

ORACLE®

PEOPLESOFT

PeopleSoft EPM 9.1: Enterprise Performance Management Fundamentals

December 2013

ORACLE®

PeopleSoft EPM 9.1: Enterprise Performance Management Fundamentals
CDSKU epm91pbr4
Copyright © 1999, 2013, Oracle and/or its affiliates. All rights reserved.

Trademark Notice

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

License Restrictions Warranty/Consequential Damages Disclaimer

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

Warranty Disclaimer

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

Restricted Rights Notice

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

Hazardous Applications Notice

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Third Party Content, Products, and Services Disclaimer

This software or hardware and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

Alpha and Beta Draft Documentation Notice

If this document is in preproduction status:

This documentation is in preproduction status and is intended for demonstration and preliminary use only. It may not be specific to the hardware on which you are using the software. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to this documentation and will not be responsible for any loss, costs, or damages incurred due to the use of this documentation.

Contents

- Preface..... xxi**
 - Understanding the PeopleSoft Online Help and PeopleBooks..... xxi
 - PeopleSoft Hosted Documentation..... xxi
 - Locally Installed Help..... xxi
 - Downloadable PeopleBook PDF Files..... xxi
 - Common Help Documentation..... xxi
 - Field and Control Definitions..... xxii
 - Typographical Conventions..... xxii
 - ISO Country and Currency Codes..... xxiii
 - Region and Industry Identifiers..... xxiii
 - Access to Oracle Support..... xxiii
 - Documentation Accessibility..... xxiv
 - Using and Managing the PeopleSoft Online Help..... xxiv
 - PeopleSoft EPM Related Links..... xxiv
 - Contact Us..... xxiv
 - Follow Us..... xxiv
- Chapter 1: Getting Started With PeopleSoft Enterprise Performance Management..... 25**
 - PeopleSoft EPM Overview..... 25
 - PeopleSoft EPM Integrations..... 25
 - PeopleSoft EPM Implementation..... 25
 - Common Elements Used in the PeopleSoft EPM Documentation Set..... 26
- Chapter 2: Understanding PeopleSoft Enterprise Performance Management..... 29**
 - Overview..... 29
 - PeopleSoft EPM Architecture..... 29
 - Extract, Transform, and Load (ETL) in EPM..... 31
 - Operational Warehouse - Staging (OWS)..... 32
 - OWS Core Target Tables..... 33
 - OWS Error Tables..... 34
 - Operational Warehouse - Enriched (OWE)..... 35
 - Tools and Processes Associated with the OWE..... 36
 - OWE Dimension (D00) Tables..... 37
 - OWE Fact (F00) Tables..... 38
 - OWE Temporary Tables..... 40
 - Specialized Reporting Tables..... 40
 - OWE Error Tables (for Profit Manager only)..... 41
 - Multidimensional Warehouse (MDW)..... 43
 - Surrogate Keys..... 44
 - Audit Fields..... 46
 - Data Aggregation..... 46
 - MDW Dimension Tables..... 46
 - MDW Fact Tables..... 47
 - EPM Foundation Toolset..... 48
 - PeopleSoft EPM Analytical Applications..... 51
 - PeopleSoft EPM Warehouses and Reporting..... 53
- Chapter 3: Implementing PeopleSoft EPM..... 55**
 - Understanding EPM Implementation..... 55
 - EPM Installation Tasks..... 56

EPM Core Infrastructure and ETL Setup Tasks.....	57
OWE / MDW Specific Setup Tasks.....	59
Using the EPM Suite of Documentation to Locate Implementation Task Instructions.....	61
Chapter 4: Setting Up EPM Business Rules.....	65
Setting Up EPM Business Rules.....	65
Understanding PeopleSoft EPM Business Rule Setups.....	65
Reviewing Installed Products.....	66
Pages Used to Set Installation Options.....	66
ERP Products Page.....	66
Web Services Page.....	67
Specifying Your EPM Sources.....	68
Page Used to Specify EPM Sources.....	68
Understanding Source System Data and EPM.....	68
Setting Up Country and State Information.....	69
Pages Used to Set up Country and State Information.....	69
Country Table - Country Description Page.....	70
Country Table - Valid Address Page.....	71
Country Statistics Page.....	71
State Page.....	72
Defining Accounting Calendars.....	73
Pages Used to Define Accounting Calendars.....	73
Understanding Accounting Calendar Setup.....	74
Common Elements Used to Define Accounting Calendars.....	75
Calendar Builder Page.....	76
Detail Calendar Page.....	77
Summary Calendar Page.....	79
Summary Budget Period Calendar Page.....	80
Business Day Calendar Page.....	80
Daily Calendar Page.....	81
Budget Calendar Page.....	82
Frequency Definition Page.....	83
Frequency Details Page.....	84
Timespan Page.....	85
Setting Up the Gregorian Calendar [EPM Warehouses].....	87
Pages Used to Set Up the Gregorian Calendar.....	87
Day Dimension Page.....	87
Time Dimension Page.....	88
Setting Up Time Zones [EPM Warehouses].....	89
Defining Units of Measure.....	90
Page Used to Modify UOMs.....	90
Understanding how to Import Unit of Measure Values from Your Source.....	90
Unit of Measure Page.....	90
Defining and Maintaining Dimensions.....	91
Pages Used to Set Up Dimensions.....	91
Understanding Dimensions.....	92
Dimensions Page.....	93
OWE Detail for Dimension Page.....	94
MDW Detail for Dimension Page.....	96
Defining Operator Defaults.....	96
Page Used to Define Operator Defaults.....	96
Operator Defaults Page.....	96

Archiving EPM Data.....	97
Page Used to Archive EPM Data.....	97
Understanding EPM Data Archiving.....	97
Warehouse Archive Project Page.....	98
Setting Up Chart Viewing and Printing Results [Analytical Applications].....	98
Setting Up Ledger-Based nVision Reporting for EPM.....	99
Pages Used to Set Up nVision Ledger-Based Reporting in EPM.....	99
Understanding Ledger-Based Reporting and Setup in EPM.....	99
Ledger Template Page.....	101
Group Definition Page.....	101
Ledgers for a Unit Definition Page.....	102
Chapter 5: Setting Up Currency Rules for EPM.....	105
Setting Up Currency Rules for EPM.....	105
Understanding EPM Multiple Currency Processing Concepts.....	105
Currency and Calculation Types.....	105
Currency Precision.....	107
Revaluation.....	107
Currency as a ChartField.....	108
Differentiating Between the Currency Conversion Process of the EPM Warehouses and the Analytical Applications.....	108
Setting Up EPM Currency Tables.....	109
Currency Code Page.....	110
Currency Codes (Euro) Page.....	111
Setting Up Market Rates for EPM Currency Conversion.....	111
Pages Used to Manage Market Rates.....	112
Defining Market Rate Indexes.....	112
Defining Market Rate Types.....	113
Defining Market Rate Tolerances.....	113
Defining Currency Quotations for EPM Currency Conversion.....	115
Page Used to Define and Maintain Currency Quotations.....	115
Understanding Currency Quotations.....	115
Currency Quotation Method Page.....	116
Establishing Market Rates for EPM Currency Conversion.....	119
Pages Used to Establish Market Rates for EPM Currency Conversion.....	120
Market Rate Page.....	120
Rate Definition Page.....	122
Exchange Rate Detail Page.....	123
Calculating Currency Rates for EPM Currency Conversion.....	125
Pages Used to Calculate Currency Rates.....	126
Understanding Currency Calculations.....	126
Currency Exchange Calculator Page.....	127
Configuring Currency Precision for Currency Conversion.....	128
Understanding Currency Precision.....	128
Activating Currency Precision.....	128
Maintaining Currency Precision by Currency.....	129
Reporting with Currency Precision.....	129
Chapter 6: Setting Up EPM Security.....	131
Understanding EPM Security and Setups.....	131
PeopleSoft Application Security.....	131
EPM Security.....	132
Setting Up EPM Security.....	132

EPM Security Views.....	135
Specifying Field-Level Security Options.....	136
Pages Used to Specify Field-Level Security Options.....	136
Security Options Page.....	137
Security View Names Page.....	138
Request Security Processing Page.....	139
TableSet Security by Permission List Page.....	139
Unit Security by Permission List Page.....	140
Establishing Ledger Security.....	141
Pages Used to Define Ledger Security.....	141
Ledger/Perm List Page.....	141
Specifying Row and Column Level Security.....	142
Pages Used to Establish Role and User-Based Security.....	142
Defining Dimension and Metric Security.....	144
Understanding Role and User Based Security for Dimensions and Metrics.....	144
Define EPM Security Roles Page.....	146
Access To Metric Page.....	147
Role Dimension Access Page.....	147
Individual Selection Page.....	148
Constraint-based Selection Page.....	149
Select Security Column Page.....	151
Tree-based Selection Pge.....	151
User Role Access Pge.....	153
Role Security Summary Page.....	154
Review Role Dimension Access Page.....	156
Access to Metric Page.....	156
User Dimension Access Page.....	156
Individual Selection Page.....	157
Constraint-Based Selection Page.....	157
Tree-Based Selection Page (OWE Dimension Only).....	157
User Security Summary Page.....	157
Review User Dimension or User Metric Access Summary.....	159
Request Security Processing Page.....	159
Establishing Security Bridges for Your Business Intelligence Tool.....	161
Role Collection.....	161
Role.....	161
User Collection.....	163
User.....	164
Dimension Collection.....	165
Dimension.....	166
MetricCollection.....	167
Metric.....	168
JDBCCollection.....	168
Chapter 7: Preparing to Load Source Data Into EPM.....	171
Understanding ETL in EPM.....	171
ETL and the EPM Architecture.....	171
Understanding the Flow of Data Through EPM.....	172
Understanding IBM WebSphere DataStage.....	174
IBM WebSphere DataStage Terminology.....	175
DataStage Server.....	176
DataStage Client.....	176

Key DataStage Components.....	178
IBM Documentation.....	184
Understanding ETL Load Strategies in EPM.....	185
Overview of ETL Load Strategies in EPM.....	185
Incremental Loading with the DateTime Stamp.....	185
Incremental Loading Using Cyclical Redundancy Check.....	186
Incremental Loading Using the Insert Flag and Lookup Validations.....	186
Special Load Requirements.....	190
Understanding Data Validation and Error Handling in the ETL Process.....	191
Understanding the Data Validation Mechanism.....	192
Understanding the Data Completeness Validation and Job Statistic Summary Mechanism.....	195
Understanding OWE Jobs.....	210
OWE Dimension Load Jobs.....	210
OWE Fact Load Jobs.....	212
Understanding MDW Jobs.....	214
MDW Dimension Load Jobs.....	214
MDW Fact Load Jobs.....	215
Chapter 8: Using DataStage Administrator.....	219
Using DataStage Administrator.....	219
Setting DataStage Server Properties.....	220
Setting Project Properties.....	221
Project Properties - General Page.....	221
Project Properties - Permissions Tab.....	226
Project Properties - Tracing Tab.....	227
Project Properties - Schedule Tab.....	228
Project Properties - Tunables Tab.....	229
Project Properties - Sequence Tab.....	230
Chapter 9: Using DataStage Designer.....	231
Using DataStage Designer.....	231
DataStage Designer Overview.....	231
Managing Repository Objects.....	237
Editing Object Properties.....	239
Editing Job and Job Sequence Properties.....	240
Editing Server Routines.....	241
Editing the Stage Type.....	242
Specifying Table Definitions.....	243
Importing and Exporting Repository Components.....	243
Using Table Definitions.....	246
Creating Table Definitions.....	247
Importing Table Definitions.....	248
Building DataStage Jobs.....	248
Using Database and File Stages.....	249
Server Job Database Stages.....	250
Server Job File Stages.....	250
Dynamic Relational Stages.....	250
Processing Stages.....	256
Adding and Linking Stages.....	260
Compiling and Running Jobs.....	261
Criteria Checked when Compiling Jobs.....	261
Specifying Job Run Options.....	262
Editing Job Properties.....	264

Using Expressions.....	265
Accessing Expressions.....	265
Expression Editor Options.....	266
Creating Constraints.....	267
Using Hashed File Stages.....	267
Accessing Hashed File Stages.....	268
Creating Hashed File Lookups.....	269
Using Job Sequencers.....	269
Using DataStage BASIC.....	270
Chapter 10: Using DataStage Director.....	273
Using DataStage Director.....	273
DataStage Director Overview.....	273
DataStage Director Menu Options.....	274
DataStage Director Toolbar Options.....	275
DataStage Director View Options.....	275
Viewing Job Status.....	276
Scheduling Jobs.....	276
Running Jobs.....	276
Setting Parameters.....	277
Setting Limits.....	277
Setting Tracing.....	277
Running, Stopping, Resetting, and Deleting Jobs.....	277
Monitoring Jobs.....	277
Reviewing Job Events.....	278
Managing Aborted and Failed Jobs.....	280
Reviewing the Job Log to Determine Job Errors.....	281
Debugging Aborted and Failed Jobs.....	282
Chapter 11: Setting Up DataStage for EPM.....	283
Understanding IBM WebSphere DataStage Setup and Configuration.....	283
Setting Up DataStage for EPM.....	283
Setting Up DataStage Projects.....	283
Project Structure Considerations.....	283
Creating Projects.....	284
Configuring Environmental Parameters.....	286
Importing Delivered .DSX Files.....	289
Chapter 12: Defining ETL Parameters.....	293
Defining ETL Parameters.....	293
Defining ETL Parameters for the Planning and Budgeting Analytical Application.....	293
Page Used to Define Financial Asset Item Parameters.....	293
Financial Asset Item Setup Page.....	293
Defining ETL Parameters for the EPM Warehouses.....	294
Pages Used to Define ETL Parameters.....	294
Financial Mart Account Setup Page.....	295
Financial Mart GC Cled Setup Page.....	296
Financial Mart GC Flow Setup Page.....	297
Financial Rolling Average Setup Page.....	298
AP Mart Aging Setup Page.....	299
AR Mart Aging Setup Page.....	300
Defining Customer Segment Ranking Parameters for the Customer Segment Data Mart.....	301
Chapter 13: Running Initial Setup Jobs.....	303
Running Initial Setup Jobs.....	303

Verifying ETL Components Have Imported Properly.....	303
Verifying Routines.....	303
Verifying Shared Containers.....	304
Verifying ETL Jobs.....	304
Compiling ETL Jobs.....	304
Verifying ETL Job Compilation.....	304
Compiling Individual ETL Jobs.....	305
Compiling Multiple ETL Jobs.....	305
Running ETL Setup Jobs to Bring Source Data Into EPM.....	306
Running Hash Files Setup Jobs Manually.....	307
Running the Setup - OWS Jobs Manually.....	307
Running Shared Lookup Jobs.....	307
Running Setup - OWE Jobs.....	308
Running Common Dimension Jobs.....	309
Running Common Dimensions Jobs.....	309
Chapter 14: Importing Source Business Units into EPM to Create Warehouse Business Units.....	311
Understanding Warehouse Business Units, TableSet Sharing, and SetID Mapping.....	311
SetIDs and TableSet Sharing.....	311
Mapping Two SetID Based Systems Together and Warehouse Lineage.....	312
Potential SetID Conflicts.....	313
Warehouse Lineage and Source Blueprints.....	315
Understanding Warehouse Business Unit Setup.....	316
Establishing Default SetIDs - Overview.....	317
Establishing Warehouse Sources - Overview.....	317
Establishing Warehouse Lineage - Overview.....	317
Creating Warehouse Business Units with the Business Unit Creation Wizard - Overview.....	318
Creating Warehouse Business Units Manually - Overview.....	320
Working with Invalid or Unused Source Business Units - Overview.....	321
Working with PeopleSoft Human Capital Management (HCM) Source Business Units - Overview.....	321
Reviewing Warehouse and General Ledger Business Unit Properties - Overview.....	323
Creating Collision Maps - Overview.....	324
Prerequisites to Creating Warehouse Business Units.....	324
Running Prerequisite ETL Setup Jobs.....	325
Sizing Tablespaces for the Input Tables.....	325
Creating Backups of Impacted Output Tables (Optional).....	326
Enabling PS Queries for Data Verification.....	326
Verifying the State of Output Tables (Optional).....	327
Validating Business Unit and Set Control Data.....	328
Verifying Source Blueprints.....	330
Establishing Default Set IDs, TableSets, and Warehouse Lineage.....	331
Pages Used to Define Default Set IDs, Warehouse Sources, and Lineage.....	331
TableSet ID Page.....	331
Source Blueprint Page.....	332
Warehouse Lineage Page.....	334
Establishing Warehouse Business Units Using the Business Unit Creation Wizard.....	336
Pages Used to Run the Business Unit Creation Wizard.....	336
Business Unit Creation Wizard - Start Page.....	336
Business Unit Creation Wizard - Review Set Controls from Source Page.....	336
Business Unit Creation Wizard - Address Set Control Collisions Page.....	337
Business Unit Creation Wizard - Validate Mapping Page.....	337

Establishing Warehouse Business Units Manually.....	338
Pages Used to Create Warehouse Business Units Manually.....	338
Warehouse Business Unit Page.....	338
Business Unit Mapping Page.....	340
Validate Mapping Page.....	340
Reviewing Warehouse and General Ledger Business Unit Creation.....	341
Pages Used to Review Your Warehouse and General Ledger Business Units.....	342
Warehouse Business Unit Page.....	342
General Ledger Page.....	342
Defining Collision Mappings (Optional).....	343
Pages Used to Define Collision Maps.....	343
Collision Map Definition Page.....	343
Collision Map Error Report Page.....	345
Chapter 15: ETL Configurations.....	347
Using the Master Sequencer Utility to Create Master Sequencer Jobs.....	347
Using the Master Run Utility to Automatically Run Your ETL Jobs.....	349
Understanding the Master Run Utility.....	349
Running the Master Run Utility.....	351
Supporting UniCode Data.....	353
Converting Incremental Load Jobs to Destructive Load Jobs.....	353
Adding New Environmental Variables.....	359
Chapter 16: Setting Up and Working with Metadata for the Operational Warehouse - Enriched.....	363
Understanding Metadata.....	363
Metadata Terms and Objects.....	364
SQL Object ID.....	366
Metadata Utilities.....	366
Common Elements Used When Setting Up and Working with Metadata for the Operational Warehouse - Enriched.....	367
Finding Metadata Objects.....	368
Page Used to Find Metadata Objects.....	368
Metadata Search Page.....	368
Applying the Hidden Flag to Metadata Objects.....	370
Applying the Hidden Flag to Metadata Objects.....	371
Granting Access to Hidden Metadata Objects.....	371
Setting Up Record Metadata.....	371
Pages Used to Set Up Record Metadata.....	371
Understanding Record Metadata.....	372
Record Metadata Page.....	372
Record Metadata - Field Properties Page.....	374
Record Metadata - Related Metadata Page.....	376
Setting Up Rule Metadata.....	377
Page Used to Set Up Rule Metadata.....	378
Understanding Rule Metadata.....	378
Rule Metadata Page.....	378
Setting Up Tablemaps.....	379
Pages Used to Set Up TableMaps.....	379
Understanding Tablemaps.....	379
Common Elements Used to Set Up TableMaps.....	381
TableMap Page.....	382
Reference Page.....	384

SQL IDs Page.....	386
Setting Up Datamaps.....	386
Pages Used to Set Up Datamaps.....	386
Understanding Datamaps.....	387
Understanding Datamap Setup.....	387
General Properties Page.....	387
DataMap Fields Page.....	388
Constraints Page.....	390
Datamap Wizard Page.....	390
Setting Up Expressions.....	396
Page Used to Set Up Expressions.....	397
Expression Page.....	397
Using Data Sets.....	400
Page Used to Use Data Sets.....	401
DataSet Page.....	401
Setting Up Filters.....	402
Pages Used to Set Up Filters.....	403
Understanding Filters.....	403
Filter Page.....	403
Filter - Selection Criteria Page.....	404
Tree Viewer Page.....	406
Setting Up Constraints.....	407
Pages Used to Set Up Constraints.....	408
Constraint Page.....	408
Constraint - Criteria Page.....	409
Constraint - Details Page.....	410
Setting Up Metric Metadata.....	410
Page Used to Set Up Metric Metadata.....	411
Define Metric Page.....	411
Working with Record Summary Metadata.....	411
Page Used to Work With Record Summary Metadata.....	412
Record Summary Page.....	412
PF Record Summary Page.....	413
Setting Up Report Metadata.....	415
Pages Used to Set Up Report Metadata.....	415
Report Details Page.....	415
Cloning Metadata.....	416
Page Used to Clone Metadata.....	417
Metadata Clone Page.....	417
Using the Metadata Mover Utility.....	418
Pages Used to Migrate Metadata.....	418
Understanding the Metadata Mover Utility.....	418
Metadata Export - Export Definition Page.....	419
Exporting Metadata from a Source Database to a Target Database.....	421
Importing Metadata into the Target Database.....	421
Building Imported Metadata in the Target Database.....	421
Deleting Metadata.....	422
Page Used to Delete Metadata.....	422
Metadata Delete Page.....	422
Creating User-Defined Functions.....	423
OWE Dimension Correction Pages.....	423

Chapter 17: Working with Metadata Utilities.....	425
Viewing Error Messages.....	425
Auditing PeopleSoft EPM Objects.....	425
Page Used to Audit PeopleSoft EPM Objects.....	425
Understanding the Audit Utility.....	425
Audit EPM Objects Page.....	426
Reviewing Metadata Audit Log Error Messages.....	427
Performing Impact Analysis.....	427
Page Used to Perform Impact Analysis.....	427
Impact Analysis Page.....	427
Running Mass Validate.....	428
Page Used to Run Mass Validate.....	429
Mass Validate Page.....	429
Running Mass Compile.....	430
Page Used to Run Mass Compile.....	430
Compile Metadata Changes Page.....	430
Working With Tree Utilities.....	432
Pages Used to Work With Tree Utilities.....	432
Understanding Tree Utilities.....	432
Tree Compare Page.....	433
Tree Compare inquire Page.....	433
Super Tree Page.....	435
Super Tree inquire Page.....	436
Setting Up and Flattening Tree Metadata.....	436
Pages Used to Set Up and Flatten Tree Metadata.....	436
Understanding Trees.....	436
Understanding Tree Flattening in the OWE.....	437
Tree Metadata Page.....	438
Run Tree Flattener Page.....	439
Chapter 18: Setting Up Business Rules for the Operational Warehouse - Enriched.....	441
Setting Up Account Information.....	441
Pages Used to Set Up Account Information.....	441
Understanding Accounts.....	441
Account Types Page.....	442
Accounts Page.....	443
Account Nodes Page.....	444
Specifying Ledger Mapping Defaults.....	445
Pages Used to Set Up Ledger Mapper Defaults.....	445
Common Elements Used to Specify Ledger Mapping Defaults.....	446
Ledger Mapper Defaults Page.....	446
Ledger Mapper Defaults II Page.....	447
Map Ledger to Tree Page.....	448
Defining Ledger Event Codes.....	448
Page Used to Set Up Ledger Event Codes.....	449
Ledger Event Codes Page.....	449
Defining Performance Ledger Templates.....	450
Pages Used to Define Performance Ledger Templates.....	450
Understanding Performance Ledger Templates.....	450
Ledger Template - Ledger Template Page.....	451
Ledger Template - ChartFields Page.....	452
Ledger Template - Edit and Post Variables Page.....	453

Defining Detail Ledgers.....	454
Page Used to Define Detail Ledgers.....	455
Detail Ledger - Definition Page.....	455
Defining Ledger Groups.....	456
Page Used to Define Ledger Groups.....	456
EPM Ledger Group Page.....	456
Processing Roll-Ups.....	458
Pages Used to Run Roll-Up Engines.....	458
Understanding EPM Roll-Up Engines.....	459
Common Elements Used to Process Roll-Ups.....	459
Roll-Up Scenario Page.....	459
Roll-Up Calendar Page.....	460
Roll-Up Business Unit Page.....	461
Setting Up and Running Currency Conversion.....	462
Pages Used to Set Up and Run Currency Conversion.....	463
Understanding Currency Conversion for the Analytical Applications.....	463
Delivered Multicurrency Metadata.....	464
Prerequisites.....	465
Currency Conversion Rule Page.....	465
Currency Conversion Rule Set Page.....	466
Job Conversion Rule Set Page.....	467
Running the Currency Conversion Process.....	468
Message Header Page.....	469
Chapter 19: Setting Up Models and Scenarios.....	471
Understanding Models and Scenarios.....	471
Defining Models and Scenarios.....	472
Pages Used to Define Models and Scenarios.....	473
Models Page.....	473
Scenarios - Definition Page (for Historical Scenarios).....	474
Scenarios - Definition Page (for Forecast Scenarios).....	475
Scenarios - Historical Business Rules Page.....	476
Scenarios - Economic Assumptions Page (for Historical Scenario).....	477
Scenarios - Forecast Business Rules Page.....	479
Scenarios - Economic Assumptions Page (for Forecast Scenario).....	479
Warehouse Business Unit Scenario Definition Page.....	480
Chapter 20: Streamlining Processing with Jobstreams.....	483
Understanding Jobstreams.....	483
Jobstreams.....	483
Jobstream Terminology.....	486
Jobstream Processing.....	487
Jobstream Record Suites.....	487
Jobstream Chunking.....	487
Spawn Process.....	488
Resolver Engine.....	488
Resolver and Chunking.....	488
Process Monitor.....	489
Setting Up Chunking.....	489
Pages Used to Set Up Chunking.....	489
Engine Metadata Pages.....	490
Technical Scenario Page.....	490
Working with Engine Metadata.....	492

Pages Used to Work with Engine Metadata.....	493
Engine Metadata Page.....	493
Engine Metadata - State Variables Page.....	495
Engine Metadata - Rule State Variables Page.....	497
Engine Metadata - Source TableMaps Page.....	497
Engine Metadata - Source Trees Page.....	498
Setting Up Job Metadata.....	499
Page Used to Set Up Job Metadata.....	499
Job Metadata Page.....	499
Setting Up Jobstreams.....	500
Page Used to Define a Jobstream.....	500
Jobstreams Page.....	500
Linking Jobstreams Sequentially.....	502
Working with Record Suites.....	502
Pages Used to Work with Record Suites.....	503
Creating Record Suites.....	503
Record Suite Page.....	503
Jobstream Record Suites Page.....	504
Creating Additional Instances of Temporary Tables for Record Suites.....	504
Removing Extraneous Temporary Tables from Record Suites.....	505
Pages Used to Remove Temporary Tables from EPM.....	506
Understanding the Temporary Table Removal Process.....	506
Clean Up Temporary Tables Page.....	507
Cleanup Log Page.....	508
Running Jobstreams.....	509
Pages Used to Run a Jobstream and Multiple Jobstreams.....	509
Run Jobstream Page.....	509
Run Multiple Jobstream Page.....	511
Jobstream Email Notification Page.....	512
Tracking Jobs.....	513
Pages Used to Track Jobs.....	513
Record Suites Page.....	514
Record Suite History Page.....	514
Jobstream Page.....	515
Jobstream History Page.....	516
Temporary Table Page.....	516
Temporary Table History Page.....	517
Viewing Engine Messages.....	518
Pages Used to View Engine Messages.....	518
Engine Messages - Message Header Page.....	518
Engine Messages - Message Detail Page.....	519
Chapter 21: Setting Up and Using Profit Manager.....	521
Understanding Profit Manager Tools.....	521
Ledger Mapper.....	522
Setting Up Ledger Mapper.....	524
Page Used to Set Up the Ledger Mapper.....	524
Prerequisites.....	524
Ledger to Resource Mapping Page.....	524
Creating Data Manager Rules for the GL Mapper Method.....	526
Configuring Ledger Mappings for New or Configured ChartFields.....	526
Creating, Editing, and Approving Manual Journals.....	529

Pages Used to Create, Edit, and Approve Manual Journals.....	529
Understanding Manual Journals.....	530
Understanding the Journal Copy Feature.....	530
Understanding Reverse Journal Entries.....	531
Journal Entries Page.....	531
Journal Copy Options Page.....	533
Journal Reversal Page.....	535
Journal Attachments Page.....	536
Journal Approval Page.....	536
Processing and Posting Journals.....	538
Pages Used to Process and Post Journals.....	538
Understanding Journal Processing and Posting.....	539
Understanding the PF Journal Modification Component.....	541
Running PF Journal Edit and PF Journal Re-Edit.....	541
PF Journal Modification - Journal Statistic Page.....	542
PF Journal Modification - Journal Correction Page.....	543
PF Journal Modification - Error Details Page.....	544
Journal Cleanup Page.....	544
Ledger Post Page.....	545
Ledger Post - Batch Selection Page.....	546
PF Ledger Unpost Page.....	547
Unposting and Reposting Transactions.....	548
Using Balancing and Reconciliation Features.....	549
Pages Used to Set Up Balancing and Reconciliation.....	549
Understanding PF Reconciliation.....	550
Understanding Job Total Metadata.....	550
Understanding Balance Rules Metadata.....	551
Job Total Metadata Page.....	551
Balance Rules Metadata Page.....	552
PF Reconciliation Page.....	554
Job Totals Page.....	555
Using Ledger Drill Down.....	556
Pages Used to Drill Down on Ledger Data.....	556
Drill Criteria Page.....	557
Drill Details Page.....	558
Reviewing Profit Manager Error Messages.....	558
Correcting Profit Manager Fact Table Errors Using PF Modification.....	559
Understanding Profit Manager Fact Table Error Correction.....	559
PF Summary Statistics Page.....	559
Error Fields Page.....	559
Chapter 22: Using Data Enrichment Tools.....	561
Understanding Data Manager and Allocation Manager.....	561
Data Manager.....	561
Allocation Manager.....	562
Mathematical Model for Data Manager and Allocation Manager Rules.....	564
Setting Up Allocations.....	567
Setting Up Value Objects.....	568
Page Used to Set Up Value Objects.....	568
Value Object Page.....	568
Creating Indices on Allocation Manager Temporary Tables.....	570
Defining Data Manager Rules.....	570

Pages Used to Set Up Data Manager Rules.....	571
Understanding Data Manager Rules and Methods.....	572
Prerequisites.....	577
Data Manager Rules - Define Rule Page.....	578
Data Manager Rules - Define Source 1 Page.....	579
Data Manager Rules - Define Target Page.....	581
Data Manager Rules - DataSet for Source 1 Page.....	586
Data Manager Rules - DataSet for Common Dimensions Page.....	587
Data Manager Rules - SQL Page.....	588
Defining Data Manager Rule Sets.....	589
Pages Used to Define Data Manager Rule Sets.....	590
Assignment Rule Sets Page.....	590
Renaming Data Manager Rules.....	591
Page Used to Rename Data Manager Rules.....	591
Rename Data Manager Rule Page.....	591
Associating Data Manager Rule Sets With a Job.....	591
Pages Used to Associate Data Manager Rule Sets With a Job.....	592
Prerequisites.....	592
Job Association Page.....	592
Running the Data Manager Engine.....	593
Defining Allocation Manager Rules.....	594
Pages Used to Define Allocation Manager Rules.....	594
Understanding Allocation Rule Setup.....	596
Understanding Allocation Manager Fixed Source and Fixed Basis Options.....	597
Calculating Offset Target Information (Optional).....	598
Prerequisites.....	598
Fixed Dimension Setup Page.....	598
Allocation DataMap Setup Page.....	599
Rules - Method Page.....	601
Rules - Source Page.....	604
Rules - Basis Page.....	606
Rules - Source to Basis Relationship Page.....	609
Rules - Target Page.....	610
Allocation Manager Rules - Residual Page.....	611
Rules - Offset Page.....	612
Allocation Template Page.....	615
Defining Allocation Manager Rule Sets.....	618
Pages Used to Define Allocation Manager Rule Sets and Post-Processing Routines.....	618
Understanding Allocation Manager Rule Sets.....	618
Allocation Manager Rule Set Page.....	620
Result Processing Page.....	621
Renaming Allocation Manager Rules.....	624
Page Used to Rename Allocation Manager Rules.....	624
Rename Allocation Rule Page.....	624
Associating Allocation Manager Rule Sets With a Job and Processing Allocations.....	625
Pages Used to Associate Allocation Manager Rule Sets with a Job and Process Allocations.....	626
Prerequisites.....	626
Allocation Manager Rule Set Job Association Page.....	626
Run Allocations Page.....	627
Querying Allocations.....	628
Pages Used to Query Target Allocations.....	629

Allocation Manager Inquiry - Audit Parameters Page.....	629
Allocation Manager Inquiry Results Page.....	631
Allocation Manager Inquiry Results - Drill to Basis Page.....	633
Allocation Manager Inquiry Results - Drill to Source Page.....	635
Chapter 23: Creating XBRL Instance Documents in EPM.....	639
Understanding XBRL Reporting Using PeopleSoft EPM.....	639
XBRL Reporting Requirements and Reference Resources.....	639
Using the XBRL Reporting Features.....	640
Setting Up XBRL Definitions in EPM.....	641
Pages Used to Set up Reporting Definitions.....	641
Understanding XBRL Instance Document Components.....	642
Define Namespace Set Page.....	645
Define Taxonomy Page.....	646
Define Instance Header Page.....	647
Define Reporting Entity Page.....	648
Define Instance Report Page.....	649
Define Report Set Page.....	650
Define XBRL Unit of Measure Page.....	651
Define Context Page.....	652
Uploading and Maintaining XBRL Facts and Taxonomy Elements.....	653
Pages Used to Upload Facts and Taxonomy Elements.....	653
Understanding Data Upload Requirements.....	654
Define File Layout Page.....	657
Load Data Page.....	658
Maintain Taxonomy Elements Page.....	660
Maintain Footnotes Page.....	661
Maintain Instance Facts Page.....	662
Creating XBRL Instance Documents.....	663
Page Used to Create an XBRL Instance Document.....	663
Create Instance Page.....	664
Appendix A: ETL Installation and Implementation Prerequisites and Considerations.....	667
Preparing to Install and Implement IBM WebSphere DataStage.....	667
Sizing Your Database/DataStage Servers.....	667
Determining Server Configuration and Installing IBM WebSphere DataStage.....	669
IBM WebSphere DataStage Implementation Considerations.....	669
Defining a Job Execution Strategy.....	670
Appendix B: ETL Frequently Asked Questions.....	671
IBM WebSphere DataStage ETL Tools.....	671
DataStage Server Installation.....	672
DataStage Client Installation.....	672
DataStage Patches.....	673
ETL Content.....	673
Environmental Variables.....	674
Project Creation, Import, and Export.....	676
Setting Up Development and Production Environments.....	677
ETL Jobs.....	679
Survey Jobs.....	681
ETL Hashed Files.....	682
ETL Routines.....	683
ETL Job Process Flow.....	684
ETL Utilities.....	684

Running ETL Jobs.....	685
Error Handling with ETL Jobs.....	687
ETL Job Validation.....	687
Common ETL Issues.....	688
Configuring Delivered ETL Content.....	691
Appendix C: ETL Reference Documents.....	695
DSX File Import Descriptions.....	695
Environmental Parameters Information.....	695
Parameter and Source Data Files Information.....	695
Routine Descriptions.....	695
Appendix D: Using the PeopleSoft EPM Lineage Spreadsheets.....	697
Understanding the EPM Lineage Spreadsheets.....	697
Viewing Lineage Information.....	700
Finding Lineage Information for a Server Job.....	700
Identifying the List of Jobs to be Run for a Data Mart.....	704
Generating Lineage Information for a Job.....	705

Preface

Understanding the PeopleSoft Online Help and PeopleBooks

The PeopleSoft Online Help is a website that enables you to view all help content for PeopleSoft Applications and PeopleTools. The help provides standard navigation and full-text searching, as well as context-sensitive online help for PeopleSoft users.

PeopleSoft Hosted Documentation

You access the PeopleSoft Online Help on Oracle's PeopleSoft Hosted Documentation website, which enables you to access the full help website and context-sensitive help directly from an Oracle hosted server. The hosted documentation is updated on a regular schedule, ensuring that you have access to the most current documentation. This reduces the need to view separate documentation posts for application maintenance on My Oracle Support, because that documentation is now incorporated into the hosted website content. The Hosted Documentation website is available in English only.

Locally Installed Help

If your organization has firewall restrictions that prevent you from using the Hosted Documentation website, you can install the PeopleSoft Online Help locally. If you install the help locally, you have more control over which documents users can access and you can include links to your organization's custom documentation on help pages.

In addition, if you locally install the PeopleSoft Online Help, you can use any search engine for full-text searching. Your installation documentation includes instructions about how to set up Oracle Secure Enterprise Search for full-text searching.

See *PeopleTools 8.53 Installation* for your database platform, "Installing PeopleSoft Online Help." If you do not use Secure Enterprise Search, see the documentation for your chosen search engine.

Note: Before users can access the search engine on a locally installed help website, you must enable the Search portlet and link. Click the Help link on any page in the PeopleSoft Online Help for instructions.

Downloadable PeopleBook PDF Files

You can access downloadable PDF versions of the help content in the traditional PeopleBook format. The content in the PeopleBook PDFs is the same as the content in the PeopleSoft Online Help, but it has a different structure and it does not include the interactive navigation features that are available in the online help.

Common Help Documentation

Common help documentation contains information that applies to multiple applications. The two main types of common help are:

- Application Fundamentals

- Using PeopleSoft Applications

Most product lines provide a set of application fundamentals help topics that discuss essential information about the setup and design of your system. This information applies to many or all applications in the PeopleSoft product line. Whether you are implementing a single application, some combination of applications within the product line, or the entire product line, you should be familiar with the contents of the appropriate application fundamentals help. They provide the starting points for fundamental implementation tasks.

In addition, the *PeopleTools: PeopleSoft Applications User's Guide* introduces you to the various elements of the PeopleSoft Pure Internet Architecture. It also explains how to use the navigational hierarchy, components, and pages to perform basic functions as you navigate through the system. While your application or implementation may differ, the topics in this user's guide provide general information about using PeopleSoft Applications.

Field and Control Definitions

PeopleSoft documentation includes definitions for most fields and controls that appear on application pages. These definitions describe how to use a field or control, where populated values come from, the effects of selecting certain values, and so on. If a field or control is not defined, then it either requires no additional explanation or is documented in a common elements section earlier in the documentation. For example, the Date field rarely requires additional explanation and may not be defined in the documentation for some pages.

Typographical Conventions

The following table describes the typographical conventions that are used in the online help.

<i>Typographical Convention</i>	<i>Description</i>
Key+Key	Indicates a key combination action. For example, a plus sign (+) between keys means that you must hold down the first key while you press the second key. For Alt+W, hold down the Alt key while you press the W key.
... (ellipses)	Indicate that the preceding item or series can be repeated any number of times in PeopleCode syntax.
{ } (curly braces)	Indicate a choice between two options in PeopleCode syntax. Options are separated by a pipe ().
[] (square brackets)	Indicate optional items in PeopleCode syntax.
& (ampersand)	When placed before a parameter in PeopleCode syntax, an ampersand indicates that the parameter is an already instantiated object. Ampersands also precede all PeopleCode variables.
⇒	This continuation character has been inserted at the end of a line of code that has been wrapped at the page margin. The code should be viewed or entered as a single, continuous line of code without the continuation character.

ISO Country and Currency Codes

PeopleSoft Online Help topics use International Organization for Standardization (ISO) country and currency codes to identify country-specific information and monetary amounts.

ISO country codes may appear as country identifiers, and ISO currency codes may appear as currency identifiers in your PeopleSoft documentation. Reference to an ISO country code in your documentation does not imply that your application includes every ISO country code. The following example is a country-specific heading: "(FRA) Hiring an Employee."

The PeopleSoft Currency Code table (CURRENCY_CD_TBL) contains sample currency code data. The Currency Code table is based on ISO Standard 4217, "Codes for the representation of currencies," and also relies on ISO country codes in the Country table (COUNTRY_TBL). The navigation to the pages where you maintain currency code and country information depends on which PeopleSoft applications you are using. To access the pages for maintaining the Currency Code and Country tables, consult the online help for your applications for more information.

Region and Industry Identifiers

Information that applies only to a specific region or industry is preceded by a standard identifier in parentheses. This identifier typically appears at the beginning of a section heading, but it may also appear at the beginning of a note or other text.

Example of a region-specific heading: "(Latin America) Setting Up Depreciation"

Region Identifiers

Regions are identified by the region name. The following region identifiers may appear in the PeopleSoft Online Help:

- Asia Pacific
- Europe
- Latin America
- North America

Industry Identifiers

Industries are identified by the industry name or by an abbreviation for that industry. The following industry identifiers may appear in the PeopleSoft Online Help:

- USF (U.S. Federal)
- E&G (Education and Government)

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

Using and Managing the PeopleSoft Online Help

Click the Help link in the universal navigation header of any page in the PeopleSoft Online Help to see information on the following topics:

- What's new in the PeopleSoft Online Help.
 - PeopleSoft Online Help accessibility.
 - Accessing, navigating, and searching the PeopleSoft Online Help.
 - Managing a locally installed PeopleSoft Online Help website.
-

PeopleSoft EPM Related Links

[My Oracle Support](#)

[PeopleSoft Information Portal on Oracle.com](#)

[PeopleSoft Training from Oracle University](#)

[PeopleSoft Video Feature Overviews on YouTube](#)

Contact Us

[Send us your suggestions](#) Please include release numbers for the PeopleTools and applications that you are using.

Follow Us



Get the latest PeopleSoft updates on [Facebook](#).



Follow PeopleSoft on [Twitter@PeopleSoft_Info](#).

Chapter 1

Getting Started With PeopleSoft Enterprise Performance Management

PeopleSoft EPM Overview

EPM is a packaged data warehousing platform that serves both as a repository of your information for reporting and analysis, and as the foundation for the PeopleSoft EPM Warehouses and Analytical Applications. EPM provides the tools necessary to gather data from transactional, legacy, and external data sources, stage, store, and enrich that data, and make the information available for analysis.

Related Links

[Overview](#)

PeopleSoft EPM Integrations

PeopleSoft EPM is the central repository for the PeopleSoft EPM Warehouses and Analytical Applications, and fully integrates with:

- PeopleSoft transactional systems
 - IBM WebSphere DataStage
 - EPM Warehouses
 - PeopleSoft EPM Analytical Applications
-

PeopleSoft EPM Implementation

PeopleSoft Setup Manager enables you to generate a list of setup tasks for your organization based on the features that you are implementing. The setup tasks include the components that you must set up, listed in the order in which you must enter data into the component tables, and links to the corresponding documentation.

Other Sources of Information

In the planning phase of your implementation, take advantage of all PeopleSoft sources of information, including the installation guides, table-loading sequences, data models, and business process maps. A complete list of these resources appears in *PeopleSoft and the Online Library*, with information about where to find the most current version of each.

PeopleTools: PeopleSoft Setup Manager

Related Links

[Understanding EPM Implementation](#)

Common Elements Used in the PeopleSoft EPM Documentation Set

This topic lists common elements used in PeopleSoft EPM.

SetID	Provides the ID code for a tableset. A tableset is a group of tables (records) necessary to define your company's structure and processing options.
Effective Date	Establishes the date the row in the table becomes effective. It determines when you can view and change the information. Pages and batch processes that use the information use the current row.
Status	Indicates whether a row in a table is active or inactive. You cannot select inactive rows on pages or use them for running batch processes
Description	Enables you to input free flow text, up to 30 characters, that describes what you are defining.
Run Control ID	Identifies specific run control settings for a process or report.
Report ID	Identifies the report.
Program Name	Provides the EPM program name for which you are running the report or process.
When	Specifies the frequency with which you want to run a process. You can choose <i>Once</i> , <i>Always</i> , or <i>Don't</i> .
Last Run On	Indicates the date the report or process was last run.
As Of Date	Indicates the last date for which the report or process includes data.
Scenario ID	Provides an identifier for a specific scenario.
Model ID	Provides an identifier for a model. A model uniquely identifies the types of data you want to include in a scenario. For example, you might want to review revenue by region—a very high-level scope. Or, if you use Activity-Based Management, you might want to review only those activities that relate to a certain product line for certain types of resources—a very narrow scope.
Fiscal Year	Specifies the fiscal year for your scenario or process run.

Period	Specifies the accounting period for the object being defined or process being run.
Job ID	Specifies an instance of an engine.

Understanding PeopleSoft Enterprise Performance Management

Overview

PeopleSoft EPM is a comprehensive, integrated analytic business solution designed to increase the efficiency of your organization. PeopleSoft EPM helps your organization achieve operational excellence by providing insight into the information you need to drive predictability, accountability, and manage operational risk. EPM enables you to produce detailed activity analyses and resource plans, understand the cause-and-effect relationship between cost and behavior, organize strategic thinking and performance measurement, use continuous, collaborative forecasting to manage the plan and budget in real-time, and clearly communicate strategy and success measures.

EPM is supported by data warehouses, related data models, robust infrastructure and metadata, and the EPM Foundation toolset. EPM provides all the necessary tools to gather and manage data from PeopleSoft, legacy, and external data sources, enrich that data, and store it in an intuitive analytic context for you to analyze in a variety of ways and at a variety of levels. EPM enables you to deliver a single, accurate view of information across your organization.

PeopleSoft EPM Architecture

PeopleSoft EPM Warehouses and Analytical Applications are built on a foundation of specialized data warehouses, target warehouse tables, ETL jobs, metadata, and other prepackaged content that enable complex analysis and reporting of your data.

EPM target warehouse tables provide a way to consolidate and store your source transaction data. EPM target warehouse tables reside in two high-level data warehouse structures:

- the *Operational Warehouse (OW)*
- the *Multidimensional Warehouse (MDW)*

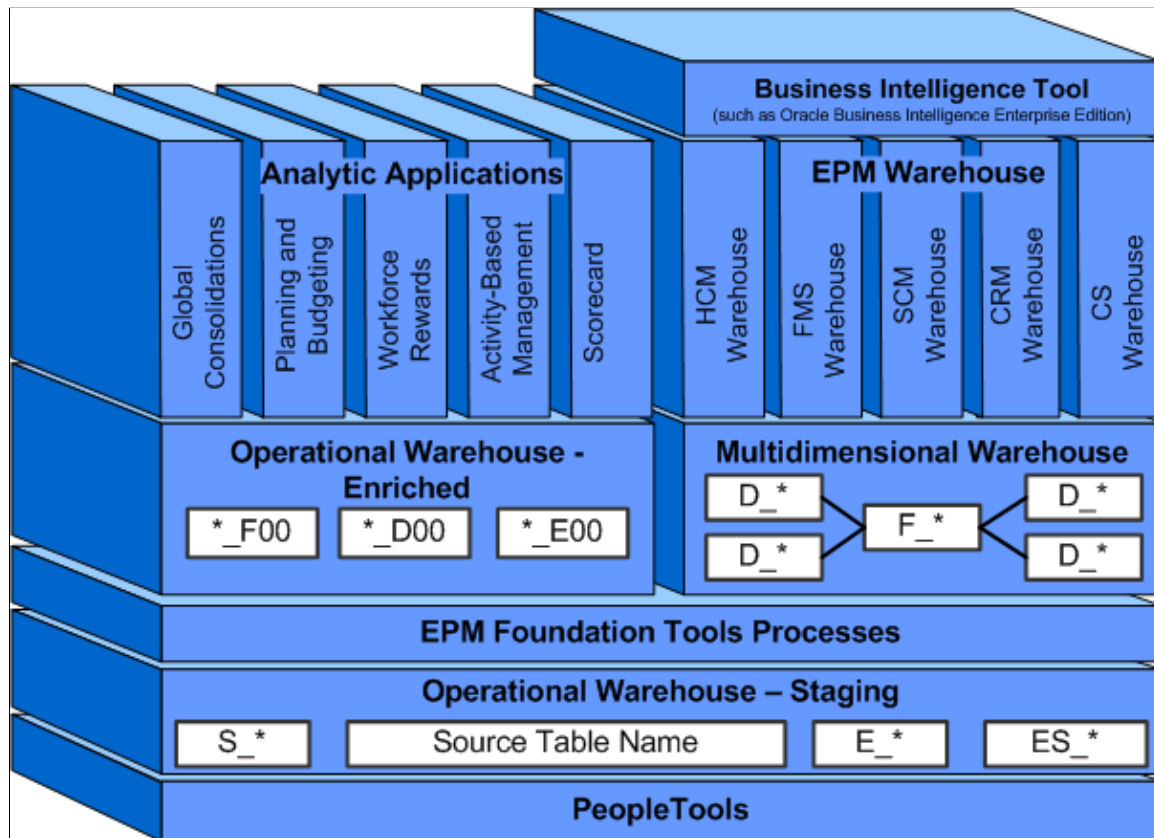
The Operational Warehouse can be further divided into the *Operational Warehouse - Staging (OWS)* and the *Operational Warehouse - Enriched (OWE)*.

Each warehouse structure has its own set of specialized target warehouse tables that are unique to that structure. For example, the Operational Warehouse - Enriched (OWE) structure stores enriched data that is arranged in a normalized format to promote complex analytics. And the Multidimensional

Warehouse (MDW) structure stores data that is arranged in a denormalized format (dimensional schema) for enhanced reporting capabilities.

Image: EPM architecture

The following graphic illustrates the various components comprising the EPM architecture and how each component relates to the others, including shared components which act as the foundation for both the EPM Warehouses and Analytical Applications.



The dual data warehouse architecture helps to:

- Isolate and channel specific source data to the appropriate data warehouse structure for individual enrichment and modeling.

PeopleSoft provides *extract, transform, and load (ETL) jobs* to extract information contained in your source systems, load it into the Operational Warehouse - Staging (OWS) structure, and migrate that data to the Operational Warehouse - Enriched (OWE) and the Multidimensional Warehouse (MDW) structures. And because the warehouse structures are logically separated, the ETL jobs can isolate and channel specific source data to the OWE or the MDW.

- Facilitate specialized, or tailored, data enrichment for your source data.

PeopleSoft provides *EPM Foundation tools and processes* (a set of specialized tools, processes, and metadata) that prepare and enrich your source data for the EPM Warehouses and Analytical Applications.

The delivered target warehouse tables, ETL jobs, Foundation tools, and other packaged content work together to provide the underlying infrastructure on which the EPM Warehouses and Analytical

Applications are built. Detailed information regarding the OWS, OWE, MDW, and EPM Foundation tools can be found in this topic.

Note: EPM *data warehouse structures* refer to the OWS, OWE, and MDW, whereas *EPM Warehouses* refer to the PeopleSoft packaged warehouse solutions available for licensing, such as the Campus Solutions Warehouse and the Human Capital Management Warehouse.

Definition of a Data Warehouse

A textbook definition of a data warehouse is: a copy of transaction data specifically structured for query and analysis.

Transactional database applications have been widely used by the corporate world for over 30 years. Although data has been entered into dedicated transaction applications for decades, it has become apparent that extracting data from these systems for analytic purposes can be cumbersome and difficult.

Data warehousing is the process of taking data from legacy and transaction database systems and transforming it into organized information in a user-friendly format to encourage data analysis and support fact-based business decision-making.

A data warehouse is a central, integrated database that contains data from one or more operational sources and archive systems in an organization. It contains a copy of transaction data that is specifically structured for query analysis.

The mission of the data warehouse is to publish an organization's data assets to most effectively support decision-making. Because the data warehouse is a decision-support system, the main criterion of success is whether the data warehouse contributes to the most important decision-making processes in the organization.

Related Links

[Operational Warehouse - Staging \(OWS\)](#)

[Multidimensional Warehouse \(MDW\)](#)

[EPM Foundation Toolset](#)

Extract, Transform, and Load (ETL) in EPM

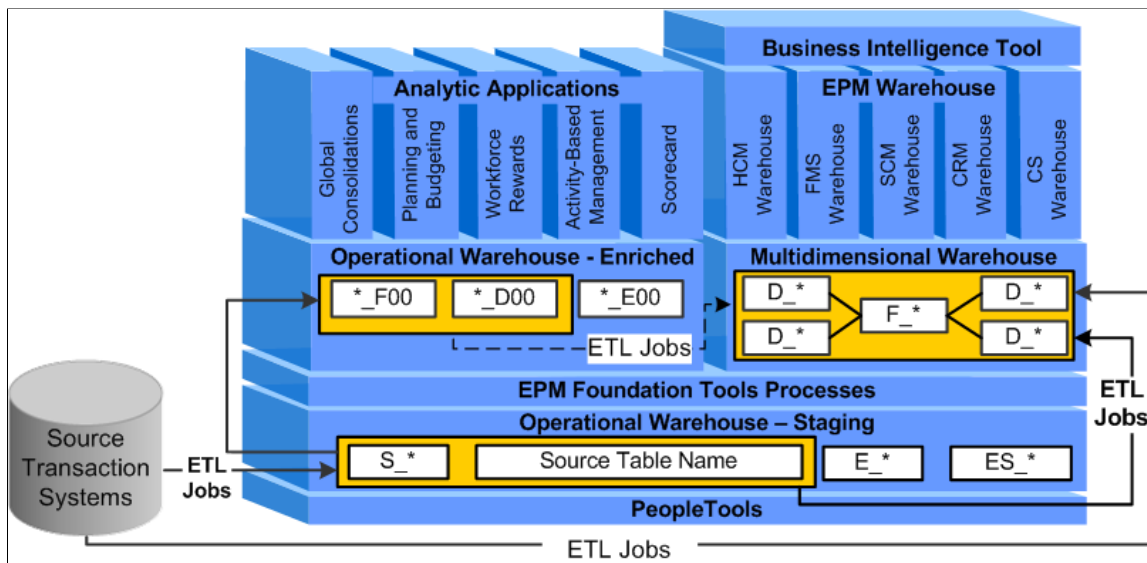
PeopleSoft has an original equipment manufacturer (OEM) agreement with IBM WebSphere to supply *extract, transform, and load (ETL)* technology that supports source data acquisition and data movement within EPM. The ETL tool, IBM WebSphere *DataStage*, is delivered with EPM.

PeopleSoft uses IBM WebSphere *DataStage* to deliver prepackaged ETL jobs that extract information contained in PeopleSoft source systems, load it into the Operational Warehouse - Staging (OWS), and migrate that data to the Operational Warehouse - Enriched (OWE) and the Multidimensional Warehouse (MDW). But ETL jobs do more than migrate data; they also identify data for extraction and ensure the

consistency and validity of your data. Because the ETL jobs are so versatile, separate tools and engines that extract, stage, and move data are not necessary.

Image: ETL process in EPM

The following graphic illustrates the various components comprising the EPM architecture and how data flows from source systems to the Operational and Multidimensional warehouses via the ETL process.



As depicted in the diagram, source transaction data is extracted into OWS tables and migrated across warehouse layers using the aforementioned ETL jobs. Also, source data is sometimes extracted directly into the MDW.

You can use IBM WebSphere DataStage to build custom jobs for mapping your data into EPM . However, PeopleSoft does not support custom jobs.

Detailed information regarding the ETL process can be found in the [Understanding ETL in EPM](#) topic.

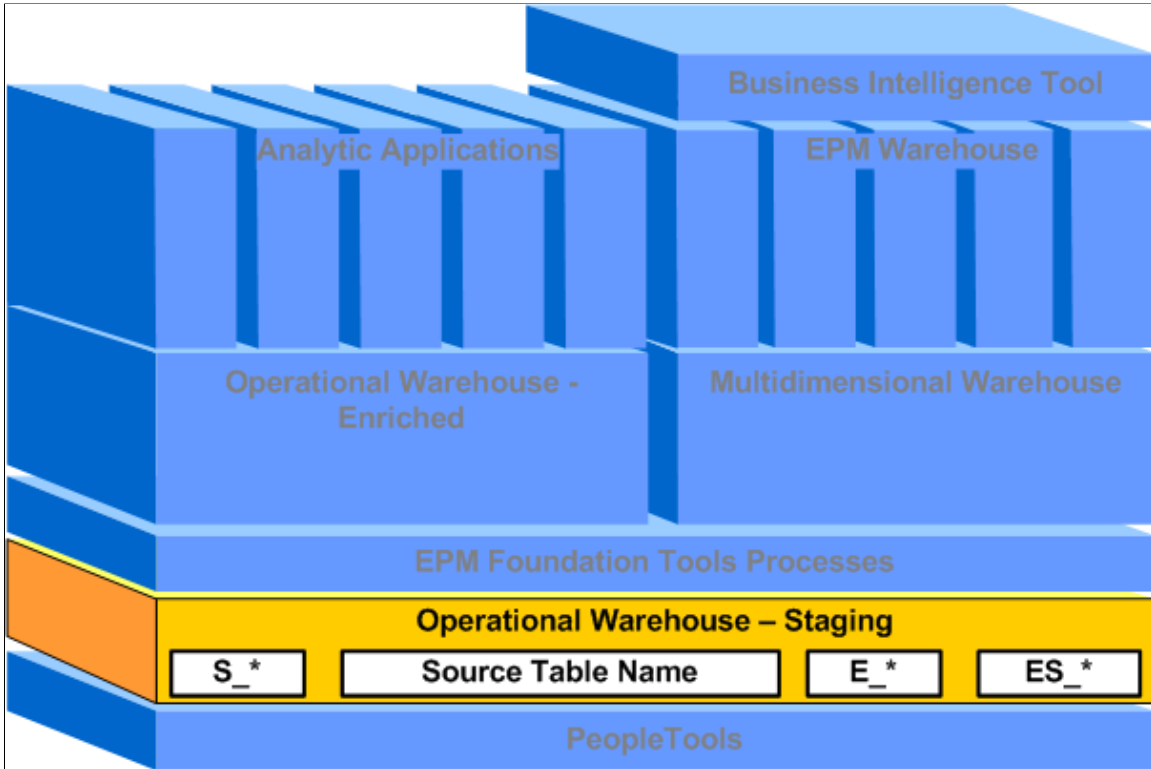
Operational Warehouse - Staging (OWS)

The OWS structure is one of two subcomponents that comprise the Operational Warehouse. The OWS acts as an entry-point for your source transaction data into EPM and can house data from one or more

of your PeopleSoft, legacy, or external source systems. The main function of the OWS is to provide a platform to offload, consolidate, and stage your source transaction data in preparation for enrichment.

Image: Operational Warehouse - Staging (OWS)

The following graphic illustrates the OWS component of the EPM architecture and the target tables that are present in the OWS.



Source data is extracted into the OWS using prepackaged ETL jobs and loaded into target staging tables. No transformations are performed on your source data during this process and the system maintains the same source-level of granularity for your data. Source tables are extracted into the OWS, including all logically related tables, to ensure your source data is semantically complete in EPM. For example, a table extracted into the OWS may have an associated related language table in the source system. The related language data from the associated table is also extracted into the OWS to maintain completeness and data integrity. Data stored in the OWS is used as input for the Operational Warehouse - Enriched (OWE) and the Multidimensional Warehouse (MDW) structures.

Note: The OWS does not contain reporting tables nor prepackaged reports built on the core OWS target tables.

OWS Core Target Tables

OWS core target tables contain data extracted from PeopleSoft source systems. OWS target tables are permanent tables (as opposed to temporary tables), and can store historical data. However, it is not the recommended location for historical data as the tables can be purged from time to time depending on your operational needs.

Generally, the structure of the OWS target tables match the structure of the source transaction tables with the addition of a source system identification column (SRC_SYS_ID) , which enables you to track the origin of your data. However, minor structural differences can occur when:

- Fields are removed from the source system record.
- New fields are added to the source system record but not used by the EPM warehouse.

For example, the following fields were added to the FIN 91 source system: P6_INCL_INTGRTN, P6_TEMPLATE_ID, and P6_WBS_LVL. But these fields are not used by the FMS Warehouse, so EPM does not propagate these fields to the corresponding OWS table.

- The OWS table is designed as a common table and stages data from more than one PeopleSoft source system.

For example, the PS_PROJECT OWS table is a common table that captures project related data from PeopleSoft Campus Solutions, Financials, and HCM source systems. The fields BURDEN_PLAN, SUMMARY_SW, and WBS_TREE_NAME are present in the Campus Solutions and HCM source system tables, but not in the Financials table.

Note: Certain OWS target tables have specific non-key columns that can be "activated" as key columns if your business requirements necessitate it.

Sample OWS Target Table

Image: OWS target table - ABS_CLASS_TBL

The following is a sample OWS target table page shown in Application Designer.

	Num	Field Name	Type	Len	Format	Short Name	Long Name
	1	SETID	Char	5	Upper	SetID	SetID
	2	ABSENCE_CLASS	Char	4	Upper	Absence Class	Absence Class
	3	SRC_SYS_ID	Char	5	Upper	Source ID	Source System Identifica
	4	DESCR	Char	30	Mixed	Descr	Description
	5	DESCRSHORT	Char	10	Mixed	Short Desc	Short Description
	6	LOAD_OWS_SBR	SRec				

OWS Target Table Naming Convention

OWS target tables use the following naming conventions:

- S_[source table name]
- [source table name]

OWS Error Tables

The OWS contains error tables used in the data validation process. The data validation process uses ETL jobs to verify the integrity and completeness of the data entering OWE and MDW target tables. The validation process can perform dimension key validation (for example, verifying that customer ID fact value has a corresponding customer ID dimension value) and general key validation (for example, verifying the pre-fact customer ID in the OWS table has a corresponding customer ID in the OWE or MDW table), as well as ensure source business unit and SetID are properly mapped to EPM values and source codes are properly mapped to EPM code values.

Data failing the validation process are sent to OWS error tables. It is important to note that the OWS error tables have a different structure than the error tables in the OWE and perform a very different function. The OWS error table mirrors the key structure and other columns of its corresponding data table and has additional fields to facilitate troubleshooting. The following OWS error table columns represent some of the columns provided for troubleshooting:

- **LOAD_OWS_SBR:** The values for these columns are copied from the failing data row. The reason for copying values from the failed data row is that it provides vital load information such as batch ID and load timestamp for the data row.
- **Target Table:** This column lists the target table for the job.
- **Failed data source table and column name:** The source table and column from which the failing data originated. Knowing the name of the failed source data table is especially useful when the job loading the failed data contains a multi-source-table join.
- **Failed Data Value:** The actual value that failed validation.
- **Lookup table and column name:** The table and column against which the failed lookup was performed.

Detailed information regarding the data validation process can be found in the [Understanding Data Validation and Error Handling in the ETL Process](#) topic.

Sample OWS Error Table

Image: OWS error table - E_ABS_CLASS_TBL

The following is a sample OWS error table page shown in Application Designer.

	Num	Field Name	Type	Len	Format	Short Name	Long Name
	1	ERROR_SID	Nbr	10	Raw B	Error Row SID	Error Row SID
	2	SETID	Char	5	Upper	SetID	SetID
	3	ABSENCE_CLASS	Char	4	Upper	Absence Class	Absence Class
	4	SRC_SYS_ID	Char	5	Upper	Source ID	Source System Identification
	5	ERR_TRACE_SBR	SRec				
	6	LOAD_OWS_SBR	SRec				

OWS Error Table Naming Convention

OWS error tables use the following naming conventions:

- PS_E_[OWS table name]
- PS_ES_[OWS table name]

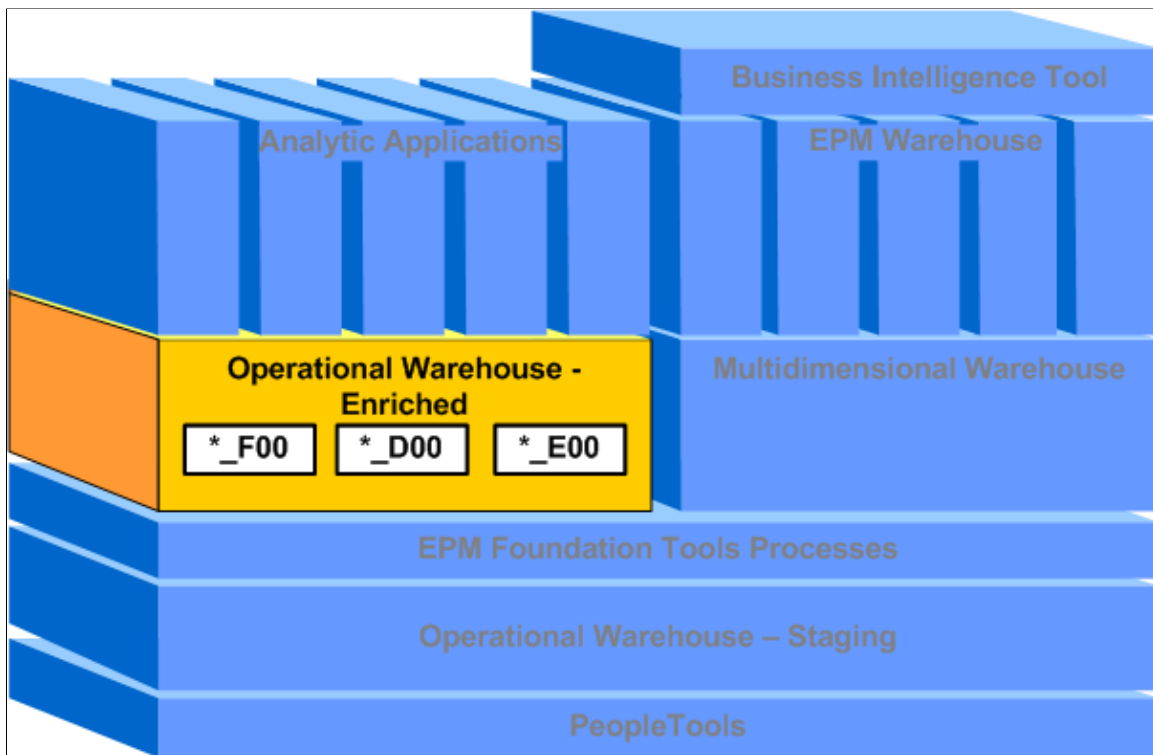
Operational Warehouse - Enriched (OWE)

The OWE structure is the second of two subcomponents that comprise the Operational Warehouse. The OWE stores enriched data that is arranged in a normalized format and mapped to warehouse business units (WBU). Enrichment can entail many transformations to your data, including (but not limited to) conversion to a common currency, common calendar, or a common ledger, or aggregating data to a

common warehouse business unit. The PeopleSoft Analytical Applications use the enriched data in the OWE to perform analysis and reporting.

Image: Operational Warehouse - Enriched (OWE)

The following graphic illustrates the OWE component of the EPM architecture and the target tables that are present in the OWE.



Data is extracted into the OWE using prepackaged ETL jobs and loaded into target dimension (D00) and fact (F00) tables. The structure of these tables are quite different from the OWS tables because they are arranged in a normalized format and organized data around warehouse business units. In addition, OWE tables are augmented with subrecords which help facilitate the ETL process and tracking data lineage. OWE tables store data permanently and can maintain history (as opposed to temporary tables which remove data at the end of an ETL job).

Tools and Processes Associated with the OWE

EPM is delivered with several tools and processes that enable you to enrich and manage the data stored in the OWE. The following are some of the tools and processes used only with the OWE:

- Performance ledger template setup.
- Detail ledger setup.
- Model and scenario setup.
- Roll-up processing.
- Profit manager.
- EPM object auditing.

- Mass validate processing.
- Mass compile processing.
- Tree utility setup.
- Data manager processing.
- Allocation manager processing.

OWE Dimension (D00) Tables

An OWE dimension table provides additional attributes about a fact for greater flexibility in reporting. Dimensions are derived from operational applications and are cleansed and transformed during data migration. Examples of dimension tables include: product, customer, channel, department, personal data, and accounts. Some of the fields associated with an OWE dimension table are:

- SetID: Key column.
- DIMENSION_ID: Key column.
- EFFDT: This is the same date as the source. If an EFFDT or an alternative date, such as a date time stamp, does not exist, the system creates one and sets it to the date the dimension data is loaded.
- EFF_STATUS.
- KEY Fields from the source table.
- SET CONTROL FIELD. This is BUSINESS_UNIT.
- RELATED LANGUAGE RECORD. This is the same with an extension of _LNG.

Sample OWE Dimension Table

Image: OWE dimension - CUSTOMER_D00

The following is a sample OWE dimension table page shown in Application Designer.

	Num	Field Name	Type	Len	Format	Short Name	Long Name
	1	SETID	Char	5	Upper	SetID	SetID
	2	CUST_ID	Char	15	Upper	Customer	Customer ID
	3	EFFDT	Date	10		Eff Date	Effective Date
	4	EFF_STATUS	Char	1	Upper	Status	Status as of Effective D
	5	NAMESHORT	Char	10	Mixed	Short Name	Short Name
	6	CUSTOMER_TYPE	Char	1	Upper	Type	Customer Type
	7	CUSTOMER_GROUP	Char	10	Upper	Cust Group	Customer Group
	8	SINCE_DT	Date	10		Since	Customer Since
	9	ADD_DT	Date	10		Date Added	Customer Added On
	10	NAME1	Char	50	Mixed	Name	Name 1
	11	NAME2	Char	40	Mixed	Name 2	Name 2
	12	NAME3	Char	40	Mixed	Name 3	Name 3
	13	SALES_PERSON	Char	8	Upper	Sales	Sales Person
	14	CORPORATE_SETID	Char	5	Upper	SetID	SetID
	15	CORPORATE_CUST_ID	Char	15	Upper	Corporate	Corporate Customer
	16	VENDOR_SETID	Char	5	Upper	Vndr SetID	Vendor SetID
	17	VENDOR_ID	Char	10	Upper	Vendor	Vendor ID
	18	FORMER_NAME_1	Char	40	Mixed	Former Name	Former Name
	19	CURRENCY_CD	Char	3	Upper	Currency	Currency Code
	20	SHIP_TO_FLG	Char	1	Upper	Ship To	Ship To Customer
	21	BILL_TO_FLG	Char	1	Upper	Bill To	Bill To Customer
	22	SOLD_TO_FLG	Char	1	Upper	Sold To	Sold To Customer
	23	CUST_LEVEL	Char	1	Upper	Level	Customer Level
	24	ADDRESS_SEQ_NUM	Nbr	3		Addr	Address Sequence Num
	25	ADDRESS_SEQ_SOLD	Nbr	3		Address	Primary Address - Sold T
	26	ADDRESS_SEQ_SHIP	Nbr	3		Address	Primary Address - Ship T
	27	TAXPAYER_ID	Char	14	Upper	Tax ID	Taxpayer ID
	28	COMPANY_SIZE	Nbr	10		Company Size	Company Size
	29	INDUSTRY_ID	Char	30	Upper	CRM Industry ID	CRM Industry ID
	30	BO_ID	Sign	31		BO ID	Business Object ID
	31	BO_ID_PARENT	Nbr	31		Parent	Parent Company
	32	SOURCE_IND	Char	4	Upper	Source?	Source Indicator
	33	PROCESS_INSTANCE	Nbr	10		Instance	Process Instance
	34	LOAD_OWE_SBR	SRec				

OWE Dimension Table Naming Convention

OWE dimension tables use the following naming convention, *[table name]_D00*

OWE Fact (F00) Tables

An OWE fact table contains measures for analyzing performance. Some of the fields associated with an OWE fact table are:

- **BUSINESS_UNIT:** This field enables the fact data to be shared across different dimensions, as they are based on SetIDs.
- **FACT KEY.**
- **ASOF_DT:** This is for non-cumulative facts (for example, account balance).

- PF_TRANS_DT: This is for cumulative facts (for example, billing transactions).
- All KEY Fields: These are required for uniqueness.
- DEFAULT VALUES include:
 - BUSINESS_UNIT: This will have a default table set to OPR_DEF_TBL_FS and a default field set to BUSINESS_UNIT.
 - EFFDT: This will have a default set to %DATE.
 - EFF_STATUS: This will have a default set to A.
 - TRANSLATE VALUES: These values, if any exist, must be set to the XLATTABLE.
- PF_TRANS_DT: This is set to the source record's transaction date. In addition, the source transaction date field is included in the data warehouse fact table.

Sample OWE Fact Table

Image: OWE fact - ACCT_REC_F00

The following is a sample OWE fact table page shown in Application Designer.

	Num	Field Name	Type	Len	Format	Short Name	Long Name
	1	BUSINESS_UNIT	Char	5	Upper	Unit	Business Unit
	2	CUST_ID	Char	15	Upper	Customer	Customer ID
	3	EFFDT	Date	10		Eff Date	Effective Date
	4	BAL_AMT	Sign	23.3		Balance	Customer Balance
	5	HI_BAL_AMT	Sign	23.3		Hi Balance	Hi Balance
	6	HI_BAL_DT	Date	10		Hi Balance	Hi Balance Date
	7	ITEM	Char	30	Upper	Item ID	Item ID
	8	ITEM_AMT	Sign	23.3		Amount	Amount
	9	ITEM_DT	Date	10		Date	Item Date
	10	DEPOSIT_BU	Char	5	Upper	Unit	Deposit Unit
	11	DEPOSIT_ID	Char	15	Upper	Deposit ID	Deposit ID
	12	PAYMENT_SEQ_NUM	Nbr	6		Seq	Payment Sequence
	13	PAYMENT_DT	Date	10		Date	Payment Date
	14	POST_DT	Date	10		Posted	Posted Date
	15	POSTED_PI	Nbr	10		Proc Inst	Last Post Process Instan
	16	AGED_DT	Date	10		Aged	Aged Date
	17	AGED_PI	Nbr	10		Proc Inst	Last Aging Process Insta
	18	PAYMENT_ID	Char	15	Upper	Payment ID	Payment ID
	19	PAYMENT_AMT	Sign	23.3		Amount	Payment Amount
	20	ENTRY_CURRENCY	Char	3	Upper	Ent Crncy	Entry Currency Code
	21	PAYMENT_CURRENCY	Char	3	Upper	Currency	Payment Currency
	22	CURRENCY_CD	Char	3	Upper	Currency	Currency Code
	23	AGED_AMT_CR_CHK	Sign	23.3		Amt Checkd	Amount Credit Checked
	24	ITEM_LINE	Nbr	6		Line	Item Line
	25	PROCESS_INSTANCE	Nbr	10		Instance	Process Instance

OWE Fact Table Naming Convention

OWE fact tables use the following naming convention, *[table name]*_F00

OWE Temporary Tables

OWE temporary tables support parallel processing. A temporary table layout and key structure differs from its respective fact or dimension table in that the organizational unit (SetID or business unit) and the effective date are not keys.

EPM is delivered with three sets of temporary tables. You can define additional sets of tables when needed. The project EPM_TEMP_TABLES contains one instance of every temporary table, enabling you to create new temporary table suites, if necessary. A temporary table layout and key structure differs from its respective fact or dimension data warehouse table in that the organizational unit (SetID or business unit) and the effective date are not keys.

Note: If you must create more temporary tables than the ones delivered with PeopleSoft EPM, see the delivered project, EPM_TEMP_TABLES. It contains one instance of every temporary table, enabling you to create new temporary table suites, if necessary.

See [Creating Additional Instances of Temporary Tables for Record Suites](#).

OWE Temporary Table Naming Convention

OWE temporary tables use the following naming convention, `[table name]_T`

Specialized Reporting Tables

The OWE features tables that have been designed specifically to enhance reporting capabilities. Those tables are the performance ledger table (PF_LEDGER_F00), performance journal table (PF_JRNL_F00), and the performance statistics table (PF_STAT_F00). Creating specialized tables in this manner enables you to move away from storing all of your accounting data in your general ledger to make your general ledger perform as it should—as a method for compliance reporting *only*.

The performance journal and performance ledger tables are described in more detail later in this documentation.

Performance Ledger Table

The performance ledger table (PF_LEDGER_F00) is a central fact table within EPM. The performance ledger table is an accumulation of monetary amount facts over a period of time. The primary function of the performance ledger table is to support PeopleSoft EPM reporting. The PF_LEDGER_F00 is the source for one of the data marts.

Note: The performance ledger table should not be confused with a general ledger from an online transaction processing (OLTP) system. The performance ledger contains all information mapped from a general ledger and enriched through one (or more) of the PeopleSoft EPM engines.

Information that has been processed through an PeopleSoft EPM engine, for instance the ABM engine or Data Manager, is stored in a temporary performance journal staging table (PF_JRNL_T).

The PF Edit engine enables you to verify the data in the temporary journal table and moves valid data to the final table, the PF_JRNL_F00. Errors are placed in the PF_JRNL_E00, the error table for the journal table. The PF Post takes the detailed information from the performance journal table, aggregates it to the desired level of summarization and posts it to the PF_LEDGER_F00 for reporting.

