu). An integrity check verifies the validity of the currently active control task (job), table, or history analysis document. For more information, see “Integrity Check” on page 264.

History Database (Default Name unihf.dat)

The history database is a relative-record entry-sequenced ISAM file that stores the results of balancing runs so that they can be used in another run of the same job or in another job. Each history record contains a job ID, cycle ID, and the control value extracted for each internal or extended internal item.

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.

Application Interface

ACR/Summary's application interface provides a choice of modes for obtaining control values for balancing purposes:

File Interface Mode

This is the typical application interface mode, in which control values are extracted directly from one or more input sources. This mode is especially well-suited for incorporating automated balancing steps into your existing applications. For more information, see "Selecting an Application Interface Mode" on page 38.

Direct Input Mode

This mode enables you to specify the control values to be stored in internal or extended internal items directly. It is useful for processing data that is not available on your system, for testing new balancing jobs, or for priming the history database. For more information, see "Understanding and Using Direct Input Mode" on page 39.

Balancing Function

The balancing function is the primary function of ACR/Summary.

In a typical batch processing stream, an application (such as general ledger or billing) consists of a job stream made up of one or more job steps. At any point along this stream you can insert a balancing step (balancing job) that will invoke the ACR/Summary balancing function to check the data as it moves through the system.

For most of your jobs, the balancing requirements can be met by including one invocation of ACR/Summary (one balancing step). If needed, a single application job-step can invoke multiple ACR/Summary balancing steps.

Each balancing step must be supplied with the following:

- An ACR/Summary job ID. This identifies a set of job definitions (audit and control requirements) that you create to be used in balancing.
- A cycle ID. This is an 8-digit cycle number and a 3-digit run number that together uniquely identify the application processing cycle that is to be verified. Cycle numbers are usually based on system dates, but any ascending number may be used.

When performed in file interface mode (the typical application interface mode), the balancing function does the following:

1. Applies your balancing job definitions to your input sources, which are usually reports or files from your application.
2. Extracts the specified control values.
3. Performs calculations specified for extracted values (in calculated items), if any.
4. Applies rules that compare and/or verify the extracted control values and calculated item values.
5. Determines whether the application job-step is in or out of balance.
6. Places the results in the history database to be used in another run of the same job or another job.
7. Generates the reports you specified, which show the results of the balancing function.

After any out-of-balance conditions are corrected, the job can be rerun.

History Analysis Function

The history analysis function enables you to create a set of definitions for extracting, analyzing, and reporting on selected data from the history database. The generated History Analysis Report enables you to do the following:

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

Utilities

Most ACR/Summary utilities help you maintain and report on your definition and history databases. For more information, see “Using the Utilities” on page 253

ACR/Summary Client File Types

Following is a description of the types of files used in ACR/Summary Client. For a diagram that shows file processing, see “Database and File Processing Diagram” on page 24.

Control Task Files (.tsk)

A .tsk file contains the definitions for a balancing job (control task) that are used by the ACR/Summary Client graphical interface.

Batch Definition Transaction Files (.def)

A .def file is created in addition to the .tsk, .tbl, or .had file when you save a job, table, or history analysis document through ACR/Summary Client. This file contains one of the following types of batch definitions:

- **Job and file (input source) definitions for a control task**—the batch equivalent of the .tsk file.
- **Definitions for a table**—the batch equivalent of the .tbl file.
- **Definitions for a history analysis document**—the batch equivalent of the .had file.

Table Files (.tbl)

A .tbl file contains the definitions for a table that are used by the ACR/Summary Client graphical interface.

History Analysis Definition Files (.had)

A .had file contains the definitions for a history analysis document that are used by the ACR/Summary Client graphical interface.

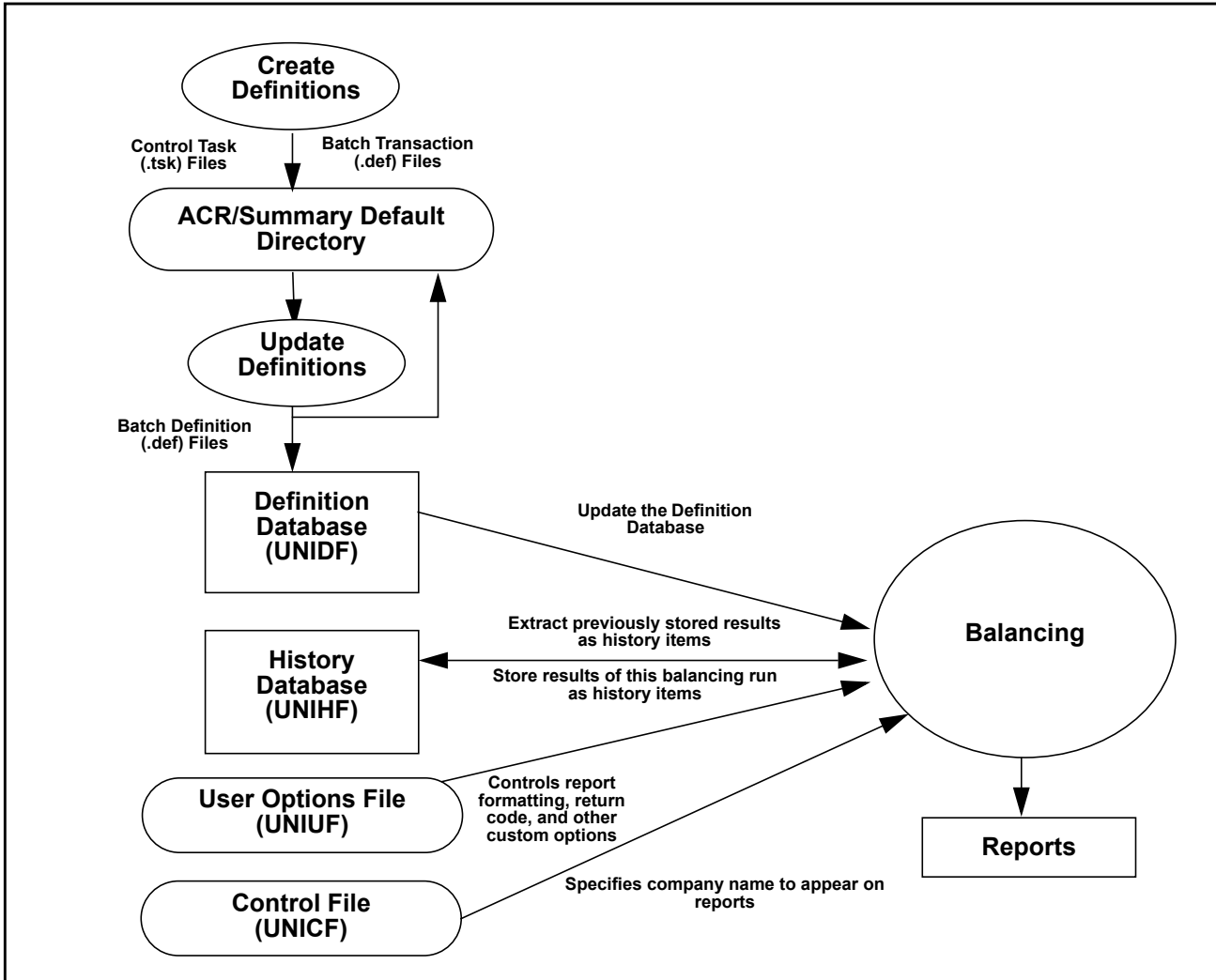
Output Report Files (.rpt)

An .rpt file is a report that can be used as input to a job or can show the results of processing.

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

- **Definition Database (default name unidf.dat)**—Contains all of the job, file, table, and history analysis definitions that have been updated to the definition database.
- **History Database (default name unihf.dat)**—Contains control values extracted as internal or extended internal items during balancing runs. These values can then be used in another run of the same job or a different job.
- **User Options file (default name uniuf.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 7, “Setting the User Options” on page 165.
- **User Control file (default name unicf.dat)**—Resides on both the PC and the ACR/Summary and ACR/Detail Server and contains licensing information as well as 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. If this file is not modified, INFOGIX, INCORPORATED will appear at the top of reports.

Database and File Processing Diagram



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

Preparing to Use ACR/Summary Client

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

- “Accessing and Exiting from ACR/Summary Client” on page 25
- “Using the ACR/Summary Client Interface” on page 26
- “Configuring ACR/Summary Client” on page 34
- “UNIX or Linux Only: Modifications for RDBMS Access” on page 36
- “Initializing the Definition and History Databases” on page 36
- “Downloading, Uploading, and Importing Definitions” on page 37
- “Setting the User Options” on page 37
- “Selecting an Application Interface Mode” on page 38
- “Analyzing Your Balancing Needs” on page 42

Accessing and Exiting from ACR/Summary Client

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

- Start ACR/Summary from your Infogix program group.
- Select **Start > Programs > Infogix > ACR Summary** (or ACR/Summary Client).
- Create a shortcut on your desktop pointing to the executable file Wsum32.exe in your installation folder and click on it.

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

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

Using the ACR/Summary Client Interface

Interface Elements

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

- **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.
- **Folders**—Each view has its own folders and folder options. You can click on 1) a folder option, 2) an item you have defined in a folder (such as an internal item), 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 27.
- **Menus**—The menu options will change depending on the type of view that is active. Help for individual menus (such as the File menu), including any associated toolbar icons, can be accessed through the product help by entering Menus in the keyword area on the Index tab and selecting the menu you are interested in.
- **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 also shows messages describing the status of actions within ACR/Summary.
- **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**.

1 ■ Preparing to Use ACR/Summary Client

Using the ACR/Summary Client Interface

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 Task (Job) View	Create or modify a control task (balancing job).	File > New > Control Task (Job) This is the default view when you open ACR/Summary.
Input Source View	Define the file(s), reports and/or program(s) from which values will be extracted for use in your balancing job.	Control Task (Job) View > Input Sources > New
User Report View	Set up a (Balancing) 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 > Balancing 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/Summary. 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
Cycle Table View	Set up a table to be used in balancing 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
	Paste	Edit Menu
	Integrity Check	Options Menu to turn on or off. Run Menu to run an integrity check.
	Database Update	Run Menu
	Balancing	Run Menu














1 ■ Preparing to Use ACR/Summary Client

Using the ACR/Summary Client Interface

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






Icons for Input Source View

Icons for Input Source View		
Icon	Command	Menu to Access This Command
	File Key	Define Menu (Input Source View)
	Begin Reference Record	Define Menu (Input Source View)
	End Reference Record	Define Menu (Input Source View)
	Reset Record	Define Menu (Input Source View)
	Embedded Key Record	Define Menu (Input Source View)
	Relative Record	Define Menu (Input Source View)

Icons for Input Source View		
Icon	Command	Menu to Access This Command
	Page/Line Record	Define Menu (Input Source View)
	RDBMS Record	Define Menu (Input Source View)
	Freeform SQL Select	Define Menu (Input Source View)
	Freeform SQL From	Define Menu (Input Source View)
	Freeform SQL Where	Define Menu (Input Source View)
	Total Record	Define Menu (Input Source View)
	Tally Record	Define Menu (Input Source View)
	Hash Record	Define Menu (Input Source View)
	Field Item	Define Menu (Input Source View)
	Selection Record	Define Menu (Input Source View)
	Reformat Record	Define Menu (Input Source View)
	Detail Record	Define Menu (Input Source View)
	Relative Record	Define Menu (Input Source View)

1 ■ Preparing to Use ACR/Summary Client



Using the ACR/Summary Client Interface




Icons for Input Source View		
Icon	Command	Menu to Access This Command
	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 190.

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)

Icons for External Translation Table View		
Icon	Command	Menu to Access This Command
	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)
	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 70.

Configuring ACR/Summary Client

What Is a Configuration?

In ACR/Summary, 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.
- 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 16.

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 save it.

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 from your PC.

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 290.

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

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 108.

Setting Up UNIX or Linux Configurations

You need to set up a separate configuration with the **Server platform** field set to UNIX for either the 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/Summary 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 or Linux*.

Initializing the Definition and History Databases

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

- The definition database stores job, file, table, or history analysis definitions.
- The history database stores the results of balancing jobs for use in another run of the same job or in another job.

You can initialize both databases at once through the Database Initialization dialog box (select **Run > Database Initialization**). Use the dialog box help to complete the fields and run the initialization.

For more information on database initialization, see “Database Initialization” on page 254.

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, instructions for this can be found in [Appendix F, “Using ACR/Workbench”](#) on page 379 in the following sections:

- [Downloading and Uploading Between z/OS and ACR/Workbench](#) on page 381
 - [Importing Definitions and Input Sources from z/OS](#) on page 382
-

Uploading to UNIX or Linux

With a UNIX configuration selected from the drop down 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 [Chapter 14, “Importing Definitions”](#) on page 243.

Setting the User Options

What Are User Options?

User options enable you to override a variety of defaults for system-wide ACR/Summary processing and reporting options in the following categories: miscellaneous, currency formats, negative signs, date/time formats, XML options, and Infogix Assure information.

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

For information on setting user options, see [“Setting the User Options”](#) on page 165.

Selecting an Application Interface Mode

What Is an Application Interface Mode?

An application interface mode is the method ACR/Summary uses to interface with your applications. It determines where and how the balancing function obtains control values for processing. The following application interface modes are available in ACR/Summary Client:

- **File interface mode**—This is the typical application interface mode. It can be used to extract control values directly from an input source, which can be a report from your application, a sequential file, or a relational database table. These input sources are tied to most manual balancing procedures, so file interface mode is the logical choice for automating existing manual balancing operations.
- **Direct input mode**—This mode enables you to specify the control values to be stored in internal or extended internal items and used in balancing directly, rather than having ACR/Summary extract them from input sources. Direct input mode is useful for the following:
 - Processing data that is not available on your system. For example, your balancing might require entry of the current prime rate, obtained by making a telephone call to the bank.
 - Testing new balancing jobs. It allows you to validate ACR/Summary definitions and produce sample control reports without processing application programs or files.
 - Priming the history database.

Following are descriptions of file interface mode and direct input mode.

Understanding and Using File Interface Mode

How File Interface Mode Works

The following briefly describes how the file interface program functions.

1. UAC2000 uses the job name and step name to find all job and file definitions.
2. The file definitions are applied against the input source files, the control values are extracted, each control value is associated with an internal item, extended internal item, a cycle number, a job step qualifier (JSQ), or a rule set number.
3. The control values are then passed to the balancing function, which retrieves additional definitions and previously stored control values, performs balancing, stores the control values and the job results in the history database, and produces the Control Report.
4. If history analysis definitions exist for the job name and step name on the definition database, the history analysis function is invoked. The history analysis definitions are processed and the History Analysis Report is created.
5. A return code indicating an in- or out-of-balance condition is passed from the balancing function back to the file interface program.
6. The file interface program uses the **Set Return Code** option in the job definitions to control further processing. This option can be set to 0000 as a step completion code, or to some other 4-digit value.

Setting Up and Running a Job in File Interface Mode

Instructions for setting up and running a job in file interface mode are provided in “Setting Up and Running a Balancing Job” on page 65.

Understanding and Using Direct Input Mode

Purposes of Direct Input Mode

Direct input mode is useful for the purposes such as the following:

- Processing data that is not available on your system. For example, your balancing might require entry of the current prime rate, obtained by making a telephone call to the bank.
- Testing new balancing jobs. It allows you to validate ACR/Summary definitions and produce sample control reports without processing application programs or files.

1 ■ Preparing to Use ACR/Summary Client

Selecting an Application Interface Mode

- Priming the history database.

How Direct Input Mode Works

1. The direct input mode program reads sets of direct input transactions for a job ID.
2. It generates the direct input transaction report and invokes the balancing function.
3. Control values are processed according to the job definitions in the definition database.
4. Depending on the options specified in basic job information, the balancing function may produce the Control Report, store the job results in the history database, and produce the Recap Report.

The file containing the direct input data transactions can contain multiple sets of data. A single execution of direct input mode can be used to process one or more balancing steps.

Setting Up and Running a Job in Direct Input Mode

The steps for running balancing in direct input mode are as follows:

1. In Control Task (Job) View, create and save your job definitions. The process is very similar to setting up definitions for a job to be run in file interface mode described in “Setting Up and Running a Balancing Job” on page 65, except no input sources (file definitions) are used (if entered, they are ignored). To create an internal or extended internal item, in Control Task (Job) View select **New** under Internal Items or Extended Internal Items and complete the corresponding dialog box. (You will specify the value of each item in step 3 below.)
2. Select **Run > Direct Input Balancing** to display the Direct Input Balancing dialog box. Complete the Data Parameters section.
3. In the Direct Input Values section, only internal and/or extended internal items that have been created as described in step 1 will appear. (Additional items must be added in the Control Task [Job] View.) Highlight an item and click **Edit** to specify its value.
4. Use the **Additional** section if you want to specify an options file, which enables you to use the cycle override and history insert features. To do this, click **Add** or **Edit** to display the File ID and File Name dialog box, where you can add, edit, or remove the options file.

5. If you are generating the Control Report, User Report, or Free-Form Report in XML format, optionally select **Override** to change the default names for the files that will be used when writing these reports to a file in XML format.
6. Click **Run**. After the direct input transactions for the job are read, the direct input transaction report is generated and the balancing function is invoked.
7. View the Control Report and any other reports you specified when you set up the job to determine if there were processing errors and to determine whether the values are in- or out-of-balance. For instructions on viewing and interpreting reports, see “Reports” on page 289. The following are general guidelines:
Processing errors—If the status line at the bottom of the display or the System Messages (SYSOUT) Report shows #U system messages from Infogix or ACR/Summary return codes, look up the messages and return codes in the messages and codes manual for further information.

Note: When this documentation was published, ACR/Summary return codes included the following:

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

Out-of-balance situations—If the Control Report shows an out-of-balance condition, determine if modifications need to be made to the job. If the job needs to be rerun, it will execute with the same cycle ID, which automatically overlays the information previously stored in the history database for that job.

If you encounter processing errors that you cannot resolve, see “Contacting Customer Support” on page 24.

Analyzing Your Balancing Needs

Before you can set up a balancing job, you need to plan exactly what the job will do. Gathering this information may seem complicated at the outset, but even for a very complex job, the process is no more difficult than the type of analysis that would have to be used by someone performing the verification manually.

The analysis you need to perform is outlined below.

Determining the Application Interface Mode

The application interface modes available were discussed in “Selecting an Application Interface Mode” on page 38. The following sections assume you are using the typical mode, file interface mode, where values to be used in balancing are extracted from an input source.

Determining the Input Source Information

An input source is an application report or file that contains values you want to extract for use in balancing. 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.
 - 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.
 - VSAM Key Sequenced—This is an ISAM file in Windows, UNIX, or Linux.
 - Direct Spool Dataset—If this file organization type is used, the job can be run on z/OS only.
 - Oracle—Direct access to an Oracle table. If this file organization type is used, the job can be run on UNIX or Linux only.
 - 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.
 - User Program Accessed File—Access to files that cannot be directly accessed by ACR/Summary. For more information, see the *ACR/Summary and ACR/Detail User Program Guide*.

- The values that should be extracted from each input source. For example: Total sales, gross profit, profit margin for previous year and current year.
- Where the records containing the values to be extracted are located in the file. This will help determine the access mode and record/field definition types you should use to retrieve the values. For more information, see “Selecting an Access Mode” on page 120.
- How the extracted values should be formatted (for example, as counts, amounts, text items, dates, or time items).

Planning for Internal and Extended Internal Items

Determine the internal and/or extended internal items that should be defined. In your balancing 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 the next section)

Planning for Calculated Items and History Items

Determine if any calculated items need to be defined. 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.

Calculated items can be conditional or non-conditional. 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.

Determine if any history items need to be defined. A history item specifies retrieval from the history database of a value that was stored in a previous run of the current job or of a another specified job. This allows the value to be used in calculated items or rules that you define for the current job run. The retrieved value can be an internal or extended internal item, job run date, job run time, or cycle number.

Planning the Balancing Rules and Messages

Rules

Balancing 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. (The results of one rule can be used in another rule).

Return Codes

The return code is a 3- or 4-digit code that you define when you create a rule.

Note: 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 to report on the results of the job unless you specify otherwise. This report shows whether the values being balanced are in- or out-of-balance. When you set up the job, you will be able to select the level of detail you want to view on the Control Report, and you can set the number of copies (1 or 2) that you want. In addition to the Control Report, you can set up the following types of reports:

- (Balancing) User Report. See “Setting Up User Reports” on page 195.
- Free-form Report. See “Setting Up Free-Form Reports” on page 187.
- Recap Report. See “Set Return Code” on page 84.

You can optionally set several system-wide options for report formatting. See “Setting the User Options” on page 165.

For a basic example of analyzing your balancing needs, see “Determining Your Balancing Requirements” on page 46 in the Tutorial.

Tutorial

This chapter contains a tutorial that demonstrates the basic procedures for setting up and running an ACR/Summary balancing job. It consists of the following sections:

- “Preparing for the Tutorial” on page 45
- “Determining Your Balancing Requirements” on page 46
- “Creating the Balancing Job” on page 49
- “Running the Job” on page 60
- “Viewing the Output Reports and Handling Errors” on page 61

Note: A tutorial for ACR/Workbench for Summary is provided in “ACR/Workbench for Summary Tutorial” on page 385.

Preparing for the Tutorial

Before you begin this tutorial, you need to:

1. Access ACR/Summary Client. For instructions, see “Accessing and Exiting from ACR/Summary Client” on page 25.
You will see the Control Task (Job) View, where you can set up your balancing job for the tutorial.
2. Familiarize yourself with the product interface. See “Using the ACR/Summary 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/Summary for UNIX or Linux must be installed on the UNIX or Linux box you will use.
See “Configuring ACR/Summary Client” on page 34 for more information.
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.

Determining Your Balancing Requirements

4. Make sure the configuration you want to use for the tutorial is selected in the list box to the right of the toolbar when you start the tutorial.
5. Ensure that an integrity check will be run automatically before the database update. An integrity check scans your definitions for inconsistencies and missing data and displays a window reporting success or identifying problems.
 Select **Options** from the menu and ensure that there is a check mark next to **Integrity check**. If not, click on **Integrity Check**.
6. Ensure that you followed the instructions in “Initializing the Definition and History Databases” on page 36.
7. Verify that you have access to the sample reports that will be used as input sources in the tutorial. The report names are Updt0601.rpt and Summ0601.rpt. 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: The reports should be in ../sumdet/samples/ (your product installation subdirectory on UNIX or Linux).
 - Local mode: c:\Infogix\Summary32\sample is the default.

Determining Your Balancing Requirements

This tutorial assumes that you have a simple batch job that runs once a month and creates two reports: the Update Report and the Summary Report. You want to set up a balancing job to take values from each report and verify whether or not they are in balance based upon rules that you will define. The reports, which are described below, will be the input sources to your balancing job.

The Update Report (Updt0601.rpt) contains the valid and invalid transaction totals for new additions, changes, and deletions.

XYZ COMPANY	UPDATE CONTROL REPORT		JUN 01	PAGE 1
	INVALID	VALID		INPUT
	TRANSACTIONS	TRANSACTIONS		TRANSACTIONS
CHANGE TRANSACTIONS	37	4,800		4,837
DELETE TRANSACTIONS	19	1,900		1,919
ADD TRANSACTIONS	4	2,000		2,004
	----	-----		-----
TOTAL	60	8,700		8,760

The Summary Report (Summ0601.rpt) contains the total number of transactions.

XYZ COMPANY	S U M M A R Y	C O N T R O L	R E P O R T	PAGE 1
REPORT OP23-1-1		RUN DATE	JUN 01	
NUMBER OF TRANSACTIONS IN OLD MASTER				100,400
NUMBER OF TRANSACTIONS IN NEW MASTER				100,450
NUMBER OF VALID INPUT TRANSACTIONS				8,700

Determine your balancing requirements as follows:

1. Decide what you want to balance.

For this tutorial, assume you want to accomplish the following:

- Verify the total number of input transactions on the Update Report. You can do this by adding the total number in the Invalid Transactions column with the total number in the Valid Transactions column and comparing the result with the total number in the Input Transactions column.
- Verify that the total from the Valid Transactions column in the Update Report is equal to the Number of Valid Input Transactions from the Summary Report.

2. Identify the field items.

A field item must be created for each value that is extracted from an input source. For this tutorial, you will identify three field items from the Update Report and one from the Summary Report as shown below. The TOTAL row of the Update Control Report below shows in bold the field values to be extracted from the report: 60 (invalid transactions), 8,700 (valid transactions, and 8,760 (input transactions)

XYZ COMPANY	UPDATE CONTROL REPORT		JUN 01	PAGE 1
	INVALID	VALID	INPUT	
	TRANSACTIONS	TRANSACTIONS	TRANSACTIONS	
CHANGE TRANSACTIONS	37	4,800	4,837	
DELETE TRANSACTIONS	19	1,900	1,919	
ADD TRANSACTIONS	4	2,000	2,004	
	----	-----	-----	
TOTAL	60	8,700	8,760	

2 ■ Tutorial

Determining Your Balancing Requirements

The NUMBER OF VALID INPUT TRANSACTIONS row of the Summary Report shown below shows in bold the field item to be extracted from the report (8,700).

XYZ COMPANY	S U M M A R Y	C O N T R O L	R E P O R T	PAGE 1
REPORT OP23-1-1		RUN DATE	JUN 01	
NUMBER OF TRANSACTIONS IN OLD MASTER			100,400	
NUMBER OF TRANSACTIONS IN NEW MASTER			100,450	
NUMBER OF VALID INPUT TRANSACTIONS			8,700	

3. Identify the internal items.

Before it can be used in balancing rules, the value from each field item must be stored as an internal item.

Note: A longer value can be stored in an extended internal item. This type of item is not included in the tutorial. For more information, see "About Internal Items and Extended Internal Items" on page 88.

4. Identify the rules that will be used to manipulate the internal items.

- Rule 1 will be: INVALID TRANSACTIONS plus VALID TRANSACTIONS should equal INPUT TRANSACTIONS.
- Rule 2 will be: VALID TRANSACTIONS from the Update Report should be equal to VALID INPUT TRANSACTIONS from the Summary Report.

5. Determine the messages you need to define for each rule.

When defining a rule, you will assign a return code to it to identify an out-of-balance situation (a situation that does not conform to the rule). Each return code will then be associated with a message. Messages are included in the Control Report that is run at the end of the balancing job. They 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 messages in the tutorial:

- If Rule 1 is out of balance, the user should call Chris King for assistance.
- If Rule 2 is out of balance, the user should call Technical Support.

Creating the Balancing Job

Before you begin, ensure that you have completed the instructions in “Preparing for the Tutorial” on page 45.

Complete the Basic (Job) Information

The Control Task (Job) View appears when you start the product. This is the default view, which you will use in this tutorial to define a new balancing job. The Basic Information folder will be 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. Complete the fields shown in the table and select **OK** to save your entries.

Field	Enter or Select	Reason/Notes
Job Name	SAMPLE	Provide a meaningful name for the job.
Step Name	UPDATE	The step name field is used to identify parts of a multi-step job.
Report Title	MONTHLY FILE UPDATE SYSTEM	A descriptive title for the Control Report.

2. Click on the **Set Return Code** option to display the associated dialog box.

Select **Yes (highest balancing Return Code)** and then click **OK**. This sets the step completion code equal to the highest return code identified when a balancing step is out of balance or in error.

Name and Save Your Job (Task) File

Save the file by pressing **CTRL + S**, by selecting **File > Save**, or by selecting the **Save** icon.

Note: To identify the icons, hover the mouse over them or see “Toolbar Icons” on page 29.

Name your file Sample.tsk.

2 ■ Tutorial

Creating the Balancing Job

Define the First Input Source

Your first input source will be the Update Report, updt0601.rpt. Input sources are defined through the Input Source View. 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	UPDATE	Provide a meaningful name for the input source.
File Name	Enter the full path and file name of the Update Report, updt0601. When you select OK , the file will be displayed in the Extract Window at the bottom of the view.	<p>If you are in UNIX or Linux mode, the file should be in ../sumdet/samples/ (your product installation subdirectory on UNIX or Linux).</p> <p>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.</p> <p>If you are in Local mode, the file should be in C:\Infogix\Summary32\sample</p>

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

Basic Information				
Name:	UPDATE			
File Description:				
File Organization:	Physical Sequential, Access Mode 1			
File Comments:				
Base File Indicator:	Do not use this file as a base			
Record/Field Definitions				
< New... >				
KYZ COMPANY	UPDATE CONTROL REPORT	JUN 01	PAGE 1	
	INVALID	VALID	INPUT	
	TRANSACTIONS	TRANSACTIONS	TRANSACTIONS	
CHANGE TRANSACTIONS	37	4,800	4,837	
DELETE TRANSACTIONS	19	1,900	1,919	
ADD TRANSACTIONS	4	2,000	2,004	
TOTAL	60	8,700	8,760	

2. Click on the **File Description** option to display the File Description dialog box. In the **File Description** field, enter UPDATE REPORT and click **OK** to save.

For the other Basic Information options, keep the defaults.

3. You are now ready to start identifying the field items you want to use from the Update Report. Recall from “Determining Your Balancing Requirements” on page 46 that you need the following field items from this report:

- INVALID TRANSACTIONS
- VALID TRANSACTIONS
- INPUT TRANSACTIONS

All of these field items reside on the report line, which contains the word TOTAL.

You first need to define an embedded key record to locate the record (report line) where the items are located. An embedded key record finds a record by searching for a text string.

Begin defining the embedded key by highlighting the word TOTAL in the displayed report with your mouse.

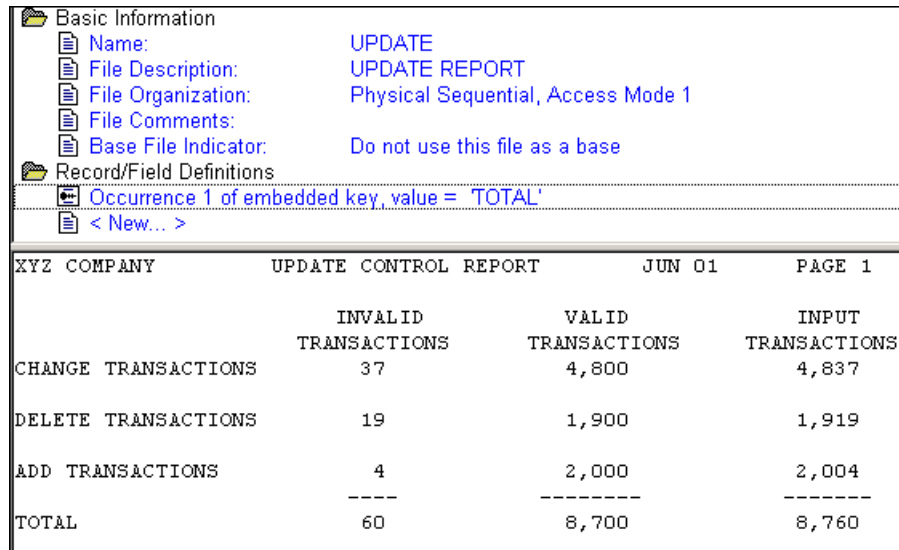
4. Select the **Embedded Key Record** icon from the toolbar to display the Embedded Key dialog box.

Because you selected the key value from the displayed report using the mouse (this is called the mark-and-capture method), the fields **Key Value**, **Start Position**, **Maximum Start Position**, **Occurrence**, and **Length** will be filled in automatically.

2 ■ Tutorial

Creating the Balancing Job

When you click **OK** to save your entries, the Input Source View is displayed showing your embedded key record:

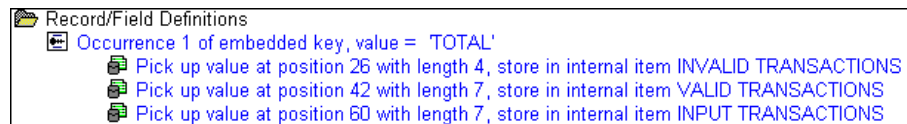


The screenshot shows a software interface with a toolbar at the top containing icons for 'Basic Information', 'Record/Field Definitions', and '< New... >'. Below the toolbar, the 'Basic Information' section lists: Name: UPDATE, File Description: UPDATE REPORT, File Organization: Physical Sequential, Access Mode 1, and Base File Indicator: Do not use this file as a base. The 'Record/Field Definitions' section shows 'Occurrence 1 of embedded key, value = 'TOTAL''. Below this is a table with the following data:

XYZ COMPANY	UPDATE CONTROL REPORT	JUN 01	PAGE 1
	INVALID	VALID	INPUT
	TRANSACTIONS	TRANSACTIONS	TRANSACTIONS
CHANGE TRANSACTIONS	37	4,800	4,837
DELETE TRANSACTIONS	19	1,900	1,919
ADD TRANSACTIONS	4	2,000	2,004
	----	-----	-----
TOTAL	60	8,700	8,760

- Now you can specify the field items for the values you want to extract from this embedded key record. Define the first field item by highlighting the number 60 with your mouse. Place the cursor a few spaces to the left of the value to allow for a larger number. This field shows the total INVALID TRANSACTIONS.
- Click the **Field Item** icon on the toolbar to open the associated dialog box. The **Position** and **Length** fields will be pre-filled.
- Define an internal item to store the extracted field value. Click **Create/Change** to open the Create/Change Internal Items dialog box and then click **<New...>** to open the Internal Item dialog box. In the **Name** field, enter INVALID TRANSACTIONS.
- Click **OK** on this dialog box, click **Close**, and then click **OK** on the Field Item dialog box to return to the Input Source view.
- Define the second field item by highlighting the number 8,700 with your mouse. Place the cursor a few spaces to the left of the value to allow for a larger number. This field shows the total VALID TRANSACTIONS.
- Click the **Field Item** icon on the toolbar to open the associated dialog box. The **Position** and **Length** fields will be pre-filled.

11. Define an internal item to store the field value. Click **Create/Change** to open the Create/Change Internal Items dialog box and then click **<New...>** to open the Internal Item dialog box. In the **Name** field, enter VALID TRANSACTIONS.
12. Click **OK** on this dialog box, click **Close**, and then click **OK** on the Field Item dialog box to return to the Input Source view.
13. Define the third field item by highlighting the number 8,700 with your mouse. Place the cursor a few spaces to the left of the value to allow for a larger number. This field shows the total INPUT TRANSACTIONS.
14. Click the **Field Item** icon on the toolbar to open the associated dialog box. The **Position** and **Length** fields will be pre-filled.
15. Define an internal item to store the field value. Click **Create/Change** to open the Create/Change Internal Items dialog box and then click **<New...>** to open the Internal Item dialog box. In the **Name** field, enter INPUT TRANSACTIONS.
16. Click **OK** on this dialog box, click **Close**, and then click **OK** on the Field Item dialog box to return to the Input Source view.



17. Close the Input Source - UPDATE view and save the job (**CTRL + S**).

2 ■ Tutorial

Creating the Balancing Job

Define the Second Input Source

Your second input source will be the Summary Report, summ0601.rpt. To begin defining it, click on **New** under Input Sources to open a new Input Source View.

1. Complete the Basic Information section following the same procedure you used for the first input source. Click **Name** to display the Name dialog box and complete only the fields shown in the following table.

Field	Enter or Select	Reason/Notes
File ID	SUMMARY	Provide a meaningful name for the input source.
File Name	Full path and file name of the Summary Report, summ0601. 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:\Infigix\Summary32\sample

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

Basic Information	
Name:	SUMMARY
File Description:	
File Organization:	Physical Sequential, Access Mode 1
File Comments:	
Base File Indicator:	Do not use this file as a base
Record/Field Definitions	
< New... >	

XYZ COMPANY	S U M M A R Y	C O N T R O L	R E P O R T	P A G E 1
REPORT OP23-1-1		R U N D A T E	J U N 0 1	
NUMBER OF TRANSACTIONS IN OLD MASTER			100,400	
NUMBER OF TRANSACTIONS IN NEW MASTER			100,450	
NUMBER OF VALID INPUT TRANSACTIONS			8,700	

2. Click the **File Description** option to display the File Description dialog box. In the **File Description** field, enter SUMMARY REPORT and click **OK** to save.

Note: For this tutorial, we will keep the default values for the other Basic Information options.

3. Recall from “Determining Your Balancing Requirements” on page 46 that you need the following field item from the Summary Report
VALID INPUT TRANSACTIONS
4. Define the record (report line) where this item is located. The value you want is on the line containing the character string NUMBER OF VALID INPUT TRANSACTIONS. You can define an embedded key record to identify the report line by searching for this character string. Highlight the words NUMBER OF VALID INPUT TRANSACTIONS in the report with your mouse.
5. Select the **Key** icon from the toolbar to display the Embedded Key dialog box. The necessary fields will be filled in automatically. Click **OK** to save your entries and view your new embedded key record definition in the Input Source View.
6. Now you can specify the value you want on this record. Highlight the number 8,700 on the report with your mouse. Place the cursor a few spaces to the left of the value to allow for a larger number. This is the total VALID INPUT TRANSACTIONS.
7. Click the **Field Item** icon on the toolbar to open the associated dialog box. The **Position** and **Length** fields will be pre-filled.
8. Define an internal item to store the field value. Click **Create/Change** to open the Create/Change Internal Items dialog box and then click **<New...>** to open the Internal Item dialog box. In the **Name** field, enter VALID INPUT TRANSACTIONS.
9. Click **OK** on this dialog box, click **Close**, and then click **OK** on the Field Item dialog box to return to the Input Source view.
10. Close the Input Source window.
11. Save the job (**CTRL + S**).

Define the First Balancing Rule

You can now define the rules 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 Balancing Requirements” on page 46 that the first rule will be: INVALID TRANSACTIONS (internal item 1) plus VALID TRANSACTIONS (internal item 2) should be equal to INPUT TRANSACTIONS (internal item 3).

- 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**.
- 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	TOTAL RECORDS RULE	
Return Code	3001	Indicates the code (0001-3999) that you want to be set if the rule is out of balance.

- Select the Comparison tab. The **Available Items** field on the right side of the screen shows the four internal items you defined for this job. We will use three of these items in this rule.
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	INVALID TRANSACTIONS	Click this to highlight.

Field	Enter or Select	Reason/Notes
Add	Click this.	Enters INVALID TRANSACTIONS in the Included Items field.
	VALID TRANSACTIONS	Click this to highlight.
Add	Click this.	Enters VALID TRANSACTIONS in the Included Items field with the + sign preceding it.

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

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

5. Click on the General Information tab and verify that the Rule is defined as follows in the Rule area at the bottom of the dialog: (INVALID TRANSACTIONS + VALID TRANSACTIONS) = (INPUT TRANSACTIONS). If not, click on the Comparison tab and make the adjustments.
6. Click **OK** to display the Direct Message dialog box.

Define the First Message

We will define a direct message to be generated when the 3001 return code defined in the previous section is issued, indicating an out-of-balance condition.

Note: The other type of message, an indirect message, enables you to reuse (in this job) one or more direct messages defined in another job.

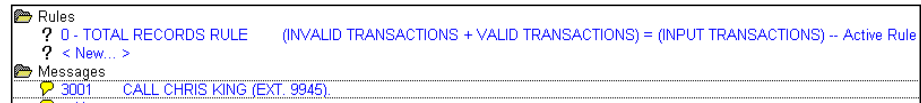
2 ■ Tutorial

Creating the Balancing Job

Complete the Direct Message dialog box fields as shown in the following table.

Field	Enter or Select	Reason/Notes
Return Code	Verify the Return Code: 3001.	
Message Text	CALL CHRIS KING (EXT. 9945).	This is an example of an instructional message notifying someone about the discrepancy.

Click **OK** to save the message and return to the Control Task (Job) View. Your first balancing rule and message will look like this:



Define the Second Balancing Rule

Recall from “Determining Your Balancing Requirements” on page 46 that the second rule will be: VALID TRANSACTIONS from the Update Report should be equal to VALID INPUT TRANSACTIONS from the Summary Report.

1. Under the Rules folder, click **<New>** to display the Rule Type dialog box.
2. Accept the default selection (Standard Rule) and click **OK** to display the Standard Rule dialog box.
3. When the dialog box opens, on the General Information tab, complete only the fields shown in the following table.

Field	Enter or Select	Reason/Notes
Name	REPORT COMPARISON RULE	
Rule Action	Set Return Code	Sets the return code shown in the Return Code field on this dialog box.
Return Code	3021	Indicates the code (0001-3999) that you want to be set if the rule is out of balance.

4. Select the Comparison tab and complete the left-hand side of the condition as shown in the following table.

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	VALID TRANSACTIONS	Click this to highlight.
Add	Click this.	Enters VALID TRANSACTIONS in the Included Items field.

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

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

6. Click on the General Information tab and verify that the Rule is defined as follows in the Rule area at the bottom of the dialog: (VALID TRANSACTIONS) = (VALID INPUT TRANSACTIONS). If not, click on the Comparison tab and make the adjustments.
7. Click **OK** to display the Direct Message dialog box.

2 ■ Tutorial

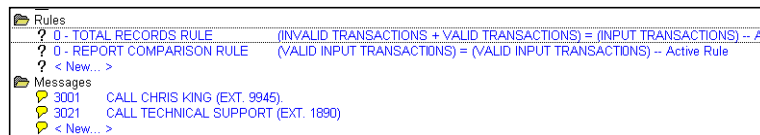
Running the Job

Define the Second Message

We will define a direct message to be generated when the 3021 return code defined in the previous section is issued, indicating an out-of-balance condition. Complete the dialog box fields as shown in the following table.

Field	Enter or Select	Reason/Notes
Return Code	Verify the Return Code: 3021.	
Message Text	CALL TECHNICAL SUPPORT (EXT. 1890).	

Click **OK** to save the message and return to the Control Task (Job) View. Your second balancing rule and message will look like this:



Save the job (**CTRL + S**).

Running the Job

Click the **Balancing** icon to display the Balancing 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, enter a meaningful number such as today's date. For this tutorial, keep the default entries for the other fields.
2. Click **Run**. 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

See “Viewing the Output Reports and Handling Errors” on page 69 for instructions.

The report will display values for the internal items, indicate that both rules are in balance, and will also show the step completion code.

2 ■ Tutorial

Viewing the Output Reports and Handling Errors

Setting Up a Balancing Job

This chapter explains what a balancing job is and provides instructions for setting one up. In addition, this chapter introduces the Basic Job Information portion of the ACR/Summary Client interface.

This chapter includes the following sections:

- “What is a Balancing Job?” on page 63
- “Preparing to Set Up a Balancing Job” on page 64
- “Setting Up and Running a Balancing Job” on page 65
- “Viewing the Output Reports and Handling Errors” on page 69
- “Working with Job Definitions” on page 70

What is a Balancing Job?

The basis of ACR/Summary's functionality is the balancing job. A typical balancing job, run in file interface mode, requires both job and file definitions, as follows:

Note: A balancing job run in direct input mode requires only job definitions. This is because the control values to be balanced are input directly by the user. For more information, see "Understanding and Using Direct Input Mode" on page 39.

- Job definitions specify how values (typically extracted from input sources using file definitions) are processed. Job definitions include definitions for:
 - Internal Items
 - Extended Internal Items
 - Extraction Variables
 - History Items
 - Calculated Items
 - Rules
 - Messages
 - Reports
- More details on job definitions are provided in Chapter 5, “Setting Up Job Definitions.”

3 ■ Setting Up a Balancing Job

Preparing to Set Up a Balancing Job

- File definitions specify how values will be extracted from one or more input sources (often reports from your application). More details on file definitions will be provided in Chapter 6, “Setting Up Input Sources.”

Note: Definitions for external translation tables, internal translation tables, cycle tables, and history analysis reports are not classified as job definitions. However, a job may require one or more types of tables, and you may need to specify definitions for a history analysis report to be run in conjunction with the job.

When a balancing job is run in file interface mode, user-defined rules are applied to data extracted from your input sources to verify the accuracy of your key business information. For more information, see “Understanding and Using File Interface Mode” on page 39.

A single application requiring verification typically needs only one balancing job, but several may be required. Each job is uniquely identified by a job ID, consisting of a job name, step name, and qualifier. For more information on qualifiers, see “Job/Step/Qualifier (JSQ)” on page 76.

When created through ACR/Summary Client, a balancing job is also called a control task and has the extension .tsk.

Preparing to Set Up a Balancing 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/Summary Client.”

- Configured ACR/Summary Client.
- Initialized your definition and history databases.
- Familiarized yourself with interface elements such as views, folders, menus, toolbar icons, and dialog boxes.
- Analyzed your balancing needs so that you know exactly what you want to accomplish.

Setting Up and Running a Balancing Job

This section provides an overview of the steps involved in setting up a balancing job through ACR/Summary Client.

These instructions are for a job that will run in file interface mode, the typical application interface mode. In file interface mode, user-defined rules are applied to data extracted from application files or reports (input sources) to verify the accuracy of the information.

Note: For information on direct input mode, which is the other application interface mode available in ACR/Summary Client, see "Understanding and Using Direct Input Mode" on page 39.

Note: For information on working with existing job definitions, see "Working with Job Definitions" on page 70.

The procedure is as follows:

1. Open a new Control Task (Job) View. This is the default view when you start ACR/Summary. You can also open this view by selecting **File > New > Control Task (Job)**.
2. Use the options in the Basic Information folder to set up the basic job information. Only the Name option must be completed for every balancing job. The other options have default settings which you can change if appropriate.

For more information on this step, refer to Chapter 4, "Setting Up Basic Job Information" on page 75 and see the dialog box help for each option.

3. Use the Input Sources folder to open the Input Source View and set up your first input source.

Note: You may prefer to define your internal and/or extended internal items (see step 7) before you complete this step.

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

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

- Location and name of the file or its record layout.
- File organization type, such as physical sequential file, Oracle database file, etc.

3 ■ Setting Up a Balancing Job

Setting Up and Running a Balancing Job

- Access mode, which determines how the file will be accessed. Access modes are not applicable to relational database sources.
- Record/field definitions. These define records to select and values to extract from these records.

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

Chapter 6, "Setting Up Input Sources" on page 115 contains detailed information on setting up input sources.

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

Internal items (regular and extended). These are storage places for the following types of data to be used in your balancing rules:

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

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 internal or extended internal items extracted in previous job runs (and stored in history records) that will be used in the current job.

Calculated items. These specify mathematical manipulation of values from internal items, extended internal items, history items, other calculated items, or literals (numeric constants). 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.

Rules. These specify the criteria that will be used to determine if the values extracted from your input sources are in or out of balance. Rules compare the extracted values and evaluate the results. The results of a rule can be evaluated in other rules. 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.

Alternate control tasks. This job definition type is applicable to z/OS only. For more information, see "Alternate Control Tasks" on page 387.

Recap items. A recap item defines the jobs to include in the Recap Report.

Additional information for creating each type of job definition referenced in this step is provided in Chapter 5, “Setting Up Job Definitions.”

5. Use the Report Information folder in Control Task (Job) View if you want to create the following customized reports:
 - Balancing User Report (often called 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 User Reports” on page 195.
 - Free-Form Report.** This is a fully customizable report on the job results. For more information, see “Setting Up Free-Form Reports” on page 187.
6. The Control Report is the primary output of an ACR/Summary balancing run. To customize the report to your needs, see “Customizing the Control Report to Your Needs” on page 293.
7. Click the **Save** icon to save the job.
8. Click the **Database Update** icon to update the definition database.

Note: When you update the definition database, ACR/Summary automatically performs an integrity check. For more information on the integrity check feature, see “Integrity Check” on page 264.

9. If you have not already done so, set up any tables required for your job. You may need the following table types:
 - **Translation table**—A table used to translate values from an input source for balancing purposes. Two types are available:
 - Internal translation table—A static table that you create and populate within ACR/Summary. It is used 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 balancing purposes. Unlike an internal translation table, an external translation table is built at run time, usually by referencing data external to ACR/Summary. External translation tables are used when an internal translation table (which performs only simple translations from one text value to another text value) will not suffice.
For more information on translation tables, see “Using Translation Tables” on page 221.

3 ■ Setting Up a Balancing Job

Setting Up and Running a Balancing Job

- **Cycle table**—A table that can be used to verify cycle numbers. For example, if a company wants a particular job to run only on Mondays, a cycle table with the Monday dates of the year could be created. For more information on cycle tables, see “Setting Up Cycle Tables” on page 209.
10. After saving each table, update the definition database with the table definitions by clicking the **Database Update** icon.
 11. Prepare to run the job. Click the **Balancing** icon to open the Balancing dialog box. The dialog box help provides field definitions and other information.