PeopleSoft EPM reporting tools support multidimensional analysis based primarily on profitability dimensions such as customer, product, and channel. You can use one, two, or more of these dimensions within your models, or configure the application to add more dimensions, or change the existing ones. No matter which dimensions you select, however, you need to consider how to populate the performance ledger table with meaningful multidimensional data.

Performance Journal Table

The performance journal table (PF_JRNL_F00):

- Contains data that is not yet summarized.
- Is a fact table, or multiple fact tables, within EPM.
- Is a collection of batches of amount facts staged for validation and posting to the performance ledger table.
- Supports drill down from reports produced against the performance ledger table.

The PF Edit engines moves data to the performance journal fact table. The PF Post process accumulates valid transactions from the performance journal table, and inserts summarized rows into the performance ledger. There is a "many to one" relationship between the performance journal and the performance ledger tables.

Performance Statistics Table

The performance statistics table (PF_STAT_F00) is similar to the performance ledger table (PF_LEDGER_F00) in its layout.

OWE Error Tables (for Profit Manager only)

The OWE contains error tables used to identify flawed data in certain OWE target tables. There are a small number of delivered OWE error tables and they are used only for Profit Manager. Profit Manager uses specific business rules to validate and format data in its related OWE target tables. If the business rules are not met, then the flawed records are written to an OWE error table and a message describing the error is written to a detail error message table (TSE table). If your load results in errors, you can use PF Modification to correct the errors. You can correct the errors using the PeopleSoft Application Designer and then migrate the corrected tables to the target. The following OWE error tables are delivered:

- BP_LED_BUDG_E00
- BP_LED_E00
- BP_LED_KK_E00
- BP_LED_PROJ_E00
- GC_JRNL_MGT_E00
- LEDGER_E00
- PF_JRNL_E00

See [Understanding Profit Manager Tools](#).

Sample OWE Error Table

Image: OWE error table - BP_LED_E00

The following is a sample OWE error table page shown in Application Designer.

Num	Field Name	Type	Len	Format	Short Name	Long Name
1	BUSINESS_UNIT	Char	5	Upper	Unit	Business Unit
2	LEDGER	Char	10	Upper	Ledger	Ledger
3	FISCAL_YEAR	Nbr	4		Year	Fiscal Year
4	ACCOUNTING_PERIOD	Nbr	3		Period	Accounting Period
5	ACCOUNT	Char	10	Upper	Acct	Account
6	ALTACCT	Char	10	Upper	Alt Acct	Alternate Account
7	STATISTICS_CODE	Char	3	Upper	Stat	Statistics Code
8	BP_CF9B_AK_SBR	SRec				
9	PROJECT_ID	Char	15	Upper	Project	Project
10	CURRENCY_CD	Char	3	Upper	Currency	Currency Code
11	BOOK_CODE	Char	4	Upper	Book Code	Book Code
12	GL_ADJUST_TYPE	Char	4	Upper	Adjustment	Adjustment Type
13	PF_EDIT_SEQ_NUM	Nbr	6		Last Edit Seq #	Last Edit Seq Number
14	POSTED_TOTAL_AMT	Sign	23.3		Total Amt	Posted Total Amount
15	POSTED_BASE_AMT	Sign	23.3		Amount	Posted Base Currency A
16	POSTED_TRAN_AMT	Sign	23.3		Transaction Amt	Posted Transaction Amo
17	BASE_CURRENCY	Char	3	Upper	Base Curr	Base Currency
18	DTTM_STAMP_SEC	DtTm	26	Scnds	DateTime	Last Update DateTime
19	ERROR_FLAG	Char	1	Upper	Err Flag	Error Flag
20	PROCESS_INSTANCE	Nbr	10		Instance	Process Instance

OWE Error Table Naming Convention

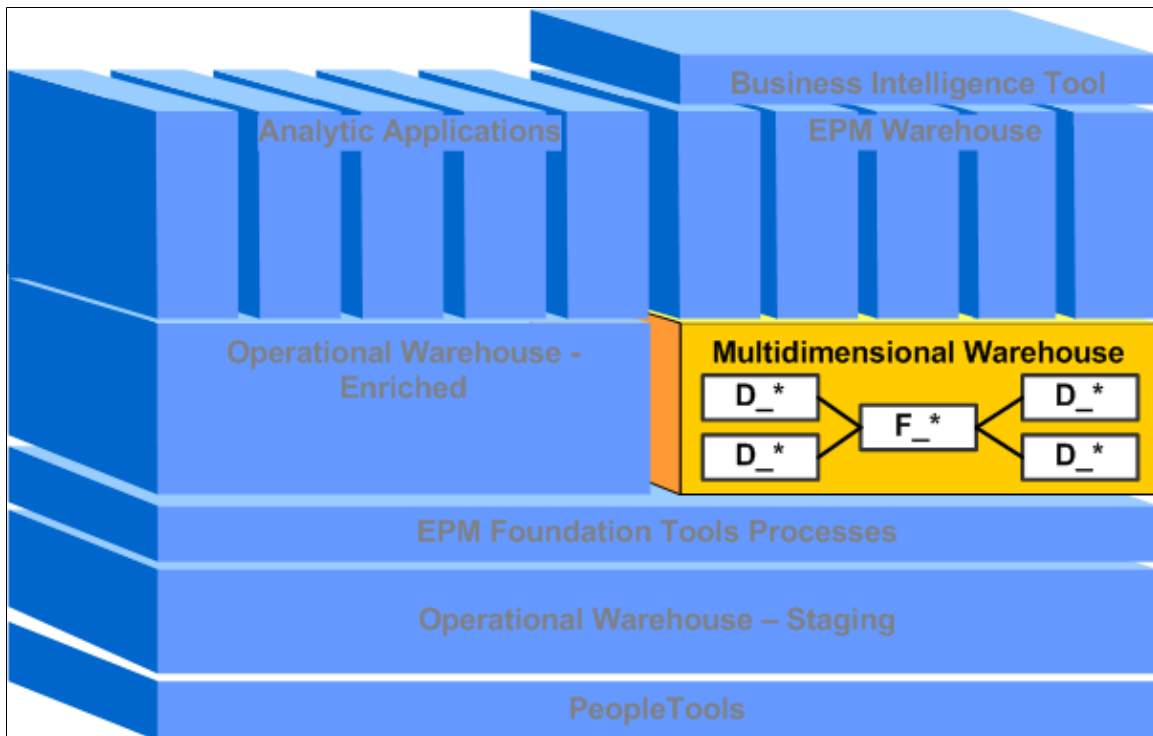
OWE error tables use the following naming convention, *[table name]_E00*

Multidimensional Warehouse (MDW)

The Multidimensional Warehouse is the third data structure in EPM.

Image: Multidimensional Warehouse (MDW)

The following graphic illustrates the MDW component of the EPM architecture and the target tables that are present in the MDW.



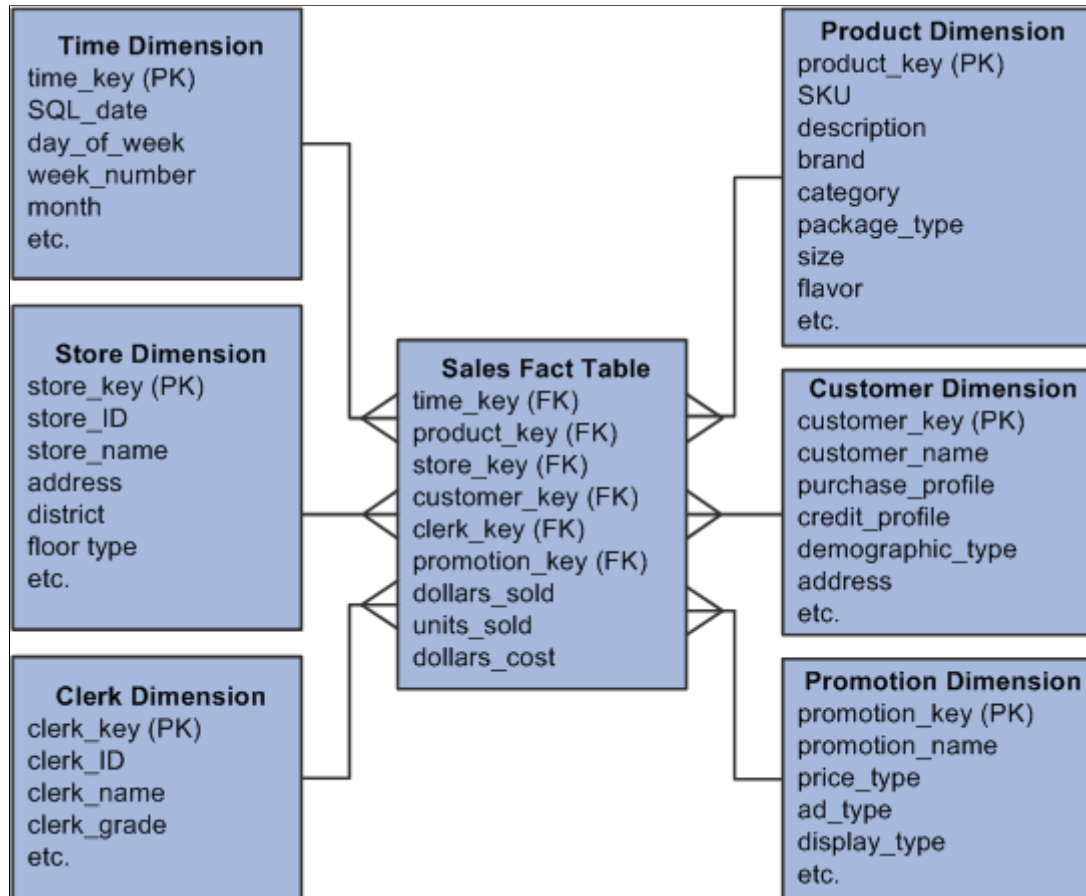
The MDW stores dimensionalized data that is grouped into one or more business processes, better known as a *dimensional schema*, used for business intelligence and ad hoc reporting. The data is stored in a *star schema* (a fact table associated with a series of dimension tables) and generally contains data loaded from the OWS.

The star schema arrangement depends entirely on *primary key* and *foreign key* relationships. A primary key is a column (or columns) in a dimension table whose values uniquely identify each row in the table. Primary keys enforce entity integrity by uniquely identifying entity instances. A foreign key is a column or columns in a fact table whose values match the primary key values of a given dimension table. This way references can be made between a fact and dimension table. Foreign keys enforce referential integrity by completing an association between two entities.

Note: MDW dimensions use a *surrogate key*, a unique key generated from production keys by the ETL process. The surrogate key is not derived from any data in the EPM database and acts as the primary key in a MDW dimension. See the next topic for more information on surrogate keys in the MDW.

Image: Dimensional Model Example

The following graphic provides an example of a star schema and its primary and foreign key relationships:



Although data loaded into the MDW is primarily derived from the OWS, there are exceptions to this rule. Profitability and Global Consolidations data for the Financial Management Solutions (FMS) Warehouse is loaded into the MDW from the OWE.

External survey data for the HCM Warehouse is loaded into the MDW from the OWE.

Online Marketing data is loaded into the MDW directly from the source system, and bypasses the Operational Warehouse entirely.

Surrogate Keys

Surrogate keys provide a means of defining unique keys whose values, with the exception of the Time and Calendar dimensions, are anonymous—that is, the value of a surrogate key has no significance to the application using it and is strictly an artificial value. The system uses surrogate keys specifically as a means of joining structures. To speed up query access, the MDW resolves PeopleSoft-specific programming constructs, such as SetIDs and effective dates and replaces them with surrogate IDs as key columns. Surrogate keys have no relationship to the business or production key. Surrogate keys are

present in dimension tables as the primary key and in fact tables as foreign keys to dimensions. However, the dimension record retains the business key as an alternate-key attribute. Surrogate keys are four-byte integers and their size does not change even when production key changes in size.

Although surrogate keys usually do not have any "intelligence," that is, their value has no meaning, in certain situations, such as the Gregorian Calendar and Time dimensions, intelligent surrogate keys are used. These intelligent keys enable the ETL process to run more quickly by providing the option of avoiding a lookup on corresponding dimensions.

Surrogate key fields usually have the suffix *_SID* (Surrogate ID).

Surrogate Keys and the ETL Process

Surrogate keys are generated from production keys using the DataStage routine *KeyMgtNextValueConcurrent()*, which receives an input parameter and a name identifying the sequence. The surrogate key can be unique per single dimension target (D) or unique across the whole (W) multidimensional warehouse. This process is enabled by the environment parameter named *SID_UNIQUENESS*. The value for this parameter is provided at run time. If the value is *D*, then this routine is called with a dimension job name for which a surrogate key must be assigned and it returns the next available number. If not, the routine is called with *EPM* as the sequence identifier.

You do not have to take any action to create surrogate keys; they are generated during the ETL process within the aforementioned DataStage routine. The DataStage routine retrieves the next surrogate key value and assigns it to the surrogate key that it is currently creating. When the ETL process copies a dimension row from the source system into the MDW, the ETL process performs a lookup on the dimension table. If the dimension row (with same business keys) does not exist in the dimension table, the process inserts a row with a new surrogate key value. If the dimension row already exists in the dimension table, the process updates the existing row with the incoming row value. When the ETL process copies a fact row from the source system into the MDW, for each dimension key in the fact row, the system performs a lookup on the dimension table and retrieves the corresponding surrogate key value. This surrogate key is the foreign key value in the fact row in the MDW. If the system does not locate a dimension value in the fact row in the dimension table, that is a data exception and an error results.

Surrogate Key Benefits

Surrogate keys provide benefits such as:

- The ability to easily and structurally conform a dimension when being sourced from multiple systems.
- Disassociation from operational system changes.

Because surrogate key generation is controlled by the warehouse, it is not influenced by operational system changes.

- The ability to handle unspecified or missing key values.
- A graceful mechanism to handle changes in history.

Multiple versions of a dimension can be maintained with different surrogate (primary) keys, yet with the same business (identifying) key.

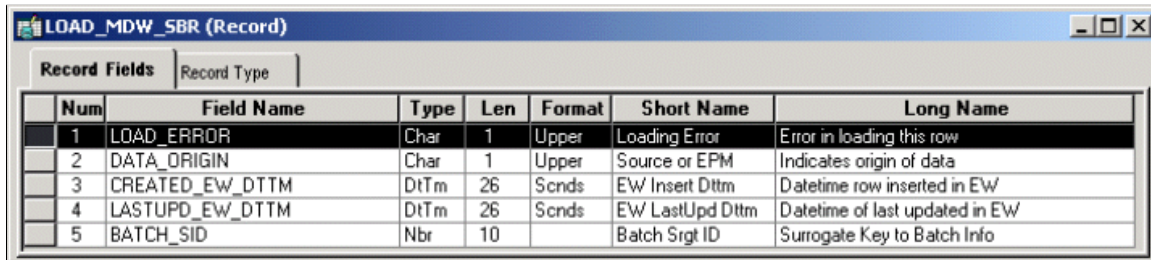
- Performance enhancement of queries, because a surrogate key is a single column numeric key, thus the joins using surrogate keys are faster than ones using multi-column business keys.

Audit Fields

Audit fields track extract, transform, and load (ETL) loading information, such as when the row was loaded or last modified or the batch in which the row was loaded. This information is included in a subrecord. The subrecord added to MDW tables is called `LOAD_MDW_SBR`. Subrecords are always added at the end of a record; no fields exist after this subrecord in any table.

Image: `LOAD_MDW_SBR` record example

The following example shows a typical `LOAD_MDW_SBR` subrecord.



Num	Field Name	Type	Len	Format	Short Name	Long Name
1	LOAD_ERROR	Char	1	Upper	Loading Error	Error in loading this row
2	DATA_ORIGIN	Char	1	Upper	Source or EPM	Indicates origin of data
3	CREATED_EW_DTTM	DtTm	26	Scnds	EW Insert Dttm	Datetime row inserted in EW
4	LASTUPD_EW_DTTM	DtTm	26	Scnds	EW LastUpd Dttm	Datetime of last updated in EW
5	BATCH_SID	Nbr	10		Batch Srgt ID	Surrogate Key to Batch Info

Data Aggregation

Tables in the MDW contain source data at the same granularity as the source system. Required data aggregation is carried out at run time by the business intelligence tool. This allows for better control of aggregation strategies by the business intelligence tool, because aggregation requirements vary from customer to customer.

MDW Dimension Tables

Dimensions are sets of related attributes that you use to group or constrain detailed information that you measure in your data mart. Dimensions are usually text (in character data type), relatively static, and often hierarchical.

Dimension tables contain surrogate keys as the primary key and are a single column key containing only the surrogate key column. Surrogate keys usually have `_SID` (surrogate ID) appended to the field name. Dimension tables retain source system business key fields as non-key attribute columns in the dimension table. However, these are not used for joins with fact tables. For example, in the Customer dimension, the original business key field `CUST_ID` is retained, if it exists in the source table, but is no longer included in the key. The `SetID` is also retained, if it exists in the source table, as a nonkey attribute; the value contained in the `SetID` is the same as in the source system.

If a dimension is `SetID`-based, the MDW table contains the source `SetID` and the performance (PF) `SetID`, which is named `SETID`.

If a dimension contains a description text, a related language table is often defined for this dimension. The ETL process populates this table if a customer requires multilanguage processing. The key for this table is the surrogate key ID, plus the language code field, `LANGUAGE_CD`, which contains the code for the additional language.

Note: You can find more information about multilanguage processing for the multidimensional warehouse in your EPM Warehouse specific documentation (for example, the *PeopleSoft EPM: Campus Solutions Warehouse*).

Shared Dimensions

Dimensions such as Account, Customer, Department, or Person are examples of shared dimensions. Shared dimensions are either exactly the same—including key structure—or an exact subset of another dimension; that is, shared dimensions are structurally identical every place in which they are used. Shared dimensions are used across all EPM warehouse products, such as the Campus Solutions Warehouse and the Financial Management Solutions Warehouse.

When using a shared dimension, the system consistently interprets attributes; hence rollups across data marts are possible and consistent. When a warehouse is provided data from multiple sources, a shared dimension is typically (but not always) built from multiple source structures.

Image: EPM conformed dimension

The following is a sample MDW shared dimension shown in Application Designer.

Num	Field Name	Type	Len	Format	Short Name	Long Name
1	CUSTOMER_SID	Nbr	10	Raw B	CUSTOMER_SID	CUSTOMER_SID
2	SETID	Char	5	Upper	SetID	SetID
3	SRC_SETID	Char	5	Upper	Source SetID	Source SetID
4	CUST_ID	Char	15	Upper	Customer	Customer ID
5	BO_ID_CUST	Nbr	31		CUST BO_ID	Customer BO_ID
6	SRC_SYS_ID	Char	5	Upper	Source ID	Source System Identifica
7	MAPPED_CUST_ID	Char	15	Upper	Mapped Cust ID	Mapped Customer ID
8	EFFDT	Date	10		Eff Date	Effective Date
9	CUSTOMER_SD	Char	20	Upper	Cust Short Desc	Customer Short Descripti
10	CUSTOMER_LD	Char	50	Upper	Customer Descr	Customer Description
11	CUSTOMER_IND	Char	1	Upper	Customer	Customer Indicator
12	CORPORATE_SETID	Char	5	Upper	SetID	SetID
13	CORPORATE_CUST_ID	Char	15	Upper	Corporate	Corporate Customer
14	CURRENCY_CD	Char	3	Upper	Currency	Currency Code
15	CUSTOMER_GROUP	Char	10	Upper	Cust Group	Customer Group
16	CUSTOMER_LVL	Char	1	Upper	Customer Level	Customer Level
17	BO_TYPE_ID	Nbr	6		BO Type ID	Business Object Type ID
18	CUSTOMER_TYPE_CD	Char	10	Upper	Type	Customer Type
19	CUSTOMER_TYPE_LD	Char	30	Upper	Cust. Type Desc	Customer Type Descripti
20	CUST_STATUS	Char	1	Upper	Status	Customer Status
21	CUST_STATUS_LD	Char	10	Upper	Cust Status Des	Customer Status Descrip
22	CUST_STAT_DT_SID	Nbr	10	Raw B	Cust Date SID	Customer Status Date SI
23	SINCE_DT_SID	Nbr	10	Raw B	Since Date SID	Since Date SID
24	ADD_DT_SID	Nbr	10	Raw B	Add date SID	Add date SID
25	DO_NOT_CONTACT	Char	1	Upper	Don't Contact	Do Not Contact
26	DO_NOT_EMAIL	Char	1	Upper	No Email	Do Not Email

MDW Dimension Table Naming Convention

MDW dimension tables use the following naming convention: D_*[table name]*.

MDW Fact Tables

MDW fact tables (F_*) contain numeric performance measurement data—such as quantity, sales, and revenue—that is used to build a data warehouse and its related reports. Facts help to quantify a company's activities. A fact is a typically an additive business performance measurement. That is, you can usually perform arithmetic functions on facts.

In a star schema, a fact table is the central table, each element of which is a foreign key derived from a dimension table. Dimension tables have a surrogate ID column that is the primary key of that dimension. A fact table may use these dimension surrogate IDs as foreign keys to the dimension table. In the dimensional model example graphic presented previously, the Sales fact table contains six foreign keys, each one matching a dimension surrounding the fact table.

Periodic Snapshot Fact Tables

Periodic Snapshots provide a view of the cumulative performance of the business at regular, predictable time intervals. Unlike a transaction fact table that loads a row of data for each event occurrence, the periodic snapshot fact table captures the event at the interval of a day, week, or month, and another capture at the interval of the next period, and so on. These periodic snapshots are stacked consecutively into the fact table. The periodic snapshot fact table often is the only place to easily retrieve a regular, predictable, trend view of the key business performance metrics.

Accumulating Fact Tables

Accumulating snapshots represent an indeterminate time span, covering the complete life of a transaction or discrete product. Accumulating snapshots almost always have multiple date stamps, representing the predictable major events or phases that take place during the course of a lifetime. Since many of these dates are not known when the fact row is first loaded, we must use surrogate date keys to handle undefined dates.

MDW Fact Table Naming Convention

MDW fact tables use the following naming convention: F_*[table name]*.

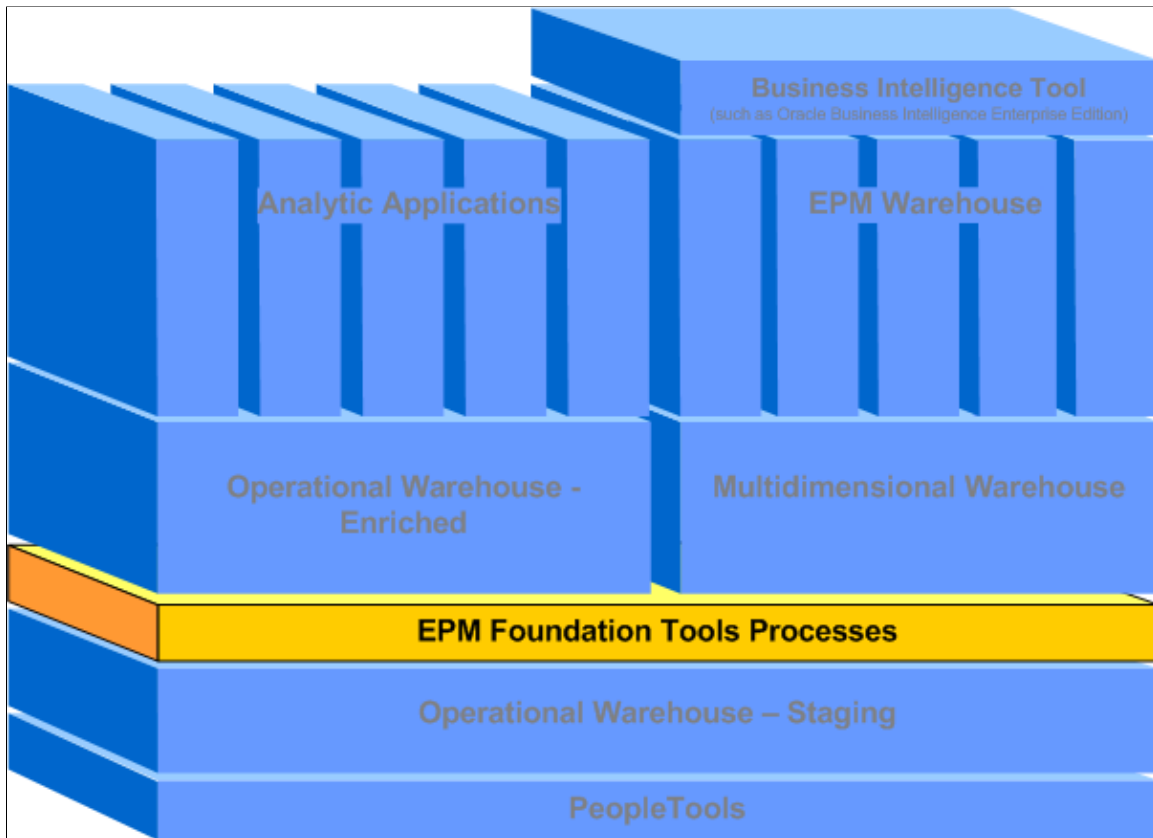
EPM Foundation Toolset

EPM is delivered with EPM Foundation tools. These set of tools enable you to enrich, audit, and manage the rich content included with EPM with a high degree of automation. For example, the Clone Metadata tool enables you to quickly and easily create a duplicate copy of your existing metadata. EPM Foundation

tools can be used with content included in the Operational Warehouse and the Multidimensional Warehouse.

Image: EPM Foundation

The following graphic illustrates how the EPM foundation tools and processes fit into the overall EPM architecture.



The following sections provide additional details about EPM Foundation tools.

Setup Tools

Implementing EPM requires that you specify parameters within the warehouse that reflect your organization's basic business processes and parameters. For example, you must define parameters for unit of measure, country, and accounting calendars in EPM .

EPM delivers several setup tools which enable you to quickly and easily setup basic information in the warehouse including unit of measure, multiple language and currency, and operator defaults.

See [Understanding PeopleSoft EPM Business Rule Setups](#).

Security Tools

EPM security enables you to set up data access at a variety of entry points and control access to meet your business needs, right down to an individual field. Security tools enable you to:

- Use application security to control access to applications, menus, and objects. You can specify which applications are available to a group of users, which menus and EPM objects they can access.

- Use row-level security, for example, to implement dimension-level access to particular products, customers, or key performance metrics. This ensures that highly sensitive data is protected.

You can also set up a specific security for the IBM WebSphere ETL tool, see [Understanding EPM Security and Setups](#)

Data Storage and Classification Tools

Implementing EPM involves configuring the system's structures to how your business operates. You can share common tables across reporting and analytical applications to minimize redundant data and system maintenance tasks.

Record metadata, for example, defines the first level of EPM metadata. It is used to identify and classify the tables that constitute the EPM data model. The record metadata identifies EPM tables as fact tables, fact reference tables, dimension tables, dimension reference tables, or transaction-dated tables. Each table is also classified to a specific data layer: the OWE or the MDW.

Tree manager provides an intuitive way to create, view, and maintain hierarchical definitions. An easy to understand user interface facilitates the creation and maintenance of trees. Tree mover enables you to moved PeopleSoft trees between different PeopleSoft application databases.

See [Understanding PeopleSoft EPM Business Rule Setups](#), and [Understanding Metadata](#).

Performance Management Related Tools

EPM utilizes shared components that provide functionality key to supporting high-volume analytical applications:

- Reusable filters and constraints stored in the metadata enable you to define sets of rules that can be shared across applications.
- Jobstreams streamline analytic processes and enable applications to run concurrently.

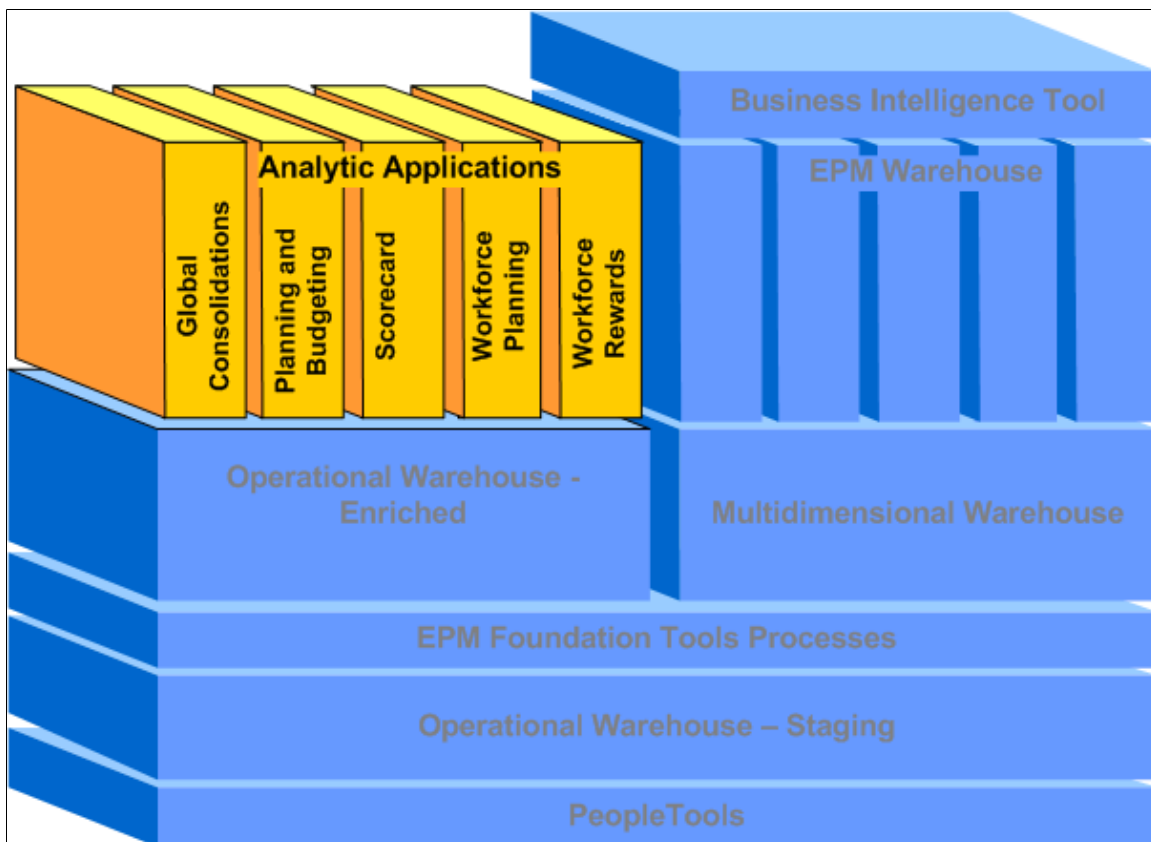
See [Understanding Metadata](#), and [Understanding Jobstreams](#).

PeopleSoft EPM Analytical Applications

EPM provides the applications necessary to analyze business situations, model business scenarios, and monitor performance.

Image: EPM Analytical Applications

The following graphic illustrates how the Analytical Applications fit into the overall EPM architecture.



PeopleSoft EPM analytical applications consist of a series of logically integrated dashboards, reports, scorecards, and models, which enable a wide range of users to access, analyze and act on integrated information within the context of a specific business processes, such as sales pipeline, accounts payable, or risk adjusted profitability analysis. These applications are built on normalized data from the OWE (rather than dimensionalized data from the MDW), and data is processed using the powerful Application Engine (AE) PeopleTool.

The analytical applications provide functional models for forecasting, trend analysis, and statistical analysis of your data to help drive decision-making within a business process. You model and analyze data using EPM PeopleSoft Internet Architecture (PIA) pages and application engines delivered with the analytical applications.

The EPM analytical applications enable you to:

- Capture organization-wide financial and workforce information in a single data model.
- Prepare and analyze plans in real time without having to move data between applications.
- Create and revise forecasts based on data from previous planning scenarios.

- View summary level information about your organization or drill back directly to transaction source system data.
- Deliver information to all levels within the organization.
- Continuously assess operational performance based on key performance indicators and historical trends.

Application Engine PeopleTool

The Application Engine is a PeopleTool that enables you to run and monitor SQL processing programs that process your data within the analytical applications. A program is defined in Application Designer and performs a specific business process. The program can consist of a set of SQL statements, PeopleCode, and program control actions that enable looping and conditional logic.

The Application Engine does not generate SQL or PeopleCode. It executes the SQL and PeopleCode included in an Application Engine action as part of a program. The Application Engine is designed for batch processing where data that must be processed without user intervention—for example, calculating salaries in payroll processing (although not printing the checks). Another example might be converting money from one currency to another.

PeopleSoft Delivered Analytical Applications

PeopleSoft delivers the following Applications:

- PeopleSoft Activity-Based Management
- PeopleSoft Application Fundamentals for Financial Services Industry
- PeopleSoft Funds Transfer Pricing
- PeopleSoft Global Consolidations
- PeopleSoft Performance Management Portal Pack
- PeopleSoft Planning and Budgeting
- PeopleSoft Project Portfolio Management
- PeopleSoft Risk Weighted Capital
- PeopleSoft Scorecard
- PeopleSoft Workforce Analytics Applications
- PeopleSoft Workforce Planning
- PeopleSoft Workforce Rewards

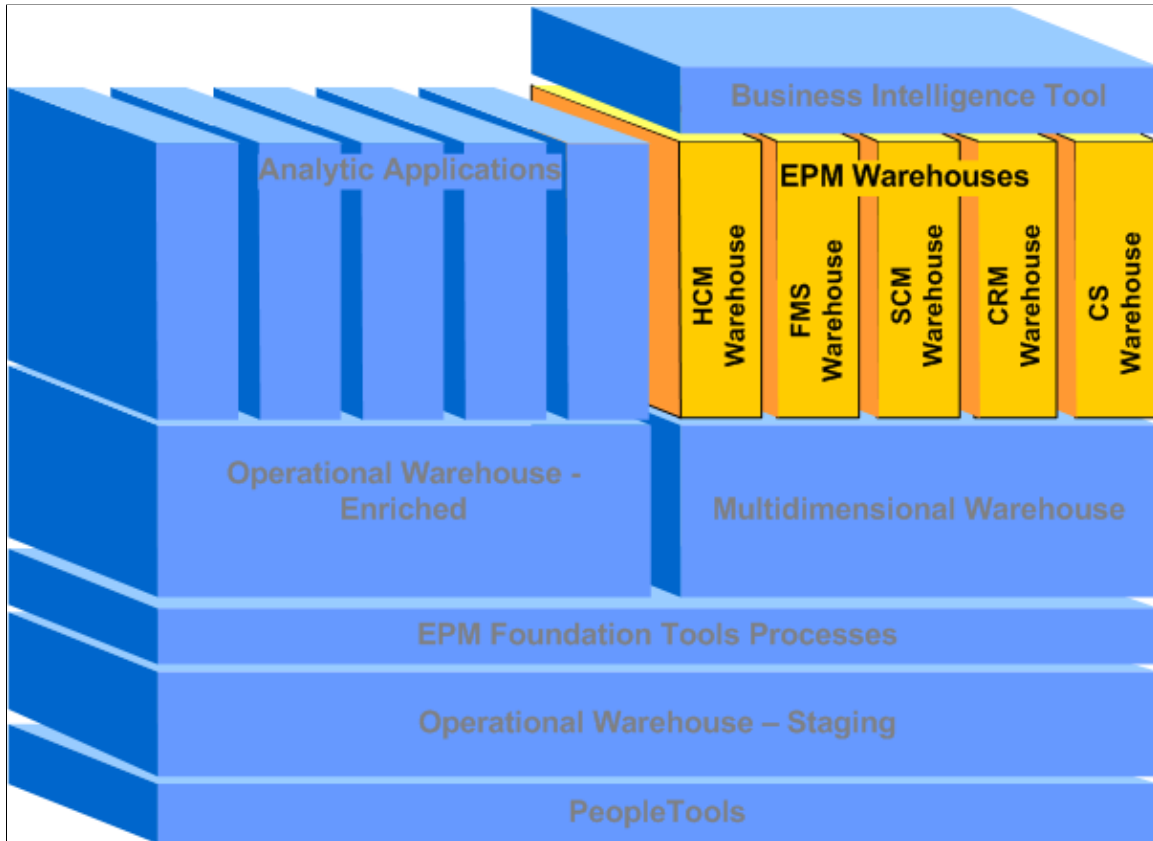
For more details on the application or applications you have licensed, please refer to the specific application documentation.

PeopleSoft EPM Warehouses and Reporting

The PeopleSoft EPM warehouses provide you with the tools and technology to manage your organization's information that is used for reporting and analysis. Each warehouse is divided into multiple subject areas, or data marts. Each data mart is aligned with a business process, which enables you to answer strategic questions essential to your organization's bottom line.

Image: EPM Warehouses

The following graphic illustrates how the EPM warehouses fit into the overall EPM architecture.



PeopleSoft provides the following EPM warehouses:

- Campus Solutions Warehouse
- Customer Relationship Management (CRM) Warehouse
- Financials Management Solution (FMS) Warehouse
- Financials Warehouse for Public Sector and Higher Education
- Human Capital Management (HCM) Warehouse
- Supply Chain Management (SCM) Warehouse

Prepackaged Content

PeopleSoft delivers the following content with each EPM warehouse:

- Extract Transform and Load (ETL) component
- Infrastructure tables and tools
- Security tables
- Staging tables
- Multidimensional Warehouse tables
- Data Models
- Measures

Reporting tables are built in the MDW to enable offloading of operational reports from your transactional systems. As part of your implementation, you need to consider which operational reports it makes sense to offload to the EPM warehouses.

The EPM warehouse documentation (such as the *PeopleSoft EPM: Campus Solutions Warehouse*) provides more details on the each of the EPM warehouses.

Implementing PeopleSoft EPM

Understanding EPM Implementation

There is no singular approach to implementing EPM; each implementation varies depending on the EPM products you license. For example, many of the implementation tasks required to implement an EPM Analytical application (such as Global Consolidations) are different from the implementation tasks required to implement an EPM Warehouse application (such as the Campus Solutions Warehouse). However, there are some implementation tasks common to both EPM product lines and generally speaking, implementing EPM requires that you:

1. Install EPM and applicable third-party products (such as IBM WebSphere DataStage).
2. Set up EPM core infrastructure (which serves as the underlying framework for the EPM Warehouses and Analytical Applications).

Examples of infrastructure setups include setting up currency codes (CURRENCY_CD_TB) and record metadata (META_REC_TBL).

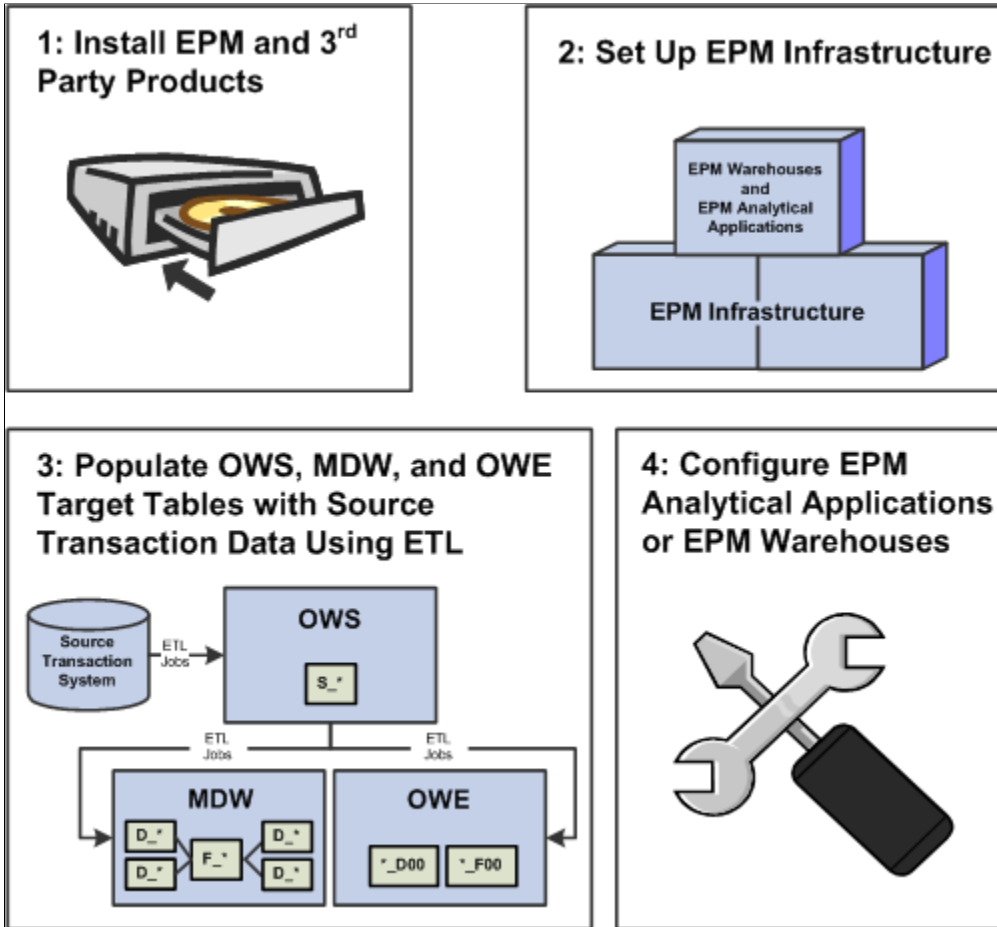
3. Populate Operational Warehouse - Staging (OWS), Multidimensional Warehouse (MDW), and Operational Warehouse - Enriched (OWE) target warehouse tables with your source transaction data using the extract, transform, and load (ETL) tool.

You also conform your disparate source transaction data using the ETL tool.

- Configure either the EPM Warehouses or Analytical Applications for your business (depending on which EPM product you license).

Image: Basic EPM Implementation Steps

This diagram illustrates the tasks that are common to implementing an EPM Analytical application and an EPM Warehouse application.



The following topics will help you better understand which implementation tasks apply to your implementation and how to use the EPM suite of documentations to locate the instructions for those tasks.

Note: PeopleSoft Setup Manager can also help you determine which specific tasks are required for your implementation by generating a list of setup tasks based on the features you license. The list of setup tasks include the components that you must set up, listed in the order in which you must enter data into the component tables, and links to the corresponding documentation.

EPM Installation Tasks

The first step in implementing EPM is to install all the necessary software on your designated machine(s).

The following table provides an *example* of some of the installation tasks you perform to implement EPM:

Task	Common, EPM Warehouses, or Analytical Applications?	Documentation Resource	Documentation Location
Pre-installation	Common	PeopleSoft Pre-Installation Checklist	See My Oracle Support.
Review hardware/software requirements	Common	PeopleSoft Hardware and Software Guide	See My Oracle Support.
Install IBM WebSphere DataStage	Common	<i>PeopleSoft Enterprise Performance Management Installation Guide</i> <i>IBM Information Server: Planning Installation and Configuration Guide</i>	Installation CD
Install EPM	Common	<i>PeopleSoft Enterprise Performance Management Installation Guide</i>	Installation CD

EPM Core Infrastructure and ETL Setup Tasks

The second step in implementing EPM is to set up EPM infrastructure tables and populating warehouse target tables with source transaction data. Some of these tasks include defining currency conversion methodology, setting up warehouse business units, and configuring ETL environmental parameters.

The core infrastructure and ETL setup tasks are grouped as follows:

- Core infrastructure setups.
- Common ETL setups.

Core Infrastructure Setups

The following table provides an *example* of some of the core EPM infrastructure setup tasks you perform to implement EPM:

Task	Common, EPM Warehouses, or Analytical Applications?	Documentation Resource	Documentation Location
Specify EPM Sources	Common	<i>PeopleSoft Enterprise Performance Management Fundamentals</i> , “Setting Up EPM Business Rules”	See Specifying Your EPM Sources .
Specify Country and State Info	Common	<i>PeopleSoft Enterprise Performance Management Fundamentals</i> , “Setting Up EPM Business Rules”	See Setting Up Country and State Information .

Task	Common, EPM Warehouses, or Analytical Applications?	Documentation Resource	Documentation Location
Define Accounting Calendars	Common	<i>PeopleSoft Enterprise Performance Management Fundamentals</i> , “Setting Up EPM Business Rules”	See Defining Accounting Calendars .
Set Up Gregorian Calendar	EPM Warehouses	<i>PeopleSoft Enterprise Performance Management Fundamentals</i> , “Setting Up EPM Business Rules”	See Setting Up the Gregorian Calendar [EPM Warehouses] .
Define Units of Measure	Common	<i>PeopleSoft Enterprise Performance Management Fundamentals</i> , “Setting Up EPM Business Rules”	See Defining Units of Measure .
Define Dimensions	Common	<i>PeopleSoft Enterprise Performance Management Fundamentals</i> , “Setting Up EPM Business Rules”	See Defining and Maintaining Dimensions .
Set Up Currency Tables	Common	<i>PeopleSoft Enterprise Performance Management Fundamentals</i> , “Setting Up Currency Rules for EPM”	See Setting Up EPM Currency Tables .
Define Market Rates	Common	<i>PeopleSoft Enterprise Performance Management Fundamentals</i> , “Setting Up Currency Rules for EPM”	See Setting Up Market Rates for EPM Currency Conversion .
Define Currency Quotations	Common	<i>PeopleSoft Enterprise Performance Management Fundamentals</i> , “Setting Up Currency Rules for EPM”	See Defining Currency Quotations for EPM Currency Conversion .
Set Up Currency Rate Calculations	Common	<i>PeopleSoft Enterprise Performance Management Fundamentals</i> , “Setting Up Currency Rules for EPM”	See Calculating Currency Rates for EPM Currency Conversion .
Set Up EPM Security	Common	<i>PeopleSoft Enterprise Performance Management Fundamentals</i> , “Setting Up EPM Security”	See Understanding EPM Security and Setups .

Common ETL Setups

The following table provides an *example* of some of the common ETL setup tasks you perform to implement EPM:

Task	Common, EPM Warehouses, or Analytical Applications?	Documentation Resource	Documentation Location
Configure IBM WebSphere DataStage for EPM	Common	<i>PeopleSoft Enterprise Performance Management Fundamentals</i> , “Preparing to Load Source Data Into EPM” and “Setting Up DataStage for EPM”	See Understanding IBM WebSphere DataStage , and Understanding IBM WebSphere DataStage Setup and Configuration .
Specify ETL parameters	Common	<i>PeopleSoft Enterprise Performance Management Fundamentals</i> , “Defining ETL Parameters”	See Defining ETL Parameters for the Planning and Budgeting Analytical Application , and Defining ETL Parameters for the EPM Warehouses .
Run initial setup jobs	Common	<i>PeopleSoft Enterprise Performance Management Fundamentals</i> , “Running Initial Setup Jobs”	See Running ETL Setup Jobs to Bring Source Data Into EPM .
Import source business units into EPM	Common	<i>PeopleSoft Enterprise Performance Management Fundamentals</i> , “Importing Source Business Units into EPM to Create Warehouse Business Units”	See Understanding Warehouse Business Units, TableSet Sharing, and SetID Mapping and Understanding Warehouse Business Unit Setup .

OWE / MDW Specific Setup Tasks

You may recall from the [Overview](#) topic that the OWE structure stores data arranged in a normalized format for the Analytical Applications, and the MDW structure stores data arranged in a dimensional schema for the EPM Warehouses. Hence, implementation tasks related to the MDW affect the EPM Warehouses and tasks related to the OWE affect the Analytical Applications.

After all installation and core EPM implementation tasks are completed, the final step is to perform specific implementation tasks that apply only to either the EPM Warehouses or the Analytical Applications. For example, if you purchased the Global Consolidations analytical application you must set up record metadata, which is an implementation task specific to the analytical applications.

MDW Specific Setups

MDW specific setups are documented in EPM warehouse documentation (for example, *PeopleSoft EPM: Campus Solutions Warehouse*).

The following table provides an *example* of some of the MDW specific setup tasks you perform to implement an EPM warehouse:

Task	Documentation Resource
Run Global Dimension jobs for your specific warehouse (Campus Solutions Warehouse, for example).	<i>Running [product name] Warehouse Implementation Jobs</i> (for example, Running Campus Solutions Warehouse Implementation Jobs) PeopleSoft EPM: [product name] Warehouse documentation (for example, <i>PeopleSoft EPM: Campus Solutions Warehouse</i>)
Run SKU jobs for your specific warehouse (Supply Chain Management Warehouse, for example)	<i>Running [product name] Warehouse Implementation Jobs</i> PeopleSoft EPM: [product name] Warehouse
Configure slowly changing dimensions	<i>Configuring Slowly Changing Dimensions</i> PeopleSoft [product name] Warehouse
Implement currency conversion	<i>Implementing Currency Conversion</i> PeopleSoft EPM: [product name] Warehouse
Implement multilanguage conversion	<i>Setting Up Multilanguage Processing and Running the Language Swap Utility</i> PeopleSoft EPM: [product name] Warehouse
Denormalize trees and recursive hierarchies	<i>Processing Trees and Recursive Hierarchies</i> PeopleSoft EPM: [product name] Warehouse

PeopleSoft EPM provides the following EPM warehouse documentation, where you can find MDW specific setup tasks to implement an EPM warehouse:

- PeopleSoft Campus Solutions Warehouse
- PeopleSoft Customer Relationship Management Warehouse
- PeopleSoft Financial Management Solutions Warehouse
- PeopleSoft Financials Warehouse for Public Sector and Higher Education
- PeopleSoft PeopleSoft Human Capital Management Warehouse
- PeopleSoft PeopleSoft Supply Chain Management Warehouse

OWE Setups

OWE specific setups are documented in this documentation and the EPM Analytical Application documentations (for example, PeopleSoft EPM: Global Consolidations).

The following table provides an *example* of some of the OWE specific setup tasks you perform to implement an EPM Analytical Application:

Task	Documentation Resource	Documentation Location
Define record metadata	<i>PeopleSoft Enterprise Performance Management Fundamentals</i> , "Setting Up and Working with Metadata for the Operational Warehouse - Enriched"	See Setting Up Record Metadata .
Set up datamaps	<i>PeopleSoft Enterprise Performance Management Fundamentals</i> , "Setting Up and Working with Metadata for the Operational Warehouse - Enriched"	See Setting Up Datamaps .
Specify ledger mapping defaults	<i>PeopleSoft Enterprise Performance Management Fundamentals</i> , "Setting Up Business Rules for the Operational Warehouse - Enriched"	See Specifying Ledger Mapping Defaults .
Implement currency conversion	<i>PeopleSoft Enterprise Performance Management Fundamentals</i> , "Setting Up Business Rules for the Operational Warehouse - Enriched"	See Setting Up and Running Currency Conversion .
Define jobstreams	<i>PeopleSoft Enterprise Performance Management Fundamentals</i> , "Streamlining Processing with Jobstreams"	See Understanding Jobstreams .
Define allocation rules	<i>PeopleSoft Enterprise Performance Management Fundamentals</i> , "Using Data Enrichment Tools"	See Defining Allocation Manager Rules .
Define your organizational structure (example of a setup specifically for the Global Consolidations analytical application)	<i>PeopleSoft EPM: Global Consolidations</i> , "Defining Your Organizational Structure"	See "Understanding Organizational Structure Requirements (<i>PeopleSoft EPM 9.1: Global Consolidations</i>)".
Establish KPIs (example of a setup specifically for the Scorecard analytical application)	<i>PeopleSoft EPM: Scorecard</i> , "Establishing and Maintaining KPIs"	See "Defining KPI Building Blocks (<i>PeopleSoft EPM 9.1: Scorecard</i>)".

Using the EPM Suite of Documentation to Locate Implementation Task Instructions

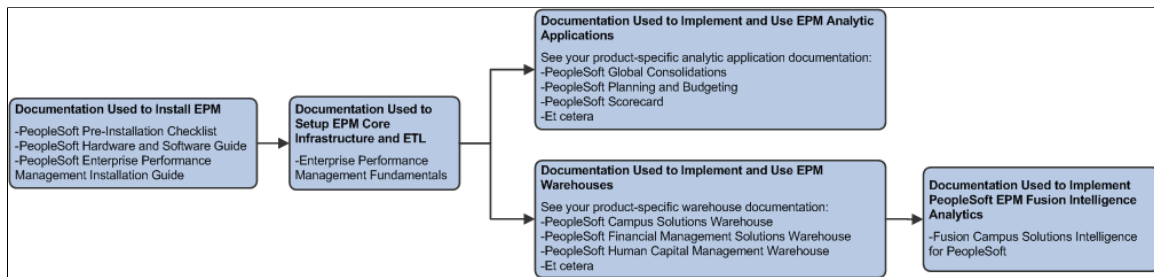
The suite of EPM Documentation are organized into four general categories:

- **EPM Installation:** This documentation contains EPM installation information, hardware and software guidelines, and third-party product installation information (such as IBM WebSphere DataStage).
- **EPM Fundamentals:** This documentation contains core infrastructure setup common to all EPM products, ETL setup and processing information, and some setups for the EPM Warehouses and Analytical Applications.
- **EPM Warehouses:** This documentation contains specific set up and configuration information particular to each EPM warehouse.
- **EPM Analytical Applications:** This documentation contains specific setup and configuration information particular to each analytical application.

The organization of the suite of EPM documentation represent the various steps required in an EPM implementation (installation, core infrastructure setup, ETL setup and processing, and EPM Warehouse or Analytical Application configuration). You progress through the different EPM documentation in the same order you progress through an implementation, using the installation documentation first and the warehouse or analytical application documentation last.

Image: EPM documentation usage order

This diagram illustrates the usage order of PeopleSoft EPM documentation.



The following table lists all available EPM documentation by category:

EPM Installation	EPM Fundamentals	EPM Analytical Application	EPM Warehouse	PeopleSoft EPM Fusion Intelligence Analytics
<i>PeopleSoft Pre-Installation Checklist</i>	<i>PeopleSoft Enterprise Performance Management Fundamentals</i>	<i>PeopleSoft Activity-Based Management</i>	<i>PeopleSoft Campus Solutions Warehouse</i>	<i>Fusion Campus Solutions Intelligence for PeopleSoft</i>
<i>PeopleSoft Enterprise Performance Management Hardware and Software Requirements Guide</i>		<i>PeopleSoft Application Fundamentals for Financial Services Industry</i>	<i>PeopleSoft Customer Relationship Management Warehouse</i>	
<i>PeopleSoft Enterprise Performance Management Installation Guide</i>		<i>PeopleSoft Funds Transfer Pricing</i>	<i>PeopleSoft Financial Solutions Management Warehouse</i>	

<i>EPM Installation</i>	<i>EPM Fundamentals</i>	<i>EPM Analytical Application</i>	<i>EPM Warehouse</i>	<i>PeopleSoft EPM Fusion Intelligence Analytics</i>
		<i>PeopleSoft Global Consolidations</i>	<i>Financials Warehouse for Public Sector and Higher Education</i>	
		<i>PeopleSoft Performance Management Portal Pack</i>	<i>PeopleSoft Human Capital Management Warehouse</i>	
		<i>PeopleSoft Planning and Budgeting</i>	<i>PeopleSoft Supply Chain Management Warehouse</i>	
		<i>PeopleSoft Project Portfolio Management</i>		
		<i>PeopleSoft Risk Weighted Capital</i>		
		<i>PeopleSoft Scorecard</i>		
		<i>PeopleSoft Workforce Analytic Applications</i>		
		<i>PeopleSoft Workforce Planning</i>		
		<i>PeopleSoft Workforce Rewards</i>		

Note: If you are implementing EPM with the assistance of a PeopleSoft consultant, the consultant can access a searchable, online version of the aforementioned documentation from My Oracle Support.

Using This Documentation to Locate Implementation Tasks

The structure of the *PeopleSoft Enterprise Performance Management Fundamentals* documentation is designed to help you locate the specific tasks required for your implementation, which depends on the EPM product you license. The structure of this documentation separates topics and implementation tasks according to whether they relate to all EPM products (common), or only the OWE and the EPM Analytical Applications. Implementation tasks for the MDW and EPM Warehouses are located in separate documentation.

Topics located in this documentation contain an overview of EPM and core infrastructure and ETL setup information necessary to implement all EPM products (warehouses and analytical applications). Other

topics located in this documentation contain setup information and optional configurations for the OWE, which are necessary for implementing the EPM Analytical Applications.

Image: Structure of the Enterprise Performance Management Fundamentals documentation.

This graphic illustrates the structure of the Enterprise Performance Management Fundamentals documentation.

EPM Fundamentals Documentation Structure	
These topics contain EPM overview, core infrastructure setup, and ETL information common to implementing all EPM warehouses and analytic applications.	Getting Started With PeopleSoft Enterprise Performance Management
	Understanding PeopleSoft Enterprise Performance Management
	Implementing PeopleSoft EPM
	Setting Up EPM Business Rules
	Setting Up Currency Rules for EPM
	Setting Up EPM Security
	Preparing to Load Source Data Into EPM
	Using DataStage Administrator
	Using DataStage Designer
	Using DataStage Director
	Setting Up DataStage for EPM
	Defining ETL Parameters
	Running Initial Setup Jobs
	Importing Source Business Units into EPM to Create Warehouse Business Units
ETL Configurations	
These topics contain setup information and optional configurations for the Operational Warehouse - Enriched, which affect the EPM analytic applications.	Setting Up and Working with Metadata for the Operational Warehouse - Enriched
	Working with Metadata Utilities
	Setting Up Business Rules for the Operational Warehouse - Enriched
	Setting Up Models and Scenarios
	Streamlining Processing with Jobstreams
	Setting Up and Using Profit Manager
	Using Data Enrichment Tools
	Creating XBRL Instance Documents in EPM

Regardless of the EPM product you purchase, you must perform the tasks documented in the core infrastructure and ETL setup topics (see preceding graphic), as these tasks are required for all EPM products. However, if you purchase an EPM Analytical Application, you must also perform the tasks documented in OWE-related topics (see preceding graphic). If you purchase an EPM Warehouse, you need not perform the tasks documented in the OWE-related topics, but you must refer to EPM Warehouse implementation tasks in your warehouse-specific documentation, for example, *PeopleSoft Campus Solutions Warehouse*.

Chapter 4

Setting Up EPM Business Rules

Setting Up EPM Business Rules

These topics provide overviews of PeopleSoft EPM business rule setups and discuss how to:

- Review installed products.
- Specify your EPM sources.
- Set up country and state information.
- Define accounting calendars.
- Set up the Gregorian calendar. [EPM Warehouses]
- Set up time zones. [EPM Warehouses]
- Define units of measure.
- Define and maintain dimensions.
- Define operator defaults.
- Archive EPM data.
- Set up chart viewing and printing results [Analytical Applications]
- Set up ledger-based nVision reporting for EPM

Note: When the term *EPM Warehouses* or *Analytical Applications* appears in the task titles below, it means that the task applies only to that particular EPM product line. When neither term appears in the task title, it means the task applies to both EPM product lines.

Understanding PeopleSoft EPM Business Rule Setups

After installing EPM, you must set up the infrastructure, key business rules, and processing variables that support the product. These rules provide the foundation on which EPM operates and can include country and state, unit of measure, and calendar rules. Because these rules act as a foundation and span the entire EPM product suite, you must perform these setup tasks prior to beginning other EPM implementation tasks documented in this guide.

The business rules and processing variables that you define for EPM are stored in EPM database tables. You define these rules and populate the EPM tables using the PeopleSoft Pure Internet Architecture (PIA) pages in the EPM Foundation setup menu. These pages are built over EPM tables that are shared across many EPM products.

Note: Please be aware that the order in which the setup tasks are presented in this topic does not necessarily indicate the actual order in which they should be performed during implementation. It is highly recommended that you use Setup Manager to determine the implementation tasks that are required for your organization and the related implementation sequence.

Reviewing Installed Products

Before you can begin working with any PeopleSoft EPM application, you must specify installation options to indicate how you plan to use the system and which applications you will be using. Installation options are defined for your entire database—they are not specific to a business unit or SetID.

This topic discusses how to:

- Review installed PeopleSoft EPM products.
- Review installed PeopleSoft Enterprise Resource Planning (ERP) products.
- Set Web Services options.

Pages Used to Set Installation Options

<i>Page Name</i>	<i>Definition Name</i>	<i>Navigation</i>	<i>Usage</i>
EPM Products	INSTALLATION_PF1	EPM Foundation, EPM Setup, Installation Analysis & Options, Installation Options, EPM Products	Review installed PeopleSoft EPM products.
ERP Products	INSTALLATION_PF2	EPM Foundation, EPM Setup, Installation Analysis & Options, Installation Options, ERP Products	Review installed PeopleSoft ERP products.
Web Services	INSTALLATION_PF3	EPM Foundation, EPM Setup, Installation Analysis & Options, Installation Options, Web Services	Set web services options.

ERP Products Page

Use the ERP Products page (INSTALLATION_PF2) to review installed PeopleSoft ERP products.

Navigation

EPM Foundation, EPM Setup, Installation Analysis & Options, Installation Options, ERP Products

Image: ERP Products page

This example illustrates the fields and controls on the ERP Products page. You can find definitions for the fields and controls later on this page.

The screenshot shows the 'ERP Products' tab selected in a navigation bar. Below the tab is a section titled 'Other Peoplesoft Products' containing a grid of checkboxes for various products. The products and their selection status are as follows:

Product Name	Selected
Asset Management	<input type="checkbox"/>
Billing	<input type="checkbox"/>
Expenses	<input type="checkbox"/>
General Ledger	<input checked="" type="checkbox"/>
Inventory	<input type="checkbox"/>
Manufacturing	<input type="checkbox"/>
Order Management	<input type="checkbox"/>
Payables	<input checked="" type="checkbox"/>
Product Configurator	<input type="checkbox"/>
Projects	<input type="checkbox"/>
Public Sector	<input type="checkbox"/>
Purchasing	<input type="checkbox"/>
Receivables	<input checked="" type="checkbox"/>
Treasury	<input type="checkbox"/>

Other PeopleSoft Products

Any other PeopleSoft ERP products that have been installed in your database are automatically selected.

Web Services Page

Use the Web Services page (INSTALLATION_PF3) to set web services options.

Navigation

EPM Foundation, EPM Setup, Installation Analysis & Options, Installation Options, Web Services

Image: Web Services page

This example illustrates the fields and controls on the Web Services page. You can find definitions for the fields and controls later on this page.

The screenshot shows the 'Web Services' tab selected in a navigation bar. Below the tab is a section titled 'PIA Server Settings' containing the following fields and controls:

- Allowed Concurrent Merges:
- Process wait time (Seconds):
- Applets
- Chart Server

Below the PIA Server Settings is a section titled 'CBM Options' containing the following field and control:

- Model Vendor: Angoss

Allowed Concurrent Merges

Enter the number of concurrent merges allowed in your installation. The default is 3, which matches the number of record suites that are delivered with your system.

Process wait time (Seconds)

Enter the process wait time. The default is 60 seconds. Waiting improves performance of processes by eliminating repeated database requests.

Warning! Because each merge job in a jobstream reads in the number of concurrent updates, changing its value while merge processes are running can have unexpected results. Changes to the process wait time affect only subsequent runs of the engine.

Applets

Select if you are using applets with PeopleSoft Activity-Based Management or Scorecard. See the application-specific documentation for more information on using the chart server and enabling applets.

Chart Server

Select if you are using the chart server to create charts through PeopleSoft Activity-Based Management, Scorecard, or the financial services industry applications (PeopleSoft Risk-Weighted Capital, Asset Liability Management, or Funds Transfer Pricing).

Related Links

[Understanding Jobstreams](#)

"Viewing Scorecard Results (*PeopleSoft EPM 9.1: Scorecard*)"

Specifying Your EPM Sources

This topic provides an overview of the relationship between source system data and EPM, and discusses how to define warehouse sources.

Page Used to Specify EPM Sources

Page Name	Definition Name	Navigation	Usage
Define Warehouse Sources	PF_WAREHOUSE_SRC	EPM Foundation, EPM Setup, Warehouse Sources & Bus Units, Define Warehouse Sources	Specify a source ID for a source transaction system and define its basic attributes, such as base language and currency.

Understanding Source System Data and EPM

EPM enables you to extract, transform, and consolidate data from multiple source transaction systems into a series of target warehouse tables in the EPM database. This means that the EPM database is able to accommodate data from several PeopleSoft sources. Although the data is commingled in the EPM database, certain EPM processes require the ability to trace the data in target warehouse tables to its original source (original source transaction system). Other EPM processes use specific attributes associated with the source transaction system to process data. For example, the *language swap utility* uses the source system's base language for multiple language processing.

Because source system information is required, you must define the source in EPM and specify the attributes associated with that source using the Define Warehouse Sources page. Information specified on this page include source system ID (SRC_SYS_ID), base currency, and base language, and is used in several different EPM processes. Most notably, the source system ID is used in the ETL process and helps to maintain source data history as well as data uniqueness. Currency and rate type are used in multiple currency processing. The source system ID and default SetID are used later to define warehouse business units.

Single Signon and Your GL Source Database

The Global Consolidations analytical application has a feature that enables you to drill down to your source GL database from the consolidation audit using the PeopleTools *single signon* feature. The single signon feature enables you to access additional PeopleSoft FSCM transaction databases without entering an ID or a password after you have already been authenticated in another PeopleSoft database.

In order to use the Global Consolidations GL drill-down feature you must set up the single signon feature for each GL source in PeopleTools. You must also use the Define Warehouse Sources page to associate the GL source system ID with its corresponding single signon portal information. This information is used to build the URL to transfer to a GL instance.

See the product documentation for *PeopleSoft EPM: Global Consolidations*, and *PeopleTools: Security Administration*

Related Links

[Understanding Warehouse Business Unit Setup](#)

[Understanding Warehouse Business Units, TableSet Sharing, and SetID Mapping](#)

Setting Up Country and State Information

This topic discusses how to:

- Review country descriptions.
- Select an address format by country.
- Validate addresses.
- Define countries for reporting.
- Specify state information.

Pages Used to Set up Country and State Information

Page Name	Definition Name	Navigation	Usage
Country Table - Country Description	COUNTRY_DEFN	EPM Foundation, EPM Setup, Common Definitions, Country and State Info, Country Table, Country Description	Review country descriptions.

Page Name	Definition Name	Navigation	Usage
Country Table - Address Format	ADDR_FORMAT_TABLE	EPM Foundation, EPM Setup, Common Definitions, Country and State Info, Country Table, Address Format	Select an address format by country. You do so by configuring address fields and field descriptions so that addresses conform to the customary address format of the specified country. After the address format is set, it appears everywhere that the system uses the address subrecord.
Country Table - Valid Address	EO_ADDR_VALIDAT	EPM Foundation, EPM Setup, Common Definitions, Country and State Info, Country Table, Valid Address	Validate addresses by adding valid combinations of address fields.
Country Statistics	COUNTRY_STAT_PNL	EPM Foundation, EPM Setup, Common Definitions, Country and State Info, Country Statistics	Specify countries for reporting.
State	STATE_DEFN	EPM Foundation, EPM Setup, Common Definitions, Country and State Info, State Table, State	Specify state, province, or other geopolitical region information.

Country Table - Country Description Page

Use the Country Table - Country Description page (COUNTRY_DEFN) to review country descriptions.

Navigation

EPM Foundation, EPM Setup, Common Definitions, Country and State Info, Country Table, Country Description

Image: Country Description page

This example illustrates the fields and controls on the Country Description page. You can find definitions for the fields and controls later on this page.

The screenshot shows a web interface for the 'Country Description' page. At the top, there are three tabs: 'Country Description' (selected), 'Address Format', and 'Valid Address'. Below the tabs, the 'Country' field is set to 'GBR'. A dropdown menu is open, showing 'Country' as the selected item. Below the dropdown, there are three input fields: '*Description:' with the value 'United Kingdom', 'Short Description:' with the value 'UK', and '2-Char Country Code:' with the value 'GB'. To the right of the '2-Char Country Code' field, there is a checked checkbox labeled 'EU Member State'.

The system displays the country code, description, and a short description for the country.

PeopleSoft delivers a fully populated country code table, which is updated as national boundaries and designations change.

2-Char Country Code (two-character country code) Enter a country code. Country codes are informational indicators that record the value-added tax (VAT) registration ID for each country where your company does business.

EU Member State (European Union member state) Select to specify that the country is a member of the European Union.

Country Table - Valid Address Page

Use the Country Table - Valid Address page (EO_ADDR_VALIDAT) to validate addresses by adding valid combinations of address fields.

Navigation

EPM Foundation, EPM Setup, Common Definitions, Country and State Info, Country Table, Valid Address

To enable this page, select the Enable Address Search check box on the Address Format page, which enables the Used in Search column. The fields that you select to be used in a search appear on the Valid Address page as columns. Enter the valid address combinations that the user can search for and select.

Country Statistics Page

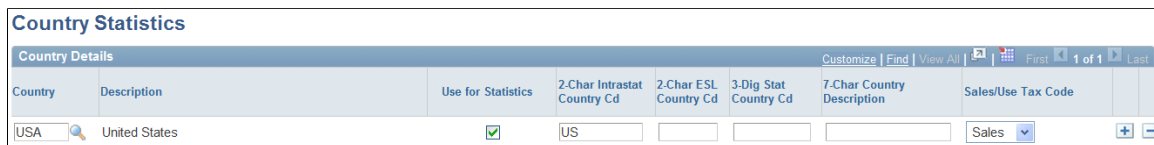
Use the Country Statistics page (COUNTRY_STAT_PNL) to specify countries for reporting.

Navigation

EPM Foundation, EPM Setup, Common Definitions, Country and State Info, Country Statistics

Image: Country Statistics page

This example illustrates the fields and controls on the Country Statistics page. You can find definitions for the fields and controls later on this page.



PeopleSoft EPM only uses this page to establish the countries for statistical purposes. So most of the information is for informational purposes only, because the reports mentioned are not available through the PeopleSoft EPM database. For more information on the reports mentioned in the following topic, refer to the appropriate PeopleSoft Financials application documentation.

Country Select the country on which you would like to report.

Use for Statistics If this check box is selected, the country recognized by the International Standards Organization (ISO) is also recognized by the European statistical offices. If the check box is not selected, the country is recognized only by the ISO. In this case, European statistical offices assume that the country is a

part of another country when producing statistical data such as the gross national product (GNP). For example, for the ISO, the principality of Monaco has its own country code, but for statistical purposes, Monaco is assumed to be part of France, and therefore has the same statistical country code as France.

2-Char Intrastat Country Cd (two-character intrastat country code)

The country identifier for the European statistical offices. This code often appears on intrastat layout forms.

2-Char ESL Country Cd (two-character ESL country code)

Used for countries that are members of the European Union. It usually appears as part of the VAT registration information on the European Sales List (ESL).

3-Dig Stat Country Cd (three-digit statistical country code)

Used as a country identifier for the European statistical offices. It usually appears on the intrastat report to identify source or destination countries to which or from which goods are shipped.

7-Char Country Descr (seven-character country description)

Used for the German international Electronic Funds Transfer (EFT) layout. It identifies countries into which or from which electronic funds are sent or received.

Sales/Use Tax Code

Informational only. This field can be set to *None*, *Sales*, or *Use*.

State Page

Use the State page (STATE_DEFN) to specify state, province, or other geopolitical region information.

Navigation

EPM Foundation, EPM Setup, Common Definitions, Country and State Info, State Table, State

Image: State page

This example illustrates the fields and controls on the State page. You can find definitions for the fields and controls later on this page.

The screenshot shows a form titled "State" with the following fields and values:

Country:	FRA	France
Postal Abbreviation:	59	
Numeric Code:	<input type="text" value="59"/>	
*Description:	<input type="text" value="Nord"/>	

PeopleSoft delivers a fully populated state code table. PeopleSoft updates the state table as changes occur. This table provides states, provinces, and equivalent geopolitical entities (for example, Dutch communities and French departments) for all supported countries. The codes are based on standard postal codes.

Numeric Code

Assign a number to a state or province for statistics and reporting purposes.

Defining Accounting Calendars

This topic provides an overview of accounting calendar setup, lists common elements, and discusses how to:

- Define a base calendar.
- Modify or add detail calendars.
- Define summary calendars.
- (Optional) Define business calendars.
- (Optional) Define daily calendars.
- (Optional) Define budget calendars.
- Define calendar frequencies.
- Define frequency details.
- Define timespans.

Pages Used to Define Accounting Calendars

<i>Page Name</i>	<i>Definition Name</i>	<i>Navigation</i>	<i>Usage</i>
Calendar Builder	CALENDAR_BUILDER	EPM Foundation, EPM Setup, Common Definitions, Calendars, Calendar Builder	Define a base calendar to be used by other calendars that you create. The calendar that you create is by default a detail calendar.
Detail Calendar	DETAIL_CALENDAR1	EPM Foundation, EPM Setup, Common Definitions, Calendars, Detail Calendar	Modify or add detail accounting calendars that match or differ from your general ledger calendars.
Summary Calendar	SUMMARY_CALENDAR	EPM Foundation, EPM Setup, Common Definitions, Calendars, Summary Calendar, Summary Calendar	Define a summary calendar to group or combine periods from detail calendars.
Summary Budget Period Calendar	SUMMARY_BP_CAL	EPM Foundation, EPM Setup, Common Definitions, Calendars, Summary BP Calendar, Summary Budget Period Calendar	Define a summary budget period calendar to group or combine periods from detail calendars.

<i>Page Name</i>	<i>Definition Name</i>	<i>Navigation</i>	<i>Usage</i>
Business Day Calendar	BUS_CALENDAR	EPM Foundation, EPM Setup, Common Definitions, Calendars, Business Day Calendar	Define a business calendar to specify your normal business week and the holidays observed by your banks.
Daily Calendar	DAILY_CALENDAR	EPM Foundation, EPM Setup, Common Definitions, Calendars, Daily Calendar	Define a daily calendar for reporting purposes.
Budget Calendar	CAL_BP_TBL	EPM Foundation, EPM Setup, Common Definitions, Calendars, Budget Calendar	Define a budget calendar for controlled budget processing.
Frequency Definition	PF_FREQUENCY_DEFN	EPM Foundation, EPM Setup, Common Definitions, Calendars, Frequency, Frequency Definition	Define calendar frequencies. Add and modify frequency definitions.
Frequency Details	PF_FREQUENCY_DTL	EPM Foundation, EPM Setup, Common Definitions, Calendars, Frequency Details	Define frequency details by assigning the frequencies to calendars.
TimeSpan	TIME_SPAN	EPM Foundation, EPM Setup, Common Definitions, Calendars, TimeSpan	Define timespans.

Understanding Accounting Calendar Setup

In PeopleSoft systems, you can establish an accounting period configuration based on the beginning and ending period dates that you normally use and combine these periods to create accounting calendars. These calendars define the time periods to which you post transactions or create reports for different ledger group and business unit combinations. You select the calendar that defines the periods for a business unit and ledger group combination.

You can maintain an unlimited number of accounting periods over any span of years. You can maintain traditional monthly periods, including an additional adjustment period, or you can define your own periods.

Note: In the PeopleSoft EPM product line, calendars are used mostly for reporting. However, for the system to properly handle data (for example, general ledger data) from your transaction database, you must be sure that the detail calendars match those in your transaction accounting system. You can move calendars from your PeopleSoft transaction database into your PeopleSoft EPM database using PeopleTools such as Data Mover.

The PeopleSoft system supports multiple calendars, so you can keep one calendar for actuals, another for budget and forecast activity, and still others for special reporting needs. Because you store calendars in tables, you can share them across business units, helping to provide consistency in period dates and easing the process of acquisition and consolidation.

PeopleSoft uses the following calendar definition options:

Calendar Builder	Use to create a base calendar from which to create other calendars such as the detail calendar.
Detail Calendar	Define detail calendars that include the number and duration of accounting periods in your fiscal year and the beginning and ending dates for each period. These calendars also identify the adjustment periods for the calendar.
Summary Calendar	Use summary calendars to group detail calendar periods for inquiries and reporting, such as for quarterly reports and semiannual reviews. In this way, your financial information is always ready to be summarized into the timespans that you use most frequently.
(Optional) Business Calendar	Use to create the business or working calendar that identifies holidays and non work days.
(Optional) Daily Calendar	Use for reporting purposes.
(Optional) Budget Period Calendar	Use to manually define fiscal and non fiscal detail budget period calendars.
Calendar Frequencies	Use to define frequency relationships among multiple calendars. You can use frequencies to create multiple relationships among calendars or assign a frequency designation to a calendar. For example, you might have a calendar with a quarterly frequency. This quarterly frequency might be based on the monthly detail calendar.
Timespans	Timespans control the number of periods for which data can be extracted from the ledger table. They enable you to easily select and retrieve information for use in allocations and inquiries. Many timespans are expressed relative to the current period, so that they automatically adapt the content of a report to the date it is produced. Others are defined for specific periods. While many commonly used timespans are included in your PeopleSoft system, you can define additional timespans on the TimeSpan page as necessary.

Common Elements Used to Define Accounting Calendars

Periods Per FY (periods per fiscal year)	Enter the number of periods in an accounting year for a calendar.
Description	Enter a description. The description appears on prompt lists, inquiries, and reports.
Descr (description)	Enter a detailed description of the calendar.

Calendar Builder Page

Use the Calendar Builder page (CALENDAR_BUILDER) to define a base calendar to be used by other calendars that you create. The calendar that you create is by default a detail calendar.

Navigation

EPM Foundation, EPM Setup, Common Definitions, Calendars, Calendar Builder

Image: Calendar Builder page

This example illustrates the fields and controls on the Calendar Builder page. You can find definitions for the fields and controls later on this page.

Calendar Builder

SetID: SHARE Calendar: 01 Periods Per FY: 12

*Description: Monthly

*Begin Date: 01/01/2010 *End Date: *Start Fiscal Year: 2010

Descr:

Calendar Method

Daily
 Weekly
 Bi-weekly
 Monthly
 Bi-monthly
 Quarterly
 Semi-Annual
 Yearly

Monthly Allocation Type: 12 period Calendar **Generate**

Year	Period	Begin Date	End Date	*Period Name	*Abbreviation
2009	1	01/01/2009	01/31/2009	January	Jan
2009	2	02/01/2009	02/28/2009	February	Feb
2009	3	03/01/2009	03/31/2009	March	Mar
2009	4	04/01/2009	04/30/2009	April	Apr
2009	5	05/01/2009	05/31/2009	May	May
2009	6	06/01/2009	06/30/2009	June	Jun
2009	7	07/01/2009	07/31/2009	July	Jul
2009	8	08/01/2009	08/31/2009	August	Aug
2009	9	09/01/2009	09/30/2009	September	Sep
2009	10	10/01/2009	10/31/2009	October	Oct
2009	11	11/01/2009	11/30/2009	November	Nov
2009	12	12/01/2009	12/31/2009	December	Dec

Begin Date

Enter the beginning date for the calendar. The system uses this date and the end date to create the range of periods for the calendar.

End Date

Enter the end date for the calendar. The system uses this date with the begin date to create the range of periods for the calendar.

Start Fiscal Year

The system enters a default year for this field based on the year that you enter in the begin date field.

Calendar Method

Select one of the following options to determine the number of periods in the calendar: daily, weekly, bi-weekly, monthly, bi-monthly, quarterly, semi-annual, or yearly. For example, a daily calendar has 365 periods and a quarterly will have 4 periods.

The system updates to display the number of periods in the Periods Per FY (periods per fiscal year) field.

Monthly Allocation Type

Use to select the appropriate period allocations for a monthly calendar. Select from the following: *12 period Calendar*, *13 period Calendar*, *445 Calendar* (4 weeks, 4 weeks, 5 weeks), *454 Calendar* (4 weeks, 5 weeks, 4 weeks), or *544 Calendar* (5 weeks, 4 weeks, 4 weeks).

Generate

Click this option to have the system generate the calendar. The generated calendar periods display begin and end dates in the grid at the bottom of the page.

Period Name and Abbreviation

Use these fields to change the generated calendar period name and abbreviation if applicable.

Note: As you define calendars, keep in mind that in all instances in which a PeopleSoft EPM warehouse business unit is related to a general ledger business unit, the calendars for the business units should be the same. Warehouse business units are described elsewhere in this topic.

After you have saved a calendar using the calendar builder, you can only modify the calendar description or long description on the Calendar Builder page. If you need to modify the saved calendar further, use the other calendar pages, (for example the Detail Calendar page).

By default, a calendar that you create with the Calendar Builder is a detail calendar.

Detail Calendar Page

Use the Detail Calendar page (DETAIL_CALENDAR1) to modify or add detail accounting calendars that match or differ from your general ledger calendars.

Navigation

EPM Foundation, EPM Setup, Common Definitions, Calendars, Detail Calendar

Image: Detail Calendar page

This example illustrates the fields and controls on the Detail Calendar page. You can find definitions for the fields and controls later on this page.

Detail Calendar

SetID: SHARE Calendar: 01

*Description:

*Periods Per FY:

Descr:

End Date Default

Month Bi-Month

Quarter Semi-Annual

Year Days

Detail Periods							Customize Find View All First 1-12 of 12 Last	
*Year	*Period	*Begin Date	*End Date	*Period Name	*Abbreviation			
2009	1	01/01/2009	01/31/2009	January	Jan	+	-	
2009	2	02/01/2009	02/28/2009	February	Feb	+	-	
2009	3	03/01/2009	03/31/2009	March	Mar	+	-	
2009	4	04/01/2009	04/30/2009	April	Apr	+	-	
2009	5	05/01/2009	05/31/2009	May	May	+	-	
2009	6	06/01/2009	06/30/2009	June	Jun	+	-	
2009	7	07/01/2009	07/31/2009	July	Jul	+	-	
2009	8	08/01/2009	08/31/2009	August	Aug	+	-	
2009	9	09/01/2009	09/30/2009	September	Sep	+	-	
2009	10	10/01/2009	10/31/2009	October	Oct	+	-	
2009	11	11/01/2009	11/30/2009	November	Nov	+	-	
2009	12	12/01/2009	12/31/2009	December	Dec	+	-	

Adjustment Periods				Customize Find View All First 1 of 1 Last	
Adjustment Period	Period Name	Abbreviation			
998			+	-	

You create detail calendars using the Calendar Builder. Use this page to modify calendars after they have been created.

Periods per FY (periods per fiscal year)

Displays the appropriate number of periods for the calendar. For example, 4 indicates a quarterly calendar.

End Date Default

This setting specifies which periodic intervals the system creates. Values are: month, bimonth (bi-monthly), quarter, semi-annual, year, and days.

If you select days, enter the number of days to include in the calendar.

The system uses the end date default setting to populate the subsequent begin and end date values in the grid at the bottom of the page.

Detail Periods

Displays the detail periods for the calendar, including the begin and end dates, period name, and abbreviation.

Adjustment Periods

Displays any adjustment periods defined for the calendar, including the period name and abbreviation.

When you enter period dates, you can define monthly calendar periods or any fiscal period that matches your accounting calendar (such as weekly or bimonthly) as long as the beginning and ending dates of successive periods don't overlap. Every day of the year must be included in a period; you cannot leave gaps between period dates. Make sure that your detail calendar includes a period for the oldest transaction that you want to enter. After installing your PeopleSoft system, you might want to make this earliest date more restrictive.

You will need to return to this page to enter ensuing years manually. You can enter several years at a time or treat the task as part of your end-of-year system maintenance.

After you define your detail calendar, you can use it to manage open periods for the generation of journals. You can also use it to define the periods that store summarized results in a summary ledger.

Note: The MODEL calendar delivered with your system contains data from 1957 to 2025. You can use the MODEL calendar rather than entering your own data; if you do so, make a copy of MODEL first, and make changes to the copy.

Summary Calendar Page

Use the Summary Calendar page (SUMMARY_CALENDAR) to define a summary calendar to group or combine periods from detail calendars.

Navigation

EPM Foundation, EPM Setup, Common Definitions, Calendars, Summary Calendar, Summary Calendar

Image: Summary Calendar page

This example illustrates the fields and controls on the Summary Calendar page. You can find definitions for the fields and controls later on this page.

Summary Calendar

SetID: SHARE Summary Calendar: 02

*Description: Summary Calendar

*Periods Per FY: 4

Detail Calendar: Weekly

Descr:

Detail Calendar Periods					
*Fiscal Year	*Period	*Period Name	*Abbreviation	*From Period	*To Period
2009	1	First Quarter	FQ		

Consider the following when setting up summary calendars. A quarterly summary calendar, for example, might have four periods, each consisting of three periods from a monthly detail calendar. A summary calendar can also combine cumulative detail calendar accounting periods to create year-to-date balances.

Summary calendars are especially useful for determining how your models will roll up for reporting purposes. As with detail calendars, you can include as many fiscal years as you need in one summary calendar.

Enter a description and periods per fiscal year to specify the number of periods in the accounting year for this calendar.

Detail Calendar Select the ID of the detail calendar on which this calendar is based. Every summary calendar must be based on a detail calendar.

Supply the fiscal year, period, period name, and abbreviation.

From Budget Period and To Budget Period Enter the detail calendar periods to be summarized in each summary period.

Note: You can define as many years on a calendar as necessary. Be sure to include any years that you use to store historical information.

Summary Budget Period Calendar Page

Use the Summary Budget Period Calendar page (SUMMARY_BP_CAL) to define a summary budget period calendar to group or combine periods from detail calendars.

Navigation

EPM Foundation, EPM Setup, Common Definitions, Calendars, Summary BP Calendar, Summary Budget Period Calendar

Image: Summary Budget Period Calendar page

This example illustrates the fields and controls on the Summary Budget Period Calendar page. You can find definitions for the fields and controls later on this page.

Summary calendars are especially useful for determining how your models will roll up for reporting purposes. Enter a description, budget period, and from/to periods to specify the number of budget periods in the accounting year for this calendar.

Business Day Calendar Page

Use the Business Day Calendar page (BUS_CALENDAR) to define a business calendar to specify your normal business week and the holidays observed by your banks.

Navigation

EPM Foundation, EPM Setup, Common Definitions, Calendars, Business Day Calendar

Image: Business Day Calendar page

This example illustrates the fields and controls on the Business Day Calendar page. You can find definitions for the fields and controls later on this page.

Business Day Calendar

SetID: SHARE Calendar: 01 *Description:

Normal Business Days

Monday Tuesday Wednesday Thursday Friday Saturday Sunday

Notes:

Holidays Customize | Find | First 1 of 1 Last

*Year	*Date	*Holiday Name
2009	12/25/2009	Christmas

Normal Business Days

Select the business days of the week to include.

Notes

Enter any notes to further describe this calendar.

In the grid at the bottom of the page enter a row for each holiday on which you know businesses won't operate.

Year and Date

Specify the year and date for any holidays that you include.

Holiday Name

Enter the name of the holiday on which your business does not operate.

In multinational corporations, you accommodate the various locations and different holidays observed by defining as many business calendars as you need for each SetID.

Note: Business calendars determine the number of workdays in each month, for reporting purposes only.

Note: If you use PeopleSoft Asset Liability Management, Funds Transfer Pricing, or Risk-Weighted Capital, you must create a business day calendar to define your holidays.

Daily Calendar Page

Use the Daily Calendar page (DAILY_CALENDAR) to define a daily calendar for reporting purposes.

Navigation

EPM Foundation, EPM Setup, Common Definitions, Calendars, Daily Calendar

Image: Daily Calendar page

This example illustrates the fields and controls on the Daily Calendar page. You can find definitions for the fields and controls later on this page.

The screenshot shows a web form titled "Daily Calendar". At the top, it displays "SetID: SHARE" and "Calendar: 12". Below this, there are several input fields and buttons:

- A field labeled "*Description:" with the value "12" entered. To its right are two buttons: "Create" and "Update".
- A field labeled "*Periods Per FY:" with the value "12" entered.
- A dropdown menu labeled "*Calendar Type:" with "Detail Calendar" selected.
- A large text area labeled "Descr:" with a small icon in the bottom right corner.

Create Click to access the Create Detail Periods secondary page, on which you can enter the fiscal year and start date for creating daily periods.

Update Click to access the Create Detail Periods secondary page, on which you can enter a new fiscal year.

Remember to set up a calendar for each fiscal year. To display the detail periods for existing calendars, click the Update button.

Note: For leap years, you must change the periods per fiscal year value from 365 to 366 to create the last period for the leap year (for example, December 31, 2004).

Budget Calendar Page

Use the Budget Calendar page (CAL_BP_TBL) to define a budget calendar for controlled budget processing.

Navigation

EPM Foundation, EPM Setup, Common Definitions, Calendars, Budget Calendar

Image: Budget Calendar page

This example illustrates the fields and controls on the Budget Calendar page. You can find definitions for the fields and controls later on this page.

Budget Calendar

SetID: Calendar:

*Description:

*Periods Per FY:

Descr:

Budget Period	*Begin Date	*End Date	*Period Name		
<input type="text" value="1999"/>	<input type="text" value="08/01/1999"/> <small>31</small>	<input type="text" value="07/31/2000"/> <small>31</small>	<input type="text" value="Period 1 - 1999-08-01"/>	<input type="button" value="+"/>	<input type="button" value="-"/>
<input type="text" value="2000"/>	<input type="text" value="08/01/2000"/> <small>31</small>	<input type="text" value="07/31/2001"/> <small>31</small>	<input type="text" value="Period 1 - 2000-08-01"/>	<input type="button" value="+"/>	<input type="button" value="-"/>
<input type="text" value="2001"/>	<input type="text" value="08/01/2001"/> <small>31</small>	<input type="text" value="07/31/2002"/> <small>31</small>	<input type="text" value="Period 1 - 2001-08-01"/>	<input type="button" value="+"/>	<input type="button" value="-"/>
<input type="text" value="2002"/>	<input type="text" value="08/01/2002"/> <small>31</small>	<input type="text" value="07/31/2003"/> <small>31</small>	<input type="text" value="Period 1 - 2002-08-01"/>	<input type="button" value="+"/>	<input type="button" value="-"/>
<input type="text" value="2003"/>	<input type="text" value="08/01/2003"/> <small>31</small>	<input type="text" value="07/31/2004"/> <small>31</small>	<input type="text" value="Period 1 - 2003-08-01"/>	<input type="button" value="+"/>	<input type="button" value="-"/>

For budget calendars, you might want to enter a descriptive calendar ID such as *AN* for an annual calendar budget.

Enter the budget periods, begin and end dates, and period names.

Frequency Definition Page

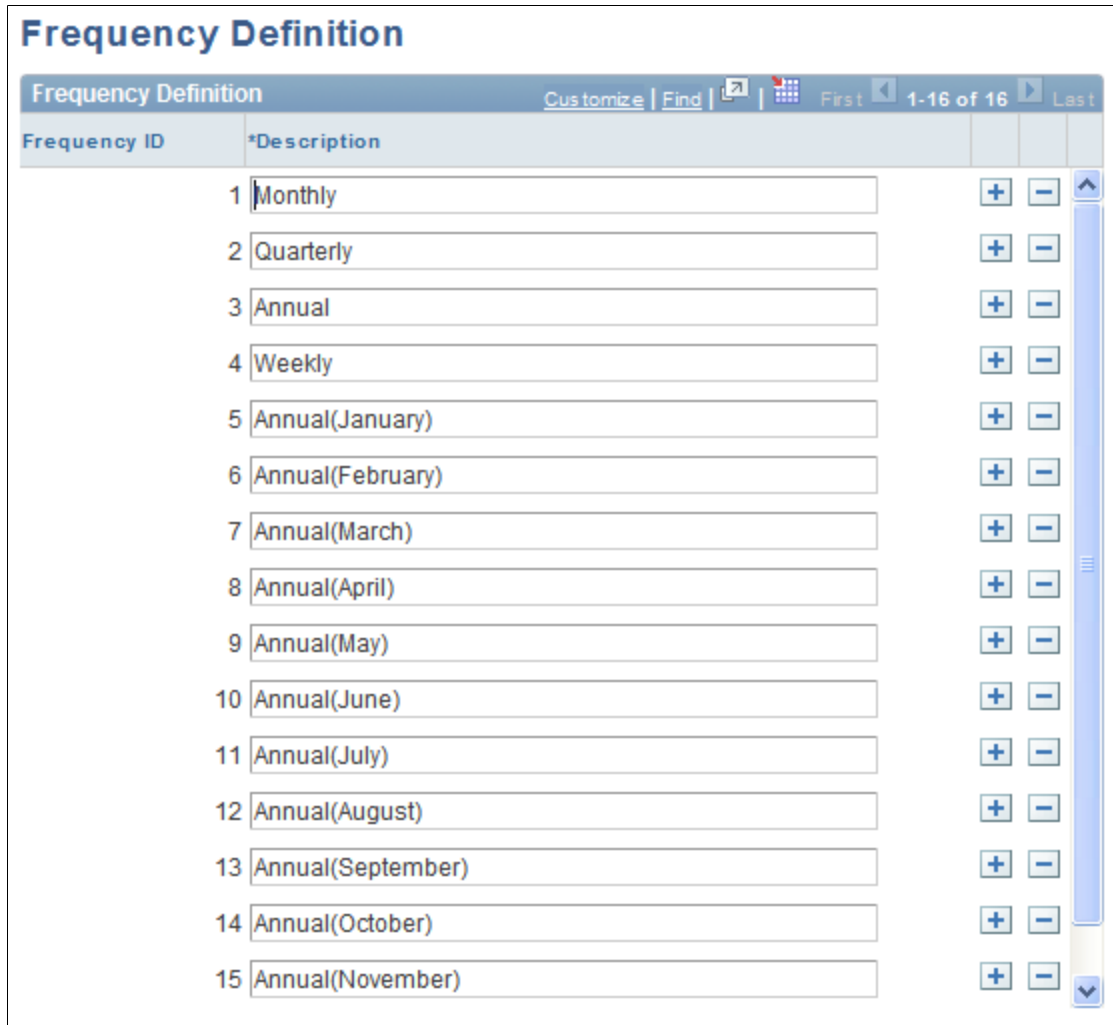
Use the Frequency Definition page (PF_FREQUENCY_DEFN) to define calendar frequencies. Add and modify frequency definitions.

Navigation

EPM Foundation, EPM Setup, Common Definitions, Calendars, Frequency, Frequency Definition

Image: Frequency Definition page

This example illustrates the fields and controls on the Frequency Definition page. You can find definitions for the fields and controls later on this page.



Define frequencies by adding a row with a description or by updating an existing description.

Note: The frequencies you define on this page will comprise the list of valid values in the Frequency Details page.

Frequency Details Page

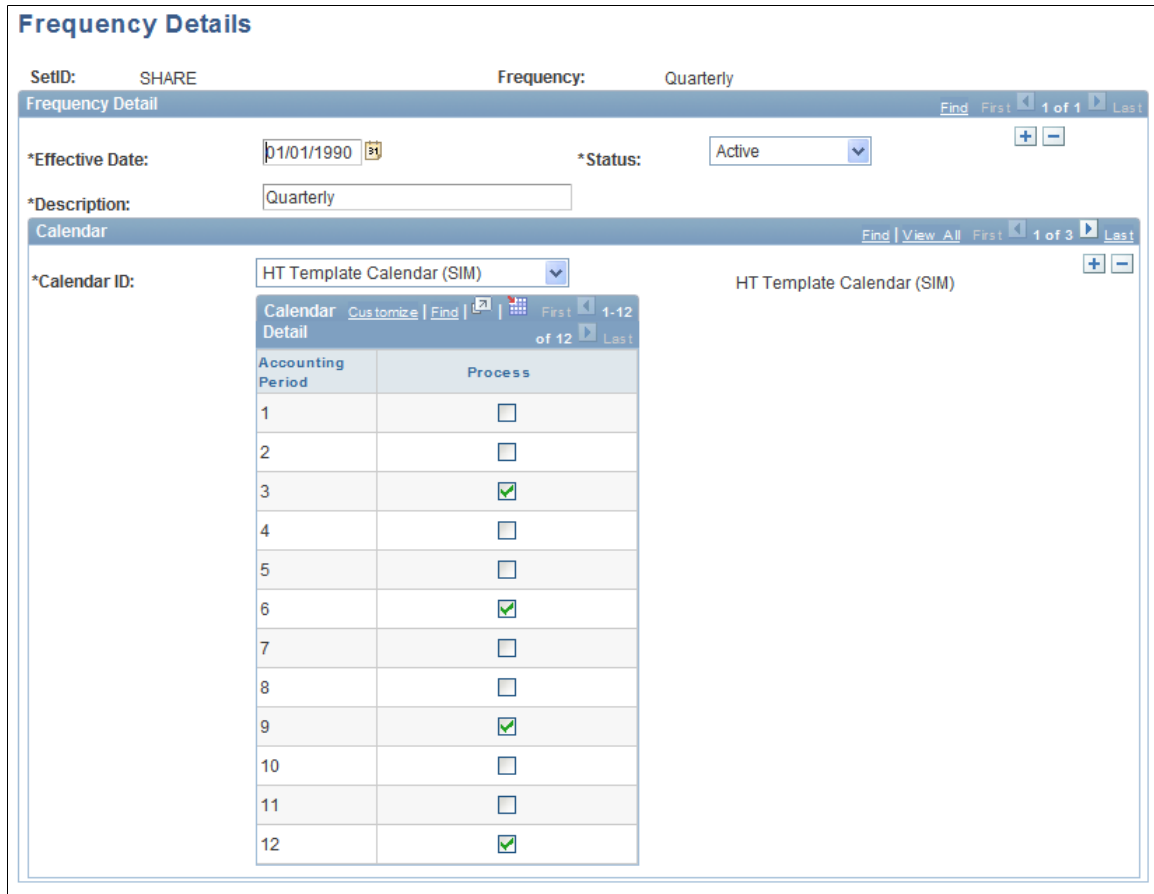
Use the Frequency Details page (PF_FREQUENCY_DTL) to define frequency details by assigning the frequencies to calendars.

Navigation

EPM Foundation, EPM Setup, Common Definitions, Calendars, Frequency Details

Image: Frequency Details page

This example illustrates the fields and controls on the Frequency Details page. You can find definitions for the fields and controls later on this page.



Calendar ID

Select a calendar. The calendar ID that you select represents the calendar that you want to associate with the frequency. You can have more than one frequency defined for a single calendar. The system updates to display calendar detail.

You can also specify all frequencies that are associated with any other frequency definition for this frequency group. You can define frequency relationships across multiple calendars.

Process

Select for any of the accounting periods that you want to process.

Timespan Page

Use the TimeSpan page (TIME_SPAN) to define timespans.

Navigation

EPM Foundation, EPM Setup, Common Definitions, Calendars, TimeSpan

Image: Time Span page

This example illustrates the fields and controls on the Time Span page. You can find definitions for the fields and controls later on this page.

Start Year and Start Period

Enter the appropriate value. The value that you enter depends on the type that you select.

End Year and End Period

Enter the appropriate value. The value that you enter depends on the type that you select.

Type

Select from the following:

Absolute Period or *Absolute Year*: You can enter any year or period that has been defined in the timespans calendar.

Relative to Current Period or *Relative to Current Year*: Enter 0 as the period or year to select the current year or period, -1 to indicate the previous year or period, 1 to indicate the next, and so forth.

Use Scenario Calendar

Select this check box to use the specified scenario calendar with this timespan. This enables you to have a scenario-based timespan.

Include Balance Forward

Select this check box if you use nVision for the consolidation ledger or any ledger with balance forward accounts. If you select this check box, nVision will compute ledger balance.

Calendar ID

Select the calendar that you want to use with this timespan.

Note: Balance forward accounts store balance forward amounts. For example, in your regular accounting system you might want to specify asset, liability, and equity accounts as balance forward accounts, but not revenue or expense accounts. The account balance forward function rolls forward the accounts on any date specified for flexible year-end processing.

See the product documentation for *PeopleSoft EPM: Global Consolidations*

Setting Up the Gregorian Calendar [EPM Warehouses]

The Calendar dimension represents calendars that fall into two broad functional categories. The Gregorian calendar (the calendar that is commonly used) has one row for each day. The business calendar represents the functionality the PeopleSoft provides to define various types of business calendars, such as detail calendars, summary calendars, budget calendars, and so on.

Gregorian calendar setup is a onetime setup as part of the MDW setup. You can generate the Gregorian calendar for a range of years. For each day in the year range, the Gregorian calendar has one row. (You do not define business calendars in the MDW; their definitions are imported from source systems and the OWE).

To set up the Gregorian calendar, use the Day Dimension (D_DAY) component and the Time Dimension (D_TIME) component.

This topic discusses how to set up the Gregorian Calendar for the EPM warehouses.

Pages Used to Set Up the Gregorian Calendar

Page Name	Definition Name	Navigation	Usage
Day Dimension	D_DAY	EPM Foundation, EPM Setup, Common Definitions, Calendars, Gregorian Calendar, Gregorian Calendar Dimension, Day Dimension	Populate the Day dimension.
Time Dimension	D_TIME	EPM Foundation, EPM Setup, Common Definitions, Calendars, Gregorian Calendar, Time Dimension, Time Dimension	Populate the Time dimension.

Day Dimension Page

Use the Day Dimension page (D_DAY) to populate the Day dimension.

Navigation

EPM Foundation, EPM Setup, Common Definitions, Calendars, Gregorian Calendar, Gregorian Calendar Dimension, Day Dimension

Image: Day Dimension page

This example illustrates the fields and controls on the Day Dimension page. You can find definitions for the fields and controls later on this page.

The screenshot shows the 'Day Dimension' configuration page. At the top, there is a header 'Day Dimension'. Below it, the 'Run Control ID' is set to '1'. There are links for 'Report Manager' and 'Process Monitor', and a 'Run' button. The main configuration area includes:

- Start Date**: A text input field with a calendar icon.
- End Date**: A text input field with a calendar icon.
- Jobstream ID**: A text input field containing 'D_DAY'. Below it, a note states '(Default Jobstream is 'D_DAY')'.
- Weekend Flag**: A section titled 'Select days in weekend' containing a table of days with checkboxes:

Weekend Flag	
<input checked="" type="checkbox"/> Sunday	<input type="checkbox"/> Monday
<input type="checkbox"/> Tuesday	<input type="checkbox"/> Wednesday
<input type="checkbox"/> Thursday	<input type="checkbox"/> Friday
<input checked="" type="checkbox"/> Saturday	

Start Date Enter the beginning date for this calendar.

End Date Enter the ending date for this calendar.

Jobstream ID Enter *D_DAY*.

Weekend Flag Check the days that represent weekend days.

Retry Enabled This field will be available for future use.

Save the page and click Run.

Click the Process Monitor link to check the status of application engine program. After a period of time, it should display the status as *Success*.

Time Dimension Page

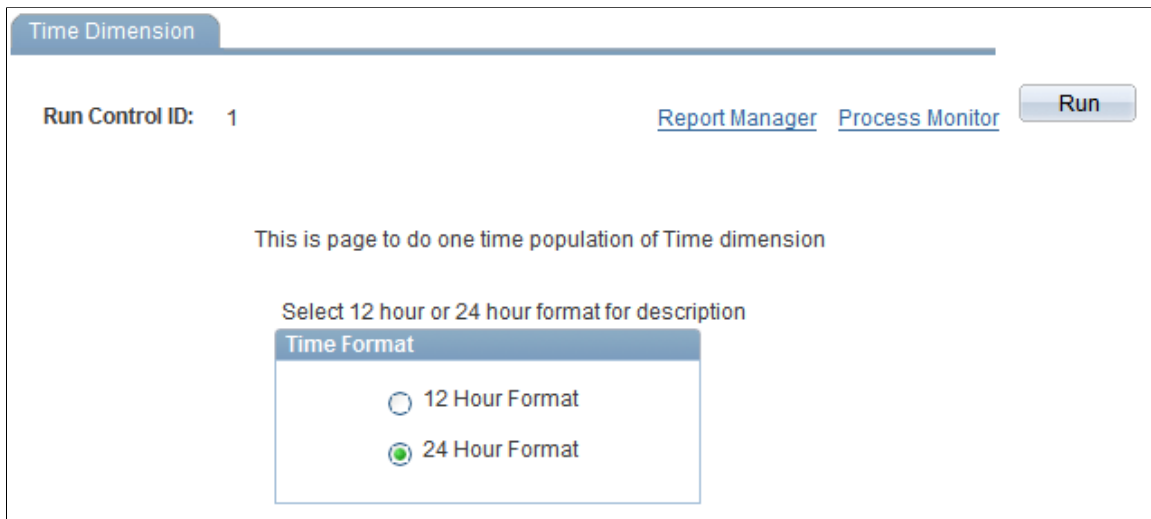
Use the Time Dimension page (*D_TIME*) to populate the Time dimension.

Navigation

EPM Foundation, EPM Setup, Common Definitions, Calendars, Gregorian Calendar, Time Dimension, Time Dimension

Image: Time Dimension page

This example illustrates the fields and controls on the Time Dimension page. You can find definitions for the fields and controls later on this page.



Time Format

Select either 12 Hour Format or 24 Hour Format.

Save the page and click Run.

Click the Process Monitor link to check the status of application engine program. After a period of time, it should display the status as *Success*.

Setting Up Time Zones [EPM Warehouses]

EPM extract, transform, and load (ETL) jobs use the date time stamp in conjunction with source records to perform incremental loads. However, data can be extracted from multiple source transaction systems, each of which might reside in different database servers that use different time zones and different date and time stamps. This can lead to mismatched dates and times between the source and the EPM database. However, EPM target warehouse tables may include a source time zone field for records that have a date-time or time value depending upon the design and requirement.

When a target warehouse table contains a time zone field, prepackaged ETL jobs populate it with your source time zone values. Not all target warehouse tables contain a time zone field and only one source time zone value can be populated in the target warehouse table.

Populating warehouse target tables with source time zone values is the first step in ensuring the synchronicity of source and EPM time zones. However, you are responsible for converting the source time zone value to the EPM database time zone.

Note: You only need provide time zone data if you are using *Enterprise Learning Management* reports.

Defining Units of Measure

Units of measure (UOM) determine how resources are quantified. Each resource must be associated with a standard unit of measure. Standardization helps you to control the units that appear in reports and enables you to use the PeopleSoft automatic conversion features. ETL jobs move unit of measure data from your source transaction system to EPM target warehouse tables. You need only use the Unit of Measure PIA page to redefine or modify your existing unit of measure data.

This topic provides an overview about importing unit of measure values from your source and discusses how to display and modify UOMs.

Page Used to Modify UOMs

<i>Page Name</i>	<i>Definition Name</i>	<i>Navigation</i>	<i>Usage</i>
Unit of Measure	UNITS_TBL1	EPM Foundation, EPM Setup, Common Definitions, Unit of Measure	Display and modify units of measure to determine how resources are quantified.

Understanding how to Import Unit of Measure Values from Your Source

EPM enables you to extract and consolidate source data from various source transaction systems. As such, certain source systems might contain UOM values that are different from those offered in EPM. This causes mismatched UOM values between the source and the EPM database. However, EPM target warehouse tables include source UOM definitions and conversion rate fields for all records that store UOM data. Prepackaged ETL jobs populate these fields with your source UOM definitions and conversion rate values.

PeopleSoft-delivered ETL jobs populate the following target warehouse tables with UOM values:

- UNITS_TBL
- UNITS_CVT_TBL
- INV_ITEM (for Supply Chain Management only)
- UOM_TYPE_INV (for Supply Chain Management only)

Populating warehouse target tables with source UOM values is the first step in ensuring the synchronicity of source and EPM UOM values. However, you are responsible for converting the source UOM value to the EPM database UOM.

Unit of Measure Page

Use the Unit of Measure page (UNITS_TBL1) to display and modify units of measure to determine how resources are quantified.

Navigation

EPM Foundation, EPM Setup, Common Definitions, Unit of Measure

Image: Unit of Measure page

This example illustrates the fields and controls on the Unit of Measure page. You can find definitions for the fields and controls later on this page.

Convert To, Conversion Rate, and Inverse Conversion Rate

Specify the conversion rules of a particular UOM.

For example, by specifying a conversion from cubic feet to cubic yards and an appropriate conversion rate, when you enter a resource amount in cubic feet, the system automatically converts the amount to cubic yards. Automatic conversion is a key feature of PeopleSoft statistical accounting. It enables you to enter resource amounts in whatever unit is readily available or convenient at the time, and converts the units to the required standard. The system also automatically calculates the inverse conversion rate.

Defining and Maintaining Dimensions

This topic provides an overview of dimensions and discusses how to:

- Define dimensions.
- Define dimension details for an Operational Warehouse - Enriched (OWE) warehouse table.
- Define dimension details for a Multidimensional Warehouse (MDW) table.

Pages Used to Set Up Dimensions

Page Name	Definition Name	Navigation	Usage
Dimension	PF_DIM_DEFN	EPM Foundation, Foundation Metadata, Metadata Creation and Editing, Define Dimension	Define dimension metadata.

Page Name	Definition Name	Navigation	Usage
OWE Detail for Dimension	PF_DIM_OWE_DETL	Click the Define link on the Dimension page	Define dimension metadata details for an OWE dimension and apply security parameters.
MDW Detail for Dimension	PF_DIM_MDW_DETL	Click the Define link on the Dimension page	Define dimension metadata details for an MDW dimension and apply security parameters.

Understanding Dimensions

Dimension metadata is associated with specific record metadata and its fields. For example, you can define the dimension DEPARTMENT that references the record DEPARTMENT_TBL and the DEPT_ID column in that record. Dimension metadata captures additional column, key, and security information that is not included with a standard datamap, such as alternate key fields and dimension security. A single dimension can be defined for both an OWE and MDW tables, enabling you to use the same dimension name for both table types.

Dimension metadata is used by the EPM Warehouses, applications, security, and KPI manager.

Dimension Security

Because EPM is delivered with no security restrictions, dimensions are also delivered unsecured. Before you can grant a user access to a dimension you must first indicate to the system that a particular dimension requires securing. Dimensions that are not secured are classified as public, or unsecured. All EPM users can view these objects.

You specify dimensions that require securing using the Dimension page. After you specify a dimension to secure you must associate that dimension with a *security join table* to complete dimension security. Security join tables are EPM database tables that store the security profiles for users along with the corresponding dimension values for which they have access. During security processing, a security join table acts as a lookup. For example, when a user is trying to access a row of data, the SQL that processes this request uses the security join table to identify the user and her access to the particular row of data.

A security join table must be created for every dimension that you plan to secure. Each security join table should match the key structure of the dimension table for which it is defined. Each row in a security join table identifies a user or security role and his access to a specific dimension value. A user who is granted access to multiple values in a single dimension table has several rows in the security join table. In the event that a user has access to an entire dimension, you can insert a single row designated *all* and prevent

the table from ballooning in size with several rows of data. A user that is granted access to multiple dimensions appears in several security join tables.

Image: Account security join table

The following is an example of a security join table.

Record Fields		Record Type					
Num	Field Name	Type	Len	Format	Short Name	Long Name	
1	PF_SY_ROLE_NAME	Char	31	Mixed	EPM Role	EPM Security Role Nam	
2	SETID	Char	5	Upper	SetID	SetID	
3	ACCOUNT	Char	10	Upper	Acct	Account	
4	LASTUPDDTTM	DtTm	26	Scnds	Last Upd DtTm	Last Update Date/Time	

The security join table model is better than a single security output table for two main reasons: Processing smaller tables is more efficient when you are inserting or deleting data, or querying the table to determine access privileges, and modeling individual security tables enable you to be in sync with the anticipated migration to data objects in future releases.

See [Defining Dimension and Metric Security](#).

Dimensions Page

Use the Dimension page (PF_DIM_DEFN) to define dimension metadata.

Navigation

EPM Foundation, Foundation Metadata, Metadata Creation and Editing, Define Dimension

Image: Dimension page

This example illustrates the fields and controls on the Dimension page. You can find definitions for the fields and controls later on this page.

Dimension

Dimension: VENDOR

Dimension Details

***Description:**

Dimension Metadata Customize ?			
	Warehouse	Define	Record Name
<input checked="" type="checkbox"/>	OWE	Define	VENDOR_D00
<input type="checkbox"/>	MDW	Define	

Notes:

This Object Type represents an EPM Vendor. The matching translate value from PF_OBJ_TYPE is "3700". [?](#)

***Owner ID:**

Last Update: 01/01/1990 12:00AM **Updated By:** PplSoft

- Warehouse** Indicates the warehouse layer that is associated with each dimension type.

- Define** Click to access the OWE Detail for Dimension or MDW Detail for Dimension page, depending on which dimension type you are defining.

- Record Name** Displays the name of the record that is associated with a particular dimension.

- Owner ID** Select an owner ID for this dimension.

The owner represents an EPM functional area (such as ABM or Budgeting) that is associated with the dimension. Assign an owner ID to help organize and group the metadata—making locating and auditing easier.

OWE Detail for Dimension Page

Use the OWE Detail for Dimension page (PF_DIM_OWE_DETL) to define dimension metadata details for an OWE dimension and apply security parameters.

Navigation

Click the Define link on the Dimension page

Image: OWE Detail for Dimension page

This example illustrates the fields and controls on the OWE Detail for Dimension page. You can find definitions for the fields and controls later on this page.

The screenshot shows the 'OWE Detail for Dimension' page with the following details:

- Dimension Name:** VENDOR, **Warehouse Name:** OWE
- Record Information Section:**
 - *Record Name: VENDOR_D00 (with a search icon and a link to 'Vendor Header')
 - Leading Key: SetID, Business Unit, None
 - *Business Key: VENDOR_ID (dropdown menu)
 - *Business Key Label: Supplier (dropdown menu)
 - Description: On same Record, Translate Value
 - Description Fieldname: NAME1 (dropdown menu)
- Security Section:**
 - Is Secured
 - Security Join Table: SPPLR_SJT (with a search icon)
- Alternate Fields Section:**
 - Table with columns: Field Name, Description
 - Row 1: Field Name: VENDOR_ID, Description: Vendor Identifier
 - Buttons: +, -
- Buttons: OK, Cancel

Record Information

Record Name Select the record that you want to associate with this dimension.

Leading Key Select a leading key for set processing.

SetID indicates that the leading key is a SetID.

Business Unit indicates that the leading key is a business unit.

None indicates that there is no leading key.

Business Key

Select a business key to serve as the primary key for the selected dimension.

Business Key Label

Select a label for your business key.

Description

Select a description for the business key field.

On Same Record indicates that the description is available in the record. If you select this option, specify a value for the Description Fieldname field.

Translate Value indicates that the description is available through the translate table.

Description Fieldname

Select the field which contains the description of the business key field.

Create Record Metadata

Select to access the Record Metadata page and define new record metadata to associate with this dimension.

If you select an existing record from the prompt, this field displays the name of the record metadata.

Security

Is Secured

Select this check box to indicate whether this dimension is secured.

If you secure this dimension, you must specify its related security join table in the Security Join Table field.

Security Join Table

Select a security join table to associate with this dimension.

The security join table is an EPM table that stores the security profiles for users along with the corresponding dimension values to which they have access. Only Dimensions that are associated with security join tables are presented to a security administrator when security is defined.

Alternate Fields

Field Name

Select an alternate name for your dimension fields.

Alternate field names share the same IDs as the primary field and can be used by different fact tables to join to the same ID.

Specifying alternate field names is optional.

MDW Detail for Dimension Page

Use the MDW Detail for Dimension page (PF_DIM_MDW_DETL) to define dimension metadata details for an MDW dimension and apply security parameters.

Navigation

Click the Define link on the Dimension page

The MDW Detail for Dimension page contains the same fields as the OWE Detail for Dimension page with the addition of the following three fields.

Surrogate Key

Select a surrogate key for this dimension.

MDW dimensions are keyed by surrogate key, and not business unit.

Source ID

Select the source transaction system that is associated with the data in the record.

Prompt View

Enter a key to map the surrogate key to the business key.

This field enables you to select data using meaningful keys instead of surrogate keys, which are generated numbers.

Defining Operator Defaults

Operator defaults enable you to assign a default business unit and SetID for each user per country. You specify operator defaults on the Operator Defaults page.

Page Used to Define Operator Defaults

<i>Page Name</i>	<i>Definition Name</i>	<i>Navigation</i>	<i>Usage</i>
Operator Defaults	OPR_DEF_TABLE_FS1	EPM Foundation, EPM Setup, Installation Analysis and Options, Personal Defaults, Operator Defaults	Specify default user preferences for PeopleSoft users.

Operator Defaults Page

Use the Operator Defaults page (OPR_DEF_TABLE_FS1) to specify default user preferences for PeopleSoft users.

Navigation

EPM Foundation, EPM Setup, Installation Analysis and Options, Personal Defaults, Operator Defaults

Image: Operator Defaults page

This example illustrates the fields and controls on the Operator Defaults page. You can find definitions for the fields and controls later on this page.

The screenshot shows the 'Operator Defaults' page with the following fields and values:

- User ID:** VP1
- Name:** Vice President of Corporate planning
- Business Unit:** CORP1
- SetID:** SHARE
- As of Date:** 01/01/2000
- Localization Country:** USA United States

Business Unit and SetID

The values that you select become the user's default values in the business unit and SetID search fields.

Localization Country

Enter the default country for this user ID.

Note: You can control the business unit, SetID, and address format default values by using the Operator Defaults page.

Archiving EPM Data

This topic provides an overview of EPM data archive and discusses how to define a warehouse archive project.

Page Used to Archive EPM Data

<i>Page Name</i>	<i>Definition Name</i>	<i>Navigation</i>	<i>Usage</i>
Warehouse Archive Project	PF_ARCH	EPM Foundation, Foundation Metadata, Other Metadata Operations, Build Archive Project, Warehouse Archive Project	Archive your EPM data.

Understanding EPM Data Archiving

You can use the warehouse archive project functionality to archive your OWE or MDW data. The Warehouse Archive Project links to the PeopleTools archive tool when you have defined your selection criteria. The archived data is deleted from the database and saved to a flat file.

Warehouse Archive Project Page

Use the Warehouse Archive Project page (PF_ARCH) to archive your EPM data.

Navigation

EPM Foundation, Foundation Metadata, Other Metadata Operations, Build Archive Project, Warehouse Archive Project

Image: Warehouse Archive Project page

This example illustrates the fields and controls on the Warehouse Archive Project page. You can find definitions for the fields and controls later on this page.

Warehouse Component

Select the component to archive: Operational Warehouse - Enrich (OWE) or Multi-Dimensional Warehouse (MDW).

Archive ID

Enter an archive ID for your project.

Archive records on or before

Select the date. Only records with a date equal to or before the date that is specified will be archived.

Build Archive Project

Click this button to build the archive project.

The system accesses the PeopleTools, Archive Data component. You will need to manage the rest of the archival process from the PeopleTools pages. For information about this component, see your PeopleSoft PeopleTools documentation.

Note: Make sure the Archive to Flat File check box is selected on the PeopleTools Record Criteria page (ARCH_PROJ).

Setting Up Chart Viewing and Printing Results [Analytical Applications]

Some analytical applications use charts. To view these charts, the AVS charting tool must be installed. In addition, complete the following procedure to set up required options:

1. Navigate to EPM Foundation, EPM Setup, Installation Analysis & Options, Installation Options, Web Services.

2. Select the Chart Server check box.

This enables you to use the basic charting features.

3. To enable applets, select the Applets check box.

Note that applets download code to the client workstation.

You can also print any of the pages that display results. To produce an optimal printout, set your internet browser to print background colors and images.

In Internet Explorer, follow these steps to set this option:

1. Select Tools, Internet Options.
2. Select the Advanced tab.
3. Scroll through the list of settings to view the Printing options.
4. Select the Print background colors and images check box, and click OK.

Setting Up Ledger-Based nVision Reporting for EPM

This topic discusses how to set up ledger-based nVision reporting over EPM ledger tables.

Pages Used to Set Up nVision Ledger-Based Reporting in EPM

<i>Page Name</i>	<i>Definition Name</i>	<i>Navigation</i>	<i>Usage</i>
Ledger Template	NVS_LED_TMPLT	EPM Foundation, EPM Setup, Ledger Setup, nVision Ledgers, Ledger Template, Ledger Template	Set up ledger templates for your views.
Detail Ledger	NVS_LEDGER_DETAIL	EPM Foundation, EPM Setup, Ledger Setup, nVision Ledgers, Detail Ledger, Detail Ledger	Create detail ledgers for your scenarios.
Group Definition	NVS_LEDGER_GROUP	EPM Foundation, EPM Setup, Ledger Setup, nVision Ledgers, Ledger Group, Definition	Create a ledger group for one or more detail ledgers and specify base currency for those ledgers.
Ledgers for a Unit Definition	NVS_BU_LED1	EPM Foundation, EPM Setup, Ledger Setup, nVision Ledgers, Ledgers for a Unit, Definition	Associate a calendar ID with a ledger group.

Understanding Ledger-Based Reporting and Setup in EPM

nVision is a reporting tool that can retrieve information from a PeopleSoft database and place that data into a Microsoft Excel spreadsheet for further analysis. nVision selects data from a PeopleSoft database

using a *query* or a *ledger*. nVision query-based reporting uses PS query to extract information and can be used with the EPM database. However, nVision ledger-based reporting must be modified to extract data from the EPM database. This is due to the manner in which ledger-based reporting extracts data and the unique structure of EPM ledger tables.

nVision ledger-based reporting uses the LEDGER key to distinguish sets of ledger data and identify which rows should be extracted from ledger tables. However, the LEDGER key does not exist in the EPM ledger tables (such as PF_LEDGER_F00). Instead, EPM ledger tables use the key, PF_SCENARIO_ID, to distinguish sets of data. Consequently, the ledger-based reporting process must be modified to accommodate the PF_SCENARIO_ID used by EPM ledger tables.

To incorporate the PF_SCENARIO_ID key into the nVision ledger-based reporting process, you must create a view of your EPM ledger tables and map the PF_SCENARIO_ID column to the LEDGER column in the view. EPM delivers a view of the PF_LEDGER_F00 table called PF_LEDGER_VW, so you do not need to create one. You only need to create additional ledger views for those ledger tables associated with EPM analytical applications (for example, GC_LEDGER_F00 for Global Consolidations). For more information about application ledger tables that require a view, see the documentation for your PeopleSoft EPM analytical application.

In addition to mapping keys with a ledger view, you must also set up related nVision ledger pages in PIA. The EPM nVision PIA pages populate tables that are used during the ledger-based reporting process (for example, PF_LED_DEFN). When nVision processes a request, it looks for the SET_ID and LEDGER keys located in the PF_LED_DEFN table. The PF_LED_DEFN table is tied to the PF_LED_TEMPLATE table via the LEDGER key, and that connection is used to determine the related data associated with a particular ledger view and PF_SCENARIO_ID. Most of this information is defined in the EPM nVision PIA pages, which are described in the subsequent topics.

See *PeopleSoft PeopleTools: PS/nVision*

See *PeopleSoft PeopleTools: PeopleSoft Application Designer Developers Guide*

Ledger-Based Reporting Setup

The following steps are required to implement ledger-based nVision reporting for EPM:

1. Create a ledger template for each ledger view in your database.

The Ledger Template page populates the LED_TMPLT_TBL table, which stores the ledger view information, used to tie the LEDGER key to the PF_SCENARIO_ID key.

2. Create a detail ledger definition for each scenario.

The Detail Ledger page populates the LED_DEFN_TBL table, which stores the LEDGER and SET_ID keys, used for lookup by nVision reporting.

3. Create a ledger group for each detail ledger.

The Group Definition page enables you to create a ledger group for one or more detail ledgers and specify base currency for those ledgers.

4. Create a ledger group for each detail ledger.

The Ledgers for a Unit Definition page enables you to associate a calendar ID with a ledger group.

Ledger Template Page

Use the Ledger Template page (NVS_LED_TMPLT) to set up ledger templates for your views.

Navigation

EPM Foundation, EPM Setup, Ledger Setup, nVision Ledgers, Ledger Template, Ledger Template

Image: Ledger Template page

This example illustrates the fields and controls on the Ledger Template page. You can find definitions for the fields and controls later on this page.

The screenshot shows the 'Ledger Template' page with the following fields and values:

- Ledger Template:** TEMPLATE01
- *Description:** Ledger Template 01
- *Record (Table) Name:** DEPARTMENT_TBL
- *Posted Total Amount:** BUDGET_DEPTID
- Secured Rept VW:** (empty field)

Record (Table) Name

Select a view to associate with this template.

Note: The record name you select must match the view defined for the EPM ledger record.

Posted Total Amount

Select the name of the record column that holds your posted total amount value.

Secured Rept VW (Secured Report View)

Enter the name of an alternate secured reporting view for the template.

This field enables you to use an alternate view for reporting and overrides the view specified in the Record (Table) Name field.

You can create a ledger template using a specific record, but allow the reporting of that record be accomplished through a different view.

Group Definition Page

Use the Group Definition page (NVS_LEDGER_GROUP) to create a ledger group for one or more detail ledgers and specify base currency for those ledgers.

Navigation

EPM Foundation, EPM Setup, Ledger Setup, nVision Ledgers, Ledger Group, Definition

Image: Ledger Group Definition page

This example illustrates the fields and controls on the Ledger Group Definition page. You can find definitions for the fields and controls later on this page.

The screenshot shows the 'Ledger Group Definition' page. At the top, there is a 'Definition' section with the following fields: SetID (SHARE), Ledger Group (BUDGETS), and Description (Budgets). Below this is a 'Ledger Template' field with a search icon. The 'Ledger Details' section is a table with the following columns: Ledger, Primary, and Base Currency. The table contains one row with the following values: Ledger (empty), Primary (checked), and Base Currency (USD). The table has a toolbar with 'Customize', 'Find', 'First', '1 of 1', and 'Last' buttons.

Ledger Template Select a ledger template to associate with the ledger group.

Ledger Details

Ledger Select a ledger scenario to associate with the ledger group.

Primary Select this check box to indicate the ledger scenario is the primary ledger for this ledger group.

Note: You must specify at least one primary ledger record.

Base Currency Select the base currency for the selected ledger record.

Ledgers for a Unit Definition Page

Use the Ledgers for a Unit Definition page (NVS_BU_LED1) to associate a calendar ID with a ledger group.

Navigation

EPM Foundation, EPM Setup, Ledger Setup, nVision Ledgers, Ledgers for a Unit, Definition

Image: Ledgers for a Unit Definition page

This example illustrates the fields and controls on the Ledgers for a Unit Definition page. You can find definitions for the fields and controls later on this page.

The screenshot shows a web interface for defining ledgers for a business unit. At the top, there is a tab labeled 'Definition'. Below the tab, the 'Business Unit' is set to '00001'. The main section is titled 'BU Ledger Groups' and contains a table with the following data:

	Ledger Group	*Calendar ID		
1	AA	R	+	-
2	BA	R	+	-

The Ledgers for a Unit Definition page enables you to associate a calendar ID with a ledger group.

Note: The calendar ID should match the calendar ID that is specified on the Warehouse Business Unit Scenario Definition page.

Chapter 5

Setting Up Currency Rules for EPM

Setting Up Currency Rules for EPM

PeopleSoft enables you to manage financial information in multiple currencies. PeopleSoft provides specific input, processing, and reporting features that support the European Common Currency (euro), currency conversions, remeasurement, revaluation, translation, and a complete audit trail of all multi-currency processing.

You can define and maintain tables that describe currency codes, exchange rates, market rates, and currency rate types. All PeopleSoft products use the same market rate and currency pages and tables, which enables you to administer centralized currency controls throughout the integrated product lines.

These topics provide an overview of multiple currency processing concepts and discuss how to:

- Set up EPM currency tables
- Set up market rates
- Define currency quotations
- Establish market rates
- Calculate currency rates
- Configure currency precision

Note: The multi-currency processing setup tasks documented in this topic are common to both the EPM Warehouses and the Analytical Applications. However, additional multi-currency processing setup tasks are required for the EPM Warehouses and [Setting Up and Running Currency Conversion](#) the Analytical Applications:

EPM Warehouses: see the topic entitled 'Implementing Currency Conversion' in your specific EPM warehouse documentation (for example, the *PeopleSoft Campus Solutions Warehouse*).

Analytical Applications: see [Setting Up and Running Currency Conversion](#) in this documentation.

Understanding EPM Multiple Currency Processing Concepts

Before you begin to process multiple currencies, you should understand the concepts behind multi-currency processing in EPM.

Currency and Calculation Types

PeopleSoft software uses terminology associated with currency that is consistent with generally accepted accounting principles and the Financial Accounting Standards Board (FASB) accounting standards.

Currency refers to the denomination of a monetary transaction. PeopleSoft applications use a currency code (CURRENCY_CD) to identify and track individual currencies. Although the system does not require it, you should use International Standards Organization (ISO) currency codes. PeopleSoft applications have no limit on the number of currencies that you can use.

Important currency terms are:

Base currency	Primary currency in use for a business unit. Each business unit must have one base currency. This is usually the local currency for the organization, but accounting rules or other circumstances might dictate that it be different. In PeopleSoft EPM, you assign a base currency to each warehouse business unit.
European Common Currency (euro)	Common currency adopted by participating European countries (effective January 1, 1999). PeopleSoft applications enable you to use <i>triangulation</i> to convert currencies that newly participate in but are not yet replaced by the euro.
Foreign currency	Any currency that a business unit uses other than its base currency for doing business is referred to as a foreign currency. Some foreign currencies are used for reporting only, some are for input only, but most are available for both input and reporting.
Functional currency	Defined in FASB 52 as "currency of the primary economic environment in which a foreign entity operates." It is not an interchangeable term for base currency. When the functional currency differs from the base currency, FASB 52 requires an additional translation (called <i>remeasurement</i>) from base to functional currency.
Reporting currency	Used for financial reports such as consolidated financial statements.

Important currency calculation types are:

Exchange rate	Expresses the value of one currency in terms of another. Recognized types of exchange rates include <i>spot</i> (immediate), <i>current</i> , <i>negotiated</i> (discount and premium forward rates), <i>average</i> , and <i>historical</i> rates. PeopleSoft applications support any number of exchange rate types.
Conversion	Exchange of one currency for another currency. In PeopleSoft applications, <i>conversion</i> refers to the expression of the value of foreign currency transactions in the base currency.
Market rate	Encompasses a number of different rate types including exchange rates, interest rates, stock exchange indexes, economic indicators, and commodity prices.
Remeasurement	Changing the unit of measure from the base currency of a business unit to its functional currency. This is required

whenever a business unit's books are maintained in a currency other than the functional currency.

Translation

Expressing ledger balances in terms of another currency, such as when balances maintained in the base or functional currency are restated in a different reporting currency. In the case of translation, gains and losses are recognized solely from fluctuations in the exchange rate.

Currency Precision

Currency dictates the *precision* of monetary amounts. For example, United States dollar amounts have two digits to the right of the decimal and Japanese yen have none. The system addresses currency precision as follows:

- PeopleSoft software provides currency-sensitive amount fields with a standard length of 23.3, or 23 digits to the left of the decimal point and 3 digits to the right of the decimal point.
- By default, the system rounds all currency-sensitive amount fields to the currency precision of the associated currency. This action is a PeopleSoft PeopleTools option that you can deactivate.
- All numbers on SQR reports are currency-sensitive. For reporting with PS/nVision, the display is equal to the field precision, but you can increase the number of decimal places.

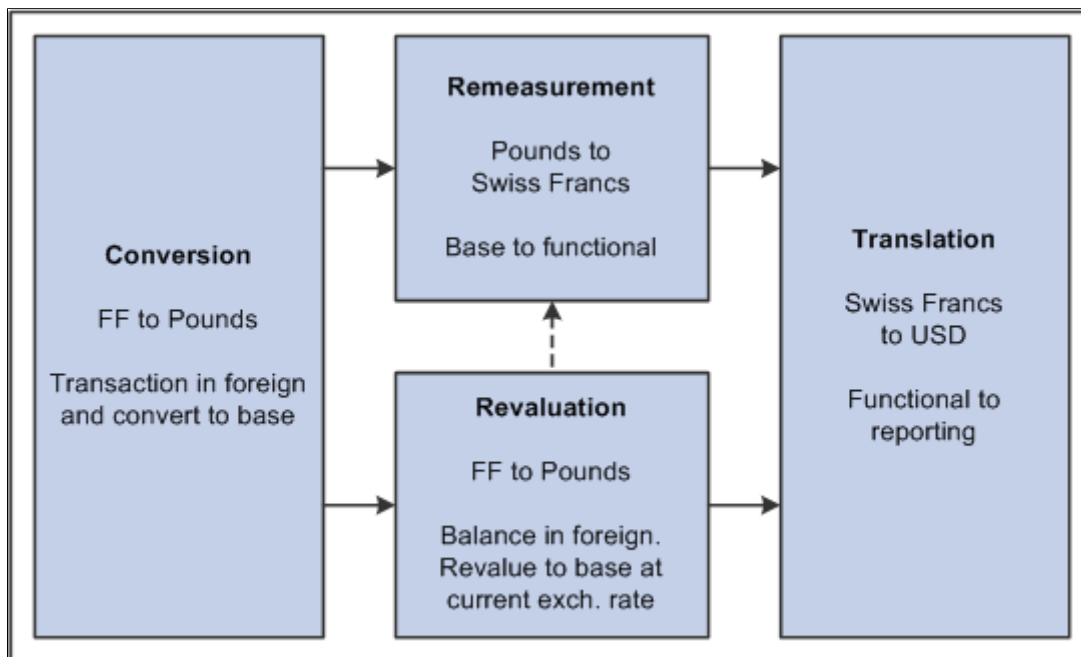
Revaluation

When you adjust the base currency value of balance sheet accounts that are maintained in a foreign currency, this is called *revaluation*. You generally perform revaluations at the end of each accounting period to reflect the actual base currency value of assets and liabilities as exchange rates fluctuate between the base and foreign currencies. You make adjusting entries to the accounts that are being revalued with

an offsetting entry to a revaluation gain or loss account. The gain or loss account is sometimes referred to as an unrealized exchanged gain or loss.

Image: Revaluation

This diagram illustrates the revaluation process.



In this example, a London-based subsidiary of a Swiss company records a purchase made in Mexican pesos. The Swiss company is owned by a United States corporation. The following table correlates the terminology and the currencies:

Foreign currency	MXN (Mexican Peso)
Base currency	GBP (British pounds)
Functional currency	CHF (Swiss francs)
Reporting currency	USD (United States dollars)

Currency as a ChartField

You get the best results when you record multi-currency transactions with a currency ChartField. This approach enables you to record multiple currencies in the same ledger and reinforces the concept of a ledger's role as a record for an entire category of information (such as actuals, budgets, forecasts, or commitments).

Differentiating Between the Currency Conversion Process of the EPM Warehouses and the Analytical Applications

Separate currency conversion processes are required in EPM, one for the *EPM Warehouses* and one for the *Analytical Applications*. After performing the basic multi-currency processing setup tasks in this topic, you must perform additional multi-currency processing setup tasks that are specific either to the EPM

Warehouses or the Analytical Applications. It is important to understand the difference between the two currency conversion processes, as they are quite different.

The following table describes the differences between the EPM Warehouse and Analytical Application currency conversion processes:

Subject	Analytical Application Currency Conversion	EPM Warehouse Currency Conversion
Technology Platform	Based on Application Engine (AE) technology for seamless integration with application processing that is also based on AE technology.	Based on ETL technology.
Set Processing	Business unit is required in set processing.	Business unit is optional in set processing. This allows for a single conversion process to convert all transaction amounts for global reporting.
Business/Conversion Rules	Currency conversion involves complex rules for compliance reporting and simulation. The complex rules are stored in various EPM metadata.	Currency conversion is used to convert monetary amounts to a common currency for trend analysis. Trend analysis requires a simple currency conversion rule based on an exchange date and rate type that does not require extensive rule setup.

The remaining currency conversion setup tasks required for the EPM Warehouses can be found in your specific EPM warehouse documentation (for example, *PeopleSoft Campus Solutions Warehouse*).

The remaining currency conversion setup tasks required for the Analytical Applications can be found in this documentation.

See [Setting Up and Running Currency Conversion](#).

Setting Up EPM Currency Tables

Currency code pages define each currency that you use. To meet your multicurrency requirements, PeopleSoft supports the euro and delivers the Currency Code table with many common ISO standard currencies. The table also supports the ISO standard of zero, two, and three decimal positions.

PeopleSoft-delivered ETL jobs move currency code data from your source transaction system to EPM target warehouse tables. You need only use the currency code PIA pages to redefine or modify your existing currency code data.

This topic discusses how to:

- Maintain currency codes.
- Update the status of a euro currency code.

Currency Code Page

Use the Currency Code page (CURRENCY_CD_TABLE) to maintain currency codes and the currency precision by currency.

Navigation

EPM Foundation, EPM Setup, Common Definitions, Currencies and Rates, Currency Codes

Image: Currency Code page

This example illustrates the fields and controls on the Currency Code page. You can find definitions for the fields and controls later on this page.

Currency Code

Currency Code: ESP

Definition Find | View All First 1 of 1 Last

*Effective Date: 01/01/1900 *Status: Active

*Description: Spanish Peseta

Short Description: Peseta

Currency Symbol: Pts

Country: ESP Spain

Decimal Positions: 0

Scale Positions:

Currency Symbol

PeopleSoft delivers many currencies with a currency symbol such as \$ for Australian dollar (AUD) or ? for British pound (GBP). You can enter new symbols for delivered currencies or for currencies that you might add.

Country

Displays the code for the country from which the currency originates.

Decimal Positions

Enter the number of decimal positions that should appear in the notation for the currency. For example, use two decimal positions for Australian dollars (5.00 AUD), but no decimal positions for Japanese yen (500 JPY).

Scale Positions

Enter the scale positions that you want to round for this currency. Scale positions control how many numbers appear to the left of the decimal when displayed. The data is stored with full precision in the database. For example, if you want all dollar amounts in the millions displayed as the number of millions without the zeros, enter 6 as your scale position. In this case, 24,

000,000 is displayed as 24, but is stored in the database as 24,000,000.

Note: The data on this page is stored in the Currency Code table. The values on this table are effective-dated. The software is shipped with the Currency Code table in compliance with ISO standards for decimal positions. You can increase the number of decimals to a maximum of three.

Note: PeopleSoft updates the Currency Code table and the fully populated country, state, and province code tables as national boundaries and designations change.

Related Links

[Configuring Currency Precision for Currency Conversion](#)

Currency Codes (Euro) Page

Use the Currency Codes (Euro) page (CURRENCY_EURO) to update the status of a euro currency code.

Navigation

EPM Foundation, EPM Setup, Common Definitions, Currencies and Rates, Currency Code (Euro)

Image: Currency Code (Euro) page

This example illustrates the fields and controls on the Currency Code (Euro) page. You can find definitions for the fields and controls later on this page.

The screenshot shows the 'Currency Code (Euro)' page with the following fields and values:

Currency Code:	ESP
*Status:	Active
Description:	Spanish Peseta
Short Description:	Peseta

Status Displays whether a currency is an *active* participant in the euro.

Note: Do not attempt to modify the currency quotation methods for currencies that are linked to the euro.

Setting Up Market Rates for EPM Currency Conversion

The PeopleSoft approach to market rates and currency conversion is driven by the need to accommodate business practices related to the European Common Currency (euro). In addition to currency exchange rates, PeopleSoft supports the many different types of global market rates, such as interest rates, stock exchange indexes, and economic indicators.

PeopleSoft-delivered ETL jobs move market rate data from your source transaction system to EPM target warehouse tables. You need only use the market rate PIA pages to redefine or modify your existing market rate data.

This topic discusses how to:

- Define market rate indexes.
- Define market rate types.
- Define market rate tolerances.

Pages Used to Manage Market Rates

<i>Page Name</i>	<i>Definition Name</i>	<i>Navigation</i>	<i>Usage</i>
Market Rate Index	RT_INDEX_TBL	EPM Foundation, EPM Setup, Common Definitions, Currencies and Rates, Market Rate Index	Describe the indices for which you track rates. Typical market rate indexes include LIBOR, Bloomberg foreign exchange, and Reuters foreign exchange. An index categorizes the various market rates that you track.
Market Rate Type	RT_TYPE_TBL	EPM Foundation, EPM Setup, Common Definitions, Currencies and Rates, Market Rate Type	Define market rate types. Rate types include commercial, floating, average, and historical.
Market Rate Definition	RT_RATE_DEF_TBL	EPM Foundation, EPM Setup, Common Definitions, Currencies and Rates, Market Rate Definition,	Define tolerance limits for rates and determine what action should occur if a new rate occurs outside of the tolerance limit. The fields on this page differ according to the rate category of the market rate index.

Defining Market Rate Indexes

Use the Market Rate Index page (RT_INDEX_TBL) to describe the indices for which you track rates.

Typical market rate indexes include LIBOR, Bloomberg foreign exchange, and Reuters foreign exchange. An index categorizes the various market rates that you track.

Navigation

EPM Foundation, EPM Setup, Common Definitions, Currencies and Rates, Market Rate Index

Image: Market Rate Index page

This example illustrates the fields and controls on the Market Rate Index page. You can find definitions for the fields and controls later on this page.

The screenshot shows a web form titled "Market Rate Index". The form contains the following fields and controls:

- Index:** A text field containing the value "MODEL".
- *Rate Category:** A dropdown menu with "Exchange Rate" selected.
- *Description:** A text field containing the value "Default".
- Default Exchange Rate Index:** A checkbox that is checked.

Rate Category

Select a Rate Category from the list: Options include: *Commodity Price, Economic Indicator, Exchange Rate, Futures Price, Interest Rate, Other, Stock Exchange, Index, or Stock Price.*

Default Exchange Rate Index

If you are entering exchange rate indexes, select this check box to indicate which index should be used to retrieve currency exchange rates. You can specify only one index code as the default.

Defining Market Rate Types

Use the Market Rate Type page (RT_TYPE_TBL) to define market rate types.

Rate types include commercial, floating, average, and historical.

Navigation

EPM Foundation, EPM Setup, Common Definitions, Currencies and Rates, Market Rate Type

Enter a description and short description for each market rate type that you use.

Defining Market Rate Tolerances

Use the Market Rate Definition page (RT_RATE_DEF_TBL) to define tolerance limits for rates and determine what action should occur if a new rate occurs outside of the tolerance limit.

The fields on this page differ according to the rate category of the market rate index.

Navigation

EPM Foundation, EPM Setup, Common Definitions, Currencies and Rates, Market Rate Definition,

Image: Market Rate Definition page

This example illustrates the fields and controls on the Market Rate Definition page. You can find definitions for the fields and controls later on this page.

Rate Definition

Market Rate Definition

Index: MODEL Default

Rate Category: Exchange Rate

From Currency:

Code:

Term	From Currency	To Currency	Maximum Variance	*Error Type	
0	ADP	CAD	2.50	Warning	+ -
0	ADP	USD	2.50	Warning	+ -
0	AED	CAD	2.50	Warning	+ -
0	AED	USD	2.50	Warning	+ -
0	AFA	CAD	2.50	Warning	+ -
0	AFA	USD	2.50	Warning	+ -
0	ALL	CAD	2.50	Warning	+ -
0	ALL	USD	2.50	Warning	+ -

The information that appears on this page depends on the type of index that you select. For example, if you select an index with interest rates, interest rates appear on this page.

Market rate definitions specify the valid term, currency, and other appropriate field combinations for market rates. For example, if you have a market rate definition for an exchange rate with a term of 30, a from currency of *EUR*, and a to currency of *USD*, you can enter a rate using this combination in the market rate table. If you do not have a market rate definition, the system creates one for you using the default values.

From Currency Code

Enter the *from* currency code.

Refresh

Click the Refresh button to limit the display to the specified *from* currency code.

Term

Enter the desired term (expressed in days). A zero term indicates a spot rate.

From Currency

Enter the appropriate *from* currency. This value is used with the *to* currency value as part of an exchange rate pair. When you use triangulation, include a definition for each of the currency pairs involved in the triangulation.

To Currency	Enter the appropriate <i>to</i> currency. This value is used with the <i>from</i> currency value as part of an exchange rate pair.
Maximum Variance	Indicate the percentage of variance that is allowed when the user maintains the market rate. The system generates an error message if the change exceeds the tolerance. The default value is 2.50 (2.5 percent).
Error Type	Enter the type of error processing that should occur if the maximum variance is exceeded. Values are: <p><i>None:</i> No error processing occurs, and the new rate is used even though it exceeds the limit.</p> <p><i>Stop:</i> Processing halts, and the system prevents you from saving the new rate.</p> <p><i>Warning:</i> Default value. A warning appears; you can ignore it and save the new rate.</p>

Defining Currency Quotations for EPM Currency Conversion

This topic provides an overview of currency quotations and discusses how to maintain currency quotation methods.

Page Used to Define and Maintain Currency Quotations

<i>Page Name</i>	<i>Definition Name</i>	<i>Navigation</i>	<i>Usage</i>
Currency Quotation Method	CURR_QUOTE_PNL	EPM Foundation, EPM Setup, Common Definitions, Currencies and Rates, Currency Quotation Method	Maintain a currency quotation method for each <i>from</i> currency and <i>to</i> currency pair.

Understanding Currency Quotations

PeopleSoft supports direct and indirect rate quotation, quote units, and triangulation. These are flexible and accurate tools that enable you to convert and manage multicurrency operations.

The currency quotation method controls how a stored rate is displayed and how an entered rate is interpreted and stored in the database. You maintain a currency quotation method for each *from* currency and *to* currency pair.

You do not typically maintain rates online for currency pairs that triangulate. Instead, the Cross rate/ Triangulation Generation SQR determines the cross rate by using the rates between the *from* currency and the reference currency, and between the reference currency and the *to* currency. Currency quotation methods must be set up correctly to yield the desired triangulation results.

PeopleSoft-delivered ETL jobs move currency quotation data from your source transaction system to EPM target warehouse tables. You need only use the Currency Quotation Method PIA page to redefine or modify your existing currency quotation data.

Note: Define currency quotation options before you enter and calculate the rates.

Currency Quotation Method Page

Use the Currency Quotation Method page (CURR_QUOTE_PNL) to maintain a currency quotation method for each from currency and to currency pair.

Navigation

EPM Foundation, EPM Setup, Common Definitions, Currencies and Rates, Currency Quotation Method

Image: Currency Quotation Method page

This example illustrates the fields and controls on the Currency Quotation Method page. You can find definitions for the fields and controls later on this page.

The screenshot displays the 'Currency Quotation Method' page. At the top, it identifies the 'From Currency Code' as 'ESP Spanish Peseta' and the 'To Currency Code' as 'EUR euro'. Below this, the 'Quote Method' section shows an 'Effective Date' of '01/01/1997' and a status of 'Active'. The 'Rate Quotation Basis' section is set to 'Indirect' with '*Quote Units' of '1' and 'Auto Reciprocate' checked. The 'Triangulation Options' section includes a 'Triangulate' checkbox, a 'Reference Currency' field, and three sub-sections: 'Primary Visual Rate', 'Cross-Rate', and 'Recalculate', each with radio buttons for 'Not Applicable' and 'Allow Override'.

Rate Quotation Basis

Direct

Determine whether you want the rates for a currency pair quoted directly. For example, in converting United States dollars (USD) to euro (EUR), a direct quote would indicate that USD 1 = EUR x.xxxx. In this case, euros are quoted directly against the United States dollar.

Indirect	Determine whether you want the rates for a currency pair quoted indirectly. For example, in converting United States dollars (USD) to euros (EUR), an indirect quote would indicate that $\text{USD } x.xxxx = \text{EUR } 1$.
Quote Units	Sometimes called "scaling factors," these preserve decimal position. You can enter any value in this field, although quote units generally are on a scale of 10 (such as 10, 100, 1000). The default value for this field is <i>1</i> .
Auto Reciprocate	Select this check box to create or update the rate for the reciprocal currency pair automatically whenever an exchange rate is added or updated. For example, when you enter a new USD to GBP rate, the GBP to USD rate will be updated automatically. You can only apply this feature to currency pairs for which quotation methods have been established.

Note: Currency pairs that triangulate must be classified as either direct or indirect to use in displaying the calculated cross rate. Two fields store the rate conversion factor: RATE_DIV and RATE_MULT. The currency conversion formula is always: $(\text{From currency} \div \text{RATE_DIV}) \times (\text{RATE_MULT}) = \text{To currency}$

Triangulation Options

Select the Triangulate check box to have the system convert two currencies through a third currency (the reference currency). Triangulation is used in hyperinflationary environments in which all conversions to the local currency are done through a more stable currency such as USD.

Note: Triangulation was initially used for European countries participating in the euro. However, since 1999 all countries participating in the euro are quoted directly against the foreign currency. Any countries newly participating in the euro might be initially subject to triangulation, however. The triangulation example below and any other examples in this topic that show triangulation, use a fictional country, with a currency code of NEW, that has just joined the euro. This country is subject to triangulation.

The following are examples of indirect quotation, direct quotation with quote units, and triangulation:

USD 100 to GBP (indirect) = $(\text{USD } 100 \times 1.6) \times 1 = \text{GBP } 62.50$.

CHF 1000 to German marks (DEM) (direct with units) = $(\text{CHF } 1000 / 100) \times 119.335 = \text{DEM } 1193.35$.

USD 100 to NEW (triangulate) = $(\text{USD } 100 / 1.25) \times 6.8 = \text{NEW } 544$.

For example, to convert from USD to NEW with triangulation, you perform two conversions:

- Convert the USD amount to the reference currency using the appropriate triangulated rate.

The triangulated rate uses the USD to EUR component of the USD to NEW triangulated rate that is stored in RATE_DIV.

- Convert the reference currency to NEW using the fixed exchange rate.

The exchange rate uses the EUR to NEW component of the USD to NEW triangulated rate that is stored in RATE_MULT.

Typically, you do not maintain triangulation rates directly. Instead, you process these and all rates through the Cross/Reciprocal Rate Calculator.

Select a reference currency through which the *from* currency will be converted.

You must consider three exchange rates for triangulated currency pairs:

- The rate between the *from* currency code and reference currencies.
- The rate between the reference and the *to* currency code currencies.
- The cross rate between the *from* currency code and *to* currency code currencies.

Primary Visual Rate

Select one of the three conversion rates as the primary rate that appears on primary pages and reports.

Cross Rate

Select the Allow Override check box to enable users to override the cross rate for a triangulated currency pair. If this option is deselected, you can change the components of only the triangulated rate.

Recalculate

If you select the Allow Override check box, you must maintain triangulation accuracy by specifying which currency pair the system should use to recalculate if the cross rate is overridden.

Automatic Reciprocation of Quote Methods

The Currency Quotation Method page automatically provides reciprocal methods. For example, if you define the conversion of USD to NEW as indirect, this record is automatically created to indicate a quote method of direct. If you change the quote method on the NEW to USD record, the USD to NEW record is updated automatically.

Note: This example uses a hypothetical currency NEW that has just begun participating in the euro and is still subject to triangulation for an initial period.

Using the conversion of USD to NEW as an example, this table shows each possible field value and its corresponding reciprocal value.

Field	Value <i>(for USD to NEW)</i>	Reciprocal Value <i>(for NEW to USD)</i>
Quotation Basis	Direct Indirect	Indirect Direct
Quote Units	Any valid value	Same value

Field	Value (for USD to NEW)	Reciprocal Value (for NEW to USD)
Rate Decimal Positions	4 (default value)	Same value
Auto Reciprocate	Yes No	Yes No
Triangulate	Yes No	Yes No
Reference Currency	Any valid value	Same value
Primary Visual Rate	From - To (USD - NEW) From - Ref (USD - EUR) Ref - To (EUR - NEW)	From - To (NEW - USD) Ref - To (EUR - USD) From - Ref (NEW - EUR)
Cross rate Allow Override	Yes No	Yes No
Cross rate Recalculate	From - Ref (USD - EUR) Ref - To (EUR - NEW)	Ref - To (EUR - USD) From - Ref (NEW - EUR)

Establishing Market Rates for EPM Currency Conversion

PeopleSoft-delivered ETL jobs move market rate data from your source transaction system to EPM target warehouse tables. You need only use the market rate PIA pages to redefine or modify your existing market rate data.

This topic discusses how to:

- Define market rates.
- Establish rate definitions.
- Maintain exchange rates.
- Load market rates.