From the Balancing dialog box, you can do any of the following:

 - Optionally click **Overrides** to specify run-time overrides for selected options from the Basic Job Information folder.
 - Use the File Interface Input section to:
 - Add an input source that is not defined in the job, remove a selected input source, or edit a selected input source. In this case, if your job contains qualifiers, you will need to add additional input source file names for each job name, step name, qualifier in addition to the base.
 - Specify an options file, which will enable you to use the cycle override or history insert feature to set or insert the cycle number. For more information, see “Options File (UNIOF.DAT) Features for Cycle Processing” on page 202.
 - Request a Trace Report on input sources that use access mode 6 or 7. For more information, see “Trace Report - UNIETR.RPT” on page 298.
 12. Click **Run** to run the job.

Viewing the Output Reports and Handling Errors

View the reports to see if there were processing errors and to determine whether your input sources are in- or out-of-balance. For instructions on viewing and interpreting reports, see “Working with Reports on ACR/Summary Client” on page 289.

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 system messages or ACR/Summary return codes, look up the messages and return codes in the messages and codes manual for further information.

Note: When this documentation was published, ACR/Summary return codes included 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 will execute with the same cycle ID, which automatically overlays the information previously stored in the history database for that job.

If you encounter errors that you cannot resolve, see “Contacting Customer Support” on page 24.

Working with Job Definitions

This section provides instructions for saving, opening, copying, pasting, and deleting existing definitions.

Saving Definitions

When you save a job, table, or history analysis document created through ACR/Summary Client, you specify a name and location where the files containing your definitions will be stored. You then need to run a database update in order to add the definitions to the definition database before you can use the definitions you created in a job run.

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 a job, file, table, or set of history analysis definitions and modify it, you should save the changes and run the database update again.

Opening Existing Definitions

To open an existing job, table, report, or history analysis document, select **File > Open**. When the Open dialog box appears, you can control the type of files displayed by changing the setting in the **Files of Type** field. After you select the file you want, the appropriate view will open displaying the selected job, table, etc.

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. The 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 Record/Field 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 or Free-Form Report, 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**.

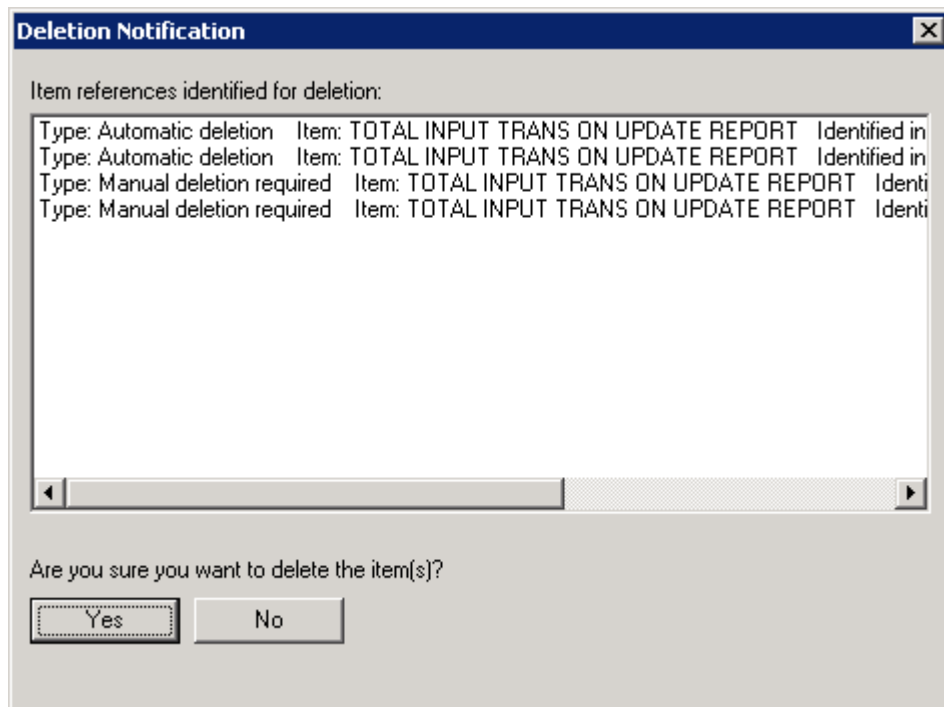
Deletion Notification dialog box for Referenced Items

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

3 ■ Setting Up a Balancing Job

Working with Job Definitions

An example of this dialog box is shown below:



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 or Free-Form Report.

Lines in format 1 identify references to the item to be deleted that currently appear in the User Report or Free-Form Report. When 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.

Format 3: Manual Deletion Information for an Alternate Balancing Rule

"Item *rule name* deletion omitted in User Report definitions"

"Item *rule name* deletion omitted in Free-Form Report definitions"

Where *rule name* is the name of the alternate balancing rule to be deleted.

Lines in format 3 (not shown in the sample dialog box above) indicate that you marked an alternate balancing rule for deletion, and your job definitions include a User Report and/or Free-Form Report. You must delete any references to the deleted rule from these reports 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

For this information, see "Character to use as the time separator (instead of the default colon)." on page 161.

3 ■ Setting Up a Balancing Job

Working with Job Definitions

Setting Up Basic Job Information

As explained in “Setting Up and Running a Balancing Job” on page 65, you start creating your job by completing the Basic Information options at the top of the Control Task (Job) View. This chapter provides an overview of these options. It contains the following sections:

- “Basic (Job) Information Options” on page 75
- “Name” on page 76
- “Length of Descriptions” on page 77
- “Report Headings” on page 77
- “Store History” on page 77
- “Print Control Reports” on page 77
- “Extraction File Options” on page 78
- “Access User Exit” on page 83
- “Print Recap Report” on page 83
- “Set Return Code” on page 84
- “Cycle Processing” on page 84
- “Automatic Rerun/Audit Trail Options” on page 84
- “Round Results” on page 84
- “Use 22-character Numeric Format” on page 85
- “Control Report XML Option” on page 85
- “Adding Job Comments” on page 85

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.

4 ■ Setting Up Basic Job Information

Name

Name

The Control Task (Job) Name dialog box allows you to assign a job ID, name the Control Report, specify a model job ID (if you are creating a model user for job modeling), and specify file control balancing. For more information on job modeling, see “Modeling Job Definitions” on page 268. For more information on file control balancing, see the dialog box help.

Job ID

The Job ID is made up of 3 parts, the job name, step name, and qualifier. The **Qualifier** field is optional. This field is completed when you want to create a job/step/qualifier (JSQ).

Job/Step/Qualifier (JSQ)

The JSQ feature enables multiple invocations of the balancing function with the same job name/step name. The base job (the job that will run first) is created with a job name and step name and blanks in the Qualifier field. Each JSQ is assigned a unique qualifier to indicate the processing sequence. Use 00, 01, etc., or alpha values such as AA, A1, etc. After the base job runs each JSQ is run in ascending order. Within any JSQ, the items from other JSQs can optionally be referenced as history items.

JSQs are useful in the following situations:

- A job needs to have more than the maximum of 100 internal items, extended internal items, and/or calculated items. You could use one or more JSQs to process the additional items.
- You need to use a unique naming convention for business entities (divisions, departments, companies, etc.).

Control Report Title

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

Model Definition Section

The Model Definition section of the dialog box enables you to assign a model job ID if you are referencing definitions from another job. For more information, see “Modeling Job Definitions” on page 268.

Job Definition Section

The Job Definition section enables you to specify file control balancing, which is applicable only if you will be running your job on z/OS.

Length of Descriptions

This dialog box allows you to specify the maximum number of characters allowed for descriptions of internal items, extended internal items, history items, calculated items, and rules.

Report Headings

The Report Headings dialog box enables you to:

- Modify any one of the top three lines of your Balancing Control Report.
- Enter your own descriptive titles for the two columns on the right side of the report. The default values for these columns are Input and Output.

Store History

The Store History dialog box allows you to specify whether you want to store history records for this job in the history database. Each history record contains the job ID, cycle ID, and the control value extracted for each internal or extended internal item. You can also specify how many history records you want to keep.

History records stored in the history database can be used in another run of the same job or in another job.

Print Control Reports

This option opens the Job Output Options dialog box, where you can set your preferences for the Control Report, which is the primary output of a balancing job.

4 ■ Setting Up Basic Job Information

Extraction File Options

For information on customizing this report, see “Control Report - UNIACR.RPT” on page 292.

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

Extraction File Options

The Extraction File Options dialog box allows you to create an formatted extract file containing information extracted during balancing. Depending on your specifications, the file can contain any combination of the following: internal items, extended internal items, history items, calculated items, rules, and messages. The extract file is a flat file named uniextr.dat that can be read into another job as input. It can provide a means of downloading balancing results for use in PC applications, passing results to other platforms, and generating correction transactions based on the balancing results.

The layout of the extract file is as follows:

Extract File (UNIEXTR) Record Layout

Header Record Layout

Fld#	Description	Position	Length	Format	Value
1	Job ID	1-18	18	X(18)	
2	Cycle ID	19-29	11	X(11)	
3	Record Type	30-31	2	X(2)	A1
4	Run Date	32-37	6	X(6)	
5	Run Time	38-43	6	X(6)	
6	Reserved	44-200	157	X(157)	Blanks

Internal Item Record Layout

Fld#	Description	Position	Length	Format	Value
1	Job ID	1-18	18	X(18)	
2	Cycle ID	19-29	11	X(11)	
3	Record Type	30-31	2	X(2)	B1
4	Internal Item Number	32-34	3	X(3)	001-100
5	Internal Item Value Sign	35	1	X(1)	+ or -
6	Value	36-50	15	X(15)	Control Value
7	Description	51-130	80	X(80)	
8	Reserved	131-200	80	X(80)	Blanks

Extended Internal Item Record Layout

Fld#	Description	Position	Length	Format	Value
1	Job ID	1-18	18	X(18)	
2	Cycle ID	19-29	11	X(11)	
3	Record Type	30-31	2	X(2)	G1
4	Internal Item Number	32-34	3	X(3)	001-100
5	Internal Item Value Sign	35	1	X(1)	+ or -
6	Value	36-115	80	X(80)	Control Value
7	Description	116-195 80	X(80)		
8	Reserved	196-200	5	X(5)	Blanks

4 ■ Setting Up Basic Job Information

Extraction File Options

History Item Record Layout - Record 1 of 2

Fld#	Description	Position	Length	Format	Value
1	Job ID	1-18	18	X(18)	
2	Cycle ID	19-29	11	X(11)	
3	Record Type	30-31	2	X(2)	C1
4	History Item Number	32-34	3	X(3)	001-100
5	History Item Value Sign	35	1	X(1)	+ or -
6	Value	36-115	80	X(80)	Control Value
7	Description	116-195	80	X(80)	
8	Reserved	196-200	5	X(5)	Blanks

History Item Record Layout - Record 2 of 2

Fld#	Description	Position	Length	Format	Value
1	Job ID	1-18	18	X(18)	
2	Cycle ID	19-29	11	X(11)	
3	Record Type	30-31	2	X(2)	C2
4	History Item Number	32-34	3	X(3)	001-100
5	EITB Job ID	35-52	18	X(18)	
6	EITB Cycle ID	53-63	11	X(11)	
7	Relative Cycle	64-66	3	X(3)	-99 to +00
8	Reserved	67-200	134	X(134)	Blanks

Calculated Item Record Layout - Record 1 of 2

Fld#	Description	Position	Length	Format	Value
1	Job ID	1-18	18	X(18)	
2	Cycle ID	19-29	11	X(11)	
3	Record Type	30-31	2	X(2)	D1
4	Calculated Item Number	32-34	3	X(3)	01-100
5	Calculated Value Sign	35	1	X(1)	+ or -
7	Calculated Value	36-65	30	X(30)	
7	Description	66-145	80	X(80)	Blanks
8	Reserved	146-200	55	X(55)	

Calculated Item Record Layout - Record 2 of 2

Fld#	Description	Position	Length	Format	Value
1	Job ID	1-18	18	X(18)	
2	Cycle ID	19-29	11	X(11)	
3	Record Type	30-31	2	X(2)	D2
4	Calculated Item Number	32-34	3	X(3)	001-100
5	Formula	35-154	120	X(120)	
6	Reserved	155-200	46	X(46)	Blanks

4 ■ Setting Up Basic Job Information

Extraction File Options

Balancing Rule Record Layout - Record 2

Fld#	Description	Position	Length	Format	Value
1	Job ID	1-18	18	X(18)	
2	Cycle ID	19-29	11	X(11)	
3	Record Type	30-31	2	X(2)	E2
4	Rule Number	32-34	3	X(3)	001-100
5	Formula	35-154	120	X(120)	
6	Tolerance Value Sign	155	1	X(1)	+ or -
7	Tolerance Value	156-185	30	X(30)	
8	Reserved	186-200	15	X(15)	Blanks

Balancing Rule Record Layout - Record 2 of 2

Fld#	Description	Position	Length	Format	Value
1	Job ID	1-18	18		X(18)
2	Cycle ID	19-29	11	X(11)	
3	Record Type	30-31	2	X(2)	E2
4	Rule Number	32-34	3	X(3)	001-100
5	Formula	35-114	120	X(120)	
6	Tolerance Value Sign	115-115	1	X(1)	+ or -
7	Tolerance Value	116-185	30	X(30)	
8	Reserved	186-200	15	X(15)	Blanks

Balancing Rule Record Layout - Record 6 (Special Instructions)

Fld#	Description	Position	Length	Format	Value
1	Job ID	1-18	11	X(11)	
2	Cycle ID	19-29	11	X(11)	
3	Record Type	30-31	2	X(2)	F1
4	Special Instruction Number	32-34	3	X(3)	01-100
5	Return Code	35-38	4	X(4)	0000-9999
6	Text	39-118	80	X(80)	
7	Reserved	119-200	82	X(82)	Blanks

Access User Exit

The Access User Exit dialog box enables you to select the option you want for the File Interface user exit program (UNIEXIT). UNIEXIT is a COBOL program that you can modify to perform custom processing.

UNIEXIT controls the ability to set return codes and kick off additional jobs based on balancing results. It works in conjunction with the options you set in the Set Return Code and the Standard Rule dialog boxes.

This dialog box affects processing only 1) when you are running the job in file interface mode and 2) when you have modified the standard COBOL shell program.

Print Recap Report

The Print Recap Report dialog box allows you to indicate whether to print the Recap Report after the job is run. A Recap Report can be used to provide a high-level summary of one or multiple jobs for daily processing, for variable cycle processing, or for a complete system or subsystem. If you want a Recap Report, you must use the Recap Items option that appears near the bottom of the Control Task (Job) View to define each Recap Item. Each recap item identifies a job to be included in the report. For more information, see “Recap Items” on page 112.

Set Return Code

The return code is a code that you define when you create a rule. 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.

The code is generated when the rule detects an out-of-balance condition. 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.

The return code appears on the status bar and in the Control Report.

The Set Return Code dialog box allows you to specify whether the highest return code generated will actually be set and whether an abend will be forced. You can choose one of the following:

- No, do not set the highest return code. Reset the highest return code generated to 0000.
- Yes, set the highest return code generated.
- Abend. This option is applicable only if you are running jobs on z/OS.

Cycle Processing

The Cycle Processing dialog box allows you to specify how the cycle ID (made up of the cycle number and run number) will be assigned to each job run.

For more information on cycle processing, see “Setting Up Cycle Processing” on page 199.

Automatic Rerun/Audit Trail Options

This dialog box is applicable only if you are running jobs on z/OS.

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.

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.

Control Report XML Option

The Control Report XML Option 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.

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**.

4 ■ Setting Up Basic Job Information

Adding Job Comments

Setting Up Job Definitions

Job definitions specify how values (typically extracted from input sources) are processed. This chapter provides information for setting up several types of job definitions. All of the definitions referenced here are set up through folders that appear on the Control Task (Job) View. Typically, you will complete the Basic Information folder options as described in Chapter 4, “Setting Up Basic Job Information” on page 75 before you set up the job definitions described here. This chapter contains the following sections:

- “About Internal Items and Extended Internal Items” on page 88
- “Internal Items” on page 90
- “Extended Internal Items” on page 90
- “Extraction Variables” on page 91
- “History Items” on page 96
- “Calculated Items” on page 97
- “Rules” on page 99
- “Defining Messages” on page 108
- “Alternate Control Tasks” on page 112
- “Recap Items” on page 112
- “Using Batch Options to Control Item Numbers” on page 113

Note: Your specifications for reports are also technically part of your job definitions. For information on reports, see the following:

- Setting Up Basic Job Information on page 75 (see sections relevant to reports in the list at the beginning of the chapter)
 - Setting Up Free-Form Reports on page 187
 - Setting Up User Reports on page 195
 - Appendix B, “Reports” on page 289
-

About 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 record/field 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.

Both regular and extended internal items are given names by you and numbers by the system. The numbers are assigned sequentially and are displayed in the Control Task (Job) View.

You can use a maximum of 100 regular and 100 extended internal items in the same balancing job. If you need more than 100 internal items in a job, you can use a job/step/qualifier (JSQ). For more information, see “Job/Step/Qualifier (JSQ)” on page 76.

Differences Between Regular and Extended Internal Items

Internal Items

- 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

- In the interface, 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 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 136.

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.

Internal Items

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

Setting Up Internal Items

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

- From the Input Source View, through one of the following dialog boxes: Field Item dialog box, Detail Field dialog box, Total Record dialog box, Tally Record, or Hash Record dialog box
On any of these dialog boxes, after specifying that the value should be stored in an Internal Item, if you have not yet created an internal item to store the results, you click **Create**. This opens the Internal Item dialog box.
- Directly from the Control Task (Job) View by clicking **New** under Internal Items in the Input Sources folder. You will need to do this if you want to set up your internal items before defining their input sources, or if you want to define an internal item to store a calculated item in the history database.

The dialog box help provides field definitions and other information.

Extended Internal Items

Extended internal items were explained in the “Differences Between Regular and Extended Internal Items” on page 88.

Setting Up Extended Internal Items

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

- From the Input Source View, through one of the following dialog boxes: Field Item dialog box, Detail Field dialog box, Total Record dialog box, Tally Record, or Hash Record dialog box
On any of these dialog boxes, after specifying that the value should be stored in an extended internal item, if you have not yet created an extended internal item to store the results, you click **Create**. This opens the Extended Internal Item dialog box where you specify the item.

- Directly from the Control Task (Job) View by clicking **New** under Extended Internal Items in the Input Sources folder. You will need to do this if you want to set up your extended internal items before defining their input sources, or if you want to define an extended internal item to store a calculated item in the history database.

The dialog box help provides field definitions and other information.

Extraction Variables

Extraction variables are storage places that hold regular or extended data from file definitions or external translation table definitions for further processing. You can specify up to 999 extraction variables per job.

Extraction Variables from File Definitions and Table Definitions are Not Shared

In ACR/Summary for Windows, extraction variables are not shared between jobs and external translation tables. An extraction variable you create for use in a file definition can be referenced only by another file definition. An extraction variable you create for use in an external translation table definition can be referenced only by another external table definition.

Specifying Extended Formatting for an Extraction Variable

An extraction variable should be in extended format 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 that an extraction variable is extended, select **Use extended format** on every file definition and/or external translation table definition that references it, as detailed below.

Consistency in Specifying Extended Format for an Extraction Variable

When setting the selecting the **Use extended format** field for an extraction variable, you must be consistent. For example, for extraction variable 1 in a table definition, you should not select **Use extended**

5 ■ Setting Up Job Definitions

Extraction Variables

format in one file or table definition and, when you reference extraction variable 1 in another file definition, leave **Use extended format** deselected.

Warning: If you are not consistent in selecting **Use extended format** 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.

Extraction Variables in (Record/Field) File Definitions

Extraction Variable Uses in File Definitions

The following table summarizes extraction variable uses in file definitions:

Access Mode	Definition	Source	Target
1 through 4	Field Item	Input Source	Extraction Variable
1 through 4	Embedded Key ¹	Extraction Variable	Embedded Key
5	Tally Item	Input Source	Extraction Variable
5	Total Item	Input Source	Extraction Variable
5	Hash Item	Input Source	Extraction Variable
6 and 7	Detail Field	Input Source	Extraction Variable
6 and 7	Detail Field	Extraction Variable	Internal Item
6 and 7	Detail Field	Extraction Variable	Extraction Variable
6 and 7	Selection Field ²	Extraction Variable	Selection Criteria

1 The use of extraction variables to store a value from an embedded key and pass it to another embedded key is called embedded key substitution. Embedded key substitution is explained in Setting Up Embedded Key Substitution on page 93 below.

2 If the input type is extraction variable, the value from the specified extraction variable can be evaluated against one or more values or ranges. If the input type is Input Area, the value from the input area can be evaluated against the value of a specified extraction variable.

Defining Extraction Variables for Record/Field (File) Definitions

An extraction variable to be used in a file definitions is defined by specifying its name and description in the Extraction Variable dialog box. The extraction variables that you define in extraction definitions at the control task (job) level will appear in the Extraction Variables folder in the Control Task (Job) View.

To define an extraction variable for use in file definitions, 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.
- In the Control Task (Job) View, select **New** under Extraction variables and complete the Extraction Variable dialog box.

If the extraction variable is extended, select **Use extended format** in every file definition that references it and identify it as extended in the **Name** field of the Extraction Variable dialog box. See “Specifying Extended Formatting for an Extraction Variable” on page 91.

Automatic Generation of Extraction Variable Definitions Specified in Record/Field (File) 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 Control Task (Job) Window. Saving the job will move them to the definition database.

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 generated descriptions.

Storing and Saving Extraction Variables Specified in Record/Field (File) Definitions

The extraction variables that you define when creating record/field (file) definitions will be saved as part of the job definitions.

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

Setting Up Embedded Key Substitution

In access modes 1 through 4, extraction variables can be used in embedded key substitution. This technique provides you with more flexibility in locating values across multiple input sources.

5 ■ Setting Up Job Definitions

Extraction Variables

To use embedded key substitution, in one input source, save a field item as an extraction variable. Then use the value of that extraction variable as the value of an embedded key in a second input source.

When defining the embedded key in the second input source, do the following:

- Enter the position and length of the value that you want to be equal to the extraction variable value.
- In the **Use extraction variable as the embedded key** field, specify the extraction variable number you want to use. When you view the embedded key definition, the extraction variable will be in the format (V-*nnn*), where *nnn* is the extraction variable number. Because you want to use the extraction variable from the first input source, be sure to deselect **Reset variables** in the File Organization dialog box.

Example of Embedded Key Substitution

Input Source 1:

```
1111 abcd
2222 efgh
```

Input Source 2:

```
2222 1234
1111 3456
```

For input source 1, suppose you want to define 1111 as an embedded key and then pick up the value starting on position 6 on that line. For input source 2, you want to find the value of the embedded key defined in input source 1 and pick up the value starting on position 6 on that line.

To do this, in input source 1, you could set up your definitions as follows:

- Define 1111 as an embedded key by setting the start position to 1 and the length to 4.
- Use a field item to pick up the value at position 1 for a length of 4 and assign it to extraction variable 1ST EMB KEY VALUE. Use a field item to pick up the value at position 6 for a length of 4 and assign it to internal item 1, named CODE1.

Your definitions for input source 1 would look like this:

Occurrence 1 of embedded key, value = '1111'

Pick up value at position 1 with length 4, store in extraction variable 1ST EMB KEY VALUE

Pick up value at position 6 with length 4, store in internal item CODE1

In input source 2, you could set up your definitions as follows:

- Define the value of the extraction variable 1ST EMB KEY VALUE as an embedded key. Set the start position to 1 and the length to 4 because those are the positions in the input file you want to match to the extraction variable value. In the **Use extraction variable as the embedded key** field, enter 1ST EMB KEY VALUE, which is the name of the extraction variable you specified in input source 1.
- Use a field item to pick up the value at position 6 for a length of 4 and assign it to internal item 2, named CODE2.

Your definitions for input source 2 would look like this:

Occurrence 1 of embedded key, value = '1ST EMB KEY VALUE'

Pick up value at position 6 with length 4, store in internal item CODE2

Extraction Variables in External Translation Table Definitions

In external translation table definitions, extraction variables are identified by number only. If the extraction variable is extended (see “Specifying Extended Formatting for an Extraction Variable” on page 91), select **Use extended format** in every table definition that references it.

The extraction variables defined when creating an external translation table will be saved as part of table definitions.

For more information on using extraction variables in translation tables, see the following sections:

- “Using Extraction Variables in External Translation Tables” on page 232
- “Combining Data for Translation Using Extraction Variables” on page 232

History Items

What is a History Item?

A history item specifies retrieval from the history database of a value that was stored in a previous run of the current job or of a different job. The retrieved value can be an internal or extended internal item, job run date, job run time, or cycle number. A history item allows the value to be used in calculated items or rules that you define for the current job run. 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

In the History Item, the 'Infogix Assure' option enables you to retrieve history from Infogix Assure..

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 245.

What is Variable Cycle Processing?

Variable cycle processing is an optional feature you can specify when you define a history item.

When you use this feature, the retrieval of the history item will be based on the run date and time and/or the cycle ID of another job stored in the history database.

Variable cycle processing allows you to set up a stopping point in history to proactively account for variations in your processing schedule. For example, you may have a daily job that balances to a weekly job. The weekly job would attempt to bring in the results from five daily jobs. On a holiday week, you may only have three daily jobs. Variable cycle processing allows you to tell the system to only bring in the cycles that have run since the last weekly job.

For a comprehensive explanation of the feature, see “Understanding Variable Cycle Processing” on page 211.

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.

Setting Up Variable Cycle Processing

To specify variable cycle processing, in the History Item dialog box, click **Variable Cycle**. The Variable Cycle dialog box will be displayed.

The dialog box help provides field definitions and other information. For a comprehensive explanation of the feature, see “Understanding Variable Cycle Processing” on page 211.

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.

Calculated items can be conditional or non-conditional. 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. For more on conditional calculated items, see below.

The values of calculated items are not stored in history. If you want to store the value of a calculated item in the history database, you must assign it to an internal (or extended internal) item.

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

5 ■ Setting Up Job Definitions

Calculated Items

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 245.

Conditional Calculated Items

A calculated item can be conditional. To make it conditional, set the Conditional Calculated Item field to Y 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.
- 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 Calculated item to process field on the Calculated Items panel was set to Y 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.

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.

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 106.
- The Formula tab, which defines the computation.

The dialog box help provides field definitions and other information.

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. (If you need multiple sets of rules, you can have up to 10 sets of 100 rules each. See “What are Alternate Balancing Rules?” on page 100.) If one or more rules are out of balance, the entire job is considered out of balance.

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.

5 ■ Setting Up Job Definitions

Rules

Two types of rules can be created in ACR/Summary.

- “Standard Rules” on page 100
- “Conditional Rules” on page 102

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 245.

What are Alternate Balancing Rules?

Alternate balancing rules are additional sets of balancing rules that can be defined for a single job. A rule is made part of a rule set by assigning a 1-digit rule set identifier (1-9) when the rule is created. Alternate balancing rule sets are useful when the same job is being run by different departments or for different processing periods.

Up to 10 different rule sets can be defined for a job. Each rule set can contain up to 100 rules. These may be either standard or conditional rules. Rules 1-100 are in rule set 0, rules 101-200 in rule set 1, and so forth.

When it is time to run balancing, you enter the number of the rule set you want to use in the **Control Field** on the Balancing dialog box.

For example, if you want all of your daily balancing to use one set of rules, you would create the rules and store them all in rule set 0. You could then create another set of rules needed to run the monthly job and store those rules in rule set 1. When you run the job for daily processing, you would enter 0 in the **Control Field** on the Balancing dialog box. When you run the job for monthly processing, you would enter 1 in this field.

Standard Rules

What is a Standard Rule?

A standard rule compares values from any combination 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.

Active Standard Rules

An active standard rule defines criteria for a comparison (using items, literals, and constants as described above) that is evaluated when balancing 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 4-digit code of your choice as the completion code for the job if this rule is out-of-balance. If more than one rule is out-of-balance, the highest code will be set.
- Trigger an abend of the balancing 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

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 balancing 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 balancing 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

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. 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.

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 two tables on the following page 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.

5 ■ Setting Up Job Definitions

Rules

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.

If True Table:

If True	Then	Description
R1	Process R3	If “rule 1 is in balance” is true, then process rule 3.
¬R1	Process R3	If “rule 1 is not in balance” is true, then process rule 3.
R1 + R2	Process R3	If “rule 1 is in balance and rule 2 is in balance” is true, then process rule 3.
R1 + ¬R2	Process R3	If “rule 1 is in balance and rule 2 is not in balance” is true, then process rule 3.
R1 R2	Process R3	If “rule 1 is in balance or rule 2 is in balance” is true, then process rule 3.
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 False Table:

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.
¬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.

If False	Then	Description	What Will Happen
R1 + ¬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 ¬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 ¬R1 | ¬R2 is false” statement above:

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.

What happens if you use AND and OR connectors in the same If statement?

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/Summary 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.

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 “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.

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 balancing 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 balancing 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 rules. 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.

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.

5 ■ Setting Up Job Definitions

Defining Messages

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,

SYSOUT 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

ACR/Summary has two types of messages: direct and indirect.

For messages that are used frequently across multiple jobs, you can set up a global template that is used by all balancing rules. See “Creating Global Messages” on page 166.

Direct Messages

If you created a rule and defined a new return code for the rule, the Direct 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, or create an indirect message.

Indirect Messages

An indirect message enables you to use messages defined in another job for the current job. Specify a return code to associate with the indirect message, specify the job ID for the job that has the messages you want to use, and specify the first and last lines of the message text to include.

5 ■ Setting Up Job Definitions

Defining Messages

Message Examples

The following table shows sample messages for two jobs and three global messages defined for return code 3000. The table includes the job ID, the message line or sequence number, the return code (RC), the message type, and the message text or the job ID for an indirect message

Job ID	Number	RC	Type	Message Text or Job ID
CLAIMS10 ACRSTEP	001	1001	Direct	DATA ENTRY ERROR IN AMOUNT {I7}
CLAIMS10 ACRSTEP	002	1002	Direct	CLAIMS ADJUSTMENTS NEEDED
CLAIMS10 ACRSTEP	003	2000	Direct	NO INPUT FOR PUERTO RICO
CLAIMS10 ACRSTEP	004	3000	Direct	PROGRAM ERROR
CLAIMS20 ACRSTEP	001	1500	Indirect	CLAIMS10 ACRSTEP 001- 002
CLAIMS20 ACRSTEP	002	3600	Direct	WRONG INPUT FILE USED
GLOBAL RC3000	001	3000	Direct	PROGRAM LOGIC ERROR
GLOBAL RC3000	002	3000	Direct	CONTACT RESPONSIBLE MANAGER
GLOBAL RC3000	003	3000	Direct	LOG PROBLEM IN P6000

Job CLAIMS10

Job CLAIMS10 ACRSTEP include four direct messages, one defined for each of the following return codes: 1001, 1002, 2000 and 3000.

If balancing is run for this job and each of the four return codes is issued, return code 3000 and the following messages will print on the Control Report and any other reports, as specified:

- DATA ENTRY ERROR IN AMOUNT \$1,234,000
- CLAIMS ADJUSTMENTS NEEDED
- NO INPUT FOR PUERTO RICO
- PROGRAM ERROR

The first message, which includes an item value, shows the actual item value.

The global messages for return code 3000 will not be used because this job has a message that uses return code 3000. Global messages are generated only if there is no corresponding direct message in the current job.

Job CLAIMS20

Job CLAIMS20 ACRSTEP includes two messages: an indirect message defined for return code 1500, and a direct message defined for return code 3600. The indirect message points to message numbers 001 and 002 in CLAIMS10.

If balancing is run for this job and generates a return code of 1500, the indirect message will point to message line numbers 001 and 002 of the Job CLAIMS10 definitions. The following messages will appear on the Control Report with a return code of 1500:

- DATA ENTRY ERROR IN AMOUNT {I7}
- CLAIMS ADJUSTMENTS NEEDED

Because the first message is an indirect message, it does not show the actual item value on the report.

Global Messages

Three global messages have been defined. For additional information, see “Creating Global Messages” on page 166.

Alternate Control Tasks

This job definition type is applicable to z/OS only. For more information, see “Alternate Control Tasks” on page 387.

Recap Items

What Are Recap Items?

Each recap item identifies a job to be included in the Recap Report. Recap reports can optionally be used to provide a high-level summary (showing the status and return code) for the identified history records. You can produce a summary of daily processing, variable cycle processing, or a complete system or subsystem.

This report is often run in batch with the last job of the day. If a recap item is entered for each job that runs that day, the Recap Report can be scanned to find any problems that may have occurred.

Regardless of whether you define recap items, the report will run only if you have set the Print Recap Report option in the Basic Information folder to Always print the report.

Setting Up Recap Items

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

Even if recap items are specified, the Recap Report will be generated only if you complete the Print Recap Report dialog box, which can be accessed from the Basic Information section of the Control Task (Job) View.

The dialog box help provides field definitions and other information.

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/Summary Client.

The Batch Options dialog box (**Options > Batch Options**) enables you to do the following:

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 these 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 are 300, 600, 900, for history items, and you create 4 history items. Your generated item numbers would be 300, 600, 900, and 1200. The last item number, 1200, would be larger than the maximum number, 999, allowed for history items, calculated items, and rules. 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."

5 ■ Setting Up Job Definitions

Using Batch Options to Control Item Numbers

You could then 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.

Setting Up Input Sources

This chapter explains how to set up input sources for balancing purposes. When you set up an input source, you typically define the input source file, specify how it will be accessed, and provide definitions that will enable ACR/Summary to locate the pertinent records and extract the required values.

This chapter includes the following sections:

- “What Is an Input Source?” on page 115
- “Preparing to Set Up an Input Source” on page 116
- “General Procedure for Setting Up an Input source” on page 117
- “Selecting an Access Mode” on page 120
- “Understanding Field Formats” on page 136
- “Setting Up an Input Source for the Selected Access Mode” on page 139
- “Setting Up an Input Source for Relational Data” on page 149
- “Date and Cycle Number Extraction and Processing” on page 156
- “Time Extraction and Processing” on page 161
- “Deleting File Definitions” on page 162
- “Extracting Directly from a Delimited Data File” on page 162

What Is an Input Source?

An input source is a file (often a report from your application) from which values will be extracted to be used in a balancing job. The file organization type can be any of the types shown on the File Organization dialog box:

- Physical sequential. A physical sequential file or PDS member.
- DB2 table. Direct access to a DB2 table is applicable only if you are running jobs on z/OS.
- VSAM key sequenced. This is an ISAM file in Windows, UNIX, or Linux mode.
- Direct spool dataset. This file type is applicable only if you are running jobs on z/OS.
- Oracle. Direct access to an Oracle database table (UNIX or Linux mode only).

6 ■ Setting Up Input Sources

Preparing to Set Up an Input Source

- ODBC. Access to a relational database table via your ODBC driver. Available in all configuration modes.
- User Program. 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*.

Multiple input sources can be defined for a single job. When you completed the analysis described in “Determining the Input Source Information” on page 42, you gathered information that will help in setting up each input source.

Preparing to Set Up an Input Source

Gathering Information About Your Input Sources

Before you can set up your input sources, you must have gathered the basic information (such as the path and file name of each input source, the file organization type of each input source, where the records containing the values to be extracted are located in the file, etc.) described in “Determining the Input Source Information” on page 42.

If you cannot use the mark-and-capture method described in the following section, you need to review your file and note the exact positions, lengths, and formats of each field you want to use in a file definition.

Preparing to Use Mark-and Capture

For non-relational database input sources, you can streamline the process of creating file definitions for an input source file by using the mark-and-capture method. This is the process of selecting data from the input file or its COBOL record layout, which is displayed in the Extract Window in Input Source View, with the mouse. In order for the file or record layout to be displayed, it must be in a location that is accessible to ACR/Summary Client. If the file is not currently accessible to ACR/Summary Client, you need to download it in text (ASCII/CR/LF) format.

For more information on mark-and-capture, see step 7 in the “General Procedure for Setting Up an Input source” on page 117.

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.

General Procedure for Setting Up an Input source

Following is the general procedure for setting up an input source. The dialog box help provides additional information.

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 149.

1. Complete the instructions in the previous section, “Preparing to Set Up an Input Source” on page 116. In particular, we recommend that you download input files as needed to use mark-and-capture (described in step 7 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, and 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:

You must complete the **File ID** field by providing a name for this input source.

Complete one of the following fields (unless your input source is a relational database table. In that case see “Setting Up an Input Source for Relational Data” on page 149):

- Complete the **File Name** field if this input source is an ASCII text file accessible to ACR/Summary Client.
- Complete the **Record Layout** field if the COBOL record layout of this input source file is accessible to ACR/Summary 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 ACR/Summary Client, it will display in the Extract Window, enabling you to use the mark-and-capture method to create file definitions

6 ■ Setting Up Input Sources

General Procedure for Setting Up an Input source

You can optionally reuse file definitions that are in the definition database for an existing input source file by entering the 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 (record/field) definitions for the input source file specific to this job.

Note: Another way to reuse existing file definitions, which does enable you to merge new definitions with existing definitions, is through file modeling, which is explained in Chapter 17, “Modeling Job and File Definitions” on page 267.

4. Complete the File Organization dialog box, making special note of the following:
You need to specify an access mode. This is the method that will be used to read the file. See “Selecting an Access Mode” on page 120 for a description of each access mode, so that you can make an appropriate selection.
The dialog box fields that are required for the specified type will display. The dialog box help provides field definitions and other information.
5. Skip this step if you are using access mode 6 or 7.
Complete the Base File Indicator dialog box only if you want to use this set of file definitions as a base model to create definitions for another input source, the model user. For more information, see Chapter 17, “Modeling Job and File Definitions”).
6. Complete the File Comments dialog box if you want to paste (use **CTRL + V** to paste) 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.
7. Complete the Record/Field (file) Definitions section of the view.
Multiple file definitions may be required. File definitions limit the records that will be scanned for extraction and specify the values to be extracted and where the extracted values will be stored.
The access mode will determine the file definition types you can use. The general procedure for creating a file definition is as follows:

Mark-and-Capture Method

- a. If you are using mark-and-capture (see “Preparing to Use Mark-and Capture” on page 116), the input source will be displayed in the Extract Window. Use the mouse to select the first character of the data to be extracted.

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.

Manual Method

If you are not using mark-and-capture, make sure you have the exact positions, lengths, and formats of each field that you wish to select.

Note: For more information about how dates are extracted, processed, and stored, see Date and Cycle Number Extraction and Processing on page 156.

- b. Display the dialog box for the file definition type you want to create using one of the following methods:
 - If you are using mark-and-capture, you can click the icon for the file definition type you want in the toolbar. (You must click on or make mouse selection from the input source displayed in the Extract Window to activate the icons.)
 - Select the file definition type you want from the **Define** menu.
 - Select **New** under Record/Field Definitions and then select the type of definition you want to create from the Extraction Field Type dialog box. This method is not suitable for mark-and-capture.
 - c. Complete the dialog box for the file definition type you are creating. If you are using the mark-and-capture method, the **Key Value**, **Position**, and **Length** fields (if they exist on the dialog box) will be populated for you, but you can change them.
8. Repeat step a through step c to create any additional file definitions needed for this input source.
 9. Click the **Save** icon to save the job with the new input source and its file definitions.
 10. Repeat step 1 through step 9 to create any additional input sources needed for the job.

6 ■ Setting Up Input Sources

Selecting an Access Mode

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

Note: The following sections will provide more specific instructions on setting up file definitions:

- For information to help you select an appropriate access mode, see [Selecting an Access Mode](#) on page 120.
 - For procedures for defining file definitions for access modes 1 through 7, see [Setting Up an Input Source for the Selected Access Mode](#) on page 139.
-

Selecting an Access Mode

For all input source types except relational database tables (these have the file organization/type of DB2 Table, Oracle, or ODBC), you complete the **Access Mode** field on the File Organization dialog box to specify how the input source file will be read.

The specified access mode determines the file definition types that you can use to locate, format, and extract data from the input source.

The following sections:

- Summarize access mode information in a table to help you choose the appropriate mode for your input source.
- Describe each access mode in detail.

After you select an access mode, see “[Setting Up an Input Source for the Selected Access Mode](#)” on page 139.

Brief Description of Each Access Mode

The following table provides a high-level summary of access mode information. For more detail, see “Detailed Description of Each Access Mode” on page 122.

Abbreviations:

PS = physical sequential (ASCII text files)

VSAM = VSAM key sequenced files

(these are ISAM files in Windows, UNIX, or Linux)

DS = Direct spool data sets

Access Mode	Available for these File Types	Description
1 - First X Records ¹	PS, VSAM, DS	Selects the first X records (X is the number of records to be selected as specified in the Max. Record Count field in the File Organization dialog box) from the input source and places them in the file access area.
2 - Last X Records ¹	PS, DS	Selects the last X records (X is the number of records to be selected as specified in the Max. Record Count field in the File Organization dialog box) from the input source and places them in the file access area.
3 - X records based on file key(s) ¹	PS, VSAM, DS	Selects records based on file key(s) and places them in the file access area. Record selection stops when one of the following occurs 1) the maximum specified in the Max. Record Count field in the File Organization dialog box has been reached or 2) the stop key is found.
4 - X records from the nth record ¹	PS, DS	Selects the nth record and x number of records after it up to the maximum (as specified in the Max. Record Count field in the File Organization dialog box) and places them in the file access area.
5 - All records (total/tally/hash)	PS, DS	Processes all records in the input source and derives a total, tally (count), and/or hash value of one of the following: <ul style="list-style-type: none"> ■ All records in the file (if no key value is specified). ■ Records that match a specified key value. ■ Records that do not match a specified key value. For total records, the field to be totaled is specified in the Field Information section of the Total Record dialog box. A hash value of all records in a file can be useful for comparing 2 files to determine whether they are identical.

6 ■ Setting Up Input Sources

Selecting an Access Mode

Access Mode	Available for these File Types	Description
6 - Select records that match 1 line	PS, VSAM	Processes all records in the input source using selection groups. All criteria from all selection groups must be met on the same record.
7 - Select records that match multiple lines	PS, VSAM	Same as access mode 6 except that, although all criteria from all selection groups must be met, they need not be met on the same record.
1 - See "File Access Area" in the next section for an explanation of this term.		

Detailed Description of Each Access Mode

Note: When you select a **File Organization/Type** in the File Organization dialog box, the access modes available for that type display for selection in the **Access Mode** field.

Access Mode 1—Selects the First X Records

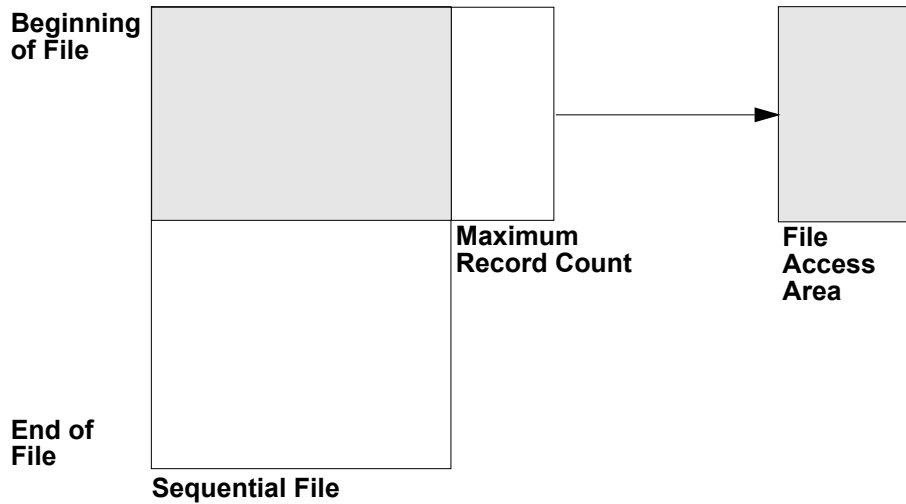
This access mode selects the first 1 to 99,999 records from the input source and places them in the file access area.

File Access Area

The file access area used in access modes 1 through 4 is a temporary storage place for the application records that will be scanned for extractable data. Using a file access area eliminates the need to scan all of the records in the input file.

The number of records to be selected is specified in the **Max. Record Count** field in the File Organization dialog box.

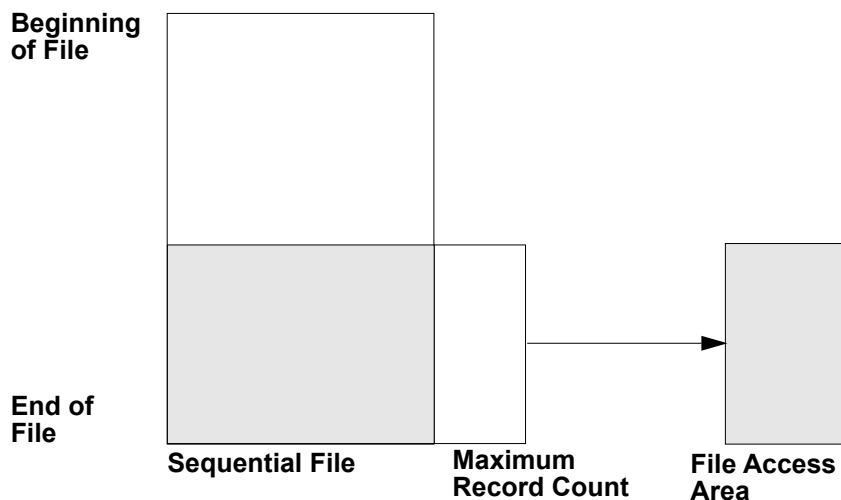
This mode is appropriate for input sources where the values to be extracted reside at the top of the file. The use of a file access area makes processing quicker because only part of the file is searched.



Access Mode 2—Last X Records

This access mode selects the last 1 to 99,999 records from the input source and places them in the file access area. (For an explanation of this term see “File Access Area” on page 122.) The number of records to be selected is specified in the **Max. Record Count** field in the File Organization dialog box

This mode is appropriate for input sources where the values to be extracted reside at the bottom of the file. The use of a file access area makes processing quicker because only part of the file is searched.



6 ■ Setting Up Input Sources

Selecting an Access Mode

Access Mode 3—X records based on file key

Access Mode 3 selects records based on one or multiple file keys and places them in the file access area. (For an explanation of this term see “File Access Area” on page 122.)

- Use one file key when you want to specify a starting point for selecting a block of records.
- Use multiple file keys where the values to be extracted reside on records that contain a common character string.

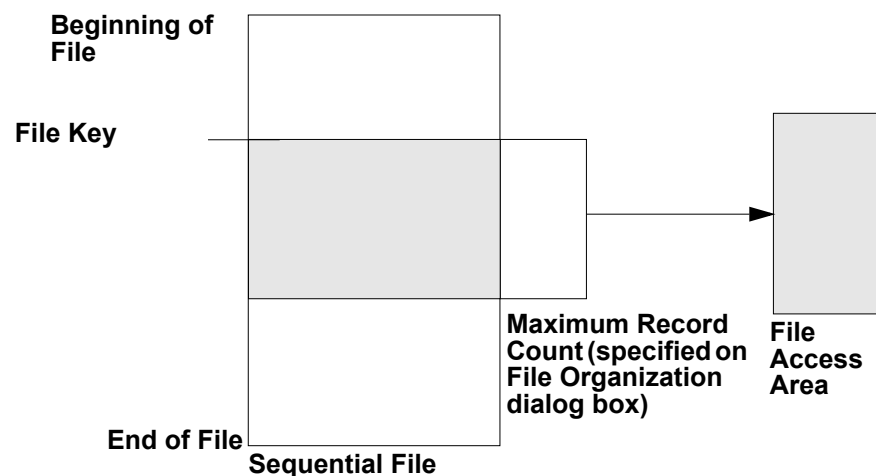
Record selection stops when one of the following occurs 1) the maximum specified in the **Max. Record Count** field in the File Organization dialog box has been reached or 2) the stop key (explained below) is found as explained below.

What is a File Key?

A file key allows you to specify a character string (the key value) to be searched for in the input file. To be considered a match, the record must contain the file key value beginning at the specified start position and having the specified length. For example, a file key might specify the key value DIVISION TOTALS SECTION starting in column 20 for a length of 23. Only a record containing this value at this location in the file is considered a match.