Pages Used to Establish Market Rates for EPM Currency Conversion

<i>Page Name</i>	<i>Definition Name</i>	<i>Navigation</i>	<i>Usage</i>
Market Rate	RT_RATE_PNL	EPM Foundation, EPM Setup, Common Definitions, Currencies and Rates, Market Rates, Market Rate	Define market rates.
Rate Definition	RT_RATE_DEF_SEC	Click the Rate Definition link on the Market Rates page	Establish rate definitions.
Exchange Rate Detail	EXCH_RT_DTL	Click the Exchange Rate Detail button on the Market Rates page. This page can also be accessed from other pages in the system.	Maintain exchange rates.

Market Rate Page

Use the Market Rate page (RT_RATE_PNL) to define market rates.

Navigation

EPM Foundation, EPM Setup, Common Definitions, Currencies and Rates, Market Rates, Market Rate

Image: Market Rates page

This example illustrates the fields and controls on the Market Rates page. You can find definitions for the fields and controls later on this page.

Market Rates

Market Rate

Index:	MODEL	Default	Rate Definition
Rate Category:		Exchange Rate	
Rate Type:	CRRNT	Current Rate	
Term:	0		
From Currency Code:	ESP	Spanish Peseta	
To Currency Code:	EUR	euro	

Rate Find | View All First 1 of 1 Last

Effective Date:	*Rate:	
12/15/2002	166.38600000	+ -

Market rates can be loaded to the RT_RATE_TBL from any external source using the DataStage ETL tool. Use this page to verify that the market rates were loaded correctly.

Rate

The Rate field displays the *visual rate* or, in the case of triangulated exchange rates, the *primary visual rate*. The visual rate is typically the cross rate, but it can also be one of the other component rates of the triangle.

You can edit the rate for non triangulated rates and for triangulated rates if a quotation method has been defined for the currency pair and the Cross rate Allow Override check box is selected on the Currency Quotation Method page. If an override is not allowed, you can update the exchange rate values to and from the reference currency on the Exchange Rate Detail page.

Changing a Triangulated Cross Rate

If you change a triangulated cross rate, the system recalculates one of the component rates. This can result in the cross rate being recalculated in a manner that is slightly different from the one that you entered. For example, you start with a triangulated rate of RM=6.80000000 and RD=1.25000000 for a cross rate of 5.44000000. If you change the cross rate to 5.43550000:

- The system first recalculates RD to 1.25103486.
- The system then recalculates the cross rate to 5.43550001 based on the first recalculation.

In rate maintenance, you must accept the recalculation. However, in a situation such as journal entry, a warning message enables you to override triangulation and to use the exact rate that you entered, which results in the rate being stored as RM equal to 5.43550000 and RD equal to 1.

You can edit the Rate field except when *all* of these conditions are true:

- The rate is triangulated.
- The primary visual rate is the cross rate.
- The Allow Override check box on the Currency Quotation Method page is deselected for the exchange rate's quotation method.

Note: Typically, you do not maintain triangulated exchange rates online. Instead, maintain the rates of the *from* currency to the *reference* currency and the *reference* currency to the *to* currency, and then run the Cross rate Reciprocal SQR (EO9030.SQR) to define the triangulated exchange rates.

Reciprocal Currency Pairs

If a quotation method has been defined for the currency pair and if the Auto Reciprocate check box is selected, creating or maintaining a rate for a currency pair automatically creates or updates the rate of the reciprocal currency pair. For example, if you change the USD to GBP rate, the GBP to USD rate automatically is automatically updated. You can only automatically reciprocate currency pairs for which quotation methods have been defined.

If a rate definition does not already exist for the currency pair, one will be automatically created with the default values of 2.5 percent maximum variance and warning message processing.

Rate Definition Page

Use the Rate Definition page (RT_RATE_DEF_SEC) to establish rate definitions.

Navigation

Click the Rate Definition link on the Market Rates page.

Image: Market Rate - Rate Definition page

This example illustrates the fields and controls on the Market Rate - Rate Definition page. You can find definitions for the fields and controls later on this page.

Term	From Currency	To Currency	Maximum Variance	*Error Type
0	ESP	EUR	2.50	Warning

OK Cancel

Maximum Variance

You can modify the maximum variance—that is, the percentage of variance allowed when you maintain the market rate. If the change exceeds the tolerance, an error results. The default value is *2.50* (2.5 percent).

Error Type

You can modify the type of error that results when the tolerance defined in the Maximum Variance field is exceeded during data entry. Error type values are:

None: No error processing occurs. The new rate is used even though it exceeds the limit.

Stop : Processing halts. The system prevents you from saving the new rate.

Warning: A warning appears. You can ignore it and save the new rate.

Note: The results of changing the rate definition do not take effect until you save the Market Rates page.

Click OK to return to the Market Rates page.

Exchange Rate Detail Page

Use the Exchange Rate Detail page (EXCH_RT_DTL) to maintain exchange rates.

Navigation

Click the Exchange Rate Detail button on the Market Rates page. This page can also be accessed from other pages in the system.

Image: Market Rate - Exchange Rate Detail page

This example illustrates the fields and controls on the Market Rate - Exchange Rate Detail page. You can find definitions for the fields and controls later on this page.

Exchange Rate Detail

Rate Quotation Basis: Indirect

Quote Units: 1

Triangulate: N

Reference Currency:

Current Quote
166.38600000 ESP = 1 EUR

Historic Quote
Not Applicable

Exchange Rate		
From	To	Rate
ESP	EUR	166.38600000

OK Cancel

The read-only fields include:

Rate Quotation Basis

Displays the quotation basis for the exchange rate as it is defined on the Currency Quotation Method page. If no quotation method is defined, the quotation basis is *Direct*.

Quote Units

Displays the quote units for the exchange rate as defined on the Currency Quotation Method page. If no quotation method is defined, the quote unit is *1*.

Triangulate

Displays the triangulated setting for the exchange rate as it is defined on the Currency Quotation Method page. If no quotation method is defined, the triangulated setting is *N*.

Reference Currency

Displays triangulated exchange rates only, and shows the reference currency used in the triangulated exchange.

Current Quote

Displays the current exchange rate used to convert the from currency to the to currency. A direct, non-triangulated rate shows quote units (or 1) on the left side of the equal sign and

the visual rate on the right. For example: 1 USD = 1.40000000 CAD.

A triangulated rate displays two component rates of the triangle: the rate for converting the *from* currency to the *reference* currency, and the rate for converting the *reference* currency to the *to* currency.

Historic Quote

Displays a quote to indicate the quotation method originally used by a historic exchange rate if the system determines that the original quotation method of the historic rate differs from the current quotation method. This field displays the following information:

- A quote, if the historic rate has converted the *from* currency to the *to* currency directly using a calculated reciprocal rate, but the current quotation method for the currency pair is now indirect.
- A quote, if the historic quote method were non-triangulated and the current quote method is triangulated.
- *Not Applicable*, if the system does not determine that the historic and current quote methods are different.

Exchange Rate

Displays a single visual rate for non-triangulated exchange rates or displays all three component visual rates for triangulated exchange rates. The cross rate for triangulated exchange rates is editable only if the Allow Override check box is selected in the exchange rate's quotation method definition.

Calculating Currency Rates for EPM Currency Conversion

PeopleSoft-delivered ETL jobs move currency rate data from your source transaction system to EPM target warehouse tables. You need only use the currency rate PIA pages to redefine or modify your existing currency rate data.

This topic provide an overview of currency calculations and discusses how to:

- Calculate cross rates and reciprocal rates.
- Run the Currency Exchange Calculator tool.

Pages Used to Calculate Currency Rates

Page Name	Definition Name	Navigation	Usage
Cross/Reciprocal Rate Calc (calculator)	RUN_FIN9030	EPM Foundation, EPM Setup, Common Definitions, Currencies and Rates, Cross/Reciprocal Rate Calc, Cross/Reciprocal Rate Calculator, Cross/Reciprocal Rate Parameters	Calculate cross rates and reciprocal rates by defining parameters to run the FIN9030 SQR report.
Currency Exchange Calculator	CURRENCY_EXCHNG_PN	EPM Foundation, EPM Setup, Common Definitions, Currencies and Rates, Currency Exchange Calculator	Run the Currency Exchange Calculator to calculate the currency exchange between currencies.

Understanding Currency Calculations

PeopleSoft calculates currency rates for cross rates, triangulated rates, and reciprocal rates.

EPM utilizes two tools for currency calculations:

- The Currency Exchange Calculator tool quickly performs *ad hoc* currency conversion using the exchange rates that are stored on the market rates table.
- The Cross/Reciprocal Rate Calculator calculates exchange rates and updates the market rates table.

It performs three functions by generating the rates shown in this table:

Cross rates for nontriangulated currency pairs

For example, an organization subscribes to a rate service that provides all rates respective to the USD. Starting with a USD to Canadian dollar rate and a USD to Mexican peso rate, the system can calculate a new Canadian dollar to Mexican peso cross rate.

Triangulated rates for triangulated currency pairs

For example, the euro to NEW (a fictitious country that has just joined the euro and is subject to triangulation) fixed rate has been established on the market rate table and a new euro to USD rate has just been entered. Using this information, the process can create a new USD to NEW triangulated rate. The difference between triangulated rates and cross rates affects how the data is stored in the database. When calculating a cross rate, you actually create a new rate. When calculating a triangulated rate, the individual components of the source rates are stored on the target.

Reciprocal rates for those currency pairs that are not automatically reciprocated

For example, using a USD to CAD rate as the source, the process calculates the CAD to USD reciprocal. If quote methods are in place, the visual rate remains the same and a difference exists in how the data is stored in the database (RATE_MULT and RATE_DIV are inverse). If quote methods are not used, the

process calculates an inverse rate, meaning that the visual rates differ.

Currency Exchange Calculator Page

Use the Currency Exchange Calculator page (CURRENCY_EXCHNG_PN) to run the Currency Exchange Calculator to calculate the currency exchange between currencies.

Navigation

EPM Foundation, EPM Setup, Common Definitions, Currencies and Rates, Currency Exchange Calculator

Image: Currency Exchange Calculator page

This example illustrates the fields and controls on the Currency Exchange Calculator page. You can find definitions for the fields and controls later on this page.

The screenshot shows a web form titled "Currency Exchange Calculator". It contains the following fields and controls:

- *From Amount:** A text input field containing the value "1.000".
- *From Currency Code:** A dropdown menu with a magnifying glass icon.
- *To Currency Code:** A dropdown menu containing the value "USD" with a magnifying glass icon.
- *Exchange Rate Type:** A dropdown menu containing the value "OFFIC" with a magnifying glass icon.
- *Effective Date:** A date picker field showing "04/01/2010" with a calendar icon.
- Converted Amount:** A text input field containing the value "0.00".
- Calculate:** A button located at the bottom left of the form.

This tool enables you to enter a rate or an amount in a currency other than the base currency, or to compute an exchange using an alternative rate type.

From Amount	Enter a <i>from</i> amount. The currency exchange is based on the <i>from</i> amount that you enter and the current exchange rate set up on the Market Rates page.
From Currency Code	Select the currency code from which to calculate the exchange amount.
To Currency Code	Select the currency code to which to calculate the exchange amount.
Exchange Rate Type	Select the type of exchange rate to use for this calculation.
Converted Amount	Displays the converted amount. The system automatically calculates this amount when you save the page.

Note: Do not decrease the number of decimals after you have created transactions for that currency; the system will not properly round the previous rounded amount fields with the new precision.

Configuring Currency Precision for Currency Conversion

This topic provides an overview of currency precision and discusses how to:

- Activate currency precision.
- Maintain currency precision by currency.
- Report with currency precision.

Understanding Currency Precision

According to the ISO standard, currency precision can range from zero decimals to three decimals. For example, USD amounts have two digits to the right of the decimal, and JPY have none. To support this dynamic currency precision, the system delivers all of its currency-sensitive amount fields with a standard length of 23.3, or 23 digits to the left of the decimal and three digits to the right. A control currency on the same record exists to control the display and processing of such amount fields.

PeopleSoft applications round all currency-sensitive amount fields to the currency precision of the controlled currency during all online or background processes. For example, in a database that contains amount fields with a length of 23.3, JPY are rounded to 123.000 and USD are rounded to 123.230. The system does not place a nonzero after the decimal for a JPY amount or after the second digit to the right of the decimal for a USD amount.

Although amount fields are stored in the database with decimal placeholders, the system displays amount fields with the precision that is appropriate for the currency. For example, it displays JPY as 123 and USD as 123.23. When you enter an amount, you cannot enter more than the defined precision. If you attempt to do so, the system treats the entry as an online error.

PeopleCode programs and background processes round all currency-sensitive amount fields to the currency precision of the controlled currency.

PeopleSoft-delivered ETL jobs move currency precision data from your source transaction system to EPM target warehouse tables. You need only use the currency precision PIA pages to redefine or modify your existing currency precision data.

Related Links

[Configuring Currency Precision for Currency Conversion](#)

Activating Currency Precision

Currency precision is a PeopleSoft PeopleTools option. When it is selected using the PeopleTools Options page, all features of currency precision are activated. When the option is deselected, all amount fields behave as if no controlled currency exists. The system displays amount fields as defined in the PeopleSoft Application Designer and rounds them to the number of decimals defined in the Application Designer.

Note: If you deselect the multicurrency check box, the system only supports the default amount field size of 15.3—it does not support the larger amount field size of 23.3. After you deselect this check box, selecting it again does not automatically round existing transaction amounts.

Maintaining Currency Precision by Currency

Use the Currency Code page to access the currency code table, in which you define the decimal position by currency. The values in this table are effective dated. The software is shipped with the currency code table in compliance with ISO standards for decimal positions. You can increase the number of decimals to a maximum of three.

Warning! Do *not* decrease the number of decimals after transactions are entered in that currency; the system does not properly round the previously rounded amount fields with the new precision.

Related Links

[Currency Code Page](#)

Reporting with Currency Precision

Most PeopleSoft SQR reports display currency-controlled amounts with the number of decimal places that are defined by the associated currency. For example, a JPY amount appears as 123 on a report, and a United USD amount appears as 123.23.

Amounts on PS/nVision (Microsoft Excel) reports appear as two-decimal-place numbers. If you want to show three decimal places on these reports, you must configure the reports to do so.

Third-party reporting tools used by PeopleSoft do not fully support numeric fields greater than 15 digits. Microsoft Excel uses an eight-byte float for numeric fields, which causes values to be truncated after the fifteenth digit.

For any of these reporting tools, the accuracy of the results is:

- Hundreds of trillions of yen (precision = 0).
- Trillions of dollars (precision = 2).
- Hundreds of billions of dinar (precision = 3).

For example, if you populate a 23.3 numeric database amount field with the number 2, the following table illustrates the number that is displayed in each type of report.

Number of Digits	Excel	SQR
16	2,222,222,222,222.220	2,222,222,222,222.222
17	22,222,222,222,222.200	22,222,222,222,222.220
18	222,222,222,222,222.000	222,222,222,222,222.200

Chapter 6

Setting Up EPM Security

Understanding EPM Security and Setups

This topic discusses:

- PeopleSoft application security
- EPM security
- Setting up EPM security
- EPM security views

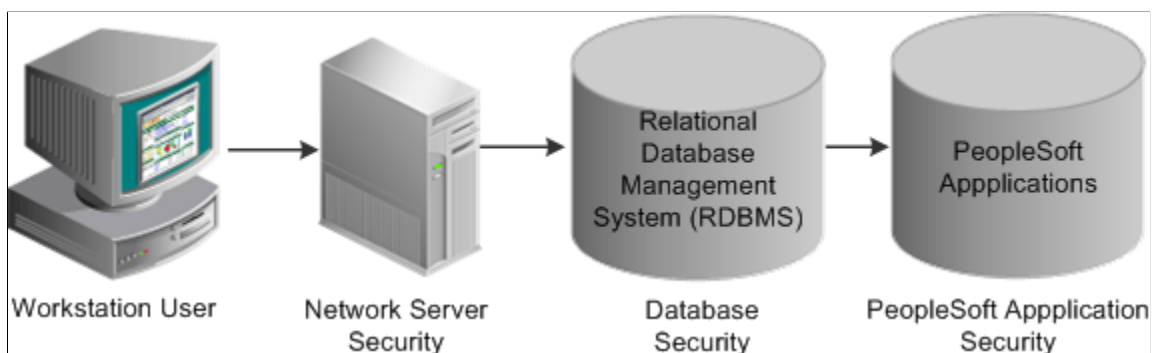
PeopleSoft Application Security

PeopleSoft applications use multilevel security to enable you to successfully manage shared data environments. You set up data access at different entry points within your system and define the most efficient path to data across business groups, tables, departments, pages, and so forth. You have full control over security definitions, selecting options to create a matrix that enables or restricts user access to data through a series of authorizations.

Security access covers three areas: networks, databases, and applications. Network security controls the overall point of entry into your system hardware and software resources. Database security narrows the scope of a user's information access. At the application level, security extends to the field level. This diagram illustrates PeopleSoft application security levels:

Image: Levels of security in PeopleSoft applications

This example illustrates the Levels of security in PeopleSoft applications.



Application Security Terms

This table describes the various types of PeopleSoft application security:

Security Type	Location	Function
Network	Network software	Controls entry into the network and authorizes rights to use shared resources.
Relational Database Management System (RDBMS)	Operating system	Controls access to the database.
User	PeopleTools	Controls access to application pages, functions, and business components.
Object	PeopleTools	Controls access to objects or object groups that are used in application development.
Query	PeopleTools	Defines table row sets that are accessed for performing system queries.
Row-level	PeopleTools and PeopleSoft applications	Controls access to the subset of data rows within tables that the user is authorized to review or update.
Field-level	PeopleCode	Controls access to individual fields on pages.

EPM Security

EPM security controls access to specific data within the EPM database and enables you to grant users access to specific rows, columns, or fields in an EPM database table. Data-access privileges within the warehouse can be defined for both Operational Warehouse - Enriched (OWE) and Multidimensional Warehouse (MDW) tables, and for dimension and fact tables.

You can specify EPM security access using any of the following methods:

- Field-level security
- Ledger security
- Dimension (row-level) security
- Metric (column-level) security

EPM security provides a single point of entry for defining and maintaining data access rules across all the EPM Warehouses and Analytical Applications.

Setting Up EPM Security

EPM security is extremely flexible and enables you to restrict user access to EPM database tables in a variety of ways, including field level, row level, and column level restrictions. PeopleSoft-delivered ETL jobs move security data from your source transaction system to EPM target warehouse tables. You must

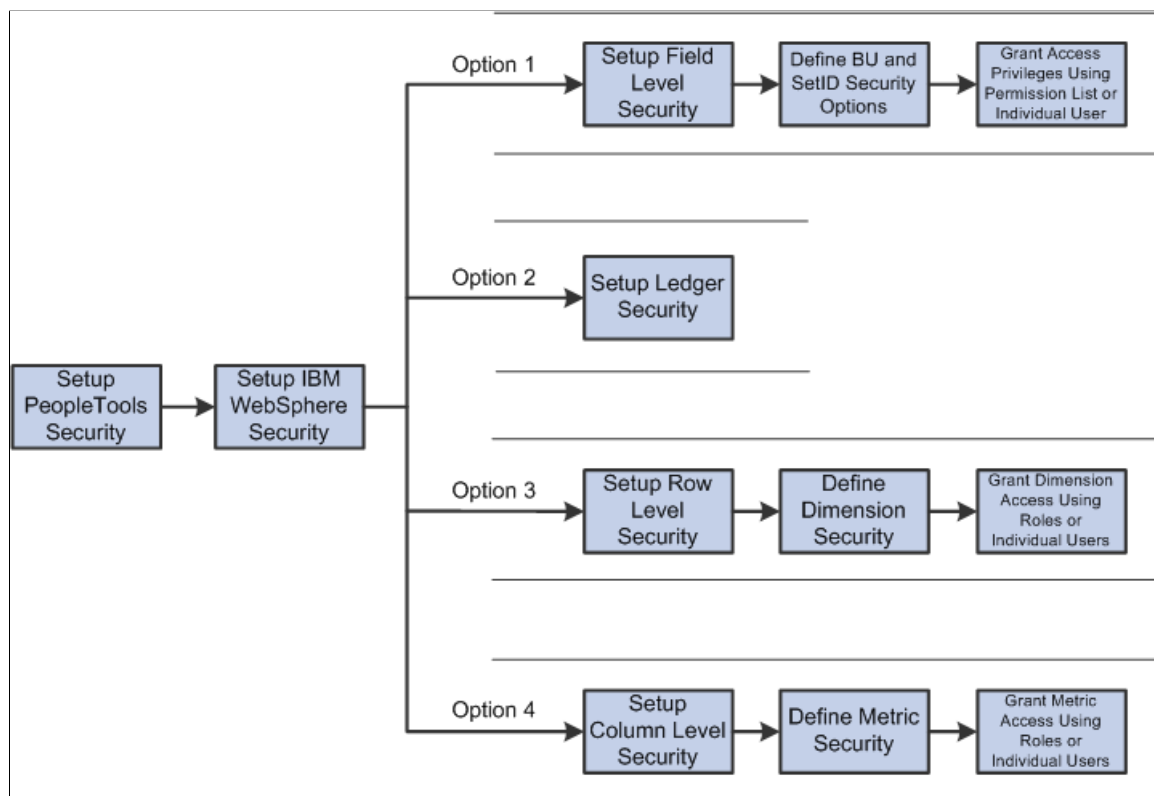
set up PeopleTools and IBM WebSphere DataStage security, but you need only use the security pages documented in this topic to redefine or modify your existing security data.

You have the option of implementing only one type of EPM security restriction or all four types:

- Field-level security.
- Ledger security.
- Row-level security.
- Column-level security.

Image: EPM Security Setup Process Flow

The following security setup process flow demonstrates the different security setup options:



As demonstrated in the diagram, you can implement any combination of field level, ledger, row level, and column level security restrictions.

Note: Refer to your EPM Warehouse and Analytical Application documentation for any special security setups required for those products.

PeopleTools security

PeopleTools security provides user authentication and application-level security. Users are authorized to access the EPM database using PeopleTools login security. Users may be granted or denied access to each application and components therein. EPM security works together with PeopleTools security, so you must properly define PeopleTools security prior to setting up EPM security.

See *PeopleSoft PeopleTools: Security Administration*

IBM WebSphere DataStage Security

PeopleSoft EPM delivers the IBM WebSphere extract, transform, and load (ETL) tool to load your source data into the Operational Warehouse - Staging (OWS) and migrate the data to the OWE and the MDW. The tool builds three layers of security into the repository and its objects. Access is granted through groups and users.

The three layers of security are:

1. A user role that is created by the EPM Production Manager.

The Production Manager assigns Windows user groups to one of four DataStage user categories. For example, if you belong to the Windows Administrators user group, you might be assigned to the DataStage Production Manager category and have full access to all areas of a DataStage project. This is the default for DataStage users.

2. Project privileges that range from browse-only to full Production Manager privileges.
3. Object-level safeguards that prevent users from jointly accessing or overwriting the same object.

For example, if a user is working on a job, that object is locked so that no other user can access it. Locks are automatically created to avoid repository contamination.

The following DataStage user categories are defined in PeopleSoft EPM: the EPM Production Manager, Developer, Operator, and None. The Production Manager has full access to all areas of a DataStage project, including repository objects, and can create and manipulate protected projects. The Production Manager grants access to all other groups and users. Developers have full access to all areas of a DataStage project and can create and modify ETL jobs. This is the default setting. Operators have permission to run and manage DataStage jobs. Users who are assigned to None do not have permission to sign in to DataStage.

See *IBM Information Server: Administration Guide*

Field Level Security

Field level security enables you to restrict users to specific fields on your EPM database tables based on business units and SetIDs (because the tables are keyed by SetID or business unit).

To set up field level security you must:

1. Enable field level security and specify related options.
2. Associate security views with specific users, permission lists, business units, or SetIDs.
3. Apply the aforementioned field level security parameters.
4. Associate specific users or permission lists with the business unit and/or SetID restrictions (data group restrictions) you defined in the previous steps.

See [Specifying Field-Level Security Options](#).

Ledger Security

Ledger security enables you to restrict users to specific EPM ledger tables. You can restrict access to the tables by user or permission list.

See [Establishing Ledger Security](#).

Dimension (Row Level) and Metric (Column Level) Security

Row level security enables you to secure individual rows of an EPM dimension table. Column level security enables you to secure individual columns of an EPM database table. Usually, these are measures on fact tables, but may also be attributes on a dimension table that contain sensitive data, such as Employees' Salary or the Social Security Number columns. With row level and metric level support, you can restrict access by individual user or security roles.

To set up row level security you must:

1. Define dimension security.
2. Associate the dimension with a security join table.

Security join tables are EPM database tables that store the security profiles for users along with the corresponding dimension values for which they have access.

3. Define users and security roles.
4. Assign dimension rows to the user or security role.

To set up column level security you must:

1. Define metric security.
2. Define users and security roles.
3. Assign metric to the user or security role.

See [Defining Dimension and Metric Security](#).

See [Understanding Role and User Based Security for Dimensions and Metrics](#).

EPM Security Views

Security views are SQL SELECT statements that filter out data rows whose key values are not needed as valid access parameters. The result is that users who are authorized to access SetIDs or business units see only a subset of values from these edit table values. PeopleSoft EPM delivers prepackaged security views for most securable objects in EPM and the views tell the system which table views to reference. To use these security views in PeopleTools, use the views as a search view.

Object	Delivered EPM Security View
Personal Data	PERSONAL_SRCH
Job Data	JOB_SRCH
Job Code	JOBCODE_SRCH
Location	LOCATION_SRCH
Scenario	SCENARIO_SRCH

You can alter these prepackaged views or build your own.

Specifying Field-Level Security Options

You can set up field-level security for business units and SetIDs because EPM warehouse tables are keyed by either SetID or business unit. The Security Options page enables you to enable this type of security. The page updates one row in the INSTALLATION_FS table. Only one row is ever in this table (only one type of security can be active at a time). For example, if you select operator security, only selected fields (business unit, SetID, or both) have security applied.

After specifying field-level security options you must associate security views with specific users, permission lists, business units, or SetIDs and run the apply security application engine process (which reads the INSTALLATION_FS table, determines the security type, such as *none*, *permission list* or *operator*, and applies your selected field-level security parameters).

After running the apply security process, you must associate specific users or permission lists with the business unit and/or SetID restrictions (data group restrictions) you defined in the previous steps.

This topic discusses how to:

- Set up field-level security options.
- Set up security view names.
- Establish SetID security by permission list or user ID.
- Establish business unit security by permission list or user ID.

Pages Used to Specify Field-Level Security Options

Page Name	Definition Name	Navigation	Usage
Security Options	SECURITY_OPTIONS	EPM Foundation, EPM Security, Advanced, Security Options	Set up field-level security for business units and SetIDs.
Security View Names	SECURITY_VIEWS	EPM Foundation, EPM Security, Advanced, Security View Names	Set up security view names.
Request Security Processing	RUN_FIN9001	EPM Foundation, EPM Security, Advanced, Apply Security Setups, Request Security Processing	Run the Apply Security process to apply the security setup throughout the system.
TableSet Security by Permission List	SEC_SETID_CLS	EPM Foundation, EPM Security, Advanced, TableSet Security by Perm (Permission) List	Establish SetID security by permission list. Valid values for permission list are taken from your PeopleTools security setup.

Page Name	Definition Name	Navigation	Usage
TableSet Security by User ID	SEC_SETID_OPR	EPM Foundation, EPM Security, Advanced, TableSet Security by User ID	Establish SetID security by user ID. Valid values for user are taken from your PeopleTools security setup.
Unit Security by Permission List	SEC_BU_CLS	EPM Foundation, EPM Security, Advanced, Unit Security by Perm (Permission) List	Establish business unit security by permission list.
Unit Security by User ID	SEC_BU_OPR	EPM Foundation, EPM Security, Advanced, Unit Security by User ID	Establish business unit security by user ID.

Security Options Page

Use the Security Options page (SECURITY_OPTIONS) to set up field-level security for business units and SetIDs.

Navigation

EPM Foundation, EPM Security, Advanced, Security Options

Image: Security Options page

This example illustrates the fields and controls on the Security Options page. You can find definitions for the fields and controls later on this page.

The screenshot shows the 'Security Options' page with two main sections:

- Type of Security:** Three radio button options:
 - No Security
 - Operator Level Security
 - Permission List Level Security
- Secured Fields:** A grid of checkboxes for selecting fields to secure:
 - Unit
 - Setid
 - Ledger
 - Book
 - Project
 - Analysis Group
 - Pay Cycle

You can implement security using any of the following fields:

- Unit: Business unit, the primary key for all transaction data.
- SetID: The primary key for all accounting structure and rules tables.
- Ledger
- Book
- Project
- Analysis Group
- Pay Cycle

For either field, you can set the security in the following ways:

No Security

Disables PeopleSoft application security. All users who are authorized to access a page may select any valid SetID or business unit.

Operator Level Security

Enables PeopleSoft application security. Users are limited to the SetIDs and business units that are specified by their user IDs.

Permission List Level Security

Enables PeopleSoft application security. Users are limited to the key fields that are specified by the permission list that is assigned to their user IDs. All users in a permission list have the same level of security.

Security View Names Page

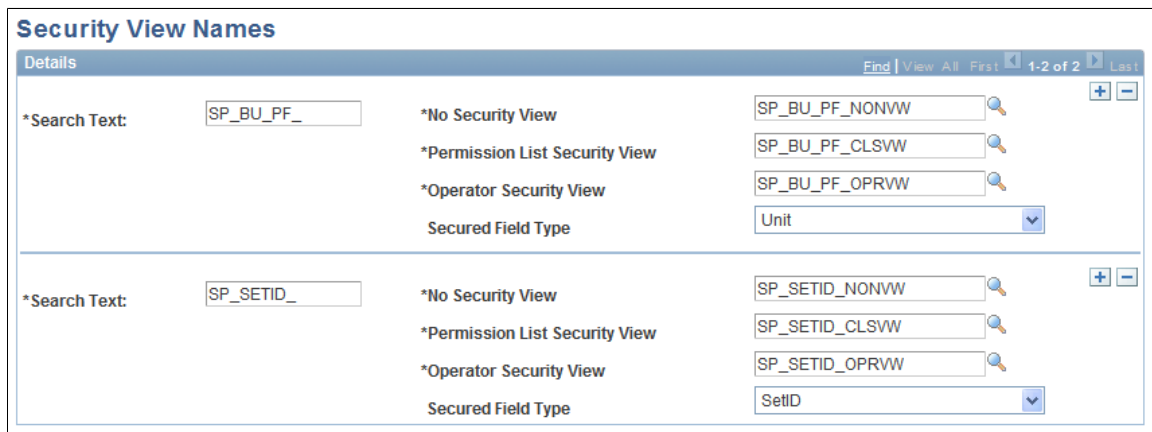
Use the Security View Names page (SECURITY_VIEWS) to set up security view names.

Navigation

EPM Foundation, EPM Security, Advanced, Security View Names

Image: Security View Names page

This example illustrates the fields and controls on the Security View Names page. You can find definitions for the fields and controls later on this page.



After you have set up your security views, you can specify which users or permission lists can access the pages that contain secured field values using the Security View Names page. Within each page, you also can hide specific fields from particular permission lists. The Security View Names page also indicates the type of field (business unit or SetID) secured by each view.

The system stores the list in the SEC_VIEW_NAMES table, where you can review or update this information. You can add any security views that you configure for your system.

View names use one of three file extensions to reflect the type of security that you selected when you specified your security options:

- _NONVW: Indicates that no security has been selected for that view.
- _OPRVW: Indicates that operator (user) security has been selected for that view.

- `_CLSVW`: Indicates that permission list security has been selected for that view.

Use the Search Text field to search for a security view table. This field lists the view name prefixes that are supplied by each of your applications. When you run the Apply Security Setup process, the process uses these prefixes to search the system for view names that begin with these prefixes. If a view name begins with a prefix from this list, the process changes the view name extension to match the security type that you selected when you specified security options. The type of field (business unit or SetID) that is secured by each view is embedded in the view name.

After you have selected your security options and defined your security views, you must run the Apply Security FIN9001 SQR process for the options to take effect.

Related Links

[EPM Security Views](#)

Request Security Processing Page

Use the Request Security Processing page (RUN_FIN9001) to run the Apply Security process to apply the security setup throughout the system.

Navigation

EPM Foundation, EPM Security, Advanced, Apply Security Setups, Request Security Processing

Run the Apply Security process to apply the security parameters you set up on the Security Options and Security View Names pages. The Apply Security process reads the INSTALLATION_FS table to determine the security type (none, permission list, operator) and then switches the prompt table views to the new view names, based on SEC_VIEW_NAMES. The process searches for the values in the Search Text field on the Security View Names page and replaces the table names as specified.

Note: This process should not be confused with the request security processing process (PF_SECURITY) which is discussed later in this topic. The PF_SECURITY process applies security parameters specified for security roles and users.

TableSet Security by Permission List Page

Use the TableSet Security by Permission List page (SEC_SETID_CLS) to establish SetID security by permission list or use the TableSet Security by User ID page (SEC_SETID_OPR) to establish SetID security by user ID.

Valid values for permission list are taken from your PeopleTools security setup.

Valid values for user are taken from your PeopleTools security setup.

Navigation

EPM Foundation, EPM Security, Advanced, TableSet Security by Perm (Permission) List

EPM Foundation, EPM Security, Advanced, TableSet Security by User ID

Image: TableSet Security by Permission List page

This example illustrates the fields and controls on the TableSet Security by Permission List page. You can find definitions for the fields and controls later on this page.

For each user ID or primary permission list, select the SetID or SetIDs that the users or permission list can access.

Unit Security by Permission List Page

Use the Unit Security by Permission List page (SEC_BU_CLS) to establish business unit security by permission list and use the Unit Security by User ID page (SEC_BU_OPR) to establish business unit security by user ID.

Navigation

EPM Foundation, EPM Security, Advanced, Unit Security by Perm (Permission) List

EPM Foundation, EPM Security, Advanced, Unit Security by User ID

Image: Unit Security by Permission List page

This example illustrates the fields and controls on the Unit Security by Permission List page. You can find definitions for the fields and controls later on this page.

For each user ID or primary permission list, select the warehouse business unit or units that the user or permission list can access.

Establishing Ledger Security

You can determine which users have access to ledger data by securing access to specific ledgers. For a specified ledger you can grant access to users by permission list or user ID.

Pages Used to Define Ledger Security

<i>Page Name</i>	<i>Definition Name</i>	<i>Navigation</i>	<i>Usage</i>
Ledger/Perm List	SEC_LEDGER_CLS	EPM Foundation, EPM Security, Advanced, Ledger Security by Permission List, Ledger/Perm List	Select a ledger to associate with a security permission list.
Ledger by User ID	SEC_LEDGER_OPR	EPM Foundation, EPM Security, Advanced, Ledger Security by User ID, Ledger by User ID	Select a ledger to associate with a user ID.

Ledger/Perm List Page

Use the Ledger/Perm List page (SEC_LEDGER_CLS) to select a ledger to associate with a security permission list and use the Ledger by User ID page (SEC_LEDGER_OPR) to select a ledger to associate with a user ID.

Navigation

EPM Foundation, EPM Security, Advanced, Ledger Security by Permission List, Ledger/Perm List

EPM Foundation, EPM Security, Advanced, Ledger Security by User ID, Ledger by User ID

Use these pages to select a ledger to associate with a security permission list or user ID.

Specifying Row and Column Level Security

This topic provides an overview of role and user-based security and discusses how to:

- Define security roles.
- Grant dimension and metric access to security roles.
- Assign users to security roles.
- View summary for security roles.
- Grant dimension and metric access to a user.
- View summary for a secured user.
- Request security processing

Pages Used to Establish Role and User-Based Security

<i>Page Name</i>	<i>Definition Name</i>	<i>Navigation</i>	<i>Usage</i>
Define EPM Security Roles	PF_SY_ROLE_DEFN	EPM Foundation, EPM Security, Security By Role, Define EPM Security Roles	Define security roles.
Access To Metric	PF_SY_ROLE_METR	EPM Foundation, EPM Security, Security By Role, Role Metric Access, Access To Metric	Assign a metric to a security role.
Role Dimension Access	PF_SY_ROLE_ALL_MDW	EPM Foundation, EPM Security, Security By Role, Role Dimension Access, Role Dimension Access	Assign a dimension table to a security role and specify high-level access privileges. The object name of this page changes depending on whether an OWE or MDW dimension is selected.
Individual Selection	PF_SY_ROLE_LIST	EPM Foundation, EPM Security, Security By Role, Role Dimension Access, Individual Selection	Specify row-level access to the dimension based on SetID and dimension key fields.

Page Name	Definition Name	Navigation	Usage
Constraint-based Selection	PF_SY_ROLE_CONS	EPM Foundation, EPM Security, Security By Role, Role Dimension Access, Constraint-based Selection	Specify row-level access to the dimension based on constraint.
Select Security Column	PF_SY_COLUMN_PG	Automatically accessed when you select a constraint on the Constraint-based Selection page that contains two or more columns and the system cannot distinguish the column you want to include in your security parameters	Select one column to include in your constraint.
Tree-based Selection	PF_SY_ROLE_TREE	EPM Foundation, EPM Security, Security By Role, Role Dimension Access, Tree-based Selection	Specify row-level access to the dimension based on an existing tree hierarchy that is defined for the dimension. You can use the tree to grant a user access to specific nodes, leaves, or details in the tree. Used with OWE dimensions only.
User Role Access	PF_SY_USER_ROLES	EPM Foundation, EPM Security, Security By User, User Role Access	Assign a user to a security role.
Role Security Summary	PF_SY_ROL_SUMMARY	EPM Foundation, EPM Security, Security By Role, Role Security Summary	Review access privileges and other information for a security role.
Review Role Dimension Access	PF_SY_ROLE_DETAIL	Click the <i>Details</i> link on the Role Security Summary page.	Review additional details about a dimension that is associated with the selected security role
Access To Metric	PF_SY_ROLE_METR	EPM Foundation, EPM Security, Security By User, User Metric Access, Access To Metric	Assign a metric to a user.
User Dimension Access	PF_SY_ROLE_ALL	EPM Foundation, EPM Security, Security By User, User Dimension Access, User Dimension Access	Assign a dimension table to a user and specify high-level access privileges.
Individual Selection	PF_SY_ROLE_LIST	EPM Foundation, EPM Security, Security By User, User Dimension Access, Individual Selection	Specify row-level access to the dimension based on SetID and dimension key fields.
Constraint-based Selection	PF_SY_ROLE_CONS	EPM Foundation, EPM Security, Security By User, User Dimension Access, Constraint-based Selection	Specify row-level access to the dimension based on constraint.

Page Name	Definition Name	Navigation	Usage
Select Security Column	PF_SY_COLUMN_PG	Automatically accessed when you select a constraint on the Constraint-based Selection page that contains two or more columns and the system cannot distinguish the column you want to include in your security parameters.	Select one column to include in your constraint.
Tree-based Selection	PF_SY_ROLE_TREE	EPM Foundation, EPM Security, Security By User, User Dimension Access, Tree-based Selection	Specify row-level access to the dimension based on an existing tree hierarchy that is defined for the dimension. You can use the tree to grant a user access to specific nodes, leaves, or details in the tree. Used with OWE dimensions only.
User Security Summary	PF_SY_SUMMARY	EPM Foundation, EPM Security, Security By User, User Security Summary	Review access privileges and other information for a user.
Review User Dimension Access	PF_SY_USER_DETAIL	Click the <i>Details</i> link on the User Security Summary page.	Review additional details about a dimension that is associated with the selected user.
Request Security Processing	RUN_PF_SECURITY	EPM Foundation, EPM Security, Advanced, Request Security Processing	Apply security parameters for security roles and users by running the request security processing (PF_SECURITY) process.

Defining Dimension and Metric Security

Because EPM is delivered with no security restrictions, dimensions and metrics (also known as fact-columns) are also delivered unsecured. Before you can grant a user access to a dimension or metric, you must first indicate to the system that a particular dimension or metric requires securing. The pages used to define dimension and metric security are discussed in the security topic of this documentation.

See [Defining Dimension and Metric Security](#).

Note: Dimensions and metrics that are not secured are classified as public, or unsecured. All EPM users can view public objects.

Understanding Role and User Based Security for Dimensions and Metrics

After you designate dimensions and metrics that require securing, you must grant users access to those objects. You can grant security access to an individual user or to a specific security *role*.

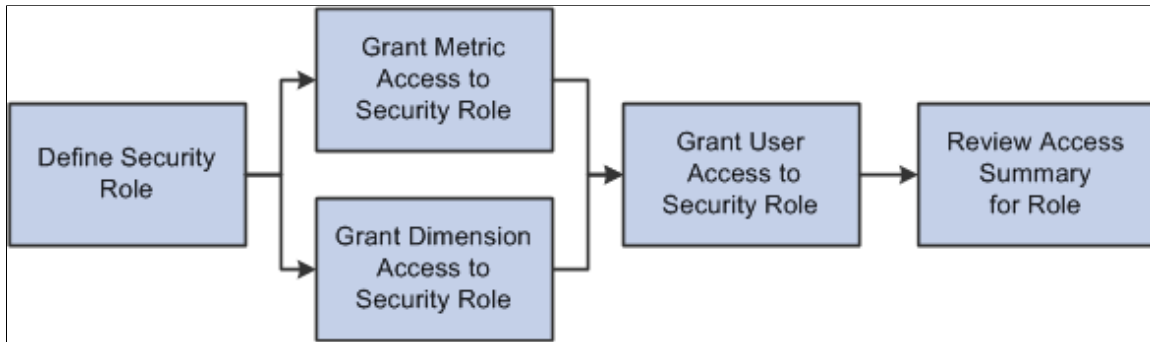
EPM security enables you to create security roles. A security role is a set of data access privileges that are assigned to one or more users. A user who is assigned to a specific role inherits all access privileges that

are associated with that role. A user can belong to multiple roles. In this case, the user would inherit the combined privileges that are defined for all roles.

To set up security roles, define the role, assign dimension and metric access privileges to the role, and then assign users to the role.

Image: Process flow - security role setup

The following diagram depicts this process.



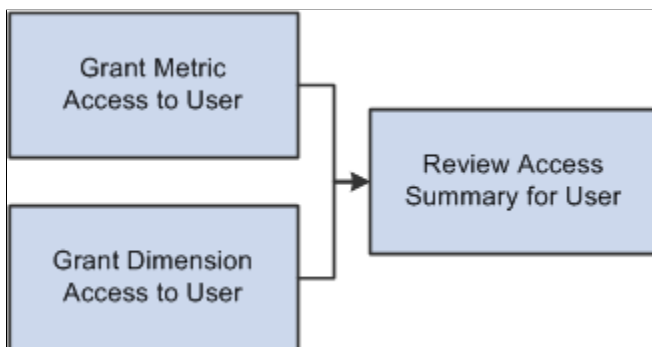
If you have established security roles for your PeopleTools security, you can import the roles into the EPM database using the Run Security Processing page.

See [Request Security Processing Page](#).

EPM security also enables you to define access privileges for individual users. To set up user access privileges, assign dimension and metric access privileges to a specific user.

Image: Process flow - user security setup

This example illustrates the Process flow - user security setup. .



Dimension Security and Individual, Constraint, and Tree Based Selections

EPM security provides three methods to specify row-level security for your dimension:

- Individual-based definition: Enables you to specify row-level security using the SetID and dimension key fields.
- Constraint-based definition: Enables you to specify row-level security by associating a constraint with a dimension table.

The constraint limits access to a dimension by acting as the WHERE clause in a SQL statement—for example, SELECT Account ID FROM Account Dimension WHERE Account ID = Northwest.

Set ID and Constraint ID are used to specify constraint access.

- Tree-based definition: Enables you to specify row-level security using existing tree hierarchies that are defined for a dimension.

You can use the tree to grant a user access to specific nodes, leaves, or details in the tree. Tree hierarchy use is limited to OWE tables only.

Processing Role and User Based Security Parameters

After the security rules have been set up, the EPM security application engine (PF_SECURITY) process must be run. This processes the access as defined in the Role Dimension access pages and flattens the data to the individual dimension members and populates the security join tables specified in dimension metadata.

Define EPM Security Roles Page

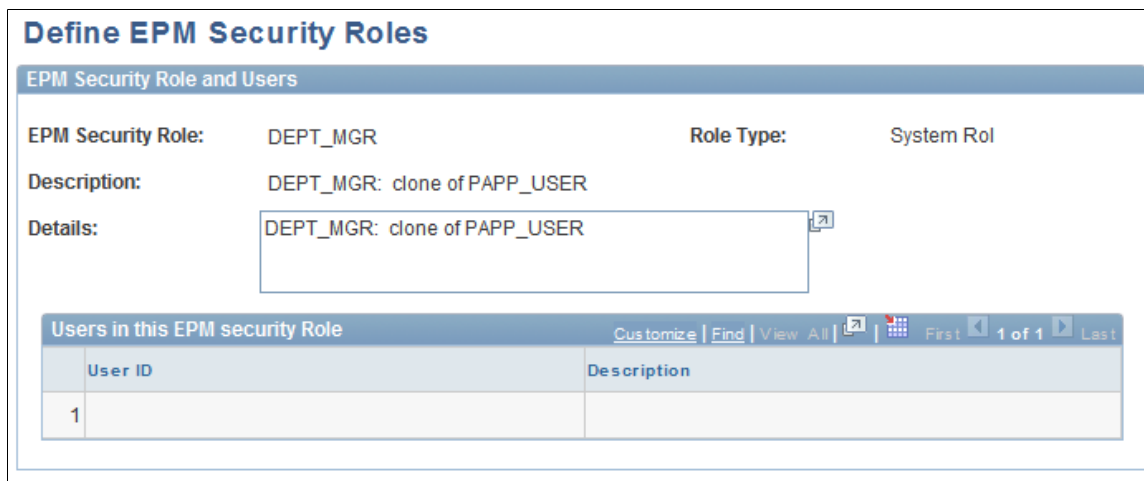
Use the Define EPM Security Roles page (PF_SY_ROLE_DEFN) to define security roles.

Navigation

EPM Foundation, EPM Security, Security By Role, Define EPM Security Roles

Image: Define EPM Security Roles page

This example illustrates the fields and controls on the Define EPM Security Roles page. You can find definitions for the fields and controls later on this page.



EPM Security Role Displays the name of the security role that you are defining.

Role Type Displays the type of role that is being defined.

Possible role types include *EPM Role*, *System Role*, *User Role*, *WFA Generated Role*.

Users in This EPM Security Role

User ID Displays the users who are associated with this role.

Access To Metric Page

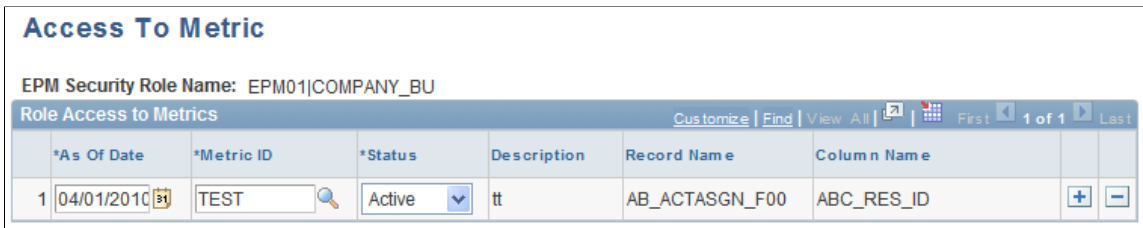
Use the Access To Metric page (PF_SY_ROLE_METR) to assign a metric to a security role.

Navigation

EPM Foundation, EPM Security, Security By Role, Role Metric Access, Access To Metric

Image: Access To Metric page

This example illustrates the fields and controls on the Access To Metric page. You can find definitions for the fields and controls later on this page.



Metric ID Displays the metric that you are associating with a particular security role.

Record Name Displays the record that is associated with the selected metric.

Column Name Displays the column that is associated with the selected metric.

Role Dimension Access Page

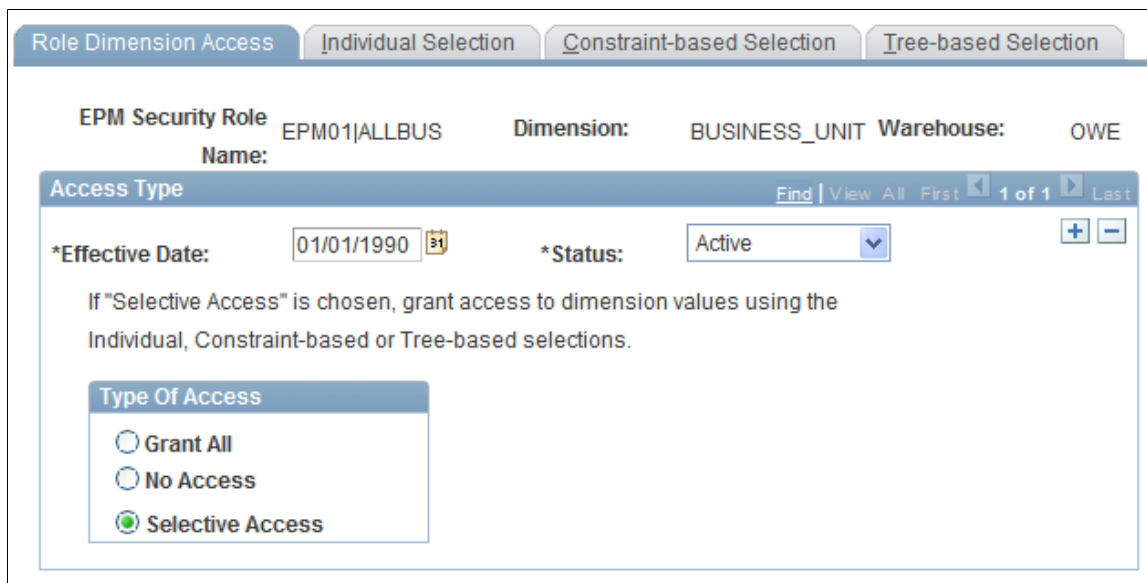
Use the Role Dimension Access page (PF_SY_ROLE_ALL_MDW) to assign a dimension table to a security role and specify high-level access privileges.

Navigation

EPM Foundation, EPM Security, Security By Role, Role Dimension Access, Role Dimension Access

Image: Role Dimension Access page

This example illustrates the fields and controls on the Role Dimension Access page. You can find definitions for the fields and controls later on this page.



Dimension Displays the dimension that you are associating with a particular security role.

Warehouse Displays the warehouse layer that is associated with the selected dimension.

Type of Access

Grant All Select this option to grant the role access to the entire dimension.

No Access Select this option to bar the role from accessing the entire dimension.

Selective Access Select this option to grant the role access to specific rows in the dimension.

You can specify rows individually based on SetIDs, using a constraint or using a hierarchy tree that is defined for the dimension (tree hierarchies are available only for OWE dimensions) .

Individual Selection Page

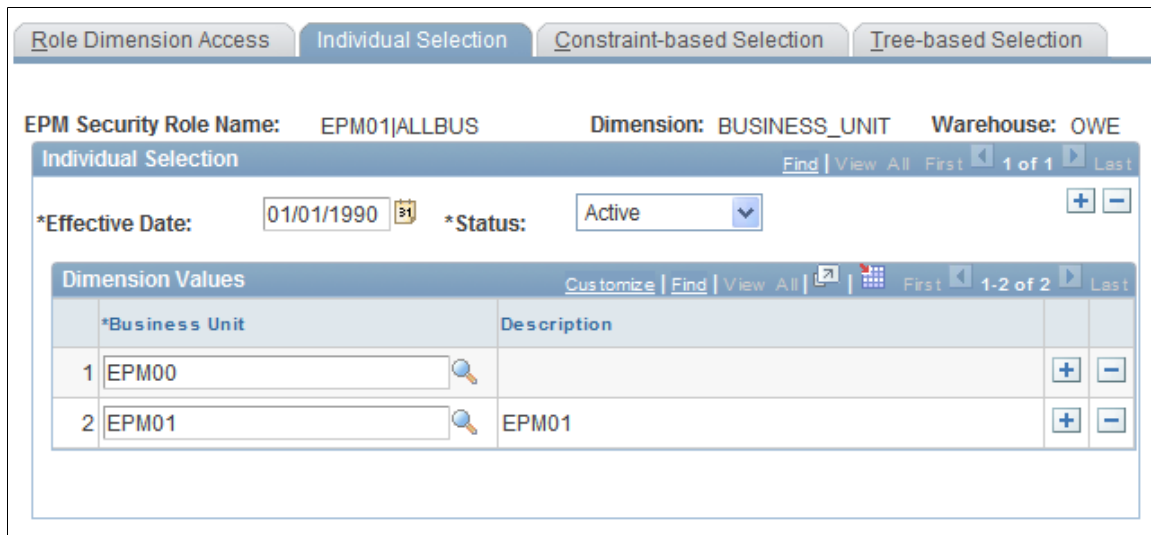
Use the Individual Selection page (PF_SY_ROLE_LIST) to specify row-level access to the dimension based on SetID and dimension key fields.

Navigation

EPM Foundation, EPM Security, Security By Role, Role Dimension Access, Individual Selection

Image: Individual Selection page

This example illustrates the fields and controls on the Individual Selection page. You can find definitions for the fields and controls later on this page.



Dimension Displays the dimension that you are associating with a particular security role.

Warehouse Displays the warehouse layer that is associated with the selected dimension.

Dimension Values

SetID or Business Unit Enter the SetID or business unit that is associated with the dimension rows that you want to secure.

This field can display either SetID or Business Unit, depending on the dimension you select. In some instances, there is no value displayed for the field .

Dimension Key Enter the dimension key that is associated with the dimension.

Because this is a dimension key field, the name of this field changes depending on the selected dimension. For example, if the Product (PRODUCT) table were selected, *Product ID*would be displayed because it is the dimension key for that table.

Constraint-based Selection Page

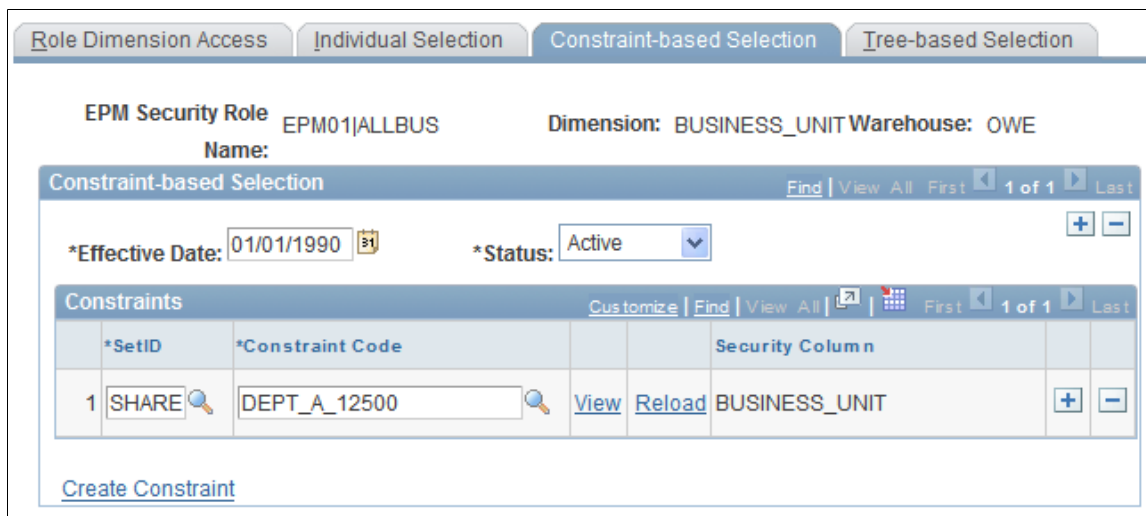
Use the Constraint-based Selection page (PF_SY_ROLE_CONS) to specify row-level access to the dimension based on constraint.

Navigation

EPM Foundation, EPM Security, Security By Role, Role Dimension Access, Constraint-based Selection

Image: Constraint-based Selection page

This example illustrates the fields and controls on the Constraint-based Selection page. You can find definitions for the fields and controls later on this page.



Dimension Displays the dimension that you are associating with a particular security role.

Warehouse Displays the warehouse layer that is associated with the selected dimension.

Constraint-based Selection

SetID Enter the SetID that is associated with the dimension rows that you want to secure.

Constraint Code Enter the constraint that you want to associate with the selected dimension rows.

You must have a constraint defined before you can access it here. If you do not have a constraint defined, you can use the Create Constraint link to create a new constraint.

Note: The Select Security Column page displays if the constraint you select contains two or more columns and the system cannot distinguish the column you want to include in your security parameters.

Reload Click to refresh the constraint definition if you have changed it after it was included in a security role.

Security Column Displays the field from the constraint that is used as the column to restrict access.

Create Constraint

Click to access the Constraints page and define a constraint.

If you have not created a constraint for the selected dimension rows, you can do so in the Constraints page.

Select Security Column Page

Use the Select Security Column page (PF_SY_COLUMN_PG) to select one column to include in your constraint.

Navigation

Automatically accessed when you select a constraint on the Constraint-based Selection page that contains two or more columns and the system cannot distinguish the column you want to include in your security parameters

When you select a constraint (on the Constraint-based Selection page) that contains two or more columns and the system cannot distinguish the column you want to include in your security parameters, the Select Security Column page is accessed automatically. The page displays the columns available to use in the constraint you selected. You must choose just one of the columns for the constraint. Select the column you want to include by clicking the column name in the Key ID field.

Tree-based Selection Page

Use the Tree-based Selection page (PF_SY_ROLE_TREE) to specify row-level access to the dimension based on an existing tree hierarchy that is defined for the dimension.

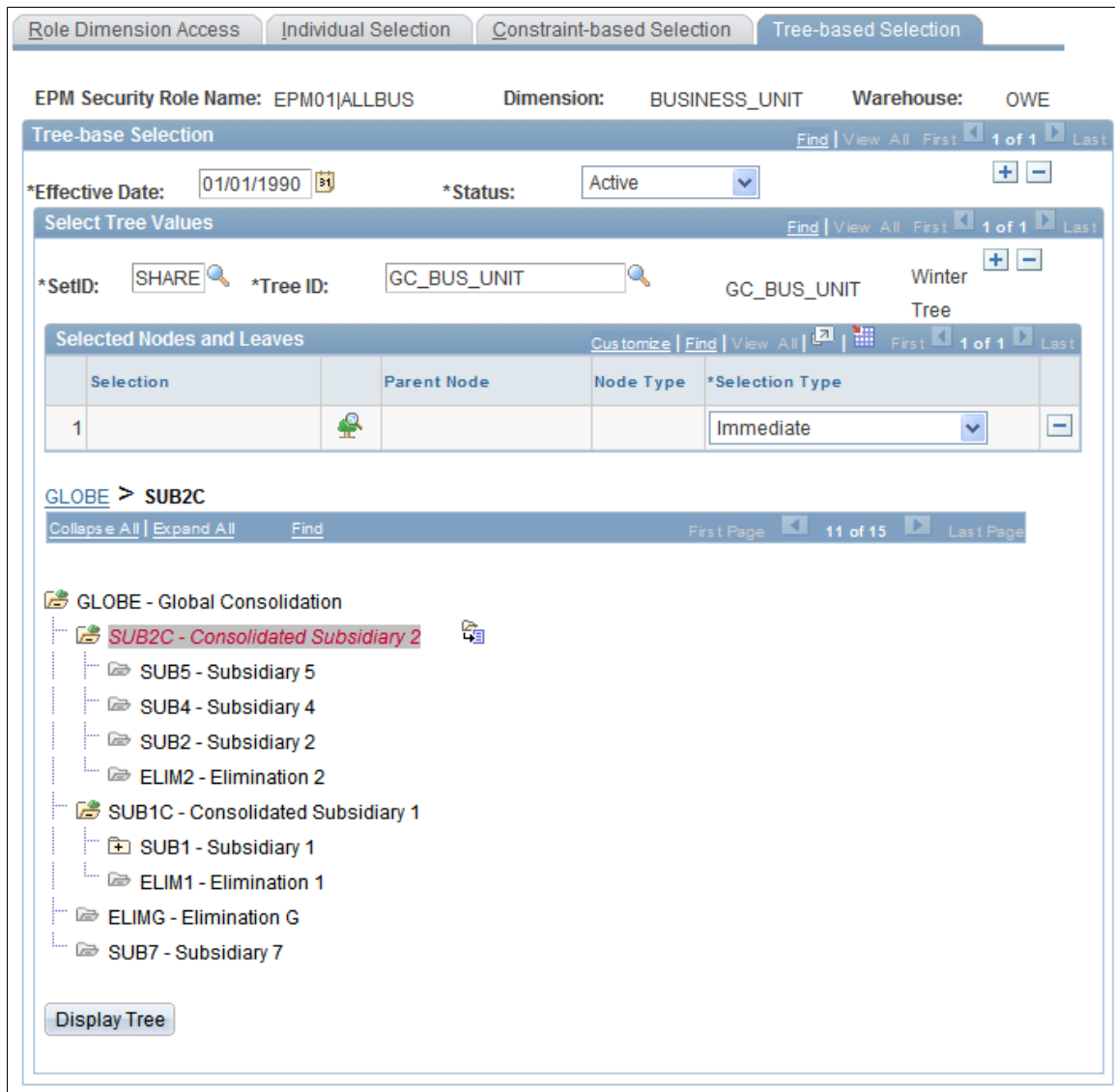
You can use the tree to grant a user access to specific nodes, leaves, or details in the tree. Used with OWE dimensions only.

Navigation

EPM Foundation, EPM Security, Security By Role, Role Dimension Access, Tree-based Selection

Image: Tree-based Selection page

This example illustrates the fields and controls on the Tree-based Selection page. You can find definitions for the fields and controls later on this page.



Dimension

Displays the dimension that you are associating with a particular security role.

Warehouse

Displays the warehouse layer that is associated with the selected dimension.

Select Tree Values

SetID

Enter the SetID that is associated with the dimension rows that you want to secure.

Tree ID	Enter the hierarchy tree that you want to use to specify the dimension rows.
Selection	Displays the selected tree node value.
Find Selected Value	Click the Find Selected Value button to display the selected node at the top of the hierarchy tree and make it easier for you to locate the node with which you are working.
Parent Node	Displays the parent node of the selected node. This field is blank if the selected node is a root node.
Node Type	Displays the node type of the selected node. Values can be <i>Node</i> or <i>Detail</i> .
Selection Type	Specify the level of detail to include with the selected node. Different values are available for your selection, depending on whether you have selected a node or a leaf from the hierarchy tree. If a node is selected, you can specify <i>This Node Only</i> , <i>Immediate Children</i> , <i>Node and Immediate Children</i> , <i>All Descendants</i> , or <i>Node + All Descendants</i> . If a leaf is selected, you can specify <i>Immediate Child Leaves</i> or <i>All Descendant Leaves</i> . If the leaf has a range of values, you must select <i>Immediate Child Leaves</i> . Trees with duplicate leaves are not supported.
Add to Node Selection List	Click the Add to Node Selection List button to add the selected node to the selection list. You must add a node to the selection list before the fields in the Selected Nodes and Leaves group box displays node values.
Display Tree	Click to display the hierarchy tree.

User Role Access Pge

Use the User Role Access page (PF_SY_USER_ROLES) to assign a user to a security role.

Navigation

EPM Foundation, EPM Security, Security By User, User Role Access

Image: User Role Access page

This example illustrates the fields and controls on the User Role Access page. You can find definitions for the fields and controls later on this page.

User Role Access

User ID: EXEC1

Roles					
	EPM Security Role Name		Role Type	Description	
1	SHARE PL_ACCNODE		EPM Role	PL account nodes	+ -
2	SHARE PL_ALL_CHANNELS		EPM Role	All Planning Channels	+ -
3	SHARE PL_ALL_CUSTOMER		EPM Role	All Planning Customers	+ -
4	SHARE PL_ALL_DEPTS		EPM Role	All Planning Departments	+ -
5	SHARE PL_ALL_OPERUNIT		EPM Role	All Planning Operating Units	+ -
6	SHARE PL_ALL_PRODS		EPM Role	All Planning Products	+ -
7	SHARE PL_BU_GRP1		EPM Role	Planning BU Group 1	+ -
8	SHARE PL_JOBCODE		EPM Role	Planning Jobcode for Headcount	+ -
9	SHARE PL_SCENARIO		EPM Role	Planning Scenarios	+ -

- User ID** Displays the user for whom you are granting role access.
- EPM Security Role Name** Enter the security role that you want to associate with the selected user.

Role Security Summary Page

Use the Role Security Summary page (PF_SY_ROL_SUMMARY) to review access privileges and other information for a security role.

Navigation

EPM Foundation, EPM Security, Security By Role, Role Security Summary

Image: Role Security Summary page

This example illustrates the fields and controls on the Role Security Summary page. You can find definitions for the fields and controls later on this page.

Role Security Summary

EPM Security Role Name:

Dimension			
Dimension Name	Warehouse	Edit Access	Details
1 BUSINESS_UNIT	OWE	Edit Access	Details

[Add Dimension Access](#)

Metrics

[Edit Metric Access](#)

Note: Changes will not take effect until security is rebuilt

EPM Security Role Name

Enter the security role for which you want to see a summary of access privileges.

Display Summary

Click to display the security role details and refresh the view.

Dimension

Dimension Name

Displays the dimensions that are associated with the selected security role.

Warehouse

Displays the warehouse layer that is associated with the selected dimension.

Edit Access

Click to access the Role Dimension Access page and edit the security role's access to the dimension.

Details

Click to access the Review Role Dimension Access page and examine additional details about the secured dimension, such as the SetID or dimensionID.

Add Dimension Access

Click to access the Role Dimension Access page and grant the selected security role access to another dimension.

Metrics

Click the Edit Metric Access link to access the Role Metric Access page and edit the security role's access to a metric.

Review Role Dimension Access Page

Use the Review Role Dimension Access page (PF_SY_ROLE_DETAIL) to .

Navigation

Click the *Details* link on the Role Security Summary page.

Image: Review Role Dimension Access page

This example illustrates the fields and controls on the Review Role Dimension Access page. You can find definitions for the fields and controls later on this page.

Review Role Dimension Access	
EPM Security Role Name: EPM01 CITY_BU Dimension: BUSINESS_UNIT Warehouse: OWE	
Access via Roles Customize Find View All First 1 of 1 Last	
Business Unit	Last Update Date/Time
1 EPM00	09/13/2004 1:47:28PM
Return	

Use this page to review additional details about your dimension that is associated with a particular security role.

Access to Metric Page

Use the Access To Metric page (PF_SY_ROLE_METR) to assign a metric to a user.

Navigation

EPM Foundation, EPM Security, Security By User, User Metric Access, Access To Metric

The fields on this page are identical to the fields on the Access to Metric page for security roles. The only difference is that the fields on this page represent individual user access privileges and not a security role.

User Dimension Access Page

Use the User Dimension Access page (PF_SY_ROLE_ALL) to assign a dimension table to a user and specify high-level access privileges.

Navigation

EPM Foundation, EPM Security, Security By User, User Dimension Access, User Dimension Access

The fields on this page are identical to the fields on the Role Dimension Access page. The only difference is that the fields on this page represent individual user access privileges and not a security role.

Individual Selection Page

Use the Individual Selection page (PF_SY_ROLE_LIST) to specify row-level access to the dimension based on SetID and dimension key fields.

Navigation

EPM Foundation, EPM Security, Security By User, User Dimension Access, Individual Selection

The fields on this page are identical to the fields on the Individual Selection page for security roles. The only difference is that the fields on this page represent individual user access privileges and not a security role.

Constraint-Based Selection Page

Use the Constraint-based Selection page (PF_SY_ROLE_CONS) to specify row-level access to the dimension based on constraint.

Navigation

EPM Foundation, EPM Security, Security By User, User Dimension Access, Constraint-based Selection

The fields on this page are identical to the fields on the Constraint-based Selection page for security roles. The only difference is that the fields on this page represent individual user access privileges and not a security role.

Tree-Based Selection Page (OWE Dimension Only)

Use the Tree-based Selection page (PF_SY_ROLE_TREE) to specify row-level access to the dimension based on an existing tree hierarchy that is defined for the dimension.

You can use the tree to grant a user access to specific nodes, leaves, or details in the tree. Used with OWE dimensions only.

Navigation

EPM Foundation, EPM Security, Security By User, User Dimension Access, Tree-based Selection

The fields on this page are identical to the fields on the Tree-based Selection page for security roles. The only difference is that the fields on this page represent individual user access privileges and not a security role.

User Security Summary Page

Use the User Security Summary page (PF_SY_SUMMARY) to review access privileges and other information for a user.

Navigation

EPM Foundation, EPM Security, Security By User, User Security Summary

Image: User Security Summary page

This example illustrates the fields and controls on the User Security Summary page. You can find definitions for the fields and controls later on this page.

User Security Summary

User ID: Display Summary

Dimension Customize Find View All First 1-3 of 3 Last					
	Dimension Name	Warehouse	Access Type	Edit Access	Details
1	ACCOUNT	MDW	User	Edit Access	Details
2	ACCOUNT	OWE	User	Edit Access	Details
3	DEPARTMENT	OWE	User	Edit Access	Details

[Add Dimension Access](#)

Metrics

[Edit Metric Access](#)

Roles Customize Find View All First 1-4 of 4 Last		
	EPM Security Role Name	Description
1	Compensation Administrator	Compensation Administrator
2	EOPP_USER	Common Portal User
3	PAPP_USER	Enterprise Portal User
4	PeopleSoft User	PeopleSoft User

[Edit Role Assignments](#)

Note: Changes will not take effect until security is rebuilt

User ID Enter the user for which you want to see a summary of access privileges.

Display Summary Click to display security details for the user and refresh the view.

Dimension

Dimension Name Displays the dimensions that are associated with the selected user.

Warehouse	Displays the warehouse layer that is associated with the selected dimension.
Edit Access	Click to access the User Dimension Access page and edit the user's access privileges to the dimension.
Details	Click to access the Review User Dimension Access page and examine additional details about the secured dimension, such as the SetID or dimensionID.
Add Dimension Access	Click to access the User Dimension Access page and grant the selected user access to another dimension.

Metrics

Click the Edit Metric Access link to access the User Metric Access page and edit the user's access to a metric.

Roles

EPM Security Role Name	Displays the security roles that are associated with selected user.
Edit Role Assignments	Click to access the User Role Access page and edit the user's privileges that are associated with the role.

Review User Dimension or User Metric Access Summary

Use the Review User Dimension Access page (PF_SY_USER_DETAIL) to review additional details about a dimension that is associated with the selected user.

Navigation

Click the *Details* link on the User Security Summary page.

The fields on this page are identical to the fields on the Review Role Dimension Access page. The only difference is that the fields on this page represent individual user access privileges and not a security role.

Request Security Processing Page

Use the Request Security Processing page (RUN_PF_SECURITY) to apply security parameters for security roles and users by running the request security processing (PF_SECURITY) process.

Navigation

EPM Foundation, EPM Security, Advanced, Request Security Processing

Image: Request Security Processing page

This example illustrates the fields and controls on the Request Security Processing page. You can find definitions for the fields and controls later on this page.

The screenshot shows the 'Request Security Processing' page with the following fields and controls:

- User ID:** VP1
- Run Control ID:** DVP1
- Process Information:**
 - *Description:** Security - Warehouse
 - When:** Always (dropdown)
 - Send Email Notification
 - [Specify Email Parameters](#)
 - EPM Role:** Analyst
 - User ID:** AASH
 - Dimension:** ACCOUNT
 - Warehouse:** MDW
 - *As Of Date:** 04/01/2010
 - Enter Business Unit to use for RecordSuite allocation:
 - *Business Unit:** CORP1
 - Rerun Option
 - *Jobstream ID:** SECURITY
 - Program:** PF_JOBSTREAM
 - As Of Date:**
- Buttons:** Report Manager, Process Monitor, Run
- Dropdown:** Copy System Role to EPM
 - Rebuild Security Only
 - Copy Roles, Rebuild Security
 - Copy Roles Only

EPM Role

Enter the EPM security role that you want to process.

If you leave this field blank, all security roles are processed.

Note: You cannot process a security role and a user at the same time.

User ID

Enter the user you want to process.

If you leave this field blank, all users are processed.

Note: You cannot process a security role and a user at the same time.

Dimension

Enter the dimension that you want to process.

If you leave this field blank, all dimensions are processed.

Warehouse

Enter the warehouse structure that is associated with the dimension you select for processing.

Business Unit

Enter the business unit that you want to process.

Business unit is used to determine which record suite is used for the security job.

Jobstream ID

Enter the Jobstream ID for the warehouse security.

Note: This jobstream is not secured, all users can access and run it. However, only an administrator should run this jobstream.

Rerun Option	Select this check box to rerun the security parameters process.
Copy System Role to EPM	
Rebuild Security Only	Select this option if you want only to rebuild the security join tables.
Copy Roles, Rebuild Security	Select this option if you want to rebuild the security join tables and import PeopleTools security roles into the EPM database.
Copy Roles Only	Select this option if you want only to import PeopleTools security roles into the EPM database.

Establishing Security Bridges for Your Business Intelligence Tool

This topic provides an overview of security bridges and discusses PeopleSoft-delivered Application Programming Interfaces (APIs).

Role Collection

This class is a collection class of individual Role classes. The collection will inherit the normal Java methods to traverse the collection, for example, next(), first() and so forth. Also methods such as add(), remove() can be used and implemented as an ArrayList.

Public Methods

Name	RoleCollection(JDBCConnection dbCon)
Returns	Nothing
Parameters	JDBCConnection dbCon. A valid JDBC Connection.
Description	This is the constructor used to instantiate the Role collection.
Name	LoadAllRoles()
Returns	Nothing
Parameters	None
Description	This method clears out the current elements in the collection and load it with all the currently defined Roles in EPM.

Role

This class defines a single Role defined in the EPM row-level security framework.

Public Methods

Name	Role(JDBCConnection dbCon, String Name);
Returns	Nothing
Parameters	JDBCConnection dbCon. A valid JDBC Connection.
Description	The constructor for the Role class. Takes JDBC Connection and RoleName as parameters. Given a rolename, the constructor will fill up the other properties such as Type and Description.
<hr/>	
Note: There is no public constructor of the format Role(), you cannot instantiate this class without a name.	
<hr/>	
Name	getUsers();
Returns	List of UserNames
Parameters	None
Description	The method will return a List of User Names that belong to this Role. A standard Java List object can be used to implement this.
Name	getAccessibleDimensions();
Returns	List of Dimension Names
Parameters	None
Description	The method will return a List Dimension names that this Role can access.
Name	hasAccess (String DimName);
Returns	integer 0 or 1
Parameters	Dimension Name
Description	The method returns a Boolean specifying whether the Dimension referenced by DimensionName parameter is available to the Role or not.
Name	hasAccess(String DimName, int DimValue);
Returns	integer 0 or 1
Parameters	Dimension Name, A surrogate key value
Description	The method returns 0 or 1 specifying whether this Role has access to a particular value in a dimension.
Name	hasAllAccess(String DimName);
Returns	integer 0 or 1
Parameters	Dimension Name

Description	The method returns a Boolean specifying whether this Role has ALL access to the Dimension referenced by DimensionName parameter.
Name	getDimensionValues(String DimName);
Returns	List of Values
Parameters	Dimension Name
Description	The method returns a List of values in the Dimension referenced by DimName that are available to the Role. The method will first check to ensure that the Dimension Name is a valid Secured dimension in the system, and it is accessible to the Role. The method will return a List of values from the appropriate security join table. If the Role has ALL access to this dimension, the List will contain only one value with the pre-determined surrogate key value for ALL. Will return an empty List if the dimension is not available to the Role.
Name	getName();
Returns	String Name
Parameters	None
Description	Get method for Name
Name	getType();
Returns	String Name
Parameters	None
Description	Get method for Type
Name	getDescription();
Returns	String Descr
Parameters	None
Description	Get method for Description
Name	getLongDescription();
Returns	String Description
Parameters	None
Description	Getter method for Long Description

User Collection

This class contains the same information as the Role Collection class. The only difference is that the values for this class represent user collection and not role collection.