Processing differs depending on whether one file key or multiple file keys are specified, as follows:

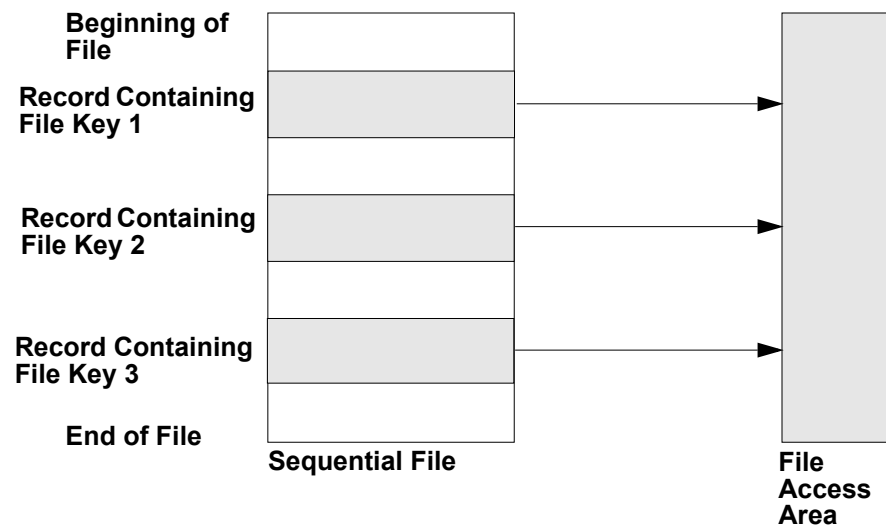
One File Key



If only one file key is specified, the application file is read sequentially from the beginning until a record is found that contains the key value string beginning at the specified position. If found, that record and a block of

subsequent records are selected are copied to the file access area. (For an explanation of this term, see “File Access Area” on page 122.) The block of records to be copied ends when either the **Max Record Count** as specified on the File Organization dialog box is met or the end of file is reached.

Multiple file keys



If multiple file keys are specified, each is assigned a file key item number. The file key with the lowest number is called the start file key, and the file key with the highest number is called the stop file key.

The file is read sequentially from the beginning until a record matching the start file key is found. If found, that record is copied to the file access area (if no match is found, no records are selected from the file). After a record matching the start key is found, each subsequent record is read and compared to all of the file keys. If a record matches any of the file keys (except the stop key), the record is selected and copied to the file access area.

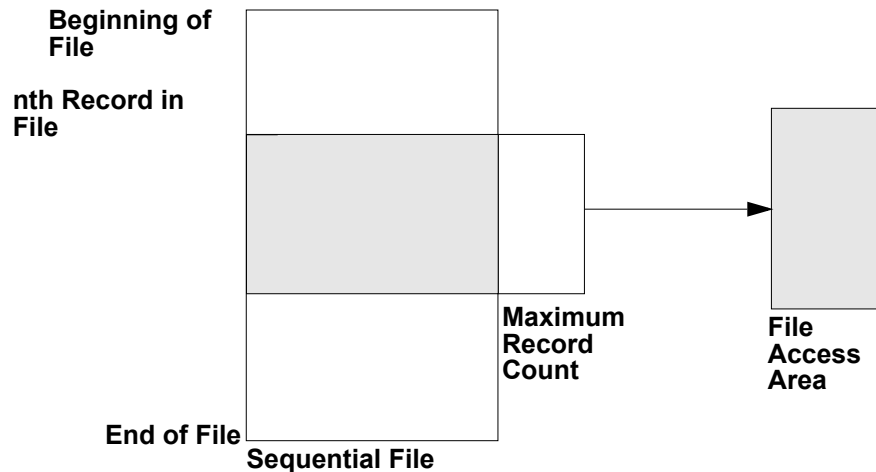
Selection stops when 1) the stop file key is found, 2) Max Record Count specified on the File Organization dialog box is met, or 3) the end of file is reached.

6 ■ Setting Up Input Sources

Selecting an Access Mode

Access Mode 4—X records from the nth record

This access mode selects the nth record and x number of records after it and places them in the file access area. (For an explanation of this term, see “File Access Area” on page 122.) For example, if you want to pick up records 10 through 250, you would tell the system to pick up 241 records starting with record number 10.



Access Mode 5—All records (total/tally/hash)

This access mode processes all records in the input source and derives a total, tally, and/or hash value of one of the following:

- All records in the file (if no key value is specified).
- Records that match a specified key value.
- Records that do not match a specified key value.

For total records, the field to be totaled is specified in the Field Information section of the Total Record dialog box.

Hash Record Purposes and Example

Hashing can be helpful for determining whether two files are identical.

For example, you could define hashing of all records of an input source in two jobs, job A and job B. The hash value from job A needs to be stored in the history database.

In Job B you could use a rule to compare the hash value it obtains with the hash value from job A. If the hash values were the same, you would know that the input files hashed in the two jobs were identical.

If ACR/Connector is used, hash values from jobs that reside on different platforms can be compared. The files being hashed must be in the same format, for example, both ASCII or both EBCDIC.

Access Mode 6—Select records that match 1 line

This access mode typically evaluates every record in the input source using selection groups, 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, and Detail Fields” on page 129) to obtain data for use in balancing rules.

When to Use Access Mode 6

Access modes 6 and 7 both use selection groups. It will be easier to set up your definitions using access mode 6 (rather than access mode 7) when the following conditions exist:

- You do not need to extract values from the same location on multiple records (lines).
- The records containing the data you want are not always in the same sequence as the file definitions in the selection group.

Selection Fields

This information applies to both access mode 6 and access mode 7.

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, which consists of specific positions of the current record or 2) data from specific positions in a saved extraction variable.
- You can choose from 15 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.

6 ■ Setting Up Input Sources

Selecting an Access Mode

- Multiple selection fields can be combined for evaluation with other file definition types that are valid for access mode 6 and 7 into a selection group.

Selection Groups

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

A selection group consists of the following file definition types:

- One selection field or a set of consecutive selection fields at the **group level**. Group-level selection fields are not indented in the display. Consecutive selection fields are joined by AND or OR

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.

- Optionally, 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.

- For an explanation of the other file definition types available in access modes 6 and 7, see “Relative Records, Reformat Records, and Detail Fields” on page 129.

Demote or Promote 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, which work as follows:

- 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, if 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.

The logical operator that is currently being used is displayed to the right of the first selection field in the group or subgroup.

Toggle 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.

Relative Records, Reformat Records, and Detail Fields

One or more of the following record types can be defined at the group level, the subgroup level (in access mode 6 only), or both. These records will be processed if the record passes the preceding group or subgroup evaluation.

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.

A **reformat record** can be used to reformat and/or move data from the currently selected record or from a literal. (The reformatting/moving is done in memory only and does not change the input source or literal). For more information, see “Understanding Reformat Fields - Access Modes 6 and 7” on page 145.

6 ■ Setting Up Input Sources

Selecting an Access Mode

One or more **detail fields**. This field type does one of the following:

- Extracts data from specified positions of 1) the currently selected record or 2) an extraction variable.
- Tallies (counts) the selected records (no values are extracted).

The extracted value can be assigned to an internal item, extended internal item, or extraction variable and can be stored as one of the following:

- A total of the extracted values.
- The date-count value for the last date extracted.
- The last text value extracted.
- A tally of the selected records.

Access Mode 6 Example: Simple Selection Group with No Subgroups

Definitions:

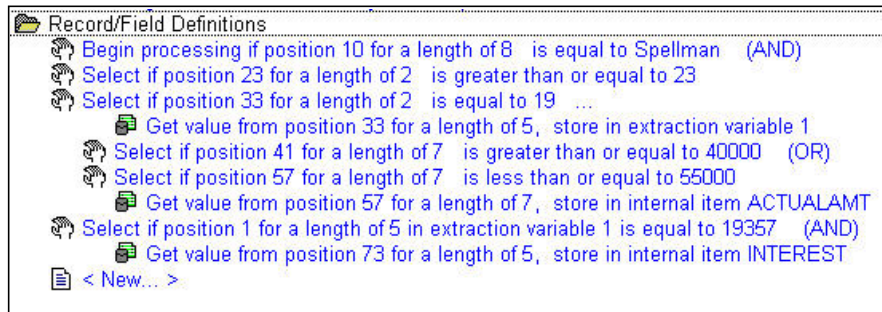
```
Select if position 33 for a length of 2 is equal to 19 (AND  
Get value from position 57 for a length of 7, store in internal item ACTUALAMT
```

This is an example of a simple selection group with one selection field. Using the equal selection type, it evaluates the specified positions of the input area against one constant. If the record passes, the other field in the group, a detail field (the line beginning Get value) is processed, and the data in the indicated positions of the detail field is extracted and assigned to an internal or extended internal item. (Here, the value is assigned to an internal item.)

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

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

Access Mode 6 Example: Two Selection Groups, One with a Subgroup
 This example shows two selection groups at the group level, the first of which contains one selection group at the subgroup level.



Processing the First Selection Group

The first selection group starts with three consecutive selection fields at the group level joined by AND. Notice the following:

- The hand icon (not the word Select) indicates a selection field. Here the first selection field starts with the word Begin.
- The third selection field at the group level has three dots at the end, indicating that the value in the specified positions is being compared to one or more values in addition to 19. You can click on the **Add** button to open the dialog box and enter or view the additional comparison values or ranges.
- Because AND is the logical operator at the group level, the current record must meet the criteria of all three of the group-level selection fields in order for the detail field (the line beginning Get value from position 33) to be processed.
 If the record passes, 1) the resulting value will be stored in extraction variable 1, and 2) the record will be passed on to the subgroup (indented) for evaluation. If the record fails, the subgroup will be skipped.
- The subgroup begins with two consecutive indented selection fields joined by OR. If the record satisfies the criteria of either of these fields, the record will pass and the detail field (the line beginning Get value from position 57) defined at the subgroup level will be processed.

6 ■ Setting Up Input Sources

Selecting an Access Mode

Processing the Second Selection Group

Selection groups are evaluated independently. The record will be passed on for testing by the second selection group regardless of whether it passes or fails the first.

The second selection group consists of one selection field (the line beginning `Select` if position 1 for a length of 5) that evaluates the specified positions in extraction variable 1 using the equal selection type against a constant. If these values are equal, the record will pass, and the detail field (the line beginning `Get` value from position 73) defined for this selection group will be processed.

Processing the Next Record

After all file definitions have been processed against the first record, the next record will be read and evaluated against the same definitions, starting with the first selection statement (the line beginning `Begin processing if`).

Access Mode 7—Select records that match multiple lines

Like access mode 6, access mode 7 evaluates records in the input source using selection groups, but the processing is different, and access mode 7 lacks some of flexible features of access mode 6.

When to Use Access Mode 7

Access mode 7 will be a better choice than access mode 6 when both of the following conditions exist:

- You want to extract values from the same location on multiple records (lines).
- The records containing the data you want are always in the same sequence as the file definitions in the selection group.

Similarities 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, relative records, reformat records, and detail fields.

Differences Between Access Modes 6 and 7

Access modes 6 and 7 differ in some important ways:

- Access mode 7 does not support OR selection and selection subgroups (nested selection groups).

- In access mode 6, every record in the input source is typically evaluated against all selection groups, starting with at the top of the definitions, before the next record is read. This is not always true in access mode 7, as detailed in the next sections.

Access Mode 7 Processing

In access mode 7, the following occurs:

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.
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 Processing Example

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

6 ■ Setting Up Input Sources

Selecting an Access Mode

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.
*All steps are described in Access Mode 7 Processing on page 133.					

Access Mode 7 Example

Assume that your input file is the one shown below. You want to extract the balance for DANVERS that follows the ADAMS record. The balance you want for DANVERS is always after the first record for ADAMS.

LAST NAME	BALANCE
SMITH	123.4999
JONES	45.8770
SPELLMAN	1.4235
ADAMS	229.3030
DANVERS	103.0001

In access mode 7, you could retrieve this value (103.0001) with very few record/field definitions as shown below:

Each record would be read and tested against the first selection group (the line beginning Select if position 1 for a length of 5) only until the ADAMS record passes this group. Then the following records would be read and tested beginning with the second selection group (the line beginning Select if position 1 for a length of 7).

To do this in access mode 6 would require more definitions. You would have to use an extraction variable, as shown below.

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:

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	000000012341111
(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
```

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/Summary, 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.

6 ■ Setting Up Input Sources

Understanding Field Formats

Formats for Windows-Based Column Types

The following table shows how Windows-based column types are formatted. 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/Summary 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
Date	Numeric = Packed
Time	Numeric = Packed
Timestamp	Numeric = Packed

Setting Up an Input Source for the Selected Access Mode

After you have selected an access mode based on the information in “Selecting an Access Mode” on page 120, you can use one of the procedures provided in this section to set up an input source.

This section assumes you will be using the mark-and-capture method and that the file itself is displayed in the Extract Window at the bottom of the Input Source View.

If you are using a record layout, you will not be able to select specific fields with the mouse. Instead, click on the line of the value you want to identify.

If you are not using mark-and-capture, you will need to know the exact positions, lengths, and formats of each field that you wish to use in a file definition so that you can enter these in the dialog box.

Setting Up an Input Source Using Access Mode 1, 2, 3, or 4

Access modes 1 through 4 are similar in that they all pull records from the input source and place the data in a temporary storage place called a file access area. Only the records in the file access area are scanned for data extraction. Using a file access area makes processing more efficient by reducing the number of records that need to be scanned for extractable data.

Review the “General Procedure for Setting Up an Input source” on page 117. That section provides general procedures that apply regardless of the access mode you are using. Then complete the following procedure to set up an input source for access mode 1, 2, 3, or 4:

1. If you are using UNIX or Linux mode and you want to use mark-and-capture to select data from your input sources, you must first download the input files to a location accessible to ACR/Summary Client.
2. From Control Task (Job) View, select **Input Sources > New** to display the Input Source View.
3. In the Basic Information section of the view, complete the Name dialog box using the instructions in the dialog box help.
4. Complete the File Organization dialog box.

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.

6 ■ Setting Up Input Sources

Setting Up an Input Source for the Selected Access Mode

5. If you are not going to use this file as a base model for file modeling, skip the Base File Indicator dialog box. File modeling is explained in “Modeling File Definitions” on page 272.

You can now create your file definitions in the Record/Field Definitions folder, as explained in the following steps.

6. If you are not using access mode 3, skip this step.
If you are using access mode 3, you need to define one or more file keys definitions, which will be used to populate the file access area.

To create a file key definition, select the file key value with the mouse, then click the **File Key** icon and complete the dialog box. Create one or multiple file keys, depending on the following:

- Use one file key when you want to specify a starting point for selecting a block of records.
- Use multiple file keys where the values to be extracted reside on records that contain a common character string.

7. If needed, create one or more of the following file definition types to limit the records included in the file access area.

To create a definition, select the key value (for begin or end reference records only), then click the icon for any of the following file definition types and complete the dialog box:

- Begin reference record. Marks the beginning of a subset of records (called a reference area) within the file access area that will be scanned for extractable data.
- End reference record. Marks the end of a subset of records (called a reference area) within the file access area to be scanned for extractable data.

Note: When using begin/end reference records, you must define the end reference record immediately after the begin reference record with no intervening file (record/field) definitions. For example, in Input Source View, your begin and end reference record definitions might look like this:

```
Set begin reference record with key value 'A'  
Set end reference record with key value 'B'  
Occurrence 1 of embedded key, value = '1A'
```

- **Reset record.** Resets the beginning and/or ending point of the reference area. This would be useful, for example, if you were trying to capture information about different departments in your organization but you didn't know the order in which they appear. Using a reset record, you could have the system return to the top of the file access area, reset the reference point(s), and search again for information on another department.
- **Relative record.** Finds a record relative to the last record found. This is used when the record you want to find does not have a unique character string that can be used to identify it.
- **Page/line record.** Specifies the page and line number where a record you want to scan for extraction is located. When a page/line record is used, you usually will not need to specify begin/end reference points because a page/line record identifies a specific record. A page/line record can be used only if the following are true:
 - The records are in fixed block addressing (FBA) format based on a report page number and line number.
 - The record appears on the same page and line in every report used with the definitions you are creating.
- **Embedded key record.** Embedded keys are used to identify records that contain data that you want to extract from the selected file access area.

Note: Embedded keys are the method to tell ACR/Summary what fields should be extracted. Embedded Keys should always be specified to pick up data. If no embedded keys are specified the results may be inconsistent between platforms and options.

You can use a value from one of the following as the value of your embedded key:

- **A literal.** For example, if you want to extract a dollar amount from a record that contains the word TOTAL, you could identify TOTAL as the embedded key and then use a field item to identify the dollar amount to be extracted.
 - **An extraction variable.** Defining the value of an extraction variable as the value of an embedded key enables you to use the technique explained in "Setting Up Embedded Key Substitution" on page 93.
8. Define a field item to extract the value required for balancing and to specify where the extracted value will be stored. The value can be stored in one of the following:

6 ■ Setting Up Input Sources

Setting Up an Input Source for the Selected Access Mode

- An internal or extended internal item. This can be used directly in a rule.
- An extraction variable. Extraction variables are temporary storage areas used to store extracted values for further processing. For more information on extraction variables, see “Extraction Variables” on page 91.

To create the field item definition, select the key value, then click the **Field Item** icon and complete the dialog box.

Note: If the **Field Type** is Date, the date will be extracted as a date count value based on the date format you select.

To understand how date count values are calculated when extracting a date with a 2-digit year, see *Calculation of Century Values* on page 160.

If the **Field Type** is Time, the time will be extracted as a time count value based on the time format you select.

9. Repeat **step 6** through **step 8** for each additional field from which values should be extracted from this input source.
10. Click the **Save** icon to save the job with the new input source and its file definitions.

Setting Up an Input Source Using Access Mode 5 (Total/Tally/Hash)

This access mode processes all records in the input source and derives a total, tally, and/or hash value of one of the following:

- All records in the file (if no key value is specified).
- Records that match a specified key value.
- Records that do not match a specified key value.

Note: Hashing of all records in a file can be used to determine whether 2 files are identical. For more information on how hash records are used, see *Hash Record Purposes and Example* on page 126.

Before you start, review the “General Procedure for Setting Up an Input source” on page 117. That section provides general procedures that apply regardless of the access mode you are using. Then complete the following procedure to set up an input source for access mode 5:

1. If you are using UNIX or Linux mode and you want to use mark-and-capture to select data from your input sources, you must first download the input files to a location accessible to ACR/Summary Client.
2. Display the Input Source View. From Control Task (Job) View, select **Input Sources > New**.
3. In the Basic Information section of the view, complete the Name dialog box using the instructions in the dialog box help.
4. Complete the File Organization dialog box.
5. Complete the Base File Indicator dialog box only if you are using file modeling. You can now create your file definitions in the Record/Field Definitions folder.
6. Create one or more of the following file definition types as needed:
To create a total, tally, or hash record, select the key value (if you are using one), then click the **Total Record**, **Tally Record**, or **Hash Record** icon and complete the dialog box.

When you define any of the record types, you can specify that the extracted value will be stored in one of the following:

- An internal or extended internal item. This can be used directly in a rule.
- An extraction variable. Extraction variables are temporary storage areas used to store extracted values for further processing. For more information on extraction variables, see on extraction variables, see “Extraction Variables” on page 91.

Setting Up an Input Source Using Access Mode Modes 6 and 7

Before you begin, review the “General Procedure for Setting Up an Input source” on page 117. That section provides general procedures that apply regardless of the access mode you are using. Then review “Access Mode 6—Select records that match 1 line” on page 127 and “Access Mode 7—Select

6 ■ Setting Up Input Sources

Setting Up an Input Source for the Selected Access Mode

records that match multiple lines” on page 132. The way selection groups work depends on whether you are in access mode 6 or 7. Then complete the following procedure:

1. If you are using UNIX or Linux mode and you want to use mark-and-capture to select data from your input sources, you must first download the input files to a location accessible to ACR/Summary Client.
2. Display the Input Source View. To do this, from Control Task (Job) View, select **Input Sources > New**.
3. In the Basic Information section of the view, complete the Name dialog box using the instructions in the dialog box help.
4. Complete the File Organization dialog box.
5. Optionally create a relative (advance) record. 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.
6. Create the field types required for each selection group. Following is a typical procedure:
 - a. Create a selection field.

Select the value in the Extract Window (if you are selecting from the input area using mark-and-capture) click the **Selection** icon and complete the dialog box. Some of the fields will be completed for you.

Otherwise click the **Selection** icon and complete the dialog box. If you are not using mark-and-capture, you will have to specify the positions you want to capture manually.

If you are selecting from an extraction variable, you will need to specify the extraction variable number.

If the records containing the value you want to extract contain no unique key value, see step b.
 - b. If you are selecting from the input area, and the current record does not contain a unique key value, you need to create a relative record after the selection field. 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, click the **Relative Record** icon and complete the dialog box.

- c. Repeat [step a](#) again as needed to define any additional 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 and nested selection groups are supported, as explained in “[Access Mode 6—Select records that match 1 line](#)” on page 127.
- d. If you need to change the positions and/or format of a field before picking it up using a detail record, create a reformat record. Click the **Reformat** icon and complete the dialog box. For more information on reformat fields, see “[Understanding Reformat Fields - Access Modes 6 and 7](#)” on page 145.
- e. Create one or more detail fields. A detail field indicates the values to extract from the selected input records or extraction variable, how the extracted values will be processed for storage, and whether the result will be stored in an internal item, extended internal item, or extraction variable.

To create a detail field, select the value in the Extract Window (if you are using mark and capture to extract from an input source), then click the **Detail** icon and complete the dialog box.
- f. Repeat [step a](#) through [step e](#) to define any additional selection groups.
- g. Click the **Save** icon to save the job with the new input source and its file definitions.

Understanding Reformat Fields - Access Modes 6 and 7

Reformat fields are used in access modes 6 and 7 to change the positions and/or formats of fields from the input source or from a specified literal.

Reformat fields are often used to combine multiple fields from an input source to create one continuous string of characters. The reformatted string can then be used as a selection or picked up as a detail field.

When reformat field definitions are used, a copy of the input record is made in memory and the reformatting is done to this copy. The data in the input source or literal is not reformatted.

You can then reformat the copied record with data and/or with literals supplied in the reformat field definitions. Any combinations of data and literals can be reformatted.

For example, suppose your input source contains a social security number you want to pick up. In the input source, the number is formatted with hyphens as follows:

111-22-3333

6 ■ Setting Up Input Sources

Setting Up an Input Source for the Selected Access Mode

You want to pick it up without the hyphens. This would require 2 reformat records:

- One record to move the 22 to the left 1 position.
- One record to move the 3333 to the left 2 positions.

If you do this, the result will be 111223333**33** (in memory) because the last 2 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.

The Reformat Field dialog box consists of the following sections:

- The **Input Field information** section specifies the position, length, and format of the field in the input source.
- The **Output Field information** section specifies the position, length, and format of the reformatted field.

A reformat field can be defined before or after a selection statement, as follows:

Reformat Before Selection

Reformat fields defined **before** selection (before a selection group) stand alone and are not part of the group. These reformat fields will be applied to each and every input record before any other definitions are processed. The subsequent selection or detail field definitions will refer not to the original input data position, but to the reformatted positions and data.

Reformat After Selection

Reformat fields can also be defined **after** selection (within a selection group and/or between the selection fields). These reformat fields are part of the selection group and are only processed against input records satisfying the selection group's criteria. If the reformat field is processed, the value will be changed for all subsequent selection groups.

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 08/01/29 from text to numeric, you will get 08001029.

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 the reformat 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.

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.

6 ■ Setting Up Input Sources

Setting Up an Input Source for the Selected Access Mode

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---|--- 70---|---
123456789|123456789|123456789|123456789|123456789|123456789|123456789|123456789|123
\\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 data source 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.

6 ■ Setting Up Input Sources

Setting Up an Input Source for Relational Data

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**, the system builds the query dynamically based on your extraction definitions. See “Using the RDBMS Extraction Method” on page 153.

Note: Extraction definitions for relational database tables cannot be ported across platforms.

Using Freeform SQL Statements

The freeform SQL statements method enables you to use tested, valid SQL queries in ACR/Summary 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/Summary 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/Summary, 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/Summary 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.
 - Freeform SQL From Clause dialog box: The statement must begin with FROM.
 - 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.

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. (If you select DB2 Table, and you want the SQL to display in SYSOUT at run-time, select **Display SQL in SYSOUT**.) Select the **Use Freeform SQL Statements** field.

If **File Organization/Type** = ODBC, the data source 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** field (do not use the browse button).

4. Optionally complete the File Comments dialog box.
5. In the Record/Field Definitions folder, click the **Freeform SQL Select** icon. Paste in the SELECT clause from your valid SQL query.
6. Click the **Freeform SQL From** icon. Paste in the FROM clause from your query.

7. Click the **Freeform SQL Where** icon. 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 the extraction variable from the list in the lower section of the dialog box, 2) Select **Use Extended Format** and/or **Use Numeric Type**, as appropriate, and 3) select **Add Extraction Variable to Where Clause**. For information on extraction variable codes, see “Extraction Variable Codes Using Freeform SQL Statements” on page 155.
8. Create one or more field items to extract or accumulate column values from the selected rows and store them in an internal item, extended internal item, or extraction variable.
9. If you want to define additional selection criteria and pick up additional values, repeat steps 5 through 8. Repeating these steps also enables you to do a table join. (Each of these defines a SQL WHERE clause. Multiple RDBMS records in a row (followed by one or more field items) are ANDed.)
10. Click the **Save** icon to save the job with the new input source and its file definitions.

Using the RDBMS Extraction Method

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.

1. Display the Input Source View. From Control Task (Job) View, select **Input Sources > New** to display the Input Source View.
2. In the Basic Information section of the view, complete the Name dialog box.
3. Complete the File Organization dialog box. (If you select DB2 Table, and 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.

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)

6 ■ Setting Up Input Sources

Setting Up an Input Source for Relational Data

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. Optionally complete the File Comments dialog box.
5. Click in the Record/Field Definitions folder, click the **RDBMS Record** icon, and complete the information for the primary relational database table that contains the data to be extracted.
6. Click the **RDBMS Record** icon again and complete the data for each additional table from which you want to extract data for balancing. The **Browse** button is available only if the File/Organization type specified is ODBC.
7. Complete the Record/Field Definitions section of the view.
 - a. Click the **RDBMS Record** icon and complete the RDBMS Record dialog box. The RDBMS record defines a SQL WHERE clause, which provides criteria to select rows to retrieve from the table. Each RDBMS record, with its corresponding field items (created in the step below), makes up a selection group.

Note: You can include an extraction variable in the **Column Name** or **Argument Value** field, or both. To include an extraction variable, enter the codes described in Extraction Variable Codes for the RDBMS Extraction Method on page 156.

 - b. Create one or more field items to extract or accumulate column values from the selected rows and store them in an internal item, extended internal item, or extraction variable.
 - c. If you want to define multiple selection groups, repeat step a and step b. Multiple selection groups are concatenated into a query.
8. Click the **Save** icon to save the job with the new input source and its file definitions.

Extraction Variable Codes for Relational Database Extraction

This section provides guidelines and valid formats for extraction variable codes that can be used in Where statements.

It contains the following sections:

- "Extraction Variable Codes Using Freeform SQL Statements" in the next section.
- "Extraction Variable Codes for the RDBMS Extraction Method" on page 156.

Extraction Variable Codes Using Freeform SQL Statements

Entering the Codes

If you are using freeform SQL statements, extraction variables can be entered in the Freeform SQL Where clause dialog box by selecting from a list. The codes can appear on either or both sides of the comparison. For example:

```
Field name1 or extraction variable = field name2 or extraction variable
```

Format for a Regular Extraction Variable Using Freeform SQL Statements

: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 using Freeform SQL Statements

: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
```

6 ■ 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.

Extraction Variable Codes for the RDBMS Extraction Method

If you are using the RDBMS extraction method, extraction variables can be entered manually in the RDBMS Record dialog box.

Format for a Regular Extraction Variable using the RDBMS Extraction Method

V- nnn

where nnn is the number of the extraction variable (for example, 001).

V- 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,

V-001 \wedge V-002 = 'upto16characters'
space space

Format for an Extended Extraction Variable using the RDBMS Extraction Method

XV- nnn

where nnn is the number of the extraction variable (for example, 001).

Two text type extended extraction variables can be combined to make a text value up to 160 characters long. For example:

XV-001 \wedge XV-002 = 'upto160characters'
space space

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

Note: Extraction variables can also be used in a selection fields, as explained in Extraction Variable Uses in File Definitions on page 92.

Specifying the Position, Length, and Format of the Date to Extract

Position

When using the partial cycle number format, the **Position** field should show the first position of the portion of the date you want to extract. For example, if you want to extract the century portion of the date 01/15/2008, the **Position** field should show the first position of the century.

For more information on the partial cycle number format, see “Partial Cycle Number Format” on page 158.

Length

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:

01/15/2008 is a length of 10.

Jan. 01, 2008 is a length of 13.

Length for Partial Date (to Store as the Cycle Number)

When specifying a partial date the **Length** field should be the length of the portion you want to extract (2 or 3).

For more information on the partial date format, see “Partial Cycle Number Format” on page 158.

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 Format Options

Partial Cycle and As Is Formats

Following is an explanation of the partial date and as is options. These date formats are available only when extracting dates for use in assigning the cycle number.

6 ■ Setting Up Input Sources

Date and Cycle Number Extraction and Processing

Partial Cycle Number Format

The partial cycle number format is available on the Field Item dialog box when **Field Type** is set to Cycle Number. It enables you to extract a 2-digit century (cc), 2-digit year (yy), 2-digit month number (mm), 2-digit day number (dd), or 3-character month (mmm for JAN, FEB, and so on).

Conversion of partial dates to cycle numbers:

The extracted characters will be converted to ccyyymmdd numeric format, with the portions of the date not selected populated with zeros.

For example, if the date in the input file is 01/12/2008:

- And you use the partial cycle number option to extract the century, the resulting cycle number will be 20000000.
- And you use the partial cycle number option to extract the month, the resulting cycle number will be 00001200. You might use this option if you do not want the century value to be automatically populated with 20 for cycle numbers whose year value is less than or equal to 80.

For information on specifying position and length when using the partial cycle number format, see “Position” on page 157 and “Length” on page 157.

Accept date As Is

This format is available in the Field Item dialog box when the **Field Type** is Cycle Number. The as is date format indicates that, in general, the extracted value will be used as the cycle number with no changes. There are some exceptions, as noted below.

- If you extract fewer than 8 digits, the system will fill the leftmost positions with zeros.
- If you specify a value for extraction that includes a 3-character month, the digits extracted will not be converted, but the 3 characters representing the month will be converted to 2-digit format. For example, if the value specified is 01.JAN.2005, the cycle number will be 01012005.

Date Formats That Use DD, MM, CC, YY, JJJ, and/or MMM

Note: The remaining 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.

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.

Note: The associated batch transaction value is not shown in the options for the **Date Format** field in the Field Item dialog box.

yy/mm/dd (1)

mm/dd/yy (2)

mm/dd/ccyy (3)

dd/mm/yy (4)

yy/jjj (5)

dd/mm/ccyy (6)

cyy/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).

6 ■ 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.

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) as described in “Calculation of Century Values” on page 160, 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) as described in “Calculation of Century Values” on page 160, extracted dates that are used as cycle numbers are stored internally in ccyyymmdd format.

Display and Printing of Dates and Times

The default format for display and printing of dates is yy/mm/dd.

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 specify Y for the **Print century before year** field on the Date Formats tab.

The default format for display and printing of times is hh:mm:ss.

You can control the following defaults through your user options as described in “Setting User Options” on page 165:

- Order of the date elements.
- Character to use as the date separator (instead of the default forward slash).
- 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.

Note: 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 the internal and/or extended internal items. If a date count value is more than 145731 (December 31, 1999), the century value was assumed to be 20.

- Character to use as the time separator (instead of the default colon).

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 TTTTTTTTTTTT' format where -99999999 is the relative day and TTTTTTTTTTTT is the formatted time.

When the print format is 'Date & Time', time element is printed as 'DDDDDDDD TTTTTTTTTTTT' where DDDDDDDD is the formatted date and TTTTTTTTTTTT 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:

- 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 326.

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 record/field 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 326.
- 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.

6 ■ 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 `uniuf.dat`, enable you to override a variety of system defaults. User options include the following categories: miscellaneous, currency formats, date/time formats, negative signs, XML options, and Infogix Assure information.

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

This chapter includes the following sections:

- “The User Options File (`uniuf.dat`)” on page 165
- “Setting User Options” on page 165
- “Using Global Messages” on page 166
- “Setting User Options for UNIX or Linux” on page 168

The User Options File (`uniuf.dat`)

User options are stored in the user options file (`uniuf.dat`). This file:

- Is accessed for every balancing 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.
- On UNIX, Linux, or Windows, is shared by both ACR/Summary and ACR/Detail.

Setting User Options

Setting User Options in Windows Mode

If you set user option in Windows mode, the user options will be stored in the user options file (`uniuf.dat`) on the Windows server. These options will apply to all ACR/Summary jobs run by all users on the Windows server.

7 ■ Setting the User Options

Using Global Messages

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:
 - Use user option file specified by server.
 - Use local user option file (will always be copied to server)
 - 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 “Setting User Options for UNIX or Linux” on page 168.

Using Global Messages

A global message is a message that will print in the Control Report when any job issues its associated return code, providing no message has been defined for that return code at the job level.

Creating Global Messages

To define global messages, do the following:

1. Define a global messages job name. Use any unique, valid job name. You can define the name in one of the following ways:
 - Complete the **Global Messages Job Name** field on the User Options dialog box - Miscellaneous tab.
 - If you are using a UNIX or Linux configuration or running jobs in batch on UNIX or Linux, specify a global messages job name in the UNIXOPT record of the User Options File, uniuf.dat. For more information, see “Execution Options Record - UNIXOPT Record Layout” on page 168.

2. Create a separate global messages job for each return code for which global messages should be generated. Each global messages job must follow these guidelines:
 - The job name should be the global messages job name specified in step 1.
 - The step name should be *RCnnnn*, where *nnnn* is the return code for which you want to specify one or more global messages.
 - The qualifier field must be left blank.

For example, if MYGLOBAL is the global messages job name, and the return code for which you want global messages to be generated is 3000, the job ID would be MYGLOBALRC3000.
 - A report title must be entered.
 - Define the messages. In Control Task (Job) View, click **New** under the Messages folder, select Direct Message, and create the first message to be associated with the return code indicated in the step name. Up to 100 messages may be created.

Note: In a global messages job, the system does not process the entry in the Direct Message dialog box's **Return Code** field. All messages you create will be associated with the return code indicated in the job's step name, regardless of your entry in the **Return Code** field.

If you want to create global messages for multiple return codes, you must create multiple global messages jobs.

3. Save each global messages job and update the definition database.

Processing of Global Messages

When you run any job that generates a return code for which there is no message defined at the job level, the following processing will take place:

1. The system searches the user options file for the global messages job name.
2. One of the following will take place:
 - If no global messages job name has been defined in the user options file, the message NONE will be printed on the Control Report for the return code.
 - If a global messages job name is found, the system will continue to the next step.

7 ■ Setting the User Options

Setting User Options for UNIX or Linux

3. The definition database will be searched for a job that has both the global messages job name and a step name in the format *RCnnnn*, where *nnnn* is the return code generated.

Setting User Options for UNIX or Linux

Overview

If you want to customize the user options for running jobs from ACR/Summary Client in UNIX or Linux mode, you need to edit the user options (UNIUF.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. The records on *uniuf.dat* that you may want to customize are UNIXOPT and UNIEXEO.

Note: ACR/Summary and ACR/Detail share a single *uniuf.dat* file on the UNIX or Linux host. Some options that are valid on ACR/Summary are not valid on ACR/Detail, however. For more information, see the *ACR/Detail User Guide for UNIX or Linux*. Modifications you make on the UNIX or Linux host to *uniuf.dat* options that are valid on both products will apply to all scripts and to all users of both products, if you have both.

See the following sections for explanations of each available option:

- “Execution Options Record - UNIXOPT Record Layout” on page 168
- “Bypass Processing Option - Global Level - UNIEXEO” on page 182

Execution Options Record - 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.

Following is the record layout. For more information, see “UNIXOPT - Detail on Each Option” on page 172.

Note: ACR/Summary and ACR/Detail share a single user options file. Information about options not applicable to this product is included in the footnotes.

Setting the User Options ■ 7

Setting User Options for UNIX or Linux

Field	Description	Position	Length	Format	Value
1	Type	1-10	10	X(10)	UNIXOPT
2	Dual History Option	11	1	X(1)	Y, N, blank
3 ¹	Automatic Rerun Option	12	1	X(1)	Y, N, blank
4 ²	Currency Information Provided	13	1	X(1)	Y, N, blank = N
5	Thousands Place Marker	14	1	X(1)	
6	Decimal Place Marker	15	1	X(1)	
7	Currency Symbol	16-21	6	X(6)	
8	Second Currency Symbol	22-27	6	X(6)	
9	Currency Padding Length	28	1	9(1)	0-8
10	Currency Symbol Location	29	1	9(1)	2,4,6 (front, back, both)
11	Currency Symbol Length	30	1	9(1)	1-6
12	Negative Sign	31-36	6	X(6)	
13	Second Negative Sign	37-42	6	X(6)	
14	Negative Sign Padding Length	43	1	9(1)	0-8
15	Negative Sign Location	44	1	9(1)	2,4,6 (front, back, both)
16	Negative Sign Length	45	1	9(1)	1-6
17 ³	Date Information Provided	46	1	X(1)	Y, N, blank=N
18	Date Output Format	47-49	3	X(3)	YMD, MDY, etc.
19	Date Separator	50	1	X(1)	
20	Date Zero Suppress Indicator	51	1	X(1)	Y, N
21	Century Indicator	52	1	X(1)	Y, N
22	WTO Message Option	53	1	X(1)	Y, N, blank

7 ■ Setting the User Options

Setting User Options for UNIX or Linux

Field	Description	Position	Length	Format	Value
23	History Inserts	54	1	X(1)	Y, N, blank
24	Print Commas for Counts	55	1	X(1)	Y, N, blank=Y
25	Print Commas for Amounts	56	1	X(1)	Y, N, blank=Y
26	22-Character Numeric Formatting	57	1	X(1)	Y, N, blank=N
27	Reserved (used by ACR/Detail only)	58	1	X(1)	Blanks
28 ⁴	Control Report XML Option	59	1	X(1)	1 or blank, 2, 3, 4, 5
29 ⁵	User Report XML Option	60	1	X(1)	1 or blank, 2, 3
30 ⁶	XML DB Fail Option	61	1	X(1)	1 or blank, 2, 3
31 ⁷	XML DB Message Size (in 10,000s)	62-65	4	X(4)	Blank, 0000 through 1677
32 ⁸	Free-Form Report XML Option	66	1	X(1)	1 or blank, 2, 3
33	Time Separator	67-67	1	X(1)	
34 ¹¹	Time Output Format	68-68	1	X(1)	1, 2, 3 or Blank
35 ¹²	Time Output Notation	69-69	1	X(1)	1, 2 or Blank
36 ⁹	Job Name for Global Messages	70-77	8	X(8)	
37 ¹⁰	Rerun History Audit trail Indicator	78	1	X(1)	Y, N, blank

Field	Description	Position	Length	Format	Value
38	Send Alternate Job ID	79-79	1	X(1)	Y, N, Blank
39	Reserved	80-80	1	X(1)	Blank

1 Automatic Rerun Option:

Y = Use automatic rerun, system-wide, except if job level is set to 2 (Invoke rerun only if system level is not set to do so).

N = Do not use automatic rerun, system-wide, except if job level is set to 2 (Invoke rerun only if system level is not set to do so).

Blank = Use the balancing-step-level specification.

2 Currency Information Provided: Enter Y if you plan to specify any currency options in fields 5–11.

3 Date Information Provided: Enter Y if you plan to specify any date options in fields 18–21.

4 Control Report XML Option

1 or blank - Do not generate the report in XML format, 2 - Generate, write copy 1 to file, 3 - Generate, autoload copy 1 to database, 4 - Generate, write copy 2 to file, 5 - Generate, autoload copy 2 to database

5 User Report XML Option:

1 or blank - Do not generate the report in XML format, 2 - Generate, write to file, 3 - Generate, autoload to database

6 XML DB Fail Option:

1 or blank - display warning and continue, 2 - Display warning message, write message to file and continue, 3 - Display error and stop

7 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

8 Free-Form Report XML Option:

1 or blank - Do not generate the report in XML format, 2 - Generate, write to file,

3 - Generate, autoload to database

9 Job Name for Global Messages: If you set up an ACR/Summary job containing global messages, enter the job's eight-character job name in this field. Leave the step name and qualifier blank.

10 Rerun History Audit trail Indicator:

Y =Use rerun history audit trail, system-wide, N =Do not use rerun history audit trail, system-wide

Blank = Use the job-definition-level specification.

11 Time Output Formats: 1- HH:MM 2 or blank - HH:MM:SS 3 - HH:MM:SS.NN

12 Time Output Notations: 1- 12-hour format (AM or PM will be printed) 2 or blank – 24-hour format

UNIXOPT - Detail on Each Option

Detail on the UNIXOPT options can be found on the pages indicated below:

“Dual History Option” on page 172

“Automatic Rerun Option” on page 173

“Currency Format Options” on page 173

“Negative Sign Format Options” on page 175

“Date/Time Format Options” on page 176

“Write to Operator Messages Option” on page 178

“History Insert Option” on page 178

“Print Commas for Counts” on page 178

“Print Commas for Amounts” on page 178

“Use 22-character Numeric” on page 178

“XML Options” on page 180

“Global Messages Option” on page 181

“Rerun History Audit Trail Option” on page 181

Dual History Option

What Is the Dual History Option?

When ACR/Summary stores history records, they are written to the primary history database. The dual history option is a 1-character field that allows the histories to be written to a secondary history database in addition to the primary history database. The secondary history database is updated whenever the primary history database is updated. Valid values for the option are `Y` (Yes) to store history to a secondary database, `N` (No) to store history only to the primary database, or `Blank` to store history only to the primary database [i.e. use the default of `N` (No)].

Enabling the Dual History Option

To use the dual history option, do the following:

1. Turn on the dual history option by specifying `Y` (Yes) on the UNIXOPT record.

2. Use either the `uacinit.sh` (Initialize Databases) script or the ACR/Summary Client user interface to initialize a secondary History database.

Note: It is very important to remember that when a database is initialized it is erased. When using this function, be careful to specify a database name that differs from your primary history database name.

See Appendix D, "Using ACR/Summary for UNIX or Linux in Batch" on page 347 for instructions on running the script.

3. Use either the `uachfc.sh` (Copy History) script or the ACR/Summary Client Copy History feature to copy the primary History database to the secondary History database. Do this whenever you wish to synchronize the two databases.

See Appendix D, "Using ACR/Summary for UNIX or Linux in Batch" on page 347 for instructions on running the script.

4. Give the file name of the secondary history database name to each ACR step. You can do this by setting the following environment variable:

```
dd_UNIHF2 = <path and filename>; export dd_UNIHF2
```

Automatic Rerun Option

This option is not applicable on UNIX or Linux. Leave a space in the `uniuf.dat` file to indicate that the system should skip this 1-character field.

Currency Format Options

The currency format options allow you to customize the currency information appearing in your ACR/Summary 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.

7 ■ Setting the User Options

Setting User Options for UNIX or Linux

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 English usage 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-
```

Decimal Place Marker

This is a 1-character symbol to denote the decimal place. In English usage, 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
```

7 ■ Setting the User Options

Setting User Options for UNIX or Linux

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.

Date/Time Format Options

The date/time format options allow you to customize the date and time information appearing in your ACR/Summary reports. This affects the headings on your reports as well as any items defined as date and time formats and not overridden at a lower level.

The following list shows the order of precedence used by ACR/Summary for the date/time format options in effect. Precedence is shown from high to low. Notice that the user options file overrides any of the other options shown.

1. Options File
2. File Interface Script Parameter
3. Extracting from the Application Interface
4. Automatic Cycle Processing
5. Cycle Reference

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 08/10/31 for October 31, 2008.

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 (/). In this example, a hyphen is used to separate the month, day, and year; as in 01-01-06.

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 08/01/01 (yy/mm/dd) on your Control Report. By changing the century indicator to Y, you can make the date appear as 2008/01/01 (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 (:). In this example, a colon is used to separate the hour, minutes, and seconds; as in 10:09:45.

7 ■ Setting the User Options

Setting User Options for UNIX or Linux

Write to Operator Messages Option

The user option to allow you to write messages to the console is not available in a UNIX or Linux environment.

History Insert Option

This option is a 1-character code that allows you to insert history records with a Cycle ID less than those already in the History database.

Valid values are `Y` (Yes) to permit the insertion of history records, `N` (No) to disallow the insertion of history records, or `Blank` to disallow the insertion of history records [i.e. use the default of `N` (No)].

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]).

Use 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).

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.

7 ■ Setting the User Options

Setting User Options for UNIX or Linux

XML Options

Control Report XML Option

The Control Report XML option allows you to specify an option for generating the Control Report in XML format. Valid values are as follows:

- 1 or blank. Do not generate the Control Report in XML format.
2. Generate, write XML to file (Copy 1).
3. Generate, write to XML database (Copy 1).
4. Generate, write XML to file (Copy 2).
5. Generate, write to XML database (Copy 2).

User Report XML Option

The User Report XML option allows you to specify an option for generating the User Report in XML format. Valid values are as follows:

- 1 or blank. Do not generate the User Report in XML format.
2. Generate, write XML to file.
3. Generate, write to XML database.

Free-Form Report XML Option

The Free-Form Report XML option allows you to specify an option for generating the Free-Form Report in XML format. Valid values are as follows:

- 1 or blank. Do not generate the User Report in XML format.
2. Generate, write XML to file.
3. Generate, write to XML 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.

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 16,777,000 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.

Global Messages Option

This option enables you to specify a global messages job name. You need to specify a global messages job name if you want to define global messages. For more information, see “Using Global Messages” on page 166.

Rerun History Audit Trail Option

This option is not applicable on UNIX or Linux. Leave a space in the uniuf.dat file to indicate that the system should skip this 1-character field.

7 ■ Setting the User Options

Setting User Options for UNIX or Linux

Bypass Processing Option - Global Level - UNIEXEO

The Bypass Processing Option flags Infogix products to stop running and overrides return code settings globally by automatically setting a return code of 0000. To directly specify global processing options in the user options file, you need to create a UNIEXEO record that conforms to the record layout shown below

FIELD	DESCRIPTION	POS.	LENGTH	FORMAT	VALUE
File Interface					
1	Type of user options card	1	10	X(10)	UNIEXEO
2	Master on/off switch to shut off or turn on ACR/Summary balancing	11	1	X(1)	blank=on (default), 1=on (default), 0=off
3	Override any jobs that normally set a zero completion code	12	1	X(1)	blank or N = no override necessary (default), Y=override to set completion code, X=override to force a user abend
4	Override any jobs that normally set a completion code	13	1	X(1)	blank or Y = no override necessary (default), N=override to set zero completion code, X = override to force a user abend
5	Override any jobs that normally force a z/OS user abend	14	1	X(1)	blank or X = no override necessary (default), N = override to set zero completion code, Y = override to set completion code
6	Reserved	15	66	X(66)	Blanks
Program Interface					
1	Type of user options card	1	10	X(10)	UNIEXEO

FIELD	DESCRIPTION	POS.	LENGTH	FORMAT	VALUE
2	Master on/off switch to shut off or turn on ACR/Summary balancing	11	1	X(1)	blank=on (default), 1=on (default), 0=off
3	Set unconditional zero ACR return code after balancing	12	3	X(3)	NNN
4	Reserved	15	66	X(66)	Blanks

The UNIEXEO option allows global overrides of return code processing in file interface mode. These overrides include the following:

1. Override all jobs that normally set a return code and abend to set a return code only or set a zero return code.
2. Override all jobs that normally set a return code only to set a return code and abend or set a zero return code.
3. Override all jobs that normally set a zero return code to set a return code only or to set a return code and abend.

See the *Messages and Codes* manual for help with the generated #U messages.

To eliminate the override warning message in file interface mode, conform to the record layout as shown. Warning messages are not issued if the override was specified in the job.

If invalid UNIEXEO options are specified, #U messages will be generated. See the *Messages and Codes* manual for help with these messages.

- #USY045E: INVALID MASTER PROCESS ON/OFF SWITCH
- #USY046E: INVALID SET ZERO RETURN CODE OVERRIDE OPTION
- #USY047E: INVALID SET RETURN CODE OVERRIDE OPTION
- #USY048E: INVALID ABEND RETURN CODE OVERRIDE OPTION

Examples of when you would use File Interface Global Override follow.

7 ■ Setting the User Options

Setting User Options for UNIX or Linux

Example 1

Situation: You want to make sure all ACR/Summary jobs set a zero return code regardless of what appears in the Set Return Code field or Rule Action Code field.

Solution: Set field #3 in the UNIEXEO card to “blank” or “N”. Set field #4 to “N”. Set field #5 to “N”.

Result: Any job with a Set Return Code = N will be unaffected and no warning messages will be issued. Any job with a Set Return Code = Y or Rule Action Code = S will be overridden to set a zero return code and warning message #USY043W will be issued. Any job with a Set Return Code = X or a Rule Action Code = A will be overridden to set a zero return code and warning message #USY042W will be issued.

Example 2

Situation: You have decided that all jobs with a Set Return Code = Y or a Rule Action Code = S should be overridden to set a return code and abend while all jobs with a Set Return Code = N should be overridden to set a return code.

Solution: Set field #3 in the UNIEXEO card to “Y”. Set field #4 to “X”. Set field #5 to “blank” or “X”.

Result: All jobs with a Set Return Code = N will be overridden to set a return code and warning message #USY044W will be issued. All jobs with a Set Return Code = Y or a Rule Action Code = S will be overridden to set a return code and abend and warning message #USY042W will be issued. All jobs with a Set Return Code = X or a Rule Action Code = A will be unaffected and no warning messages will be issued.

Infogix Assure Information Record Layouts

The Infogix Assure option enables you to retrieve history from Infogix Assure. Infogix Assure provides a web service for the history retrieval.

CAURL1, CAURL2, CAURL3 and CAURL4 Record Layouts

To directly specify the Infogix Assure server URL in the user options file, you need to have CAURL1-CAURL4 (280 characters) records that conform to the record layouts shown below.

CAURL1 Record Layout

Field#	Description	Pos.	Length	Format	Value
1	Type	1-10	10	X(10)	CAURL1.
2	URL	11-80	70	X(70)	Position 1-70 of the Infogix Assure URL.

CAURL2 Record Layout

Field#	Description	Pos.	Length	Format	Value
1	Type	1-10	10	X(10)	CAURL2.
2	URL	11-80	70	X(70)	Position 71-140 of the Infogix Assure URL.

CAURL3 Record Layout

Field#	Description	Pos.	Length	Format	Value
1	Type	1-10	10	X(10)	CAURL3.
2	URL	11-80	70	X(70)	Position 141-210 of the Infogix Assure server URL.

CAURL4 Record Layout

Field#	Description	Pos.	Length	Format	Value
1	Type	1-10	10	X(10)	CAURL4.
2	URL	11-80	70	X(70)	Position 211 - 280 of the Infogix Assure server URL.

7 ■ Setting the User Options

Setting User Options for UNIX or Linux

CAUSERID Record Layout

To directly specify the Infogix Assure server user ID (for Infogix Assure history retrieval) in the user options file, you need to have a CAUSERID record that conforms to the record layout shown below.

Field#	Description	Pos.	Length	Format	Value
1	Type	1-10	10	X(10)	CAUSERID
2	User ID	11-80	70	X(70)	Infogix Assure server User ID

CAPASSWD Record Layout

To directly specify the Infogix Assure server password (for Infogix Assure history retrieval) in the user options file, you need to have a CAPASSWD record that conforms to the record layout shown below.

Field#	Description	Pos.	Length	Format	Value
1	Type	1-10	10	X(10)	CAUSERID
2	Password	11-80	70	X(70)	Infogix Assure server User password.

CATMZONE Record Layout

To directly specify and store the Infogix Assure server time zone to retrieve history in the user options file, you need to have a CATMZONE record that conforms to the record layout shown below.

Field#	Description	Pos.	Length	Format	Value
1	Type	1-10	10	X(10)	CATMZONE
2	Time Zone	11-70	60	X(60)	Time Zone

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 187
- “Creating a Free-Form Report” on page 187
- “Viewing the Internal Code Representations for Objects” on page 193
- “Report Sample” on page 193
- “Deleting a Free-Form Report” on page 194
- “Printing a Free-Form Report” on page 194

What is a Free-Form Report?

The Free-Form Report is a customizable report on a balancing job run and its results. 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, (company name from the Control File), job ID, cycle number, return codes, messages, report title, internal items, extended internal items, history items, calculated items, rules, item descriptions, and free-form text. You determine where to place the information on the report. If specified, this report can be generated by running balancing in file interface mode.

The file name for the report is UNIUFR.RPT.

Creating a Free-Form Report

Note: If you use job modeling to model the definitions of a free-form report, you cannot merge the model job's free-form report definitions with the current job's. If you create a free-form report in your current job, that report will override the report in the model job. This will happen even if you set your current job's free-form report print options to Do not print the report. For more information on job modeling, see Chapter 17, “Modeling Job and File Definitions” on page 267.

Procedure

Create the report as follows:

1. Optionally see “Setting the User Options” on page 165 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 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.
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—You may want to look at the help for the Edit Menu, View Menu, Draw Menu, Object Menu, and Report Menu, all of which 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 190.

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 290.
6. Create the heading and other report components using objects. Start by creating the heading. 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, text string, etc. For more information, see “Working with Report Objects” on page 189.

7. Save your definitions and update the definition database.
8. Run the job to automatically create the report.
9. View the report by selecting the **Open Reports** icon. The output file name is UNIUFR.RPT.

Working with Report Objects

Following is the procedure for working with 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 190.
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.

8 ■ Setting Up Free-Form Reports

Creating a Free-Form Report

- 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.












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 189.

Icons for Free-Form Report View		
Icon	Command	Menu to Access This Command
	Select (Click to select and object. Click and drag to select multiple objects.)	Draw Menu
	Page Break (Adds a page break where you click)	Draw Menu
	Run Date (of the report)	Draw Menu
	Run Time (of the report)	Draw Menu
	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
	Return Code	Draw Menu
	Message	Draw Menu
	String (enter a text string)	Draw Menu

8 ■ Setting Up Free-Form Reports

Creating a Free-Form Report

Icons for Free-Form Report View		
Icon	Command	Menu to Access This Command
	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
	History Item	Draw Menu
	Calculated Item Description	Draw Menu
	Calculated Item	Draw Menu
	Rule Variance	Draw Menu
	Rule Description	Draw Menu
	Page Number	Draw Menu
	Properties	Object Menu (in Free-Form Report View)
	Left Justify	Object Menu (in Free-Form Report View)

Icons for Free-Form Report View		
Icon	Command	Menu to Access This Command
	Center Justify	Object Menu (in Free-Form Report View)
	Right Justify	Object Menu (in Free-Form Report View)
	As is. To understand how this option will affect your report, see step 4 on page 189.	Object Menu (in Free-Form Report View)

Viewing the Internal Code Representations for Objects

The Internal Representation dialog box (**Report > Internal Code>**) displays the internal code representation for each object in the Free-Form Report, along with the positioning for each. Information required for interpreting the codes is provided in the dialog box help.

Report Sample

CYCLE	RUN TIME	JOBNAME/STEPNAME	USER NAME	
20050201	11.34.31	J0BA050/STEP001	INFOGIX, INCORPORATED	
***WEEKLY REPORT GENERATED FRIDAY MORNINGS AT 9:00 A.M.				
Summary Free Form Report				
DESCRIPTION	SALES	RETURNS	CASH ADVANCES	COMMENTS
ACCEPTED	\$279,666.11	\$1,891.14	\$.00	
OVERRIDES	\$.00	\$.00	\$.00	*As Reported by Department
REJECTS	\$.00	\$.00	\$.00	*As provided in File 100R
AMOUNT IN	\$.00	\$.00	\$.00	
AMOUNT OUT	\$.00	\$.00	\$.00	
SUB-TOTAL	\$279,666.11	\$1,891.14	\$.00	
TOTALS ARE VERIFIED IN J0BA100/STEP010 CONTACT THE OPERATIONS DEPARTMENT AT X1890 FOR SCHEDULING CHANGES				

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 balancing 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 195
- “Setting Up a User Report” on page 195
- “Deleting a User Report” on page 197
- “User Report Samples with Field Definitions” on page 197

What is a User Report?

The User Report is a customized report that prints any of the data that can be included in the Control Report in the order that you specify.

If specified, this report will be generated by running balancing in file interface mode.

The file name for the report is UNIUSR.RPT.

Setting Up a User Report

Set up the report as follows:

1. Several system-wide options for report formatting can be set in your user options. See “Setting the User Options” on page 165.
2. In the Report Information section at the bottom of the Control Task (Job) View, double-click Balancing User Report to access the User Report View.
3. Complete the Basic information section:
 - a. Complete the User Report Format dialog box to specify one of the following report formats.
 - Original format - up to 4 columns. If you select this option, any column information that is too wide to fit will be truncated unless you select Wrap Text. Column width will be 18 or 22 characters based on whether you specified 22-character formatting.

9 ■ Setting Up User Reports

Setting Up a User Report

- Enhanced format - up to 8 columns, up to 250 characters wide, and position and length must be entered. If you select this format, you can optionally change the Page Width field to any value between 70 and 250. You can specify the column start positions and widths as long as they fit into the report width you specify. For enhanced format, you can choose fixed spacing and specify the number of spaces to separate each column. Valid values are 0-99.
 - b. Complete the User Report Headings dialog box with the report title and column headings. The available fields will depend on your report format.
 - c. Complete the User Report XML Option dialog box if you want to set a job-level option for generating the User Report in XML format to override the corresponding option in your user options.
4. If you want to define a one-line header containing your choice of information that will appear above the column headings on each page of the report, use the User Report Page Headers dialog box.
 5. Define what will appear in each column on each line using the User Report Detail Lines dialog box. The available fields will depend on your report format. You can also specify an optional description of each report line that will appear in the leftmost column. If your calculated items include totals, your user report can show these totals.

Note: When reserving space for text items in the Length field, add 2 to the number of characters in the text value when calculating the required length, because text values are printed between single quotes.

6. Use the Report Header/Trailer Options dialog box if you want to define up to 2 header lines and 2 trailer lines (in addition to the standard report header identifying the product, release, date, time, type of report, etc.) to appear on each page.
In addition, you can use this dialog box to specify that you want to print up to 100 combined Header/Trailer/Message lines showing out-of-balance messages. These will print at the end of the last page.
7. Save your work and update the definition database before running the job.
8. When you run balancing, you can use the Override Options dialog box if you want to set selected run-time overrides.

Deleting a User Report

To delete the User Report that you have created, 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 Samples with Field Definitions

Standard Format

ACR/S release# DATE: yy/mm/dd TIME: 15:02:09	XYZ COMPANY BALANCING USER REPORT ENERGY REPORT - 12/2005	COPYRIGHT INFOGIX, INC. PAGE 001 DATA BASE VERSION: 0000006/0000005
Energy Usage by Type and Date		
DESCRIPTION	RECEIPTS	DATE
NATURAL GAS	74,814,392,365.85	yy/mm/dd
OIL	1211217599245.34	yy/mm/dd
COAL	\$1,289,336,185.67	yy/mm/dd
IN BALANCE		(RETURN CODE = 0000)

Enhanced Format

ACR/S release# DATE: yy/mm/dd TIME: 15:21:02	XYZ COMPANY BALANCING USER REPORT USER REPORT ENHANCED FORMAT	COPYRIGHT INFOGIX, INC. PAGE 001 DATA BASE VERSION: 0000014/0000012		
ENHUSR /STEP1 / Inventory per Company by Count, Amount, and Date				
DATE	COUNT 1	AMOUNT 1	AMOUNT 2	COMPANY
YY/06/18	65,329	\$842.00	\$847.00	'CROSS MANUFACTURING'
YY/04/15	76,783	\$1,674.00	\$1,674.00	'PARKER EQUIPMENT'
YY/10/14	56,298	\$1,254.00	\$1,254.00	'BIC RESEARCH'
YY/12/03	85,234	\$2,697.00	\$2,697.00	'ROSEART PICTURES'
YY/01/07	42,735	\$392.00	\$362.00	'SANFORD STEEL'
YY/05/05	81,351	\$628.00	\$628.00	'DIXON MASONRY'
YY/02/29	42,333	\$791.00	\$791.00	'FABER MARKETING'
OUT OF BALANCE		(RETURN CODE = 1224)		

9 ■ Setting Up User Reports

User Report Samples with Field Definitions

Report Fields

The fields can be interpreted as follows:

Field	Description
Description	This is an optional description that prints in the far left column only if the report is in the standard format.
Page heading	This is an optional one-line header containing your choice of information that will appear above the column headings on each page of the report.
Column headings	These are headings for up to 4 columns (8 for the enhanced format) specified by the user.
Detail lines	The lines below the heading line can be populated by values from internal items, extended internal items, history items, calculated items, rule return codes, literals, fill characters, or blanks.
Header/trailer lines	Additional information may be defined to appear in custom header and/or trailer lines that print on each report page in addition to the standard report header. Not shown in the example.

Setting Up Cycle Processing

Cycle processing determines the way cycle IDs, which identify individual job runs, will be set and optionally verified or overridden. This chapter explains cycle IDs and describes how to control cycle ID assignment through the cycle processing options. It also explains the cycle processing features of the Options File. This chapter contains the following sections.

- “What is a Cycle ID?” on page 199
- “Reasons for Using Run Numbers Other than 000” on page 200
- “Cycle Processing Overview” on page 201
- “Using Similar Cycle Processing Options for Related Jobs” on page 202
- “Options File (UNIOF.DAT) Features for Cycle Processing” on page 202
- “Order of Precedence for Cycle ID Assignment” on page 206

What is a Cycle ID?

A cycle ID is a means of uniquely identifying each run of a job. The cycle ID consists of the following:

- An 8-digit cycle number. This must be numeric and greater than zero. Often the best choice for a cycle number is the system date in Gregorian or Julian date format. The cycle number can also be a date extracted directly from the input source. However, cycle numbers do not have to be dates. Any ascending number can be used.
- A 3-digit run number. The run number defaults to 000. The default is often sufficient. The next section describes some cases where the non-default run numbers are needed.

Cycle IDs are processed in ascending order. Cycle IDs are especially important for storing, identifying, and retrieving values from history records (containing results of previous job runs) from the history database.

Reasons for Using Run Numbers Other than 000

The cycle ID consists of a cycle number and a run number. The default run number is 000. The most frequent cases in which run numbers other than the 000 are required are as follows:

Case 1

Run numbers are required for a job (we will call it Job A) when both of the following are true:

- A date is used for the cycle number portion of the cycle ID.
- The job (for example, Job A) runs and stores history multiple times on the same day. In this case a unique run number is required for each run so that each cycle ID will be unique. This enables the results of each run to be stored in history. Otherwise subsequent runs of the job on the same date would overwrite previous runs.

Job	Cycle Number	Run Number
Job A	20051127	001
Job A	20051127	002

Case 2

Suppose that in addition to Job A (described above), another job (we will call it Job B) runs on the **same day** as Job A. The following are true of Job B:

- A date is used for the cycle number portion of the cycle ID.
- Job B needs to retrieve a number of cycles of history from Job A. In this case, Job B's cycle ID must be equal to or higher than that of any cycle IDs of Job A that Job B needs to retrieve.

Job	Cycle Number	Run Number
Job A	20051127	001
Job A	20051127	002
Job B	20051127	999

Cycle Processing Overview

Cycle processing is the process of generating or verifying cycle IDs.

When you set up a job, you can specify the option that will be used to assign or verify the cycle ID in the **Cycle Generation** field on the Cycle Processing dialog box. The cycle processing options available in this field can be summarized as follows (see the dialog box help for additional information):

- **None.** Enables you to set the cycle ID (both cycle number and run number) yourself using either of the following methods:
 - Set up your file definitions so that the cycle number will be extracted from an input source.
 - Enter the cycle and run number in the Balancing dialog box when you run balancing.
- **J0, J9, G0, G1, G9.** These options are used to set the cycle number based on the J (Julian) or G (Gregorian) system date. They provide various choices for automatic assignment of the run number.
- **X0, X1, X9.** These options are used when you set up file definitions for extracting the cycle number from an input source and you want the run number to be assigned automatically based on your specification.
- **Verification from cycle table.** This option enables you to have the cycle number extracted from an input source and verified against a table of valid cycle numbers. When you select this option you must also do the following:
 - Set up your file definitions to extract the cycle number from an input source.
 - Set up the a cycle table containing the valid cycle numbers. For instructions, see “Setting Up Cycle Tables” on page 209.
 - Set the **Cycle Table Name** field to the name of the cycle table you created in the previous step.
- **Cycle number set from reference job.** This is called cycle reference processing. The cycle ID (both the cycle number and run number) will be set to be the same as the cycle ID of the most recent run of the job you specify in the **Reference Job ID** field.

Cycle reference processing is particularly useful when you want all jobs in an application system job stream to use the same cycle number. For example, you can ensure that two jobs will always be given the same cycle number regardless of whether they run before or after midnight.

Using Similar Cycle Processing Options for Related Jobs

Related jobs should use similar cycle processing techniques. Examples of how jobs can be related include:

- Jobs that use history items from a previous run of the same job or a different job. History items that are retrieved are identified by both job ID and cycle ID. For more information on history items, see “History Items” on page 96.
- Jobs that use variable cycle processing, where history items are retrieved based on the run date and time and/or the cycle ID of another job. For more information on history items, see “What is Variable Cycle Processing?” on page 96.

Options File (UNIOF.DAT) Features for Cycle Processing

The options file (uniof.dat) provides two options (cycle override and history insert) for setting the cycle ID. The options file is referenced when the job runs. Any other method of setting the cycle ID will be overridden.

You must create the options file manually using the record layout provided. You can run the job through ACR/Summary Client or use a batch file or script. An options file can be used in both file interface mode and direct input mode. For detailed instructions, see “Using the Options File” on page 203.

Cycle Override Option

When the cycle override option is specified, and the corresponding job runs, its cycle ID will be reset to the cycle ID specified in the uniof.dat record. This cycle ID will be passed to the balancing function.

In addition:

- The cycle ID may be assigned to other runs depending on your entries in the Job ID fields.
- The run number may be incremented in subsequent runs based on your entry in the multi-run indicator field.

History Insert Option

Use this option when a job runs out of order and you want to put it back in the right order.

When the history insert option is specified, and the corresponding job runs, its cycle ID will be reset to the cycle ID specified in the uniof.dat record. This cycle ID will be passed to the balancing function. Up to this point, the history insert option works like the cycle override option.

However, unlike the cycle override option, the history insert option allows the history record for this run to be inserted in the history database even if its cycle ID is lower than that of the other cycle IDs in the database. This kind of insert is normally prohibited.

Guidelines for Options File Processing

- Both the cycle number and the run number must be entered in the options file record. You cannot use a CYC# parameter.
- If a duplicate cycle ID is used, an error message indicating that an insert will not take place will be issued. Utilize the run number to insert additional records with the same cycle number.
- If the record that is being inserted has the lowest cycle ID for a particular job ID and the number of histories being stored per job ID has been exceeded, the new record will not be inserted and an error message will be issued.
- If, for the same job run, you include two options files (one specifying a cycle override and one specifying a history insert) or one options file containing two records for the same job ID (one specifying a cycle override and one specifying a history insert), the setting for the first file/record encountered will be used.
- Asterisks can be entered in the Job ID fields as specified in the “Options File (uniof.dat) Record Layout” on page 204.

Using the Options File

To use the options file, do the following:

1. Create the options file (uniof.dat) based on the “Options File (uniof.dat) Record Layout” on page 204. On Windows, UNIX, or Linux, you can put the file in the installation folder/directory for ACR/Summary or in any folder you want.

10 ■ Setting Up Cycle Processing

Options File (UNIOF.DAT) Features for Cycle Processing

2. If you are using the options file for history insert, select **Allow insert of history records** in the User Options dialog box - Miscellaneous tab.
3. Do one of the following:
 - If you are using file interface mode, in the Balancing dialog box, click **Add** to display the File ID and File Name dialog box.
 - In the **File ID** field, enter UNIOF.
 - In the **File Name** field, enter or browse to select the path and file name of the options file. The file name will be uniof.dat.
 - If you are using direct input mode, in the Direct Input Balancing dialog box, click **Add** in the Additional section to display the File ID and File Name dialog box.
 - In the **File ID** field, enter UNIOF.
 - In the **File Name** field, enter or browse to select the path and file name of the options file. The file name will be uniof.dat.
 - If you are using the options file in production on UNIX or Linux, set the dd_UNIOF environment variable in uac2000.sh to point to the options file as in the following example:

```
dd_UNIOF = UNIOF.dat; export dd_UNIOF
```

Where UNIOF.dat is your options file name.
 - If you are using the options file in Windows batch mode, edit your .bat file to include your user options file as in the following example:

```
SET UNIOF=UNIOF.dat
```

Where UNIOF.dat is your options file name.

Options File (uniof.dat) Record Layout

The options file must be created using the following record layout. The file can contain up to 100 records.

FLD#	DESCRIP.	POS.	LENGTH	FORMAT	VALUE
1	Comment Indicator	1	1	X(1)	Blank or * comment)
2	Transaction Code	2-3	2	X(2)	01 - Cycle Override, 02 - History Insert
3	Reserved	4-5	2	X(2)	Blank
4	<u>Job ID</u> ¹	6-23	18	X(18)	Job ID.

Options File (UNIOF.DAT) Features for Cycle Processing

FLD#	DESCRIP.	POS.	LENGTH	FORMAT	VALUE
	Job ¹	6-13	8	X(8)	Enter a specific job name or fill the field with asterisks.
	Step ¹	14-21	8	X(8)	Enter a specific step name or fill the field with asterisks.
	Qualifier ¹	22-23	2	X(2)	Enter a qualifier or fill the field with asterisks.
5	<u>Cycle ID</u>	24-34	11	9(11)	Cycle ID fields (cycle number and run number) must be completed.
	Cycle Number	24-31	8	9(8)	The cycle number to be assigned.
	Run Number	32-34	3	9(3)	The run number to be assigned.
6	Multi-Run Indicator	35	1	X(1)	Y - When cycle override is specified, increments the cycle number after every use in the current job step Blank or N - does not increment the cycle number.
7	Run Date & Time ²	36-47	12	X(12)	These fields apply only when the history insert option is specified.
	Date ²	36-41	6	9(6)	When the history insert option is specified, enter the date in YYMMDD format.
	Time ²	42-47	6	9(6)	When the history insert option is specified, enter the time in HHMMSS format.
	Reserved	48-49	2	X(2)	Blanks
	Spaces	50-79	30	X(30)	Blanks

¹ Job ID fields: If you want the cycle ID specified to be applied to all jobs, steps, and JSQs, fill the job, step, and qualifier fields with asterisks [*] (or fill only the job and step fields with asterisks and leave the qualifier field blank). To restrict the JSQs to be processed, you can fill the job and step fields with asterisks and enter a specific qualifier in the Qualifier field.

²The run date and time entered will be written to the database as the run date and time as if the job actually ran on the date/time specified. The reports will show this date and time also.

Order of Precedence for Cycle ID Assignment

Cycle IDs can be assigned through the options provided in the **Cycle Generation** field on the Cycle Processing dialog box. In addition, Three other methods of setting cycle IDs are available. The order of precedence for these options is as follows:

1. **Options file.** This file is described in “Options File (UNIOF.DAT) Features for Cycle Processing” on page 202. It can exist on the PC, Windows server, z/OS, UNIX, and Linux.
2. **File interface parameter (balancing runtime parameter on Windows or file interface script parameter on UNIX or Linux).** This is a runtime parameter that can be used to specify a cycle number and run number.
3. **UNICNUM and UNICNUM1 user exits (applies to jobs running on z/OS only).** User exits provide a way to modify the COBOL code to perform additional processing. The balancing function invokes the UNICNUM and UNICNUM1 user exit programs after the cycle ID is set. Programmers can modify either of these user exits to perform generation, manipulation, or verification of the cycle ID.
4. **Extracting from the application interface**
The method you use to extract the cycle number depends upon the application interface you are employing. In file interface mode, you use file definitions to extract data from an input source (usually a report or data file). In direct input mode, the cycle number is explicitly entered into the direct input data record 1 transaction.
5. **Automatic cycle processing**
With automatic cycle processing, the system generates all or a portion of the cycle ID for you. Automatic cycle processing will compute a cycle ID for a job run using the date format you choose and (in most cases) incrementing the run number by 1. Run numbers allow you to maintain multiple runs of a balancing step for a given cycle number and are typically used when the same job runs multiple times on the same day. Up to 999 runs can be retained using run numbers.

6. Cycle reference processing

Cycle reference processing is a method of automatic cycle processing where the current job obtains its cycle ID from a specified reference job in the history database. This feature is particularly useful when you want all jobs in an application system job stream to use the same cycle ID even if the job stream spans multiple dates.

10 ■ Setting Up Cycle Processing

Order of Precedence for Cycle ID Assignment

Setting Up Cycle Tables

In ACR/Summary, cycle tables are used for situations requiring cycle number verification, such as when the date is manually entered into a batch file or a system date file or when a job should only run on specific dates.

This chapter provides a description of cycle tables and their use and describes how to set up a cycle table. This chapter includes the following sections:

- “Purposes of Cycle Tables” on page 209
- “Procedure for Setting Up a Cycle Table” on page 210

Purposes of Cycle Tables

Using Cycle Tables in Balancing Jobs

In the balancing function, cycle tables are used to verify cycle numbers. For example, if a company wants a particular job to run only on Mondays, a cycle table with all of the Monday dates of the year could be created. If a job runs with a cycle number that is not in the cycle table, processing will stop and issue an error message indicating that no entry was found in the cycle table.

Cycle tables saved and updated to the definition database before they can be used. Each cycle tables is identified with a table name and can hold up to 1000 entries.

The cycle table name to be used in a balancing job is specified in the Cycle Processing dialog box (**Control Task (Job) View > Basic Information > Cycle Processing**).

11 ■ Setting Up Cycle Tables

Procedure for Setting Up a Cycle Table

For 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 with the name, TUESDAYS. Using a calendar, you would begin entering the Tuesday dates in ascending order as follows using July 4, 2005 as the beginning of the table.

Cycle Dates:

20060704

20060711

20060718

20060725

20060801

This job will only run when it finds a match in the cycle table.

Cycle Tables in History Analysis

In history analysis, you can use a cycle table to verify the data being retrieved from the history database. Cycles are retrieved based on the history analysis definitions. If the corresponding cycle number is not in the cycle table, it will not be used in the history analysis process.

When setting up history analysis definitions, you specify the cycle table name in the Cycle Table Name dialog box (**History Analysis View > Basic Information > Cycle Table Name**).

Procedure for Setting Up a Cycle Table

1. Cycle tables are set up through the Cycle Table View. To display this view, select **File > New > Table (External/Internal/Cycle) > Cycle Table**.
2. From the Cycle Table View, complete the Table Name dialog box.
3. Complete the Cycle Table Entry dialog box.
4. Click the **Save** icon to save the table.
5. Update the definition database with the new table definition.

Understanding Variable Cycle Processing

This chapter explains variable cycle processing. It reviews cycle IDs and relative cycle processing and then explains how variable cycle processing works. For a tutorial, troubleshooting tips, and frequently asked questions on variable cycle processing, see the separate guide *ACR/Summary Variable Cycle Processing*.

This chapter contains the following sections:

- “Review of Cycle IDs” on page 211
- “What Is Relative Cycle Processing?” on page 211
- “What Is Variable Cycle Processing?” on page 213
- “Choosing a Variable Cycle Processing Option” on page 214

Review of Cycle IDs

Before learning about relative or variable cycle processing, you need to understand cycle IDs. Refer to “What is a Cycle ID?” on page 199 for a complete explanation. The examples in this chapter use dates because they are easy to understand.

Cycle IDs can be assigned various ways. In the examples in this chapter, you will assign them as an online option.

What Is Relative Cycle Processing?

Relative cycle processing allows your ACR/Summary job to retrieve history items based on a comparison of the current cycle to previous cycles.

The current cycle is always classified as +000, the previous cycle is -001, the one previous to that is -002, and so forth. You can retrieve up to the -998 cycle.

12 ■ Understanding Variable Cycle Processing

What Is Relative Cycle Processing?

So, to retrieve a history item from a run four cycles in the past, you would choose a relative cycle of -004. For a job running on Friday, the relative cycle for Thursday would be -001 and for Wednesday it would be -002, as shown below.

	Relative Cycle
Daily Job - Monday	-004
Daily Job - Tuesday	-003
Daily Job - Wednesday	-002
Daily Job - Thursday	-001
Daily Job - Friday	+000 (current)

Relative Cycle Processing

The cycle accumulation option allows you to automatically total all history items retrieved.

If you specify cycle accumulation, then the relative cycle number refers to the oldest cycle retrieved. If you do not specify cycle accumulation, then only oldest cycle is retrieved.

Of course, there may be processing situations where the current cycle (+000) is not available and yet you've specified it in your job.

For this situation, you can choose to do one of the following:

- Use the most recent cycle record as the +000 relative cycle. Subsequent relative cycles are then based from this history.

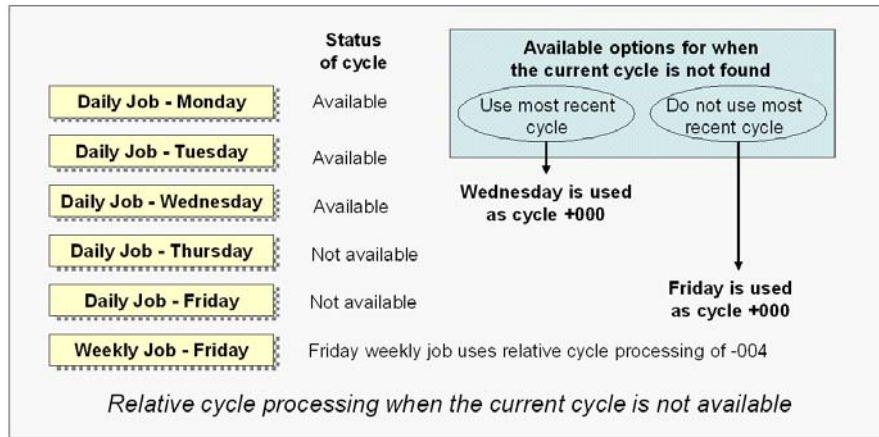
This option is specified by entering Y in the **Should most recent cycle be retrieved when cycle does not match current cycle?** field.

- Do not adjust the cycle numbers.

This option is specified by entering N in the **Should most recent cycle be retrieved when cycle does not match current cycle?** field.

On Windows, UNIX, and Linux, via the graphical interface (ACR/Summary Client or ACR/Workbench for Summary), these options are available on the History Item dialog box via the **Retrieve most recent cycle** and **If item is not found** fields.

The following example shows how this option works in a situation where some jobs are unavailable. In this case, the Thursday and Friday jobs are not available.



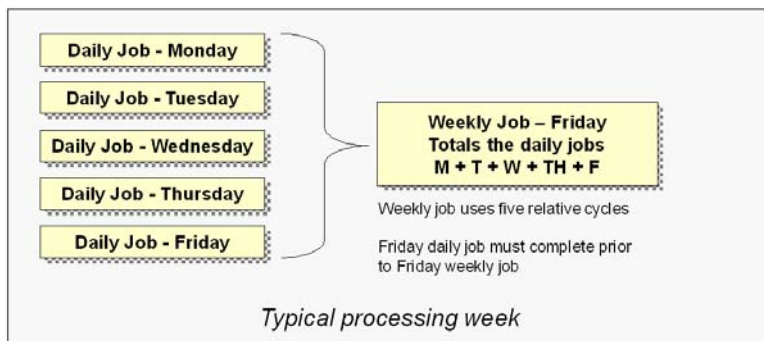
Notice the situation for the Do not use most recent cycle option. The Friday values are not available, but the ACR/Summary job must still use it. In this situation, the ACR/Summary job will use zeroes as the values for that cycle.

What Is Variable Cycle Processing?

Variable cycle processing allows you to control the retrieval of a history item based on the run date and time or the cycle ID of another job or both. This is unlike relative cycle processing, which relies exclusively upon cycle IDs.

The purpose of variable cycle processing is to accommodate situations where the run date and time of other jobs that provide the history items are different than usual.

For example, you might have a weekly job that performs balancing that uses the history items from daily jobs:



12 ■ Understanding Variable Cycle Processing

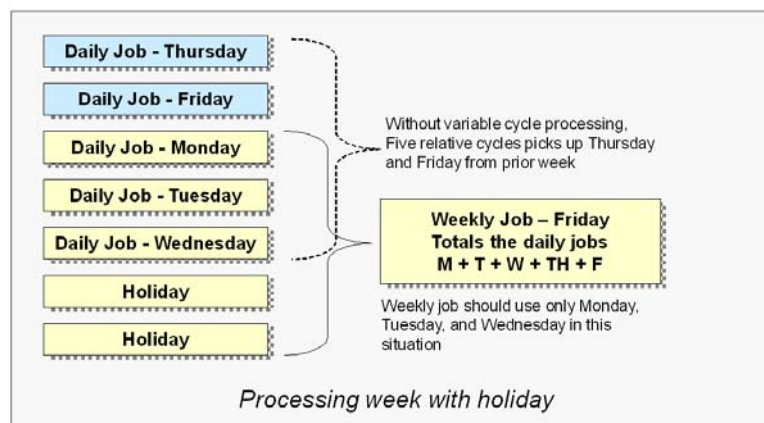
Choosing a Variable Cycle Processing Option

In this situation, the cycle processing indicates five relative cycles to use for the weekly job.

However, the daily job might not run every day. For example, during Thanksgiving week in the U.S., the daily processing for Thursday and Friday might not take place.

If your ACR/Summary controls are for the financial industry, any holiday might mean a cycle with no processing, and therefore no historical data to use in a subsequent balancing job.

In our example, if the “five relative cycles” were used for input to the weekly job, the job would use input data from Thursday and Friday of the prior week.



By using variable cycle processing, you could specify that only the daily job cycles that have run since the last time the weekly job ran will be picked up by the weekly job and used in balancing.

Variable cycle processing relies upon the cycle IDs or run dates of retrieved history items. By making comparisons, ACR/Summary can correctly choose to use or not use the history item.

ACR/Summary offers selections that match your processing environment. These options are available in the **Variable Cycle** panel, as shown below. Each option is discussed in detail in “Choosing a Variable Cycle Processing Option” on page 214.

Choosing a Variable Cycle Processing Option

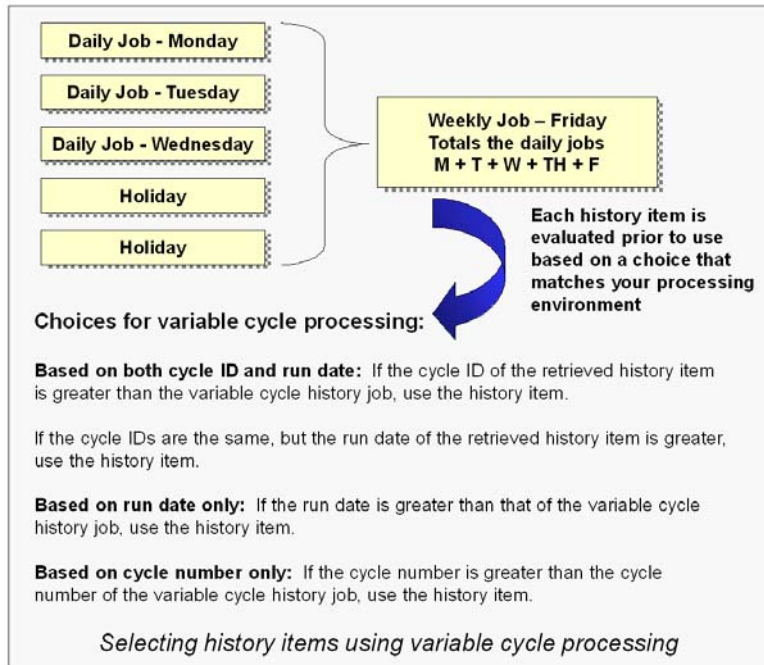
When setting up variable cycle processing, you will need to make choices that reflect your processing environment. You have four choices:

- No variable cycle processing (option 1, the default)
- Compared to cycle ID and most recent run date and time (option 2)

- Compared to most recent run date and time (option 3)
- Compared to most recent cycle number (option 4)

By carefully choosing the appropriate option for your site, you can be sure that the history item that ACR/Summary uses for balancing is the correct one.

Options 2, 3, and 4 are described in detail in this section.



Compared to Cycle ID and Most Recent Run Date/Time

This option uses both the cycle ID and the run date and time to select the history item.

This option will use a retrieved history item for balancing if either of the following is true:

- a. The retrieved history item cycle ID is greater than the variable cycle history job's cycle ID. (See "Example A" on page 216.)
- b. The cycle IDs are equal and the run date and time of the retrieved history item is greater than the run date and time of the variable cycle history job's cycle ID. (See "Example B" on page 217.)

When to Use this Option

Use this option when your balancing jobs use the same cycle ID format and it is possible that the retrieved history item cycle ID could be the same as the current job cycle ID.

12 ■ Understanding Variable Cycle Processing

Choosing a Variable Cycle Processing Option

Example A

The following example shows which history is retrieved when the history is normally retrieved from the day before, but it's not available. ACR/Summary selects the first available history item with an earlier cycle ID. This option accommodates holidays or any other situation where data must be retrieved from the last available history.

Daily Jobs – Provide the history item				
Day	Cycle	Run No.	Run date	Run time
Monday	20060304	001	060304	10:05
Tuesday	20060305	001	060305	10:15
Wednesday	20060306	001	060306	10:02
Holiday			060307	10:12
Holiday			060308	10:03

Daily Job:
Retrieved history cycles have a cycle ID greater than that of the variable cycle job.

Weekly Job – Needs history item for balancing				
Day	Cycle	Run No.	Run date	Run time
Saturday	20060309	001	060309	12:15

Currently Running Job:
Retrieves the history cycles with a cycle ID greater than that of the variable cycle job.

Variable Cycle History Job				
Day	Cycle	Run No.	Run date	Run time
Saturday	20060302	001	060302	10:05

Variable Cycle History Job:
Provides the cycle ID and run date and time used for comparison to determine the histories to include.

Example B

The following example shows how the this option can locate history when the cycle IDs are the same.

Daily Jobs – Provide the history item				
Day	Cycle	Run No.	Run date	Run time
Monday	20060304	001	060304	10:05
Tuesday	20060305	001	060305	10:15
Wednesday	20060306	001	060306	10:02
Thursday	20060307	001	060307	10:12
Friday	20060308	001	060308	09:45

Daily Job:
When a cycle ID for the daily history is the same as that of the variable cycle history, items are retrieved based on run date and time.

Weekly Job – Needs history item for balancing				
Day	Cycle	Run No.	Run date	Run time
Saturday	20060309	001	060308	12:15

Currently Running Job:
Processes the selected cycles (see Variable Cycle History Job below) and retrieves items based on run date and time.

Variable Cycle History Job				
Day	Cycle	Run No.	Run date	Run time
Monday	20060304	001	060304	10:00

Variable Cycle History Job:
Provides the cycle ID and run date and time used for comparison to determine the histories to include.

Compared to Most Recent Run Date and Time

This option relies on the run date and time only for selecting a history item.

This option will use a retrieved history item for balancing if its run date and time are greater than the run date and time of the variable cycle job ID.

When to Use this Option

Use this option if your two jobs did not store history in the same cycle number format. For example, if one format is Gregorian and the other format is Julian.

12 ■ Understanding Variable Cycle Processing

Choosing a Variable Cycle Processing Option

Processing Logic Is Different for this Option

Note that when you choose this option, the processing logic is different:

- When filtering in history records for external item and variable cycle, histories will not be filtered based on cycle number.
- The Variable Cycle Most Recent Indicator will not be used.
- The Equal Cycle Comparison has no effect since *Compared to most recent run date and time* will not exclude histories based on cycle number.

Example

This example shows how ACR/Summary retrieves history based on run date and time alone.

Daily Jobs – Provide the history item				
Day	Cycle	Run No.	Run date	Run time
Monday	03042006	001	060304	10:05
Tuesday	03052006	001	060305	10:15
Wednesday	03062006	001	060306	10:02
Thursday	03072006	001	060307	10:12
Friday	03082006	001	060308	09:45

Weekly Job – Needs history item for balancing				
Day	Cycle	Run No.	Run date	Run time
Friday	20060308	001	060308	10:05

Variable Cycle History Job				
Day	Cycle	Run No.	Run date	Run time
Saturday	20060302	001	060302	10:05

Daily Job:
Daily and weekly jobs have a different cycle format. History is retrieved based on run date and time alone.

Currently Running Job:
Processes the selected cycles (see Variable Cycle History Job below) and retrieves items based on run date and time.

Variable Cycle History Job:
Provides the run date and time used for comparison to determine the histories to include.

Compared to Most Recent Cycle Number

This option uses cycle numbers only, no run numbers. This means it will never retrieve histories that were run on the same cycle number.

This option will use a retrieved history item for balancing if its cycle number is greater than the cycle number of the variable cycle history job.

When to Use this Option

Use this option if you always want to retrieve history items from prior cycles. This option is intended for those situations where both of the following are true:

- You use the date for the cycle number.
- If the source job runs on the same day, you do not want the history item.

Example

This example shows how ACR/Summary retrieves history based on cycle number alone and ignores the run number.

Daily Jobs – Provide the history item				
Day	Cycle	Run No.	Run date	Run time
Monday	03022006	002	060302	10:05
Tuesday	03032006	001	060303	10:15
Wednesday	03042006	001	060304	10:02
Thursday	03052006	001	060305	10:12
Friday	03062006	001	060306	09:45

Weekly Job – Needs history item for balancing				
Day	Cycle	Run No.	Run date	Run time
Friday	03062006	001	060306	10:05

Variable Cycle History Job				
Day	Cycle	Run No.	Run date	Run time
Monday	03022006	001	060302	10:05

Daily Job:
Retrieved history cycle number is greater than the variable cycle number.

Currently Running Job:
Ignores cycles with same cycle number as the variable cycle history job (even if the variable cycle job has a smaller run number) and retrieves from cycle numbers greater than that of the variable cycle history job.

Variable Cycle History Job:
Provides the cycle ID used for comparison to determine the histories to include.

12 ■ Understanding Variable Cycle Processing

Choosing a Variable Cycle Processing Option

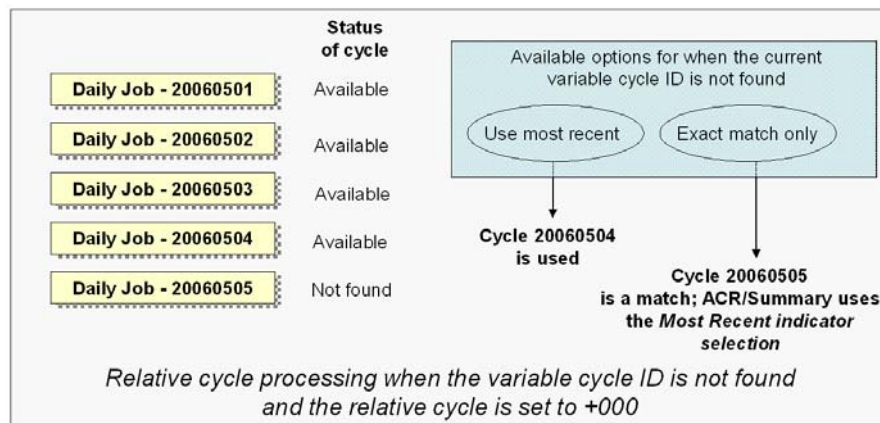
When the Variable Cycle ID Is Not Found

For those situations when the variable cycle ID is not found, you can set the **Most Recent Indicator**.

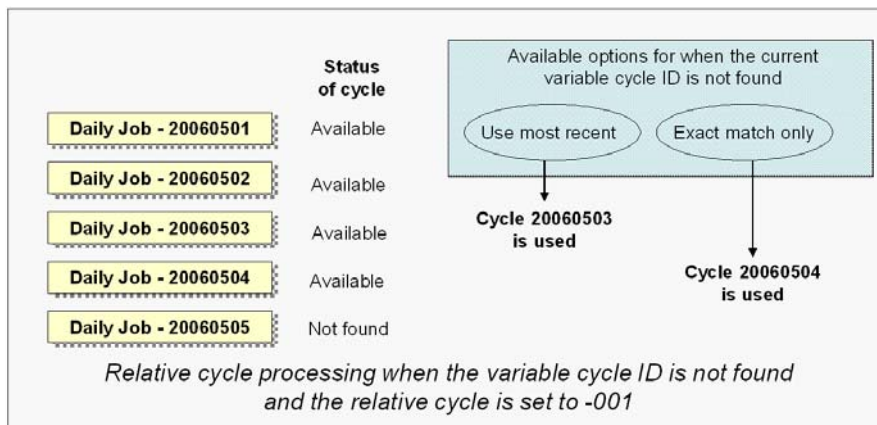
This option lets ACR/Summary automatically choose which values to use based on one of the following:

- Use the most recent cycle as +000. For example, if yesterday's run is the most recent cycle, then yesterday's cycle becomes +000.
- No adjustment; only an exact match is considered +000. In this case, the value you use for the **Most recent indicator** applies.

The graphic below shows how both options work when the relative cycle is set to +000.



The graphic below shows how both options work when the relative cycle is set to -001.



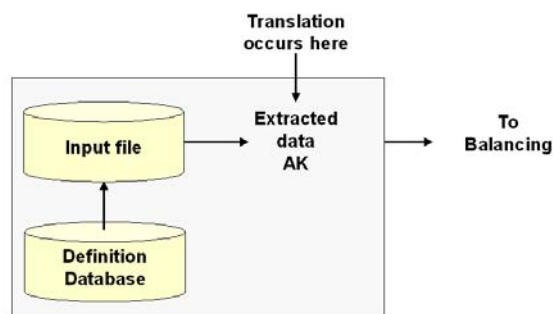
Using Translation Tables

Translation tables are used to translate values from an input source for balancing purposes. This chapter documents the translation process and helps you select and create the type of translation table you need (internal or external). This chapter includes the following sections:

- “Translation Processing” on page 221
- “Internal Vs. External Translation Tables” on page 222
- “Using External Translation Tables” on page 224
- “External Translation Table Trace Report” on page 233
- “Using Internal Translation Tables” on page 240

Translation Processing

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 balancing process.

13 ■ Using Translation Tables

Internal Vs. External Translation Tables

In the following graphic, the extracted state abbreviation AK is translated into the state code 01.



Internal Vs. External Translation Tables

External and internal translation tables share the following characteristics:

- They are both used to translate values from an input source for balancing purposes.
- They are both used to translate field items for access modes 1-5 or to translate detail fields for access modes 6-7.

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 a table consisting 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/Summary job.

For example, suppose you are balancing values from two reports, and you decide to use a region number (01, 02, etc.) as the field value. However, the region number does not show up on one of the reports. Only the region name (Dallas, New York, etc.) appears. To address this situation, you would 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/Summary. This data source can be a physical sequential file, Web page, or another type of file accessible via a user-provided program.

Note: When there is no external data source, you can use constant translation data. See *Types of External Data Sources Supported* on page 228.

Once the path and file name of the external data source have been defined in ACR/Summary, the column values in the external source can change without the need to modify the translation table definitions.

In the translation process, the values to be translated are passed from field items (for access modes 1-5) or from detail fields (for access modes 6-7) to input-output parameters and then compared against values in the external translation table. If a value finds a match in the external translation table, it will be replaced with a translated output value that is then sent back to the original field item or detail field or stored as an extraction variable.

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 balancing jobs.

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. Display the External Translation Table View by selecting **File > New > Table (External/Internal/Cycle) > External Translation Table**.
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 231).
 - b. Identify the **Table 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 228.
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**—Specify this if you want the table to be built one time for each job run. This is the default.
 - **On each lookup**—Specify this if you want the table to 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 folder 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.

See “Combining Data for Translation Using Extraction Variables” on page 232 for more information.

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.

13 ■ Using Translation Tables

Using External Translation Tables

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.