User

This class contains information about a single User defined in EPM Security.

Public Methods

Name	User(JDBCConnection dbCon, String Name);
Returns	Nothing
Parameters	JDBC Connection, String UserName
Description	The constructor for the User class. Takes Name as the one and only parameter.
Name	getRoles();
Returns	List of RoleNames
Parameters	None
Description	The method will instantiate a List of Role Names that contain this User.
Name	getAccessibleDimensions();
Returns	List of Dimension Names
Parameters	None
Description	The method will return a List of Dimension names that this User can access. This will be a combined list of all dimensions that the user can access via all his roles.
Name	hasAccess (String DimName);
	getAccessibleMetrics()
Returns	integer 0 or 1
Parameters	Dimension Name
Description	The method returns a Boolean specifying whether the Dimension referenced by DimensionName parameter is available to the User or not. This method queries all the Roles for this user to determine if the user has access to the dimension.
Name	hasAccess(String DimName, int DimValue);
Returns	integer 0 or 1
Parameters	Dimension Name, A surrogate key value
Description	The method returns 0 or 1 specifying whether this User has access to a particular value in a dimension. This method queries

all the Roles for this user to determine if the user has access to the dimension and value.

Name hasAllAccess(String DimName);

Returns integer 0 or 1

Parameters Dimension Name

Description The method returns a Boolean specifying whether this User has ALL access to the Dimension referenced by DimensionName parameter. If any of the user's Roles has ALL access, the user is deemed to have ALL access.

Name getDimensionValues(String DimName);

Returns List of Values

Parameters Dimension Name

Description The method returns a List of values in the Dimension referenced by DimName that are available to the User. The method first checks to ensure that the Dimension Name is a valid Secured dimension in the system, and it is accessible to the User. This is a combined list of values from all the user's Roles. The method returns a List of values from the appropriate SJT. If the User has "all" access to this dimension, the List will contain only one value with the pre-determined surrogate key value for "all". Will return an empty List if the dimension is not available to the User.

Name getName();

Returns String Name

Parameters None

Description Get method for Name

Dimension Collection

This class contains information about the list of dimensions defined as secured dimensions in EPM Security. Each element of the collection is a Dimension class that represents one secured Dimension in the EPM system. The collection will inherit the normal Java methods to traverse the collection, for example, next(), first() and so forth. Also methods such as add(), remove() can be used.

Public Methods

Name DimensionCollection(JDBCConnection dbCon)

Returns Nothing

Parameters JDBC Connection

Description	This is the constructor used to instantiate the Dimension collection. This class is derived from one of the Java Collection classes, possibly ArrayList, or LinkedList. Most of the methods needed are inherited from the Collection class, Next(), Previous(), HasNext(), size() and so forth. The constructor will instantiate an empty collection.
Name	LoadAllDimensions()
Returns	Nothing
Parameters	None
Description	This method will clear out the current elements in the collection and load it with all the currently secured dimensions in EPM.

Dimension

This class contains information about a single Dimension defined in the EPM database.

Public Methods

Name	Dimension (JDBC Connection, String Name)
Returns	String Name
Parameters	JDBC Connection
Description	This is the constructor for this class. Takes a Dimension Name as parameter. The constructor will validate that the dimension specified by Name is a valid secured dimension in EPM. If it is an invalid name, or the dimension is not secured, the constructor will fail.
Name	getRoles()
Returns	List of Role Names
Parameters	Nothing
Description	This method will return a list of all Role names that have access to this dimension.
Name	getUsers()
Returns	List of User Names
Parameters	Nothing
Description	This method will return a list of all User names that have access to this dimension.
Name	getDimName()

Returns	String Name
Parameters	None
Description	Get method for Name.
Name	getDimTableName()
Returns	String TableName
Parameters	None
Description	Get method for Table Name.
Name	getDimKeyName()
Returns	String KeyName
Parameters	None
Description	Get method for Key Name.
Name	isSecured()
Returns	Boolean
Parameters	None
Description	Get method for isSecured.

MetricCollection

This class contains information about metric collection.

Public Methods

Name	MetricCollection(JDBCCConnection dbCon)
Returns	Nothing
Parameters	A valid JDBCCConnection
Description	This is the constructor used to instantiate the Metric collection. The constructor will instantiate an empty collection.
Name	LoadAllMetrics()
Returns	Nothing
Parameters	None
Description	This method will clear out the current elements in the collection and load it with all the currently secured metrics in EPM.

Metric

This class contains information about the metrics (columns) setup in EPM Security. This is for the Column-level security.

Public Methods

Name	Metric(JDBCConnection dbCon , String Name)
Returns	Nothing
Parameters	JDBCConnection dbCon , String Name
Description	This is the constructor for this class. Takes a Dimension Name as parameter. There is no other public constructor. You need at least a Dimension Name to instantiate this class. The constructor will validate that the dimension specified by Name is a valid secured dimension in EPM. If it is an invalid name, or the dimension is not secured, the constructor will fail.
Name	getColName()
Returns	String Column Name
Parameters	None.
Description	This method will return the column name that the metric defines.
Name	getRecName()
Returns	String Record Name
Parameters	None
Description	This method will return the Record name that the metric defines.

JDBCConnection

This class is used to connect to a database using a JDBC driver.

Public Methods

Name	makeConnection()
Returns	Nothing
Parameters	None
Description	This method will connect to a JDBC database. The parameters needed to connect will be read from jdbcconnection.property file.
Name	makeConnection(String theDriverName, String theDbURL)

Returns	Nothing
Parameters	Driver Name, DB URL string
Description	This method will connect to a JDBC database. The parameters needed to connect are passed into the method.
Name	makeConnection(String theDriverName, String theDbURL, String theUserName, String thePassword)
Returns	Nothing
Parameters	JDBC Driver Name to be used to attempt the connection, URL for database to connect, User Name, Password
Description	This method will connect to a JDBC database. The parameters needed to connect are passed into the method.
Name	closeConnection()
Returns	Nothing
Parameters	None
Description	This method will close the connection.
Name	getDbResultSet(String pTheQuery)
Returns	ResultSet
Parameters	Query sent to the database.
Description	This method will execute the query and return a ResultSet object.

Preparing to Load Source Data Into EPM

Understanding ETL in EPM

This topic discusses:

- ETL and the EPM Architecture.
- Data Flow through EPM.
- IBM WebSphere DataStage.

ETL and the EPM Architecture

The PeopleSoft delivered ETL process enables you to extract data from disparate source transaction systems, integrate the data in a single EPM database, transform and enrich the data, and load it into specific EPM data models that are optimized for analysis and reporting. This process is facilitated by the best-in-class data integration platform *IBM WebSphere DataStage* and PeopleSoft delivered ETL *jobs*.

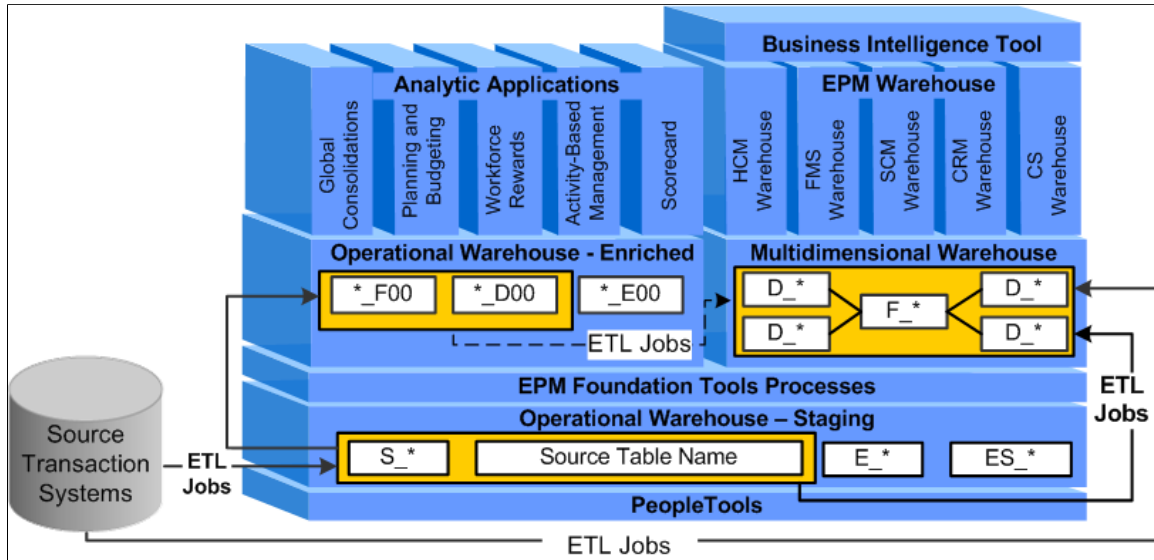
The ETL process migrates data across all layers of EPM warehouse structures and consists of two load types:

- *Stage I Load*: Consists of all ETL jobs that extract data from your source transaction system and load it into Operational Warehouse - Staging (OWS) tables. Also included in this type of load (but less common) are ETL jobs that extract data from your source transaction system and load it directly into Multidimensional Warehouse (MDW) tables.
- *Stage II Load*: Consists of all ETL jobs that extract data from the OWS tables and load it into the Operational Warehouse - Enriched (OWE) or the Multidimensional Warehouse (MDW) tables. Also

included in this type of load (but less common) are ETL jobs that extract data from the OWE and load it into the MDW.

Image: ETL in EPM

The following diagram depicts the flow of data through each layer of the EPM architecture using ETL.



After your data is extracted from the OWS it is loaded into specialized data models (target warehouse tables designed to aggregate or enrich your data), which are used by the Analytical Applications and EPM Warehouses for reporting and analysis.

Understanding the Flow of Data Through EPM

Each EPM data warehouse requires a unique set of ETL jobs to populate corresponding target tables with data. Data warehouse target tables may have missing or inaccurate data in them if you do not run all applicable jobs in the proper sequence.

The following topics provide an overview of the ETL jobs required to populate each data warehouse layer with data.

Moving Data Into the OWS

You use ETL jobs to move data into the OWS from your PeopleSoft source system. The following is an overview of the steps required to bring data into the OWS:

1. Run initial setup (OWS) jobs.
2. Run source business unit extract jobs.
3. Run shared lookup jobs.
4. Run CSW OWS jobs (for CSW Warehouse implementation only).
 Run CRM OWS jobs (for CRM Warehouse implementation only).
 Run FMS OWS jobs (for FMS Warehouse implementation only).
 Run HCM OWS jobs (for HCM Warehouse implementation only).

Run SCM OWS jobs (for SCM Warehouse implementation only).

Moving Data Into the OWE

You use ETL jobs to move data into the OWE from the OWS. The following is an overview of the steps required to bring data into the OWE:

1. Run the setup - OWE jobs.
2. Run common dimension jobs.
3. Some EPM warehouses require OWE data.

For these warehouses see steps below in, 'Moving Data Into the MDW.'

Moving Data Into the MDW

There are three methods of bringing data into the MDW:

- Extracting data from the OWS and moving it into the MDW.

This is the most common method and the majority of your data is moved into the MDW in this way.

- Extracting data from the OWE and moving it into the MDW.

Certain EPM warehouses use this method, which brings enriched, business unit-based data into the MDW.

For example, the Profitability data mart in the FMS Warehouses uses OWE data that is output from the Global Consolidations analytical application.

- Extracting source data directly from a PeopleSoft source system and moving it into the MDW.

This method bypasses the OWS and is only used when large volumes of data must be extracted, such as data used for the Marketing data mart in the CRM Warehouse.

You use ETL jobs to move data into the MDW. The following is an overview of the steps required to bring data into the MDW:

1. Run Global Dimension Jobs for Campus Solutions Warehouse

Run Global Dimension Jobs for CRM Warehouse

Run Global Dimension Jobs for FMS Warehouse

Run Global Dimension Jobs for HCM Warehouse

Run Global Dimension Jobs for SCM Warehouse

2. Run Local Dimension Jobs for Campus Solutions Warehouse

Run Local Dimension Jobs for CRM Warehouse

Run Local Dimension Jobs for FMS Warehouse

Run Local Dimension Jobs for HCM Warehouse

- Run Local Dimension Jobs for SCM Warehouse
- 3. Run CSW SKU Jobs
 - Run CRM SKU Jobs
 - Run FMS SKU Jobs
 - Run HCM SKU Jobs
 - Run SCM SKU Jobs
- 4. Run Global-OWE Jobs for CRM Warehouse
 - Run Global-OWE Jobs for FMS Warehouse
 - Run Global-OWE Jobs for HCM Warehouse
 - Run Global-OWE Jobs for SCM Warehouse
- 5. Run CRM-OWE jobs
 - Run FMS-OWE jobs
 - Run HCM-OWE jobs
 - Run SCM-OWE jobs

For more information on the jobs required to load data into the MDW for your EPM Warehouse, see your warehouse specific documentation (for example, *PeopleSoft EPM: Campus Solutions Warehouse*).

Related Links

[Overview](#)

[Preparing to Install and Implement IBM WebSphere DataStage](#)

[Understanding IBM WebSphere DataStage](#)

[Verifying ETL Components Have Imported Properly](#)

[Understanding Warehouse Business Unit Setup](#)

[Understanding Warehouse Business Units, TableSet Sharing, and SetID Mapping](#)

Understanding IBM WebSphere DataStage

PeopleSoft has an original equipment manufacturer (OEM) agreement with IBM for its WebSphere DataStage ETL tool and bundles this offering with PeopleSoft EPM. The IBM WebSphere DataStage tool uses ETL jobs to target specific data from a PeopleSoft source database and migrate it to the OWS, OWE, and MDW tables. IBM WebSphere DataStage is comprised of a server tool and client tool, which are discussed in more detail below.

IBM WebSphere DataStage provides the following features:

- Graphical design tools for designing ETL maps (called *jobs*)
- Data extraction from a variety of data sources

- Data aggregation using SQL SELECT statements
- Data conversion using predefined or user-defined transformations and functions
- Data loading using predefined or user-defined jobs

IBM WebSphere DataStage Terminology

You should be familiar with these IBM WebSphere DataStage terms:

Term	Definition
Administrators	Administrators maintain and configure DataStage projects.
Aggregator Stages	Aggregator stages compute totals or other functions of sets of data.
Data Elements	Data elements specify the type of data in a column and how the data is converted.
Container Stages	Container stages group reusable stages and links in a job design.
DataStage Package Installer	This tool enables you to install packaged DataStage jobs and plug-ins.
Hashed File	A hashed file groups one or more related files plus a file dictionary. DataStage creates hashed files when you run a job that creates hash files (these are delivered with PeopleSoft EPM). Hashed files are useful for storing data from tables from a remote database if they are queried frequently, for instance, as a lookup table.
Hashed File Stage	A hashed file stage extracts data from or loads data into a database containing hashed files. You can also use hashed file stages as lookups. PeopleSoft ETL jobs use hashed files as lookups.
Inter-process Stage	An inter-process stage allows you to run server jobs in parallel on a symmetric multiprocessing system.
Plug-in Stages	Plug-in stages perform processing that is not supported by the standard server job stage.
Sequential File Stage	A sequential file stage extracts data from or writes data to a text file.
Transform Function	A transform function takes one value and computes another value from it.
Transformer Stages	Transformer stages handle data, perform any conversions required, and pass data to another stage.
Job	A job is a collection of linked stages, data elements, and transforms that define how to extract, cleanse, transform, integrate, and load data into a target database. Jobs can either be server or mainframe jobs.

Term	Definition
Job Sequence	Job sequence invokes and runs other jobs.
Join Stages	Join stages are mainframe processing stages or parallel job active stages that join two input sources.
Metadata	Metadata is data about data; for example, a table definition describing columns in which data is structured.

DataStage Server

The IBM WebSphere DataStage server enables you to schedule and run your ETL jobs:

Image: DataStage Sever

This diagram illustrates the DataStage Server



Three components comprise the DataStage server:

- Repository
The Repository stores all the information required for building and running an ETL job.
- DataStage Server
The DataStage Server runs jobs that extract, transform, and load data into the warehouse.
- DataStage Package Installer
The DataStage Package Installer installs packaged jobs and plug-ins.

DataStage Client

The IBM WebSphere DataStage client enables you to administer projects, edit repository contents, and create, edit, schedule, run, and monitor ETL jobs.

Three components comprise the DataStage client:

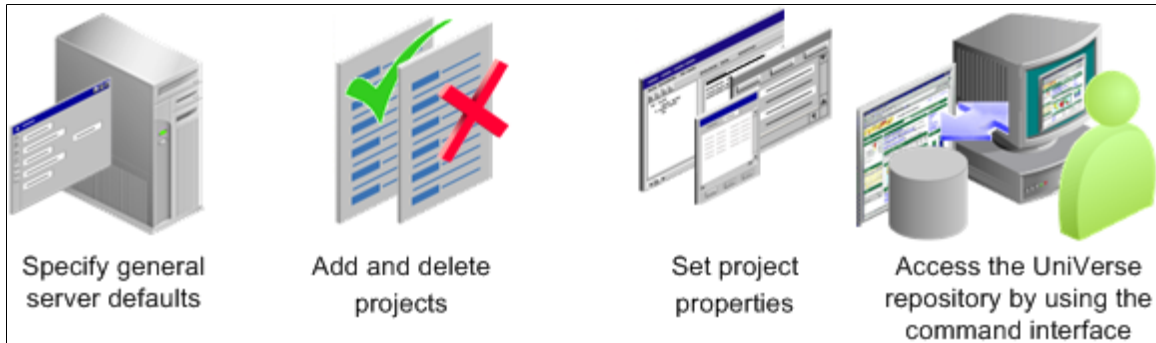
- DataStage Administrator
- DataStage Designer
- DataStage Director

DataStage Administrator

The DataStage Administrator enables you to:

Image: DataStage Administrator

This diagram illustrates the tasks that can be performed by the DataStage Administrator.



See [Setting DataStage Server Properties](#) and [Setting Project Properties](#).

DataStage Designer

DataStage Designer enables you to:

- Create, edit, and view objects in the metadata repository.
- Create, edit, and view data elements, table definitions, transforms, and routines.
- Import and export DataStage components, such as projects, jobs, and job components.
- Create ETL jobs, job sequences, containers, routines, and job templates.
- Create and use parameters within jobs.
- Insert and link stages into jobs.
- Set stage and job properties.
- Load and save table definitions.
- Save, compile, and run jobs.

See [DataStage Designer Overview](#).

DataStage Director

DataStage Director enables you to:

- Validate jobs.
- Schedule jobs.
- Run jobs.
- Monitor jobs.

- View log entries and job statistics.

See [DataStage Director Overview](#).

Key DataStage Components

IBM WebSphere DataStage contains many different components that support the ETL process. Some of these components include stages, jobs, and parameters. Only the following key DataStage components are discussed in this topic:

- DSX Files
- Jobs
- Hashed Files
- Environmental Parameters
- Shared Containers
- Routines

A complete list of all DataStage components can be found in the *WebSphere DataStage Development: Designer Client Guide*.

DSX Files

PeopleSoft delivers a *.dsx file for each functional area within EPM. As part of your installation and configuration process you import the *.dsx file into a project that has been defined in your development environment. Included in the *.dsx file are various DataStage objects that define your project. The *.dsx files are organized by functional area and contain related ETL jobs.

Note that *.dsx files use the following naming convention:

- `<file_name>.dsx`, for DataStage 8.1 customers.
For example, `Common_Uilities.dsx`.
- `<file_name>_85.dsx`, for DataStage 8.5 customers.
For example, `Common_Uilities_85.dsx`.

To see a list of the PeopleSoft-delivered *.dsx files, refer to the file "DSX Files Import Description.xls" located in the following install CD directory path: `<PSHOME>\SRC\ETL`.

Each delivered *.dsx file contains the DataStage objects described in the following topics.

ETL Jobs

PeopleSoft delivers predefined ETL jobs for use with IBM WebSphere DataStage. ETL Jobs are a collection of linked stages, data elements, and transformations that define how to extract, transform, and load data into a target database. Stages are used to transform or aggregate data, and lookup information. More simply, ETL jobs extract data from source tables, process it, then write the data to target warehouse tables.

PeopleSoft deliver five types of jobs that perform different functions depending on the data being processed, and the warehouse layer in which it is being processed:

Load Stage	Type	Description
I	Source to OWS	<p>Jobs in this category extract data from your PeopleSoft transaction system and populate target warehouse tables in the OWS layer of the warehouse.</p> <p>Source to OWS jobs assign a source system ID (SRC_SYS_ID) for the transaction system from which you are extracting data and populate the target OWS tables with that ID.</p>
I	Source to MDW	<p>Jobs in this category extract data from your transaction system and populate target dimension and fact tables in the MDW layer of the warehouse. The Online Marketing data mart is the only product to use this type of job.</p>
II	OWS to OWE	<p>Jobs in this category extract data from the OWS tables and populate target D00, F00, and base tables in the OWE layer of the warehouse.</p> <p>OWS to OWE jobs perform lookup validations for the target OWE tables to ensure there are no information gaps and maintain referential integrity. Many of the jobs aggregate your transaction data for the target F00 tables.</p>
II	OWS to MDW	<p>Jobs in this category extract data from the OWS tables and populate target DIM and FACT tables in the MDW layer of the warehouse.</p> <p>OWS to MDW jobs generate a surrogate key that helps facilitate dimension key resolution. The surrogate key value is used as the primary key in the target DIM table and as the foreign key in the FACT table. The jobs also perform lookup validations for the target DIM and FACT tables to ensure there are no information gaps and maintain referential integrity.</p>

Load Stage	Type	Description
II	OWE to MDW	Jobs in this category extract data from the OWE tables and populate target DIM and FACT tables in the MDW layer of the warehouse. Properties of this job type mirror those of the OWS to MDW job. OWE to MDW jobs generate a surrogate key that helps facilitate dimension key resolution. The surrogate key value is used as the primary key in the target DIM table and as the foreign key in the FACT table. The jobs also perform lookup validations for the target DIM and FACT tables to ensure there are no information gaps and maintain referential integrity.

All job types identified in the table are *incremental load* jobs. Incremental load jobs identify and extract only new or changed source records and bring it into target warehouse tables.

See [Understanding ETL Load Strategies in EPM](#).

ETL Jobs - Naming Convention

PeopleSoft use standard naming conventions for all ETL jobs; this ensures consistency across different projects. The following table provides the naming conventions for PeopleSoft delivered ETL jobs.

Object	Naming Convention	Example
Staging Server Job	J_Stage_[Staging Table Name]_[Source Release]_[EPM Release]	J_Stage_PS_AGING_TBL_FSCM91_EPM91
Sequencer Job	SEQ_[Staging Table Name]_[Source Release]_[EPM Release]	SEQ_J_Stage_PS_AGING_TBL_FSCM91_EPM91
CRC Initial Load Job	J_Hash_PS_[Staging Table Name]_[Source Release]_[EPM Release]	J_Hash_PS_AGING_TBL_FSCM91_EPM91
Common Lookup Load Job	J_Hash_PS_[Table Name]	J_Hash_PS_D_LRNG_ENV
MDW Dimension Job	J_Dim_PS_[Dimension Table Name]	J_Dim_PS_D_DEPT
MDW Fact Job	J_Fact_PS_[Fact Table Name]	J_Fact_PS_F_ENRLMT
OWE Dimension Job	J_D00_PS_[D00 Table Name without D00 Suffix]	J_D00_PS_ACCOMP_D00
OWE Fact Job	J_F00_PS_[F00 Table Name without F00 Suffix]	J_F00_PS_JOB_F00

Object	Naming Convention	Example
OWE Base Job	J_BASE_PS_[Base OWE Table Name]	J_BASE_PS_XYZ

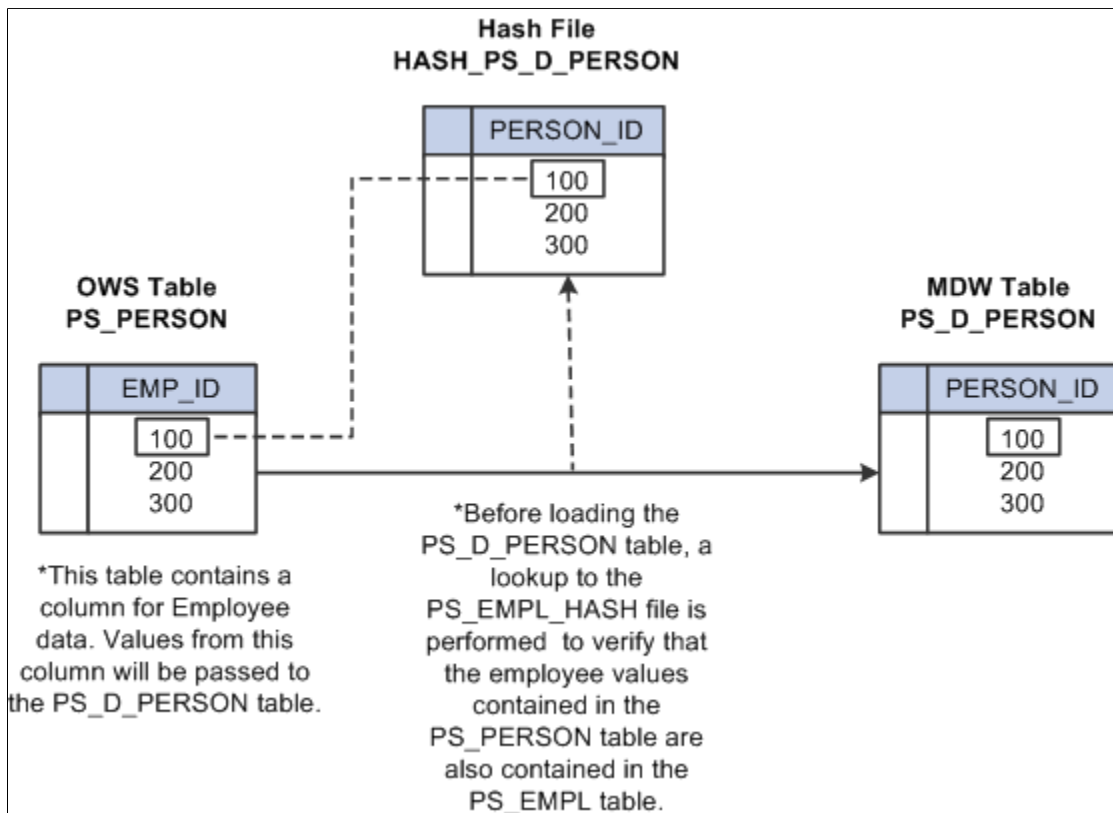
Hashed Files

Hash files are views of specific EPM warehouse tables and contain only a subset of the data available in the warehouse tables. These streamlined versions of warehouse tables are used to perform data validation (lookups) within an ETL job and select specific data from lookup tables (such as sourceID fields in dimensions).

In the validation (lookup) process the smaller hash file is accessed, rather than the base warehouse table, improving performance. The following diagram provides an example of a hash file lookup in a job.

Image: Lookup process using hash file

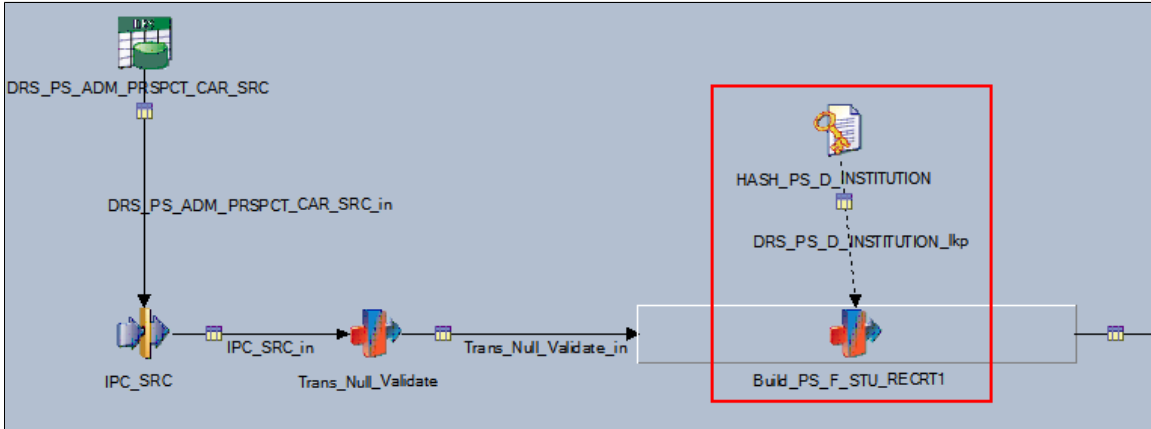
The following diagram provides an example of a hash file lookup in a job.



The following detailed view of an ETL job shows the Institution hashed file lookup in the Campus Solutions Warehouse J_Fact_PS_F_STU_RECRT job.

Image: Institution hashed file lookup in the J_Fact_PS_F_STU_RECRT job

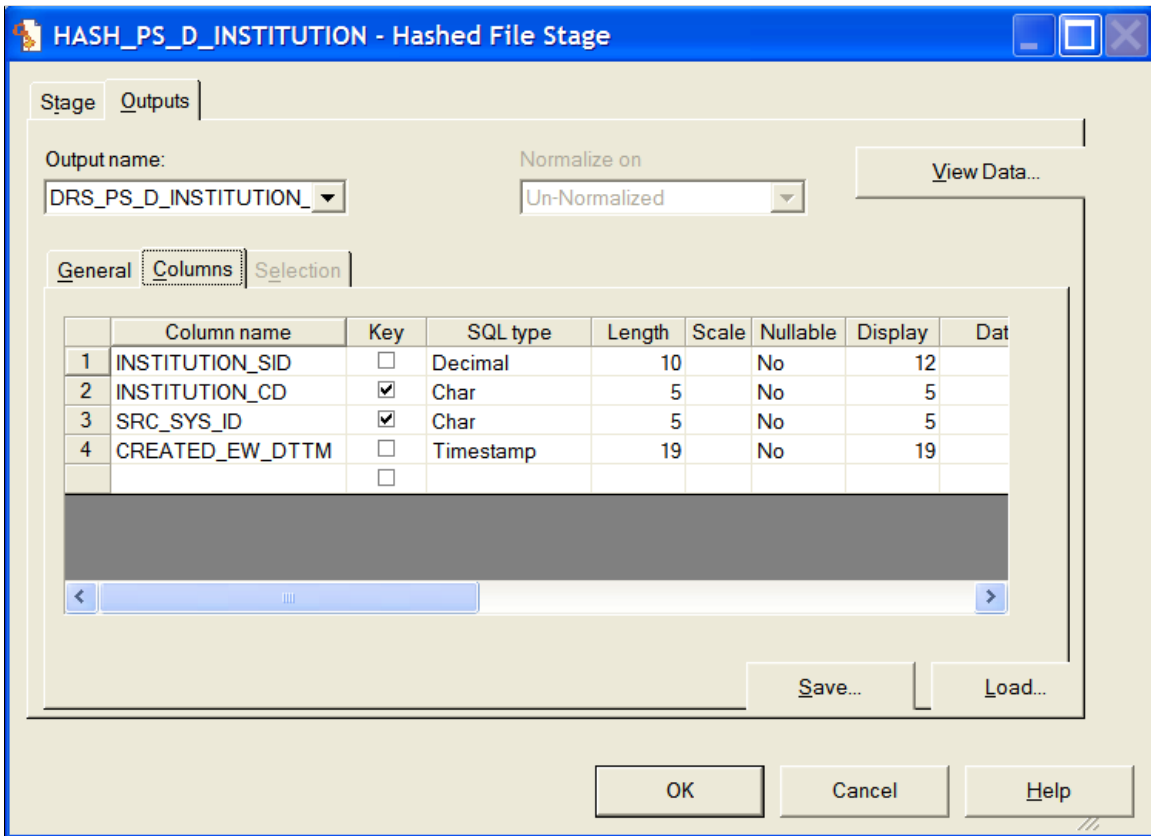
This example illustrates the Institution hashed file lookup in the J_Fact_PS_F_STU_RECRT job.



A detailed view of the hashed file stage reveals the fields (including keys) the lookup uses to validate Institution records.

Image: Hashed file stage in the J_Fact_PS_F_STU_RECRT job

This example illustrates the Hashed file stage in the J_Fact_PS_F_STU_RECRT job.

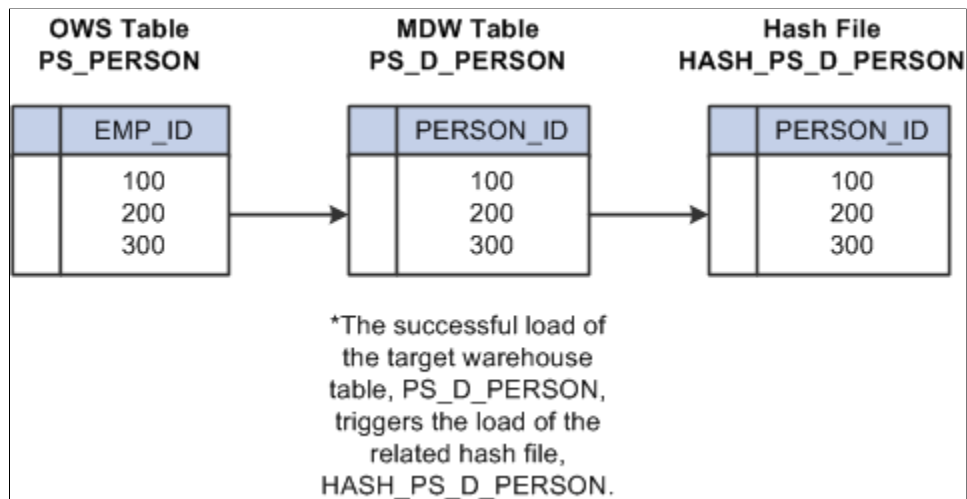


Because hash files are vital to the lookup process, jobs cannot function properly until all hash files are created and populated with data. Before you run any job that requires a hash file, you must first run all jobs that create and load the hash files—also called initial hash file load jobs.

After hash files are created and populated by the initial hash file load jobs, they are updated on a regular basis by the delivered sequencer jobs. Hash files are updated in the same job as its related target warehouse table is updated. In other words, both the target warehouse table and the related hash file are updated in the same sequencer job. The successful load of the target warehouse table in the job triggers the load of the related hash file.

Image: Hash file update process

The following diagram provides an example of the this process.



See [Understanding Data Validation and Error Handling in the ETL Process, Incremental Loading Using the Insert Flag and Lookup Validations](#).

Environmental Parameters

Environmental parameters are user-defined values that represent processing variables in your ETL jobs. Environmental parameters are reusable so they enable you to define a processing variable once and use it in several jobs. They also help standardize your jobs.

Though environmental parameters are reusable, PeopleSoft delivers specific environmental parameters for jobs related to each phase of data movement (such as the OWS to MDW jobs). Therefore, a single environmental parameter is not used across all ETL jobs, rather a subset of variables are used depending on the specific functionality of the job.

See [Environmental Parameters Information](#).

Shared Containers

Shared containers are reusable job elements. A shared container is usually comprised of groups of stages and links, and is stored in the DataStage repository. You can use shared containers to make common job components available throughout your project. Because shared containers are reusable you can define them once and use them in any number of your ETL jobs. PeopleSoft delivers the following shared containers:

- StoreLangStagingList

- StoreMaxLastUpdDttm
- StoreMaxRecordID
- StorMaxSnapDate
- StoreMaxValueDecimal
- FactStoreMaxRecordID

Routines

Routines are a set of instructions, or logic, that perform a task within a job. For example, the *ToInteger* routine converts the input value to an integer. Because routines are reusable you can use them in any number of your ETL jobs.

See [Routine Descriptions](#).

IBM Documentation

For more details on the IBM WebSphere DataStage tool and how to use it, refer to the IBM documentation listed below. You can install PDF versions of the IBM books as part of the IBM WebSphere tools install.

The following table lists the IBM documentation and the information provided.

IBM Book	Description
<i>IBM Information Server: Planning Installation and Configuration Guide</i>	Provides planning information and complete installation instructions for IBM Information Server. Also includes information about troubleshooting, validating the installation, and configuring the system.
<i>IBM Information Server: Administration Guide</i>	Describes how suite administrators can manage user access to components and features of IBM Information Server. In addition, describes how suite administrators can create and manage views of logged events and scheduled tasks for all components.
<i>WebSphere DataStage Administration: Administrator Client Guide</i>	Describes the WebSphere DataStage Administrator client and describes how to perform setup, routine housekeeping, and administration of the WebSphere DataStage engine.
<i>WebSphere DataStage Administration: Deployment Guide</i>	Describes how to package and deploy WebSphere DataStage jobs and associated objects to assist in moving projects from development to production.
<i>WebSphere DataStage Administration: Director Client Guide</i>	Describes the WebSphere DataStage Director client and explains how to validate, schedule, run, and monitor WebSphere DataStage parallel jobs and server jobs.
<i>WebSphere DataStage Administration: National Language Support Guide</i>	Describes how to use the national language support (NLS) features that are available in WebSphere DataStage when NLS is installed.

IBM Book	Description
<i>WebSphere DataStage Development: Designer Client Guide</i>	Describes the WebSphere DataStage Designer client and gives a general description of how to create, design, and develop a WebSphere DataStage application
<i>WebSphere DataStage Development: Server Job Developer Guide</i>	Describes the tools that build a server job, and supplies programming reference information

Understanding ETL Load Strategies in EPM

This topic provides an overview of ETL load strategies in EPM and discusses:

- Incremental loading using the datetime stamp
- Incremental loading using Cyclical Redundancy Check (CRC) logic
- Incremental loading using the insert flag and lookup validations

Overview of ETL Load Strategies in EPM

PeopleSoft delivers ETL jobs that extract data from your source transaction system and load it into target OWE and MDW dimension and fact tables. These jobs employ an *incremental load* strategy, which uses built-in logic to identify and load only new or updated source records. The benefit of the incremental load process is increased efficiency and faster processing during the extract and load process.

There are three types of incremental load strategies employed in PeopleSoft ETL jobs:

- Incremental loading using the datetime stamp
- Incremental loading using Cyclical Redundancy Check (CRC) logic
- Incremental loading using the insert flag and lookup validations

Note: If this is the first time you are populating your target warehouse tables with data, the incremental jobs recognize that you have no existing data in your tables and perform a complete extract of your source records. Subsequent runs of the incremental jobs will extract only new or changed records.

Incremental Loading with the DateTime Stamp

To ensure only new or changed records are extracted, EPM target tables associate a datetime stamp with each record. Please note that the datetime stamp may appear as *DTTM* or *DT_TIMESTAMP*, depending on the source from which the record originates.

When an incremental load job reads a table, it uses a built-in filter condition, `[DTTM_Column] > [%DateTimeIn('#LastModifiedDate#')]` for example, to determine whether any records in the table are new or changed since the last load. The last update date time is retrieved from the related hashed file using the `GetLastUpdDateTime` routine. If the retrieved date time is less than the current value in the *DTTM* column, the record will be updated in the EPM table. This process can be done quickly because the *DTTM* column is the only value being processed for each record.

Each time a new or updated record is loaded, the present date time stamp is recorded for the last update time stamp and is used as a basis for comparison the next time the incremental load job is run.

Note: If the last update time field is null for a record, the record is processed each time the job is executed.

Incremental Loading Using Cyclical Redundancy Check

Some source table records do not have a date timestamp column. When source table records lack a date time stamp, a *cyclical redundancy check (CRC)* must be performed to determine new or changed records. Unlike incremental loading that targets the DTTM column for each record, the CRC process must read the entire record for each record in the source table and generate a CRC value, which it uses to compare against the target warehouse record.

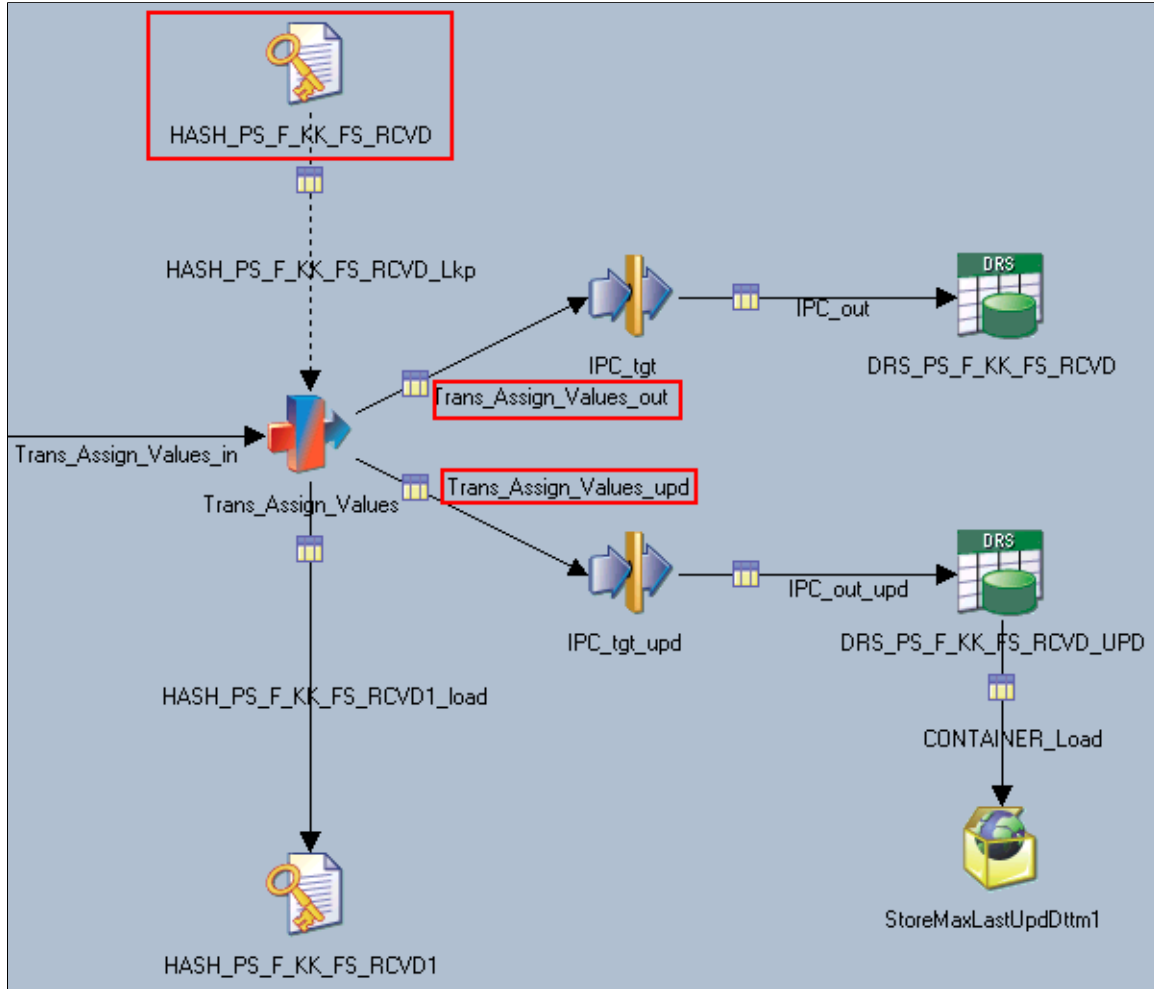
Incremental Loading Using the Insert Flag and Lookup Validations

To ensure only new or changed records are loaded to EPM target tables, some jobs use an insert flag in combination with lookup validations. The following example will illustrate this process.

In the job J_Fact_PS_F_KK_FS_RCVD, the Trans_Assign_Values transformation contains the processing logic used to load new or updated records to the target table:

Image: Trans_Assign_Values transformation

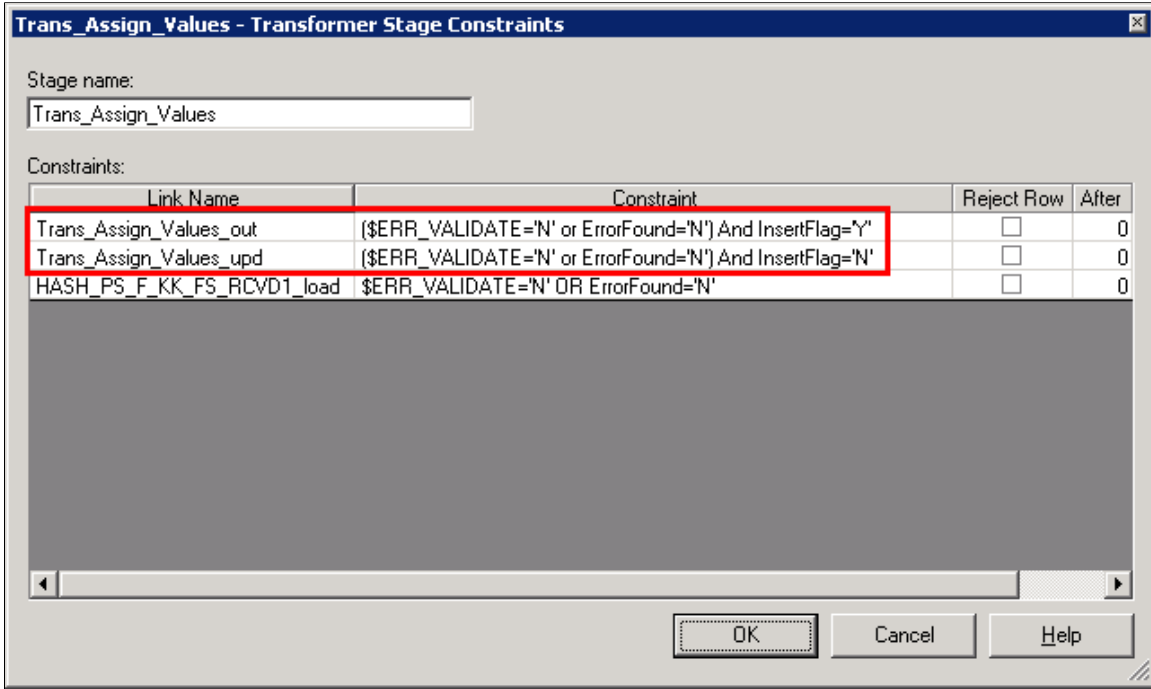
This example illustrates the Trans_Assign_Values transformation.



A closer look at the `Trans_Assign_Values` transformation shows that the `Trans_Assign_Values_out` and `Trans_Assign_Values_update` constraints are used to filter new or updated records:

Image: Trans_Assign_Values constraints

This example illustrates the `Trans_Assign_Values` constraints.



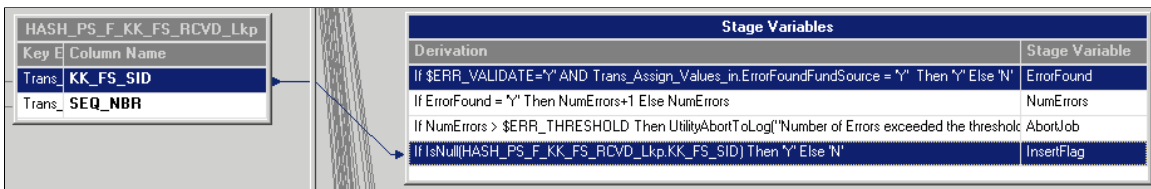
The `Trans_Assign_Values_out` constraint will insert a new record when the `InsertFlag` is set to 'Y' and the `ErrorFound` flag is set to 'N.'

The `Trans_Assign_Values_upd` constraint will update a record when the `InsertFlag` is set to 'N' and the `ErrorFound` flag is set to 'N.'

Looking at the stage variables in the same transformation, note that the `InsertFlag` value (Y or N) is set based on the `HASH_PS_F_KK_FS_RCVD_LKP`:

Image: Trans_Assign_Values transformation stage variables

This example illustrates the `Trans_Assign_Values` transformation stage variables.

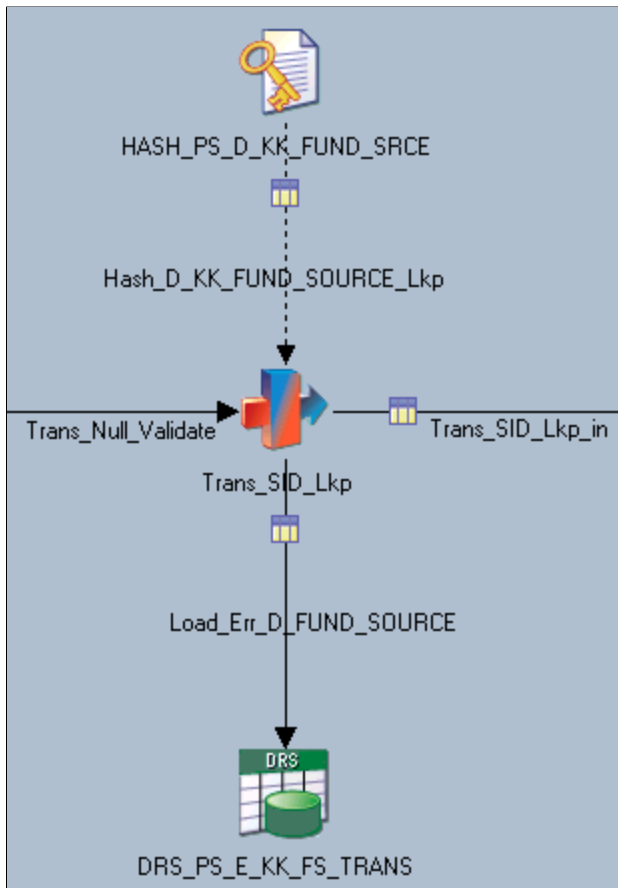


The `HASH_PS_F_KK_FS_RCVD_LKP` uses incoming values for the source keys `KK_FS_SID` and `SEQ_NBR` to determine the `SID` value for each row of fact data. If the lookup returns a null `SID` value based on the source keys, the `InsertFlag` is set to 'Y' (insert a new record). If the lookup returns an existing `SID` value based on the source keys, the `InsertFlag` is set to 'N' (update existing record).

Also note that the ErrorFound flag value (Y or N) is set based on the *ErrorFoundFundSource* stage variable. However, the value of the ErrorFoundFundSource stage variable is determined in an earlier transformation, the Trans_SID_Lkp transformation:

Image: Trans_SID_Lkp transformation

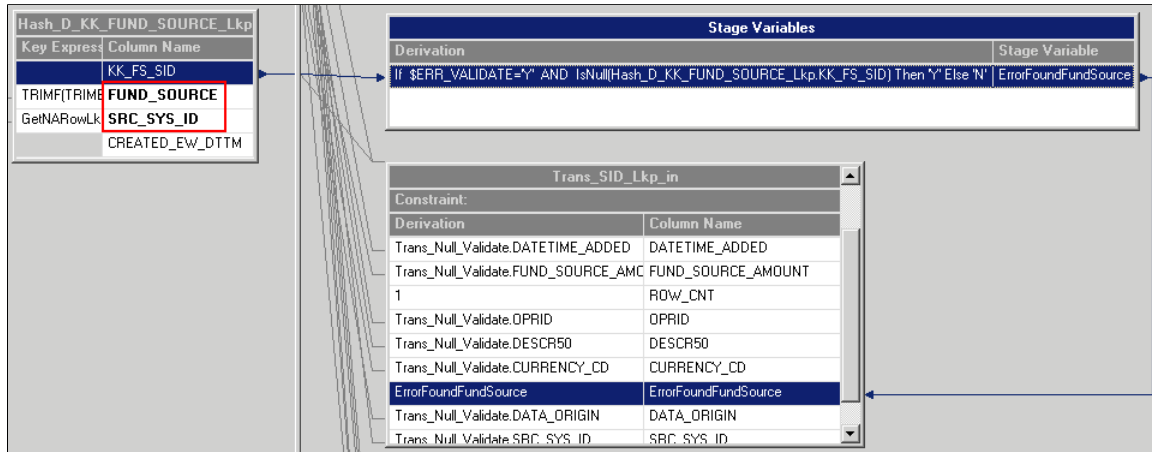
This example illustrates the f Trans_SID_Lkp transformation.



Looking at the ErrorFoundFundSource stage variable in the same transformation, note that its value is set based on the HASH_D_KK_FUND_SOURCE_LKP:

Image: ErrorFoundFundSource stage variable

This example illustrates the ErrorFoundFundSource stage variable.



The HASH_D_KK_FUND_SOURCE_LKP uses incoming values for the source keys *FUND_SOURCE* and *SRC_SYS_ID* to determine the SID value for each row of fact data. If the lookup returns a null SID value based on the source keys, the ErrorFoundFundSource is set to 'Y' (error found). If the lookup returns an existing SID value based on the source keys, the ErrorFoundFundSource is set to 'N' (no error).

Note: Normally the *\$ERR_VALIDATE* parameter is set to 'Y'. By default EPM is delivered with the value set to Y, which means that records failing validation are moved an error table.

Special Load Requirements

The complex process behind integrating and aggregating disparate source data can create some special load requirements in EPM. For example, subrecords are used extensively in EPM target tables to provide additional depth and breadth of processing.

Passing Default Values to EPM Target Tables

Due to data aggregation and other processing requirements, EPM target tables may contain columns that do not exist in your source transaction tables. Because of the differences between source and EPM columns, there are sometimes no source values to populate the EPM columns. Therefore, default values must be used to populate the EPM columns instead.

<i>Warehouse Layer</i>	<i>Data Type</i>	<i>Default Value</i>
OWS	Char	' '
	Num	0
	Date	Null
OWE	Char	' '

Warehouse Layer	Data Type	Default Value
	Num	0
	Date	Null
MDW	Char	'.'
	Num	0
	Date	Null

For MDW fact records, fact rows coming from the source normally contain a valid reference to an existing row in the dimension table, in the form of a foreign key using a business key field. However, occasionally a fact row does not contain the dimension key. To resolve this issue, each MDW dimension contains a row for *Value Not Specified*, with predefined key values of zero—for a missing numeric value—and a hyphen—for a missing character value.

PeopleSoft delivers several routines to pass default values to the EPM columns. For example, the routine *GetNumDefault* is used to pass numeric default values to a target warehouse table. A separate routine is delivered for each data type (such as varchar and numeric).

Target Table Subrecords

Subrecords are a collection of specific columns that repeat across multiple EPM target tables. Subrecords can perform a variety of functions, including tracking data to its original source and facilitating customizations that enable type 2 slowly changing dimensions. For example, the subrecord *LOAD_OWS_SBR* contains columns such as *CREATED_EW_DTTM*, *LAST_UPD_DTTM*, and *BATCH_SID* which help track target warehouse table load history.

It is important to populate subrecords with the appropriate data. Thus, it is important that you thoroughly familiarize yourself with the PeopleSoft delivered subrecords and their associated columns.

Understanding Data Validation and Error Handling in the ETL Process

Accurate reporting is completely dependent on the data stored in data warehouse OWE and MDW tables; if incomplete or incorrect data resides in these tables, reporting and analysis can be flawed. Given the considerable dependence on data in EPM tables, all source data entering EPM must be validated.

Data validations are performed when you run ETL jobs. Because we want to ensure that complete, accurate data resides in the OWE and MDW tables, data validations are embedded in the jobs that load data from the OWS to the OWE and MDW. Therefore, data that passes the validation process is loaded into OWE and MDW target tables, while data that fails the validation process is redirected to separate error tables in the OWS. This ensures that flawed data never finds its way into the target OWE and MDW tables.

Error tables log the source values failing validation to aid correction of the data in the source system. There is an error table for each OWS driver table. OWS driver tables are those tables that contain the primary information for the target entity (for example customer ID).

Data Completeness Validation and Job Statistic Summary for Campus Solutions, FMS, and HCM Warehouses

A separate data completeness validation and job statistic capture is performed against the data being loaded into Campus Solutions, FMS, and HCM MDW tables (for example, validating that all records, fields, and content of each field is loaded, determining source row count versus target insert row count, and so forth). The validation and job statistic tracking is also performed in ETL jobs. The data is output to the PS_DAT_VAL_SMRY_TBL and PS_DATVAL_CTRL_TBL tables with prepackaged Oracle Business Intelligence (OBIEE) reports built on top of the tables.

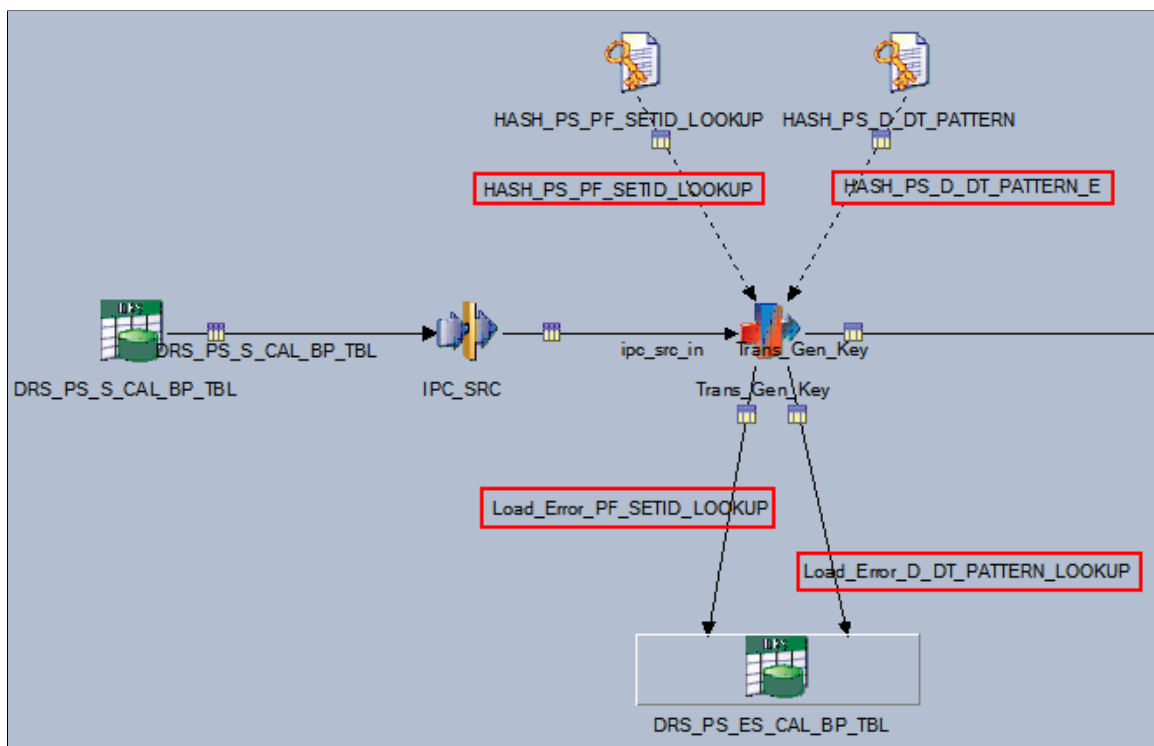
See *PeopleSoft EPM: Fusion Campus Solutions Intelligence for PeopleSoft*.

Understanding the Data Validation Mechanism

The following graphic represents the data validation-error handling process in the PeopleSoft delivered J_DIM_PS_D_DET_BUDGET job:

Image: Data validation in the J_DIM_PS_D_DET_BUDGET job

This example illustrates the Data validation in the J_DIM_PS_D_DET_BUDGET job.



Note that two hashed file validations are performed on the source data: the HASH_PS_PF_SETID_LOOKUP (which validates SetID) and HASH_PS_D_DT_PATTERN (which validates pattern code). Any data failing validation of these lookups is sent to the OWS error table (DRS_PS_ES_CAL_BP_TBL) via the Load_Error_PF_SETID_LOOKUP and Load_Error_D_DT_PATTERN_LOOKUP.

A closer look at the stage variables in the `Trans_Gen_Key` transformer stage demonstrate how the data validation process works (open the stage, then click the Show/Hide Stage Variables button on Transformer toolbar):

Image: Trans_Gen_Key transformer stage variables

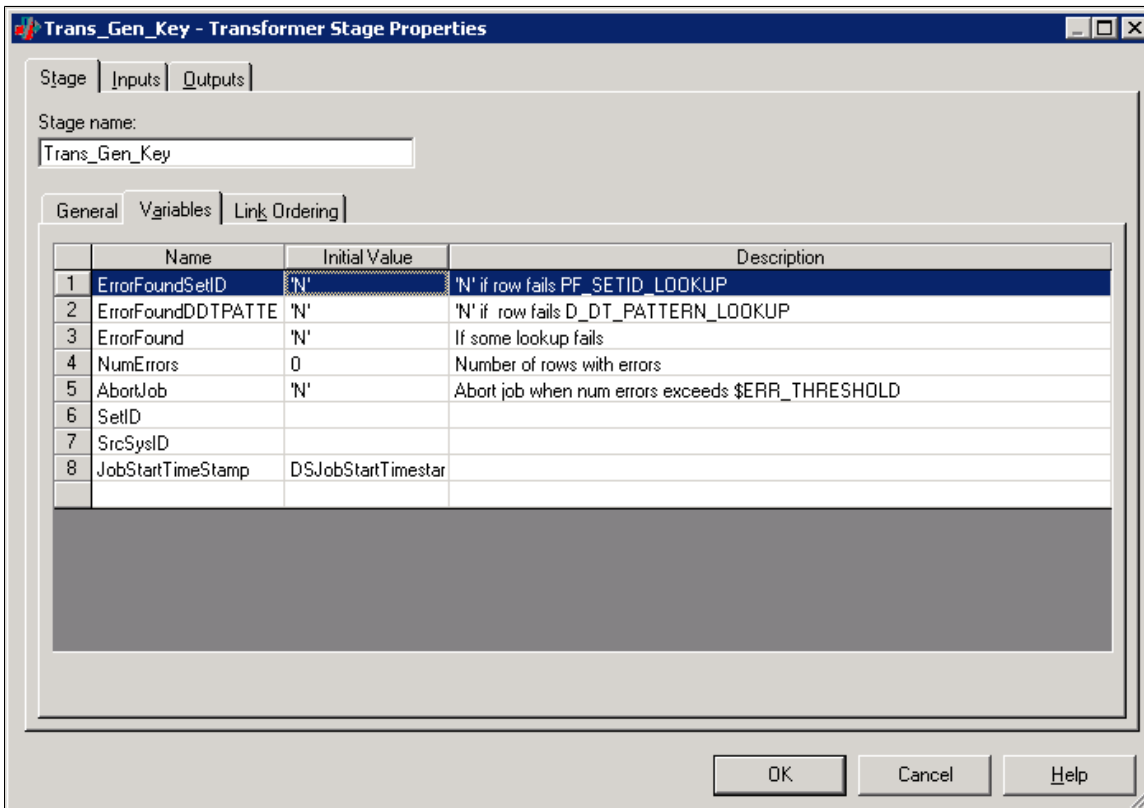
This example illustrates the `Trans_Gen_Key` transformer stage variables. .

Stage Variables	
Derivation	Stage Variable
If HASH_PS_PF_SETID_LOOKUP.NOTFOUND Then 'Y' Else 'N'	ErrorFoundSetID
IF HASH_PS_D_DT_PATTERN_E.NOTFOUND THEN 'Y' ELSE 'N'	ErrorFoundDDTPATTERN
If \$ERR_VALIDATE = 'Y' AND (ErrorFoundSetID = 'Y' OR ErrorFoundDDTPATTERN = 'Y') Then 'Y' Else 'N'	ErrorFound
If ErrorFound = 'Y' Then NumErrors+1 Else NumErrors	NumErrors
If NumErrors > \$ERR_THRESHOLD Then UtilityAbortToLog("Number of Errors exceeded the threshold limit and aborting job")	AbortJob
NullToCharDefault(ipc_src_in.SETID)	SetID
NullToCharDefault(ipc_src_in.SRC_SYS_ID)	SrcSysID
	JobStartTimeStamp

Note that the *ErrorFoundSetID* and *ErrorFoundDDTPATTERN* stage variable derivations are set to Y if the SETID lookup or pattern code validations fail.

Image: Stage variable properties

This example illustrates the Stage variable properties.



The value of the *ErrorFound* stage variable, however, depends on the values of the *ErrorFoundSetID* and *ErrorFoundDDTPATTERN* stage variables, as well as the value of the *\$ERR_VALIDATE* parameter, which can be configured to Y or N. If the *\$ERR_VALIDATE* parameter is set to Y, rows that fail

validation are written to the error table. If the value is set to N, rows that fail validation still pass to the target table.

Also note the *AbortJob* stage variable derivation uses the *\$ERR_THRESHOLD* parameter to limit the number of error records allowed in the job. If the number of error records exceed the value set for the *\$ERR_THRESHOLD* parameter, the job automatically aborts. For example, if *\$ERR_THRESHOLD* is set to 50, the job aborts if the number of records with errors exceeds 50. You can set the value of the *\$ERR_THRESHOLD* parameter to meet your specific business requirements.

Using the SetID lookup validation as an example, if a record fails validation, a Y value is assigned to the ErrorFoundSetID stage variable. If the *\$ERR_VALIDATE* parameter is also set to Y, the failed record is sent to the PS_ES_CAL_BP_TBL error table.

Image: Example constraint

This example illustrates the Example constraint.

Load_Error_PF_SETID_LOOKUP	
Constraint: \$ERR_VALIDATE = 'Y' AND ErrorFoundSetID = 'Y'	
Derivation	Column Name
'S_CAL_BP_TBL.SRC_SYS_ID "CAL_BP_TBL" S_CAL_BP_TBL	SOURCE_COLUMN_L
'PF_SETID_LOOKUP'	LOOKUP_TABLE
'SRC_SYS_ID SRC_RECNAME SRC_SETID PF_RECNAME'	LOOKUP_COL_LIST
'Y'	LOAD_ERROR
\$DATA_ORIGIN	DATA_ORIGIN
JobStartTimeStamp	CREATED_EW_DTTM
JobStartTimeStamp	LASTUPD_EW_DTTM
BATCH_SID	BATCH_SID

For records that pass validation, an N value is assigned to the ErrorFound stage variable and the records are sent to the target table.

Image: N value is assigned to the ErrorFound stage variable

N value is assigned to the ErrorFound stage variable

Trans_Gen_Key	
Constraint: ErrorFound = 'N'	
Derivation	Column Name
ipc_src_in.SETID	SRC_SETID
NullToCharDefault(HASH_PS_PF_SETID_LOOKUP.SETID)	SETID
ipc_src_in.CALENDAR_ID	CALENDAR_ID
ipc_src_in.BUDGET_PERIOD	BUDGET_PERIOD
ipc_src_in.SRC_SYS_ID	SRC_SYS_ID
ipc_src_in.BEGIN_DT	BEGIN_DT
ipc_src_in.END_DT	END_DT
ipc_src_in.PERIOD_NAME	PERIOD_NAME

Disabling Data Validation

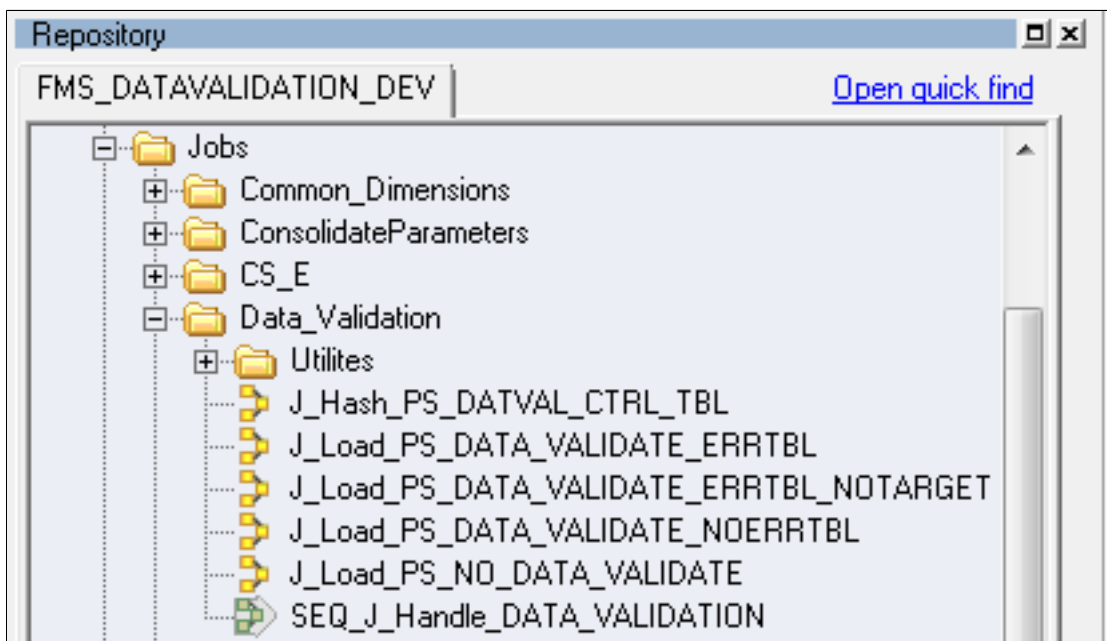
You can disable error validation in OWS jobs by configuring the value of the \$ERR_VALIDATE parameter. By default the value is set to Y, which means that records failing validation are moved an error table. If you set the \$ERR_VALIDATE value to N, records failing validation will still pass to the target table.

Understanding the Data Completeness Validation and Job Statistic Summary Mechanism

The ETL component of the new data validation feature can be found using the following navigation in the DataStage Designer repository window: Jobs, Data_Validation.

Image: Data_Validation jobs in the DataStage project tree

This example illustrates the the Data_Validation jobs in the DataStage project tree.

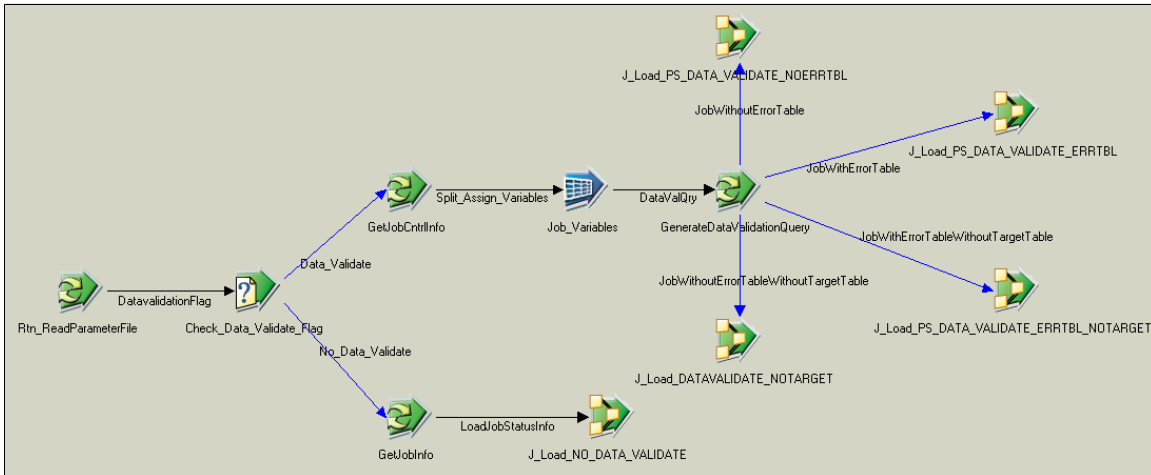


Data Validation - Job Summary Data

The ETL logic for the new data validation feature is contained in a reusable, common component called SEQ_J_Handle_DATA_VALIDATION, which is a sequencer job:

Image: SEQ_J_Handle_DATA_VALIDATION sequencer job

This example illustrates the SEQ_J_Handle_DATA_VALIDATION sequencer job.

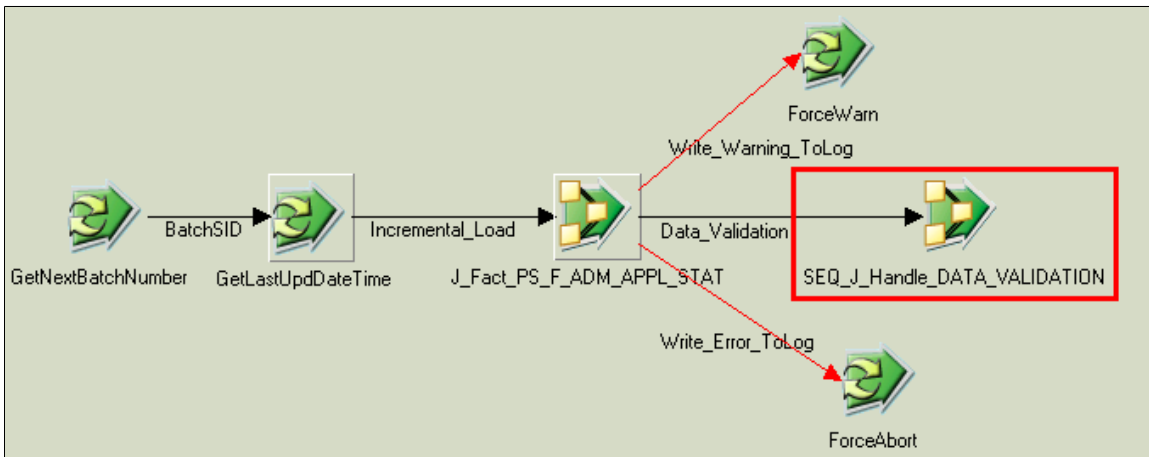


Since the logic is reusable, the sequencer job is incorporated into all the existing sequencer jobs that load the MDW. Also, because the logic is contained within a sequencer job, there is no need to modify existing server jobs to implement the logic.

Here is the SEQ_J_Handle_DATA_VALIDATION sequencer job incorporated into an existing sequencer job for the Campus Solutions Warehouse, SEQ_J_Fact_PS_F_ADM_APPL_STAT:

Image: SEQ_J_Fact_PS_F_ADM_APPL_STAT sequencer job with the data validation component

SEQ_J_Fact_PS_F_ADM_APPL_STAT sequencer job with the data validation component.



Note: If you have customized your ETL jobs, you can implement the new data validation component simply by attaching it to the associated sequencer job.

The SEQ_J_Handle_DATA_VALIDATION sequencer job consists of the following server jobs:

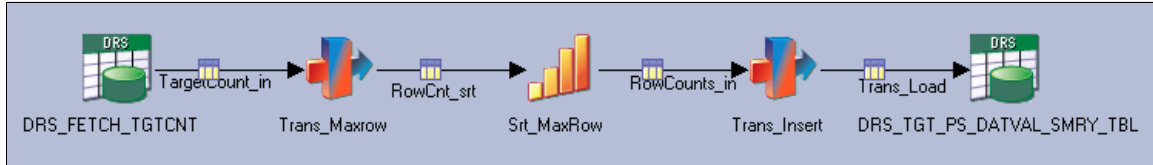
- J_Load_PS_DATA_VALIDATE_ERRTBL

- J_Load_PS_DATA_VALIDATE_NOERRTBL
- J_Load_DATAVALIDATE_NOTARGET
- J_Load_PS_DATA_VALIDATE_ERRTBL_NOTARGET

A closer look at these server jobs shows that they load the Data Validation Summary (PS_DAT_VAL_SMRY_TBL) table:

Image: J_Load_PS_DATA_VALIDATE_NOERRTBL

J_Load_PS_DATA_VALIDATE_NOERRTBL

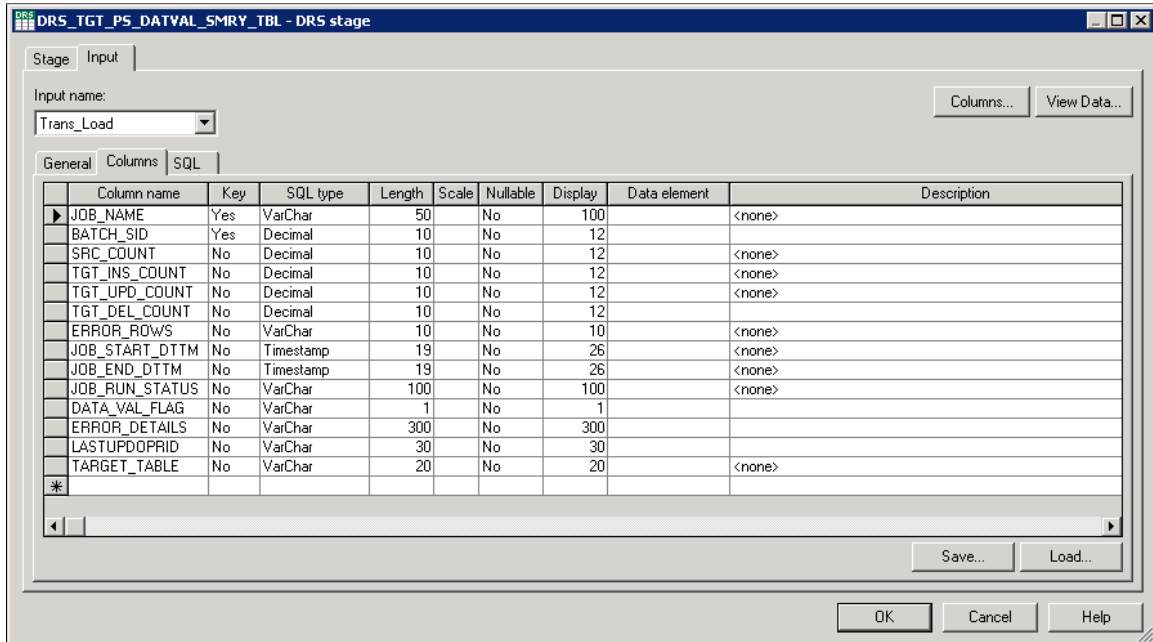


The Data Validation Summary (PS_DAT_VAL_SMRY_TBL) table consolidates all your job run statistic and error data, and is used as the foundation for the delivered data validation OBIEE reports (which are discussed in the PeopleSoft EPM Fusion Campus Solutions Intelligence for PeopleSoft documentation).