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 balancing 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 224, the Constant Table Data section 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 enter **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 record/field (file) definitions, specify translation as follows:
 - For access modes 1-5, the data to be translated will be a field item. When you complete the Field Item dialog box, in the **Table Name** field, enter or select the name of the external translation table. Select **List Missing Entry** if you want the SYSOUT report to show a warning message if the value to be translated is not found in the table.
 - For access modes 6-7, the data to be translated will be a detail field. When you complete the Detail Field dialog box, click **Use Translation Table**, select **Translate** and complete the Translation Table Information dialog box with the translation table name and other required information.
13. Save the job and update the definition database.
14. When you are ready to run the job, select the **Balancing** icon to display the Balancing dialog box.
15. Unless the **Table Dataset Organization** for the table is Constant Translation Data, you must do the following:
 - a. Select **Add** to display the File ID and File Name dialog box.
 - b. In the **File ID** field, enter the table name.
 - c. In the **File Name** field, enter or browse to select the full path and file name of the external source file for the table.
 - d. If you want to request an External Translation Table Trace Report (access modes 6 or 7 only), see “External Translation Table Trace Report” on page 233.

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 231.

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.

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.	
.	.		
.	.		

13 ■ Using Translation Tables

Using External Translation Tables

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.

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

Report

You can generate a report on the external translation table processing, as described in “External Translation Table Trace Report” on page 233.

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” on page 229 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.

13 ■ Using Translation Tables

Using External Translation Tables

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

Combining Data for Translation Using Extraction Variables

The **Indicate what the parameter refers to** field in the Input-Output Parameter dialog box enables you to combine values from multiple field items or detail fields for translation, as in the following example.

Example

Suppose you need to translate employee names to employee numbers. You need to create an external translation table which could be called Names to Numbers. Following is a sample of the external translation table columns that will be needed:

LAST NAME	FIRST NAME	EMPLOYEE NUMBER
JOHNSON	ADAM	11111111
JOHNSON	JANE	22222222
SMITH	JOHN	33333333

Field Items

In the balancing job, you would define two field items to be used in the translation:

The first field item picks up the last name.

- The **Target Area** is Extraction Variable.
- The **Extraction Variable** is assigned a name, LAST NAME.

The other field item picks up the first name.

- The **Target Area** is Extraction Variable.
- The **Extraction Variable** is assigned a name, FIRST NAME.
- In the **Table Name** field, the external translation table table, Names to Numbers, is specified, and **List Missing Entry** is selected.

Translation Table Input-Output Parameter Definitions

Because two employees have the same last name, you are going to need to include both the first name and the last name in the external translation table to determine the correct employee number. To do this you will define two input-output parameters.

In the input-output parameter for the last name, in the **Indicate what the parameter refers to** field, Extraction Variable is selected. This indicates that the translation process must read the next input-output parameter before performing the translation.

In the input-output parameter for the first name, in the **Indicate what the parameter refers to** field, Extracted Data is selected. This indicates that all of the input-output parameters necessary to perform the translation have been read.

External Translation Table Trace Report

This report, available for access modes 6 and 7, 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.

13 ■ Using Translation Tables

External Translation Table Trace Report

The information below is divided into the following sections:

- “Generating the Report” on page 234
- “External Definition List Section” on page 235
- “Table Build Rule Processing Per Record Section” on page 236
- “Formatting of Table Build Rules” on page 237

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 - UNIETR.RPT on page 298.

Generating the Report

This report will be generated when you run balancing if you request it as follows:

1. In the Balancing dialog box, under File Interface Input, click **Add**.
2. When the File ID and File Name dialog box appears, in the **File ID** field, enter the name of the external translation table.
3. In the **File Name** field, enter the full path and file name of the external source file for the table.
4. Select **Produce Trace Report**.
5. Select **External Translation Table**.
6. Select **OK** to save your settings and return to the Run Balancing dialog box.
7. Select **OK** to run balancing and generate the report.

External Definition List Section

The first section is a formatted list of the table build rule definitions. Following is an example. 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 )
```


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.

13 ■ Using Translation Tables

External Translation Table Trace Report

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.

*Column Assignment Field Format***Format**

ASGN(AAAA/BB/CCC/DDDD) EEEEE (FFF/GGG/HHHH) LIT JJJJJJJJJJJJJ

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

JJJJJJJJJJJJJ 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.

13 ■ Using Translation Tables

Using Internal Translation Tables

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

Creating Internal Translation Tables

Set up an internal translation table as follows:

1. From the Control Task (Job) View, select **File > New > Table (External/Internal/Cycle) > 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.

3. Use the Translation Table Entry dialog box to define each table entry, consisting of an input value and an output (translated) value.
 - The Input Value is the value that will be extracted using a field item 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 field item or detail field definitions.
4. Save the table. The file extension will be .tbl.
5. Incorporate the translation table into the job in one of the following ways:
 - For access modes 1-4, complete the **Table Name** field on the Field Item dialog box.
 - For access modes 6-7, select **Translate** on the Detail Item dialog box. Complete the Translation Table Information dialog box.

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 balancing 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.

Next, you would define field item information in the file definitions to extract the region name. On the Field Item dialog box, specify the name of the table in the **Table Name** field and select **List Missing Entry** to specify that any missing entry should be listed.

13 ■ Using Translation Tables

Using Internal Translation Tables

When the balancing function is executed, the region name is extracted from the report. The translation table is searched for a corresponding input entry. If an input entry is found, the table output is moved to the field value. If not, a warning message may be issued in the SYSOUT.rpt and processing may be halted or continued (depending upon the translation table option chosen).

Importing Definitions

This chapter provides documentation for importing definitions into ACR/Summary Client. It includes the following sections:

- “Purposes of Importing” on page 243
- “Procedure for Importing Definitions” on page 243
- “Understanding @FILETMP Jobs” on page 245
- “Using @@FILLER Records” on page 245

Purposes of Importing

Typically, definitions are imported for the following purposes:

- To move ACR/Summary batch definition transactions created in a prior release into the current release.
- To move ACR/Summary 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 381.
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.

14 ■ Importing Definitions

Procedure for Importing Definitions

4. Complete the dialog box based on the instructions in the dialog box help. Also see “Considerations for Completing the Import Options Dialog Box” on page 244. When you have completed the dialog box and click **OK**, a new Control Task View will be opened for each imported job, table, file, or history analysis document.
5. Click the **Save** icon to save 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 245.
- 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 383.
- 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 245.
- 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 (see step 4 on page 244).

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 244.

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.
- Integrity checking is not performed on filler records.
- 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.

14 ■ 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 hierarchical list of items under three categories: History Items, Calculated Items, and Rules. In the History Items section, three items are listed: 1 - EMPLOYEES LAST MONTH, 2 - SICK DAYS LAST MONTH, and 3 - NET MAN DAYS LAST MONTH. In the Calculated Items section, two items are listed: 1 - C001 = (EMPLOYEES + SICK DAYS) * (EMPLOYEES LAST MONTH) and 2 - C008 = (SICK DAYS) * (NET MAN DAYS). In the Rules section, two items are listed: 0 - 1 - R001 (C001) = (EMPLOYEES) -- Active Rule and 0 - 2 - R005 (SICK DAYS) = (EMPLOYEES) -- Active Rule. Blue arrows point from the text on the right to the first three items in each category. The text on the right reads: "Notice the history items, calculated items, and rules are renumbered consecutively."

The following example shows items imported with the Insert filler records when item numbers are skipped option **selected**.

The screenshot displays a hierarchical list of items under three categories: History Items, Calculated Items, and Rules. In the History Items section, the first item is EMPLOYEES LAST MONTH, followed by seven @@FILLER items, and then NET MAN DAYS LAST MONTH. In the Calculated Items section, 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). In the Rules section, 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 on the right to the first item in each category. The text on the right reads: "Notice @@FILLER always becomes the item name when importing history items, calculated items, or rules with non-consecutive numbers."

Using History Analysis

This chapter discusses the History Analysis feature and contains the following sections:

- “What is History Analysis?” on page 247
- “Creating History Analysis Definitions” on page 248
- “Report Samples” on page 250

What is History Analysis?

A history analysis job is a set of definitions for analyzing and reporting on data from the history database. The file extension for a history analysis job is `.had`.

The History Analysis Report can be created in one of three formats, each of which identifies the type of data that will appear across the horizontal axis of the report: (job IDs, relative cycles, or internal items).

Format (Horizontal Axis)	Purpose
Job ID	Allows you to look at multiple jobs to analyze and compare data from different internal items and/or relative cycles over a period of time.
Relative cycle	Allows you to look at a range of cycles and analyze trends in job IDs or internal items.
Internal item	Allows you to look at one or more internal items and analyze data from different job IDs and their associated relative cycles in order to identify changes in the items over a period of time.

Examples of each report format are provided in “Report Samples” on page 250.

A history analysis job can be run independently or in conjunction with a balancing job.

Columns and Rows

A history analysis report may have a maximum of 100 columns (although a maximum of 12 can be printed across the width of the report) and a maximum of 100 rows. Each column and row is one of two types: detail or summary. A detail column or row contains a value taken directly from the history database. A summary column or row contains the results of a computation performed on values from other columns or rows, respectively. Information in any column or row may be printed or suppressed. If printing is suppressed, the values in the column or row are still available for calculations.

Creating History Analysis Definitions

The steps for creating and running history analysis definitions are as follows:

1. From the Control Task (Job) View, select **File > New > History Analysis Document** to display the History Analysis View.

Note: To open definitions for an existing History Analysis Report, select File > Open. Change the List Files of Type field to History Analysis (*.had), and select the appropriate file.

2. Complete the Basic Information section. Keep the following in mind:
 - To specify that a history analysis job will run in conjunction with a particular balancing job, name it to match the **Job Name** and **Step Name** on the Control Task (Job) Name dialog box for the balancing job. For example, if the balancing job's job name is HIST1REP and its step name is STEP1, you would enter HIST1REPSTEP1 in the **Name** field on the History Analysis Name dialog box.
 - Use the History Analysis Format option to define a report format. For more information, see “What is History Analysis?” on page 247.
3. Complete the Column Information section. Keep the following in mind:
 - A column can be one of the following types:
 - Detail column**—Detail columns need to be defined before summary columns are defined. They contains values taken directly from the history database. The fields on the Detail Column Information dialog box will vary depending on the report format you chose in the previous step.

Summary column—Contains a value resulting from performing computations on values in other columns. For information on the computation types, see Appendix E, “Non-Biased Variance and Standard Deviation Computations”.

- Up to 12 columns can be printed across the width of the report. Information in each column can be printed or suppressed. If printing of a column is suppressed, the values in the column are still available for calculations.
4. Complete the Row Information section. Keep the following in mind:
- A History Analysis Report can have up to 100 rows. A row can be one of the following types:
 - Detail row**—Detail rows need to be defined before summary rows. They contains values taken directly from the history database.
 - Summary row**—Contains a value resulting from performing computations on values in other rows. For information on the computation types, see Appendix E, “Non-Biased Variance and Standard Deviation Computations”.
 - Format row**—Prints the specified character symbol in every column. This is useful for defining blank lines or dashed lines to separate report sections.
 - Information in each row can be printed or suppressed. If printing of a row is suppressed, the values in the row are still available for calculations.
5. Complete the Analysis Rule Information section. Keep the following in mind:
- Analysis rules make it possible to analyze history data using a variety of comparisons and computations and to report on exceptions. A maximum of 100 analysis rules may be defined for each report.
 - The following terms are used in defining analysis rules:
 - Analysis base**—The column or row to be used as the basis for comparison.
 - Analysis window**—The rows and columns to be compared against the analysis base.
 - Tolerance**—The amount of acceptable deviation from the value in the analysis base.

15 ■ Using History Analysis

Report Samples

6. Complete the Legend Information section to include an Explanation of Codes section on the last page of the history analysis report. This section will list and describe each exception code.
7. Save the file. The name of the file is not determined by the name you specified when you entered a name in the Name field on the History Analysis Name dialog box. So if this is the first time you have saved the file, you will be prompted to enter a name in the File Save As dialog box.
8. Update the definition database.
9. To run the history analysis job, do either of the following:
 - Run the balancing job that has the same name as the history analysis job (see step 2 on page 248).
 - Run the report independently by opening the file (*.had) and selecting **Run > History Analysis**.

The output file name will be UNISPRD2.RPT.

Report Samples

Job ID Format Example

Over the last five years, a large retail store has opened six new stores. Balancing jobs have been set up for each store to report total sales, gross profit, and profit margin on an annual basis. The results were stored in history.

You use the historical data to create the following report, which analyzes the increase in total sales earned per year, the percent increase in the gross profit per year, and the percent increase in the profit margin per year.

ACR/S releasenumbr		INFOGIX, INCORPORATED		COPYRIGHT INFOGIX, INC		PAGE 001	
DATE: yy/mm/dd		SUMMARY SPREADSHEET REPORT					
TIME: 17:41:10		REGIONAL SALES TOTALS		DATA BASE VERSION:0000087/0000115			
		SPREADSHEET ID=JOBID					
		CYCLE#=20000631 RUN#=000					
ROW#	DESCRIPTION	STORE 1 TOTALS	STORE 2 TOTALS	STORE 3 TOTALS	STORE 4 TOTALS	STORE 5 TOTALS	STORE 6 TOTALS
1	TOTAL SALES	\$51,432.25	\$42,094.53	\$24,870.98	\$63,980.12	\$15,665.05	\$23,587.98
2	GROSS PROFIT	\$71,005.00	\$32,028.50	\$23,660.57	\$63,001.20	\$14,660.00	\$22,669.00
3	PROFIT MARGIN	\$32,586.30	\$23,456.50	\$13,886.50	\$ 5,553.10	\$ 8,438.50	\$10,230.00
4							
5	INCREASE IN TOTAL SALES	\$ 4,500.50	\$ 1,001.25	\$ 857.22	\$ 960.58	\$ 980.75	\$ 325.87
6	PERCENT INCREASE IN GROSS PROFIT	12	8	10	18	12	5
7	PERCENT INCREASE IN PROFIT MARGIN	5	6	7	11	8	2

Relative Cycles Format

A large insurance company bills patients for services rendered by each surgeon. A balancing job produces a monthly report that identifies delinquent accounts with payments beyond 30, 60 and 90 day and stores the results in history. You use the historical data to create the following report, which analyzes the number of accounts that are delinquent beyond 30, 60 and 90 days for the last six months.

ACR/S releasenumbr		INFOGIX, INCORPORATED				COPYRIGHT INFOGIX, INC	
DATE: mm/dd/yy		SUMMARY SPREADSHEET REPORT				PAGE 001	
TIME: 15:11:15		DELIQUENT ACCOUNT TOTALS				DATA BASE VERSION:0000087/0000115	
SPREADSHEET ID=RELCTC							
CYCLE#=20000631 RUN#=000							
ROW#	DESCRIPTION	01/31/08	02/28/08	03/31/08	04/30/08	05/31/08	06/30/08
1	>30 DAYS DELIQUENT	215	213	191	112	95	68
2	>60 DAYS DELIQUENT	141	98	157	77	57	36
3	>90 DAYS DELIQUENT	80	50	60	43	32	12

Internal Item Format

A telecommunications company provides internet hosting services for four regions. Each region tracks the number of prospects, the type of service sold to those prospects, and the number of prospects that did not purchase either service. A balancing job ran last month for each region to report on these metrics. Each metric represented an internal item value that was stored in history. You create the following report to analyze by region the number of prospects for each service, the type of service that is most in demand, and the number of prospects that did not purchase the service.

ACR/S releasenumbr		INFOGIX, INCORPORATED				COPYRIGHT INFOGIX, INC	
DATE: yy/mm/dd		SUMMARY SPREADSHEET REPORT				PAGE 001	
TIME: 06:12:32		HOSTING SVC REGION REPORT				DATA BASE VERSION:0000087/0000115	
SPREADSHEET ID=INITITEM							
CYCLE#=20060821 RUN#=000							
ROW#	DESCRIPTION	REGION 1	REGION 2	REGION 3	REGION 4		
1	# OF PROSPECTS	815	613	174	242		
2	MOST DEMANDED SERVICE	T1	DSL	WIRELESS	DSL		
3	# OF PROSPECTS NO PURCHASE	624	415	61	53		

15 ■ Using History Analysis

Report Samples

Using the Utilities

This chapter documents the utilities available in ACR/Summary. It includes the following sections:

- “Running Utilities” on page 254
- “Database Initialization” on page 254
- “Database Update” on page 254
- “List Definitions” on page 255
- “Control Update Information Report Utility” on page 255
- “Build Batch Transactions” on page 256
- “Cross Reference Internal Items” on page 256
- “Cross Reference File Definitions” on page 257
- “List History” on page 257
- “Initialize History” on page 257
- “Copy History: Copy All Records” on page 258
- “Copy History: Copy Selected Records” on page 258
- “Copy History: Unload Records to a Work File” on page 258
- “Copy History: Reload Records to a Work File” on page 259
- “Delete History” on page 259
- “Reorganize History” on page 259
- “Extract History” on page 260
- “Reprint Control Report” on page 263
- “Print Management Report” on page 263
- “Update History” on page 263
- “Create Comma Delimited Text File” on page 264
- “Integrity Check” on page 264

Running Utilities

To run most utilities, select the **Database 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 Initialization** and then complete the dialog box.
-

If a utility is accessed and executed by another method, instructions are provided in the description for that utility below.

The dialog box help provides field definitions and other information.

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/Summary for the first time.

Reinitializing a database erases the contents. See the Database Initialization dialog box help for more information.

For a description and example of the report generated by this utility, see “Database Initialization Report - UNIIR.RPT” on page 302.

To run Database Initialization, select **Run > Database Initialization** and then complete the dialog box.

Database Update

This utility loads your definitions for the current job, table, or history analysis document into the definition database that is specified on the Add/Edit Configuration dialog box - file information tab. This is necessary in order to use the definitions in ACR/Summary processing.

For a description and example of the report generated by this utility, see “Build ACRUPDT Report - UNIDBR.RPT” on page 313.

To run a Database Update, select the **Database Update** icon or select **Run > Database Initialization** and then complete the dialog box.

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. Types of definitions you can select include:

- All types
- Jobs
- Files
- Internal translation tables/cycle tables
- External translation tables
- History analysis definitions
- 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 description and example of the report generated by this utility, see “Database History and Statistics Report - UNIDLR.RPT” on page 303.

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 definitions
- 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 of the report, see “Print Control Update Information Report - UNIDLRCU.RPT” on page 312.

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. Types of transactions you can select include:

- All types
- Jobs
- Files
- Internal translation tables/cycle tables
- External translation tables
- History analysis definitions
- 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.)

This utility is useful for making mass definition changes, creating an audit trail, transferring definitions between platforms, etc.

For a description and example of the report generated by this utility, see “Build ACRUPDT Report - UNIDBR.RPT” on page 313.

Cross Reference Internal Items

This utility prints a report that lists the definitions in the definition database for a specified job ID. For each internal or extended internal item defined for that job ID, the report lists the item’s description and any job ID with a related item number. Related items can include history items, calculated items, rules, field items, and history analysis definitions.

If the internal item is referenced by a calculated item from another job, and the referencing calculated item is also stored as an internal item in the same job, the report prints an indicator (*) next to the calculated item and a note at the end of the report.

For a description and example of the report generated by this utility, see “Cross Reference Internal Items Report - UNIDXR.RPT” on page 316.

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 “Cross Reference File Definitions Report - UNIDXR.RPT” on page 317.

List History

Depending on your selections in the dialog box, this utility generates a selection of reports regarding the records in the history database. It can be useful in establishing an audit trail or for troubleshooting.

For descriptions and examples of the reports that can be generated by this utility, see the following:

- “Database History and Statistics Report - UNIHLR.RPT” on page 318.
- “History Data Detail Report - UNIHSTD.RPT” on page 319.

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.

For a description and example of the report generated by this utility, see “Database Initialization Report - UNIIR.RPT” on page 302.

Copy History: Copy All Records

This utility copies all records from one history database to another. The source history database and the target history database are specified by you. This can be useful for backup purposes or to create a copy of a production database to be used in a test environment.

For a description and example of the report generated by this utility, see “Transaction Listing Report - UNITR.RPT” on page 320.

Copy History: Copy Selected Records

This utility allows transfers selected records from an existing history database directly to another file. You can select a range of histories based on job ID and/or date and time and copy them to another database. This can be useful when copying a test job into production or vice versa.

For a description and example of the report generated by this utility, see “Transaction Listing Report - UNITR.RPT” on page 320.

Copy History: Unload Records to a Work File

This utility writes the history database records to a work file.

This utility might be used to:

- Backup/restore files
- Move a history database from one host to another

For a description and example of the report generated by this utility, see “History Database Copy Report - Summary of Histories Unloaded - UNIHTR.RPT” on page 321.

Copy History: Reload Records to a Work File

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 is used after you use the Unload Records utility. It enables you to select a range of histories based on job ID and/or date and time and reload them (from the work file to which they were unloaded) into the history database.

For a description and example of the report generated by this utility, see “History Database Copy Report - Summary of Histories Reloaded - UNIHTR.RPT” on page 322.

Delete History

This utility can perform a logical delete of one or multiple history records by job ID, cycle, and date. Wildcards can be used in the **From Job ID** field to select multiple job IDs. This can be useful when a job ran that was not scheduled to run, when a job ID is no longer needed in the history database, etc.

For a description and example of the report generated by this utility, see “Transaction Listing Report - UNITR.RPT” on page 320.

Reorganize History

This utility physically deletes all logically deleted records from this history database. Run this utility regularly to ensure optimum processing efficiency.

For a description and example of the report generated by this utility, see “History Database Reorg Report - UNIHRR.RPT” on page 322.

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:

- **Default (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 264.
- **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.

The record layouts for each format are shown in the next section.

Regardless of the option you use, the data extracted includes the job ID used to store the history, its cycle ID, its overall return code, and the internal item values stored.

This utility generates the Transaction Listing Report. For more information, see “Transaction Listing Report - UNITR.RPT” on page 320.

Extract History File (Packed) Layout

Fld #	Description	Position	Length	Format	Value
1	Balancing Job Information Job Name	1-8	8	X(8)	
2	Step Name	9-16	8	X(8)	
3	Qualifier	17-18	2	X(2)	
4	Cycle Number	19-26	8	9(8)	
5	Run Number	27-29	3	9(3)	
6	Reserved	30-39	10	X(10)	
7	Run Type	40	1	X(1)	Space, D, R, F
8	Run Date	41-44	4	S9(7)	Packed
9	Extract Run Time	45-48	4	S9(7)	Packed
10	Program Name	49-56	8	X(8)	
11	Return Code	57-60	4	X(4)	
12	Rerun Date	61-64	4	S9(7)	Packed
13	Rerun Time	65-68	4	S9(7)	Packed
14	Rerun Count	69-72	4	S9(7)	Packed
15 ¹	Run Value-X	73-872	800	100 X(8)	
16 ¹	Run Value-N	73-872	800	100 S9(15)	Packed
17	Extended Run Value-X	873-8872	8000	100 X(80)	
18	Extended Run Value-N	873-8872	8000	100 S9(30) 100 Filler X(64)	Packed

1 - These positions can be populated by 100 8-byte alphanumeric fields or 100 15-byte packed fields.

16 ■ 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	Cycle Number	19-26	8	9(8)	
5	Run Number	27-29	3	9(3)	
8	Run Date	30-37	8	9(8)	
7	Run Time	38-43	6	9(6)	
8	Return Code	44-47	4	X(4)	
9	Rerun Date	48-55	8	9(8)	
10	Rerun Time	56-61	6	9(6)	
11	Rerun Count	62-69	8	9(8)	
12	Run Value Count	70-72	3	9(3)	
13	Cycle Date	73-80	8	X(8)	
14	Format Indicator	81-1880	1800	100 X(1)	
15	Run Value-X	81-1880	1800	100 X(8) 100 Filler X(9)	
16	Run Value-N	81-1880	1800	100 X(1) 100 9(16)	
17	Extended Run Count	1881- 1883	3	9(3)	
18	Extended Format Indicator	1884- 9983	8100	100 X(1)	
19	Extended Run Value-X	1884- 9983	8100	100 X(80)	
20	Extended Run Value-N	1884- 9983	8100	100 X(1) 100 9(31)	
21	Reserved	9984- 9990	7	X(7)	

Reprint Control Report

This utility reprints the (Balancing) Control Report (UNIACR.RPT) for a job using control values stored in the definition and history databases. This enables you to generate the report without rerunning the job.

Note: XML generation of the Control Report is not supported by this utility.

Because the definitions used are the ones that exist when the utility is being run, a possibility exists that the results this utility yields will differ from the original balancing run. If someone has modified the definitions being used and updated the history control values since the original run, the Control Report will show results that differ from those of the original run.

For a description and example of the Control Report, see “Control Report - UNIACR.RPT” on page 292.

Print Management Report

This utility allows you to print a report that summarizes balancing history information for management purposes.

This utility can process an unlimited number of job IDs, or it can focus on a single job ID. It also analyzes and prints the contents of the database based on user-defined selection criteria. Managers can use this utility to analyze the functioning of the system. For example, you can run a report showing all the return codes greater than 0 (all out-of-balance jobs) for a particular time period.

For a description and example of the report generated by this utility, see “Print Management Report - UNIMR.RPT” on page 323.

Update History

This utility updates (i.e., replaces) one or more control values in the history database for a specified job ID and cycle ID. This batch file 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.

For a description and example of the report generated by this utility, see “Transaction Listing Report - UNITR.RPT” on page 320.

Create Comma Delimited Text File

The Create Comma Delimited Text File utility creates a file with field values separated by commas. This utility can only be used after you have created an extract file from the Extract History utility (see “Extract History” on page 260).

The Create Comma Delimited Text File utility reads the extract file and creates a text file with sequential fields separated by commas.

The file is in the following format: Job name, step name, qualifier, cycle number, run number, run type, run date, run time, return code, rerun date, rerun time, rerun count, items

All text values are surrounded by quotation marks.

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.

Once created, the comma delimited text file can be downloaded to the desktop. Most desktop spreadsheet applications require an extension of .csv when opening a comma-delimited file, so use this extension when downloading.

For a description and example of the report generated by this utility, see “Transaction Listing Report - UNITR.RPT” on page 320.

Integrity Check

This utility examines the validity of the currently active control task (job), table, or history analysis document. It scrutinizes each item and its relationship to other items and identifies any inconsistencies.

Integrity checking is turned on/off through the Options menu. Unless turned off, an integrity check is performed when you do any of the following:

- Run a balancing job.
- Run a history analysis document.
- Update the definition database.
- Select the **Integrity Check** icon or selecting **Run > Integrity Check** from the menu.

If you have errors, the Integrity Check Results window will display messages indicating the problems and where they occurred. Results of an integrity check also appear on the System Messages Report (SYSOUT.rpt).

Correct the errors before rerunning the balancing job, history analysis document, or database update.

An integrity check is not run in the following situations:

- If you create a control task using the reserved name @FILETMP, an integrity check will not be performed on any translation tables referenced by this task.
- Filler records in history items, calculated items, or rules are not checked.
- For rules, only the left hand side (LHS) of conditional rules is checked. If the LHS contains any conditional rules with a higher rule number than the current conditional rule, an integrity check message will be issued. Standard rules are not checked.

Modeling Job and File Definitions

This chapter explains how to model job and file definitions. This feature can save you significant effort in creating and maintaining balancing jobs while reducing errors. This chapter contains the following sections:

- “What is Modeling?” on page 267
- “Modeling Terminology” on page 268
- “Modeling Job Definitions” on page 268
- “Modeling File Definitions” on page 272

What is Modeling?

Modeling is a feature that allows file definitions and job definitions stored in the definition database to be referenced by one or more balancing jobs. Consisting of a **base model** or **template** that contains the definitions, and a **model user** that uses the definitions, modeling reduces the need to recode definitions and ensures consistency between balancing jobs that have the same requirements.

A single balancing job can model job definitions, file definitions, or both.

Benefits of modeling include the following:

- Decreases the amount of maintenance required by allowing you to change the base job and have all model users utilize the change.
- Reduces the need to re-enter definitions for balancing jobs with similar extraction and/or balancing requirements.
- Decreases the size of the definition database by storing one set of core definitions that are referenced by other balancing jobs.
- Ensures balancing jobs are using the same definitions (i.e., meet the same requirements).

Modeling Terminology

The following terminology is used here when describing modeling:

Base Model

The **base model**, or **base**, contains the definitions stored on the definition database which are referenced by balancing jobs. The base definitions can exist within a balancing job, a template, or an input source.

A **template** is a set of core definitions that are referenced by a modeling job. A template consists of only file definitions or job definitions. It cannot be run independent of the balancing job that is referencing the definitions.

Model User

The **model user** uses the definitions in the base model. Based on the type of definitions modeled, the model user can be a file or a job.

There is no limit to the number of model users that can be created; the only restriction is the base and the model user must exist on the same definition database.

Base model definitions cannot be deleted by the model user. Therefore the base model and template must contain only the definitions to be modeled.

Modeling Job Definitions

Modeling job definitions allows you to have a standard set of controls that can be executed from any job. For example, you could use modeling to consistently bring in a cycle from an input source and have it applied consistently across all jobs.

You can also use modeling to consistently execute JSQs. For example, if you have multiple jobs that are similar, you can create a common base job, then define only the differences in the model users. Then, if the common elements defined in the base job change, you need only make the changes in the base job, and they will be applied to all model users.

Note: For more information on JSQs, see *Job/Step/Qualifier (JSQ)* on page 76.

Job Definitions that Can Be Modeled

Job definitions that can be modeled include:

- Internal items
- Extended internal items
- Calculated items
- History items
- Balancing rules
- Messages
- Control Report Copy 1
- User Report
- Free-Form Report

Job definitions that cannot be modeled include:

- Store history
- Set return code
- Cycle processing
- Control Report Copy 2
- Recap Report
- Round results
- 22-character numeric formatting
- Control Report XML Option
- Job comments

How Model User and Base Model Job Definitions Are Merged

When the model user job runs, the criteria described below will be applied to merge the model user job definitions and the base model definitions into a single executable set of run-time definitions:

- Some or all of the base model's item numbers (such as I-004, E-002, R-006) may be duplicated in the model user job definitions. In such cases, the duplicate items from the **model user** job definitions take precedence and will be used instead of the base model's corresponding items.

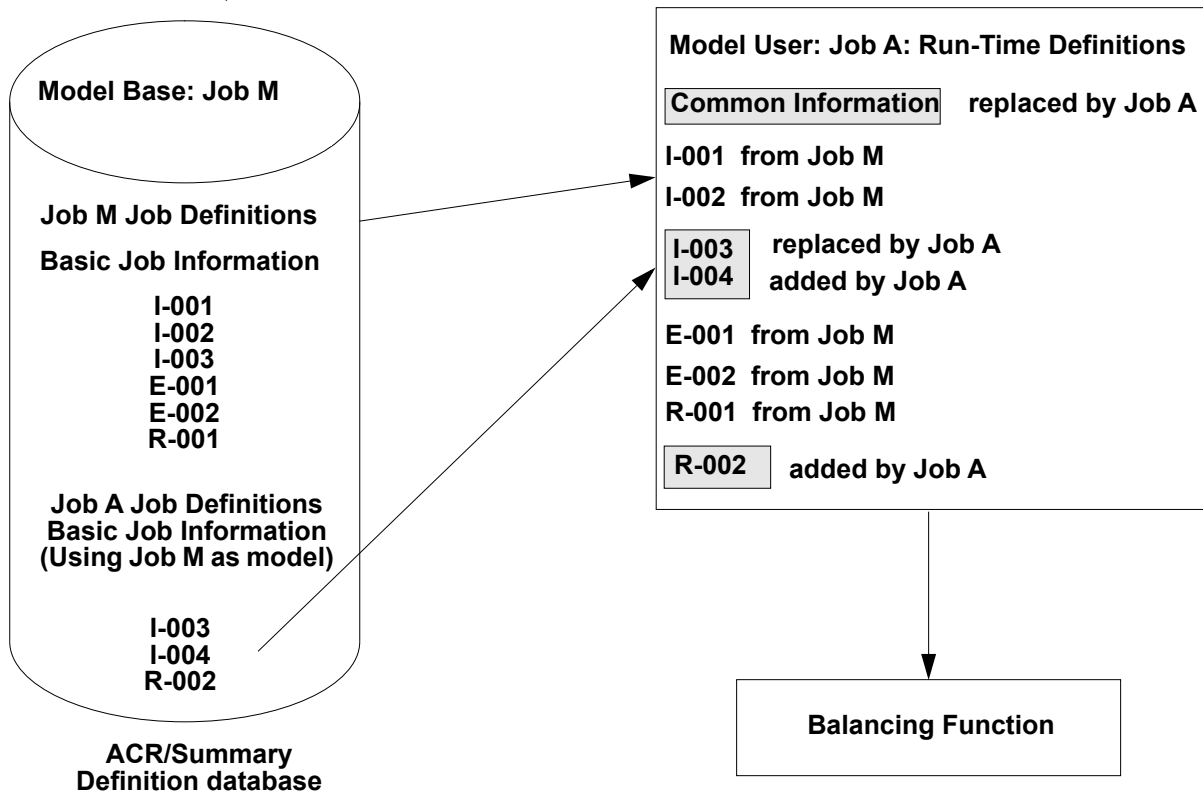
17 ■ Modeling Job and File Definitions

Modeling Job Definitions

- When the model user contains items whose item numbers are not duplicated in the base model, these items will be included when the model user job runs, in addition to the base model job definitions.

Note: Because of the nature of free-form report definitions, you cannot merge the base model's free-form report definitions with those of the model user. If you create a free-form report in your current model user job, that report will override the report in the base model. This will happen even if you set your current job's free-form report print options to Do not print the report.

The following diagram shows how a sample **model user Job A's** definitions would look at run time, after referencing **base model Job M**.

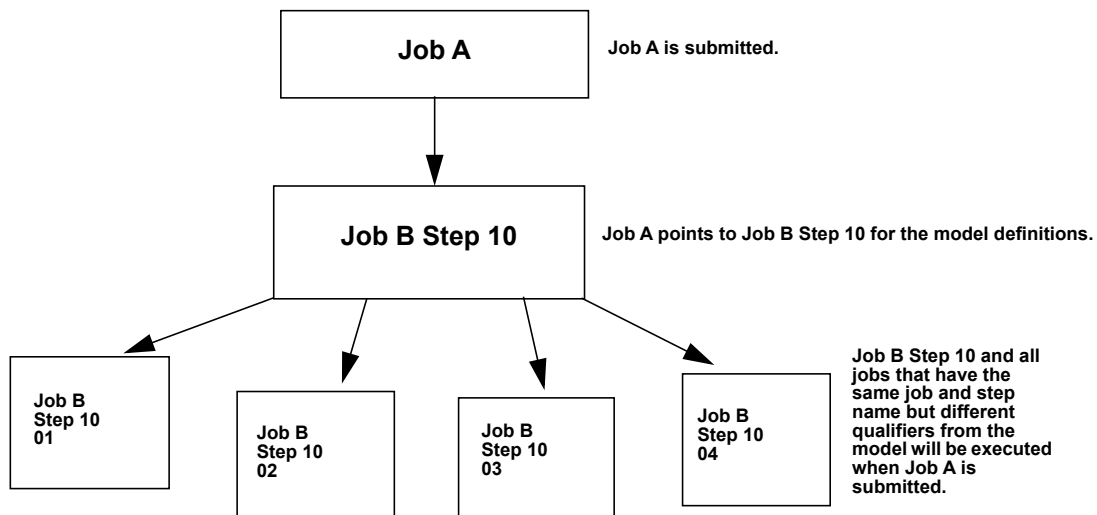


Specifying a Job as a Model User

If you want the current job to be a **model user** (that is, you want the current job to use job definitions from another job, the **base model**) you need to complete the Control Task (Job) Name dialog box's Model Definition section as follows:

1. Select one of the following options to specify how model definitions will be used with this job:
 - **Current job definitions will be merged with model Job ID's.** The current job will use both the base model definitions and the definitions that you set up within this (model user) job.
 - **This model job ID and all job IDs with the same job name and step name but with different qualifiers from the model will execute when the job is submitted.**

With this option, the **base model** that is specified in the **Model Job ID** field below will execute, along with any other jobs that share the same job name and step name as the model job but have different qualifiers. The following example illustrates this option:



2. Complete the **Model Job ID** field with the job ID of the **base model**. The model user is the job or template that contains the definitions stored on the definition database that will be referenced in the current job.

Note: Using embedded masks in history items:

In the History Job ID field for history items, an embedded mask (*) can be used for job modeling. When model definitions are used, the asterisk causes the corresponding character from the current job ID to replace the asterisk in the history item's job ID. This is also true of a mask in the Variable Cycle Job ID field. When model definitions are used, the asterisk causes the corresponding character from the current Job ID to replace the asterisk in the variable cycle job ID.

Modeling File Definitions

Purposes of File Modeling

File modeling is applicable in access modes 1 through 5. It enables you to define a set of file definitions for an input source as a base model to create definitions for another input source, the model user. File modeling is typically used when input sources have identical layouts and have similar extraction requirements. Since the values are in the same location and extracted in the same manner, modeling eliminates the need to recode the file definitions. For example, you could set up similar file definitions for similar sales reports generated by each store within a nationwide chain.

File modeling is also useful if you want to produce separate Control Reports for multiple subsets of the same report(s) or file(s). For example, a store might generate a monthly report listing sales information for every department. You could use file modeling to create similar file definitions for each department, and then set up balancing jobs to generate a Control Report for each department.

To ensure consistency in the extraction process, the base model input source and the model user input source should use the same access mode.

File Definitions that Can Be Modeled

All of the file definition types for access modes 1 through 5 can be modeled. These include:

- Begin/end/reset reference records
- Embedded keys

- File keys
- Relative record
- Total/tally/hash records
- Page/line records
- Field items

Guidelines for a Defining a Base Model for File Modeling

The base model must include only definitions needed by every model user. You cannot, when defining the model user, delete definitions from the base. However, you can add to or replace definitions from the base.

If a set of definitions for an input source that you want to use as a base model includes any definitions that will not be required by every model user, delete the extra definitions from the base. In some situations, this may mean that you need to create a base model whose definitions are not complete enough to use by themselves, but that simply serve as the starting point for all the other file definitions you want to set up.

The procedures you need to follow in order to create and use model file definitions are described in the following sections:

Specifying a Base Model for File Modeling

Specify the base model file definitions as follows:

1. Open the job that contains (or will contain) the input source that will be used as a **base model**.
2. In Input Source View, in the Base File Indicator dialog box, set the **Do you want to use this file's definitions as a base to be modeled by other files?** field to Yes.
 - If you create any new record/field definitions after specifying the input source as a base model, all records in the Record/Field Definitions section will be preceded by sequence/item numbers. A sequence/item number is a 6-digit sequence number and a 3-digit item number assigned by the system to each selection group/file definition combination when file modeling is being used. These numbers default to zero, so the sequence/item numbers will initially be (000000 / 000). In the input source view, you can identify selection groups because the first definition in each selection group begins at the left margin of the record. Additional definitions within the same selection group are slightly indented.

17 ■ Modeling Job and File Definitions

Modeling File Definitions

- If there were existing record/field definitions before you specified the input source as a base model, the menu option Edit/Generate Sequence Numbers will become enabled.
 - If an extraction definition is selected, the menu option Edit/Sequence/Item Number will become enabled.
3. If, before **step 2**, you already created some or all of the record/field definitions for this input source, delete any file definitions that will not be used by every model user. Notice that the menu option Edit/Generate Sequence Numbers will become enabled.
 4. If, before **step 2**, you did not create all the record/field definitions needed for this input source, create them now. When you do so, they will be preceded by sequence/item numbers. A sequence/item number is a 6-digit sequence number and a 3-digit item number assigned automatically to each selection group/file definition combination when file modeling is being used. These numbers default to zero, so the sequence/item numbers will initially be (000000 / 000). You can identify selection groups because the first definition in each selection group begins at the left margin of the record. Additional definitions within the same selection group are slightly indented.
 5. Now assign unique sequence/item numbers to all record/field definitions in the input source. If you do this, when you define the model user, you will be able to add new and replacement definitions if the ones from the base model are not exactly what you need. The unique sequence/item numbers should be as follows:
 - The sequence number needs to be unique for each selection group and the same for every definition within the same selection group.
 - The item number is unique for each definition within a selection group. The first definition within a selection group must have an item number of 000.
 - Leave gaps between record sequence numbers in an input source that is a base model (where the option in the Base File Indicator dialog box is set to Yes) so that when you define model users, you can add selection groups between the base model definitions as needed. To create unique sequence/item numbers using gaps, select

Edit > Generate Sequence Numbers to open the Generate Sequence Numbers dialog box. The dialog box help shows you how to set the pattern for unique sequence/item numbers and provides examples of sequence number assignment using gaps. If you use the pattern **1, 5, 10**, the definitions will look like this.

000001/000	Set begin reference record with key value '1'
000005/000	Occurrence 1 of embedded key, value = 'ITEM1'
000005/001	Pick up value at position 35 with length 2, store in Internal Item STOCK ITEM1
000010/000	Occurrence 1 of embedded key, value = 'ITEM2'
000010/001	Pick up value at position 35 with length 2, store in Internal Item STOCK ITEM2

6. If you want to edit a sequence/item number individually, select the row where the sequence/item number appears and select **Edit > Sequence/Item Number** to open the Sequence/Item Number dialog box. Follow the instructions in the dialog box help.
7. When all definitions are complete and unique sequence/item numbers have been assigned, save the definitions.
8. You now need a list of your base model file definitions and their sequence numbers to use in setting up the model user definitions. This is because definitions from the **base model** will not appear in the **model user** input source. Make a screen print of the Input Source view. Or, if you prefer, you can run the List Definitions utility to print out the file definitions and their sequence numbers.

Specifying a Model User and Adding/Replacing File Definitions

1. Ensure that you have specified the base model as described in the previous section, “Specifying a Base Model for File Modeling” on page 273.
2. Create or open the job in which you want to use file modeling
3. Create the input source that will be a model user.
4. When you complete the File Organization dialog box, in the **Model File Name** and the **Qualifier** fields, enter the file name of the input source you want to use as a base model and its qualifier, if any.
5. Make sure the option in the Base File Indicator dialog box is set to **Do not use this file as a base**.
6. If you need to create any new definitions to add to or replace definitions from the base model, open the input source you just created that will be a model user. The base model definitions will not appear in the Record/Field Definitions folder.
7. To **replace one or more of the definitions from the base model**, do the following for each replacement definition:
 - a. Create the definition. Because you entered a model file name in step 4 on page 276, the system knows you are creating a model number, and the definition will have the default sequence/item number 000000 / 000.
 - b. Refer to the list of base model file definitions and their sequence numbers that you printed in “Specifying a Base Model for File Modeling” on page 273 to get the sequence/item number of the definition in the base model that you want to replace.
 - c. Select the replacement definition, then select **Edit > Sequence/Item Number** and assign the sequence/item number of the definition you want to replace.
8. To **add one or more selection groups** to the selection groups in the base, do the following for each additional selection group:
 - a. Create the additional file definitions for the selection group. Because you entered a model file name in step 4 on page 276, the system knows you are creating a model number, and definitions will have the default sequence/item number 000000 / 000.

- b. Refer to the list of base model file definitions and their sequence numbers that you printed in “Specifying a Base Model for File Modeling” on page 273. Then choose a unique sequence number for the selection group you are adding.

Note: If you need to insert a selection group between two existing selection groups from the base model (such as between #000001 and #000005), give it a record sequence number that is in between the numbers of the two existing groups (such as #000003). If the record sequence numbers in the base model does not have gaps that allow you to add your selection group in the correct sequence, go back and change the sequence numbers in the base. See Specifying a Base Model for File Modeling on page 273.

- c. Once you have determined an appropriate record sequence number for a selection group that you want to add, select all definitions in the new selection group, then select **Edit > Sequence/Item Number**. Enter the appropriate sequence number and click **OK**. The item numbers will be generated automatically.
9. To **add one or more new definitions to an existing selection group**, do the following for each definition:
 - a. Create the additional file definition. Because you entered a model file name in [step 4 on page 276](#), the system knows you are creating a model number, and the definition will have the default sequence/item number 000000 / 000.
 - b. Refer to the list of base model file definitions and their sequence numbers that you printed in [step 8 on page 275](#). You need to assign the same sequence number to the new definition as the sequence number in the existing selection group.
 - c. Once you have determined an appropriate record sequence number for the definition you want to add, select the definition, then select **Edit > Sequence/Item Number**. Enter the appropriate sequence number and item number and click **OK**.
 10. Create an internal or extended internal item for each field value defined in the base model. Since internal and extended internal items are categorized as job definitions rather than file definitions, they will not be merged in from the base model. If you want to model job definitions too, see “Modeling Job Definitions” on page 268

17 ■ Modeling Job and File Definitions

Modeling File Definitions

11. Finish defining the balancing job and save it.

When you update the definition database and run the model user job, a copy of the base model file definitions will be merged with the file definitions you in the model user to create a new set file definitions on the definition database.

Example of File Modeling

Suppose you work at corporate headquarters for a chain of 20 retail stores. Each month, you receive a report showing all of the stores' inventory counts. You want to generate an individual Control Report for each store.

This example shows how to set up model user file definitions for Store 2 using Store 1's definitions as a base model. The procedure for setting up model user file definitions for stores 3 through 20 would be identical.

STORE1/STEP1 is a job that balances store 1's inventory. The input source definitions for this job will be the base model.

The input source file for STORE1/STEP1 is a report listing inventory information for each of the 20 stores. Store 1 and Store 2's inventory information, as displayed in the Extract Window, is shown below.

STORE 1	UNITS	IN	STOCK
ITEM1			10
ITEM1			15
STORE 2	UNITS	IN	STOCK
ITEM1			13
ITEM2			21
ITEM3			34

Specifying the Base Model Definitions in the STORE1/STEP1 Job

Specify the base model file definitions in STORE1/STEP1 as follows:

1. Open STORE1/STEP1.
2. Open the STORE1 input source in Input Source View. In the Base File Indicator dialog box, set the **Do you want to use this file's definitions as a base to be modeled by other files?** field to Yes.

Store 1's file definitions, shown below, use the store's number (1) as a beginning reference. Then, they pick up two values: counts for stock item 1 and stock item 2.

000000/000	Set begin reference record with key value '1'
000000/000	Occurrence 1 of embedded key, value = 'ITEM1'
000000/000	Pick up value at position 35 with length 2, store in Internal Item STOCK ITEM1
000000/000	Occurrence 1 of embedded key, value = 'ITEM2'
000000/000	Pick up value at position 35 with length 2, store in Internal Item STOCK ITEM2

3. The default sequence/item numbers are all zeros. To create model user definitions, you will need to change these to unique sequence/item numbers. Select the file definitions and select **Edit > Generate Sequence Numbers** to open the Generate Sequence Numbers dialog box and assign a pattern so that all the sequence/item numbers will be unique, but there will be gaps. For example, if you assigned the pattern **1, 5, 10**, your sequence/item numbers would be as in the following example.

000001/000	Set begin reference record with key value '1'
000005/000	Occurrence 1 of embedded key, value = 'ITEM1'
000005/001	Pick up value at position 35 with length 2, store in Internal Item STOCK ITEM1
000010/000	Occurrence 1 of embedded key, value = 'ITEM2'
000010/001	Pick up value at position 35 with length 2, store in Internal Item STOCK ITEM2

4. Save the file definitions.
5. Record/field definitions from the base model will not be displayed in the model user job. But you will need to know these definitions and their sequence/item numbers when you set up the model user definitions. To do this, either 1) make a screen print of the definitions for your reference or 2) run the List Definitions utility to print out the definitions and their sequence numbers.

Specifying a Model User Job, STORE2/STEP1 and Adding/ Replacing File Definitions

1. Create a second job, STORE2/STEP1, that will balance store 2's inventory.
2. Create an input source named STORE 2. This input source will be a model user. When you complete the File Organization dialog box, in the **Model File Name** and the **Qualifier** fields, enter the file name of the input source you want to use as a **base model**, STORE1.
3. Make sure the option in the Base File Indicator dialog box is set to **Do not use this file as a base**.
4. Create a replacement file definition to replace the first definition from the base model. The base model uses 1 (for Store 1) as the beginning reference. The replacement definition uses 2 (for Store 2). Remember that definitions from the **base model** will not appear in the **model user** input source. Refer to the screen print you made of the definitions from the base model when deciding what new or replacement definitions you need to add.)
5. Add a new selection group that picks up an inventory count for stock item 3. Store 1 did not carry this item but store 2 does.
The new definitions will look like this:

000000/000	Set begin reference record with key value '2'
000000/000	Occurrence 1 of embedded key, value = 'ITEM3'
000000/000	Pick up value at position 35 with length 2, store in Internal Item STOCK ITEM3

6. Referring to your screen print of the base model definitions that you printed in step 5 on page 279, assign the same sequence number to the replacement definition as the corresponding beginning reference definition in the base model. Highlight the file definition, then select **Edit > Sequence/Item Number** to display the Sequence/Item Number dialog box and edit the sequence/item number.

7. Follow the same procedure to assign new sequence numbers to the two definitions in the new selection group that will pick up the inventory count for stock item 3. Because the stock item 3 record is after the other records that already have corresponding selection groups in the base model definitions, we give the new selection group a higher number than any of the existing groups. The renumbered definitions will look like this:

000001/000	Set begin reference record with key value '2'
000015/000	Occurrence 1 of embedded key, value = 'ITEM3'
000015/001	Pick up value at position 35 with length 2, store in Internal Item STOCK ITEM3

8. Create an internal item for each field value defined in the base model plus the additional field value defined in the model user.
9. Finish defining the STORE2/STEP1 job. Save your definitions, update the definition database, and run the job. This merges the definitions we want to add and replace for the STORE2 model user with the STORE1 base model definitions. This produces the following merged model user file definitions for STORE2 in the definition database:

000001/000	Set begin reference record with key value '2' FR NEW
000005/000	Occurrence 1 of embedded key, value = 'ITEM1'
000005/001	Pick up value at position 35 with length 2, store in Internal Item STOCK ITEM1
000010/000	Occurrence 1 of embedded key, value = 'ITEM2'
000010/001	Pick up value at position 35 with length 2, store in Internal Item STOCK ITEM2
000015/000	Occurrence 1 of embedded key, value = 'ITEM3'
000015/001	Pick up value at position 35 with length 2, store in Internal Item STOCK ITEM3

17 ■ Modeling Job and File Definitions

Modeling File Definitions

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 283
- “Format of Batch Definition Transaction Records” on page 284
- “Updating the Definition Database” on page 286
- “Purging Definitions” on page 287
- “The asterisk (*) indicates to select all file definitions for purging.” on page 287

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/Summary definitions in batch.

ACR/Summary definitions include job definitions, file definitions, table definitions, and history analysis definitions. For a description of each of these definition types, see “Definition Database (Default Name unidf.dat)” on page 18.

Job, file, table, and history analysis definitions are 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/Summary 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:

A ■ Using Batch Definition Transactions

Format of Batch Definition Transaction Records

- Any edit facility (such as Notepad) can be used to code information on 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, and history analysis definitions that you have created through ACR/Summary 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/Summary 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)

This is a character string identifying the specific transaction.

- **Job Definitions**—the 18-character Job ID.
- **History Analysis Definitions**—the 16-character history analysis ID and 2 blanks.
- **Table Definitions**—the 16-character Table name and 2 blanks.
- **File Definitions**—the 10-character File ID, the 6-digit Record Sequence Number, and 2 blanks.

Transaction Code (3 Positions)

This is 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

■ **S** are History Analysis Definitions; and

The second and third characters of the code identify the definition information categories. For example, J0 (Basic Job Information), J2 (Internal Item Information), J51 (Balancing Rules Description), and F1 (File Key Information).

Transaction Number (3 Positions)

This is a number identifying the specific transaction within its category, such as internal item number or field item number.

Transaction Sequence (1 Position)

This is 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)

This is a code specifying how the transaction is to be applied to the definition database. Valid transaction types are:

Code	Meaning
A	Add a new definition record to the database.
D	Delete an existing record 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 only the transaction key.
R	Replace an existing record in the database.
*	Purge a group of records from the database. This transaction type is valid only on the Basic Job Information, Basic File Information, Basic History Analysis Information, or Table Information records (the first record for a Transaction ID). When specified, it causes all definitions for the Transaction ID key on that transaction to be deleted from the Database. This transaction is ignored if no definition records exists for that Job ID key, File ID key, or Table Name in the Database. This transaction type may not be specified together with Delete or Replace transactions for the same Transaction ID key. For more information, see Processing Batch Definition Transaction Sets on page 286 below and Purging Definitions on page 287.

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, and history analysis 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 `uacupdt.bat` (Windows) or `uacupdt.sh` (UNIX or Linux).

First, the batch definition transactions are sorted by transaction key. Then, the `uacupdt.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 edits each individual transaction to ensure that valid interrelated information is entered in the proper columns. Each transaction is also printed in the transaction validation section of the Definition Database Update Report. For example, the Number of Histories

to Retain field in Basic Job Information must be numeric and greater than zero. If it is not, an error message will appear to the right and on the line below the transaction on the Transaction Validation Report.

If all transactions are valid, transaction processing is performed. Using the transaction type from each transaction's key, the uacupdt.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 Definition Database Update Report.

Once applied to the Definition database, any of the Application Interface modes can be used to test the definitions.

Purging Definitions

To purge definitions entirely 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 Record Layout
F0	*	Basic File Information Record Layout
S0	*	History Analysis Information Record Layout
T0	*	Cycle Table or Internal Translation Table Record Layout
X0	*	External Translation Table Information Record 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.

A ■ Using Batch Definition Transactions

Purging Definitions

Reports

This appendix explains how to work with reports and provides information on individual reports (or cross-references to this information). It contains the following sections:

- “Working with Reports on ACR/Summary Client” on page 289
- “Reports Table” on page 291

Working with Reports on ACR/Summary 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 165.

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.

Display and Formatting of Reports After Generation

Icons and View Menu Options

When a report is generated and open, icons display at the top of the 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 on ACR/Summary 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 Direct Message dialog box. E-mail notification must be set up as described in “Message Processing” on page 108.
- Reports from any ACR/Summary Client processing can be published to the Web using settings in 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 and Page Number	Source	
System Messages Report	Multiple Processes	page 292
Control Report	Balancing	page 292
Recap Report	Balancing	page 296
User Report	Balancing	page 195
Free-Form Report	Balancing	page 187
Trace Report	Balancing in Access Mode 6 or 7	page 298
External Translation Table Trace Report	Balancing when using an external translation table	page 221
History Analysis Report	History Analysis	page 247
Database Initialization Report	Database Initialization	page 302
Database History and Statistics Report	List Definitions Utility - All	page 303
Index of Definition Keys Report	List Definitions Utility - All	page 303
Detailed Job Definitions Report	List Definitions Utility - Job	page 305
Detailed File Definitions Report	List Definitions Utility - File	page 307
Detailed Table Definitions Report (Internal Translation Table)	List Definitions Utility - Internal Translation	page 309
Detailed Table Definitions Report (External Translation Table)	List Definitions Utility - External Translation	page 310
Detailed Spreadsheet Definitions	List Definitions Utility	page 311
Print Control Update Information	Print Control Update Information Utility	page 312
Build ACRUPDT Report	Database Update Utility	page 313
Definition Database Update Report	Update from Batch Transactions Utility	page 313
Definition Database Cross Reference Report	Cross Reference Internal Items Utility	page 316
Definition Database Cross Reference Report	Cross Reference File Definitions Utility	page 317
Database History and Statistics Report	List History Utility - All	page 318

B ■ Reports

System Messages Report - SYSOUT.RPT

Report and Page Number	Source	
Index of ACR Histories Report	List History Utility - All	page 318
Summary of ACR Histories Report	List History Utility - All	page 319
History Data Detail Report	List History Utility - All or Detail	page 319
Transaction Listing Report	Multiple Utilities	page 320
History Database Copy Report - Summary of Histories Transferred	Copy History Utility - Copy All Records, Copy Selected Records	page 321
History Database Copy Report - Summary of Histories Unloaded	Copy History Utility - Unload Records from a Work File	page 321
History Database Copy Report - Summary of Histories Reloaded	Copy History Utility - Reload Records to a Work File	page 322
History Database Reorg Report	Reorganize History Utility	page 322
Print Management Report	Print Management Utility	page 322

System Messages Report - SYSOUT.RPT

This report is automatically created when running balancing, an integrity check, a database initialization or update, and multiple utilities.

It shows the step completion code (return code) alone if there were no errors (return code 0000). Otherwise it shows each step completion code and the corresponding error message(s).

Sample System Messages Report - Successful Balancing Run

```
STEP COMPLETION CODE IS: +00000
```

Sample System Messages Report - Unsuccessful Balancing Run

```
#UFI990E: JOBXYZ /UPDATE / NO BALANCING INFORMATION IN  
INFOGIX DATA BASE (ACRDEFN)  
#UFI990E: UAC2000 TERMINATED DUE TO 00001 ERRORS  
STEP COMPLETIO CODE IS: +4000
```

Control Report - UNIACR.RPT

The Control Report is the primary output of an ACR/Summary balancing run. (A control report for a job run previously can be generated by running the Reprint Control Report utility.) It shows you the values balanced, rules

applied, results, messages to users, and return codes. Unless you specify otherwise, this report is automatically generated when balancing is completed.

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 165.
2. By default, one copy of the report will be generated using default settings whenever you run balancing. To customize this copy and optionally specify an additional copy at the job level, from the Control Task (Job) View, select **Basic Information > Print Control Reports** and complete the Job Output Options dialog box.
Make special note of the following:
 - If you choose the 80 columns (condensed) width, the report will not show the following:
 - JOB/STEP/QUAL, item number, cycle reference, cycle run date/time or program name for history items.
 - The formula for calculated items.
 - The formula for the standard rules.
 - Conditions or rule actions for conditional rules.
 - This applies only if you are using Release 4.0 and running a job created in a previous release that now includes extended internal items:
For pre-Release 4.0 jobs only, the default for the **Extended Internal Items** field is Do not print. If you want the report to include the extended internal items section, you need to change the setting for the **Extended Internal Items** field.
You must do this for every pre-Release 4.0 that now includes extended internal items.
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 for the Control Report field that appears in the User Options dialog box - XML Options tab.
4. At run-time you can optionally select the **Overrides** button on the Balancing dialog box to display the Override Options dialog box and set selected run-time overrides.

B ■ Reports

Control Report - UNIACR.RPT

Control Report Example and Content

Following is a sample of the report with an explanation of each section. The DDNAME for the report in your output will be UNIACR.

Example: Page 1 of 2

ACR/S releasenumbr		INFOGIX, INCORPORATED				PAGE 001								
DATE: yy/mm/dd		BALANCING CONTROL REPORT				COPYRIGHT INFOGIX, INC.								
TIME: 15:36:04		SAMPLE REPORT 1				DATA BASE VERSION: 0000037/0000011								
		JOB=REPORT STEP=SAMPLE QUAL=												
		CYCLE#=00000003 RUN#=000												
A. INTERNAL ITEMS:														
ITEM NO.	DESCRIPTION					INPUT	OUTPUT							
I-001.	TEXT	INTERNAL ITEM					'TEXT1	'						
I-002.	COUNT	INTERNAL ITEM					4,021							
I-003.	AMOUNT	INTERNAL ITEM					\$321.11							
I-004.	DATE	INTERNAL ITEM					05/12/09							
EXTENDED INTERNAL ITEMS:														
X-001.	COUNT	EXTENDED ITEM					1,234,567,890,123,456,789,012,341							
X-002.	AMOUNT	EXTENDED ITEM					\$1,234,567,890,123,456,789.001111							
X-003.	TEXT	EXTENDED ITEM					'LONG TEXT OF 39 BYTES LONG FIRST 111111'							
X-004.	DATE	EXTENDED ITEM					05/12/09							
B. HISTORY ITEMS:														
ITEM NO.	JOB	STEP	QUAL	ITEM	CYCLE	CYCLE#	RUN#	RUNDATE	RUNTIME	PROGRAM	INPUT	OUTPUT		
E-001.	HISTORY	TEXT	II	REPORT	SAMPLE	I-001	-002	00000001	000	08/06/10	15:29:52	UAC2000	'TEXT1	'
E-002.	HISTORY	COUNT	II	REPORT	SAMPLE	I-002	-002	00000001	000	08/06/10	15:29:52	UAC2000	4,021	
E-003.	HISTORY	AMOUNT	II	REPORT	SAMPLE	I-003	-002	00000001	000	08/06/10	15:29:52	UAC2000	\$321.11	
E-004.	HISTORY	DATE	II	REPORT	SAMPLE	I-004	-002	00000001	000	08/06/10	15:29:52	UAC2000	05/12/01	
E-009.	HISTORY	COUNT	EI	REPORT	SAMPLE	X-001	-002	00000001	000	08/06/10	15:29:52	UAC2000	1,234,567,890,123,456,789,012,341	
E-010.	HISTORY	AMOUNT	EI	REPORT	SAMPLE	X-002	-002	00000001	000	08/06/10	15:29:52	UAC2000	\$1,234,567,890,123,456,789.001111	
E-011.	HISTORY	TEXT	EI	REPORT	SAMPLE	X-003	-002	00000001	000	08/06/10	15:29:52	UAC2000	'LONG TEXT OF 39 BYTES LONG FIRST 111111'	
OUT OF BALANCE (RETURN CODE = 3001)														
INFOGIX DATABASE: "C:\Infogix\Summary32\unidf.da(UNIDF) / "C:\Infogix\Summary32\unihf.da(UNIHF)														
PROGRAM= UAC2000 LOAD LIBRARY NAME=releasenum.ACRSWB.LOADLIB JOB START DATE=yy/mm/dd5 JOB START TIME=12:55:38														

The report sections are as follows:

Header and Footer

The header displays the product release number, the report page number, the date and time of the report, the database version you are using (showing the number of times the definition database/history database has been updated), the job ID, and the cycle ID for this run.

The footer displays the definition and history databases, the program that generated the report, and the job start date and time.

Report Sections

The sections are identified by a letter and section name, as follows:

Note: Extended data values that do not fit on the line will be printed on the next line.

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.

B. History Items. This section lists each history item number (for example, E-001) along with the job ID from which the item is being retrieved, internal (or extended internal) item number in that job, relative cycle, cycle ID, run date, run time, name of the program that produced the report (UAC2000 is the balancing program), and the control value extracted.

Example: Page 2 of 2

ACR/S releasenumbr		INFOGIX, INCORPORATED				PAGE 002						
DATE: yy/mm/dd		BALANCING CONTROL REPORT				COPYRIGHT INFOGIX, INC.						
TIME: 15:36:04		RELEASE 4.0 SMOKETEST				DATA BASE VERSION: 0000037/0000011						
		JOB=REPORT STEP=SAMPLE QUAL=										
		CYCLE#=00000003 RUN#=000										
B. HISTORY ITEMS:												
ITEM NO.	JOB	STEP	QUAL	ITEM	CYCLE	CYCLE#	RUN#	RUNDATE	RUNTIME	PROGRAM	INPUT	OUTPUT
E-008.	HISTORY DATE EI											05/12/01
	REPORT SAMPLE			X-004	-002	00000001	000	08/06/10	15.29.52	UAC2000		
C. CALCULATED ITEMS:												
ITEM NO.	CALCULATION FORMULA										RESULT	CALC MESSAGE
C-001.	I-2 PLUS 100 COUNT (I-002) + (100)										4,121	
C-002.	I-3 PLUS 100 AMOUNT (I-003) + (100)										\$421.11	
C-005.	100 PLUS X-1 COUNT (100) + (X-001)										1,234,567,890,123,456,789,012,441	
C-006.	100 PLUS X-2 AMOUNT (100) + (X-002)										\$1,234,567,890,123,456,889.001111	
C-007.	LARGE MINUS SMALL REVENUE (I-003) - (100)											**BYPASSED**
D. BALANCING RULES & RESULTS:												
RULE NO.	BALANCING RULE										RESULT	ERROR/TOLERANCE
1.	X-1 EQUAL I-2 (X-001) EQ (I-002)										**OUT OF BALANCE**	
											\$1,234,567,890,123,456,789,008,320	
2.	X-3 NOT EQUAL I-1 (X-003) NE (I-001)										IN BALANCE	
3.	LARGE LARGEST, SMALL SMALLEST IF R-001 AND R-002										ACTION BYPASSED	
										IN-BALANCE THEN SET RETURN CODE 0001 CALCULATE C-007		
E. SPECIAL INSTRUCTIONS:												
RULE NO.	CODE	INSTRUCTIONS										
1.	3001	X-1 EQUAL I-2										
										OUT OF BALANCE	(RETURN CODE = 3001)	
										(UNIDF)	USR1.SUM.HIST	
										(UNIHF)		
INFOGIX DATABASE: USR1.SUM.DEFN										PROGRAM= UAC2000		
LOAD LIBRARY NAME=UNI.TAPESHIP.R40V0M00.LOADLIB										JOB START DATE=08/06/10		
										JOB START TIME=15:36:04		

C. Calculated Items. This section lists each calculated item by number and description along with the formula used and the result. Each side of the formula (LHS - Left Hand Side) and RHS (Right Hand Side) is enclosed in parentheses.

D. Balancing Rules & Results. This section lists each rule by number and description along with 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.

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).

Recap Report - UNIRCR.RPT

This report summarizes the results of multiple jobs for daily processing, for variable cycle processing, or for a complete system or subsystem.

In batch, the report is often run with the last job of the day. If a recap item is entered for each job that runs that day, the Recap Report can be scanned to find any problems that may have occurred.

Setting Up and Running the Recap Report

To set up the report through the graphical interface, specify the balancing histories to be included in the Recap Items dialog box. Specify printing in the Print Recap Report dialog box.

The report can then be generated by running balancing in file interface mode.

Report Header

The header displays the product release number, the report page number, the date and time of the report, the job ID, and the cycle ID for this run.

Report Fields

The fields in the following example can be interpreted as follows:

Field	Description
JOB, STEP, JQ	The job ID (job name, step name, and qualifier). Matches this information in the report header.
CYC	The relative cycle for the balancing job. The current cycle is represented by 000.
CYCLE#	The cycle number used for each job ID.
RUN#	The run number used for each JOB ID.
RUNDATE	The date when the job was run.

Field	Description
RUNTIME	The time when the job was run.
RUN STATUS	The status of the job. Valid values include: IN BALANCE. All rules were in balance. OUT OF BALANCE. One or more rules were out of balance. NO BALANCING. This job did not include balancing rules. *NOT IN FIILE*. This history record was not in the history database.
CODE	The return code for the balancing result.

Recap Report Example

ACR/S releasenumbr DATE: yy/mm/dd TIME: 09:24:56	INFOGIX, INC BALANCING RECAP REPORT THIS IS THE MONTHLY JOB JOB=BCAPCHK STEP=001 QUAL= CYCLE#=00960806 RUN#=000	COPYRIGHT INFOGIX, INC PAGE 001
--	---	------------------------------------

JOB	STEP	JQ	CYC	CYCLE#	RUN#	RUNDATE	RUNTIME	RUN STATUS	CODE
BCAPCHK	001		000	00920630	000	yy/mm/dd	12:41:52	IN BALANCE	0000
BCAPCHK	001		-001	00920625	000	yy/mm/dd	18:35:30	OUT OF BALANCE	2000
BCAPCOMM	001		000	00920630	000	yy/mm/dd	12:10:39	NO BALANCING	0000
BCAPUFDT	001		000	*****	****	*****	*****	*NOT IN FILE**	****

User Report - UNIUSR.RPT

The User Report is a customized report that prints any of the data that can be included in the Control Report in the order that you specify.

This report is documented in Chapter 9, “Setting Up User Reports” on page 195.

Free-Form Report - UNIUFR.RPT

The Free-Form Report is a custom report on a balancing job run and its results. Depending on your selections, the report can include the job ID, cycle number, return codes, messages, report title, internal items, extended internal items, history items, calculated items, rules, item descriptions, and free-form text.

This report is documented in Chapter 8, “Setting Up Free-Form Reports” on page 187.

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.

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)

B ■ Reports

History Analysis Report - UNISSR2.RPT

Source	Reformatted Area is	Format	
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.

History Analysis Report - UNISSR2.RPT

This report is generated by running a history analysis job. For more information, see Chapter 15, "Using History Analysis" on page 247.

Database Initialization Report - UNIIR.RPT

This report is generated by running Database Initialization or Initialize History. It lists the initialization transactions and any validation messages and presents a summary of the transactions.

```
ACR/S releasenumbr          INFOGIX, INCORPORATED          COPYRIGHT INFOGIX, INC
DATE: yy/mm/dd              DATABASE INITIALIZATION REPORT          PAGE    001
TIME: 16:45:34

-----INPUT TRANSACTION LISTING-----
TRANS
NO.          T R A N S A C T I O N          V A L I D A T I O N M E S S A G E S
-----1-----2-----3-----4-----5-----6-----7-----8-----
HFS  0000001000000030
HFDSN  "/home/userid/unidf.dat"
DFDSN  "/home/userid/unihf.dat"

-----DATABASE INITIALIZATION SUMMARY-----
DEFINITION DSN          = "/home/userid/unidf.dat"
HISTORY DSN            = "/home/userid/unihf.dat"
NUMBER OF JOB ID KEYS  = 10
NUMBER OF ACR HISTORIES = 30
```

Database History and Statistics Report - UNIDLR.RPT

Generated by the List Definitions utility, this portion of the Definition Database Listing shows detailed information about the database including date and time of initialization and last update and number of records by type.

ACR/S releasenumber	XYZ COMPANY	COPYRIGHT INFOGIX, INC
DATE: yy/mm/dd	DEFINITION DATA BASE LISTING	DATA BASE VERSION: 0000078/*****
TIME: 13:36:15	DSN="/home/userid/unidf.dat"	
	DATA BASE HISTORY AND STATISTICS	
	DATE AND TIME OF INITIALIZATION:	yy/mm/dd AT 16:23:22
	DATE AND TIME OF LAST UPDATE:	yy/mm/dd AT 16:41:15
	ORIGINAL ESTIMATED NUMBER OF UNIQUE JOB IDS:	11
	ORIGINAL AVERAGE NUMBER OF HISTORIES PER JOB IDS:	22
	NUMBER OF JOB ID KEYS:	
	MAXIMUM:	32
	USED:	1
	AVAILABLE:	31
	NUMBER OF ACR HISTORIES (INCLUDING ADJUSTMENTS):	242
	MAXIMUM:	3
	USED:	3
	AVAILABLE:	239

Index of Definition Keys Report - UNIDLR.RPT_T

Generated by the List Definitions utility, this portion of the Definition Database Listing provides a list of definition keys by type.

B ■ Reports

Index of Definition Keys Report - UNIDLR.RPTT

DEFINITION KEY	TYPE	COMMENTS
ACCESS3 STEP06	JOB ID	
ACCMD361 000000	FILE-ID	
BOOKCLUBVERIFY	JOB ID	
BOOKCOST 000000	FILE-ID	
CARENTL COMPARE	JOB ID	
CARENTL1 000000	FILE-ID	
CARENTL2 000000	FILE-ID	
CLAIMS 000000	FILE-ID	
CLAIMS STEP1	JOB ID	
CONDRULE 000000	FILE-ID	
CURRENCY	TABLE-ID	TRANSLATION TABLE
DAILY 000000	FILE-ID	
DOMESTICSALES	JOB ID	
IFTHEN STEP1	JOB ID	
ILLINOIS 000000	FILE-ID	
JOBXYZ UPDATE	JOB ID	
JOBXYZ UPDATE TO	JOB ID	
MONTH CONVERSION	TABLE-ID	TRANSLATION TABLE
MONTHEND	TABLE-ID	CYCLE TABLE
OFFICE01 000000	FILE-ID	
OFFICE0101000000	FILE-ID	
OHIO 000000	FILE-ID	
RESULTS	SPREADSHEET-ID	
SALES DAILY	JOB ID	
SAMPLE UPDATE	JOB ID	
STATE CODE/ABBR	TABLE-ID	EXT. TRANSLATION TABLE
STATEDEF 000000	FILE-ID	
STATEOH SALES	JOB ID	
STATES SALES	JOB ID	
STATESILSALES	JOB ID	
SUMM0501 000000	FILE-ID	
TOTALS	SPREADSHEET-ID	
TOTALS STEP1	JOB ID	
TOTLCOST 000000	FILE-ID	
TOTSALQ3 000000	FILE-ID	
TRANSLATE STATES	TABLE-ID	EXT. TRANSLATION TABLE
UPDTRPT 000000	FILE-ID	
UPDT0501 000000	FILE-ID	

Detailed Job Definitions - UNIDLR.RPT

Generated by the List Definitions utility, this portion of the Definition Database Listing provides a formatted list of job definitions when the List Type selected is Jobs or All

Detailed Job Definitions - Page 1 of 2

ACR/S releasenumbr	XYZ COMPANY	COPYRIGHT INFOGIX, INC
DATE: yy/mm/dd	INFOGIX DATA BASE UTILITY	PAGE 001
TIME: 13:50:20	DEFINITION DATA BASE LISTING	DATA BASE VERSION: 0000004/*****
	DSN="/home/userid/unidf.dat"	
	DETAILED DEFINITIONS	
JOB ID: OVERPUNCTEST1		
A. ALTERNATE JOB ID KEYS:		
NONE		
B. BASIC INFORMATION:		
JOB COMMENT INFORMATION:		
001 This job generates multiple reports.		
PRODUCT:	ACR/S	PROBLEMS WITH OVERPUNCH
SYSTEM NAME:		
NUMBER OF HISTORIES RETAINED:	1	
LAST UPDATE BY:	SAMPUSR	DATE: yy/mm/dd TIME: 15:11:54 RUN OPTIONS:
RUN OPTIONS:		
STORE RUN HISTORY:	YES	
PRINT ACR:	YES	
PRINT RECAP:	NO	
SET RETURN CODE:	NO	
PRINT SECOND COPY OF ACR:	NO	
MODEL PROCESSING:	NO	
FILE CONTROL DEFINITIONS:	NO	
USER EXIT ACCESS:	NO	
CYCLE TABLE ID:		
CYCLE REFERENCE JOB ID KEY:		
CYCLE INCREMENT:	0	
AUTOMATIC CYCLE OPTION:		
AUTO RERUN REQUIRED:	USE SYSTEM DEFAULT	
HISTORY AUDIT TRAIL:	USE SYSTEM DEFAULT	
TRUNCATE/ROUND DECIMALS:	ROUND	
PRINT 22 CHARACTER NUMERIC:	NONE	
CONTROL REPORT XML EXPORT OPT:	DEFAULT TO USER OPTIONS	
OUTPUT OPTIONS:		
ENHANCEMENTS ACTIVATED:	YES	
STORE BALANCING EXTRACT FILE:	YES	
REPORT SIZE - COPY 1:	132	
REPORT SIZE - COPY 2:	132	
SECTION PRINT OPTIONS:		
INTERNAL ITEMS - COPY 1	:	PRINT SECTION
INTERNAL ITEMS - COPY 2	:	PRINT SECTION
INTERNAL ITEMS - EXTRACT	:	DO NOT STORE EXTRACT RECORD
EXTENDED INTERNAL ITEMS - COPY 1	:	PRINT SECTION
EXTENDED INTERNAL ITEMS - COPY 2	:	DO NOT PRINT THE SECTION
EXTENDED INTERNAL ITEMS - EXTRACT	:	DO NOT STORE EXTRACT RECORD
HISTORY ITEMS - COPY 1	:	PRINT SECTION
HISTORY ITEMS - COPY 2	:	PRINT SECTION
HISTORY ITEMS - EXTRACT	:	DO NOT STORE EXTRACT RECORD
CALCULATED ITEMS - COPY 1	:	PRINT SECTION
CALCULATED ITEMS - COPY 2	:	PRINT SECTION
CALCULATED ITEMS - EXTRACT	:	DO NOT STORE EXTRACT RECORD
BALANCING RULES - COPY 1	:	PRINT SECTION
BALANCING RULES - COPY 2	:	PRINT SECTION
BALANCING RULES - EXTRACT	:	DO NOT STORE EXTRACT RECORD

B ■ Reports

Detailed Job Definitions - UNIDLR.RPT

Detailed Job Definitions - Page 2 of 2

ACR/S releasenumbr		XYZ COMPANY		COPYRIGHT INFOGIX, INC													
DATE: yy/mm/dd		INFOGIX DATA BASE UTILITY		PAGE 002													
TIME: 13:50:20		DEFINITION DATA BASE LISTING		DATA BASE VERSION: 0000004/*****													
		DSN=USERID.SUM.DEFN		DETAILED DEFINITIONS													
JOB ID: OVERPUNCTEST1																	
B. BASIC INFORMATION:																	
SPECIAL INSTRUCTIONS - EXTRACT : DO NOT STORE EXTRACT RECORD																	
DESCRIPTION PRINT OPTIONS:																	
HISTORY ITEMS - COPY 1 DESC. : PRINT HISTORY ITEM DESCRIPTION																	
CALCULATED ITEMS - COPY 1 DESC. : PRINT DESCRIPTION AND FORMULA																	
BALANCING RULES - COPY 1 DESC. : PRINT DESCRIPTION AND FORMULA																	
HISTORY ITEMS - COPY 2 DESC. : PRINT HISTORY ITEM DESCRIPTION																	
CALCULATED ITEMS - COPY 2 DESC. : PRINT DESCRIPTION AND FORMULA																	
BALANCING RULES - COPY 2 DESC. : PRINT DESCRIPTION AND FORMULA																	
HISTORY ITEMS - EXTRACT DESC. : STORE FIRST RECORD LAYOUT																	
CALCULATED ITEMS - EXTRACT DESC. : STORE FIRST RECORD LAYOUT																	
BALANCING RULES - EXTRACT DESC. : STORE FIRST RECORD LAYOUT																	
C. INTERNAL ITEMS:																	
ITEM	DESCRIPTION			L/R/C	FORMAT	DEC	CI#	REQD	COM								
I-001	RIGHT ALIGNED ZONED DECIMAL AS FF4 SHOULD BE 1000			LEFT	COUNT	0		NO									
I-002	RIGHT ALIGNED ZONED DECIMAL AS FF41 SHOULD BE 10001			LEFT	COUNT	0		NO									
EXTENDED INTERNAL ITEMS:																	
ITEM	DESCRIPTION				FORMAT	DEC	CI#	REQD	COM								
X-001	EXT RT ALIGNED ZONED DECIMAL AS FF4 SHOULD BE 1,000				COUNT	0		NO									
D. EXTRACTION VARIABLES:																	
ITEM	DESCRIPTION				DEC												
V-101	DEC 4								4								
E. HISTORY ITEMS:																	
ITEM	JOB	STEP	QUAL	ITEM	CYC	CYC	RECENT	REQD	EQ	IND	JOB	STEP	QUAL	CYC	RECENT	CYC	FMT
E-001	CYCLE12	STEP1		I-002	-002	NO	NO	YES	NO	NO							
F. CALCULATED ITEMS:																	
ITEM	CALCULATION FORMULA				FORMAT	DEC	REQD	COM	COND	CYC	FMT						
C-001	ITEMS X-001 + I-002				COUNT	0	NO		N								
C-002	1 INTO @CYCLE				DATE	0	NO		N	MMDCCYY							
G. BALANCING RULES:																	
RULE	TYPE	BALANCING FORMULA			CODE	ACTION	FORMAT	TOL.	TOL.	VALUE							
R-001	ACT	ITEMS X-001 EQ XV-101						0000	NONE	TEXT NONE							
COM: CALC ITEM:001																	
J. USER REPORT ITEMS:																	
PRINT OPTION: (Z) IF IN BALANCE																	
XML OPTION: (9) DEFAULT TO USER OPT																	
FIXED SPACING: (Y) YES SPACING WIDTH: 5																	
WRAP INDICATOR: () TRUNCATE EXTENDED ITEM																	
USER REPORT TITLE																	
		COLUMN HEADING 1	COLUMN HEADING 2	COLUMN HEADING 3	COLUMN HEADING 4												
SCR 874	ALIGNMENT OF DECIMALS	COL 1 NUMBER	COLUMN SEPARATOR	CHARS													
ITEM DESCRIPTION		ITEM 1	ITEM 2	ITEM 3	ITEM 4	1234											
COLUMN ONE		I-001															
COLUMN TWO		I-002															
USER REPORT PAGE HEADERS																	
COLUMN	COLUMN	ITEM TYPE	ITEM#	WIDTH	POS.	LITERAL	HEADING										
							OPTION										
000	(L)	LITERAL		18	1	This is Literal 1:	CENTER										
000	(I)	INTRNL ITEM	1	20	20		CENTER										

Detailed File Definitions - UNIDLR.RPT

Generated by the List Definitions utility, this portion of the Definition Database Listing provides detailed information on each file ID processed when the List Type selected is Files or All. (The full heading is shown only for page 1.)

```

ACR/S releasenumber
DATE: yy/mm/dd
TIME: 15:14:39
INFOGIX, INCORPORATED
INFOGIX DATA BASE UTILITY
DEFINITION DATA BASE LISTING
DSN=R99UC47.SUM.DEFN
DETAILED FILE DEFINITIONS
COPYRIGHT INFOGIX, INC. 20159
PAGE 001
DATA BASE VERSION: 0000449/*****

FILE ID:          UGRPT1F
FILE DESCRIPTION: FLYER REPORT
A. FILE INFORMATION:

ACCESS MODE:          START OF DATA SET
DATA SET ORGANIZATION: SEQUENTIAL FILE/REPORT
NUMBER OF RECORDS SCANNED: 50
DD NAME FOR HARD COPY:
FILE MODEL - FILE ID:
LAST UPDATE BY:
DDF INFORMATION:

BASE MODEL: NO
R99UC47 DATE: 15/06/10 TIME: 10.20.49
USE DDF FIXED DATA DELIM 1 DELIM 2 DELIM 3 DELIM 4 DELIM 5 ENCLS

REFERENCES:
JOB STEP QUAL
UGRPT1 SAMPLE

C. RECORD INFORMATION:
REC ID TYPE OCC POS LEN IND DDF VALUE
000001 EMBEDDED 1 1- 1 8 NNN FF48379
REC ID FLD# POS LEN DDF FORMAT DEC ITEM IN AREA OUT AREA FIELD TYPE SIGN CYC/TIME FMT TRAN TABLE / OPTION
000001 1 97 8 ALPHA NO 1 INF FLD INT ITEM TEXT
000001 2 89 4 NUMERIC NO 2 INF FLD INT ITEM COUNT
000001 3 41 10 NUMERIC NO 3 INF FLD INT ITEM AMOUNT
000001 4 52 8 NUMERIC NO 4 INF FLD INT ITEM DATE
000001 5 61 8 NUMERIC NO 5 INF FLD INT ITEM TIME HMMSS
000001 6 70 8 NUMERIC NO 7 INF FLD INT ITEM COUNT
000001 7 11 30 ALPHA NO 1 INF FLD INT XTND TEXT
000001 8 81 3 ALPHA NO 8 INF FLD INT ITEM TEXT
000001 9 81 3 NUMERIC NO 9 INF FLD INT ITEM TEXT FLYEROFFICETABLE YES
    
```

Note: In the following Total/Tally/Hash example, the DEC (Extract Decimal Indicator) column applies only to total (not hash or tally) records.

```

FILE ID:          EXTVAR4G
FILE DESCRIPTION: SAMPLE FILE 5
A. FILE INFORMATION:
ACCESS MODE:          TOTAL/TALLY/HASH
DATA SET ORGANIZATION: SEQUENTIAL FILE/REPORT
FILE MODEL - FILE ID:
REFERENCES:          PASS EXT VAR
JOB STEP QUAL
EXTVAR4 AMSEV

C. RECORD INFORMATION:
REC ID TYPE OCC POS LEN IND VALUE
000001 HASH 28 8 N (XV-201)\
FLD# POS LEN FORMAT DEC ITEM IN AREA OUT AREA FIELD TYPE SIGN CYC FMT/PART TRAN TABLE / OPTION
1 101 EV XTND
000002 TOTAL 28 8 N (XV-201)
FLD# POS LEN FORMAT DEC ITEM IN AREA OUT AREA FIELD TYPE SIGN CYC FMT/PART TRAN TABLE / OPTION
2 10 4 NUMERIC YES 102 EV XTND R
000003 TALLY 28 8 N (XV-201)
FLD# POS LEN FORMAT DEC ITEM IN AREA OUT AREA FIELD TYPE SIGN CYC FMT/PART TRAN TABLE / OPTION
3 103 EV XTND
    
```

B ■ Reports

Detailed File Definitions - UNIDLR.RPT

ACR/S R40V4M00		INFOGIX, INCORPORATED		COPYRIGHT INFOGIX, INC. 2012		PAGE 003									
DATE: yy/mm/dd		INFOGIX DATA BASE UTILITY		INFOGIX DATA BASE LISTING		PAGE 001									
TIME: 15:02:01		DEFINITION DATA BASE LISTING		DSN=R99UC47.SUM.DEFN		DATA BASE VERSION: 0000978/*****									
FILE ID: SAMPLE		ACCESS MODE: 6		RESET EXT VAR											
FILE DESCRIPTION:		SEQUENTIAL FILE/REPORT		BASE MODEL: NO											
A. FILE INFORMATION:		USE DDF		FIXED DATA		DELIM 1 DELIM 2 DELIM 3 DELIM 4 DELIM 5 ENCLS									
ACCESS MODE:		NO													
DATA SET ORGANIZATION:		JOB		STEP		QUAL									
DD NAME FOR HARD COPY:		XSELECT1		EV											
FILE MODEL - FILE ID:		INP		LVL		LOG REL SEL									
FILE TRANSLATION TABLE:		TYP		NBR		OPR NBR KEY SAME									
DDF INFORMATION:		EXT		VAR		VALUE									
REFERENCES:		DDF													
REC-ID: 001001	SELECTION TYPE	SEQ	POS	LEN	FORMAT	TYPE	EXT	INP	LVL	LOG	REL	SEL	EXT	VALUE	DDF
	= TO VALUE 1	001	1	6	ALPHA	TEXT	NO	INP	1	AND				NUMBER	
	WITHIN RANGE	002	10	80	ALPHA	TEXT	YES	INP	1	AND				**REFER VAL BELOW**	
	SELECT FROM VALUE:	100222333444555666777888999000+ONE MUSEUM PARK EAST													
	SELECT TO VALUE:	150222333444555666777888999000+ONE MUSEUM PARK EAST													
	SELECT FROM VALUE:	200333444555666777888999000111+ONE MUSEUM PARK EAST													
	SELECT TO VALUE:	250333444555666777888999000111+ONE MUSEUM PARK EAST													
	= TO VALUE 1	003	10	31	NUMERIC	CNT	YES	INP	1	AND				**REFER VAL BELOW**	
	SELECT VALUE:	222333444555666777888999000111+													
	SELECT VALUE:	111222333444555666777888999000+													
	DETAILS:	001	10	80	ALPHA	TEXT	NO		XV	001	L	DTL			
		002	10	31	NUMERIC	SUM	NO		XV	002	ADD				
REC-ID: 002001	SELECTION TYPE	SEQ	POS	LEN	FORMAT	TYPE	EXT	INP	LVL	LOG	REL	SEL	EXT	VALUE	DDF
	= TO VALUE 1	001	1	6	ALPHA	TEXT	NO	INP	1	AND				NUMBER	
	VALUE RANGE	002	10	80	ALPHA	TEXT	YES	INP	1	AND				**REFER VAL BELOW**	
	SELECT FROM VALUE:	100222333444555666777888999000+ONE MUSEUM PARK EAST													
	SELECT TO VALUE:	150222333444555666777888999000+ONE MUSEUM PARK EAST													
	SELECT FROM VALUE:	200333444555666777888999000111+ONE MUSEUM PARK EAST													
	SELECT TO VALUE:	250333444555666777888999000111+ONE MUSEUM PARK EAST													
	= TO VALUE 1	003	10	31	NUMERIC	CNT	YES	INP	1	AND				**REFER VAL BELOW**	
	SELECT VALUE:	222333444555666777888999000111+													
	SELECT VALUE:	111222333444555666777888999000+													
	= TO VALUE 1	004	0	0	ALPHA	TEXT	YES	XV	1	AND			001	**REFER VAL BELOW**	
	SELECT VALUE:	111222333444555666777888999000+ONE MUSEUM PARK EAST													
	SELECT VALUE:	222333444555666777888999000111+ONE MUSEUM PARK EAST													
	> OR =	005	1	16	PACKED	CNT	YES	XV	1	AND			002	**REFER VAL BELOW**	
	SELECT VALUE:	111222333444555666777888999000+													
	DETAILS:	001	31	10	NUMERIC	SUM	NO							I-001	ADD

Detailed Table Definitions - External Translation Table - UNIDLR.RPT

Generated by the List Definitions utility, this portion of the Definition Database Listing provides detailed information on each external translation table processed when the type of definition selected is External Tables.

ACR/D releasenumbr		XYZ COMPANY		COPYRIGHT INFOGIX, INC		PAGE 001	
DATE: YY/MM/DD		INFOGIX DATA BASE UTILITY		DEFINITION DATA BASE LISTING		DATA BASE VERSION: 0002185/*****	
TIME: 13:36:15		DSN=USERID.SUM.DEFN		DETAILED TABLE DEFINITIONS			
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:				DATE: yy/mm/dd			
				TIME: 15:11:54			
				USE DDF			
				FIXED DATA			
				DELIM 1			
				DELIM 2			
				DELIM 3			
				DELIM 4			
				DELIM 5			
				ENCL5			
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	EQUAL TO			
2	ASSIGN:	C001	P001				
F. TABLE BUILD RULES:							
SELECTION CRITERIA:	RULE	POS	LEN	DDF	FORMAT	LH OPR	SELECTION TYP
	1	10	12		NUMERIC	INP	RH OPR
							EXT
							BEGIN
							LOCATE
							LITERAL
							111222333444
							222333444555
							333444555666
							LITERAL
							SEE BELOW
SELECTION CRITERIA:	RULE	POS	LEN	DDF	FORMAT	LH OPR	SELECTION TYP
	2	10	31		NUMERIC	INP	RH OPR
							EXT
							BEGIN
							LOCATE
							LITERAL
							SEE BELOW
SELECT VALUE:	111222333444555666777888999000+						
SELECT VALUE:	222333444555666777888999000111+						
SELECT VALUE:	333444555666777888999000111222+						
COLUMN ASSIGNMENT:	RULE	POS	LEN	DDF	FORMAT	TYPE	DATE FMT
	3	10	31		NUMERIC	COUNT	LH OPERAND
							RH OPERAND
							XV001
							LITERAL
COLUMN ASSIGNMENT:	RULE	POS	LEN	DDF	FORMAT	TYPE	DATE FMT
	4	10	31		NUMERIC	COUNT	LH OPERAND
							RH OPERAND
							XV002
							LITERAL
SELECTION CRITERIA:	RULE	POS	LEN	DDF	FORMAT	LH OPR	SELECTION TYP
	5	10	12		NUMERIC	INP	RH OPR
							EXT
							BEGIN
							LOCATE
							LITERAL
							111222333444
							LITERAL
SELECTION CRITERIA:	RULE	POS	LEN	DDF	FORMAT	LH OPR	SELECTION TYP
	6	10	31		NUMERIC	INP	RH OPR
							EXT
							BEGIN
							LOCATE
							LITERAL
							SEE BELOW
SELECT VALUE:	111222333444555666777888999000+						
SELECTION CRITERIA:	RULE	POS	LEN	DDF	FORMAT	LH OPR	SELECTION TYP
	7	1	16		PACKED	XV001	RH OPR
							EXT
							BEGIN
							LOCATE
							LITERAL
							SEE BELOW
SELECT VALUE:	111222333444555666777888999000+						
COLUMN ASSIGNMENT:	RULE	POS	LEN	DDF	FORMAT	TYPE	DATE FMT
	8	1	16		PACKED	XV001	LH OPERAND
							RH OPERAND
							LITERAL
COLUMN ASSIGNMENT:	RULE	POS	LEN	DDF	FORMAT	TYPE	DATE FMT
	9	10	6		NUMERIC	COUNT	LH OPERAND
							RH OPERAND
							C001
							LITERAL
							10
							31
							NUMERIC
							COUNT
							C002

Detailed Spreadsheet (History Analysis Definitions) Report - UNIDLR.RPT

Detailed Spreadsheet (History Analysis Definitions) Report - UNIDLR.RPT

Generated by the List Definitions utility, this portion of the Definition Database Listing provides detailed information on history analysis definitions when History Analysis or All is selected.

```

ACR/S releasenumber          XYZ COMPANY          COPYRIGHT INFOGIX, INC.
DATE: yy/mm/dd              INFOGIX DATA BASE UTILITY          PAGE 001
TIME: 14:38:39              DEFINITION DATA BASE LISTING      DATA BASE VERSION: 0000095/*****
                               DSN="/home/userid/unidf.dat"
                               DETAILED SPREADSHEET DEFINITIONS

z\
SPREADSHEET ID: RESULTS
1. SPREADSHEET INFORMATION.

DESCRIPTION:                SALES RESULTS FROM PREVIOUS MONTH
FORMAT:                      JOB ID
ROUND SUM RESULTS:          NO
CYCLE TABLE ID:
CYCLE REFERENCE JSA:
VARIABLE CYC REFERENCE JSA:
NO. OF COLUMNS PER PAGE:   4
LAST UPDATE BY:            SAMPUSR          DATE: yy/mm/dd          TIME: 15:11:54
DECIMAL PROCESSING TYPE:    NEW

ACR/S releasenumber          XYZ COMPANY          COPYRIGHT INFOGIX, INC.
DATE: yy/mm/dd              INFOGIX DATA BASE UTILITY          PAGE 001
TIME: 14:38:39              DEFINITION DATA BASE LISTING      DATA BASE VERSION: 0000095/*****
                               DSN="/home/userid/unidf.dat"
                               DETAILED SPREADSHEET DEFINITIONS

SPREADSHEET ID: TOTALS
1. SPREADSHEET INFORMATION.

DESCRIPTION:                QUARTERLY SALES REPORT
FORMAT:                      INTERNAL ITEM
ROUND SUM RESULTS:          NO
CYCLE TABLE ID:
CYCLE REFERENCE JSA:
VARIABLE CYC REFERENCE JSA:
NO. OF COLUMNS PER PAGE:   4
DECIMAL PROCESSING TYPE:    NEW
2. COLUMN DEFINITIONS.

COL#      DESCRIPTION              COL TYPE  OPT.  DETAIL COL. INFO  SUM COL INFO  HDG  MOST  HDG
                INT. ITEM NBR  TYP BEG END DEC  TYP  REC.  JUST
-----
1 TOTAL     BAKERY          SALES    DETAIL PRINT      001          TEXT NO  CENTR
2 TOTAL     DELI            SALES    DETAIL PRINT      002          TEXT NO  CENTR
3 TOTAL     PRODUCE         SALES    DETAIL PRINT      003          TEXT NO  CENTR
4 TOTAL     MONTHLY         SALES    DETAIL PRINT      004          TEXT NO  CENTR
3. ROW DEFINITIONS.