The following graphic and table provide a more detailed look at the columns included in the Data Validation Summary (PS_DAT_VAL_SMRY_TBL) table:

Image: PS_DAT_VAL_SMRY_TBL

This example illustrates the PS_DAT_VAL_SMRY_TBL.



Column	Description
JOB_NAME	Populated with the server job name for each job run. This is a composite Primary Key.

Column	Description
BATCH_SID	Contains the BATCH_SID for the corresponding job run. This is a composite Primary Key.
SRC_COUNT	Contains the row count from the source that was extracted by the job.
TGT_INS_COUNT	Contains the row count for rows inserted into the target table by the job.
TGT_UPD_COUNT	Contains the row count for rows updated in the target table by the job.
TGT_DEL_COUNT	Contains the row count for rows deleted from the target table by the job.
ERROR_ROWS	Contains the count of unique rows from the source that failed to load the target due to lookup validation failure. Will default to '-' when: <ul style="list-style-type: none"> • Unique error count cannot be captured (for example, jobs having multiple error tables). • Jobs use parent-child relationship tables, but the error data is captured at the parent table level only.
JOB_START_DTTM	Contains the server job start time. This will be the LASTUPD_EW_DTTM in the target tables.
JOB_END_DTTM	Contains the server job end time.

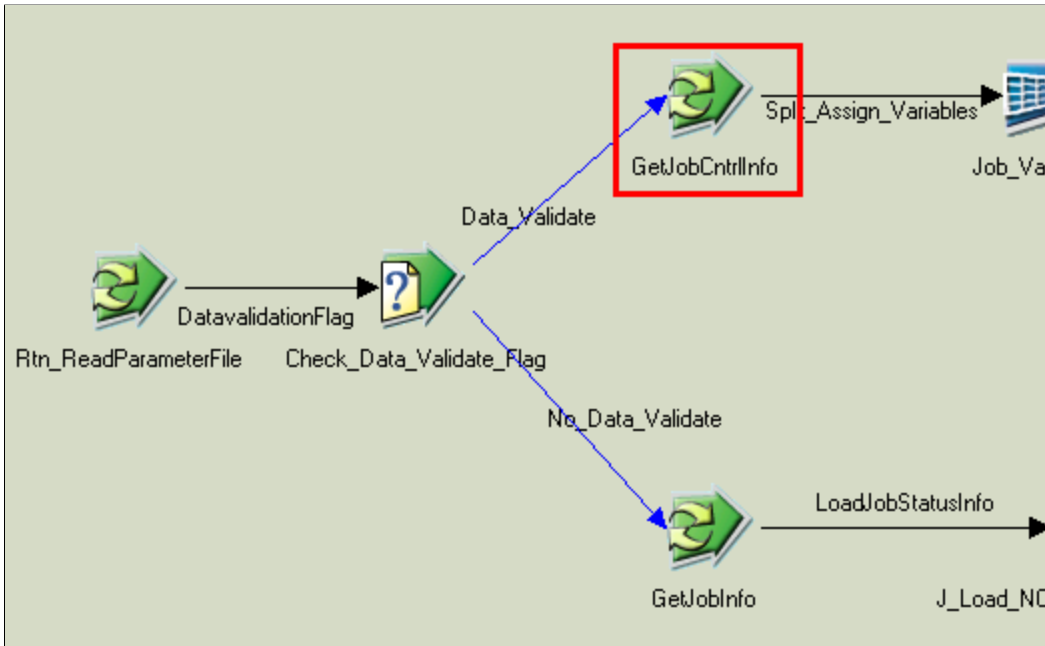
Column	Description
JOB_RUN_STATUS	<p>Contains the run status of a job and whether the data has been loaded into the target completely. Job run status values are:</p> <ul style="list-style-type: none"> • <i>Success</i>: Indicates the ETL Job ran to success and all rows from the source are loaded to the target successfully. Hence the source count will match with sum of target rows. • <i>Success with Error Rows</i>: Indicates the ETL Job ran to success but some rows are rejected to Error Table. In this case the source count will tally with sum of target rows and error rows, provided EXCEPTION_FLAG is set to 'N.' • <i>Success with Data Mismatch</i>: Indicates the ETL Job ran to success but source count does not tally with the sum of target counts and no rows are inserted into error table. • <i>Warning</i>: Indicates the ETL Job finished with warnings. In this case the source count might not tally with the sum of target rows and error rows. • <i>Failed</i>: Indicates the ETL Job aborted. In this case the source count will not tally with the sum of target rows and error rows. <hr/> <p>Note: Even if data validation is disabled for a job, job run status is still captured so that you can use the delivered Job Run Statistics OBIEE report to monitor job status.</p> <hr/>
DATA_VAL_FLAG	Flag indicating whether the data validation flag is enabled or disabled for the server job.
ERROR_DETAILS	<p>Contains the error table names and total count of rows loaded into each error table.</p> <p>When no rows are loaded into error table it will be defaulted to "_"</p>
LASTUPDOPRID	Contains the DataStage user information who triggered or ran the job.
TARGET_TABLE	Contains the target table name to which data was loaded.

Data Validation - Job Statistic Data

The SEQ_J_Handle_DATA_VALIDATION sequencer job also includes a mechanism to capture job statistic data. Note that within the sequencer job is the routine *GetJobCntrlInfo*:

Image: GetJobCntrlInfo routine in the SEQ_J_Handle_DATA_VALIDATION sequencer job

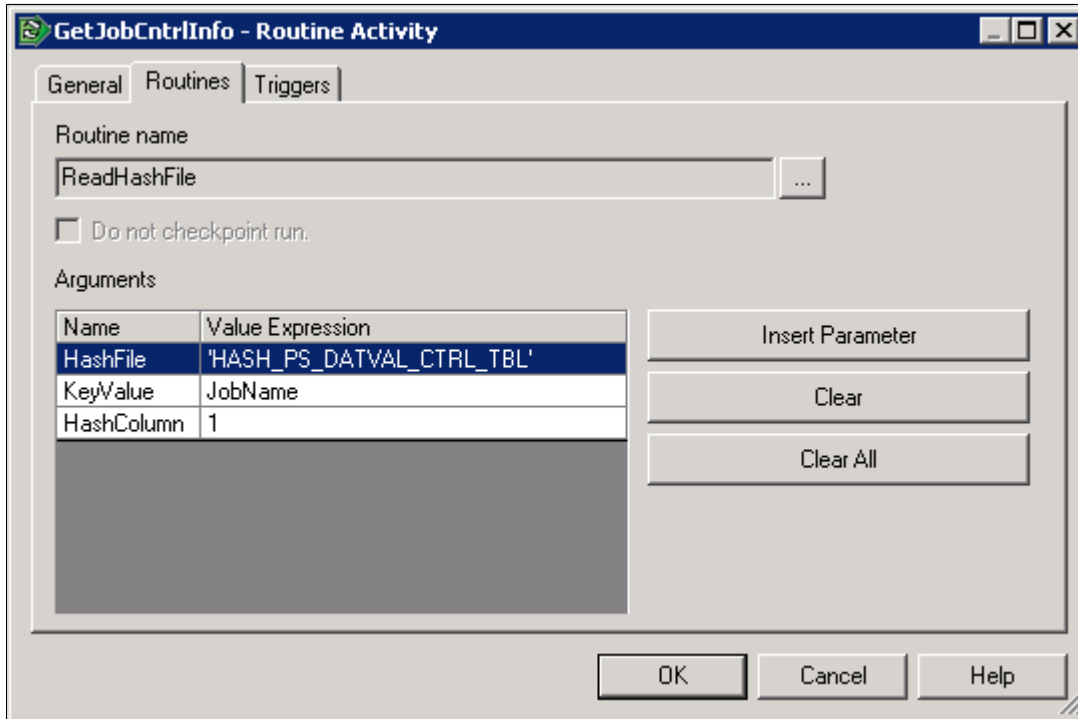
This example illustrates the GetJobCntrlInfo routine in the SEQ_J_Handle_DATA_VALIDATION sequencer job.



The GetJobCntrlInfo routine retrieves job information (such as error table and exception flag) from the Data Validation Control (HASH_PS_DATVAL_CTRL_TBL) hashed file:

Image: GetJobCntrlInfo routine detail

This example illustrates the GetJobCntrlInfo routine detail.



The job information in the hashed file is also used by the server jobs that load the Data Validation Summary (PS_DAT_VAL_SMRY_TBL) table.

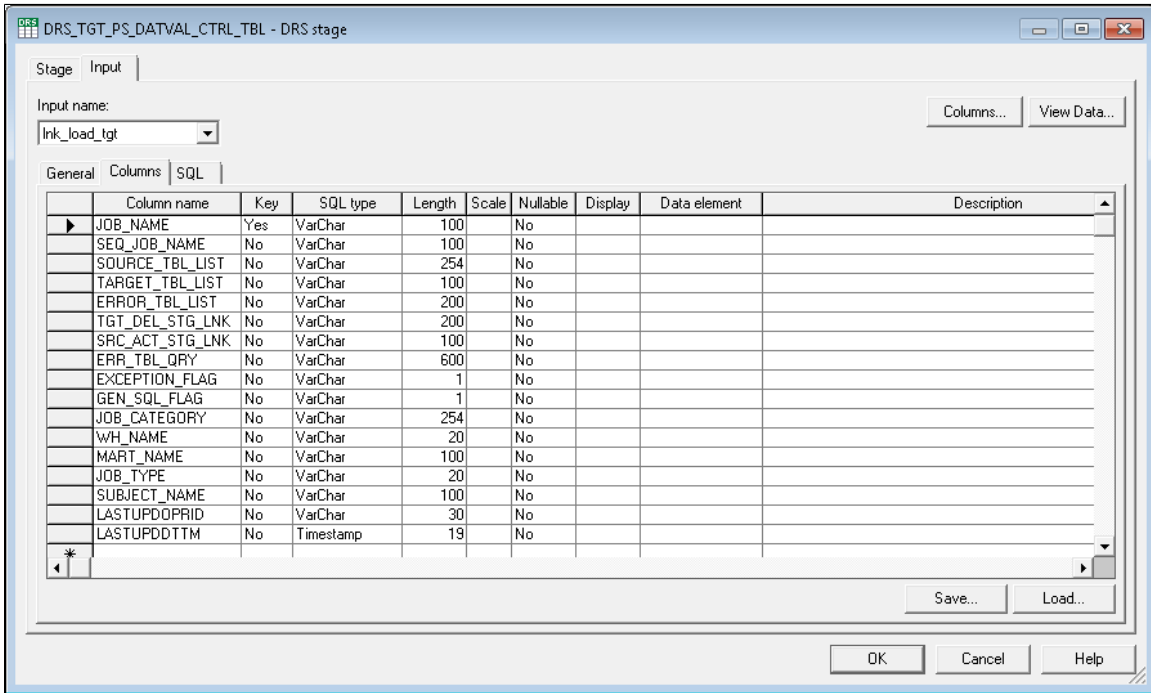
The *Data Validation Control* (PS_DATVAL_CTRL_TBL) table stores job statistic data for each OWS to MDW job run, such as source count, target count, error count, and error table list. The table is delivered prepopulated with the necessary data for the ETL jobs that perform data validation.

Note: If you have customized your ETL Jobs, you must manually update/insert data for the customized job in the PS_DATVAL_CTRL_TBL table and then run the J_Hash_PS_DATVAL_CTRL_TBL server job to update the related hashed file. See the following topic on how to perform these tasks.

The following graphic and table provide a more detailed look at the columns included in the Data Validation Control (PS_DATVAL_CTRL_TBL) table:

Image: PS_DATVAL_CTRL_TBL

This example illustrates the PS_DATVAL_CTRL_TBL.



Column	Description
JOB_NAME	Populated with the server job name that loads the OWS and MDW table. This is a primary key in this table.
SEQ_JOB_NAME	Populated with the Sequencer Name of the server job.
SOURCE_TBL_LIST	Contains the list of source table names used in the job.
TARGET_TBL_LIST	Contains the list of target target table names used in the job. This column may also contain parameter names when the target table is parameterized.
ERROR_TBL_LIST	Contains the list of error tables populated in the job.
TGT_DEL_STG_LNK	Identifies the active stage that precedes the DRS stage that performs a delete on target table, which is used to retrieve the target deleted row count.
SRC_ACT_STG_LNK	Identifies the active stage that succeeds the source DRS stage, which is used to retrieve the source row count.

Column	Description
ERR_TBL_QRY	Contains the error table query that is used to fetch the count of unique errors rows that failed to load the target.
EXCEPTION_FLAG	Contains 'Y' for jobs when source row count might not tally with the sum of target and error row counts (due to functional or job design). Otherwise, contains 'N.'
GEN_SQL_FLAG	Contains 'Y' if DRS stage uses generated sql query. Otherwise, contains 'N' if DRS stage uses user-defined SQL query.
JOB_CATEGORY	Contains the category (folder) information in the DataStage Project Repository for the job.
WH_NAME	Contains the EPM Warehouse name related to the job. The data validation component uses this value to select the appropriate parameter file at runtime.
MART_NAME	Contains the data mart name that corresponds to the parent EPM Warehouse, for the job run.
JOB_TYPE	Contains the job type information for the job. Job type values include: <ul style="list-style-type: none"> • STAGING • DIMENSION • FACT
SUBJECT_NAME	Contains the functional area that corresponds to the parent data mart, for the job.
LASTUPDOPRID	Contains the user information associated with the insert or update actions, for a given job.
LASTUPDDTTM	Contains the timestamp when data was last modified.

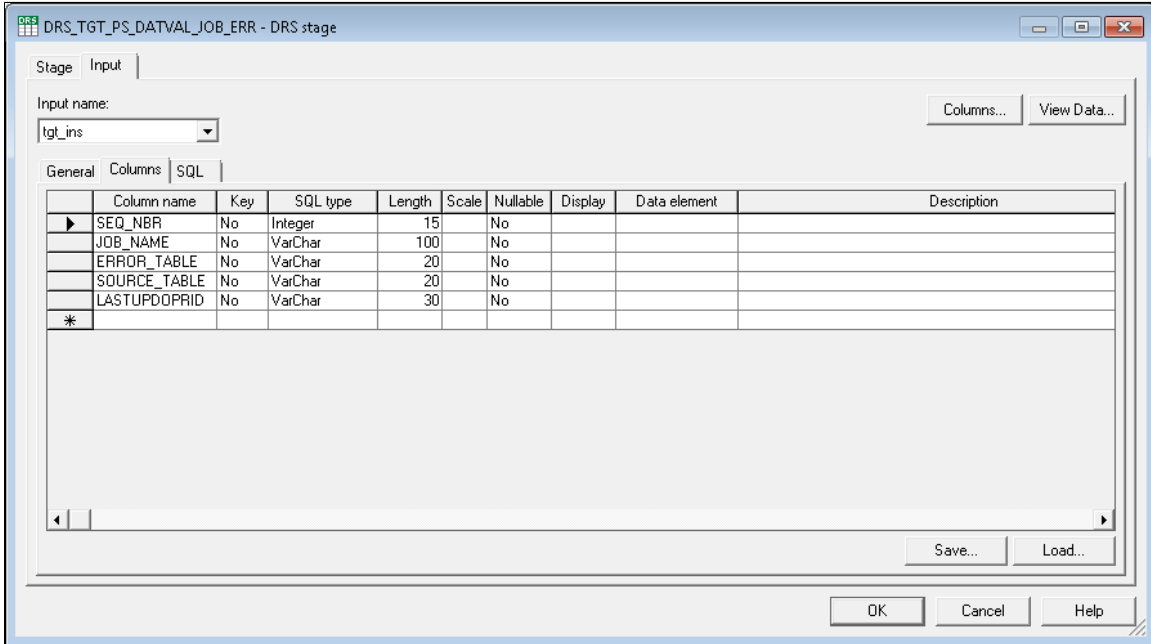
Another data validation-job statistic table, the *Data Validation Job Error (PS_DATVAL_JOB_ERR)* table, is populated from the PS_DATVAL_CTRL_TBL by an in-house ETL job not delivered with DataStage metadata.

The Data Validation Job Error table is used in Error Table OBIEE reports and contains specific error table information and corresponding source table information for each job. The Data Validation Job Error table is populated from the PS_DATVAL_CTRL_TBL by an in-house ETL job not delivered with DataStage metadata.

The following graphic and table provide a more detailed look at the columns included in the Data Validation Job Error (PS_DATVAL_JOB_ERR) table:

Image: PS_DATVAL_JOB_ERR

This example illustrates the PS_DATVAL_JOB_ERR.



Column	Description
SEQ_NBR	Contains the unique sequence number for each error table.
JOB_NAME	Populated with the server job name for each job run.
ERROR_TABLE	Contains the error table name related to the job.
SOURCE_TABLE	Contains the name of the source table used with a corresponding error table.
LASTUPDOPRID	Contains the user information associated with the insert or update actions, for a given job.

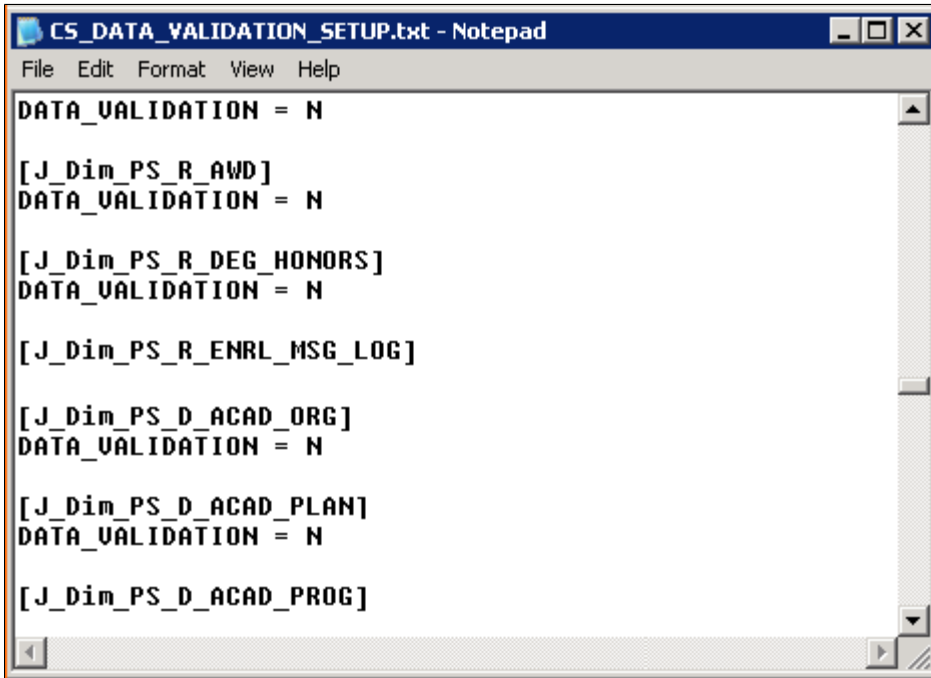
Enabling or Disabling the Data Completion Validation Feature

The Data Validation and Error Reporting feature is designed as optional, and you can enable it or disable it for each staging job using the delivered parameter files. For example, you can access the

HCM_DATA_VALIDATION_SETUP parameter file and change the DATA_VALIDATION value from 'N' (do not perform data validation) to 'Y' (perform data validation) for each job:

Image: Campus Solutions Warehouse Data Validation parameter file

This example illustrates the Campus Solutions Warehouse Data Validation parameter file.



PeopleSoft delivers a separate parameter file for each EPM Warehouse and for Global/Common dimension jobs:

- CS_DATA_VALIDATION_SETUP.txt
- FMS_DATA_VALIDATION_SETUP.txt
- HCM_DATA_VALIDATION_SETUP.txt
- GLOBAL_DATA_VALIDATION_SETUP.txt

Implementing the Data Completeness Validation Feature for Customized Jobs

PeopleSoft provides the DDL Scripts and related data files for the Data Completeness Validation setup tables, but if you have created new jobs or customized delivered jobs, you must perform the following steps to implement the Data Completion Validation feature:

1. Populate the PS_DATVAL_CTRL_TBL.

For each customized job, you must populate this table using an SQL INSERT statement. Use the following table as a guide to populating the PS_DATVAL_CTRL_TBL fields:

<i>Column</i>	<i>Description</i>
JOB_NAME	Populate this field with the server job that loads the OWS and MDW table.

Column	Description
SEQ_JOB_NAME	Populate this field with the Sequencer job that triggers the customized job.
SOURCE_TBL_LIST	<p>Populate this field with source tables used in the customized job.</p> <p>Use commas to separate multiple table names.</p>
TARGET_TBL_LIST	<p>Populate this field with target tables used in the customized job.</p> <p>If target table name is a job parameter, use this sample format:</p> <pre>#TargetTable#(PS_FF_STDNT_ENRL/PS_TF_STDNT_ENRL), where TargetTable is the Job Parameter</pre>
ERROR_TBL_LIST	Populate this field with error tables used in the customized job.
TGT_DEL_STG_LNK	<p>Populate this field with the active stage name that precedes the DRS stage that performs a delete on the target table.</p> <p>If the active stage contains more than one output link, the link name should also be specified using this format:</p> <p><i><active_stage_name>@<link_name></i> where link_name is the link connecting to the Delete DRS stage and active_stage_name is the transformer name.</p> <p>For example:</p> <pre>"PS_R STDNT_ATTR split@IPC_Del Out*-" is value for this field in J_Dim_PS_R_ STDNT_ATTR job</pre>
SRC_ACT_STG_LNK	<p>For CRC staging jobs, populate this field with the active stage name and link name that writes to the Delta hash file. Use the following format:</p> <p><i><active_stage_name>@<link_name></i> where link_name is the link that updates the delta hash file in the CRC job.</p> <p>For non CRC staging jobs, populate this field with the active stage name that follows the source DRS stage in the customized job.</p>

Column	Description
ERR_TBL_QRY	<p>Populate this field with the error table query that is used to fetch the count of unique errors rows that failed to load the target.</p> <p>For customized jobs having one error table, use the following query syntax:</p> <pre>SELECT DISTINCT <BUSINESS KEYS of source table> FROM # \$ERROR_TBL_SCHEMA#<ERROR TABLE NAME> Sample: SELECT DISTINCT SRC_SYS_ID,EFFSEQ,EFFDT,EMPL_RCD,EMPLID FROM # \$ERROR_TBL_SCHEMA#PS_E_JOB</pre> <p>This field is defaulted to "-" for customized jobs with more than one error table.</p>
EXCEPTION_FLAG	<p>Set this flag to "Y" for customized jobs with more than one error table.</p> <p>Set this flag to "Y" for customized jobs that include multiple source tables with parent/chid relationships and the error data is captured only at parent table level.</p>
GEN_SQL_FLAG	<p>Set this flag to "Y" for customized jobs that use generated SQL in the Source DRS stage.</p> <p>Set this flag to "N" when the source DRS stage includes user-defined SQL.</p>
JOB_CATEGORY	Populate this field with the category (folder) name where the job is placed within the DataStage Project Repository.
WH_NAME	Populate this field with the EPM Warehouse name related to the customized job.
MART_NAME	Populate this field with the data mart name that corresponds to the customized job.
JOB_TYPE	<p>Populate this field with the job type information for the customized job. Job type values include:</p> <ul style="list-style-type: none"> • STAGING • DIMENSION • FACT
SUBJECT_NAME	Populate this field with the functional area that corresponds to the customized job.
LASTUPDOPRID	Populate this field with the user ID associated with the insert or update actions for the customized job.

Column	Description
LASTUPDDTTM	Populate this field with the current timestamp representing when data was last modified for the customized job.

Here is a sample of SQL INSERT syntax:

```

INSERT INTO "PS_DATVAL_CTRL_TBL"
(
  JOB_NAME, SOURCE_TBL_LIST, TARGET_TBL_LIST,
  ERROR_TBL_LIST, TGT_DEL_STG_LNK, SRC_ACT_STG_LNK,
  ERR_TBL_QRY,
  EXCEPTION_FLAG, GEN_SQL_FLAG, WH_NAME, MART_NAME,
  SUBJECT_NAME, JOB_TYPE, JOB_CATEGORY,
  LASTUPDOPRID, LASTUPDDTTM, SEQ_JOB_NAME
)
VALUES
(
  'J_Fact_PS_F_KK_EXCEPTION','PS_KK_EXCPTN_TBL','PS_F_KK_EXCEPTION',
  'PS_E_KK_EXCEPTION','Trans_Assign_Values8@Trans_Assign_Values8_out_DEL*-',
  'IPC_SRC.IDENT1',
  'SELECT DISTINCT
SRC_SYS_ID,EXCPTN_TYPE,LEDGER_GROUP,KK_TRAN_LN,KK_TRAN_DT,KK_TRAN_ID
FROM #ERROR_TBL_SCHEMA#PS_E_KK_EXCEPTION',
  'N','Y','FMS','GL_And_Profitability_Mart',
  'Commitment_Control','FACT',
  'Jobs\FMS_E\GL_And_Profitability_Mart\Commitment_Control\OWS_To_MDW\Facts\=>
Base\Load_Tables\Server',
  'EPM',to_timestamp('01-JAN-53 12.00.00.000000000 AM','DD-MON-RR HH.MI.SS.F=>
F AM'),'SEQ_J_Fact_PS_F_KK_EXCEPTION'
);

```

2. Add an entry in Data Validation Parameter File.

For warehouse specific Jobs, add an entry in the corresponding parameter file. For example, customized jobs specific to Campus Solutions require an entry in the CS_DATA_VALIDATION_SETUP.txt and you could use the following format:

```

[J_Dim_PS_D_ACAD_CAR]
DATA_VALIDATION = N

```

For Common Dimensions and Global Dimensions, add an entry in GLOBAL_DATA_VALIDATION_SETUP.txt.

To enable data load validation (capturing source, target and error row counts for each run), the DATA_VALIDATION flag should be set to 'Y'.

3. Run J_Hash_PS_DATVAL_CTRL_TBL job.

Note: It is important that the information for a customized server job is accurately reflected in the PS_DATVAL_CTRL_TBL before running this job.

4. Attach the common component, SEQ_J_Handle_DATA_VALIDATION, sequencer job to the existing sequencer job that corresponds to the customized server job.

Copy the SEQ_J_Handle_DATA_VALIDATION component and paste it into the existing sequencer job that corresponds to the customized server job. Reestablish the links in the new job sequence, map the Job Name and BatchSID parameter in the Job Activity, and verify that hash file directory path, parameter file directory path, and the database related parameters are mapped correctly. Then compile the job.

When you complete these steps, the PS_DATVAL_SMRY_TBL captures job run statistics for each job run by the data validation component.

5. Populate the PS_DATVAL_JOB_ERR.

For each customized job, you must populate this table using an SQL INSERT statement. Use the following table as a guide to populating the PS_DATVAL_JOB_ERR fields:

Column	Description
SEQ_NBR	Populate this field with a unique sequence number for each error table. To avoid overlapping SEQ_NBR, use higher SEQ_NBR (values greater than 999999).
JOB_NAME	Populate this field with the customized server job name.
ERROR_TABLE	Populate this field with the error table name used in the customized job.
SOURCE_TABLE	Populate this field with the name of the source table used with the corresponding error table.
LASTUPDOPRID	Populate this field with the user ID associated with the insert or update actions for the customized job.

Here is a sample of SQL INSERT syntax:

```
INSERT INTO "PS_DATVAL_JOB_ERR"
(SEQ_NBR, JOB_NAME, ERROR_TABLE, SOURCE_TABLE, LASTUPDOPRID)
VALUES
(236, 'J_Dim_PS_D_GM_SPONSOR', 'PS_E_GM_CUSTOMER', 'PS_GM_CUSTOMER', 'dsuser');
```

6. Populate PS_DATVAL_SRC_TBL.

For each customized job, you must populate this table using an SQL INSERT statement. Use the following table as a guide to populating the PS_DATVAL_SRC_TBL:

Column	Description
SEQ_NBR	Populate this field with a unique sequence number for each error table. To avoid overlapping SEQ_NBR, use higher SEQ_NBR (values greater than 999999).
ERROR_TABLE	Populate this field with the error table name used in the customized job.
SOURCE_TABLE	Populate this field with the name of the source table used with the corresponding error table.

Here is a sample of SQL INSERT syntax:

```
INSERT INTO "PS_DATVAL_SRC_TBL"
(SEQ_NBR,ERROR_TABLE,SOURCE_TABLE)
VALUES
(351,'PS_E_WA_COMP_PIN','PS_WA_COMP_PIN_MAP');
```

Understanding OWE Jobs

When you run the ETL jobs that migrate your data from OWS tables to OWE tables, the jobs convert source SetIDs and Business Units to Warehouse SetIDs and Business Units. Therefore, source business units are consolidated into a common Warehouse Business Unit and the OWE tables are Warehouse Business Unit and Warehouse SetID based.

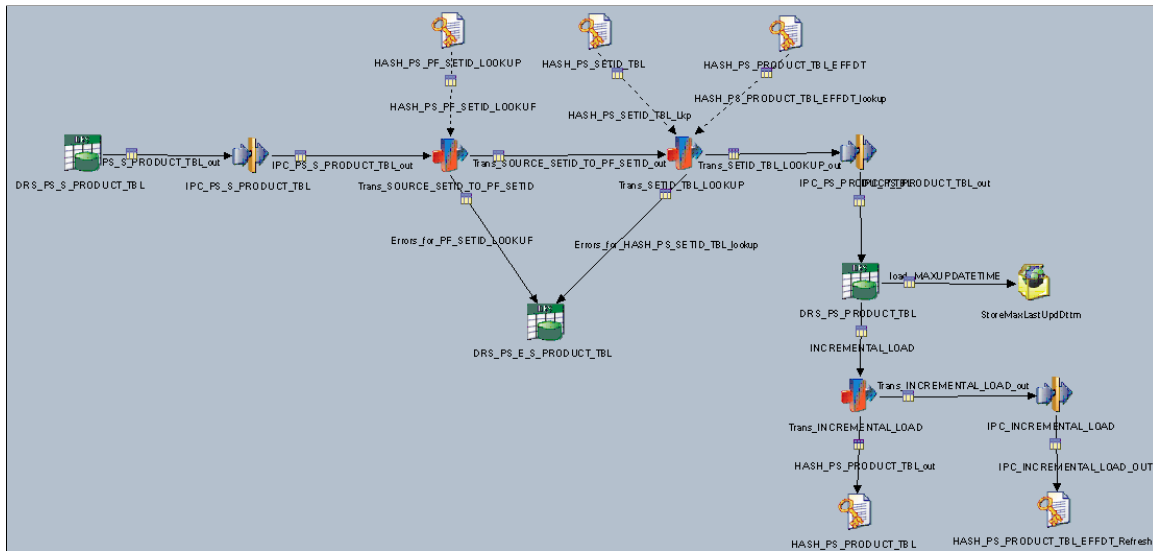
This topic provides an overview of OWE fact and dimension load jobs.

OWE Dimension Load Jobs

A typical OWE dimension job loads data from an OWS source table to a target OWE dimension table. The basic flow of an OWE dimension job starts with a DRS source stage and includes transformation stages to perform lookup validations against OWS or OWE tables, depending on the job requirements.

Image: Sample OWE Dimension Load Job (J_BASE_PS_PRODUCT_TBL)

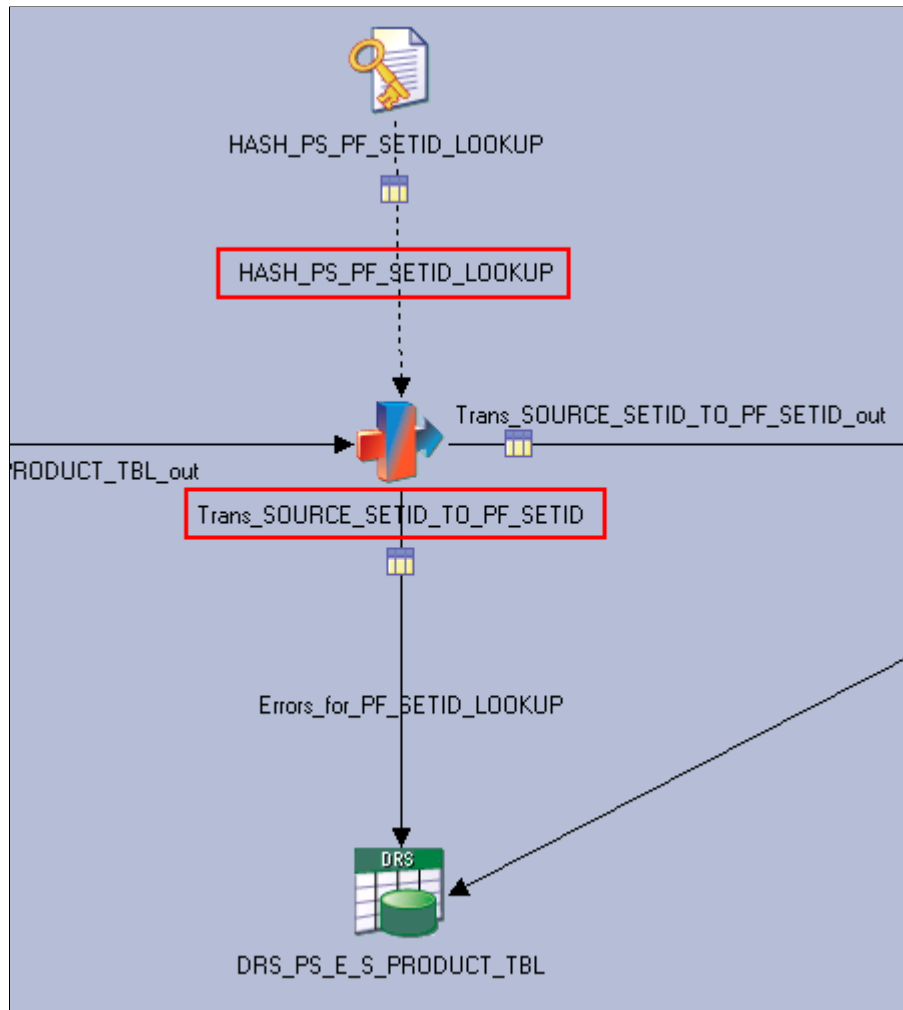
This example illustrates the Sample OWE Dimension Load Job (J_BASE_PS_PRODUCT_TBL).



In the job J_BASE_PS_PRODUCT_TBL, a SetID lookup is performed since the target dimension table is Warehouse SetID based. Specifically, the Trans_SOURCE_SETID_TO_PF_SETID transformation contains the processing logic used to convert incoming source SetIDs to a Warehouse SetID:

Image: Trans_SOURCE_SETID_TO_PF_SETID

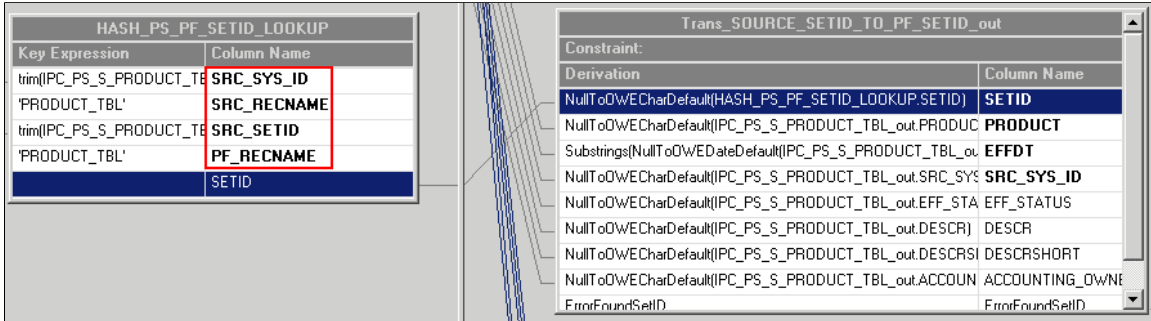
This example illustrates the Trans_SOURCE_SETID_TO_PF_SETID.



A closer look at the Trans_SOURCE_SETID_TO_PF_SETID transformation shows that the SetID lookup is performed against the HASH_PS_PF_SETID_LOOKUP hashed file:

Image: Source SetID to Warehouse SetID conversion logic

This example illustrates the Source SetID to Warehouse SetID conversion logic.



Note that the HASH_PS_PF_SETID_LOOKUP represents the PS_PF_SETID_LOOKUP table, which is one of the output tables populated when you run the Business Unit Wizard.

The HASH_PS_PF_SETID_LOOKUP uses incoming values for the source keys SRC_SYS_ID, SRC_RECNAME, SRC_SETID, and PF_RECNAME to determine the Warehouse SetID value for each row of dimension data. If the lookup returns a Warehouse SetID value based on the source keys, the Warehouse SetID is passed to the target dimension table DRS_PS_PRODUCT_TBL. If the lookup does not locate a Warehouse SetID value based on the source keys, the value is passed instead to the error table DRS_PS_E_S_PRODUCT_TBL.

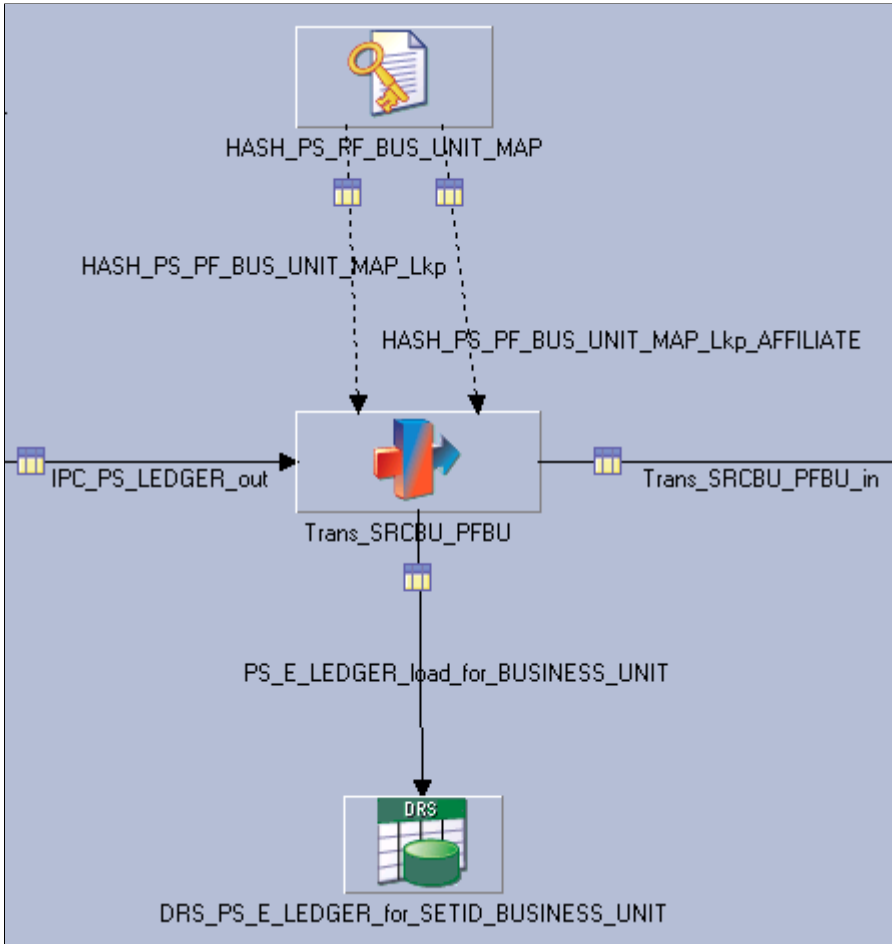
OWE Fact Load Jobs

A typical OWE fact job loads data from an OWS source table to a target OWE fact table. The basic flow of an OWE fact job starts with a DRS source stage and includes transformation stages to perform lookup validations.

In the job J_F00_PS_LEDGER, a business unit lookup is performed since the target fact table is Warehouse Business Unit based. Specifically, the Trans_SRCBU_PFBU transformation contains the processing logic used to convert incoming source business units to a Warehouse Business Unit.

Image: Trans_SRCBU_PFBU

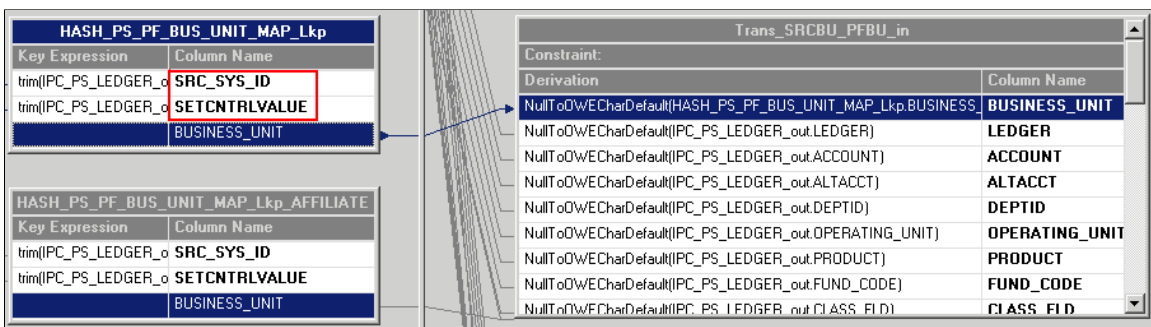
This example illustrates the Trans_SRCBU_PFBU.



A closer look at the Trans_SRCBU_PFBU transformation shows that the business unit lookup is performed against the HASH_PS_PF_BUS_UNIT_MAP hashed file:

Image: Source Business Unit to Warehouse Business Unit conversion logic

This example illustrates the Source Business Unit to Warehouse Business Unit conversion logic.



Note that the HASH_PS_PF_BUS_UNIT_MAP lookup represents the PS_PF_BUS_UNIT_MAP table, which is one of the output tables populated when you run the Business Unit Wizard.

The HASH_PS_PF_BUS_UNIT_MAP uses incoming values for the source keys *SRC_SYS_ID* and *SETCNTRLVALUE* to determine the Warehouse Business Unit value for each row of fact data. If the lookup returns a Warehouse Business Unit value based on the source keys, the Warehouse Business Unit is passed to the target fact table DRS_PS_LEDGER_F00. If the lookup does not locate a Warehouse Business Unit value based on the source keys, the value is passed instead to the error table DRS_PS_E_LEDGER_for_SETID_BUSINESS_UNIT.

Understanding MDW Jobs

This topic provides an overview of MDW fact and dimension load jobs.

MDW Dimension Load Jobs

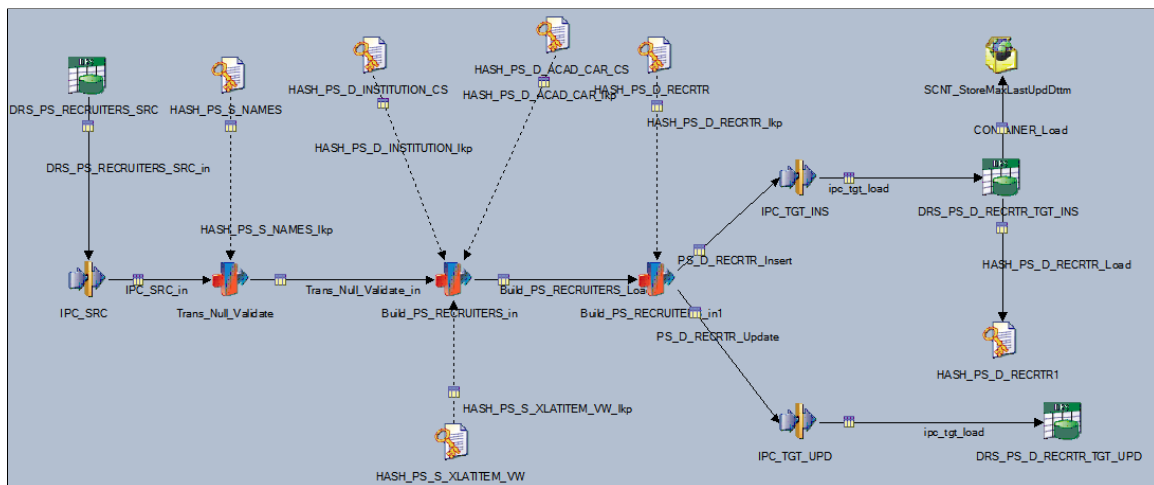
A MDW dimension job loads data from an OWS source table (or in some cases, an OWE table) to a target MDW dimension table. The basic flow of a MDW dimension job starts with a DRS source stage and includes transformation stages with data validation lookups, when necessary, using the SID.

SID validations work in the same manner as the data validations described in the [Understanding Data Validation and Error Handling in the ETL Process](#) topic, except the SID is the unique key identifier used. The job will also contain lookups for attribute values, such as description fields.

The following is an example of a typical MDW dimension load job.

Image: Sample MDW Dimension Load Job (J_DIM_PS_D_RECRTTR)

This example illustrates the Sample MDW Dimension Load Job (J_DIM_PS_D_RECRTTR).



Next the job performs a lookup on the target dimension table hash file to check if equivalent business keys are already present for each record. If the record is present, the existing SID is used. If the record is not present, a new SID is generated. The job loads valid data into the target DRS stage and updates the hash file used for incremental loading.

MDW Fact Load Jobs

A MDW fact job loads data from an OWS source table (or in some cases, an OWE table) to a target MDW fact table. The basic flow of a MDW fact job starts with a DRS source stage and includes transformation stages to validate values for SID lookup dimension tables.

Image: Sample MDW Fact Load Job (J_Fact_PS_F_CAMPUS_EVENT)

This example illustrates the Sample MDW Fact Load Job (J_Fact_PS_F_CAMPUS_EVENT).

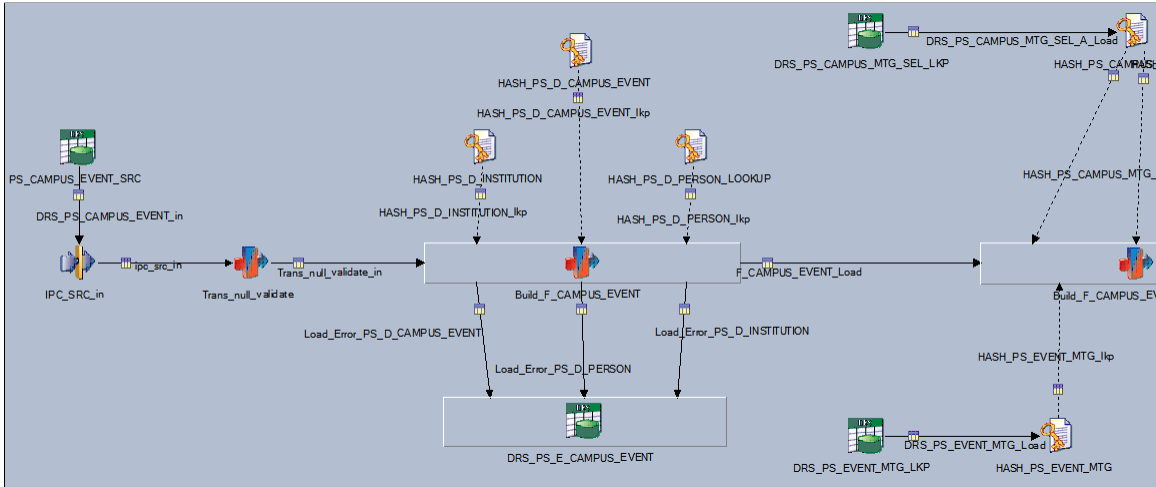
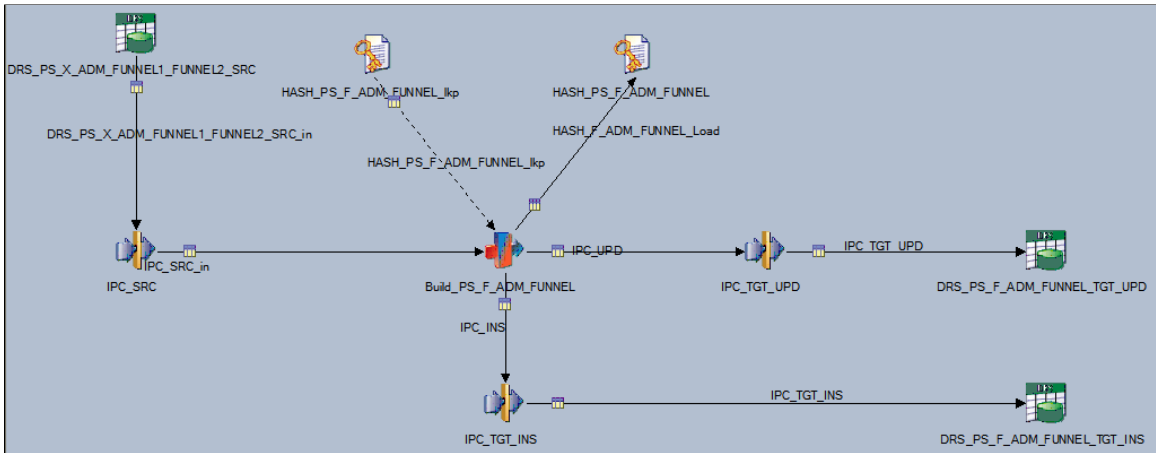


Image: Sample MDW Fact Load Job (J_Fact_PS_F_ADM_FUNNEL)

This example illustrates the Sample MDW Fact Load Job (J_Fact_PS_F_ADM_FUNNEL).

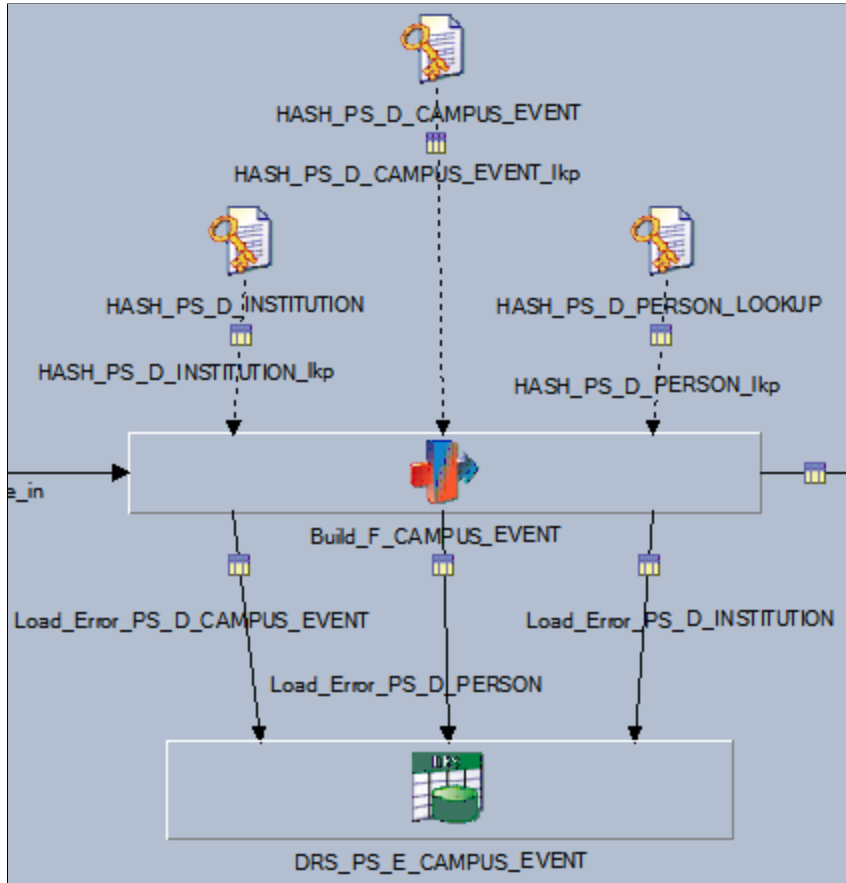


Because transaction tables are based on business unit and some dimension tables are SETID based, sometimes a *SETID indirection* lookup must be performed against the SETCTRL table to obtain the corresponding SETID for the business unit, and then use the value for the lookup. These lookups provide the values for the SID columns in the fact tables. The MDW fact job performs data validation lookups and

diverts records that fail the lookup to an OWS error table (in this case, the PS_ECAMPUS_EVENT error table).

Image: Data Validation and Error Handling in the J_Fact_PS_F_CAMPUS_EVENT Job.

This example illustrates the Data Validation and Error Handling in the J_Fact_PS_F_CAMPUS_EVENT Job.



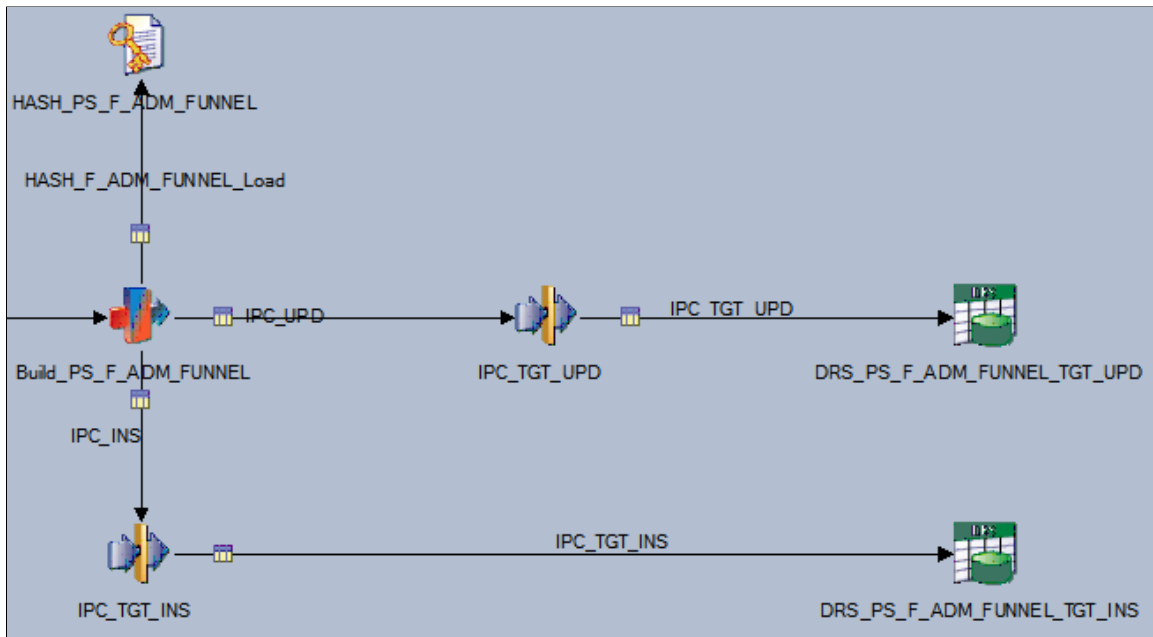
Next, data transformations are sometimes performed in transformation stages, such as aggregation of values or string manipulation.

The HASH_PS_F_ADM_FUNNEL lookup is the final validation in this job and it is required for incremental loading of the MDW target fact table (PS_F_ADM_FUNNEL). This lookup fetches the CREATED_EW_DTTM value for records in the hashed file and determines whether equivalent business keys are already present. If a matching record exists in the hashed file, the same created date time is

extracted from this lookup. The record is then updated in the target fact table. If the record is not present, a new record is inserted in the target fact table.

Image: Update and Insert Data to Target Fact Table in the J_Fact_PS_F_ADM_FUNNEL Job.

This example illustrates the Update and Insert Data to Target Fact Table in the J_Fact_PS_F_ADM_FUNNEL Job..



The job also updates the hash file used for incremental loads. A very small number of MDW fact load jobs use destructive loading, in which case the server job truncates the target table prior to loading data.

Using DataStage Administrator

Using DataStage Administrator

DataStage Administrator enables you to specify general server defaults, administer projects, and set project properties.

The DataStage Administrator window is comprised of the General and Projects tabs.

This topic discusses how to:

- Set DataStage Server properties.
- Set Project properties
- Set DataStage Server Licensing

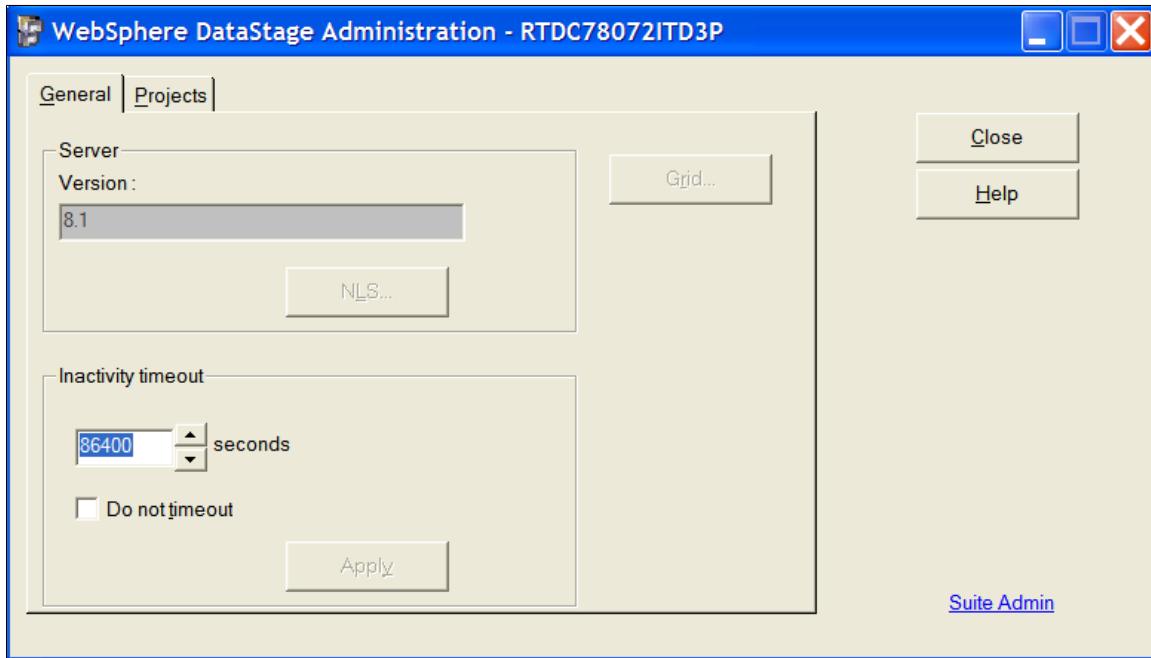
Note: This topic does not discuss all the features available for DataStage Administrator. For a complete view of DataStage Administrator functionality, please see the delivered IBM WebSphere documentation.

Setting DataStage Server Properties

Use the DataStage Administrator - General page to set DataStage server properties.

Image: DataStage Administrator - General Tab

This example illustrates the DataStage Administrator - General Tab.



You can change the following server-wide properties:

NLS

Enable or disable National Language Support (NLS). DataStage supports the language you specify during the install without any further configuration. However, if your requirements change, you can reconfigure NLS to support different languages using DataStage Administrator. Note: You can only change the NLS character set in the DataStage Administrator. You enable and disable NLS support during install.

Inactivity Timeout

Enter the number of seconds of inactivity allowed before the connection between the DataStage client and server times out.

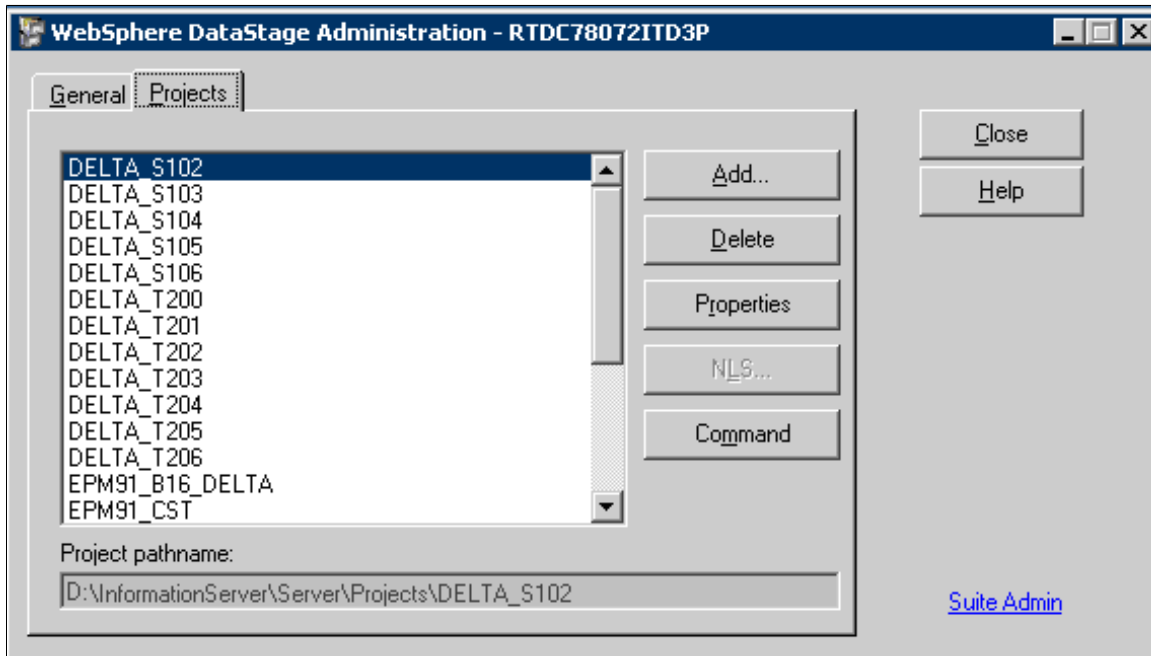
Note: Server-wide property changes made by an administrator affect all projects on the server.

Setting Project Properties

Access the DataStage Administrator - Projects tab.

Image: DataStage Administrator - Projects Tab

This example illustrates the DataStage Administrator - Projects Tab.



Using the DataStage Administrator - Projects tab, administrators can navigate to projects and:

- Add and delete projects.
- Set job administration options.
- Assign user access.
- Enable tracing on the server.
- Set up users for running scheduled jobs (Microsoft Windows only).
- Configure cache settings for hash file stages.
- Select compilation options when job sequences are created.

Project Properties - General Page

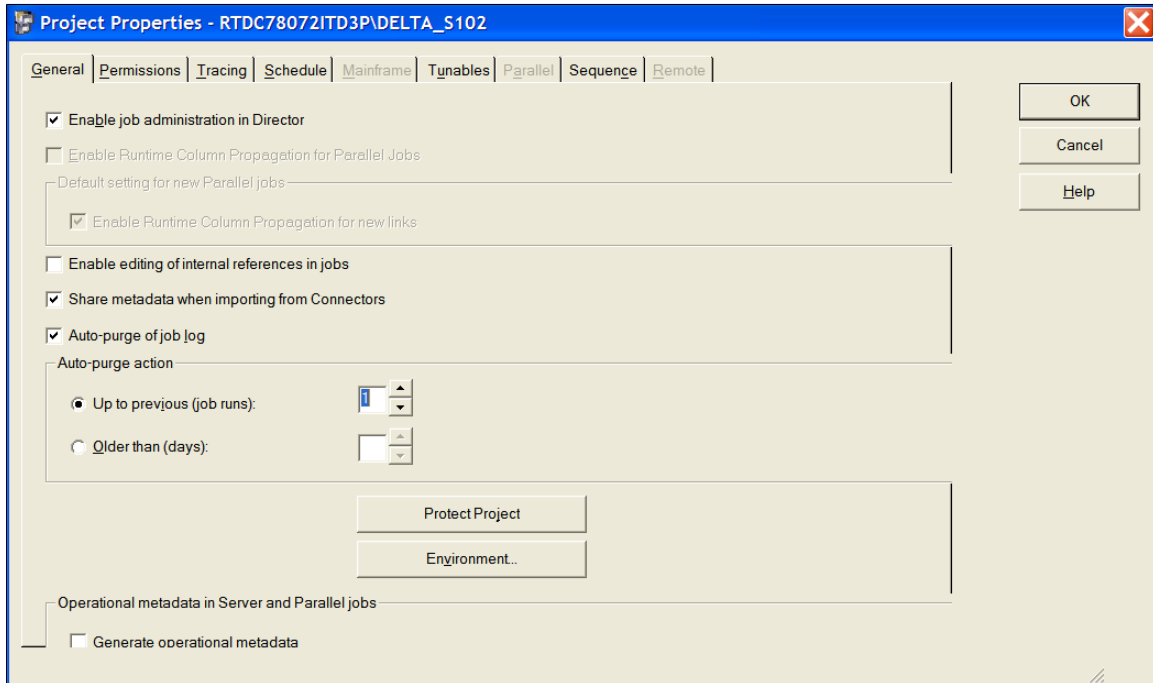
Use the Project Properties - General page to set general properties.

Navigation

Click the Properties button on the DataStage Administrator - Projects tab

Image: Project Properties - General

This example illustrates the Project Properties - General Tab.



The Project Properties - General tab includes the following options:

Enable job administration in Director

Select to use the Cleanup Resources and Clear Status File options from the Job menu of DataStage Director.

Enable Runtime Column Propagation in Parallel Jobs

If you have parallel jobs, select to enable stages to handle undefined columns during the job run. This setting propagates these columns throughout the rest of the job.

Enable remote execution of Parallel Jobs

Select to specify that parallel jobs in a project be deployed on USS systems.

Auto-purge of job log

Select to automatically delete the logs generated when you run a job, according to the criteria you select in the Auto purge action group box.

Up to previous (job runs) and Over (days old)

Select one of these options to delete jobs based on the number of job logs that you want to retain or based on the number of days old a job is. Enter the appropriate value in the adjacent field.

Protect Project

If you have Production Manager permissions, click to convert the project to a protected project to prevent its modification.

Generate Operational Metadata

Select this check box if you want parallel and server jobs in your project to generate operational metadata.

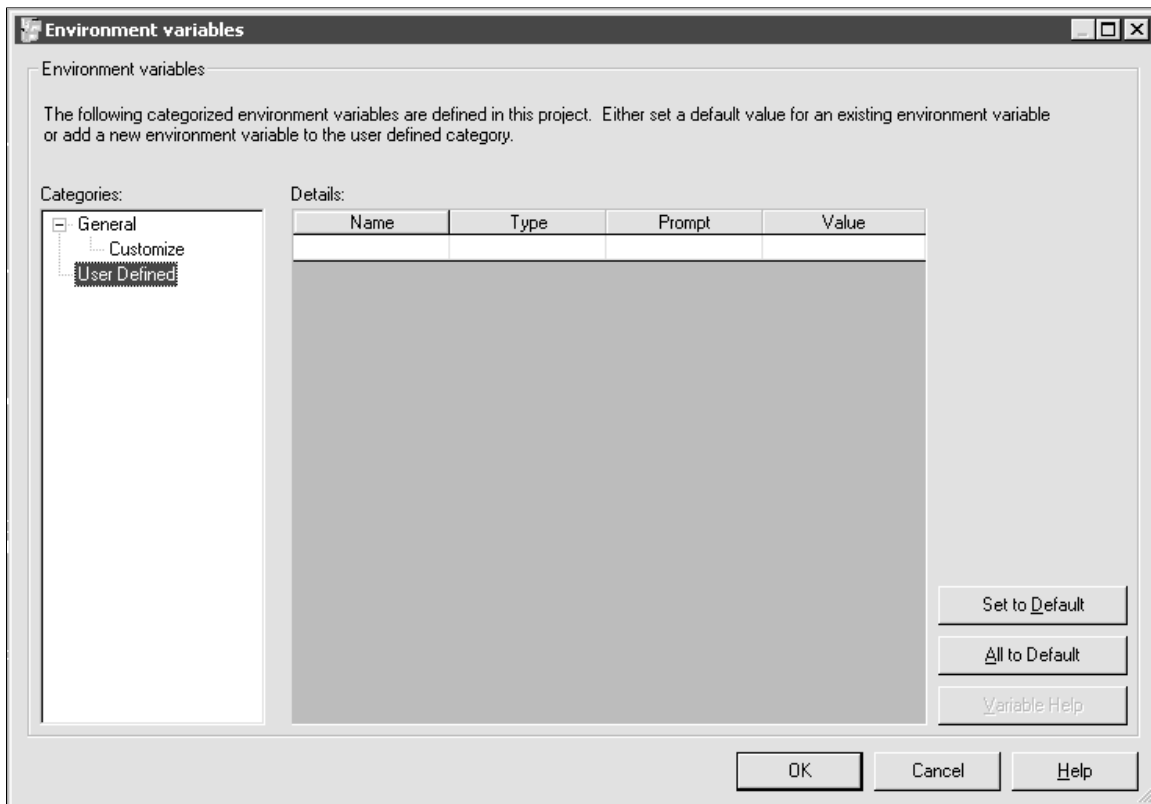
You can override this setting in individual jobs if desired.

Setting Environment Variables

Click the Environment button on the Project Properties - General tab to set project-wide environmental variables.

Image: Environment Variables Window

This example illustrates the Environment Variables Window.



DataStage Administrator enables you to create user-defined environment variables and assign default values for existing variables used throughout a project.

Changing an environment variable affects all of the jobs in the project. To change an environment variable for each job, leave the Value column empty and specify the variable value in a job parameter instead. You can also override the value when the job runs.

To set a default value for an environment variable, select the variable type from the Environment Variable Tree in the left pane, and then enter a value in the right pane.

To create a new variable, select User Defined in the Environment Variable Tree, and then enter a new variable name, prompt, and value in the right pane.

Click *Set to Default* to set the selected variable to its installed default value.

Click *All to Default* to set all currently visible variables to their installed default values.

Click *Variable Help* to get information about the selected variable.

Setting Environment Variables - Example

To configure the delivered environment parameters:

1. Open DataStage Administrator and select your project.
2. Note the project path name of the selected project and close DataStage Administrator.
3. Use the project path to navigate to the DSPARAM file.

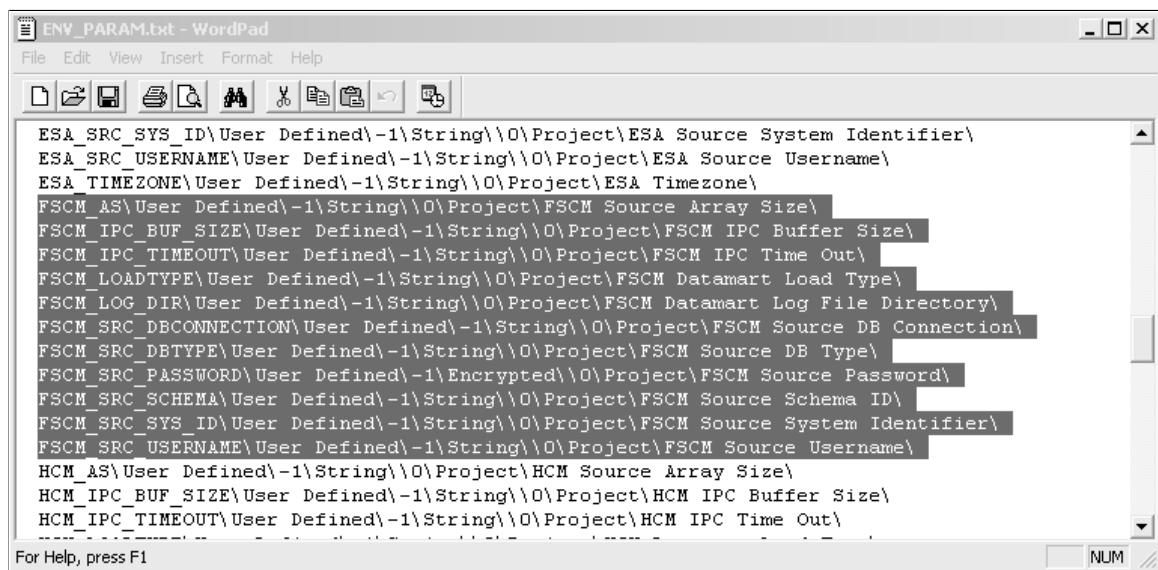
The DSPARAM file should be located in that folder.

4. Open the DSPARAM file in Notepad.
5. Search for [EnvVarDefns].
6. Open the ENV_PARAM.txt file, and then select and copy the contents of the ENV_PARAM.txt file.

You can copy specific entries based on the product.

Image: ENV_PARAM.txt File

This example illustrates the ENV_PARAM.txt File.



7. Paste the copied contents to the DSPARAM file.

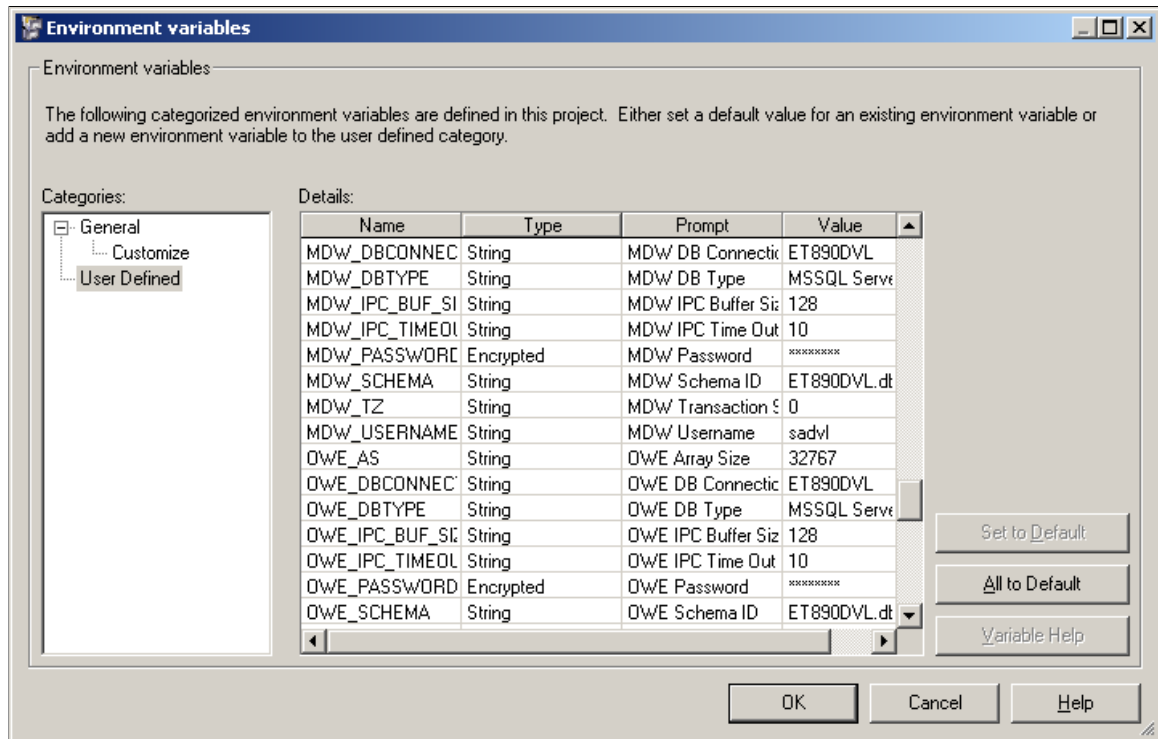
The contents should be pasted below the line that contains the [EnvVarDefns] text.