ROW#      DESCRIPTION              ROW TYPE  OPT.  -----DETAIL ROW INFO-----  SUM ROW INFO  HDG  MOST  HDG
                JOB ID KEY  CYCLE#  TYP BEG END DEC  TYP  REC.  JUST
-----
1         DETAIL PRINT  TOTALS STEP1  -003          TEXT NO  LEFT
2         DETAIL PRINT  TOTALS STEP1  -002          TEXT NO  LEFT
3         DETAIL PRINT  TOTALS STEP1  -001          TEXT NO  LEFT
4         DETAIL PRINT  TOTALS STEP1  +000          TEXT NO  LEFT
4. ANALYSIS RULES.
----ANALYSIS WINDOW----      -----ANALYSIS PARAMETERS-----
RULE      COLUMN #  ROW #      BASE      ---TOLERANCE---      RELATION  LEGEND  RETURN  FORMAT
        BEGIN END  BEGIN END  COL/ROW #  TYPE  QUANTITY  DEC  TYPE  CODE  CODE  IND
-----
1         1      3      2      4      ROW  1  NONE      0      0      LE      1      3010  A

ACR/S releasenumber          XYZ COMPANY          COPYRIGHT INFOGIX, INC.
DATE: yy/mm/dd              INFOGIX DATA BASE UTILITY          PAGE 001
TIME: 14:38:39              DEFINITION DATA BASE LISTING      DATA BASE VERSION: 0000095/*****
                               DSN="/home/userid/unidf.dat"
                               DETAILED SPREADSHEET DEFINITIONS

SPREADSHEET ID: TOTALS
5. LEGEND CODE DEFINITIONS.
LEG#      CODE      L E G E N D   T E X T
-----
1         1      SALES LOWER THAN LAST QUARTER'S AVERAGE
    
```

B ■ Reports

Print Control Update Information Report - UNIDLRCU.RPT

Print Control Update Information Report - UNIDLRCU.RPT

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	TYPE	LAST UPDATE BY	DATE	TIME
DDF2	AM5	JOB			
DDF3	AM1EMB	JOB			
DDF4	AM1REF	JOB			
DDF5	AM4REL	JOB			
DDF5	AM4REL	USER REPORT	<i>UserId</i>	yy/mm/dd	12:56:33
DDF2F1	000000	FILE			
DDF3F1	000000	FILE			
DDF4F1	000000	FILE			
DDF5F1	000000	FILE			
ABC		CYCLE TABLE			
EXTBTXEQ		EXT TRAN TABLE			
JIRA3090ETTA		EXT TRAN TABLE			
JIRA3090ETTB		EXT TRAN TABLE			
NAME TRANSLATION		INT TRAN TABLE			
UBS-DELIM-DATES		EXT TRAN TABLE			
YEARCNVINUM		EXT TRAN TABLE			
COMCALLSTEP1		HISTORY ANALYSIS			
DDF		HISTORY ANALYSIS	<i>UserId</i>	yy/mm/dd	12:56:33
HA1397		HISTORY ANALYSIS			
SCR2180R1		HISTORY ANALYSIS			
VOLANALAR1		HA REPORT			

Build ACRUPDT Report - UNIDBR.RPT

Generated by the Build Batch Transactions utility, this report shows each batch transaction record processed. The sample below has been shortened to show only the file definition transaction records (F records).

ACR/S releasenumbr		XYZ COMPANY			COPYRIGHT INFOGIX, INC				
DATE: yy/mm/dd		INFOGIX DATA BASE UTILITY							
TIME: 11:21:27		BUILD ACRUPDT REPORT			DATA BASE VERSION: 0000086/*****				
		DSN=USERID.SUM.DEFN							
TRANS NO.	RECORD TYPE	TRANSACTION							
		1	2	3	4	5	6	7	8
00001	FILE-ID	SAMPLE	000000	F0	0001*				
00002		SAMPLE	000000	F0	0001A1PFA00100	0000000		YN	
00003		SAMPLE	000000	F0	0002A		THIS IS A SAMPLE		
00004		SAMPLE	000001	F2	0001A3LOCATION:		00100	N	
00005		SAMPLE	000001	F2	0002A3		0002090002		
00006		SAMPLE	000001	F3	0011A0012041001C		A		N
00007		SAMPLE	000002	F2	0001A3-		00100		N
00008		SAMPLE	000002	F2	0002A3		0014010014		
00009		SAMPLE	000002	F3	0011A0011122001T		A		N
00010		SAMPLE	000002	F3	0021A0024292002T		A		N
00011		SAMPLE	000002	F3	0031A0071121003C		A		N
00012		SAMPLE	000003	F2	0001A3LOCATION:		00200	N	
00013		SAMPLE	000003	F2	0002A3		0002090002		
00014		SAMPLE	000003	F3	0011A0012041002C		A		N
00015		SAMPLE	000004	F2	0001A3-		00200		N
00016		SAMPLE	000004	F2	0002A3		0014010014		
00017		SAMPLE	000004	F3	0011A0011122006T		A		N
00018		SAMPLE	000004	F3	0021A0024282003T		A		N
00019		SAMPLE	000004	F3	0031A0070141005A		A		N
00020		SAMPLE	000005	F2	0001A3LOCATION:		00300	N	
00021		SAMPLE	000005	F2	0002A3		0002090002		
00022		SAMPLE	000005	F3	0011A0012041004C		A		N
00023		SAMPLE	000006	F2	0001A3-		00300		N
00024		SAMPLE	000006	F2	0002A3		0014010014		
00025		SAMPLE	000006	F3	0011A0011122004T		A		N
00026		SAMPLE	000006	F3	0021A0024262005T		A		N
00027		SAMPLE	000006	F3	0031A0069151006A		A		N

Definition Database Update Report - UNIUPDT

This report is generated by running a Database Update and 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.

For an explanation of the record structure, see “Format of Batch Definition Transaction Records” on page 284.

The sample report below has been shortened to show only the file definition transaction records (F records).

B ■ Reports

Definition Database Update Report - UNIUPDT

Page 1 of 2

ACR/S releasenumbr		XYZ COMPANY		COPYRIGHT INFOGIX, INC		PAGE 001		
DATE: yy/mm/dd		DEFINITION DATA BASE UPDATE REPORT		DSN="C:\Infogix\Summary32\unidf.da		DATA BASE VERSION: 0000089/		
TIME: 14:47:37		TRANSACTION VALIDATION		-----		-----		
TRANS NO.	1	2	3	4	5	6	7	8
	TRANSACTION							VALIDATION MESSAGES
00001	SAMPLE	000000	F0	0001*				
00002	SAMPLE	000000	F0	0001A1PSA00100	0000000			YN
00003	SAMPLE	000000	F0	0002A		THIS IS A SAMPLE		
00004	SAMPLE	000001	F2	0001A3LOCATION:			00100	N
00005	SAMPLE	000001	F2	0002A3			000009	Y
00006	SAMPLE	000001	F2	0003A300002				
00007	SAMPLE	000001	F3	0011A0000041001C		A		N
00008	SAMPLE	000001	F3	0012AN00012				
00009	SAMPLE	000002	F2	0001A3-			00100	N
00010	SAMPLE	000002	F2	0002A3			000001	Y
00011	SAMPLE	000002	F2	0003A300014				
00012	SAMPLE	000002	F3	0011A0000122001T		A		N
00013	SAMPLE	000002	F3	0012AY00011				
00014	SAMPLE	000002	F3	0021A0000292002T		A		N
00015	SAMPLE	000002	F3	0022AY00024				
00016	SAMPLE	000002	F3	0031A0000121003C		A		N
00017	SAMPLE	000002	F3	0032AN00071				
00018	SAMPLE	000003	F2	0001A3LOCATION:			00200	N
00019	SAMPLE	000003	F2	0002A3			000009	Y
00020	SAMPLE	000003	F2	0003A300002				
00021	SAMPLE	000003	F3	0011A0000041002C		A		N
00022	SAMPLE	000003	F3	0012AN00012				
00023	SAMPLE	000004	F2	0001A3-			00200	N
00024	SAMPLE	000004	F2	0002A3			000001	Y
00025	SAMPLE	000004	F2	0003A300014				
00026	SAMPLE	000004	F3	0011A0000122006T		A		N
00027	SAMPLE	000004	F3	0012AY00011				
00028	SAMPLE	000004	F3	0021A0000282003T		A		N
00029	SAMPLE	000004	F3	0022AY00024				
00030	SAMPLE	000004	F3	0031A0000141005A		A		N
00031	SAMPLE	000004	F3	0032AN00070				
00032	SAMPLE	000005	F2	0001A3LOCATION:			00300	N
00033	SAMPLE	000005	F2	0002A3			000009	Y
00034	SAMPLE	000005	F2	0003A300002				
00035	SAMPLE	000005	F3	0011A0000041004C		A		N
00036	SAMPLE	000005	F3	0012AN00012				
00037	SAMPLE	000006	F2	0001A3-			00300	N
00038	SAMPLE	000006	F2	0002A3			000001	Y
00039	SAMPLE	000006	F2	0003A300014				
00040	SAMPLE	000006	F3	0011A0000122004T		A		N
00041	SAMPLE	000006	F3	0012AY00011				
00042	SAMPLE	000006	F3	0021A0000262005T		A		N
00043	SAMPLE	000006	F3	0022AY00024				
00044	SAMPLE	000006	F3	0031A0000151006A		A		N
00045	SAMPLE	000006	F3	0032AN00069				

Definition Database Update Report - UNIUPDT

Page 2 of 2

ACR/S releasenumbr		XYZ COMPANY		COPYRIGHT INFOGIX, INC		PAGE 002	
DATE: yy/mm/dd		DEFINITION DATA BASE UPDATE REPORT		DSN="C:\Infogix\Summary32\unidf.da		DATA BASE VERSION: 0000089/*	
TIME: 14:47:37		TRANSACTION PROCESSING					
TRANS NO.	TRANSACTION			RECORD KEY	ACTION		
00001	SAMPLE	000000	F0 0001*	SAMPLE	000000	F0 000	PURGED
				SAMPLE	000001	F2 000	PURGED
				SAMPLE	000001	F3 001	PURGED
				SAMPLE	000002	F2 000	PURGED
				SAMPLE	000002	F3 001	PURGED
00002	SAMPLE	000000	F0 0001A1PSA00100	0000000			
00003	SAMPLE	000000	F0 0002A				
00004	SAMPLE	000001	F2 0001A3LOCATION:	THIS IS A SAMPLE			
00006	SAMPLE	000001	F2 0003A300002	00100			
00007	SAMPLE	000001	F3 0011A0000041001C	A			
00008	SAMPLE	000001	F3 0012AN00012				
00009	SAMPLE	000002	F2 0001A3-	00100			
00011	SAMPLE	000002	F2 0003A300014				
00012	SAMPLE	000002	F3 0011A0000122001T	A			
00013	SAMPLE	000002	F3 0012AY00011				
00014	SAMPLE	000002	F3 0021A0000292002T	A			
00015	SAMPLE	000002	F3 0022AY00024				
00016	SAMPLE	000002	F3 0031A0000121003C	A			
00017	SAMPLE	000002	F3 0032AN00071				
00018	SAMPLE	000003	F2 0001A3LOCATION:	00200			
00020	SAMPLE	000003	F2 0003A300002				
00021	SAMPLE	000003	F3 0011A0000041002C	A			
00022	SAMPLE	000003	F3 0012AN00012				
00023	SAMPLE	000004	F2 0001A3-	00200			
00025	SAMPLE	000004	F2 0003A300014				
00026	SAMPLE	000004	F3 0011A0000122006T	A			
00027	SAMPLE	000004	F3 0012AY00011				
00028	SAMPLE	000004	F3 0021A0000282003T	A			
00029	SAMPLE	000004	F3 0022AY00024				
00030	SAMPLE	000004	F3 0031A0000141005A	A			
00031	SAMPLE	000004	F3 0032AN00070				
00032	SAMPLE	000005	F2 0001A3LOCATION:	00300			
00034	SAMPLE	000005	F2 0003A300002				
00035	SAMPLE	000005	F3 0011A0000041004C	A			
00036	SAMPLE	000005	F3 0012AN00012				
00037	SAMPLE	000006	F2 0001A3-	00300			
00039	SAMPLE	000006	F2 0003A300014				
00040	SAMPLE	000006	F3 0011A0000122004T	A			
00041	SAMPLE	000006	F3 0012AY00011				
00042	SAMPLE	000006	F3 0021A0000262005T	A			
00043	SAMPLE	000006	F3 0022AY00024				
00044	SAMPLE	000006	F3 0031A0000151006A	A			
00045	SAMPLE	000006	F3 0032AN00069				

Cross Reference Internal Items Report - UNIDXR.RPT

This report is generated by running the Cross Reference Internal Items utility. For each internal item or extended internal item defined for a job ID, it lists the item's description and any job ID that has a related item number. Related items can include the following reference types: history items, calculated items, rules, field items, and history analysis definitions.

If the internal item is referenced by a calculated item from another job, and the referencing calculated item is also stored as an internal item in the same job, the report prints an indicator (*) next to the calculated item and a note at the end of the report.

For each related item, the report lists:

- Its value type (input, output, count, amount, text, or decimal positions).
- Its reference type.

The TYPE column at the right margin of the report uses information specified in the Internal Item (or Extended Internal Item) dialog box. It consists of 3 characters indicating the following:

- First character. Identifies your entry in the **Column** field indicating in which column on the Control Report the item will print: L = Left, R = Right, C=Calculated Item. This field does not appear for extended internal items.
- Second character. Identifies your entry in the **Format** field: C = Count, A = Amount, D = Date. T = Text
- Third character. If you selected Count or Amount in the Format field, shows the number of decimals for the item.

For example LCO would mean that the item is printed on the Left side of the report, print format is Count with 0 decimal places.

Cross Reference File Definitions Report - UNIDXR.RPT

Example

JOB ID	ITEM#	DESCRIPTION / REFERENCES	TYPE
ATEST STEP1	I-001	DESCRIPTION: TOTAL FOR JOHNSON AMOUNT 11111111112222TOTAL FOR JOHNSON AMOUNT 33333333334444 REFERENCES: ATEST STEP1 C-005* ATEST STEP1 R-010	LA2
ATEST STEP1	I-002	DESCRIPTION: DOUGLAS SHORT TEXT REFERENCES: ATEST STEP1 E-001 ATEST STEP1 R-003	LT0
ATEST STEP1	I-003	DESCRIPTION: TOTAL AMOUNT FOR JOHNSON COUNT REFERENCES: ATEST STEP1 E-006	LC0
ATEST STEP1	I-004	DESCRIPTION: DOUGLAS MED TEXT REFERENCES: ATEST STEP1 E-005	LT0
ATEST STEP1	I-005	DESCRIPTION: DOUGLAS AMOUNT REFERENCES: ATEST STEP1 C-005*	LA2
ATEST STEP1	I-006	DESCRIPTION: DOUGLAS COUNT REFERENCES: ATEST STEP1 R-004 ATEST STEP1 R-005	LC0
ATEST STEP1	X-001	DESCRIPTION: DOUGLAS LONG TEXT REFERENCES: ATEST STEP1 E-005	T0
ATEST STEP1	X-002	DESCRIPTION: DOUGLAS SHORT TEXT 40 REFERENCES: ATEST STEP1 E-004 ATEST STEP1 R-001 ATEST STEP1 R-002 ATEST STEP1 R-004 ATEST STEP1 R-005	T0
ATEST STEP1	X-003	DESCRIPTION: NELSON LONG NUMBER AMOUNT REFERENCES: ATEST STEP1 R-010	A2
ATEST STEP1	X-004	DESCRIPTION: NELSON LONG NUMBER COUNT REFERENCES: ATEST STEP1 E-007	C0
ATEST STEP1	X-005	DESCRIPTION: NELSON 31 1111111111222222222233333333NELSON 31 4444444444555555555566666666 REFERENCES: ATEST STEP1 E-007	A2
ATEST STEP1	X-006	DESCRIPTION: NELSON 30 REFERENCES: ATEST STEP1 C-009	A2

* - THIS CALCULATED ITEM IS STORED AS AN INTERNAL ITEM. RUN THE CROSS REFERENCE UTILITY TO DETERMINE ANY INDIRECT REFERENCES.
*****END OF REPORT*****

Cross Reference File Definitions Report - UNIDXR.RPT

This report is generated by running the Cross Reference File Definitions utility. The report lists all the file definitions used within the first database, and if specified, within the second database. The report allows you to find job definitions 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 model definitions and an indicator (#) to show which file ID is referenced by an alternate job.

B ■ Reports

Database History and Statistics Report - UNIHLR.RPT

Example

0	FILE ID	DESCRIPTION / REFERENCES	TYPE
OF1E1A	DESCRIPTION: FILE 1 A JOB1 FILEREF JOB1R1 FILEREF /JOB1 JOB1R2 FILEREF /JOB1 JOB2 FILEREF /FILE2A	DATABASE: USER01.SUM.DEFN FILEREF # FILEREF # *	PS
OF1E1A	DESCRIPTION: FILE 1 A JOB1 TEST1 JOB2 TEST2 *	DATABASE: USER02.SUM.DEFN	PS
OF1E2A	DESCRIPTION: FILE 2 A JOB2 FILEREF	DATABASE: USER01.SUM.DEFN	PS

* - USE FILE AS MODEL DEFINITIONS
- ALTERNATE JOB/PRIMARY JOB
*****END OF REPORT*****

Database History and Statistics Report - UNIHLR.RPT

Generated by the List History utility, this section of the History Data Base Listing contains 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 Job IDs, and the number of histories on file.

DATE AND TIME OF INITIALIZATION:	yy/mm/dd	AT	13:27:28
DATE AND TIME OF LAST UPDATE:	yy/mm/dd	AT	15:40:13
NUMBER OF JOB ID KEYS:			
MAXIMUM:			32
USED:			21
AVAILABLE:			11
NUMBER OF ACR HISTORIES (INCLUDING ADJUSTMENTS):			
MAXIMUM:			300
USED:			31
AVAILABLE:			269

Index of ACR Histories - UNIHLR.RPT

Generated by the List History utility, this section of the History Data Base Listing contains a list of every job ID in the history database along with the number of histories (cycles) for each job ID.

JOB ID	NUMBER OF HISTORIES	JOB ID	NUMBER OF HISTORIES	JOB ID	NUMBER OF HISTORIES	JOB ID	NUMBER OF HISTORIES
BOOKCLUBVERIFY	1	CARENZL COMPARE	1	CLAIMS STEP1	1	DOMESTICSALES	1
IFTHEN STEP1	1	JOBYZ UPDATE	5	JOBYZ UPDATE TO	2	SALES DAILY	1
SAMPLE UPDATE	1	STATEOH SALES	1	STATES SALES	1	TOTALS STEP1	4

Summary of ACR Histories - UNIHLR.RPT

Generated by the List History utility, this section of the History Data Base Listing contains information about each Job ID in the history database including cycle and run numbers, run date, run time, return code, and status of reruns and/or updates. This section will be generated based on the options you choose.

ACR/S releasenumbr		XYZ COMPANY				COPYRIGHT INFOGIX, INC							
DATE: yy/mm/dd		HISTORY DATA BASE LISTING				DATA BASE VERSION: *****/0000011							
TIME: 14:08:45		DSN="/home/userid/unihf.dat"											
0		SUMMARY OF ACR HISTORIES											
0													
JOB ID	CYCLE#	RUN#	RETURN CODE	STATUS	RUN DATE	RUN TIME	---LAST RERUN IF ANY---	DATE	TIME	NO.	DEFN	VERSION	PROGRAM

FONSERV	STEP1	19960330	000	3001		yy/mm/dd	08:12:43				0000086	0000010	UAC2000
		19960330	000	3001	***UPD	yy/mm/dd	08:12:43				0000086	0000009	UAC2000
IFTHEN	STEP1	00960823	000	0000		yy/mm/dd	08:08:36				0000086	0000008	UAC2000
JOEXYZ	UPDATE	96062005	000	0000		yy/mm/dd	07:49:04				0000085	0000005	UAC2000
JOEXYZ	UPDATE	TO 00000001	000	0000		yy/mm/dd	07:43:53				0000085	0000002	UAC2000
		96062004	000	0000		yy/mm/dd	07:45:12				0000085	0000004	UAC2000
		96062005	000	0000		yy/mm/dd	07:49:08				0000085	0000006	UAC2000

History Data Detail Report - UNIHSTD.RPT

Generated by the List History utility, this report shows either selected history records or every history record and the values stored for it. It is generated as a separate report when you are running List History with the Report Type field set to any option that includes Detail.

Transaction Listing Report - UNITR.RPT

This report is generated when you run any of a variety of processes. These include:

- Direct input balancing
- Copy History, selecting Copy all records
- Delete History
- Extract History
- Update History
- Create Comma Delimited File

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. Following are two examples.

Utility Transaction Listing - After Copy All Records

ACR/S releasenumbr	INFOGIX, INCORPORATED	COPYRIGHT INFOGIX, INC	
DATE: yy/mm/dd	INFOGIX DATA BASE UTILITY		PAGE 001
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			#UTL000I: PROCESSING COMPLETED WITHOUT ERRORS

ACR Test Report - Input Transaction Listing (After Direct Input Balancing)

ACR/S releasenumbr	INFOGIX, INCORPORATED	COPYRIGHT INFOGIX, INC	1999
DATE: yy/mm/dd	ACR TEST REPORT	PAGE	001
TIME: 15:39:13			
-----INPUT 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 DIRECTINSTEP1 011.0000881206000			
00002 DIRECTINSTEP1 01201+0000000000000628+000000000000172+0000000000000800			
00003 DIRECTINSTEP1 01202+0000000000754300'SUBTOTAL'			+0000000000754300
00004 DIRECTINSTEP1 021.0000881206000			
00005 DIRECTINSTEP1 02201+0000000000000987+0000000000000879			
00006 DIRECTINSTEP1 031.0000881206000			
00007 DIRECTINSTEP1 03201+0000000000745623+0000000000754623			
00008 DIRECTINSTEP1 041.0000881206000			
00009 DIRECTINSTEP1 04201+0000000000458723+000000000058623			
00010 DIRECTINSTEP1 061.0000881206000			
00011 DIRECTINSTEP1 06201+000000000008000+000000000000050'TRY THIS'			
00012 DIRECTINSTEP1 06202+000000000000004+00000000001.0005+000000000000045			
00013 DIRECTINSTEP1 06203'TRY THIS'			
00014 DIRECTINSTEP1 071.0000881206000			
00015 DIRECTINSTEP1 07201+000000000001001+000000000000057+0000000000000450			
00016 DIRECTINSTEP1 07202+000000000000001+000000000000013+0000000000000980			
00017 DIRECTINSTEP1 07203+000000000001000+000000000000057+0000000000000450			
00018 DIRECTINSTEP1 07204+000000000000001+000000000000057+0000000000000980			
*****END OF REPORT*****			

History Database Copy Report - Summary of Histories Transferred - UNIHTR.RPT

History Database Copy Report - Summary of Histories Transferred - UNIHTR.RPT

Generated by the Copy History utility with the Copy selected records option, this report lists each history record copied with information about the record.

Note: (The Copy all records option generates a Transaction Listing Report, as explained in Transaction Listing Report - UNITR.RPT on page 320.)

ACR/S releasenumbr		XYZ COMPANY				COPYRIGHT INFOGIX, INC				PAGE 001
DATE: yy/mm/dd		INFOGIX DATA BASE UTILITY				DATA BASE VERSION: *****/0000015				
TIME: 15:01:11		HISTORY DATA BASE COPY REPORT				SUMMARY OF HISTORIES TRANSFERRED				
		FROM: "/home/userid/unihf.dat"				TO: "/home/userid/uachf1.dat"				
JOB ID	CYCLE#	RUN#	CODE	RUN DATE	RUN TIME	TYPE	RERUN DATE	RERUN TIME	DATABASE VERSION	PROGRAM NAME
SALES	DAILY	99960501	000	0000	yy/mm/dd	14:59:21			0000095 0000013	UAC2000
SALES	DAILY	99960502	000	0000	yy/mm/dd	15:00:33			0000095 0000014	UAC2000

History Database Copy Report - Summary of Histories Unloaded - UNIHTR.RPT

Generated by the Copy History utility with the Unload records to a work file option, this report lists each history record unloaded with information about the record.

ACR/S releasenumbr		XYZ COMPANY				COPYRIGHT INFOGIX, INC				PAGE 001
DATE: yy/mm/dd		INFOGIX DATA BASE UTILITY				DATA BASE VERSION: *****/0000015				
TIME: 15:08:57		HISTORY DATA BASE UNLOAD REPORT				SUMMARY OF HISTORIES UNLOADED				
		FROM: "/home/userid/unihf.dat"				TO: "/home/userid/uachf1.dat"				
JOB ID	CYCLE#	RUN#	CODE	RUN DATE	RUN TIME	TYPE	RERUN DATE	RERUN TIME	DATABASE VERSION	PROGRAM NAME
JOBXYZ	UPDATE	96062004	000	3010	yy/mm/dd	07:44:34			0000093 0000002	UAC2000
JOBXYZ	UPDATE	TO 96062004	000	0000	yy/mm/dd	07:44:38			0000093 0000003	UAC2000
JOBXYZ	UPDATE	96062005	000	0000	yy/mm/dd	07:53:04			0000094 0000004	UAC2000
JOBXYZ	UPDATE	TO 96062005	000	0000	yy/mm/dd	07:53:08			0000094 0000005	UAC2000
FLYER	DAILY	19960503	000	0000	yy/mm/dd	08:50:08			0000094 0000007	UAC2000
FLYER	DAILY	19960504	000	0000	yy/mm/dd	08:51:48			0000094 0000009	UAC2000
CARENTL	COMPARE	99960503	000	0000	yy/mm/dd	14:57:48			0000095 0000012	UAC2000
SALES	DAILY	99960501	000	0000	yy/mm/dd	14:59:21			0000095 0000013	UAC2000
SALES	DAILY	99960502	000	0000	yy/mm/dd	15:00:33			0000095 0000014	UAC2000

B ■ Reports

History Database Copy Report - Summary of Histories Reloaded - UNIHTR.RPT

History Database Copy Report - Summary of Histories Reloaded - UNIHTR.RPT

Generated by the Copy History utility with the with the Reload records from a work file option, this report lists each history record reloaded with information about the record.

ACR/S releasenumbr		XYZ COMPANY				COPYRIGHT INFOGIX, INC				PAGE 001	
DATE: yy/mm/dd		INFOGIX DATA BASE UTILITY				DATA BASE VERSION: *****/****					
TIME: 15:11:18		HISTORY DATA BASE RELOAD REPORT				SUMMARY OF HISTORIES RELOADED					
FROM: "/home/userid/uachf1.dat"		RETURN		TO: "/home/userid/unihf.dat"		RERUN		RERUN		DATABASE VERSION PROGRAM	
JOB ID	CYCLE#	RUN#	CODE	RUN DATE	RUN TIME	TYPE	DATE	TIME	DEFN	/HISTORY	NAME
CARENTL COMPARE	99960503	000	0000	yy/mm/dd	14:57:48				0000095	0000012	UAC2000
FLYER DAILY	19960503	000	0000	yy/mm/dd	08:50:08				0000094	0000007	UAC2000
FLYER DAILY	19960504	000	0000	yy/mm/dd	08:51:48				0000094	0000009	UAC2000
JOEXYZ UPDATE	96062004	000	3010	yy/mm/dd	07:44:34				0000093	0000002	UAC2000
JOEXYZ UPDATE	96062005	000	0000	yy/mm/dd	07:53:04				0000094	0000004	UAC2000
JOEXYZ UPDATE TO	96062004	000	0000	yy/mm/dd	07:44:38				0000093	0000003	UAC2000
JOEXYZ UPDATE TO	96062005	000	0000	yy/mm/dd	07:53:08				0000094	0000005	UAC2000
SALES DAILY	99960501	000	0000	yy/mm/dd	14:59:21				0000095	0000013	UAC2000
SALES DAILY	99960502	000	0000	yy/mm/dd	15:00:33				0000095	0000014	UAC2000

History Database Reorg Report - UNIHRR.RPT

This report is generated when you run the Reorganize History utility. The report lists the including Job ID, Cycle Number, Run Number, etc. for logical records deleted. It will be blank if no logical records were in the database when the reorg was run.

Example

ACR/S releasenumbr		INFOGIX, INCORPORATED				COPYRIGHT INFOGIX, INC				PAGE 001	
DATE: yy/mm/dd		INFOGIX DATA BASE UTILITY				DATA BASE VERSION: *****/0000163					
TIME: 16:25:38		HISTORY DATA BASE REORG REPORT				SUMMARY OF DROPPED ACR HISTORIES					
JOB ID	CYCLE#	RUN#	CODE	RUN DATE	RUN TIME	TYPE	DEFN	HISTORY	PROGRAM	LOAD	LIBRARY
CLIFF SU	20200406	000	0000	yyyy/mm/dd	08:59:23	***DEL	0000595	0000024	UAC2000	R99UC02	LOADLIB
FFREFORMSTEP01	20190331	000	3003	yyyy/mm/dd	09:51:30	***UPD	0000594	0000020	ADSRAXAA	TSC	LINKLIB
INVENTORCLAS01	01 00970401	000	3081	yyyy/mm/dd	10:25:35	***UPD	0000532	0000019	UAC2000	UNI.UNISEC.PLUS	.C24.B04
MISCITEMSTEP01	20200406	000	0000	yyyy/mm/dd	11:59:23	***DEL	0000595	0000024	UAC2000	R99UC02	LOADLIB

Print Management Report - UNIMR.RPT

This report analyzes and prints the contents of the history database based on user-defined selection criteria. Managers can use this utility to analyze the functioning of the system. For example, you can run a report showing all the return codes greater than 0 (all out-of-balance jobs) for a particular time period. The criteria include:

- Return code
- Job ID
- Absolute time
- Relative time
- Absolute cycle ID
- Relative Cycles
- Cycle reference job

Balancing Summary (Print Management) Report (Page 1 of 2)

ACR/S releasenumbr	XYZ COMPANY	COPYRIGHT INFOGIX, INC	
DATE: YY/MM/DD	INFOGIX DATA BASE UTILITY		PAGE 001
TIME: 09:33:43	INFOGIX BALANCING SUMMARY		DATA BASE VERSION: 0000086/0000010
SELECTION CRITERIA:			
	PARAMETER	FROM	TO
	-----	-----	-----
	RETURN CODES	0000	9999
	JOB ID KEYS	ALL	
	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:	DEFINITIONS = "/home/userid/unidf.dat"		
	HISTORY = "/home/userid/unihf.dat"		

Balancing Summary (Print Management) Report (Page 2 of 2)

ACR/S releasenumbr	XYZ COMPANY	COPYRIGHT INFOGIX, INC						
DATE: YY/MM/DD	INFOGIX DATA BASE UTILITY		PAGE 002					
TIME: 09:33:43	INFOGIX BALANCING SUMMARY		DATA BASE VERSION: 0000086/0000010					
		RETURN						
JOB ID	ACR REPORT TITLE	CYCLE	RUN CODE	RUN DATE	RUN TIME	RERUNS	ADJ	RESULTS
	-----	-----	-----	-----	-----	-----	-----	-----
FONSERV STEP1	TOTAL COMPARISONS	19960330	000 3001	97/08/30	08:12:43	0		OUT OF BALANCE
IFTHEN STEP1	BIMONTHLY SALES REPORT	00960823	000 0000	97/08/30	08:08:36	0		IN BALANCE
JOBXYZ UPDATE	MONTHLY FILE UPDATE SYSTEM	96062005	000 0000	97/08/30	07:49:04	0		IN BALANCE
JOBXYZ UPDATE TO	MONTHLY FILE UPDATE SYSTEM	00000001	000 0000	97/08/30	07:43:53	0		IN BALANCE
		96062004	000 0000	97/08/30	07:45:12	0		IN BALANCE
		96062005	000 0000	97/08/30	07:49:08	0		IN BALANCE

B ■ Reports

Print Management Report - UNIMR.RPT

Using ACR/Summary for Windows in Batch

When you run ACR/Summary, batch files (*.bat) are created and executed. The definitions used by the batch files are typically created through ACR/Summary Client or ACR/Workbench.

This appendix assists you in customizing the batch files for your batch environment. In addition, this appendix describes how to migrate your jobs from test to production.

This appendix contains the following sections:

- “Conditionally Executing a Step Based on a Return Code” on page 325
- “Before You Begin” on page 326
- “Editing Batch Files” on page 327
- “Batch File Quick-Reference with Page Numbers” on page 328
- “Migrating Test Definitions to Production” on page 345

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/Summary 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/Summary
CALL UAC2000
REM Check return code
if ERRORLEVEL 0 goto OK
REM Not OK, so run detail job
CALL UDS2000
: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/Summary 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/Summary’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/Summary installation folder `C:\Infogix\Summary32` and add the following contents:

```
[XFH-DEFAULT]
EXPANDTAB=OFF
```

2. Add the following statement to the `uac2000` execution `.skl .bat` file:

```
SET EXTFH=C:\Infogix\Summary32\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 ships 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 extfh.cfg.

To override the environment variable FILEMAXSIZE, do the following:

1. Create the file extfh.cfg in the ACR/Summary installation folder C:\Infogix\Summary32, and add the following contents:

```
[XFH-DEFAULT]
FILEMAXSIZE=8
IDXFORMAT=8
FILEPOINTERSIZE=8
```

2. Add the following statement to the uac2000 execution .skl .bat file:

```
SET EXTFH=C:\Infogix\Summary32\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 326.
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 ACRSWB=?????????  
SET RPT=?????????  
SET COBCONFIG=%ACRSWB%\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 uac2000.bat (Run Balancing in File Interface Mode), 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:

- “uac2000.bat - Run Balancing in File Interface Mode” on page 330
- “uac3000.bat - Run Balancing in Direct Input Mode” on page 334
- “uacdfb.bat - Build Definitions” on page 334
- “uacdfi.bat - List Definitions” on page 335
- “uacdfx.bat - Cross-Reference Internal Items” on page 336
- “uachfc.bat - Copy History” on page 336

- “uachfd.bat - Delete History” on page 337
- “uachfe.bat - Extract History” on page 337
- “uachfl.bat - List History” on page 338
- “uachfr.bat - Reorganize History” on page 338
- “uachfrp.bat - Recalculate/Reprint ACR Report” on page 339
- “uachft.bat - Transfer History (Selected Records)” on page 339
- “uachftu.bat - Unload History” on page 340
- “uachftr.bat - Reload History” on page 341
- “uachfu.bat - Update History” on page 341
- “uacinit.bat - Initialize Databases” on page 342
- “uacmr01.bat - Print Management Report” on page 343
- “uacupdt.bat - Update Definitions” on page 343
- “uan2000.bat - History Analysis Reporting” on page 343
- “uacupdt.bat - Update Definitions” on page 343
- “uan2000.bat - History Analysis Reporting” on page 343
- “uaccsv.bat - Create Comma Delimited Text File” on page 344

uac2000.bat - Run Balancing in File Interface Mode

This batch file is used to run the balancing function, described in the section “Balancing Function” on page 20.

Refer to “Editing Batch Files” on page 327 and to the following tables when editing this batch file.

Required Environment Variables Table

Variable	Definition
ACRSWB	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.
JOBVAR	Type the path and a file name of your choice for the JOBVAR environment variable file. The file is dynamically assigned when the job is run.

Run-time Parameters (PARM Options)

Parm	Specified as:
Job name	PARM='JNAM=jjjjjjj' where jjjjjjj is the 1- to 8-character job name. Example: JNAM=CUST
Step name	PARM='STEP=ssssssss' where ssssssss is the 1- to 8-character step name. Example: STEP=STEP1
Job-step qualifier	PARM='JQ=qq' where qq is the 2-character JSQ. Example: JQ=02
Cycle Number	PARM='CYC#=nnnnnnnn' where nnnnnnnn is the 8-digit Cycle Number. Example: CYC#=20060815
Run Number	PARM='RUN#=nnn' where nnn is the 3-digit Run Number.
Control Field	PARM='CNTL=01Xr' where r is the rule set number for Alternate Balancing Rules.

Parm	Specified as:
Store History	<p>PARM='ROPT=abcdefgh'</p> <p>where:</p> <p>a is the store history override option (Y, N, or I). Y = Store history every time the balancing job runs; N = Never store history; I = Store history only when the ACR/Summary job is in balance. Positions b-h are reserved. Use a total of seven dots to fill positions b-h so as not to override the options they control. Example, ROPT=Y.....</p>
Rerun	<p>PARM='RERUN=Y'</p> <p>where RERUN causes the UAC2000 program to rerun the last cycle processed. When cycles other than the last cycle need to be rerun, use the PARM='CYC#=nnnnnnnn,' option.</p>
SYSOUT	<p>PARM='SYSOUT'</p> <p>where SYSOUT causes the UAC2000 program to produce a SYSOUT report even when no #U messages are created. If you specify the SYSOUT parameter, an informational header message will be generated whenever the job runs. The message will say, "USY0001: INFOGIX JOB STARTED."</p>
Bypass	<p>PARM='BYPASS'</p> <p>causes the UAC2000 program to bypass balancing.</p>
File Work Area Size	<p>PARM='CORE=nnnn'</p> <p>where nnnn is the size in 'K' of the area that the File Interface Program will use in obtaining storage for the file work area.</p>
Control Report XML Option	<p>PARM='CRXML=n'</p> <p>where n is one of the following:</p> <ol style="list-style-type: none"> 1 - do not generate the Control Report in XML format 2 - generate, write to file (Copy 1) 3 - generate, write to database (Copy 1) 4 - generate, write to file (Copy 2) 5 - generate, write to database (Copy 2)

C ■ Using ACR/Summary for Windows in Batch

Batch File Quick-Reference with Page Numbers

Parm	Specified as:
User Report XML Option	PARM='URXML=n' where n is one of the following: 1 - do not generate the User Report in XML format 2 - generate, write to file 3 - generate, write to database
Free-Form Report XML Option	PARM='FRXML=n' 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, write to database
Debug options	PARM='DEBUG=abcdefgh' where: a is the XML Writer debug option (Y, N, blank or (.)). b is the Visibility API debug option (Y, N, blank or (.)). c is the Infogix Assure History debug option (Y, N, blank or (.)). d-h are reserved. Processing options: Y = Debug on. N = Blank. (.) = Debug off. Use a total of five dots to fill positions d-h to avoid overriding the options they control. Example, DEBUG=YYY.....

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

In access modes 6 and 7, 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 semicolons. In general, the maximum length of all concatenated pathnames is 8192 characters. For more information on concatenation, see the comments in the uac2000.skl file.

For record lengths greater than 256, uncomment out the SWBIOBUFF parameter.

C ■ Using ACR/Summary for Windows in Batch

Batch File Quick-Reference with Page Numbers

E-mail Notification

If you are using e-mail notification, uncomment out the SET SMTPHOST and SMTPPORT parameters.

External Translation Tables

When using external translation tables, you must set the file IDs and specify the paths of the files the table will use.

uac3000.bat - Run Balancing in Direct Input Mode

This mode enables you to specify the control values to be stored in the internal or extended internal items and used in balancing directly, rather than having ACR/Summary extract them from input sources. For more information, see “Selecting an Application Interface Mode” on page 38.

Refer to “Editing Batch Files” on page 327 and to the following tables when editing this batch file.

Required Environment Variables Table

Variable	Definition
ACRSWB	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.

uacdfb.bat - Build Definitions

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. Types of transactions you can select include:

- All types
- Jobs
- Files
- Internal Translation Tables/Cycle Tables
- External Translation Tables
- History Analysis definitions
- Jobs and Associated Files

This utility is useful for making mass definition changes, creating an audit trail, transferring definitions between platforms, etc.

Refer to “Editing Batch Files” on page 327 and to the following tables when editing this batch file.

Required Environment Variables Table

Variable	Definition
ACRSWB	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 transactions file.

uacdf.bat - 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. Types of definitions you can specify include:

- All definitions (blank)
- Jobs (J)
- Files (F)
- Internal Tables (T)
- External Tables (X)
- History Analyses (S)
- Jobs and Associated Files

Refer to “Editing Batch Files” on page 327 and to the following tables when editing this batch file.

Required Environment Variables Table

Variable	Definition
ACRSWB	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.

uacdfx.bat - Cross-Reference Internal Items

This utility produces a report listing the definitions in the definition database for a specified job ID or a range of Job IDs. For each internal or extended internal item defined for that job ID, the report lists the item's description and any job ID that has a related item number. Related items can include history items, calculated items, rules, field items, and history analysis definitions.

Refer to “Editing Batch Files” on page 327 and to the following tables when editing this batch file.

Required Environment Variables Table

Variable	Definition
ACRSWB	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.

uachfc.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.

Refer to “Editing Batch Files” on page 327 and to the following tables when editing this batch file.

Required Environment Variables Table

Variable	Definition
ACRSWB	Path where UNIRUN.EXE resides (BASE\Cobol).
RPT	Path where the report files will be generated.
UNIHF1	Path and file name of the history database.
UNIHF2	Path and file name of the new history database.

uachfd.bat - Delete History

This utility can perform a logical delete of one or multiple history records by job ID, cycle, and date. Wildcards can be used in the **From Job ID** field to select multiple job IDs. This can be useful when a job ran that was not scheduled to run, when a job ID is no longer needed in the history database, etc.

NOTE: The physical history records are not removed from the file until the Reorganize History batch file is executed.

Refer to “Editing Batch Files” on page 327 and to the following tables when editing this batch file.

Required Environment Variables Table

Variable	Definition
ACRSWB	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.

uachfe.bat - 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. For more information and the layouts that can be generated, see “Extract History” on page 260.

Refer to “Editing Batch Files” on page 327 and to the following tables when editing this batch file.

Required Environment Variables Table

Variable	Definition
ACRSWB	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.
UNIHFWD	Path and file name of the exact file.

uachfl.bat - List History

This utility generates a selection of reports regarding the records in the history database. For descriptions and examples of the reports that can be generated by this utility, see the following:

- “Reports Table” on page 291
- “Transaction Listing Report - UNITR.RPT” on page 320
- “History Data Detail Report - UNIHSTD.RPT” on page 319
- “Summary of ACR Histories - UNIHRLR.RPT” on page 319
- “Database History and Statistics Report - UNIHRLR.RPT” on page 318
- “Index of ACR Histories - UNIHRLR.RPT” on page 318

Refer to “Editing Batch Files” on page 327 and to the following tables when editing this batch file.

Required Environment Variables Table

Variable	Definition
ACRSWB	Type the default working directory.
RPT	Path where the report files will be generated.
UNIHF	Path and file name of the history database.

uachfr.bat - Reorganize History

This utility physically deletes all logically deleted records from this history database. Run this utility regularly to ensure optimum processing efficiency.

Refer to “Editing Batch Files” on page 327 and to the following tables when editing this batch file.

Required Environment Variables Table

Variable	Definition
ACRSWB	Path where UNIRUN.EXE resides (\$BASE\Cobol).
RPT	Path where the report files will be generated.
UNIHF1	Path and file name of the history database.
UNIHF2	Path and file name of the reorganized history database.
UNIDF	Path and file name of the definition database.

uachfrp.bat - Recalculate/Reprint ACR Report

This utility reprints the (Balancing) Control Report (UNIACR.RPT) for a job using control values stored in the definition and history databases. This enables you to generate the report without rerunning the job.

Because the definitions used are the ones that exist when the utility is being run, a possibility exists that the results this utility yields will differ from the original balancing run. If someone has modified the definitions being used and updated the history control values since the original run, the Control Report will show results that differ from those of the original run.

Refer to “Editing Batch Files” on page 327 and to the following tables when editing this batch file.

Required Environment Variables Table

Variable	Definition
ACRSWB	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.
UNIDF	Path and file name of the definition database.

uachft.bat - Transfer History (Selected Records)

This utility transfers selected records from an existing history database directly to another file. You can select a range of histories based on job ID and/or date and time and copy them to another database. This can be useful when copying a test job into production or vice versa.

Refer to “Editing Batch Files” on page 327 and to the following tables when editing this batch file.

uachftr.bat - Reload History

This utility is used after you use the Unload Records utility. It enables you to select a range of histories based on job ID and/or date and time and reload them (from the work file to which they were unloaded) into the history database.

Refer to “Editing Batch Files” on page 327 and to the following tables when editing this batch file.

Required Environment Variables Table

Variable	Definition
ACRSWB	Path where UNIRUN.exe resides (\$BASE\Cobol)
RPT	Path where the report files will be generated
UNIHF2	Path and file name of the target history database.
UNIDF	Path and file name of the definition database.
UNIHFW	Path and file name of the work file.

uachfu.bat - Update History

This batch file updates (i.e., replaces) one or more control values in the history database for a specified job ID and cycle ID. This batch file 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.

Refer to “Editing Batch Files” on page 327 and to the following tables when editing this batch file.

Required Environment Variables Table

Variable	Definition
ACRSWB	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.

uacinit.bat - Initialize Databases

This utility initializes (creates or re-creates) the definition and/or history database. Both databases must be initialized before you can begin using ACR/Summary for the first time.

Reinitializing a database erases the contents.

Refer to “Editing Batch Files” on page 327 and to the following tables when editing this batch file.

Required Environment Variables Table

Variable	Definition
ACRSWB	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 Parameters (PARM options)

Option	Use
-- blank --	Initialize both the definition and history databases.
DF	Initialize the definition database only.
HF	Initialize the history database only.

uacmr01.bat - Print Management Report

This utility allows you to print a report that summarizes balancing history information for management purposes.

Refer to “Editing Batch Files” on page 327 and to the following tables when editing this batch file.

Required Environment Variables Table

Variable	Definition
ACRSWB	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.

uacupdt.bat - Update Definitions

This utility loads your definitions for the current job, table, or history analysis document into the definition database. This is necessary in order to use the definitions in ACR/Summary processing.

Refer to “Editing Batch Files” on page 327 and to the following tables when editing this batch file.

Required Environment Variables Table

Variable	Definition
ACRSWB	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.

uan2000.bat - History Analysis Reporting

The History Analysis Reporting batch file extracts, analyzes, and reports on selected history data. This can be used to spot fluctuations in your balancing processes. For more information, see “Using History Analysis” on page 247.

Refer to “Editing Batch Files” on page 327 and to the following tables when editing this batch file.

C ■ Using ACR/Summary for Windows in Batch

Batch File Quick-Reference with Page Numbers

Required Environment Variables Table

Variable	Definition
ACRSWB	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.

uaccsv.bat - Create Comma Delimited Text File

This utility converts an extract file that has been created using the Extract History utility to an ASCII text file containing sequential fields separated by commas and text values surrounded by quotation marks. For more information, see “Create Comma Delimited Text File” on page 264.

Refer to “Editing Batch Files” on page 327 and to the following tables when editing this batch file.

Required Environment Variables Table

Variable	Definition
ACRSWB	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.

Migrating Test Definitions to Production

What are Definitions?

Definitions are data balancing 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 definitions 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 balance the data as desired.

Testing Definitions

Once you have set up definitions, you must ensure that they balance 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

Use the instructions in “[uacdfb.bat - Build Definitions](#)” on page 334 to execute the batch file, which will build batch transaction cards directly from the test definition database.

Updating the Production Definition Database

Use the instructions in “uacupdt.bat - Update Definitions” on page 343 to edit and run the udsupdt.bat batch file, which will build batch transaction cards directly from the test definition database.

When you executed the uacdfb.bat batch file in the previous section, a file was created and a file name was chosen. As you execute the uacupdt.bat batch file to update production, the file name used for building batch definitions should be used as input to the variable UNIINP.

Using ACR/Summary for UNIX or Linux in Batch

When you run ACR/Summary for UNIX or Linux, scripts are executed. The job and file definitions used by the scripts are typically created through ACR/Summary Client or ACR/Workbench.

This appendix assists you in customizing the ACR/Summary scripts for your batch environment. In addition, this appendix describes how to migrate your jobs from test to production.

This appendix contains the following sections:

- “Conditionally Executing a Step Based on a Return Code” on page 347
- “Before You Begin Your Edits” on page 349
- “Editing the Scripts” on page 350
- “Script Quick-Reference with Page Numbers” on page 352
- “Migrating Test Definitions to Production” on page 368

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/Summary step that precedes it. You can do this by editing the script. For example, following is the last part of `uac2000.sh`, the script for balancing in file interface mode.

D ■ Using ACR/Summary 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 UAC2000  
#-----  
RUNCMD=${RUNCMD-cobrun} ; export RUNCMD  
CC=`SRUNCMD UAC2000 $PARM | grep "STEP COMPLETION" | awk '{print $4}'`  
#-----  
# cleanup transient files  
#-----  
for frame 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:

$$\text{Step completion code} / 256 = x.y$$

$$256 * y = \text{exit status code}$$

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:

$$3000 / 256 = 11.71875$$

$$256 * .71875 = 184$$

Before You Begin Your Edits

This section gives you important information that you should know before attempting to set up and execute any of the scripts.

Location of Scripts

Scripts are located in the administrative account's scripts subdirectory. These can be modified or used as templates. Original scripts reside in another ("backup") location: *installdirectory/sumdet/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/Summary on this UNIX or Linux host. For more information, see “Setting User Options for UNIX or Linux” on page 168.

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 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 the Scripts

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, require additional edits, which are described in the script-specific documentation.

Script Quick-Reference with Page Numbers

The following table lists each script, its function, and the page where you can find out more.

- “uac2000.sh - Balancing in File Interface Mode” on page 352
- “uac3000.sh - Balancing in Direct Input Mode” on page 357
- “uacdfb.sh - Build Definitions” on page 357
- “uacdfi.sh - List Definitions” on page 358
- “uacdfx.sh - Cross-Reference Internal Items” on page 359
- “uachfc.sh - Copy History” on page 359
- “uachfd.sh - Delete History” on page 360
- “uachfe.sh - Extract History” on page 360
- “uachfl.sh - List History” on page 361
- “uachfr.sh - Reorganize History” on page 361
- “uachfrp.sh - Recalculate/Reprint ACR Report” on page 362
- “uachft.sh - Transfer History (Selected Records)” on page 363
- “uachftu.sh - Unload History” on page 363
- “uachftr.sh - Reload History” on page 364
- “uachfu.sh - Update History” on page 365
- “uacinit.sh - Initialize Databases” on page 365
- “uacmr01.sh - Print Management Report” on page 366
- “uacupdt.sh - Update Definitions” on page 366
- “uan2000.sh - History Analysis Reporting” on page 367
- “uaccsv.sh - Create Comma Delimited Text File” on page 367

uac2000.sh - Balancing in File Interface Mode

This script is used to run the balancing function, described in the section “Balancing Function” on page 20.

Refer to “Editing the Scripts” on page 350 and to the following tables when editing this script.

Required Environment Variables Table

Variable	Definition
ACRSWB	Path where UNIRUN.EXE resides (\$BASE\Cobol).
RPT	Path where the report files will be generated.
dd_UNIDF	Path and file name of the definition database.
dd_UNIHF	Path and file name of the history database.
JOBVAR	Type the path and a file name of your choice for the JOBVAR environment variable file. The file is dynamically assigned when the job is run.

Run-time Parameters (PARM options)

Example:

PARM=CYC#00890815,NOEXTR,NOSORT

Parm	Specified as:
Job name	PARM=JNAMjjjjjjj where jjjjjjj is the 1- to 8-character job name.
Step name	PARM=STEPssssssss where ssssssss is the 1- to 8-character step name.
Job-step qualifier	PARM=JQqq where qq is the 2-character JSQ.
Cycle Number	PARM=CYC#nnnnnnnn where nnnnnnnn is the 8-digit Cycle Number.
Run Number	PARM=RUN#nnn where nnn is the 3-digit Run Number.
Control Field	PARM=CN TL01Xr where r is the rule set number for Alternate Balancing Rules.
Store History	PARM=ROPT=abcdefgh where a is the store history override option (Y, N, or I). Y = Store history every time the balancing job runs; N = Never store history; I = Store history only when the ACR/Summary job is in balance. Positions b-h are reserved. Use a total of seven dots to fill positions b-h so as not to override the options they control. For example, 'ROPT=Y.....'

D ■ Using ACR/Summary for UNIX or Linux in Batch

Script Quick-Reference with Page Numbers

Parm	Specified as:
Rerun	PARM=RERUN where RERUN causes the UAC2000 program to rerun the last cycle processed. When cycles other than the last cycle need to be rerun, use the PARM=CYC#nnnnnnnn, option.
SYSOUT	PARM=SYSOUT where SYSOUT causes the UAC2000 program to produce a SYSOUT report even when no #U messages are created. If you specify the SYSOUT parameter, an informational header message will be generated whenever the job runs. The message will say, "USY0001: INFOGIX JOB STARTED."
Bypass	PARM=BYPASS causes the UAC2000 program to bypass balancing.
File Work Area Size	PARM=COREnnnn where nnnn is the size in 'K' of the area that the File Interface Program will use in obtaining storage for the file work area. When deciding whether to use this parameter, consider the following: <ul style="list-style-type: none">■ It is useful only in situations where one execution of uac2000.sh will result in more than one balancing run being done (i.e. if multiple JSQs will be executed).■ It will help avoid fragmentation of storage in those cases where multiple balancing runs are performed during one execution of uac2000.sh.■ If you use this parameter, nnnn should be large enough to accommodate the largest work area needed by the files that will be processed. To find the largest area needed, use the following formula on every file that will be processed during the run: $\text{nnnn} = (\text{rrrr} * (\text{sss} + 8)) / 1024$ where rrrr is the number of records that will be specified on the Basic File Information Record and sss is the length of a record on that file. Set your region size according to one of the following criteria: If no Access Mode 2 files will be accessed during this execution, then: Region size = core parameter size + 1000K If Access Mode 2 files will be accessed during this execution, then: Region size = 2*(core parm size) + 1000K

Parm	Specified as:
Control Report XML	PARM='CRXMLn' where n is one of the following: 1 - do not generate the Control Report in XML format 2 - generate, write XML to file (Copy 1) 3 - generate, write to database (Copy 1) 4 - generate, write to file (Copy 2) 5 - generate, write to database (Copy 2)
User Report XML	PARM='URXMLn' where n is one of the following: 1 - do not generate the User Report in XML format 2 - generate, write to file 3 - generate, write to database
Free-Form Report XML	PARM='FRXMLn' 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, write to database
Debug options	PARM='DEBUG=abcdefgh' where: a is the XML Writer debug option (Y, N, blank or (.)). b is the Visibility API debug option (Y, N, blank or (.)). c is the Infogix Assure History debug option (Y, N, blank or (.)). d-h are reserved. Processing options: Y = Debug on. N = Blank. (.) = Debug off. Use a total of five dots to fill positions d-h to avoid overriding the options they control. Example, DEBUG=YYY.....

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 statement from the Input Sources section of the script using a #.

Concatenation of Input Sources

In access modes 6 and 7, 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 `SWBIOBUFF` parameter.

External Translation Tables

When using external translation tables, you must set the file IDs and specify the paths of the files the table will use.

uac3000.sh - Balancing in Direct Input Mode

This script is used to run balancing in direct input mode. For more information on direct input mode, see “Selecting an Application Interface Mode” on page 38.

Refer to “Editing the Scripts” on page 350 and to the following tables when editing this script.

Required Environment Variables Table

Variable	Definition
ACRSWB	Path where UNIRUN.EXE resides (\$BASE\Cobol).
RPT	Path where the report files will be generated.
dd_UNIDF	Path and file name of the definition database.
dd_UNIHF	Path and file name of the history database.

uacdfb.sh - Build Definitions

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. Types of transactions you can select include:

- All types
- Jobs
- Files
- Internal Translation Tables/Cycle Tables
- External Translation Tables
- History Analysis definitions
- Jobs and Associated Files

This utility is useful for making mass definition changes, creating an audit trail, transferring definitions between platforms, etc.

Refer to “Editing the Scripts” on page 350 and to the following tables when editing this script.

Required Environment Variables Table

Variable	Definition
ACRSWB	Path where UNIRUN.EXE resides (\$BASE\Cobol).
RPT	Path where the report files will be generated.
dd_UNIDF	Path and file name of the definition database.
dd_UNIUPDT	Path and file name of the batch transactions file.

uacdf.sh - 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. Types of definitions you can specify include:

- All definitions (blank)
- Jobs (J)
- Files (F)
- Internal Tables (T)
- External Tables (X)
- History Analyses (S)
- Jobs and Associated Files

Refer to “Editing the Scripts” on page 350 and to the following tables when editing this script.

Required Environment Variables Table

Variable	Definition
ACRSWB	Path where UNIRUN.EXE resides (\$BASE\Cobol).
RPT	Path where the report files will be generated.
dd_UNIDF	Path and file name of the definition database.

uacdfx.sh - Cross-Reference Internal Items

This utility produces a report listing the definitions in the definition database for a specified job ID or a range of Job IDs. For each internal or extended internal item defined for that job ID, the report lists the item's description and any job ID that has a related item number. Related items can include history items, calculated items, rules, field items, and history analysis definitions.

Refer to “Editing the Scripts” on page 350 and to the following tables when editing this script.

Required Environment Variables Table

Variable	Definition
ACRSWB	Path where UNIRUN.EXE resides (\$BASE\Cobol).
RPT	Path where the report files will be generated.
dd_UNIDF	Path and file name of the definition database.

uachfc.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.

Refer to “Editing the Scripts” on page 350 and to the following tables when editing this script.

Required Environment Variables Table

Variable	Definition
ACRSWB	Path where UNIRUN.EXE resides (BASE\Cobol).
RPT	Path where the report files will be generated.
dd_UNIHF1	Path and file name of the history database.
dd_UNIHF2	Path and file name of the new history database.

uachfd.sh - Delete History

This utility can perform a logical delete of one or multiple history records by job ID, cycle, and date. Wildcards can be used in the **From Job ID** field to select multiple job IDs. This can be useful when a job ran that was not scheduled to run, when a job ID is no longer needed in the history database, etc.

NOTE: The physical history records are not removed from the file until the Reorganize History script is executed.

Refer to “Editing the Scripts” on page 350 and to the following tables when editing this script.

Required Environment Variables Table

Variable	Definition
ACRSWB	Path where UNIRUN.EXE resides (\$BASE\Cobol).
RPT	Path where the report files will be generated.
dd_UNIHF	Path and file name of the history database.

uachfe.sh - 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. For more information, including the layouts that can be generated, see “Extract History” on page 260.

Refer to “Editing the Scripts” on page 350 and to the following tables when editing this script.

Required Environment Variables Table

Variable	Definition
ACRSWB	Path where UNIRUN.EXE resides (\$BASE\Cobol).
RPT	Path where the report files will be generated.
dd_UNIHF	Path and file name of the history database.
dd_UNIHFV	Path and file name of the exact file.

uachfl.sh - List History

This utility generates a selection of reports regarding the records in the history database. For descriptions and examples of the reports that can be generated by this utility, see “Database History and Statistics Report - UNIHLR.RPT” on page 318.

Refer to “Editing the Scripts” on page 350 and to the following tables when editing this script.

Required Environment Variables Table

Variable	Definition
ACRSWB	Type the default working directory.
RPT	Path where the report files will be generated.
dd_UNIHF	Path and file name of the history database.

uachfr.sh - Reorganize History

This utility physically deletes all logically deleted records from this history database. Run this utility regularly to ensure optimum processing efficiency.

Refer to “Editing the Scripts” on page 350 and to the following tables when editing this script.

D ■ Using ACR/Summary for UNIX or Linux in Batch

Script Quick-Reference with Page Numbers

Required Environment Variables Table

Variable	Definition
ACRSWB	Path where UNIRUN.EXE resides (\$BASE\Cobol).
RPT	Path where the report files will be generated.
dd_UNIHF1	Path and file name of the history database.
dd_UNIHF2	Path and file name of the reorganized history database.
dd_UNIDF	Path and file name of the definition database.

uachfrp.sh - Recalculate/Reprint ACR Report

This utility reprints the (Balancing) Control Report (UNIACR.RPT) for a job using control values stored in the definition and history databases. This enables you to generate the report without rerunning the job.

Because the definitions used are the ones that exist when the utility is being run, a possibility exists that the results this utility yields will differ from the original balancing run. If someone has modified the definitions being used and updated the history control values since the original run, the Control Report will show results that differ from those of the original run.

Refer to “Editing the Scripts” on page 350 and to the following tables when editing this script.

Required Environment Variables Table

Variable	Definition
ACRSWB	Path where UNIRUN.EXE resides (\$BASE\Cobol).
RPT	Path where the report files will be generated.
dd_UNIHF	Path and file name of the history database.
dd_UNIDF	Path and file name of the definition database.

uachft.sh - Transfer History (Selected Records)

This utility transfers selected records from an existing history database directly to another file. You can select a range of histories based on job ID and/or date and time and copy them to another database. This can be useful when copying a test job into production or vice versa.

Refer to “Editing the Scripts” on page 350 and to the following tables when editing this script.

Required Environment Variables Table

Variable	Definition
ACRSWB	Path where UNIRUN.exe resides (\$BASE\Cobol)
RPT	Path where the report files will be generated
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_UNIHFV	Path and file name of the work file.

uachftu.sh - Unload History

This utility writes the history database records to a work file.

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, use the Reload History script.

Refer to “Editing the Scripts” on page 350 and to the following tables when editing this script.

D ■ Using ACR/Summary for UNIX or Linux in Batch

Script Quick-Reference with Page Numbers

Required Environment Variables Table

Variable	Definition
ACRSWB	Path where UNIRUN.exe resides (\$BASE\Cobol)
RPT	Path where the report files will be generated
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.

uachftr.sh - Reload History

This utility is used after you use the Unload Records utility. It enables you to select a range of histories based on job ID and/or date and time and reload them (from the work file to which they were unloaded) into the history database.

Refer to “Editing the Scripts” on page 350 and to the following tables when editing this script.

Required Environment Variables Table

Variable	Definition
ACRSWB	Path where UNIRUN.exe resides (\$BASE\Cobol)
RPT	Path where the report files will be generated
dd_UNIHF2	Path and file name of the target history database.
dd_UNIDF	Path and file name of the definition database.
dd_UNIHFW	Path and file name of the work file.

uachfu.sh - Update History

This script updates (i.e., replaces) one or more control values in the history database for a specified job ID and cycle ID. 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.

Refer to “Editing the Scripts” on page 350 and to the following tables when editing this script.

Required Environment Variables Table

Variable	Definition
ACRSWB	Path where UNIRUN.EXE resides (\$BASE\Cobol).
RPT	Path where the report files will be generated.
dd_UNIHF	Path and file name of the history database.

uacinit.sh - Initialize Databases

This utility initializes (creates or re-creates) the definition and/or history database. Both databases must be initialized before you can begin using ACR/Summary for the first time.

Reinitializing a database erases the contents.

Refer to “Editing the Scripts” on page 350 and to the following tables when editing this script.

Required Environment Variables Table

Variable	Definition
ACRSWB	Path where UNIRUN.EXE resides (\$BASE\Cobol).
RPT	Path where the report files will be generated.
dd_UNIDF	Path and file name of the definition database.
dd_UNIHF	Path and file name of the history database.

D ■ Using ACR/Summary for UNIX or Linux in Batch

Script Quick-Reference with Page Numbers

Run-time Parameters (PARM options)

Option	Use
-- blank --	Initialize both the definition and history databases.
DF	Initialize the definition database only.
HF	Initialize the history database only.

uacmr01.sh - Print Management Report

This utility allows you to print a report that summarizes balancing history information for management purposes.

Refer to “Editing the Scripts” on page 350 and to the following tables when editing this script.

Required Environment Variables Table

Variable	Definition
ACRSWB	Path where UNIRUN.EXE resides (\$BASE\Cobol).
RPT	Path where the report files will be generated.
dd_UNIDF	Path and file name of the definition database.
dd_UNIHF	Path and file name of the history database.

uacupdt.sh - Update Definitions

This utility loads your definitions for the current job, table, or history analysis document into the definition database. This is necessary in order to use the definitions in ACR/Summary processing.

Refer to “Editing the Scripts” on page 350 and to the following tables when editing this script.

Required Environment Variables Table

Variable	Definition
ACRSWB	Path where UNIRUN.EXE resides (\$BASE\Cobol).
RPT	Path where the report files will be generated.
dd_UNIDF	Path and file name of the definition database.
dd_UNIINP	Path and file name of the batch definitions file.

uan2000.sh - History Analysis Reporting

The History Analysis Reporting script extracts, analyzes, and reports on selected history data. This can be used to spot fluctuations in your balancing processes. For more information, see “Using History Analysis” on page 247.

Refer to “Editing the Scripts” on page 350 and to the following tables when editing this script.

Required Environment Variables Table

Variable	Definition
ACRSWB	Path where UNIRUN.EXE resides (\$BASE\Cobol).
RPT	Path where the report files will be generated.
dd_UNIDF	Path and file name of the definition database.
dd_UNIHF	Path and file name of the history database.

uaccsv.sh - Create Comma Delimited Text File

This utility converts an extract file that has been created using the Extract History utility to an ASCII text file containing sequential fields separated by commas and text values surrounded by quotation marks. For more information, see “Create Comma Delimited Text File” on page 264.

Refer to “Editing the Scripts” on page 350 and to the following tables when editing this script.

Required Environment Variables Table

Variable	Definition
ACRSWB	Path where UNIRUN.EXE resides (\$BASE\Cobol).
RPT	Path where the report files will be generated.
dd_UNIDF	Path and file name of the definition database.
dd_UNIHF	Path and file name of the history database.

Migrating Test Definitions to Production

Overview

What are Definitions?

Definitions are data balancing 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 balance the data as desired.

Testing Definitions

Once you have set up definitions, you must ensure that they balance 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

Use the instructions in “uacdfb.sh - Build Definitions” on page 357 to execute the script, which will build batch transaction cards directly from the test definition database.

Updating the Production Definition Database

Use the instructions in “uacupdt.sh - Update Definitions” on page 366 to edit and run the udsupdt.sh script, which will build batch transaction cards directly from the test definition database.

When you executed the uacdfb.sh script in the previous section, a file was created and a file name was chosen. As you execute the uacupdt.sh script to update production, the file name used for building batch definitions should be used as input to the variable UNIINP.

D ■ Using ACR/Summary for UNIX or Linux in Batch

Migrating Test Definitions to Production

Non-Biased Variance and Standard Deviation Computations

The history analysis feature includes two computation types available in the **Computation** field on the Summary Column Information and Summary Row Information dialog boxes 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.

This appendix contains the following sections:

- “Field Descriptions and Formulas” on page 371
- “Examples” on page 373

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 373, 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 Σ 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:

E ■ 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.

E ■ Non-Biased Variance and Standard Deviation Computations

Examples

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:

$$\text{Variance} = \frac{n \sum x^2 - (\sum x)^2}{n^2}$$

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:

$$\text{Variance (Non - Biased)} = \frac{n \sum x^2 - (\sum x)^2}{n(n - 1)}$$

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

E ■ 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

E ■ Non-Biased Variance and Standard Deviation Computations

Examples

Using ACR/Workbench

This appendix documents ACR/Workbench for Summary.

ACR/Workbench for Summary provides the same user interface and all of the functionality of ACR/Summary 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 balancing 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 Summary and ACR/Summary Client are identical, this appendix documents only functionality that is specific to ACR/Workbench for Summary. For all other functionality, see the remainder of this manual.

This appendix contains the following sections:

- “Restrictions for Mainframe Users” on page 379
- “ACR/Workbench for Summary Configuration Modes” on page 380
- “Downloading and Uploading Between z/OS and ACR/Workbench” on page 381
- “Importing Definitions and Input Sources from z/OS” on page 382
- “ACR/Workbench for Summary Tutorial” on page 385
- “Features Applicable only to ACR/Workbench for Summary Jobs Uploaded to z/OS” on page 386

Restrictions for Mainframe Users

If you upload and run definitions created through ACR/Workbench on the mainframe, the following features will not apply:

- E-mail notification
- Web Extraction
- Web Publishing

ACR/Workbench for Summary Configuration Modes

ACR/Workbench can be used in Local mode, Windows mode, UNIX mode, or Linux mode, depending on your licensing.

The Windows, UNIX, and Linux configuration modes, which are also available in ACR/Summary Client, are documented in “Configuring ACR/Summary Client” on page 34.

Local Mode

In Local mode, all of the ACR/Summary components and files that will be used reside and run on the PC. In this mode you can set up, test, and run balancing 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 balancing 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 108 and/or Web publication as described in “Distribution of Reports” on page 290.

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

Downloading Definitions and Input Sources from z/OS

Types of Files to Download

You may want to download the following from the mainframe to ACR/Workbench:

- Existing ACR/Summary definitions for modification and testing.
If you download definitions, you must import them after downloading them. For more information, see “Importing Definitions” on page 243. Files having the .def extension that have been downloaded from z/OS 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 use ACR/Workbench to streamline the process of creating file definitions by using the mark-and-capture method, which is described in “Preparing to Use Mark-and Capture” on page 116. To download from the mainframe for this purpose, the file should be downloaded in text (ASCII/CR/LF) format.

Download Procedure

To download a file from z/OS, 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 which points to the directory where your 3270 emulation SEND and RECEIVE commands are located.

In ACR/Workbench, select **File > Download** while using a Local configuration to display the File Download dialog box with the fields required for downloading from z/OS.

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 388.

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. For more information, see “UNINTC - Not (¬ ASCII 173) to Caret (^) utility” on page 387.

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.

Importing Definitions and Input Sources from z/OS

General instructions for importing are provided in “Importing Definitions” on page 243. 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 for Summary and ACR/Summary for the Mainframe

ACR/Workbench for Summary differs from ACR/Summary 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/Summary 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/Summary for z/OS to ACR/Workbench, 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 Summary Tutorial

This tutorial shows you how to set up and run an ACR/Summary balancing job in Local mode using ACR/Workbench.

Preparing for the Tutorial

Before you begin this tutorial, you need to:

1. Start ACR/Workbench for Summary. **Select Start > Programs > Infogix > ACR/Workbench for Summary.** (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 balancing job for the tutorial.
2. Familiarize yourself with the product interface, which is the same as that of ACR/Summary Client. See “Using the ACR/Summary Client Interface” on page 26.
3. 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 integrity checking is turned on. Before a database update, an integrity check scans your definitions for inconsistencies and missing data and displays a window reporting success or identifying problems.
Select **Options** from the menu and ensure that there is a check mark next to **Integrity check**.
5. If you have not done so, follow the instructions in “Initializing the Definition and History Databases” on page 36.
6. Verify that you have access to the sample reports that will be used as input sources in the tutorial. Locate Updt0601.rpt and Summ0601.rpt. In Local mode, these reports should be in C:\Infogix\Summary32\sample.

Determining Your Balancing Requirements

Refer to “Determining Your Balancing Requirements” on page 46 for instructions.

Creating and Running the Balancing Job

The remainder of the tutorial is almost identical to the tutorial in Chapter 2, “Tutorial” on page 45.

Note: Skip all sections for UNIX or Linux.

- Follow the instructions in “Complete the Basic (Job) Information” on page 49.
- Follow the instructions in “Define the First Input Source” on page 50 except that when you complete path and file name in the **File Name** field in the Name dialog box, the file upd0601.rpt will be in the Sample folder on the local PC.
- Follow the instructions in “Define the Second Input Source” on page 54 except that when you complete path and file name in the **File Name** field in the Name dialog box, the file summ0601.rpt will be in the Sample folder on the local PC.
- Follow all of the remaining instructions for the tutorial, beginning with “Define the First Balancing Rule” on page 56 and ending with “Viewing the Output Reports and Handling Errors” on page 61.

Features Applicable only to ACR/Workbench for Summary Jobs Uploaded to z/OS

In addition to the features of ACR/Summary Client described in the remainder of this manual, the following features are available for ACR/Workbench jobs that will be uploaded and run on z/OS.

- Application interface modes
 - Program Interface Mode. This mode uses simple source code additions to invoke ACR/Summary directly from an application program. It is typically used for new applications or applications whose programs are being developed or modified.
 - Spool monitor mode. This mode enables you to capture spooled output, including console messages, for verification.
- Input Sources

Features Applicable only to ACR/Workbench for Summary Jobs Uploaded to z/OS

- DB2 table. Direct access to a DB2 table, as opposed to using ODBC.
- Direct spool dataset
- Alternate Control Tasks

An alternate control task is an alias for the (primary) job ID defined for the job. It is typically used to submit JCL to run with production job names in a test environment.

To set up an alternate control task, click **New** in the Alternate Control Tasks folder in Control Task (Job) View to display the Alternate Control Task dialog box.
- Automatic Rerun/audit Trail Options

The automatic rerun/audit trail options in the Basic Job Information folder in Control Task (Job) View enable you to set the following options at the job level.

 - The automatic rerun option allows reruns of a job to be made without manual intervention using the UACARUN program of ACR/Summary for z/OS.
 - The audit trail option enables you to store all of the reruns for the job in the history database, including the control values and balancing results. The stored reruns remain linked to the active history and can be viewed by running the List History utility.
- Set Return Code Abend option

The Set Return Code dialog box enables you to specify whether the highest return code generated will actually be set and, in Local mode only, to specify whether an abend will be forced.
- Utilities
 - UNINTC - Not (¬ ASCII 173) to Caret (^) utility

UNINTC is usually used when transferring files from the PC 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 for using this utility are as follows:

 - 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.

F ■ Using ACR/Workbench

Features Applicable only to ACR/Workbench for Summary Jobs Uploaded to z/OS

■ UNICTN - Caret (^) to Not (¬ ASCII 173) utility

UNICTN is usually used when transferring files from the mainframe to the PC. It converts the caret symbols (^) in a file to not symbols (¬ ASCII 173) if your 3270 emulator does not perform this conversion.

The steps for using this utility are as follows:

- a. Download the batch definition transactions to 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.

Glossary

A

access mode

This file organization method determines how the input source file is read and processed.

- **Access Mode 1** selects the first X records from the input source and places them in the file access area, a temporary storage area.
- **Access Mode 2** selects the last X records from the input source and places them in the file access area, a temporary storage area.
- **Access Mode 3** selects records based on file keys and places them in the file access area, a temporary storage area.
- **Access Mode 4** selects the nth record and the records after it and places them in the file access area, a temporary storage area.
- **Access Mode 5** processes all records in an input source using selection groups and derives either a total of a particular amount, a tally of the records, or the hash value of the file.
- **Access Mode 6** processes all records in the input source using selection groups. All criteria must be met on the same record.
- **Access Mode 7** processes all records in the input source using selection groups. All selection criteria must be met, but not on the same record.

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 balancing rules

This feature allows you to define additional sets of balancing rules for a single job.

alternate balancing rule set

A numeric identifier indicates that an alternate balancing rule is part of a rule set. Alternate balancing rule sets are useful when the same job is run by different departments or for different processing periods.

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 history analysis rule.

application interface mode

This is the bridge between your application and the ACR/Summary application. The application interface mode offers different processing modes to extract data from your application into ACR/Summary.

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.

automatic cycle processing

This method of cycle processing allows the system to generate the cycle number or the entire cycle ID automatically, usually based on the system date.

B

balancing

Balancing uses file definitions to extract control values from application files and reports, applies rules that calculate, compare, and verify the extracted control values, and produces the requested balancing reports.

balancing extract file

This output file contains formatted data and balancing results from a job run. The results in the extract file can be passed to other platforms or downloaded for use in a PC application.

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.

base model

In modeling, the base model is the balancing job, template, or input source that contains the definitions to be modeled.

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.

begin reference record

This file definition type, for access modes 1 through 4, marks the beginning of a subset of records, called a reference area, within the file access area to be scanned for the data to extract.

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 or on Free-form Reports and User Reports.

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.

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 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 generation

Cycle generation determines how the cycle ID is assigned, either manually or automatically.

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 *ccyyymmdd* format.

cycle override

Cycle override is an option that passes cycle IDs to the balancing function, when implemented through the options file. This option is typically used during a rerun to override a cycle ID that was set by another method. When in effect, if a cycle override for the current job ID is found in the options file, the associated cycle ID is set as the current cycle ID.

cycle processing

This process generates or verifies cycle IDs. Three types of cycle processing are available: cycle reference processing, automatic cycle processing, and cycle tables.

cycle reference processing

In this method of cycle ID generation, the job gets its cycle number from a previously executed reference job. This type of processing is useful when a job stream spans multiple system dates and you want to keep the same cycle number during the job stream.

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. In balancing, a cycle table can verify the cycle number before assigning it to a job run.

D**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 and history 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.

direct input mode

This application interface mode lets you specify the control values to execute a run. This mode is used to process data that is not available on your system, to test new balancing jobs, and to prime the history database.

DSN

The dataset name (DSN or DSName) represents the catalogued name of a particular computer file.

E**embedded key**

This file definition type, for access modes 1 through 4, locates a record by searching for a text string within the selected file access area.

end reference record

This file definition type, for access modes 1 through 4, marks the end of a subset of records, called a reference area, within the file access area to be scanned for the data to extract.

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. You can create a formatted extract file that contains information extracted during balancing and use the information to generate correction transactions.

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.

field item

This file definition type, for access modes 1 through 4, or for relational database tables, defines the position and length of the data to extract from the specified record. The extracted data can be stored in an internal item, extended internal item, or extraction variable.

file access area

File access area is a work area for records selected from the input source. Access modes 1 through 4 use the file access area to avoid processing all the records in the input file.

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 key

This file definition type, for access mode 3, specifies a character string (the key value) and the position of the first character of the key value. The system searches the input file, and if it finds the character string, selects individual records or a block of records and places them in the file access area.

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.

FNQ

See file name qualifier.

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.

H

hash record

This file definition type, for access mode 5, computes the hash value of a sequential file and stores the result in an internal item or extraction variable. Hashing can be used to determine whether two files are identical.

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

A history analysis definition specifies how to set up the processes and reports to analyze and report on data from the history database.

history analysis document

See history analysis definition.

history analysis ID

This is a unique name, up to 16-characters long, for the set of history analysis process definitions.

History Analysis Report

This report shows the results of a history analysis job.

history analysis rule

A history analysis rule compares values in the analysis window with values in the analysis base.

History Data Detail Report

This report shows history records and their values for a range of jobs, steps, and qualifiers.

history database

This database is a relative-record 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.

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 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.

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.

job-step qualifier (JSQ)

The job-step qualifier is an optional 2-character identifier that is the last part of the job ID. Job-step qualifiers distinguish multiple invocations of a balancing run with the same job and step name.

JSQ

See job-step qualifier.

L

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. Messages can be direct or indirect.

model definition

See modeling.

model user

In modeling, this is the file or job that uses the definitions from the base model.

modeling

This feature allows file definitions and job definitions stored in the definition database to be referenced by one or more balancing jobs. Modeling reduces the need to recode definitions and ensures consistency between balancing jobs that have the same requirements.

O

options file

The options file (UNIOF) is a file that you create manually to use the cycle override and history insert features, which are related to cycle processing.

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.

P**page/line record**

This file definition type, for access modes 1 through 4, selects records in fixed-block addressing (FBA) format based on a report page number and line number.

process control

This table build rule specifies the action to take after a selection group has been satisfied.

processing error

See error message.

program interface mode

This application interface mode requires a COBOL compiler. It uses simple source code additions to your application to extract the control values and call the appropriate program interface program. This mode is best used for new applications or applications in a state of development or modification.

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 that indicates that the input file is processed more than one time for a balancing run. See also job-step qualifier.

R**RDBMS record**

This file definition type, for relational database tables, provides criteria to search the source table for the desired rows. You can then use field item definitions to extract or accumulate column values from the selected rows.

Recap Report

This user-defined report summarizes the results of multiple jobs.

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.

reference area

This is a logical area that is between the beginning and end reference points. The reference area can include all or a subset of records in the file access area.

reference record

See begin reference record and end reference record.

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, available in all access modes except 5, 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 physically deletes all logically deleted records from the history database.

reset reference record

This file definition type, for access modes 1 through 4, resets the beginning or ending point of the reference area to the entire file access area.

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.

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 set

See alternate balancing rule set.

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.

sequence number

The sequence number is an automatically- assigned unique identifier for the selection group and file definition combination. In modeling, you can change this number to correspond with the model job.

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.

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, 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.

tally item

See tally record.

tally record

This file definition type, for access mode 5, counts all records or counts records that meet specific criteria and associates the resulting count with an internal or extended internal item.

total item

See total record.

total record

This file definition type, for access mode 5, sums the values in a specified field and places the sum in an internal or extended internal item.

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.

UNINTC utility

This utility converts not symbols to caret symbols.

UNIOF

See options file.

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/Summary 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 a history item based on the run date and time or the cycle ID of another job or both. Variable cycle processing accommodates situations where the run date and time of other jobs provide the basis for specifying history items in the current job.

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 241

@FILETMP 241

A

access mode

access mode 1 122

access mode 2 123

access mode 3 124

access mode 4 126

access mode 5 126

access mode 6 127

access mode 7 132

selecting 120

access mode 6 and 7

differences 132

similarities 132

access mode 7

when to use 134

Access User Exit dialog box 83

ACR Test Report 437

ACR/Summary

components 17

overview 14

ACR/Summary Client

access 25

configuration 34

exiting 25

input types supported 16

tutorial 45

ACR/Workbench 493

configuration modes 494

features for jobs uploaded to z/OS 500

tutorial 499

advance record 129

alternate balancing rule 100

alternate control task 112, 501

analyzing your balancing needs 42

application interface mode

file interface mode 20

O mode 20

overview 20, 38

program interface 500

selecting 42

spool monitor 500

Automatic rerun/audit trail options 501

B

balancing function 20, 64

balancing job

see job

basic job information

options 75

setting up 75

batch

see Windows, running in batch, UNIX,
running in batch, and Linux, running
in batch

batch definition transaction

see also transaction layouts

file type (.def) 22

overview 279

purging 283

record

defining a job with 15

format 280

transaction code 280

transaction ID 280

transaction key 280

transaction list 284

transaction number 281

transaction sequence 281

transaction type 281

batch file
 list of 444
 locating and copying 442
 tab-delimited input sources 442
 uac2000.bat 446
 uac3000.bat 449
 uaccsv.bat 459
 uacdfb.bat 449
 uacdfi.bat 450
 uacdfx.bat 451
 uachfc.bat 451
 uachfd.bat 452
 uachfe.bat 452
 uachfl.bat 453
 uachfr.bat 453, 456
 uachfrp.bat 454
 uachft.bat 454
 uachfu.bat 455, 456
 uacinit.bat 457
 uacmr01.bat 458
 uacupdt.bat 458
 uan2000.bat 458
batch options 113
begin reference record 140
Build Batch Transactions utility 252

C

calculated item 43, 97
 conditional 98
 setting up 99
Caret to Not Utility 495, 502
century values
 calculation 156
 verifying 157
commas
 see print commas 174
comments, job 85
components of ACR/Summary 17
conditional calculated item 98
configuration 16, 34
 see also Local mode, UNIX mode, Linux mode, and Windows mode
 setting up 34
Control Report 410
Control Report XML Option dialog box 85

Control Report XML Option, UNIX or Linux 175
Control Task (Job) Name dialog box 76
control task file type (.tsk) 22
Control Update Information Report utility 251
Copy History reports 438, 439
Copy History utility
 copy all records 254
 copy selected records 254
 reload records 255
 unload records 254
Create Comma Delimited Text File utility 260
Cross Reference File Definitions utility 253
Cross Reference Internal Items utility 252
currency format options, UNIX or Linux 169
customer support 24
cycle ID 193
cycle numbers
 as is format 154
 display and printing 157
 extraction 153
 format options 154
 partial date format 154
 storage of 156
cycle override, implementing 197
cycle processing 193
 options 195
 order of precedence 200
Cycle Processing dialog box 85
cycle reference processing 195
cycle table 18, 203
 in a balancing job 203
 in history analysis 204
 setting up 204

D

Database History and Statistics Report 420, 435
Database Initialization utility 250
Database Update utility 250
databases
 see also definition database and history database
 securing 19

- date format options
 - UNIX or Linux 172
 - dates
 - display and printing 157
 - extraction 153
 - format options 154
 - storage of 156
 - DB2
 - access 36
 - DB2 table
 - access 147
 - DB2 table input sources 500
 - defaults for ACR/Summary
 - see also user options
 - overriding 37
 - definition database 18
 - default file name (unidf.dat) 23
 - initializing 36
 - updating 18, 70
 - updating in batch 282
 - Definition Database Cross Reference Report
 - file definitions 434
 - internal items 433
 - Definition Database Update Report 430
 - definitions
 - see also job definitions, file definitions, table definitions, and history analysis definitions
 - downloading 116, 495
 - importing 37, 239
 - opening existing 70
 - saving 70
 - definitions referenced by deleted items
 - deleting 71
 - Delete History utility 255
 - delimited data, direct extraction of 158
 - detail field 130
 - defining 145
 - Detailed File Definitions 424
 - Detailed Job Definitions 422
 - Detailed Spreadsheet (History Analysis) Definitions Report 428
 - Detailed Table Definitions
 - External Translation Table Report 427
 - Internal Translation Table Report 426
 - dialog boxes 26
 - direct input mode 20, 38, 39
 - direct message 109
 - Direct spool dataset input sources 500
 - downloading
 - definitions 116, 495
 - from z/OS 495
 - input sources 495
 - dual history option, UNIX or Linux 168
- ## E
- embedded key record 141
 - end reference record 140
 - environment file
 - global, for Oracle 36
 - environment variables
 - see also environment file
 - setting in scripts 467
 - setting in Windows batch files 444
 - error handling 69
 - extended internal item 88, 90
 - external translation table 18, 219
 - characteristics of 218
 - combining data for translation 228
 - constant translation data 224
 - creating 220
 - example 225
 - physical sequential file as source 224
 - view 32
 - VSAM Key Sequenced file as source 224
 - VSAM ONEROW processing 227
 - Web page as source 225
 - External Translation Table Trace Report 229
 - Extract file
 - record layout 78
 - Extract History utility 256
 - Extraction File Options dialog box 78
 - extraction variable
 - general information 91
 - using in external translation tables 227
- ## F
- field formats for file definitions 136

- field item
 - defining 141
- file definitions 18, 63
 - see also modeling
 - deleting 157
- file interface mode 20, 38, 39
- file key
 - defining 140
 - definition 124
- file types in ACR/Summary Client 22
- filler record 241
- folders 26, 27
- Free-Form Report 181
 - As is option 183
 - creating 182
 - fonts 184
 - internal code representation 187
 - justification options 183
 - sample 187
 - working with objects 183
- Free-Form Report XML Option, UNIX or Linux 175
- freeform SQL statements 148
- G**
- global message option, UNIX or Linux 176
- graphical interface 17
- H**
- hash record 126
 - defining 126, 142
- history analysis 18, 21, 243
 - file type (.had) 22
 - formats 243
 - report samples 246
 - standard deviation 485
 - variance 485
- History Analysis Report 420
- History Data Detail Report 436
- history database 18
 - default file name (unihf.dat) 23
 - initializing 36
- History Database Copy Report 438, 439
- History Database Reorg Report 439
- history insert option
 - implementing 197
- history insert option, UNIX or Linux 174
- history item 96
 - embedded masks. See model definitions, using embedded masks
 - setting up 96
- I**
- icons
 - see toolbar icons
- Import Options dialog box 240, 496
- importing
 - from z/OS 496
- importing definitions 37, 239
- Index of ACR Histories Report 435
- Index of Definition Keys Report 421
- indirect message 109
- Initialize History utility 253
- input source 115
 - see also analyzing your balancing needs
 - for jobs to be uploaded to z/OS 500
 - planning for 42
 - setting up 117
 - for access mode 5 142
 - for access modes 1 through 4 139
 - for access modes 6 and 7 143
 - for relational data 147
 - view 30
- input types supported in ACR/Summary Client 16
- installation manuals 12
- integrity check 260
- internal item 88
 - setting up 90
- internal translation table 18, 218
 - characteristics of 218
 - creating 236
 - sample 237

item number 113
 see also filler record
 display or hide 114
 generation error 113
 moving definitions to and from z/OS 496,
 497
 pattern assignment 113

J

job
 see also Linux, running in batch
 see also UNIX, running in batch
 see also Windows, running in batch
 defining 15
 preparing to set up 64
 running
 through ACR/Summary Client 15
 through ACR/Workbench 15
 with UNIX or Linux scripts 15
 with Windows batch files 15
 setting up 65
 Job Comments dialog box 85
 job definitions 18, 63
 see also modeling
 calculated item 43, 97
 copying and pasting 70
 deleting 71
 extended internal item 88
 extraction variable 91
 history item 96
 internal item 88
 rules 99
 setting up 87
 Job Output Options dialog box 77
 job/step/qualifier 76, 264
 JSQ
 see Job/Step/Qualifier

L

Length of Descriptions dialog box 497, 498
 Linux
 see also Linux mode
 DB2 access 36
 ODBC access 36
 Oracle access 36
 restrictions 12
 running in batch 463
 changing user options 466
 conditionally executing a step 463
 list of scripts 468
 locating scripts 465
 location of environment file 465
 maximum file size 466
 migrating test definitions to production
 483
 preparing to use scripts 465
 setting environment variables 467
 specifying run-time parameters 467
 scripts 15, 463
 setting user options
 bypass processing 177
 Control Report XML Option 175
 date format 172
 Free-Form Report XML Option 175
 global message 176
 history insert 174
 messages 174
 use 22-character numeric format 174
 User Report XML Option 175
 XML DB Error Option 175
 XML DB Message Size 176
 uploading from 37
 user options
 currency format 169
 dual history 168
 negative sign 171
 print commas 174
 return code override 177
 Linux mode 16
 setting up a Linux configuration 35

Index

Linux user options 164

udsuf.dat 164

UNIEXEO 164

UNIXOPT 164

List Definitions reports 420, 421, 422, 424, 426, 427, 428

List Definitions utility 251

List History reports 435, 436

List History utility 253

Local mode 494

M

mark-and-capture method 116, 118

menus 26

message

direct 109

examples 110

global 162

indirect 109

overview 108

processing 108

setting up 108

migrating test definitions to production

UNIX or Linux 483

Windows 460

model definitions

using embedded masks 268

modeling 263

file definitions 268

job definitions 264

terminology 264

N

negative sign format options, UNIX or Linux 171

Not to Caret Utility 496, 501

O

ODBC

access 36, 115, 147

ONEROW processing, VSAM 227

options file 196

cycle override 196

history insert 197

record layout 198

Oracle

access 36, 115, 147

Order of precedence 107

output report file type (.rpt) 22

P

page/line record

defining 141

print commas option, UNIX or Linux 174

Print Control Update Information Report 429

print free-form report

dialog box 84

Print Management Report 440

utility 259

Print Recap Report dialog box 83

processing flow diagram for ACR/Summary 16

program interface mode 500

purging definitions 283

Q

qualifier

see Job/Step/Qualifier

R

recap item 112

setting up 112

Recap Report 414

reformat field

defining 144, 145

reformat record 129

relational data 147

relative record 129

defining in access modes 1 through 4 141

defining in access modes 6 and 7 144

Reorganize History Report 439

Reorganize History utility 255

Report Headings dialog box 77

- reports 407
 - Control Report 410
 - Copy History 438, 439
 - Cross Reference Report for file definitions 434
 - Cross Reference Report for internal items 433
 - Database History and Statistics 435
 - Database History and Statistics Report 420
 - Definition Database Update Report 430
 - Detailed File Definitions Report 424
 - Detailed Job Definitions Report 422
 - Detailed Spreadsheet (History Analysis) Definitions Report 428
 - Detailed Table Definitions
 - External Translation Table Report 427
 - Internal Translation Table Report 426
 - e-mail distribution 408
 - External Translation Table Trace Report 229
 - History Analysis Report 420
 - History Data Detail Report 436
 - History Database Copy Report
 - Summary of Histories Reloaded 439
 - Summary of Histories Transferred 438
 - Summary of Histories Unloaded 438
 - History Database Listing Report 435, 436
 - History Database Reorg Report 439
 - Index of ACR Histories 435
 - Index of Definition Keys Report 421
 - List Definitions reports 420
 - Print Control Update Information Report 429
 - Print Management 440
 - Recap Report 414
 - Reorganize History 439
 - Summary of ACR Histories 436
 - SYSOUT Report 410
 - table of all reports 409
 - Trace Report 416
 - Transaction Listing 437
 - viewing 33
 - Web distribution 408
 - working with 407
- Reprint Control Report utility 259
- reset reference record
 - defining in access modes 1 through 4 141
- restrictions
 - for jobs uploaded to the mainframe 493
 - for Windows, UNIX, and Linux 12
- return code
 - override option, UNIX or Linux 177
- Round Results dialog box 85
- rule 99
 - alternate balancing 100
 - conditional 102
 - active 103
 - evaluation of 103
 - inactive 103
 - processing when item is not found 106
 - reporting status 107
 - setting up 107
 - standard 100
 - active 101
 - inactive 101
- run number
 - reasons for using 194
- run-time parameters
 - specifying in scripts 467
 - specifying in Windows batch files 444

S

script 15
 see also Linux, running in batch
 see also UNIX, running in batch
 list of 468
 locating and copying 465
 preparing to use 465
selection field 127
 defining 144
selection group 127, 128
 examples for access mode 6 130
Set Return Code dialog box (in Basic (job) Information) 84
 abend option 501
spool monitor mode 500
status bar 26
Store History dialog box 77
Summary of ACR Histories Report 436
SYSOUT Report 410
System Messages Report (SYSOUT.RPT) 410

T

table
 see also external translation table, internal translation table, and cycle table
 definitions 18
 file type (.tbl) 22
tally record 126
 defining 142
toolbar icons 26, 28, 29
 control task (job) view 29
 external translation table view 32
 Free-Form Report view 184
 input source view 30
 report view 33
total record 126
 defining 142
Trace Report 68, 416
Transaction Listing Report 437
translation processing 217
translation table 217

tutorial

 ACR/Summary Client 45
 Local mode 499
 Windows, UNIX, and Linux modes 45

U

uac2000.bat - Run Balancing in File Interface Mode 446
uac3000.bat - Run Balancing in Direct Input Mode 449
uaccsv.bat - Create Comma Delimited Text File 459
uacdfb.bat - Build Definitions 449
uacdfi.bat - List Definitions 450
uacdfx.bat - Cross-Reference Internal Items 451
uachfc.bat - Copy History 451
uachfd.bat - Delete History 452
uachfe.bat - Extract History 452
uachfl.bat - List History 453
uachfr.bat - Reorganize History 453
uachfrp.bat - Recalculate/Reprint ACR Report 454
uachft.bat - Transfer History (Selected Records) 454
uachftr.bat - Reload History 456
uachftu.bat - Unload History 455
uachftu.bat - Update History 456
uacinit.bat - Initialize Databases 457
uacmr01.bat - Print Management Report 458
uacupdt.bat - Update Definitions 458
uan2000.bat - History Analysis Reporting 458
unicf.dat 23
UNICTN
 see Caret to Not Utility
unidf.dat 18, 23
UNIEXEO, UNIX or Linux 177
UNIEXIT 83
uniextr.dat 78
unihf.dat 23
UNINTC
 see Not to Caret Utility
uniuf.dat 23, 161

- UNIX
 - see also UNIX mode
 - DB2 access 36
 - ODBC access 36
 - Oracle access 36
 - restrictions 12
 - running in batch 463
 - changing user options 466
 - conditionally executing a step 463
 - list of scripts 468
 - locating scripts 465
 - location of environment file 465
 - maximum file size 466
 - migrating test definitions to production 483
 - preparing to use scripts 465
 - setting environment variables 467
 - specifying run-time parameters 467
 - scripts 15, 463
 - setting user options
 - bypass processing 177
 - Control Report XML Option 175
 - date format 172
 - Free-Form Report XML Option 175
 - global message 176
 - history insert 174
 - messages 174
 - use 22-character numeric format 174
 - User Report XML Option 175
 - XML DB Error Option 175
 - XML DB Message Size 176
 - uploading from 37
 - user options
 - currency format 169
 - dual history 168
 - negative sign 171
 - print commas 174
 - return code override 177
- UNIX mode 16
 - setting up a UNIX configuration 35
- UNIX user options 164
 - udsuf.dat 164
 - UNIEXEO 164
 - UNIXOPT 164
- UNIXOPT 164
- Update History utility 259
- uploading
 - to UNIX or Linux 37
 - to z/OS 496
- Use 22-character Numeric Format dialog box 85
- use 22-character numeric format, UNIX or Linux 174
- user control file (uniuf.dat) 23
- user interface 26
 - see also views, folders, menus, toolbar icons, status bar, and dialog boxes
- user options 37, 161, 494
 - see also UNIX user options, Linux user options, UNIXOPT, udsuf.dat, and UNIEXEO
 - file 23, 161
- User Report 189
 - deleting 188, 191
 - sample 191
 - setting up 189
- User Report XML Option, UNIX 175
- utilities 21, 249
 - Build Batch Transactions utility 252
 - Copy History utility, copy all records 254
 - Copy History utility, copy selected records 254
 - Copy History utility, reload records 255
 - Copy History utility, unload records 254
 - Create Comma Delimited Text File utility 260
 - Cross Reference File Definitions utility 253
 - Cross Reference Internal Items utility 252
 - Database Initialization utility 250
 - Database Update utility 250
 - Delete History utility 255
 - Extract History utility 256
 - Initialize History utility 253
 - integrity check 260
 - List Definitions utility 251
 - List History utility 253
 - Print Management Report utility 259

Index

Reorganize History utility 255
Reprint Control Report utility 259
running 250
Update History utility 259
Utility Transaction Listing Report 437

V

variable cycle processing 96, 97, 205
views 26, 28
 control task (job) 29
 external translation table 32
 Free-Form Report 184
 input source 30
 report view 33
VSAM ONEROW processing 227

W

Windows
 see also Windows mode
 batch files 15
 restrictions 12
 running in batch
 conditionally executing a step 441
 list of batch files 444
 locating batch files 442
 migrating test definitions to production
 460
 setting environment variables 444
 specifying run-time parameters 444
 tab-delimited input sources 442
Windows mode 16
 setting up a Windows configuration 34
 setting user options 161
Workbench
 see ACR/Workbench
write to operator option, UNIX or Linux 174

X

XML DB Error Option, UNIX or Linux 175
XML DB Message Size, UNIX or Linux 176

Z

z/OS
 downloading from 495
 importing from 496
 uploading definitions to 496