8. Save the DSPARAM file.
9. Open DataStage Administrator, navigate to the Environmental Variables window, and select the User-Defined category.

You should add values to the environment parameters to successfully run an ETL job.

Image: Environment Variables Window - Add Values

This example illustrates the Environment Variables Window - Add Values.

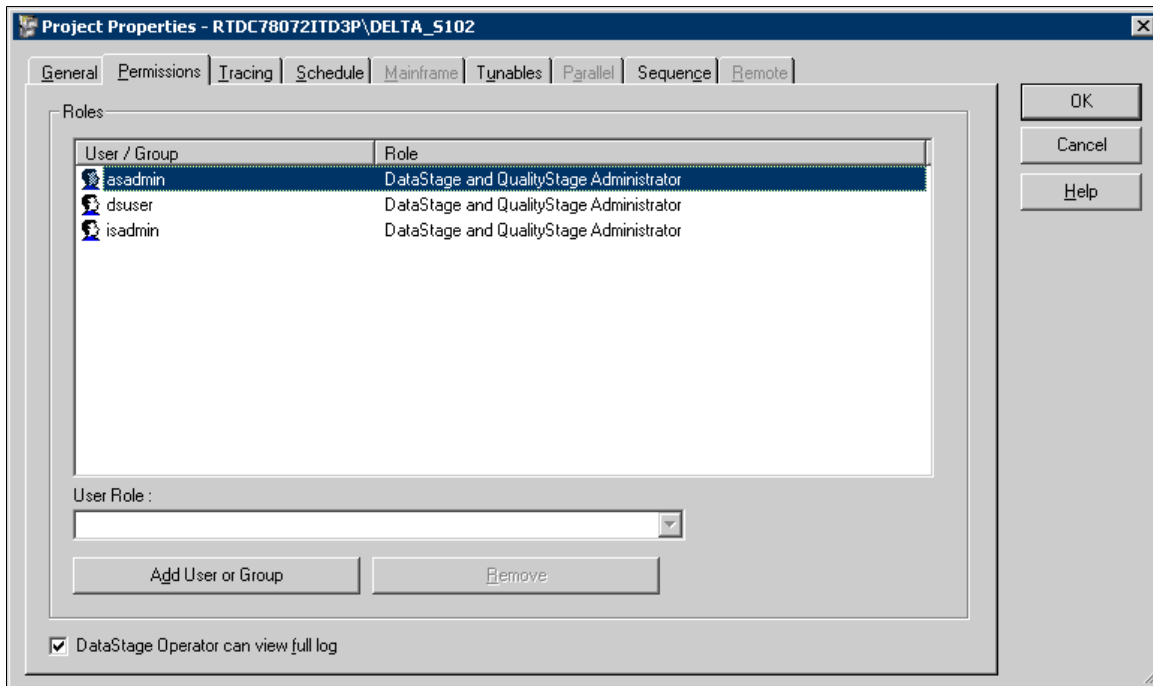


Project Properties - Permissions Tab

Access the Project Properties - Permissions tab:

Image: Project Properties - Permissions Tab

This example illustrates the Project Properties - Permissions Tab.



Before any user can access WebSphere DataStage they must be defined in the Suite Administrator tool as a DataStage Administrator or a DataStage User. As a DataStage administrator you can define whether a DataStage user can access a project, and if so, what category of access they have.

Use the Permissions tab to add groups and assign users to groups. These groups are in turn allocated the role of DataStage Administrator or DataStage User. Any users belong to an administrator group will be able to administer WebSphere DataStage. You can also grant user group access to a project and assign a role to the group.

When setting up users and groups, these still have to have the correct permissions at the operating system level to access the folders in which the projects reside.

The Permissions page contains the following controls:

- **Roles:** this window lists all the users and groups who currently have access to this project and lists their roles. Note that this window will always include users who have been defined as DataStage Administrators in the Suite Administrator tool, and you cannot remove such users from the list or alter their user role.
- **User Role:** this list contains the four categories of WebSphere DataStage user you can assign. Choose one from the list to assign it to the user currently selected in the roles window.
- **Add User or Group:** click this to open the Add Users/Groups dialog box in order to add a new user or group to the ones listed in the roles window.
- **Remove:** click this to remove the selected user or group from those listed in the roles window.

- DataStage Operator can view full log: by default this check box is selected, letting a WebSphere DataStage operator view both the error message and the data associated with an entry in a job log file. To hide the data part of the log file entry from operators, deselect this check box. Access to the data is then restricted to users with a developer role or better.

Project Properties - Tracing Tab

Access the Project Properties - Tracing tab:

Image: Project Properties - Tracing Tab

This example illustrates the Project Properties - Tracing Tab.



Use the Project Properties - Tracing tab to enable or disable tracing, and view or delete trace files.

Enabling tracing activity on the server helps diagnose project problems. By default, server tracing is disabled.

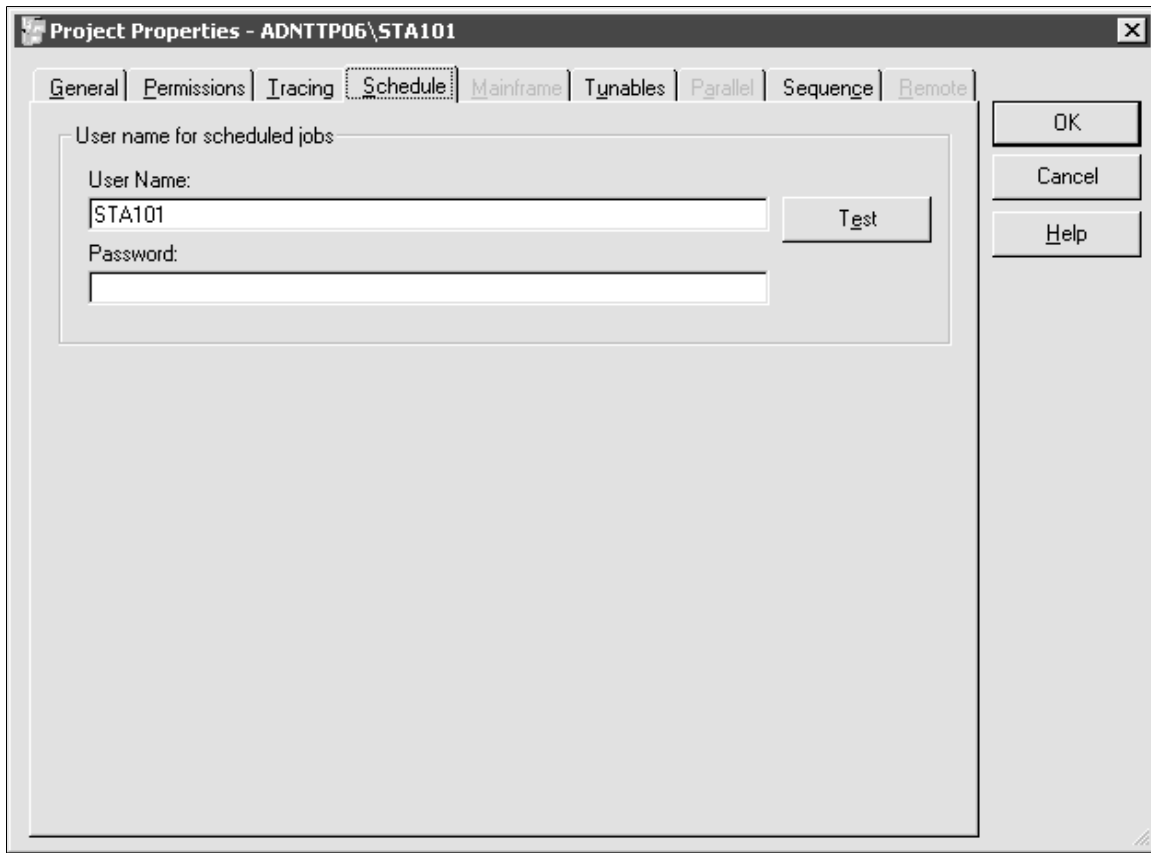
When you enable tracing, server activity attached to a specific project is written to trace files. Users can use the information saved in trace files to identify the cause of a project problem.

Project Properties - Schedule Tab

Access the Project Properties - Schedule tab:

Image: Project Properties - Schedule Tab

This example illustrates the Project Properties - Schedule Tab.



Use the Project Properties - Schedule tab to modify system authority user name for scheduling jobs. DataStage uses the Microsoft Windows Schedule service to schedule jobs. By default, jobs run under the Microsoft Windows system authority user name. However, this user name may not have sufficient rights, so you may need to change the assigned user name.

To verify that the user name exists, click the Test button. The system schedules and runs a job using the name that you entered.

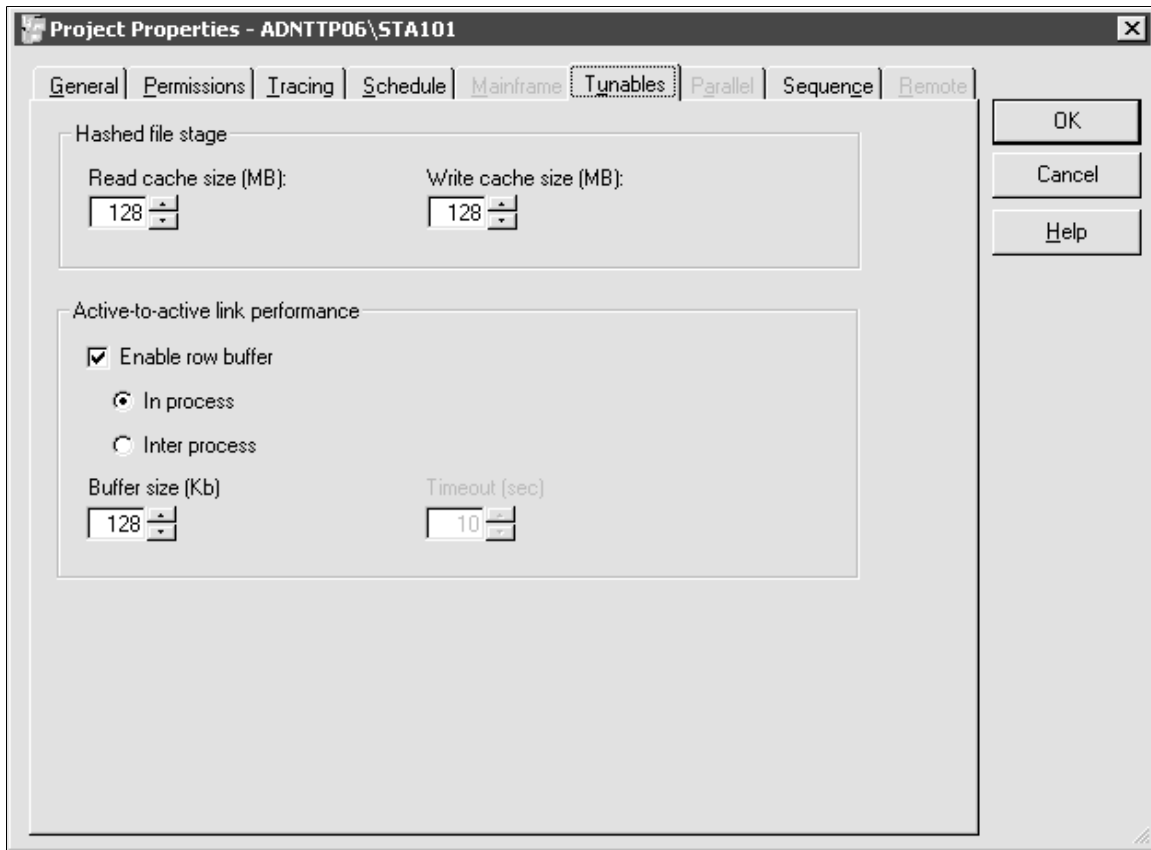
Note: The Schedule tab is only available on Microsoft Windows.

Project Properties - Tunables Tab

Access the Project Properties - Tunables tab:

Image: Project Properties - Tunables Tab

This example illustrates the Project Properties - Tunables Tab.



Use the Project Properties - Tunables tab to set up caching details for hashed file stages and row buffering to improve the performance of server jobs.

When data is referenced repeatedly, for instance in a lookup, storing the data in memory rather than on disk can improve performance. To support this performance improvement, when a hash file stage writes records to a hash file, the data can be cached rather than written to the hash file immediately. Similarly, when a hash file stage is reading a hash file, you can preload the file to memory, which makes subsequent access to the data faster. The hash file stage area of the Tunables tab enables you to adjust the sizes of both the read and write cache sizes.

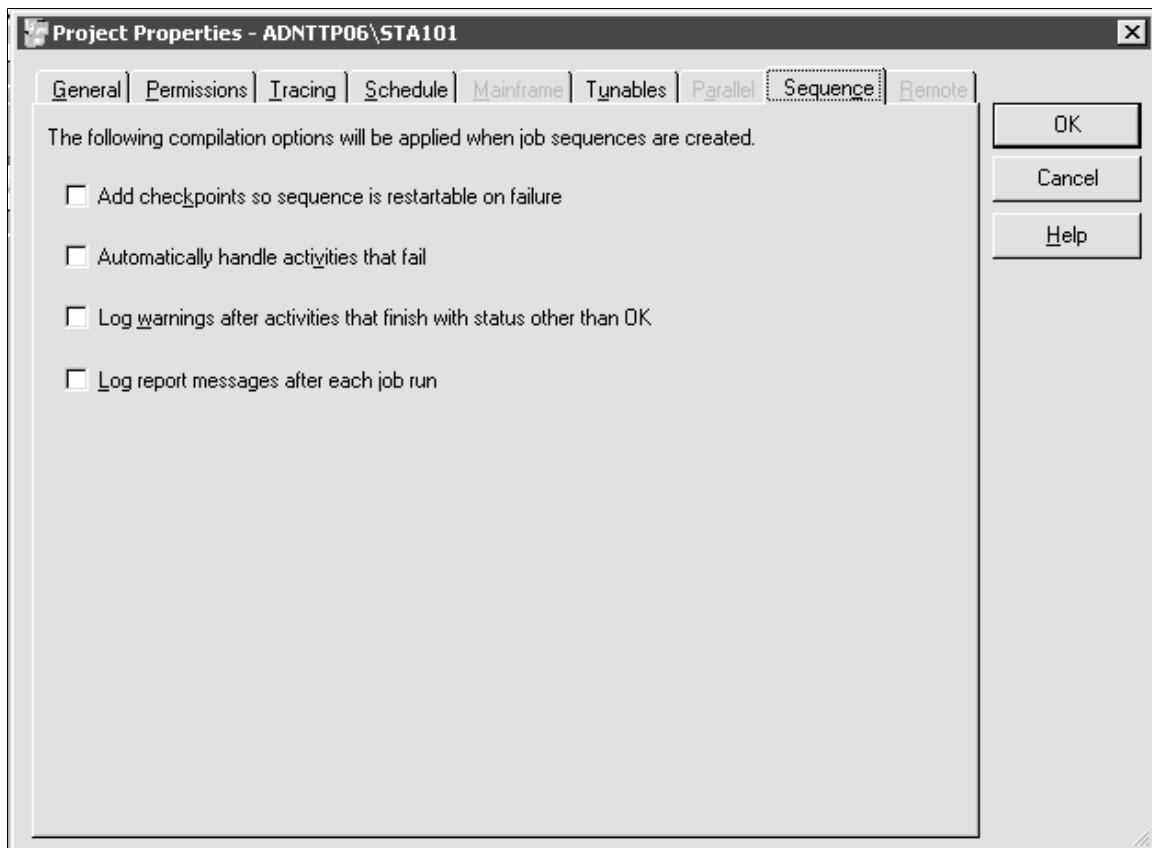
Another way to improve performance is with the use of row buffering. Row buffering enables connected active stages to pass data by using buffers (memory) rather than passing data row by row.

Project Properties - Sequence Tab

Access the Project Properties - Sequence tab:

Image: Project Properties - Sequence Tab

This example illustrates the Project Properties - Sequence Tab.



Use the Project Properties - Sequence tab to add checkpoints to a job sequence and enable automatic handling of failures during sequence runs.

You can insert checkpoints in job sequences to enable the sequence to be restarted if one of the jobs in the sequence fails. Checkpoints enable you to see where the problem is, fix it, and then rerun the sequence from the point at which it left off.

Chapter 9

Using DataStage Designer

Using DataStage Designer

These topics provide an overview of DataStage Designer and discuss how to:

- Manage Repository Objects
- Edit Object Properties
- Import and Export Repository Components
- Use Table Definitions
- Build DataStage Jobs
- Use Database and File Stages
- Add and Link Stages
- Compile and Run Jobs
- Edit Job Properties
- Use Expressions
- Create Constraints
- Use Hashed File Stages
- Use Job Sequencers
- Use DataStage BASIC

Note: This topic does not discuss all the features available for DataStage Designer. For a complete view of DataStage Designer functionality, please see the delivered IBM WebSphere documentation.

DataStage Designer Overview

The DataStage Designer is the primary interface to the metadata repository and provides a graphical user interface that enables you to view, edit, and assemble DataStage objects from the repository needed to create an ETL job.

An ETL job should include source and target stages. Additionally, your server job can include transformation stages for data filtering, data validation, data aggregation, data calculations, data splitting

for multiple outputs, and usage of user-defined variables or parameters. These stages allow the job design to be more flexible and reusable.

DataStage Designer enables you to:

- Create, edit, and view objects in the repository.
- Create, edit, and view data elements, table definitions, transforms, and routines.
- Import and export DataStage components, such as projects, jobs, and job components.
- Analyze the use of particular items in a project.
- Edit and view user-defined object properties.
- Create jobs, job sequences, containers, and job templates.
- Create and use parameters within jobs.
- Insert and link stages into jobs.
- Set stage and job properties.
- Load and save table definitions.
- Save, compile, and run jobs.

DataStage Designer Window

The DataStage Designer window, which is the graphical user interface used to view, configure, and assemble DataStage objects, contains the following components:

- *Repository Window*: Displays project objects organized into categories. By default, the Repository window is located in the upper left corner of the Designer window. The project tree displays in this pane and contains the repository objects belonging to a project.
- *Tool Palette*: Contains objects that you add to your job design, such as stage types, file types, database types, and processor objects. You can drag these objects from the Palette into the Diagram window. By default, this window is displayed in the lower left corner, of the Designer window. This window appears to be empty until you open or create a job.
- *Diagram Window*: Serves as the canvas for your job design. You drag, drop, and link stages and processor objects to create jobs, sequencers, and templates.
- *Property Browser*: Displays the properties of the currently selected stage of the job that is open in the Diagram window. By default, this window is hidden. To open it, select View, Property Browser from the menu bar, and then click a stage to see its properties.

The following diagrams show the layout of the DataStage Designer window components:

Image: DataStage Designer Window - Layout View

This diagram illustrates the layout view of the DataStage designer window.

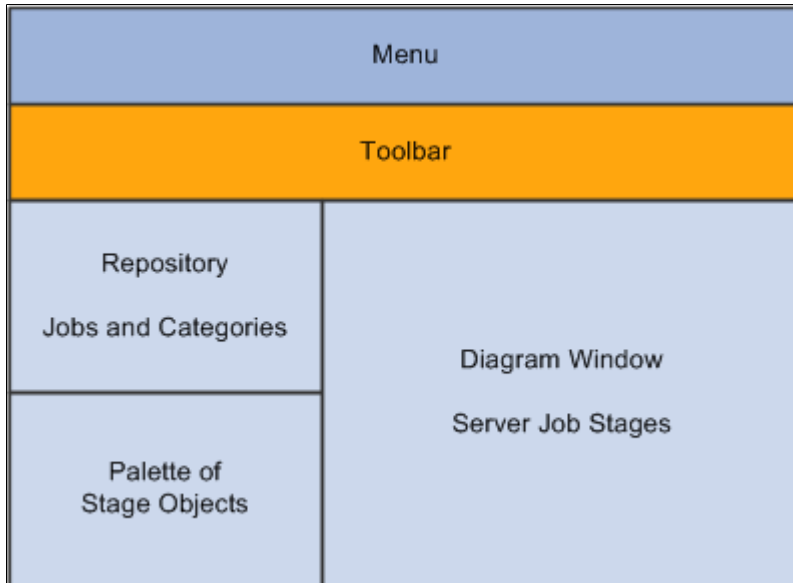
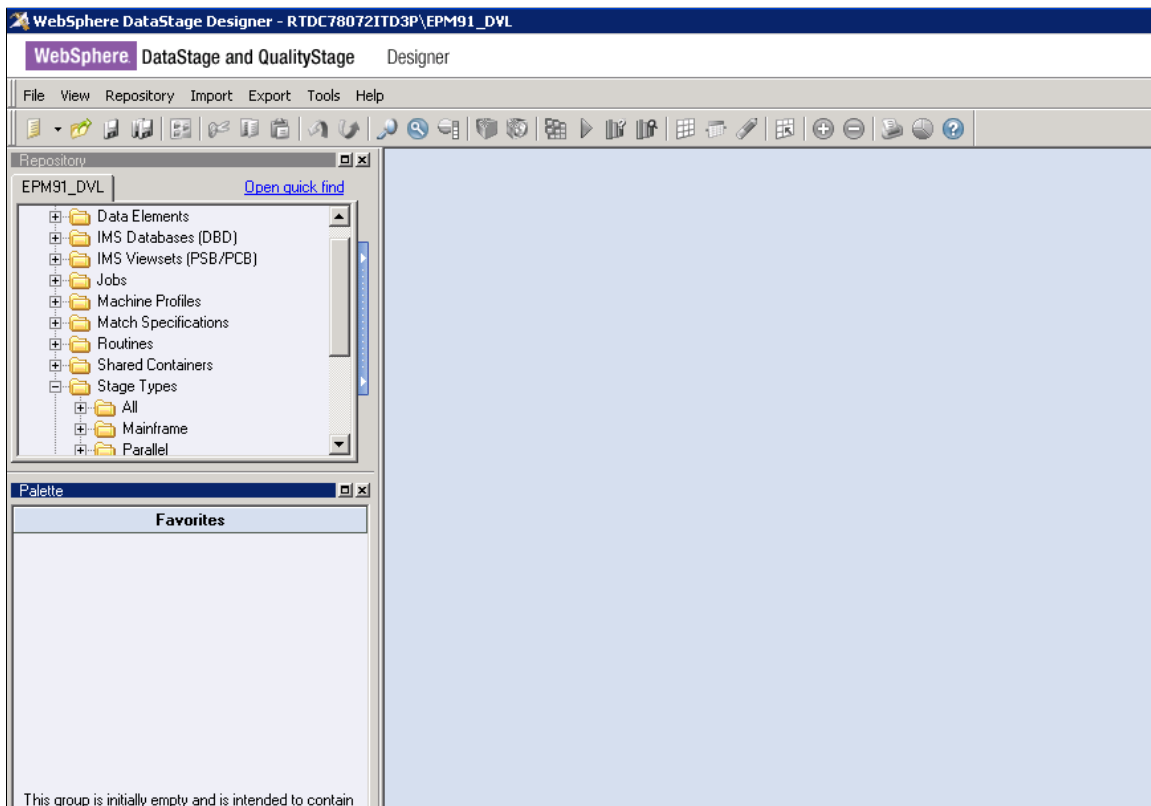


Image: DataStage Designer Window

This example illustrates the DataStage Designer Window.



The display area is in the right pane of the DataStage Designer window and displays the contents of a chosen object in the project tree.

By Default, the Designer window contains the Repository window, Tool Palette, and Diagram window. You can optionally view the Property Browser by selecting View, Property Browser from the menu bar .

The display of Designer windows and toolbars can be shown or hidden by selecting the appropriate option from the View menu. You can dock, undock, or rearrange the Designer windows.

Designer Menus

Most Designer menu items are also available in the toolbars. The following are some additional options that are available through the menus:

Designer Menu Item	Description
View, Customize Palette	Customize your palette.
View, Property Browser	Enables you to view and edit properties of a DataStage object.
Import	Enables you to import ETL projects, jobs, or other components that you export from another system, as well as DataStage components, such as table definitions, from text files or XML documents.
Export	Enables you to export DataStage objects in the form of text files with the file extension <i>.dsx</i> .
Tools, Run Multiple Job Compile	Enables you to compile all your jobs at the same time.
Tools, Run Director	Invoke the Director module, and log you into your project automatically.

Designer Toolbar

The Designer toolbar displays the following buttons:

Image: DataStage Designer Toolbar

This example illustrates the DataStage Designer Toolbar.



This table describes the Designer toolbar buttons:

Designer Toolbar Button	Description
New	Open the New window where you can open a new DataStage object.
New (arrow down button)	Display options associated with the New command on the toolbar.
Open	Display the Open window that enables you to open an existing or recently opened repository object.

Designer Toolbar Button	Description
Save	Save the current job or container.
Save All	Save all open jobs or containers.
Job Properties	Open the Job Properties window for the current job open in the Diagram window.
Cut	Cut a specific object or text and temporarily stores it.
Copy	Copy a specific object or text and temporarily stores it.
Paste	Paste the temporarily stored object or text.
Undo	Undo the last task performed.
Redo	Redo the last task performed.
Quick Find	Search for DataStage objects using the quick find feature.
Advanced Find	Search for DataStage objects using the advanced find feature.
Data Flow Analysis	Use this function to display the data lineage for a column definition to see where in the job design that the column definition is used, display the source of the data for selected column or columns, display the target for the data for selected column or columns.
Construct Local Container	Create a local job container.
Construct Shared Container	Create a shared container reusable by other jobs.
Compile	Compile the current job.
Run	Run the current job.
Grid Lines	Show or hide a grid in the Diagram window.
Link Markers	Show or hide markers on the links.
Toggle Annotations	Show or hide annotations in the diagram window. You enter annotations by dragging the Annotation object from the Palette.
Stage Validation errors	See visual cues for parallel jobs or parallel-shared containers. The visual cues display compilation errors for every stage on the canvas, without you having to actually compile the job. The option is enabled by default

Designer Toolbar Button	Description
Snap to Grid	When the grid is shown and Snap to Grid is enabled, align objects that you drag with the grid.
Zoom In	Magnify the diagram display.
Zoom Out	Shrink the diagram display.
Print	Print the current diagram window.
Generate Report	Generate an HTML report of a server, parallel, or mainframe job or shared container. You can view this report in a standard Internet browser.
Help on View	View context-sensitive help.

Debug Toolbar

The Debug toolbar provides basic functions for testing and troubleshooting your jobs.

The Debug toolbar can be accessed by selecting View, Debug and displays the following buttons:

Image: Debug Toolbar

This example illustrates the Debug Toolbar.



This table describes the Debug toolbar buttons:

Debug Toolbar Button	Description
Set target debug job	Enables you to select the job you want to debug.
Start/Continue Debugging	Start or stop running in debug mode.
Next Link	Run the job until you come to the next link.
Next Row	Run until you get to the next row.
Stop Job	Stop the job run.
Set debug Job Parameters	Set job parameters.
Edit Breakpoints	Change breakpoints (pauses that you have inserted into the run).
Toggle Breakpoints	Enable or disable breakpoints.

<i>Debug Toolbar Button</i>	<i>Description</i>
Clear All Breakpoints	Clear breakpoints.
View Job Log in Director	Open the job log in the Director module.
Show/Hide Debug Window	Display or hide the debug window.

All of the Debug toolbar options are also available from the Debug menu.

Managing Repository Objects

You can use DataStage Designer to view job categories, which serve to organize repository objects.

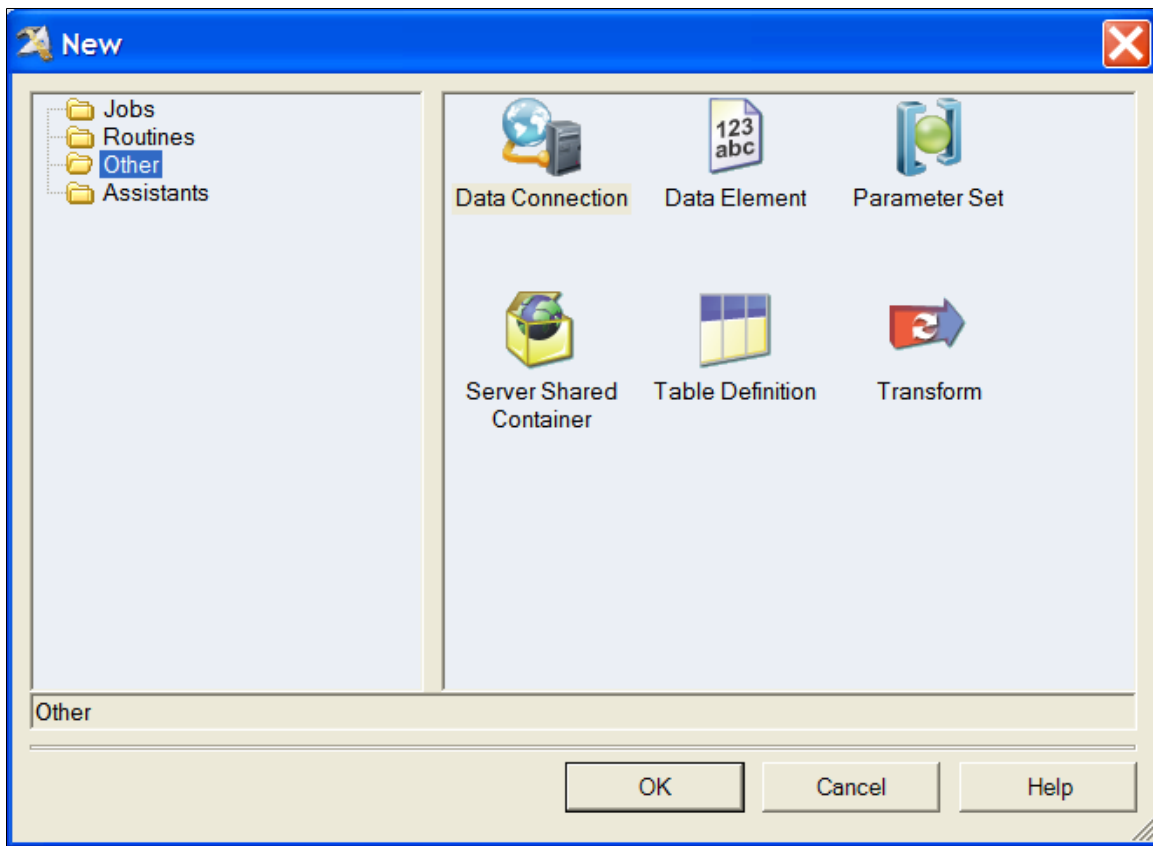
You can view the following repository objects within a job category:

- Data Elements.
- Jobs.
- Routines.
- Shared Containers.
- Stage Types.
- Table Definitions.
- Transforms.

You can also create new repository objects:

Image: Create new Data Element

This example illustrates the Create new Data Element.



You can also copy, rename, edit, delete, or move an item using the File menu commands or the item level shortcut menu.

Editing Object Properties

Object properties consist of descriptive information and other types of information, depending on the object type. Using DataStage Designer you can:

Image: DataStage Designer - Object Properties

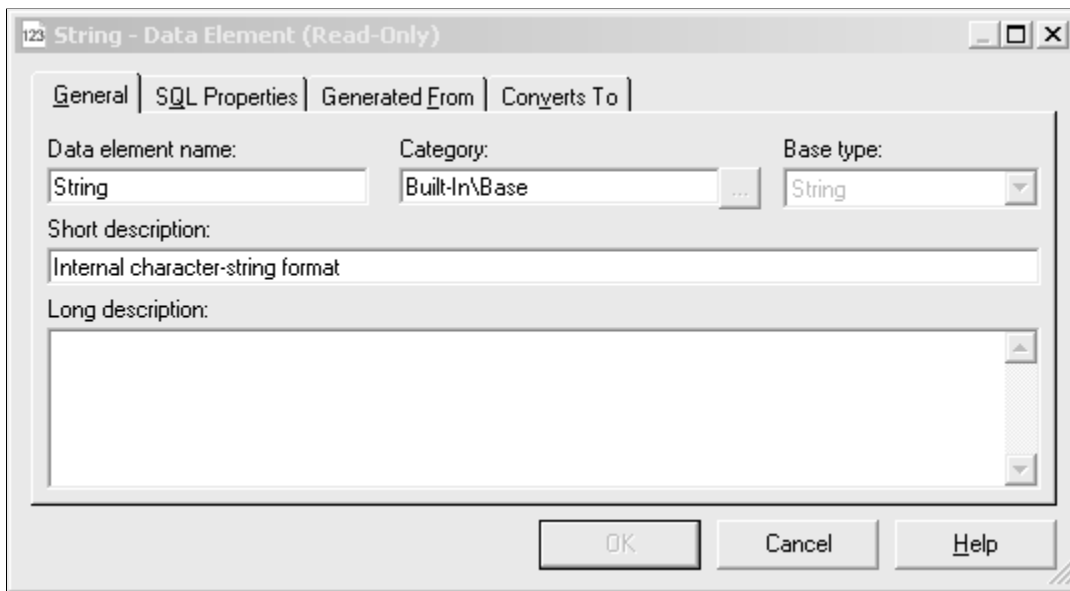
This graphic illustrates the tasks that can be performed using object properties in the DataStage Designer.



The following is an example of an object property for the String data element:

Image: Object Property - String Data Element

This example illustrates the Object Property - String Data Element.



Editing Job and Job Sequence Properties

DataStage Designer enables you to:

Image: DataStage Designer - Job/Job Sequence Properties

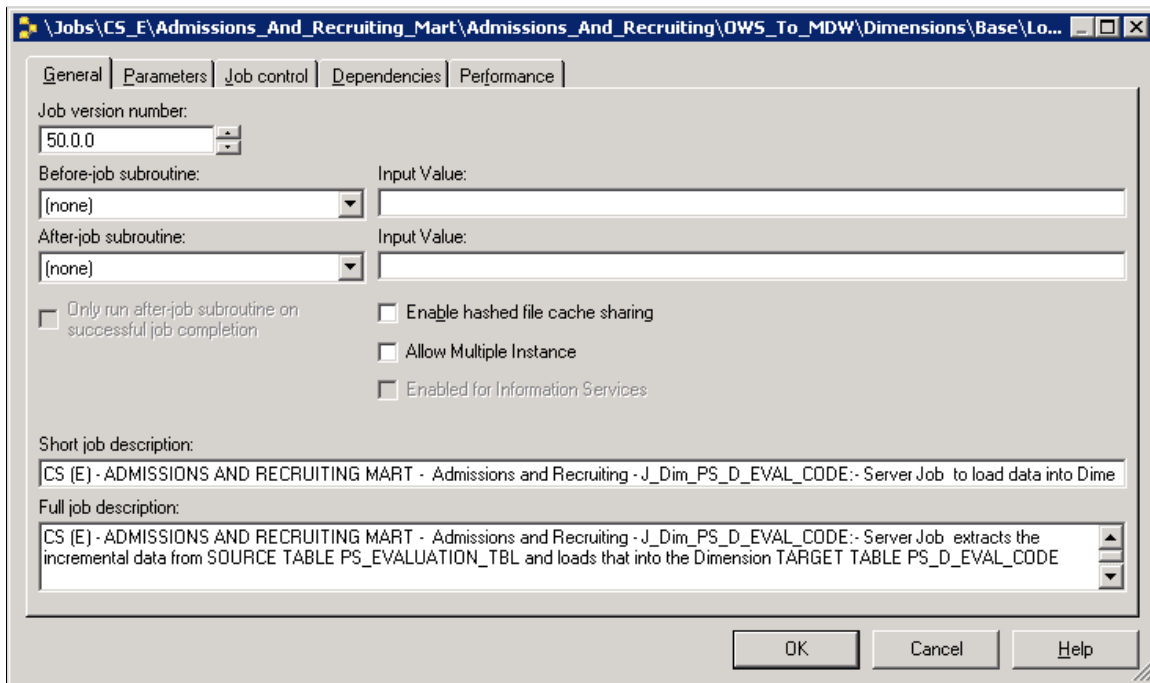
This graphic illustrates the tasks that can be performed for job and job sequence properties in the DataStage designer.



The following is an example of a server job property:

Image: Server Job Properties

This example illustrates the Server Job Properties.

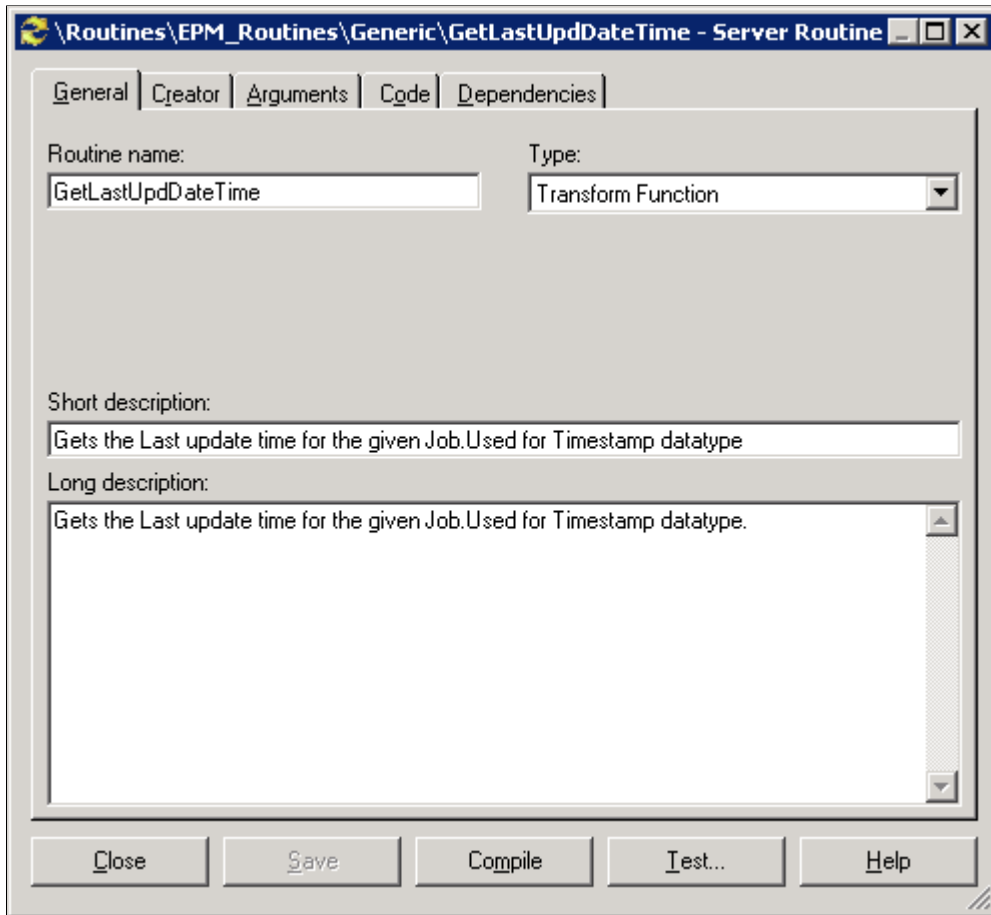


Editing Server Routines

You can create, edit, or view server routines using the Routine window. Argument names in built-in routines cannot be changed.

Image: Server Routine Properties

This example illustrates the Server Routine Properties.



The following components are classified as routines:

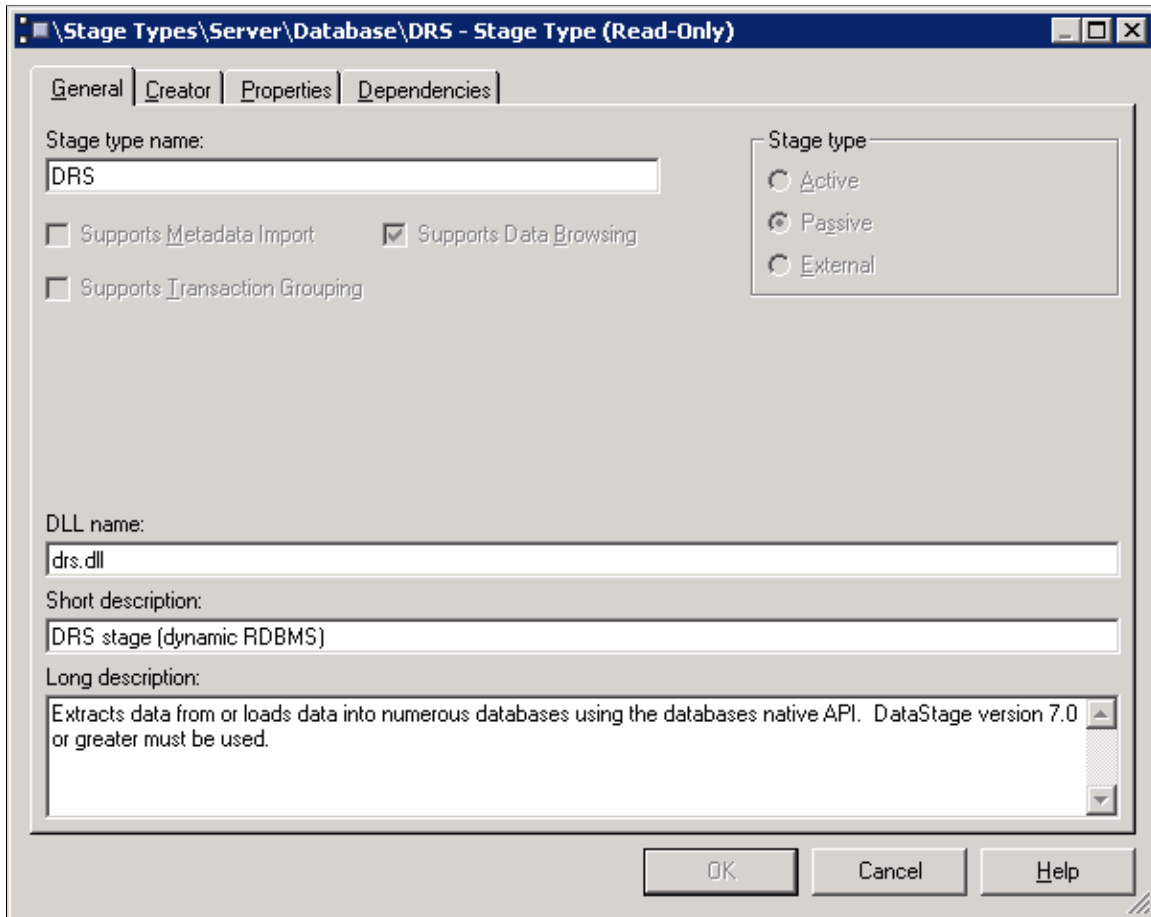
- Transform functions.
- Before/after subroutines.
- Custom UniVerse functions.
- ActiveX (OLE) functions.

Editing the Stage Type

The Stage Type category in the project tree contains all the stage types that you can use in your jobs. Properties of WebSphere DataStage's pre-built stages are read-only.

Image: DRS Stage Type Properties

This example illustrates the DRS Stage Type Properties.



You can create or edit object properties for the following stage types:

- Custom Plug-in Stages.
- Parallel Job Custom Stages.

DataStage Designer enables you to create and register plug-in stages to perform specific tasks that the built-in stages do not support. You need to register custom plug-in stages before you can use them. In addition, DataStage Designer enables you to create custom parallel stage types.

Specifying Table Definitions

DataStage Designer enables you to:

Image: DataStage Designer - Table Definitions

This graphic illustrates the tasks that can be performed for table definitions in the DataStage Designer.

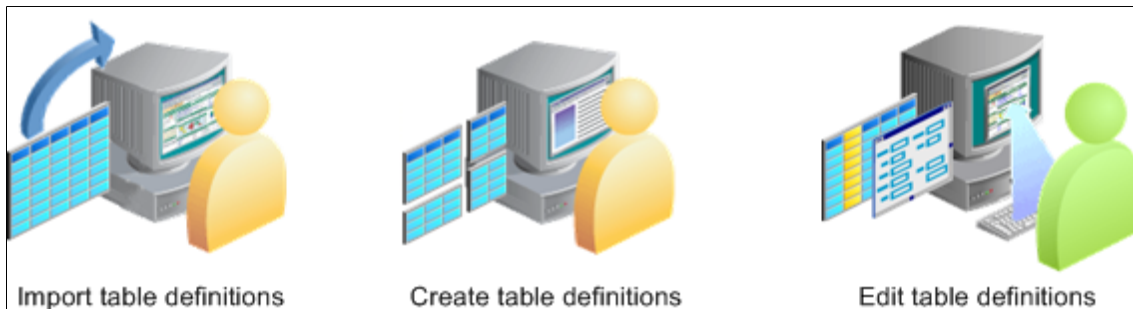


Table definitions:

- Specify the data structure used by each stage in a DataStage job.
- Are stored in the repository and are shared by all jobs in a project.
- Are required for each data source and data target.
- Can be imported, manually created, or edited.

Importing and Exporting Repository Components

Using the DataStage Designer import and export facilities enable you to move jobs or other components between projects. You can also move projects, jobs, or components from one system to another. In addition, you can import components from text files or XML documents, and you can export to XML documents. XML documents can be used as a convenient way to view descriptions of repository objects using a web browser.

Importing

The DataStage Designer import facility enables you to import:

- ETL projects, jobs, or other components that you export from another system.
- DataStage components, such as table definitions, from text files or XML documents.

You can use the Import facility to import table definitions from a variety of file types, including sequential files, ODBC, and XML.

Image: Import Menu Item

This example illustrates the Import Menu Item.

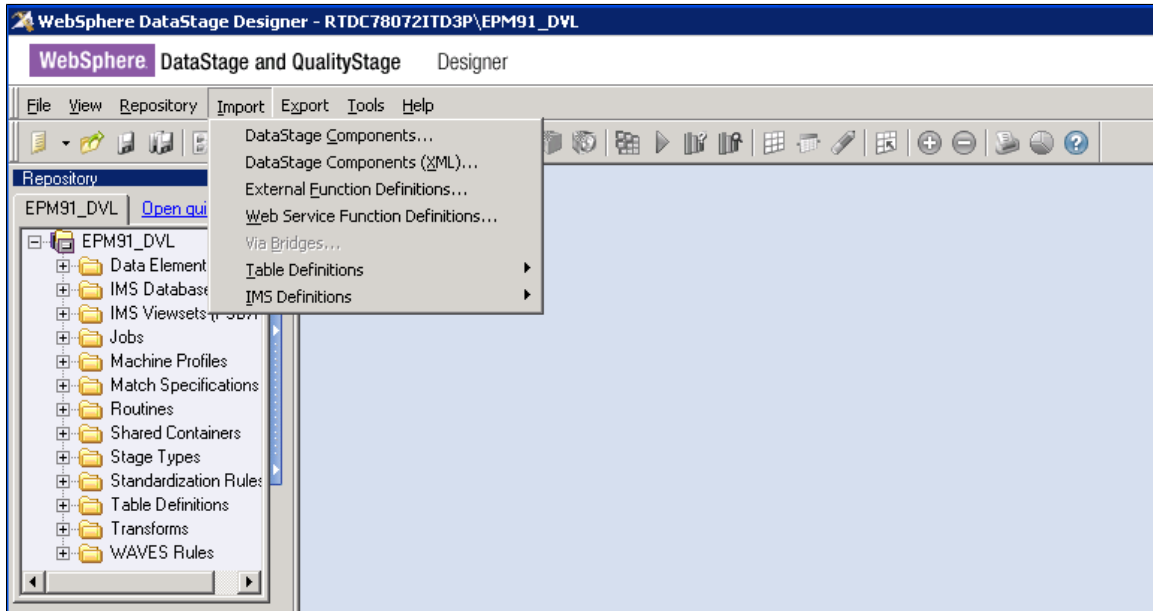
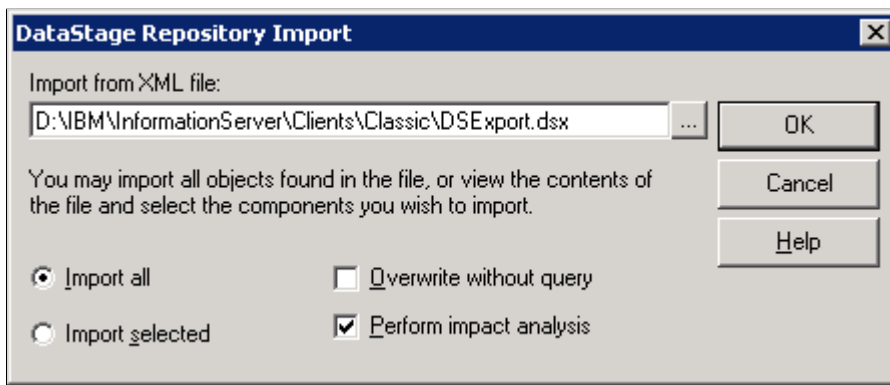


Image: Import DSX Component

This example illustrates the Import DSX Component.



Exporting

The DataStage Designer export facility enables you to export:

- ETL projects, jobs, or other components.
- Jobs or other components to XML documents.
- Job executables.
- Package server jobs using the Packager Wizard.

When you export projects or components, by default they are stored in text files with the file extension .dsx. You can also export to XML files by selecting the appropriate check box in the Export window. You also have the option to append the exported items to an existing file.

Image: Export Menu Item

This example illustrates the Export Menu Item.

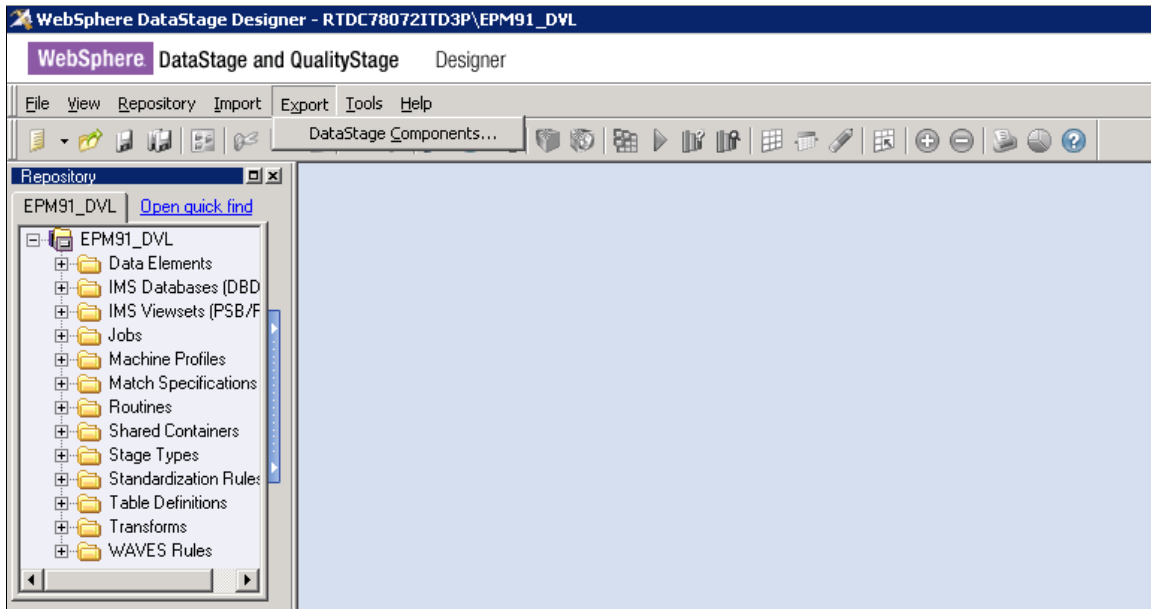
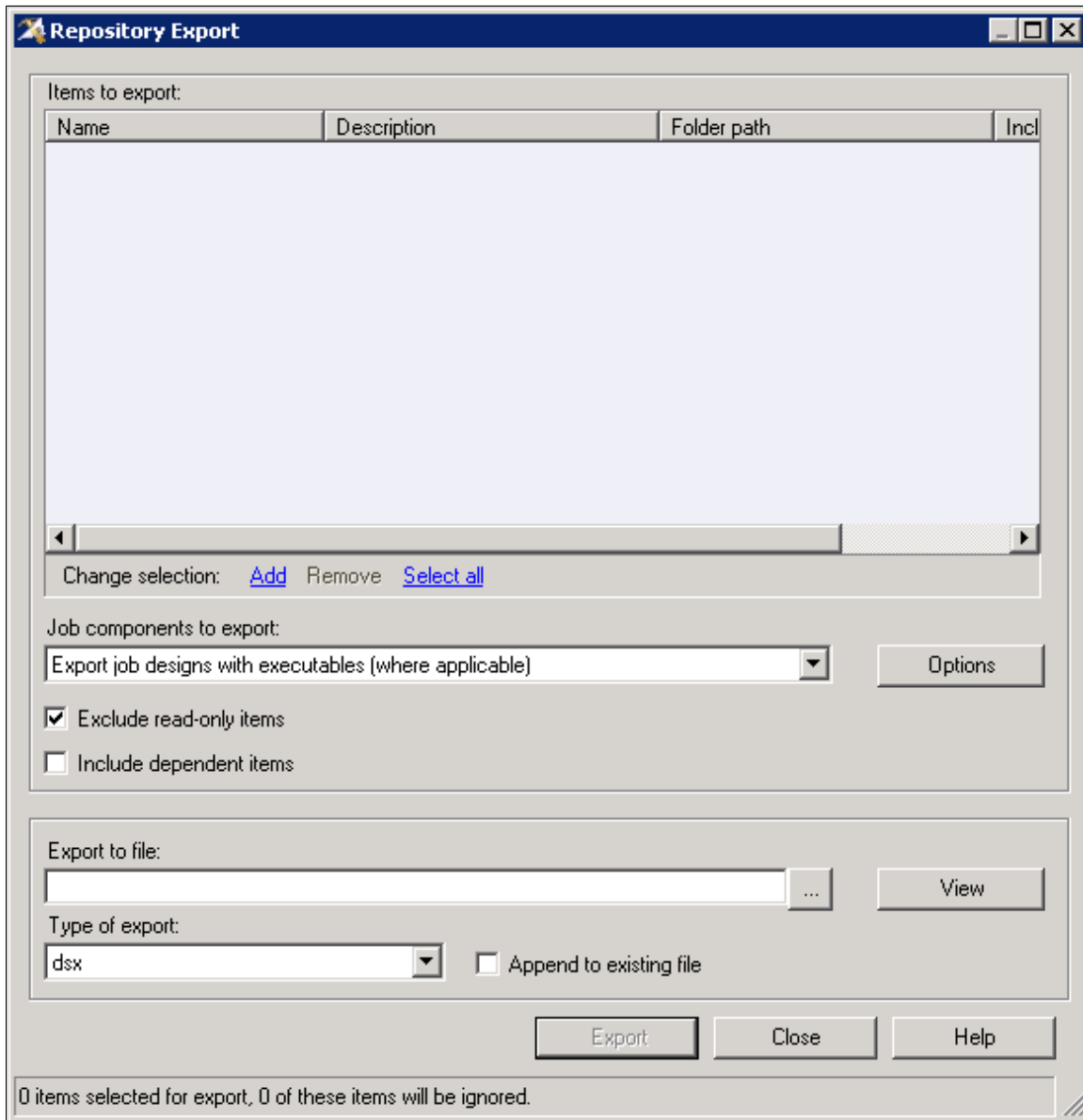


Image: Export - Repository Export window

This example illustrates the Export - Repository Export window.



Using Table Definitions

Table definitions are:

- DataStage components that specify the metadata used at each stage of a job.
- Stored in the Repository.
- Shared by all the jobs in a project.

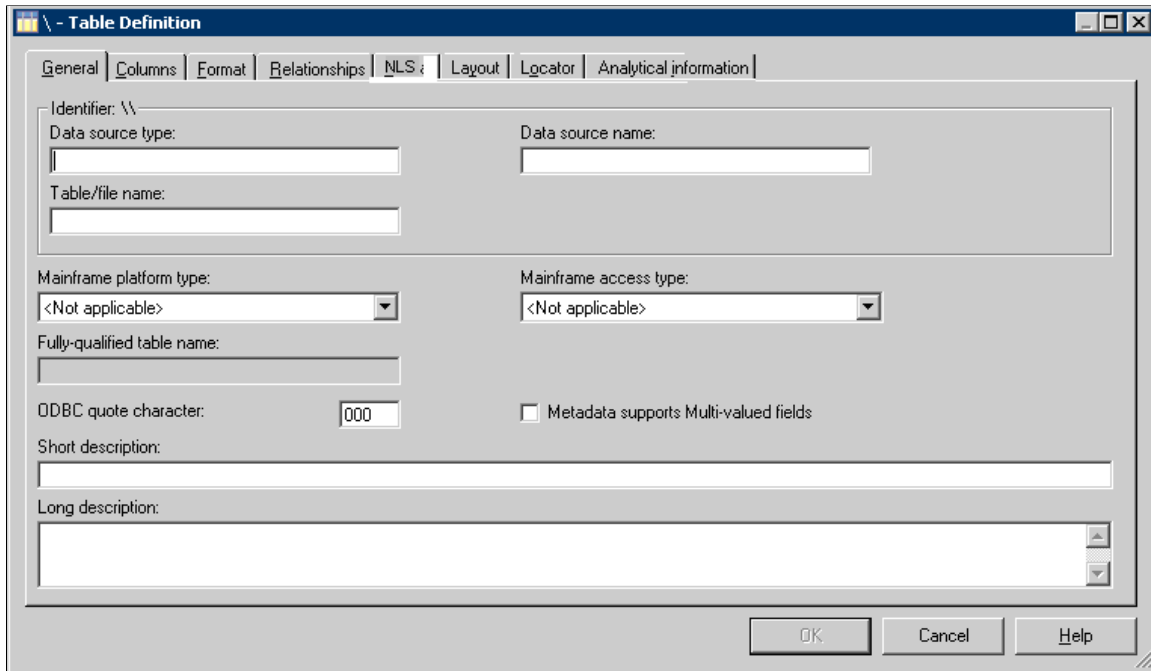
You need a table definition for each data source stage or data target stage you use in your job. You can import, create, or edit a table definition using DataStage Designer.

Creating Table Definitions

To create a new Table Definition, select New Table Definition from the Table Definition menu. The Table Definition window appears:

Image: Table Definition Window

This example illustrates the Table Definition Window.



The Table Definition window has these tabs:

- General

The General tab contains the data source type, data source name, table or file name, and other general information about the table definition.

- Columns

The Columns tab contains a grid displaying the column definitions for each field in the table definition.

- Format

The Format tab contains file format parameters for sequential files used in DataStage jobs

- Relationships

The Relationships tab displays the details of any relationship this table definition has with other tables, and allows you to define new relationships.

- NLS

If NLS is enabled, the NLS tab is enabled and contains the name of the map to use for the table definition.

- Layout

The Layout tab displays the schema format of the column definitions in a table.

- Locator

Using the Locator tab you can view and edit the data resource locator associated with the table definition. The data resource locator is a property of the table definition that describes the real world object from which the table definition was imported. The labels and contents of the fields in this window depend on the type of data source or target from which the locator originates.

- Analytical Information

The Analytical Information tab displays information about the table definition generated by Information Analyzer.

- Parallel

The Parallel tab displays detailed format information for the defined metadata for parallel jobs.

Importing Table Definitions

You can directly import a table definition from a source or target database. You can import table definitions from ODBC data sources, plug-in stages, UniVerse tables, hash files, UniData files, or sequential files.

In the DataStage Designer Repository window, right-click on Table Definitions. Select Import.

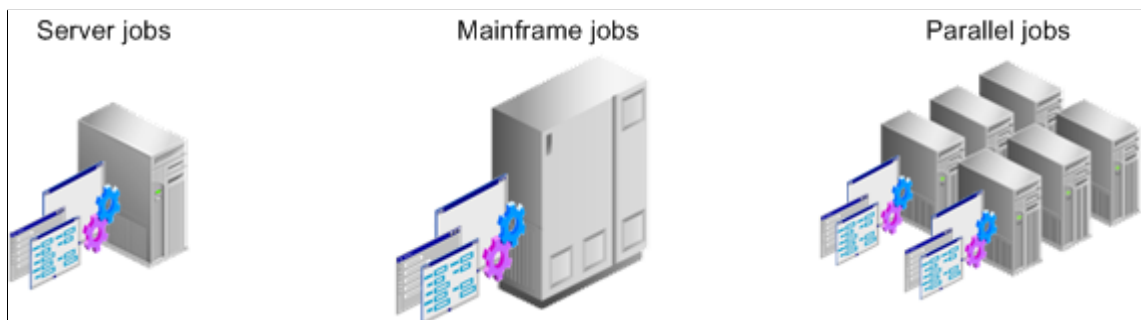
You can select the type of table definition data source from the available options.

Building DataStage Jobs

DataStage provides these types of jobs:

Image: DataStage Jobs

This graphic illustrates the jobs that can be created.



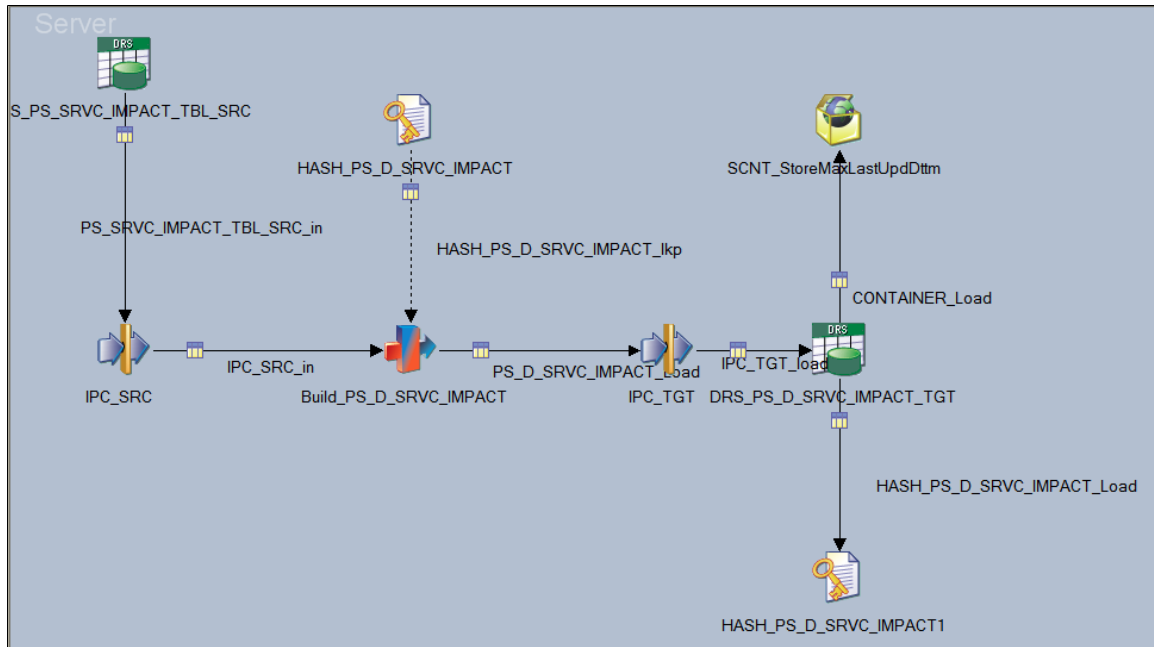
- Server Jobs: Run on the DataStage Server.
- Mainframe Jobs: Available only if you have installed Enterprise MVS Edition and uploaded it to a mainframe, where they are compiled and run.

- Parallel Jobs: Available only if you have installed the Enterprise Edition and run on DataStage servers that are SMP, MPP, or cluster systems.

The following is an example of one of the delivered Campus Solutions Warehouse server jobs:

Image: Sample Delivered Job - J_Dim_PS_D_SRVC_IMPACT

This example illustrates the Sample Delivered Job - J_Dim_PS_D_SRVC_IMPACT.



Perform the following steps to build a job:

1. Define optional project-level environment variables in DataStage Administrator.
2. Define optional environment parameters.
3. Import or create table definitions, if they are not already available.
4. Add stages and links to the job to indicate data flow.
5. Edit source and target stages to designate data sources, table definitions, file names, and so on.
6. Edit transformer and processing stages to perform various functions, include filters, create lookups, and use expressions.
7. Save, compile, troubleshoot, and run the job.

Using Database and File Stages

Database stages represent data sources or data targets.

DataStage provides three types of stages:

- Server Job Database Stages

- Server Job File Stages
- Dynamic Relational Stages
- Processing Stages

Each stage has a set of predefined and editable properties.

Server Job Database Stages

The following are some of the delivered server job database stages:

- ODBC
- UniVerse
- UniData
- Oracle
- Sybase

Server Job File Stages

The delivered server job file stages are:

- Sequential file
- Hashed file
- Complex flat file
- Folder

Dynamic Relational Stages

Dynamic Relational Stages (DRS):

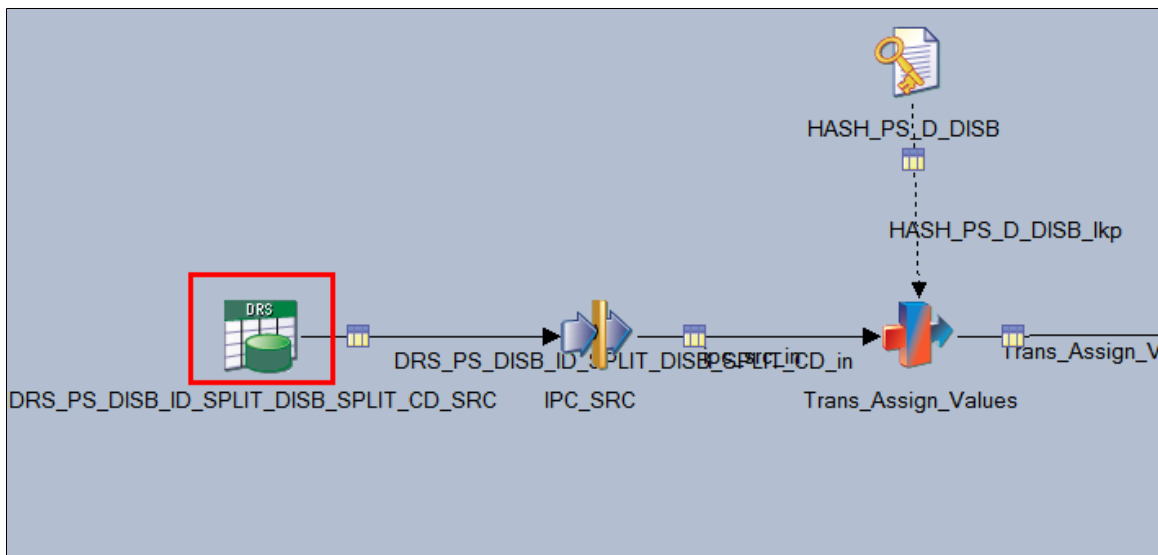
- Read data from any DataStage stage.
- Read data from any supported relational database.
- Write to any DataStage stage.
- Write to any supported relational database.

PeopleSoft-delivered ETL jobs use the DRS stage for all database sources or targets. This is represented in the Database group as "Dynamic RDBMS." When you create jobs, it is advisable to use the DRS stage rather than a specific type such as DB2 because a DRS will dynamically handle all of PeopleSoft supported database platforms.

The following example shows a DRS database stage in a delivered Campus Solutions Warehouse job:

Image: DRS Stage

This example illustrates the DRS Stage.



A DRS database stage supports the following relational databases:

- DB2/UDB
- Informix
- Microsoft SQL Server
- Oracle
- Sybase

A DRS database stage also supports any generic ODBC interface.

Editing the DRS Stage

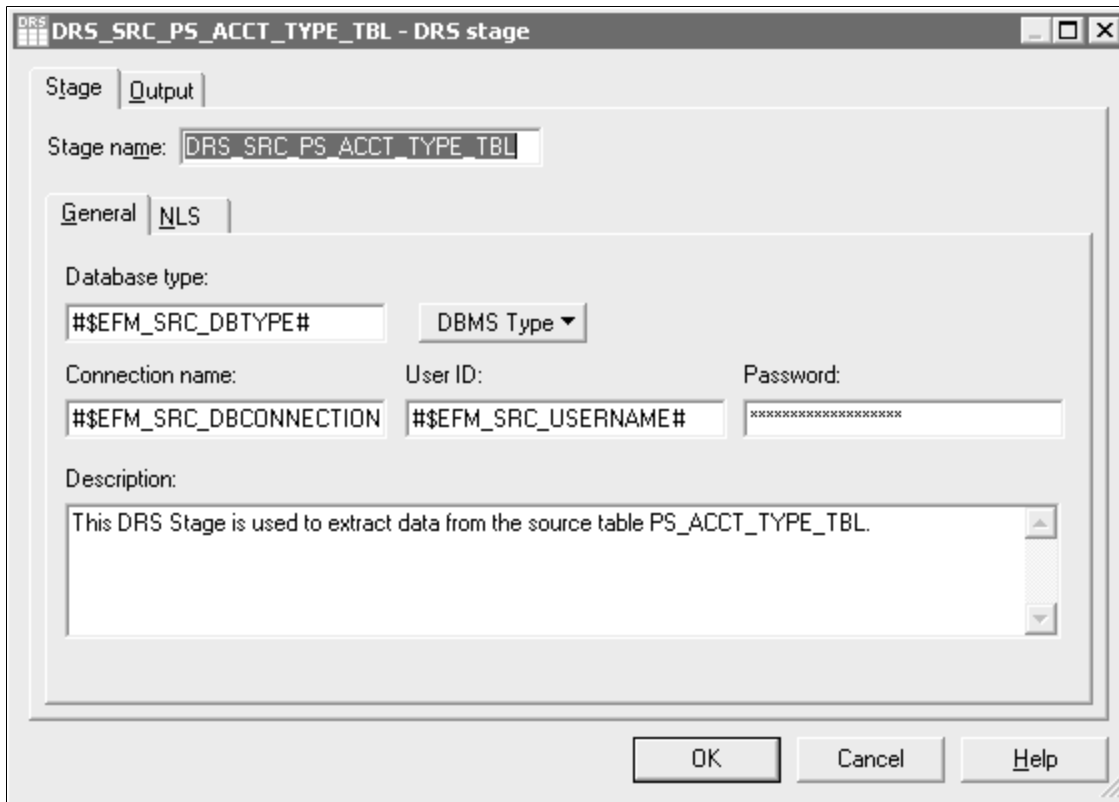
You edit the DRS properties using the DRS stage window.

1. Double-click the DRS stage to open the DRS stage window.

- The DRS stage window contains two main tabs: the Stage tab and the Output tab:

Image: DRS Stage Window

This example illustrates the DRS Stage Window.



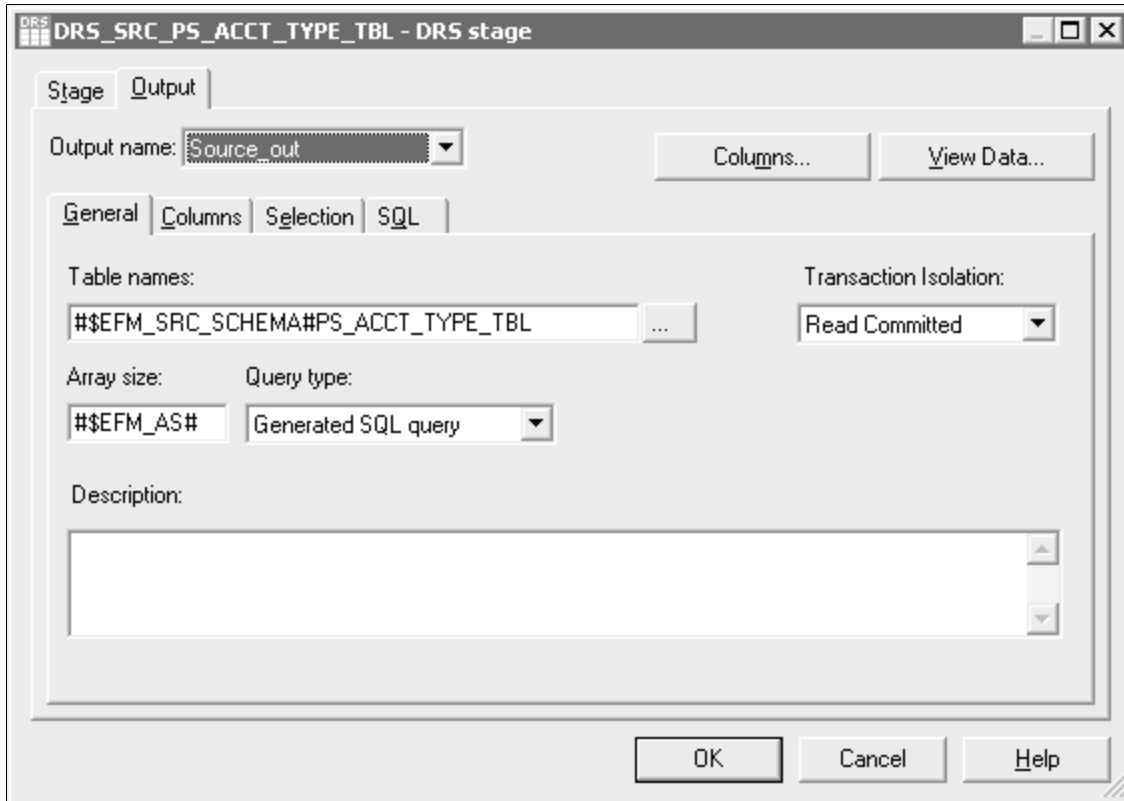
The Stage tab contains two tabs: the General tab and the NLS tab. In the General tab, you define the source database type, database or connection name, user ID, and password used in that connection. The previous example uses environment variables to define the values of these fields. If environment variables or job parameters were not used in the DRS stage, you define the actual values in these fields.

Entering Information in the Output Window

The Output tab contains General, Columns, Selection, and SQL tabs:

Image: DRS Stage Output Window

This example illustrates the DRS Stage Output Window.



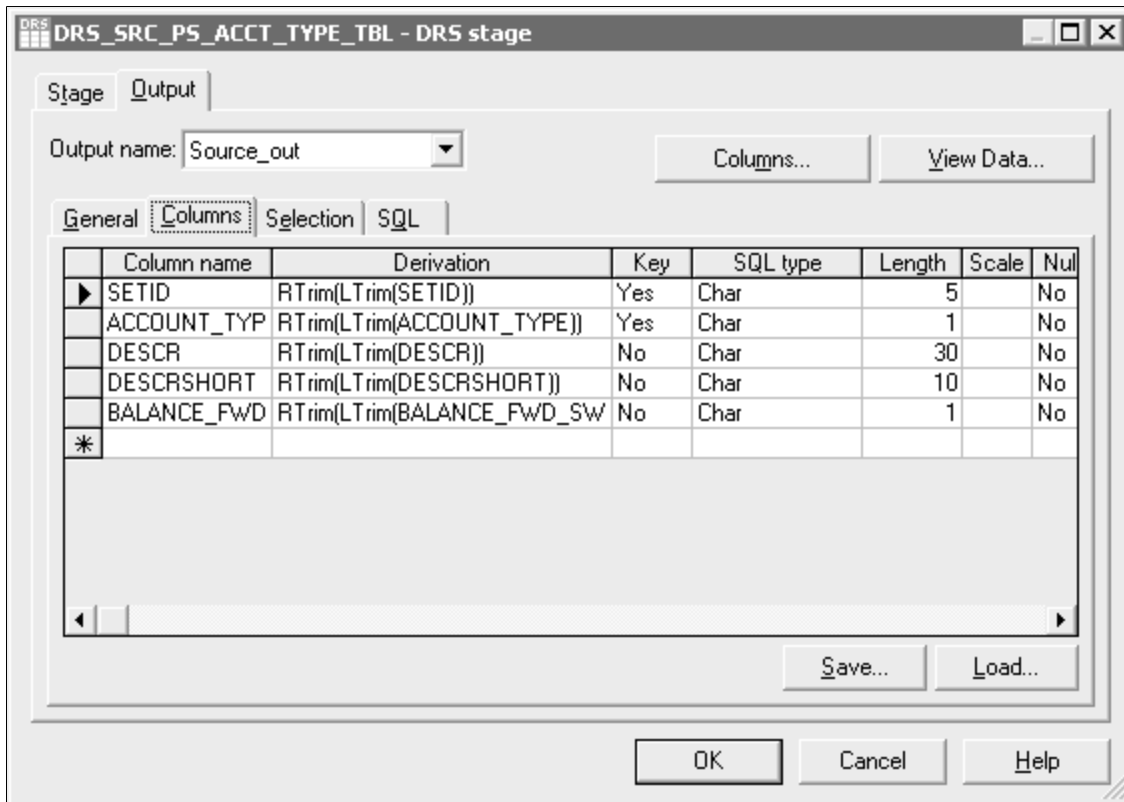
In this example, the table name listed is the source of the data that this stage uses.

The Columns window shown below enables you to select which columns of data you want to pass through to the next stage. When you click the Load button, the system queries the source table and populates the grid with all the column names and properties. You can then delete rows that are not needed.

The following example shows the Columns window:

Image: DRS Stage Output Window - Columns Tab

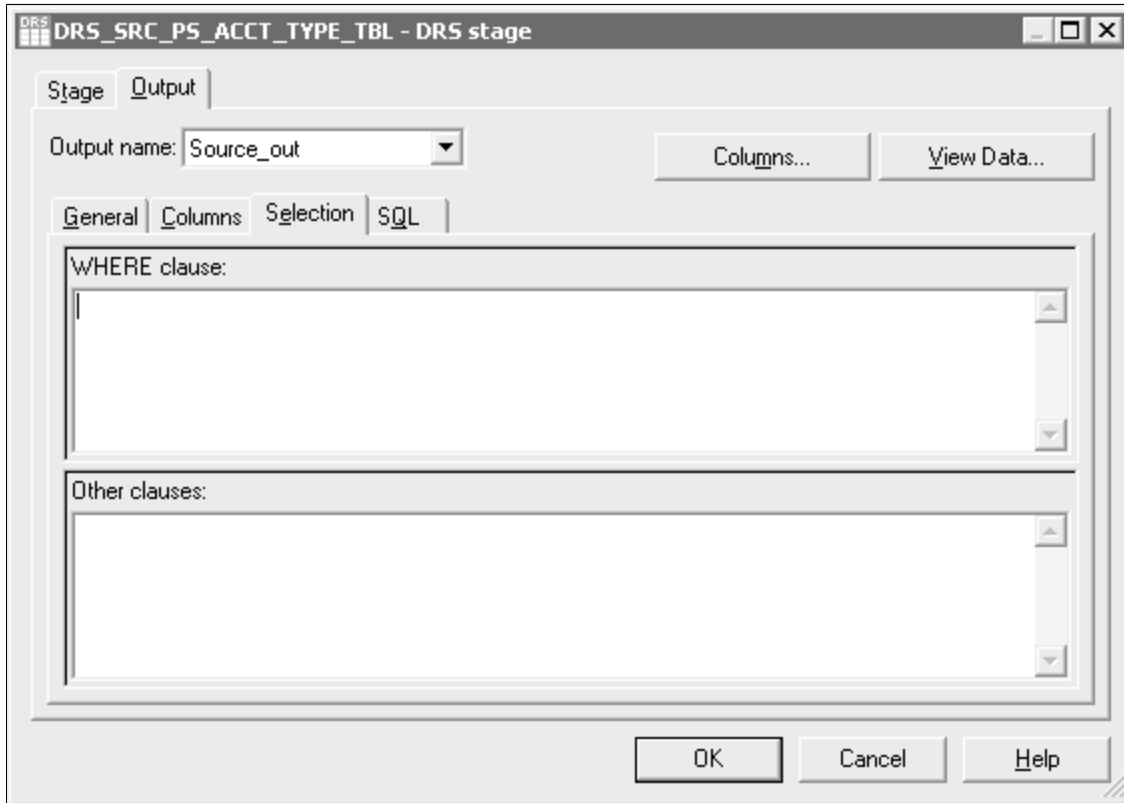
This example illustrates the DRS Stage Output Window - Columns Tab.



The Selection window enables you to enter a Structured Query Language (SQL) WHERE clause that specifies conditions when fetching data from tables.

Image: DRS Stage Output Window - Selection Tab

This example illustrates the DRS Stage Output Window - Selection Tab.

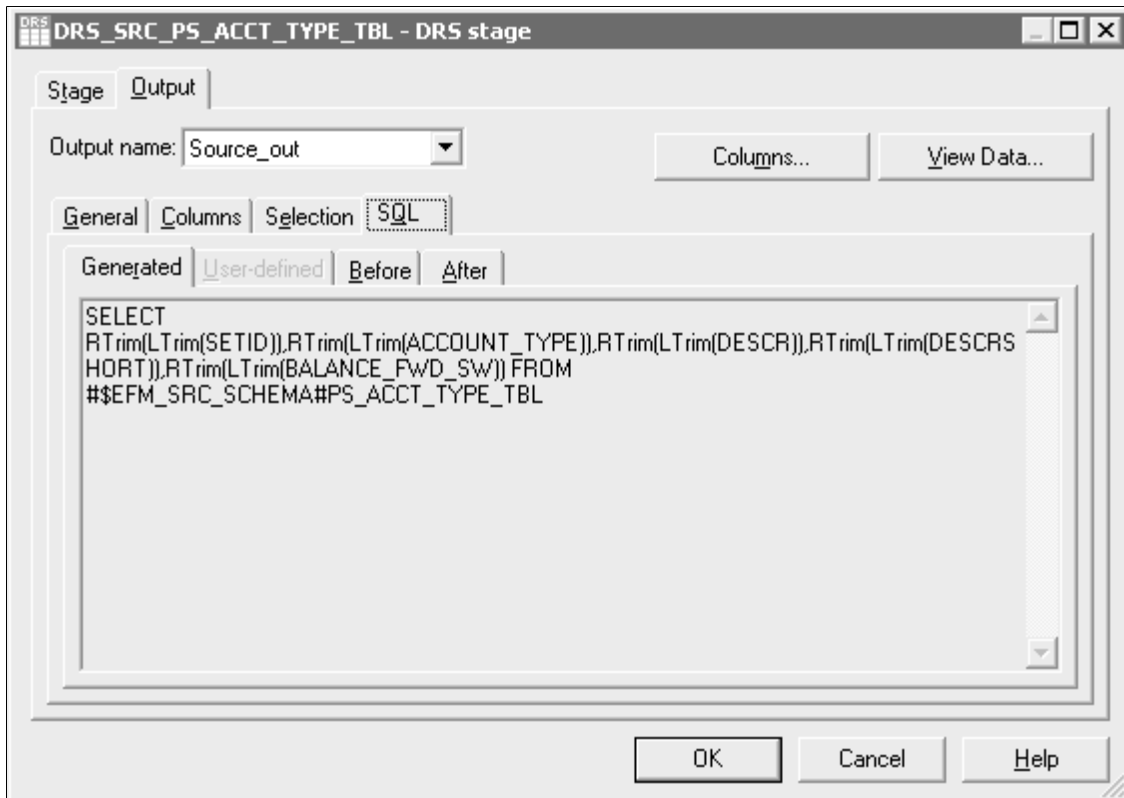


Entering a WHERE clause in the Selection window is optional.

The following shows the SQL tab of a DRS stage:

Image: DRS Stage Output Window - SQL Tab

This example illustrates the DRS Stage Output Window - SQL Tab.



The SQL tab contains the SQL statement used for the current stage.

Window Element	Usage
Generated	Shows the SQL SELECT statement that is automatically generated by this stage. It is read-only.
Before	Enter optional SQL statements executed before the stage processes job data rows. This does not appear in every plug-in.
After	Enter optional SQL statements executed after the stage processes job data rows. This does not appear in every plug-in.

Note: You can define SQL in a DRS Stage.

Processing Stages

DataStage Processing Stages:

- Reads the data from the source.
- Processes, transforms, or converts the data read from the source.

- Writes the processed data to the target.

Processing Stage Types

This table describes the different types of Processing Stages:

<i>Processing Stage</i>	<i>Description</i>
Transformer	Transformer stages perform transformations and conversions on extracted data.
Aggregator	Aggregator stages group data from a single input link and perform aggregation functions such as COUNT, SUM, AVERAGE, FIRST, LAST, MIN, and MAX.
FTP	FTP Stages transfer files to other machines.
Link Collector	Link Collectors collect partitioned data and pieces them together.
Interprocess	An InterProcess (IPC) stage is a passive stage which provides a communication channel between WebSphere DataStage processes running simultaneously in the same job. It allows you to design jobs that run on SMP systems with great performance benefits.
Pivot	Pivot, an active stage, maps sets of columns in an input table to a single column in an output table.
Sort	Sort Stages allow you to perform Sort operations.

Transformer Stages

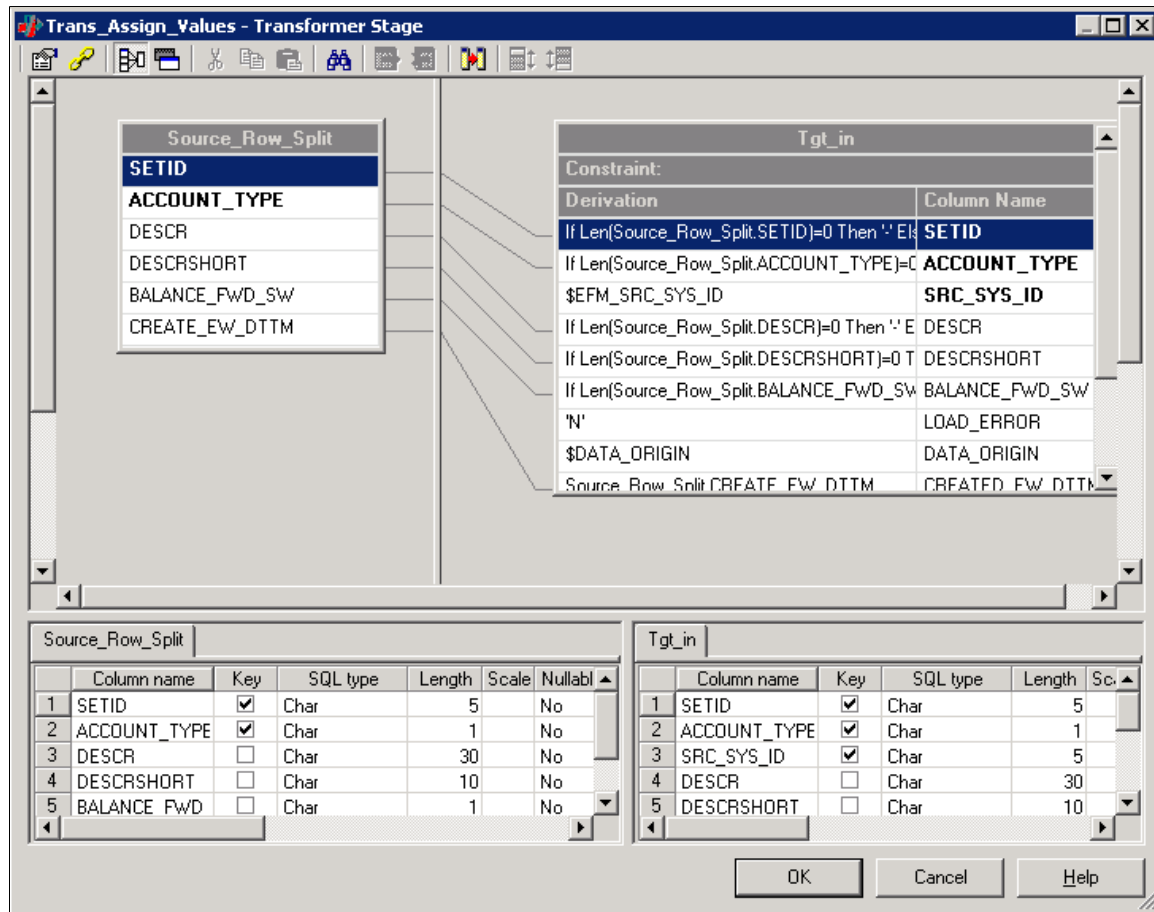
Transformer stages enable you to:

- Add, delete, or move columns.
- Apply expressions to data.
- Use lookups to validate data.
- Filter data using constraints.
- Edit column metadata and derivations.
- Define local stage variables, and before-stage and after-stage subroutines.
- Specify the order in which the links are processed.
- Pass data on to either another transformer stage, or to a target stage.

The following is an example of a delivered Transformer Stage (Trans_Assign_Values Stage):

Image: Trans_Assign_Values Transformer Stage

This example illustrates the Trans_Assign_Values Transformer Stage.



Creating Transformer Stages

You create a transformer stage by opening the Processing group in the palette, selecting the Transformer stage, and clicking in the Diagram window. After creating links to connect the transformer to a minimum of two other stages (the input and output stages), double-click the Transformer icon to open the Transformer window.

In the example above, two boxes are shown in the upper area of the window representing two *links*. Transformer stages can have any number of links with a minimum of two. Hence, there could be any number of boxes in the upper area of the window. Labeling your links appropriately makes it easier for you to work in the Transformer Stage window.

The lines that connect the links define how the data flows between them. When you first create a new transformer, you link it to other stages, and then open it for editing. There will not be any lines connecting the Link boxes. These connections can be created manually by clicking and dragging from a particular column of one link to a column in another link, or by selecting the Column Auto-Match button on the toolbar.

Using the Transformer Stage Toolbar

The following buttons appear on the Transformer Stage toolbar:

Image: Transformer Stage Toolbar

This example illustrates the Transformer Stage Toolbar.



This table describes the buttons provided with the Transformer Stage toolbar

Transformer Toolbar Button	Usage
Stage Properties	<p>Define stage inputs and outputs when you link the transformer with other stages.</p> <p>Specify before-stage and after-stage subroutines (optional).</p> <p>Define stage variables.</p> <p>Define order in which input and output links are processed if there is more than one input or output link.</p>
Constraints	Enter a condition that filters incoming data, allowing only the rows that meet the constraint criteria to flow to the next stage.
Show All or Selected Relations	If you have more than two links in the transformer, you can select one link and click this button to hide all connection lines except for those on the selected link. With only two links present, clicking this button hides or displays all connections.
Show/Hide Stage Variables	Show or hide a box that displays local stage variables that can be assigned values in expressions, or be used in expressions.
Cut, Copy, Paste, Find/Replace	These are standard Windows buttons.
Load Column Definition	Load a table definition from the repository, or import a new one from a database.
Save Column Definition	Save a column definition in the repository so that it can be used in other stages and jobs.
Column Auto-Match	Automatically sets columns on an output link to be derived from matching columns on an input link. You can then go back and edit individual output link columns where you want a different derivation.
Input Link Execution Order	Order the reference links. The primary data link is always processed first.
Output Link Execution Order	Order all output links.

Adding and Linking Stages

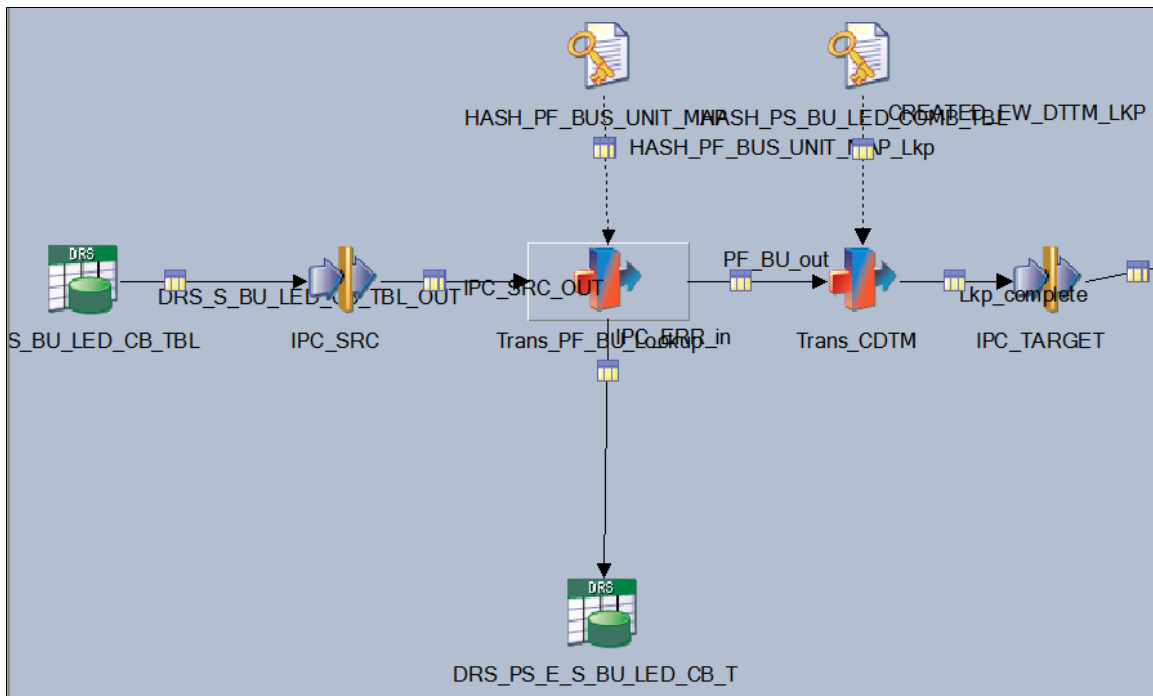
Stages represent inputs, outputs, and transformations within a job. Links join the stages together and show the flow of data within the job.

You add stages and links to a job by clicking the stage type or link in the palette and then clicking in the diagram window.

The following example shows a job that contains stages and links:

Image: Sample Job with Stages and Links - J_BASE_PS_BU_LED_COMB_TBL

This example illustrates the Sample Job with Stages and Links - J_BASE_PS_BU_LED_COMB_TBL.



A stage typically has at least one input or one output. However, some stages can have multiple inputs and output to more than one stage.

Different types of job have different stage types. The stages that are available in the DataStage Designer are dependent on the job type that is currently open in the DataStage Designer.

Adding Stages

To add a stage to a job, click a stage type in the palette, and click in the Diagram window.

The stages are located as follows:

- Database stages are located in the Database palette group.
- File stages are located in the File palette group.
- Processing stages are located in the Processing group.

This group includes the Transformer and Pivot stages used in PeopleSoft-delivered jobs.

If the link is red, then the link is broken. Start and end the drag motion in the center of each stage to ensure that you have linked the stages correctly.

Adding Links

To add a link between stages, you click the Link object in the General palette group, and then click and drag the cursor from one stage to another.

Another option is to right-click on one stage and drag the link to another stage.

By default, new links are named. However, we recommend that you rename all of your links to reflect their purpose and avoid confusion when you are editing transformers and stage properties.

Compiling and Running Jobs

Before running a job you must always:

Image: Compiling a Job

This graphic illustrates that you must save and compile jobs prior to running them.



To compile a job, click the Compile button on the DataStage Designer toolbar. After compiling the job, the result appears in the display area. If the result of the compilation is *Job successfully compiled with no errors*, you can schedule or run the job. If an error is displayed, you can click the Show Error button to highlight the stage where the problem occurs. Ensure that you have specified all the input and output column definitions, directory paths, file names, and table names correctly.

Criteria Checked when Compiling Jobs

The link to the source data stage is called the *primary link*. All other input links are called *reference links*.

During compilation, the following criteria in the job design are checked:

- **Primary Input:** If you have more than one input link to a Transformer stage, the compiler checks that one is defined as the primary input link.
- **Reference Input:** If you have reference inputs defined in a Transformer stage, the compiler checks that these are not from sequential files.

- **Key Expressions:** If you have key fields specified in your column definitions, the compiler checks that there are key expressions joining the data tables.
- **Transforms:** If you have specified a transform, the compiler checks that this is a suitable transform for the data element.

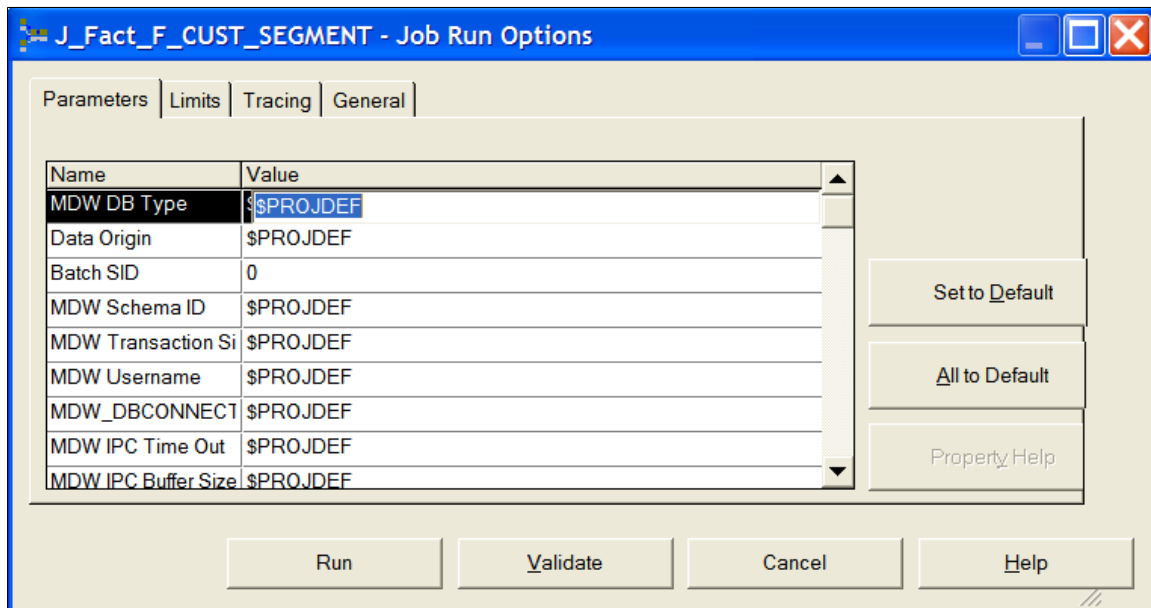
Specifying Job Run Options

After compiling jobs, they become executable. The executable version of the job is stored in your project along with your job design.

To run a job, click the Run button on the DataStage Designer toolbar. After clicking the Run button the Job Run Options window appears, where you can specify information on running a server job.

Image: Job Run Options Window

This example illustrates the Job Run Options Window.



In the Parameters tab, you enter specific parameter values for the job. You specify job parameters in the job properties window. You can create job-specific parameters or use an environment variable defined in DataStage Administrator. When running jobs, the parameters required to run the job are displayed in the Parameters tab of the Job Run Options window. If you specified default values in your job properties, these are displayed in the Parameters tab.

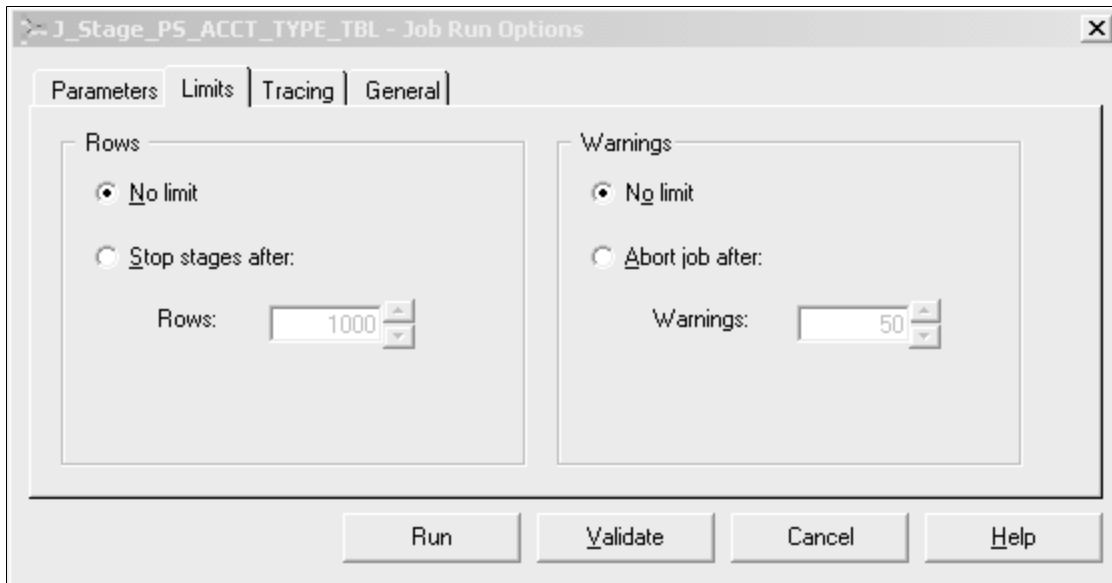
When setting values for environment variables, you can specify either *\$PROJDEF*, *\$ENV*, or *\$UNSET* special values:

- When you use *\$ENV*, DataStage uses the current setting for the environment variable.
- When you use *\$PROJDEF*, the current setting for the environment variable is retrieved and set in the job environment. This allows the environment variable value to be used anywhere in the job. If the value of that environment variable is subsequently changed in DataStage Administrator, the job picks up the new value without the need for recompiling.
- When you use *\$UNSET*, DataStage explicitly unsets the environment variable.

In the Limits tab, you specify any run time limits.

Image: Job Run Options Limits Tab

This example illustrates the Job Run Options Limits Tab.

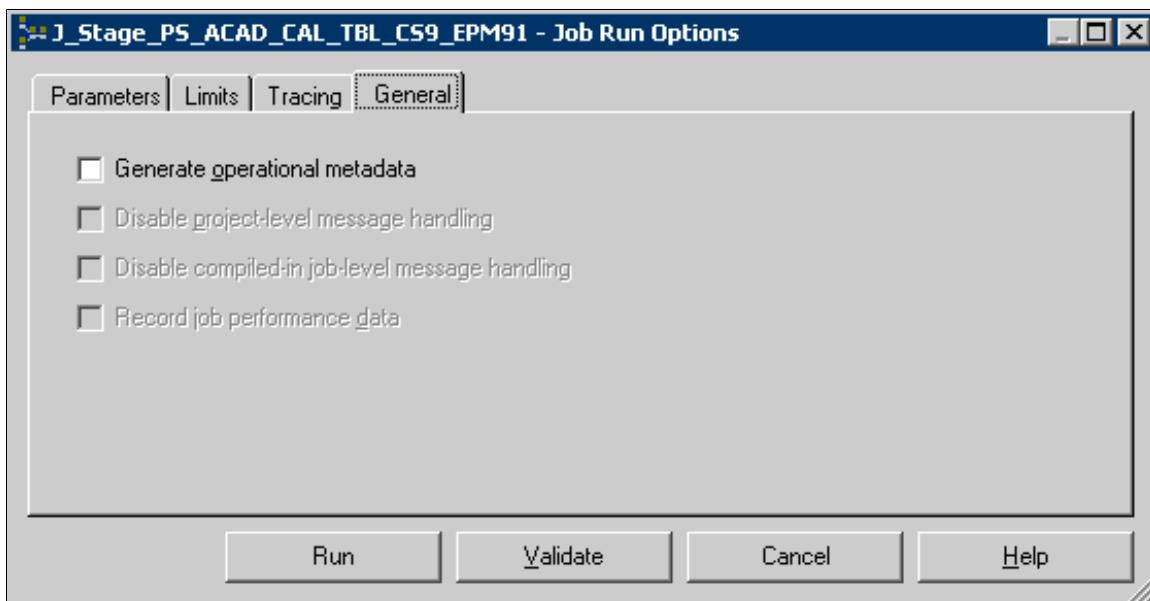


You can specify whether stages in the job should be limited in how many rows they process and whether runtime error warnings should be ignored.

You specify whether the job should generate operational metadata in the General tab.

Image: Job Run Options General Tab

This example illustrates the Job Run Options General Tab.



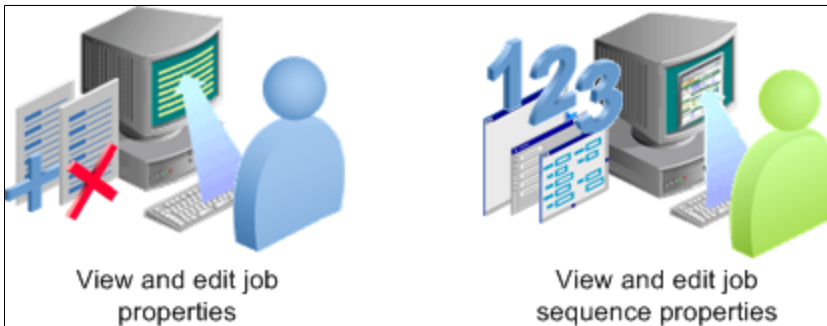
You can also disable any message handlers specified for the job run in the General tab.

Editing Job Properties

The Job Properties window enables you to:

Image: Editing Job Properties

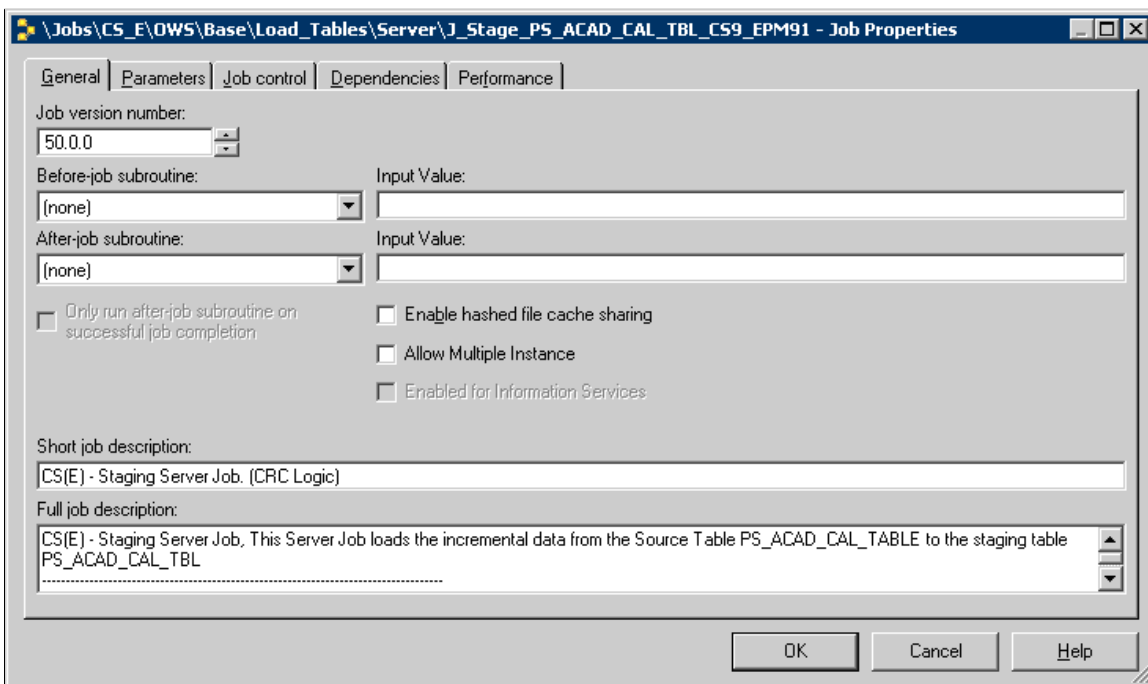
This graphic illustrates that you can view and edit job properties and job sequence properties.



To edit job properties, click the Job Properties button on the DataStage Designer toolbar.

Image: Job Properties Window

This example illustrates the Job Properties Window .



The Job Properties window contains the following tabs:

Job Properties Page	Description
General	Enter name, category description, version number, before and after job subroutines, and their input values.

Job Properties Page	Description
Parameters	Define parameters to represent processing variables. Operators can be prompted for values at run time.
Job Control	Set up a job control routine using BASIC functions to call and run other jobs from the current job. You can also set up job control by using the Sequence Editor in the Designer module.
Dependencies	Enter any dependencies that this job has on functions, routines, or other jobs.
Performance	Displays options for improving performance.

Using Expressions

Expressions define a value that is evaluated at run time.

Simple expressions can contain:

- A string or numeric constant, for example, percent or 42.
- A variable name.
- A built-in or user-defined BASIC function.

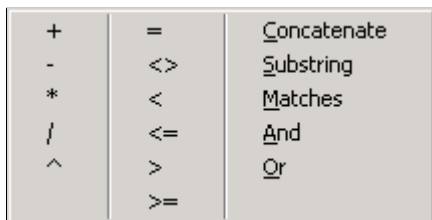
A complex expression can contain a combination of constants, variables, operators, functions, and other expressions.

Accessing Expressions

You can access expressions by double-clicking a Transformer Stage within a job. Next you double-click the Derivation cell for any column in a transformer link and the Expression Editor opens. You can type an expression directly into the editor, or use the menu by clicking the Suggest button on the right side.

Image: Expression Suggestion Menu

This example illustrates the Expression Suggestion Menu.

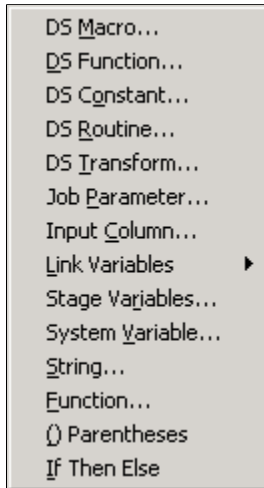


Different menus display depending on whether you right-click an input link, output link, the stage variable table, or links area background. The different menus are also dependent on what type of job you are working on (Server, Parallel, or Mainframe). The output link menu includes operations on

Derivations. The input link menu includes operations on key expressions. The stage variable menu includes operations on stage variables.

Image: Expression Editor Main Menu

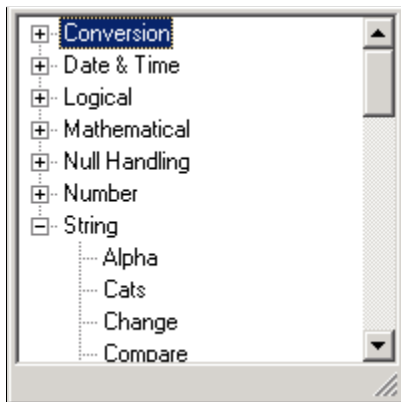
This example illustrates the Expression Editor Main Menu.



To insert a function in your expression, click the Suggest button and select Function. The following choices are displayed:

Image: Suggested functions

This example illustrates the Suggested functions.



Click the plus sign next to the function type to display the functions categorized under them.

Expression Editor Options

This table describes the menu options for the Expression Editor:

<i>Expression Editor Menu Options</i>	<i>Description</i>
DS Macro	Insert a built-in DataStage BASIC Macro.
DS Function	Insert a built-in DataStage BASIC function.

Expression Editor Menu Options	Description
DS Constant	Insert a predefined constant.
DS Routine	Insert a routine from the repository.
DS Transform	Insert a transform from the repository.
Job Parameter	Insert a parameter.
Input Column	Insert an input column name.
Link Variables	Insert a link variable
Stage Variables	Insert a stage variable.
System Variables	Insert a system variable.
String	Insert a string.
Function	Insert a function.
() Parentheses	Insert parentheses.
If Then Else	Insert If Then Else logic into the expression.

Creating Constraints

Constraints, like expressions, enable you to filter or limit data based on criteria that you enter.

You can define a constraint or specify a reject link by both selecting an output link and clicking the *Edit constraints* button on the toolbar, or by double-clicking the output link header Constraint entry.

You can specify a constraint for each output link from a Transformer stage. You can also specify a particular link as a reject link. Reject links output rows that have not been written to any other output links from the Transformer stage.

In the example above, using the expression `InsertFlag="Y"` as a constraint verifies whether the lookup stages return a value. If no values were returned by the lookup stages, the `InsertFlag` field is set to N.

Using Hashed File Stages

Using hashed files improves job performance by enabling validation of incoming data rows without having to query a database each time a row is processed. These are called lookups. The hashed file can also be placed locally, eliminating time that would be spent accessing a remote server.

You can create hashed files to use as lookups in your jobs by running one of the delivered hash file jobs, or you can create a new job that creates a target hashed file. In many of the delivered PeopleSoft sequence jobs, the appropriate hashed file is refreshed as the last step following the load of the data table, which ensures synchronized updates to the data in the hashed file for use in future lookups.

Hashed file stages:

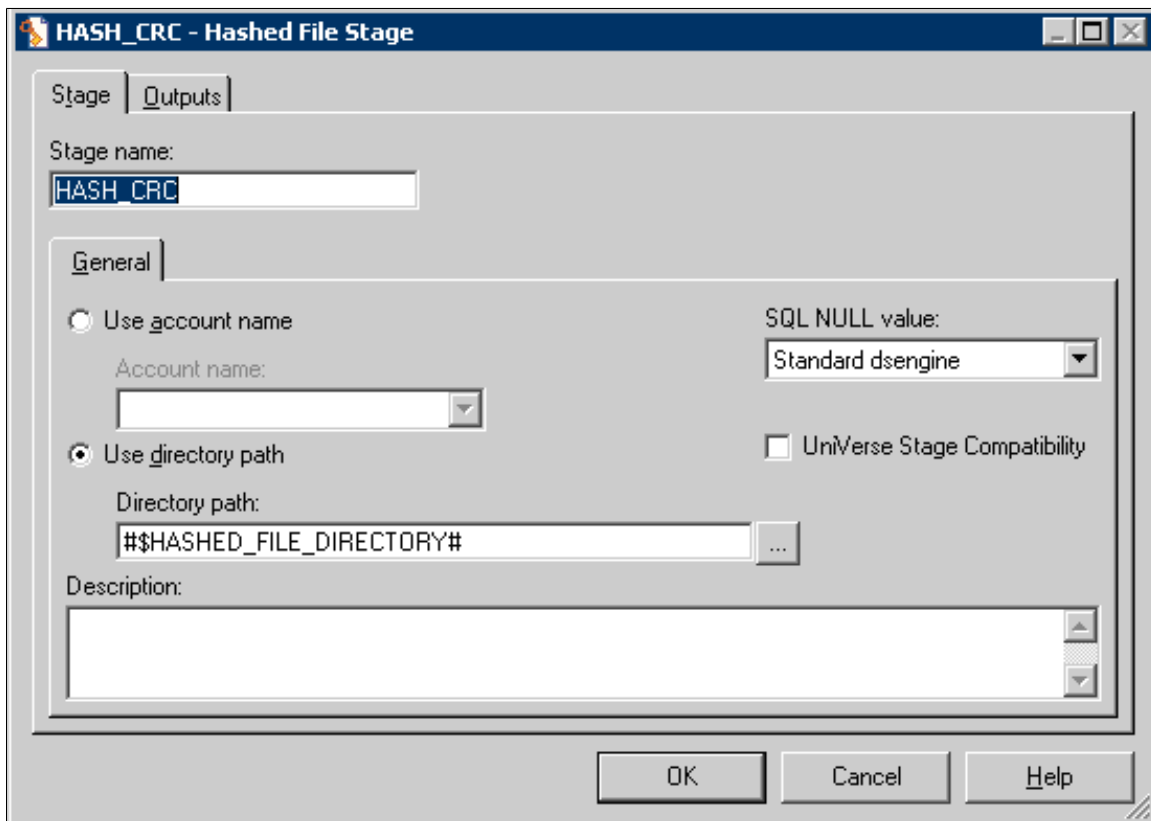
- Represent hashed files, which use a specific algorithm for distributing records in one or more groups, typically to store data extracted from a database.
- Can be used to extract or write data, or to act as an intermediate file in a job.
- Are most commonly used as reference tables or lookups based on key fields.
- Can have any number of inputs or outputs.
- Can be static or dynamic.

Accessing Hashed File Stages

To access a hashed file stage, double-click the hashed file stage in a job.

Image: Hashed File Stage Properties Window

This example illustrates the Hashed File Stage Properties Window.



The Hashed File Stage window contains the following tabs:

Window Element	Usage
Stage tab	Define whether an account name or a directory path accesses the hashed file.

<i>Window Element</i>	<i>Usage</i>
Inputs or Output tab	If the stage has an input, the Input tab is available. If it has an output, the Output tab is available.
Inputs - General tab	Enter the hashed file name. Select whether to create a new file, if none exists. Select whether to clear the file before writing to it.
Inputs - Columns tab	Select which columns of data will be written to the file.

Creating Hashed File Lookups

Lookups are references that enable you to compare each incoming row of data to a list of valid values, and then accept or reject that row based on the validation result.

DataStage job stages can have two types of input links:

- A Stream link represents where the data flow will flow, and is displayed as a solid line.
- A Reference link represents a table lookup, and is displayed as a dotted line.

Stream links, represented by solid lines, can connect either active or passive stages. Reference links, shown as dotted lines, are only used by active stages. Their purpose is to provide information that may affect how the data is changed, but they do not supply the actual data to be changed.

Typically, hashed files are used as lookups because they are much quicker to access than querying a database. Hashed files used as lookups usually contain only one or two key columns against which incoming data can be validated.

Before you can create the lookup, you must first create a hashed file containing the values to be used as a reference. To add a lookup stage to a job, you select the hashed file stage from the File palette, enter the directory path and file name of the hashed file, and link the hashed file stage to a transformer stage.

You use a DRS stage as a lookup when your lookup requires that use of relational operators, such as \geq and \leq .

Using Job Sequencers

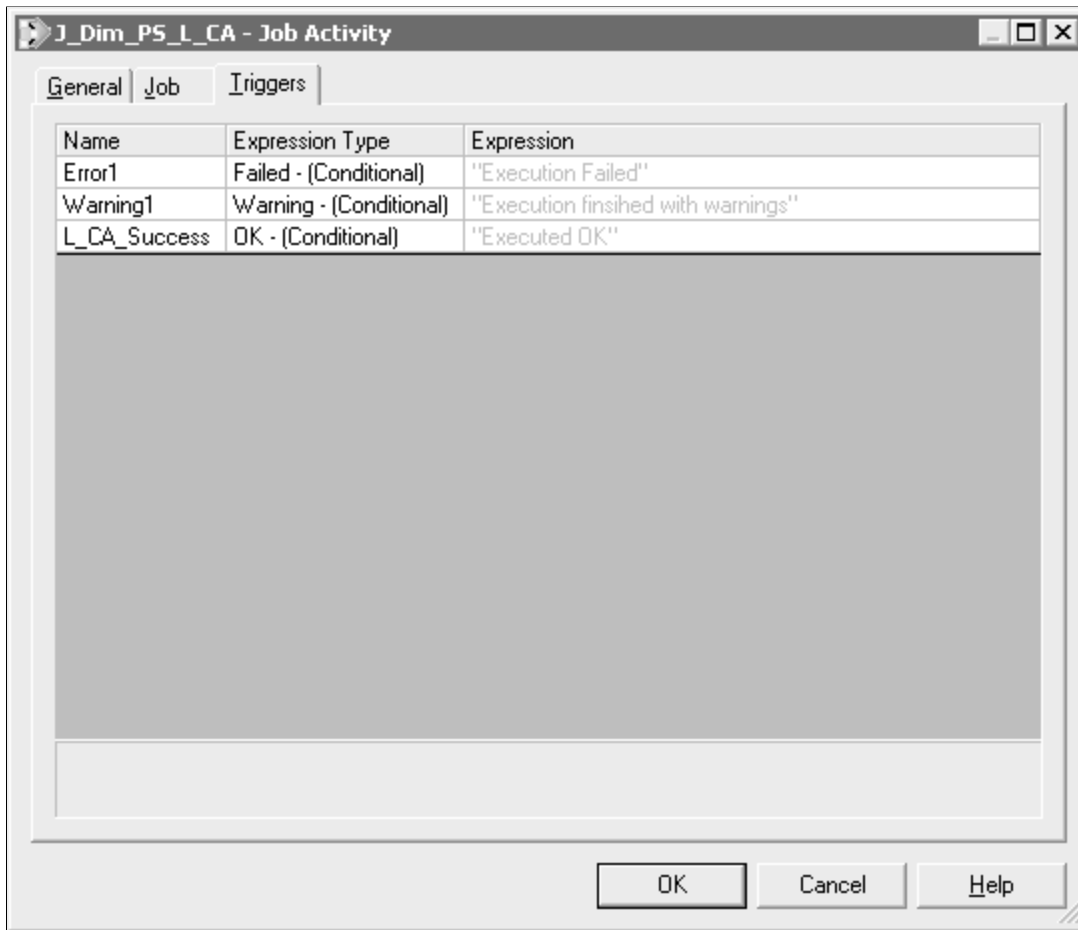
Job Sequencers enable you to:

- Set up a sequence of server jobs to run.
- Specify control information such as different course of action depending on load type or other criteria.

Designing job sequencers is similar to designing server jobs. You create the job sequence in DataStage Designer, add activities from the palette, and then join or sequence activities together using links. You control the flow of the activity sequence using triggers.

Image: Job Sequencer Trigger Window

This example illustrates the Job Sequencer Trigger Window.



Once you have defined a job sequence, you can schedule and run the job from DataStage Director.

Note: Job sequencers can also be run using DataStage Director.

Using DataStage BASIC

If you need to load data from a non-PeopleSoft source, you usually do not have to perform any programming tasks: you can use the delivered transforms and routines, using the delivered jobs and sequences as templates. For more complex jobs, you can use DataStage BASIC to:

- Define custom routines.
- Define custom transforms.
- Define derivation, expressions, and constraints in the transformer stage.

- Define before-job or after-job subroutines.
- Define job control routines.

PeopleSoft provides ETL jobs for loading data from PeopleSoft applications into EPM. Some of the PeopleSoft jobs use custom routines using DataStage BASIC.

Using DataStage Director

Using DataStage Director

These topics provide an overview of DataStage Director and discuss how to:

- View Job Status
- Schedule Jobs
- Run Jobs
- Monitor Jobs
- Review Job Events

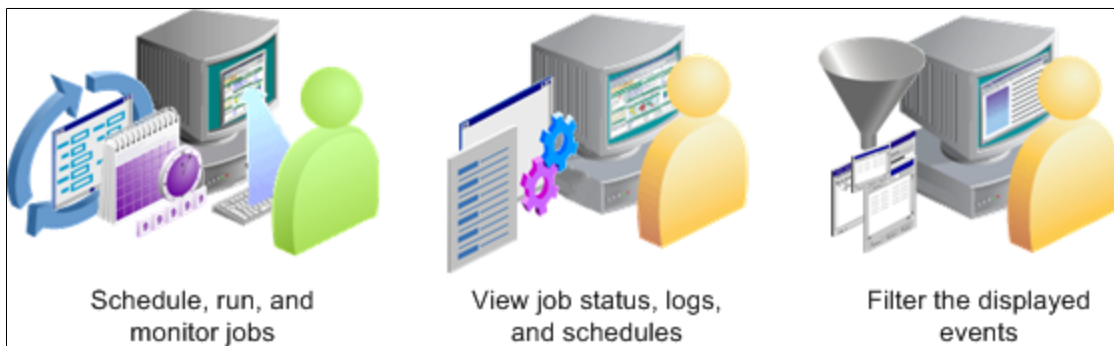
Note: This topic does not discuss all the features available for DataStage Director. For a complete view of DataStage Designer functionality, please see the delivered IBM WebSphere documentation.

DataStage Director Overview

DataStage Director enables you to:

Image: DataStage Director Properties

This graphic illustrates the tasks that can be performed by the DataStage Director.



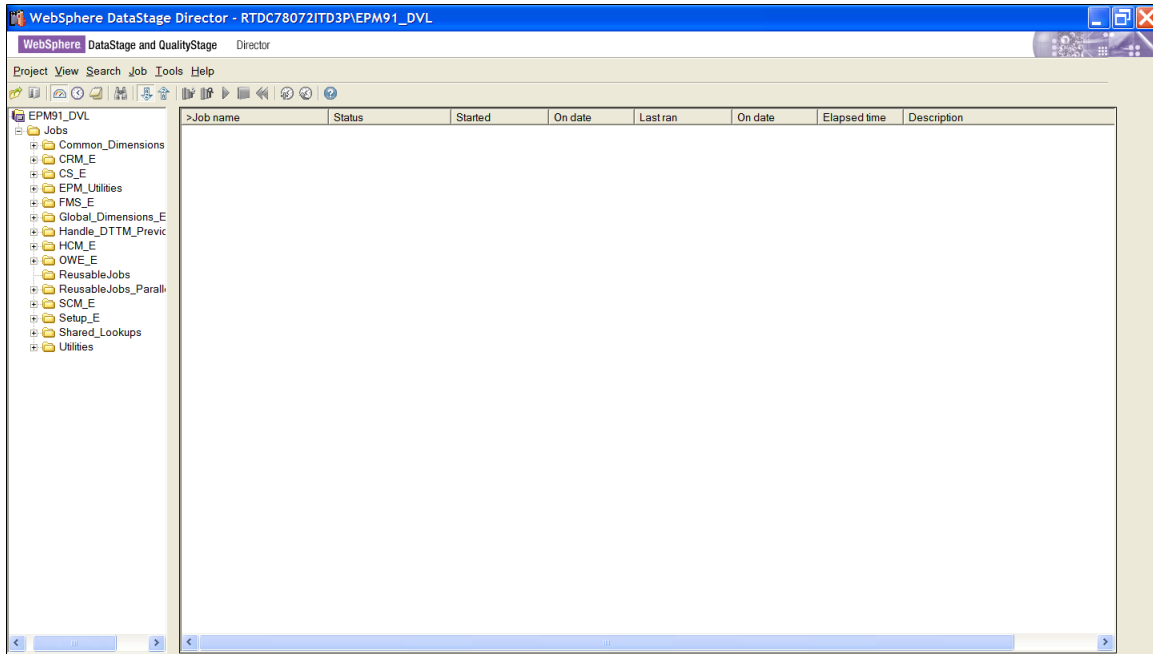
The DataStage Director window is divided into two panes:

- The Job Category pane lists all of the jobs in the repository.

- The right pane shows one of three views: Status view, Schedule view, or Log view.

Image: DataStage Director Window

This example illustrates the DataStage Director Window.



DataStage Director Menu Options

This table describes DataStage Director menu options:

Menu Option	Description
Project	Open another project, print, or exit.
View	Display or hide the toolbar, status bar, buttons, or job category pane, specify sorting order, change views, filter entries, show more details, or refresh the screen.
Search	Start a text search dialog box.
Job	Validate, run, schedule, stop, or reset a job, purge old entries from the job log file, delete unwanted jobs, clean up job resources (if this is enabled), set default job parameter values.
Tools	Monitor running jobs, manage job batches, start the DataStage Designer.
Help	Displays online help.

DataStage Director Toolbar Options

The following buttons appear on the DataStage Director toolbar:

Image: DataStage Director Toolbar

This example illustrates the DataStage Director Toolbar.



This table describes the buttons provided with the DataStage Director toolbar

Toolbar Button	Description
Open	Open a project.
Print	Print the current list or log.
Status	Select the Job Status view.
Schedule	Select the Schedule view.
Log	Select the Job Log view.
Find	Search for a job.
Ascending	Sort the list in ascending order.
Descending	Sort the list in descending order.
Run Now	Run the currently selected job.
Stop	Stop the job run.
Reset	Reset the job and cancel any changes to the target.
Add to Schedule	Add a job to the schedule.
Reschedule	Change the job schedule.
Help	View online help.

DataStage Director View Options

DataStage Director has three view options:

- The Status view displays the status, date and time started, elapsed time, and other run information about each job in the selected repository category.
- The Schedule view displays job scheduling details.
- The Log view displays all of the events for a particular run of a job.

Viewing Job Status

The Job Status view in DataStage Director can display these possible states:

- *Compiled*: The job has been compiled, but not run since compilation.
- *Not compiled*: The job has not compiled successfully.
- *Running*: The job is currently running.
- *Finished*: The job has finished running.
- *Finished (see log)*: The job has finished, but there are warning messages or rows were rejected.
- *Stopped*: The job was stopped by the operator.
- *Aborted*: The job finished prematurely.
- *Validated OK*: The job has been validated with no errors.

Scheduling Jobs

You can schedule a job to run in a number of ways:

- Once today at a specified time.
- Once tomorrow at a specified time.
- On a specific day and at a particular time.
- On the next occurrence of a particular date and time.
- Daily at a particular time.

To schedule a job In DataStage Director, select the Schedule button from the toolbar.

You can schedule each job to run on any number of occasions using different job parameters, if necessary.

Note: Microsoft Windows restricts job scheduling to administrators. You need to be logged in as an administrator to use the DataStage scheduling features.

Running Jobs

Each time that you run or schedule a job, you can:

- Change the job parameters and parameter values that are associated with the job.
- Override default limits for row processing and warning messages that are set for the job run.
- Set tracing options.

You set job options in the Job Run Options dialog box. They appear automatically when you start to run or schedule a job.

Setting Parameters

You can use the default parameter values, or enter another value. To reinstate the default values, click the Set to Default or All to Default button. Some job parameters, like dates, may be variables that you must enter for each job run.

If no parameters are set for a job, the system does not display the Parameters tab.

Setting Limits

Select the Limits tab to override any default limits for row processing and warning messages that are set for the job run. Click the Validate button to test the new settings or click the Run button to run the job.

Setting Tracing

Tracing helps analysts troubleshoot jobs. You can generate tracing information and performance statistics for server jobs.

The options on this page determine the amount of diagnostic information that is generated the next time a job is run. Diagnostic information is generated only for the active stages in a chosen job.

When the job runs, a file is created for each active stage in the job. The files are named using the format *jobname.stagename.trace*, and are stored in the &PH& subdirectory of your DataStage server installation directory.

Running, Stopping, Resetting, and Deleting Jobs

To run a job, select Job, Run Now, or click the Run button on the toolbar.

You can stop or reset a job. If a job is stopped or aborted, it might be difficult to trace where it ended. By resetting a job, you set it back to a state that can be run and, optionally, return your target files to their original state. If a job has aborted, it must be reset before you can run it again.

You can remove old or unwanted jobs from your project from either the Director or Designer.

Monitoring Jobs

Job monitoring enables you to review job progress through the links and to see the number of rows that are being processed.

To monitor a job In DataStage Director, select a job and then select Tools, New Monitor:

The Monitor window displays summary information about relevant stages in a job. It contains a tree structure that displays stages in a job and their associated links. For server jobs, it shows active stages. Active stages perform processing rather than reading or writing to a data source.

The Link type column displays up to four types of links:

Link Type	Description
<<Pri	Primary link
<<Ref	Reference link
<<Out	Output link
<<Rej	Reject link

To see detailed information, double-click a link.

Setting the Refresh Interval

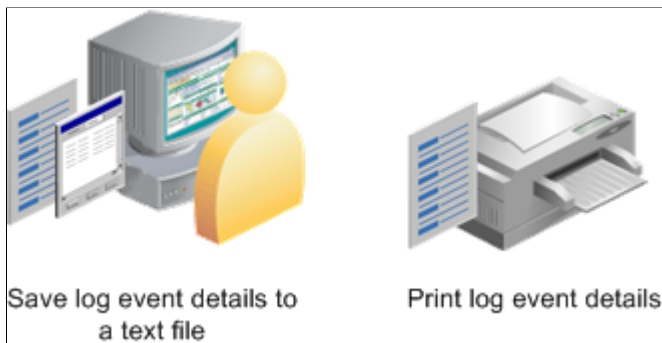
The Monitor window display is updated with new information from the server at regular intervals. You can set how often the updates occur by specifying a time, in seconds, in the Interval field. Click the arrow buttons to increase or decrease the value, or enter the value directly. The default setting is 10. The minimum value is 5. The maximum value is 65.

Reviewing Job Events

DataStage Director enables you to:

Image: DataStage Director - Reviewing Job Events

This graphic illustrates tasks that can be performed by the DataStage Director.



You can filter:

- Jobs that appear in the Job Status view.
- Events that appear in the Job Log view.

Filtering Jobs

To filter jobs In DataStage Director, select View, Status, and then View Filter Entries.

This table describes job filtering options:

Window Element	Description
Include	Specify which jobs to include. Your choices are: <ul style="list-style-type: none"> • All jobs • Jobs matching <i><a string criteria that you enter></i> (case-sensitive)
Exclude	Specify which jobs to exclude. Your choices are: <ul style="list-style-type: none"> • No jobs • Jobs matching <i><a string criteria that you enter></i> (case sensitive)
Job Status	Your choices are: <ul style="list-style-type: none"> • All • All, except "Not compiled" • Terminated normally • Terminated abnormally
Released jobs	Include only released jobs.

Filtering Events

To organize log event details, you can filter events to select which log events are displayed, based on age, date/time, or event type. You can also specify the maximum number of entries that will appear.

To filter events In DataStage Director, select View, Log, then View, Filter Entries.

This table describes the events filtering options:

Window Element	Description
From:	Defines the earliest event to include: <ul style="list-style-type: none"> • Oldest. • Start of last run. • A specific date and time.
Through	Defines the most recent event to include: <ul style="list-style-type: none"> • Newest. • A specific date and time.

Window Element	Description
Limit	Limit the number of events that are displayed: <ul style="list-style-type: none">• Select all entries.• Last <i>N</i> entries.
Type	Select one or more types of events that the system will display: <ul style="list-style-type: none">• Information• Warning• Fatal• Reject• Other

Printing Jobs

To print jobs in DataStage Director, select Project, Print.

You can send the event log details either to your printer or to a text file.

Managing Aborted and Failed Jobs

This topic provides information on how you can manage aborted and failed jobs and discusses how to:

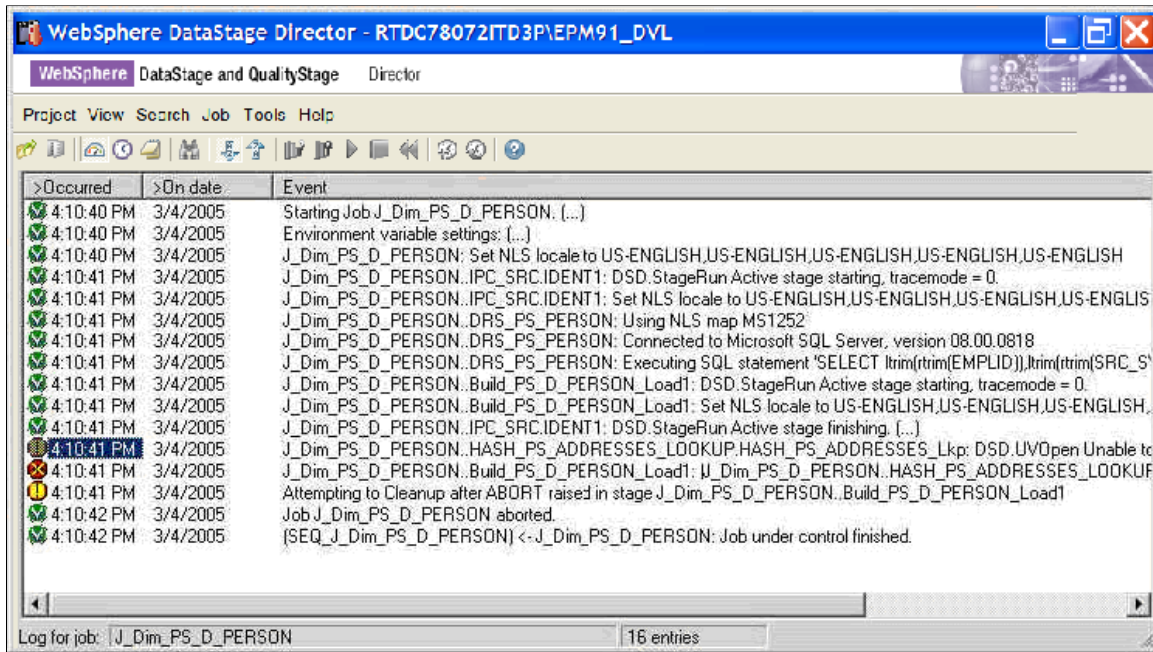
- Review the job log to determine job errors.
- Debug aborted and failed jobs.

Reviewing the Job Log to Determine Job Errors

The first step in managing aborted or failed jobs is to use DataStage Director to review the job log, which provides job run information.

Image: Detailed Job Log View of Aborted and Failed Jobs

This example illustrates the Detailed Job Log View of Aborted and Failed Jobs.



Using the job log you can first determine which jobs require your attention. Note that the job log displays which jobs aborted or failed.

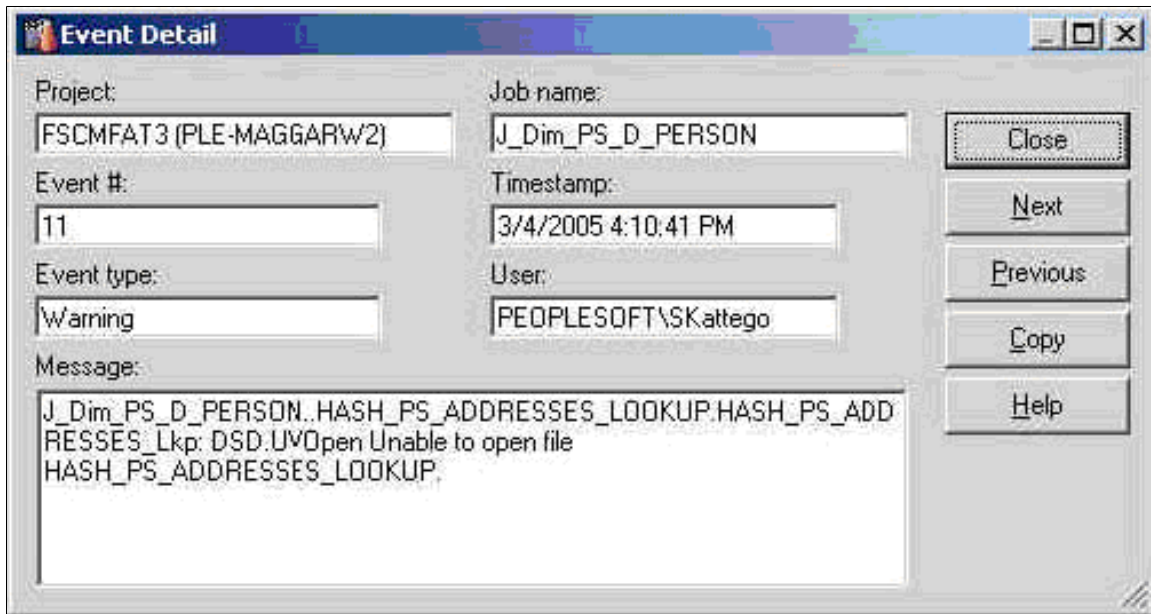
Jobs status are color-coded as follows:

- Green (V): Informational. Success condition.
- Yellow (I): Failed with warnings.
- Red (X): Error messages.

You can double-click an aborted or failed job to view details about the job.

Image: Event Detail for Failed Job

This example illustrates the Event Detail for Failed Job.



A common cause for jobs aborting is that dependent hash files do not exist. This happens when a hash file that a job performs a lookup on has not been pre-created. The hash file load jobs have to be run. As you can see in the screen above, event details suggest that the job is missing the hash file HASH_PS_ADDRESSES_LOOKUP, which is required.

Debugging Aborted and Failed Jobs

Once you know which job has an issue, you can access the job in DataStage Designer and view the job with *performance statistics* on. This shows successful links in green and failed links in red, and helps target the specific part of the job design that failed. The performance statistics also gives the number of rows that have been transmitted through each link, again which information can be useful for debugging a job. Datastage Designer also provides advanced debugging features that can help developers set break points and watch variable values.

Jobs that run with more than on array size or transaction size usually result in a warning message; the job log displays a warning message relevant to each row of data in some instances. For example, if a job has a right string truncated error when inserting into the target database, the log specifically provides the row data that failed.

To address this type of issue, configure the job to limit the rows to process so there are less rows processed during job execution. This restricts the job run time and the log will also be smaller and more manageable.

Setting Up DataStage for EPM

Understanding IBM WebSphere DataStage Setup and Configuration

This topic provides an overview of DataStage setup and configuration.

Setting Up DataStage for EPM

In order for IBM WebSphere DataStage to work properly with EPM, you must perform the following setup and configuration steps:

- Create projects: Projects are used to import your *.dsx file metadata, which include ETL jobs.
- Setup and configure environmental parameters: Every ETL job uses environmental parameters and you must configure the default parameters.
- Import *.dsx files into projects: You must specify a project to hold your DataStage metadata, including ETL jobs.

Setting Up DataStage Projects

This topic discusses considerations regarding the design of your DataStage projects and how to create a project.

Project Structure Considerations

Before you create your DataStage projects, you should consider how many projects you need to create. The number of projects you need depend on the following factors:

- The PeopleSoft source transaction system that you are using.
- The license codes for the Data Marts and/or EPM Applications you have purchased, or plan to implement.
- Disk space management based on storing hash files, log files, and *.dsx files.
- Common jobs necessary to all products (you may wish to create a separate project for the common jobs if you are implementing more than one data mart).

Based on the preceding information, you have the following options:

- Create one project per PeopleSoft source.

For example, you can create a project called HCM_EPM91 for bringing data from a PeopleSoft Human Capital Management (HCM) source database.

- Create one project per EPM Warehouse (such as HCM or SCM warehouses).

For example, you can create a project called CRM_EPM90 to handle all CRM-related Data Marts or EPM applications.

- Create one project for all the EPM Warehouses.

For example, you can create a project called EPM90_ALL which will manage ETL jobs relating to all of the EPM Warehouses.

Because you need to create projects based on the preceding requirements, ensure that you have understood your long-term needs and requirements before creating suitable projects for importing the jobs.

Note: Certain operating systems, such as Unix, have a limit to the number of objects that can be created under one project or directory. Please check your operating system specifications before proceeding with project creation.

Note: There is no naming convention for projects.

Creating Projects

Use the following steps to create DataStage projects:

1. In the DataStage Administrator, enter the following information in the Attach to DataStage box:

- *Domain:* Enter the name of the domain server machine.

You can enter *localhost* if the client and server are installed on the same machine.

- *User name:* Enter the user name that is required to log onto the machine on which the domain server is installed.

The user name is the Windows login (if the server is running on a Windows platform) or the Administrator user that was used during the install (if the server is running on a Unix platform).

- *Password:* Enter the password that is required to log onto the machine on which the domain server is installed.
- *DataStage server:* Select the name of the server machine on which the DataStage engine is installed.

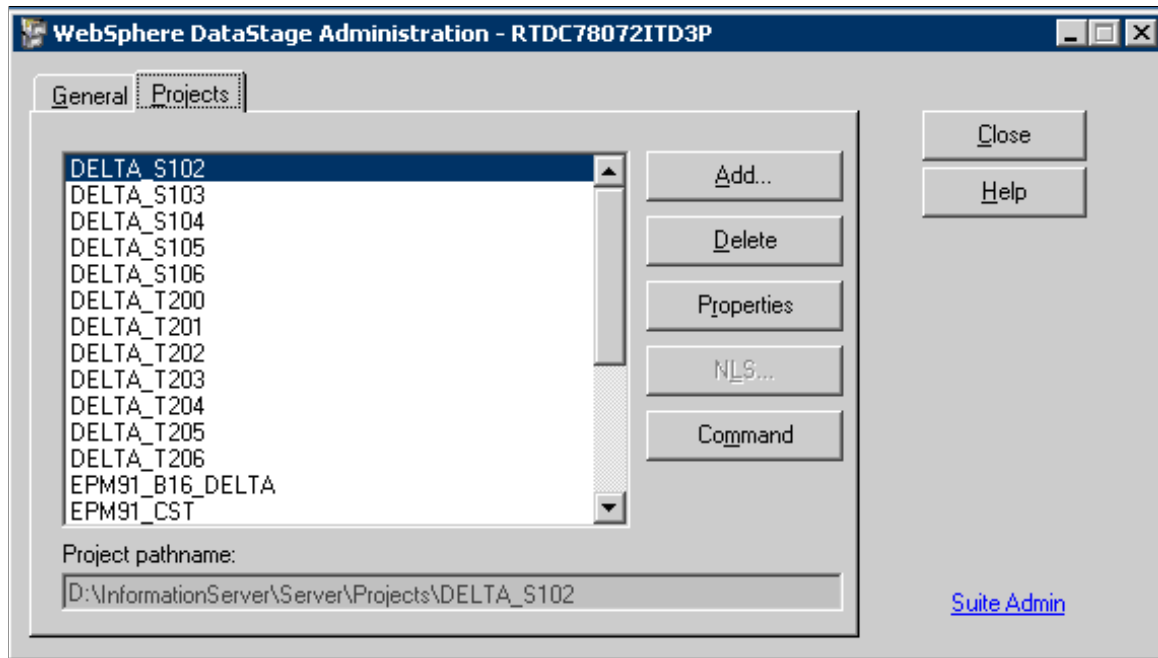
2. Click OK.

3. Click the Projects tab to view a list of available projects on the server.

- Click the Add... button.

Image: DataStage Administrator - Projects tab

This example illustrates the DataStage Administrator - Projects tab.



The Add Project box displays.

- Enter the project name in the *Name* field . You can specify your own naming convention in this field.

Remember the project name is case-sensitive and you cannot change the project name after you have imported the dsx files. If you want to update or change the project, you only have the option to create a new project and import the dsx files again. DataStage Administrator automatically appends the project to the default location for a new project.

Click the Browse button to select another location if the default location is not desirable (due to disk space constraints). Do not select the Create protected project check box as this would cause the project to be read-only.

- Click OK to create the project on the server.

This creates an empty project on the DataStage server. The updated list of projects displays after the copy is finished.

- Repeat steps five through seven to create any additional projects.
- When you are finished, click Close.

See [Setting Project Properties](#).

Related Links

WebSphere DataStage Administration: Administrator Client Guide

Configuring Environmental Parameters

This topic discusses how to copy the DSPARAMS file and add values for environmental variables. You must configure default environmental parameters for each project you are using, as every job uses a subset of the environmental variables.

Copying the DSPARAMS File

Copying the DSPARAMS file enables you to bypass entering the parameters manually. Use the following steps to copy the DSPARAMS file:

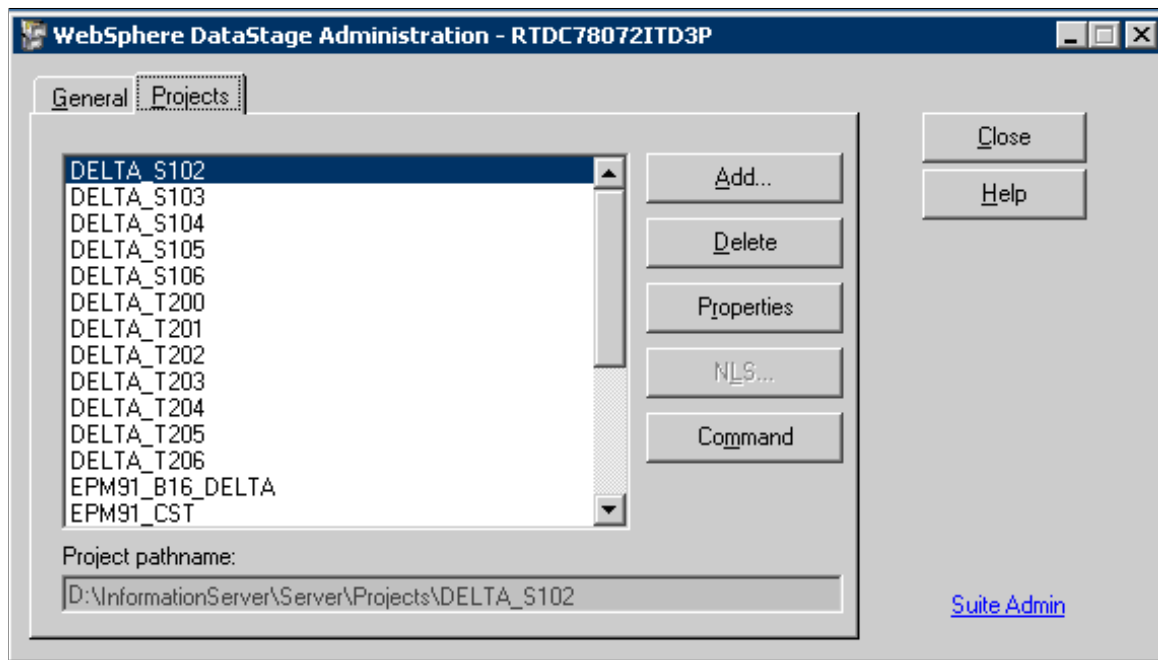
1. Locate the DSPARAMS file using the following path on the installation CD:

<PSHOME>\SRC\ETL.

2. In DataStage Administrator, navigate to the projects tab to determine the project home directory.

Image: DataStage Administrator - Projects tab

This example illustrates the DataStage Administrator - Projects tab.



The project home directory path can be found under Project Pathname field.

3. Select your project name.
4. Use the project home directory path to navigate to your DSPARAMS file.
5. Rename the DSPARAMS file located in the project home directory folder to *DSPARAMS_old* and paste the copied file.

Now, you can see a DSPARAMS file under this path.

6. Repeat steps one through five for each project you have created.

Adding Values for Environmental Variables

You must add suitable values for each environmental parameter you plan to use. For example, if you are only implementing an EPM Warehouse, you need not update OWE related parameters.

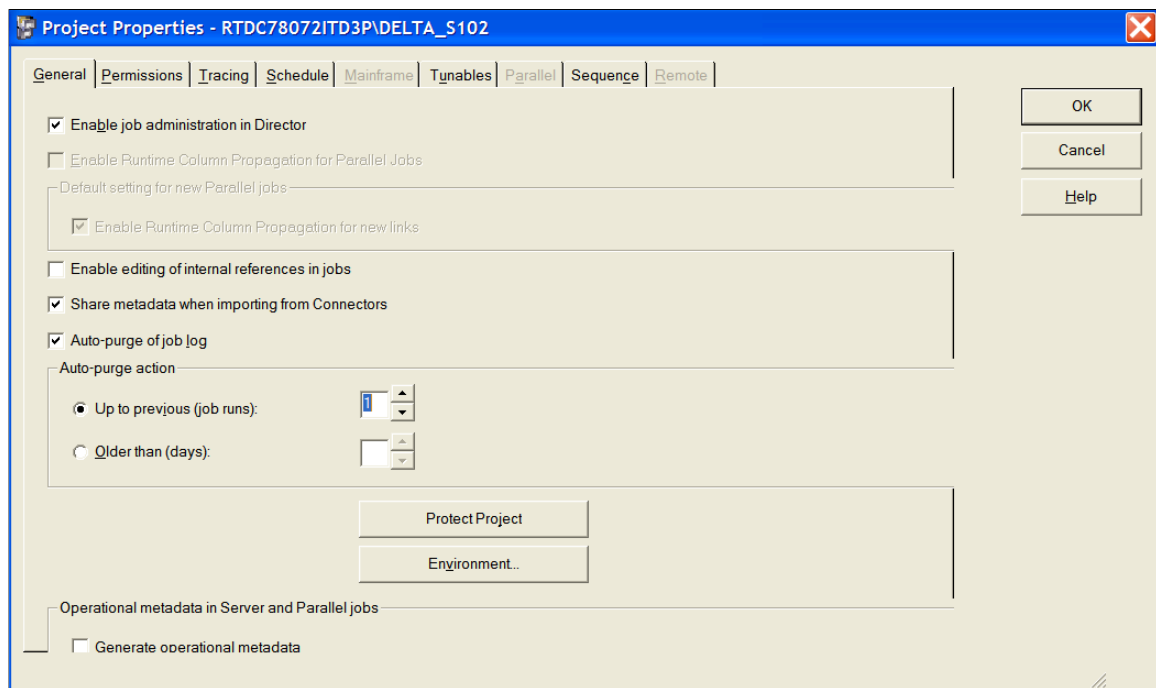
Use the following steps to add values for environmental variables:

1. In DataStage Administrator, select the Projects tab of the DataStage Administrator box to view a list of available projects on the server.
2. Select the project you would like to configure.
3. Click the Properties button.

The Project Properties window displays.

Image: Project Properties - General tab

This example illustrates the Project Properties - General tab.

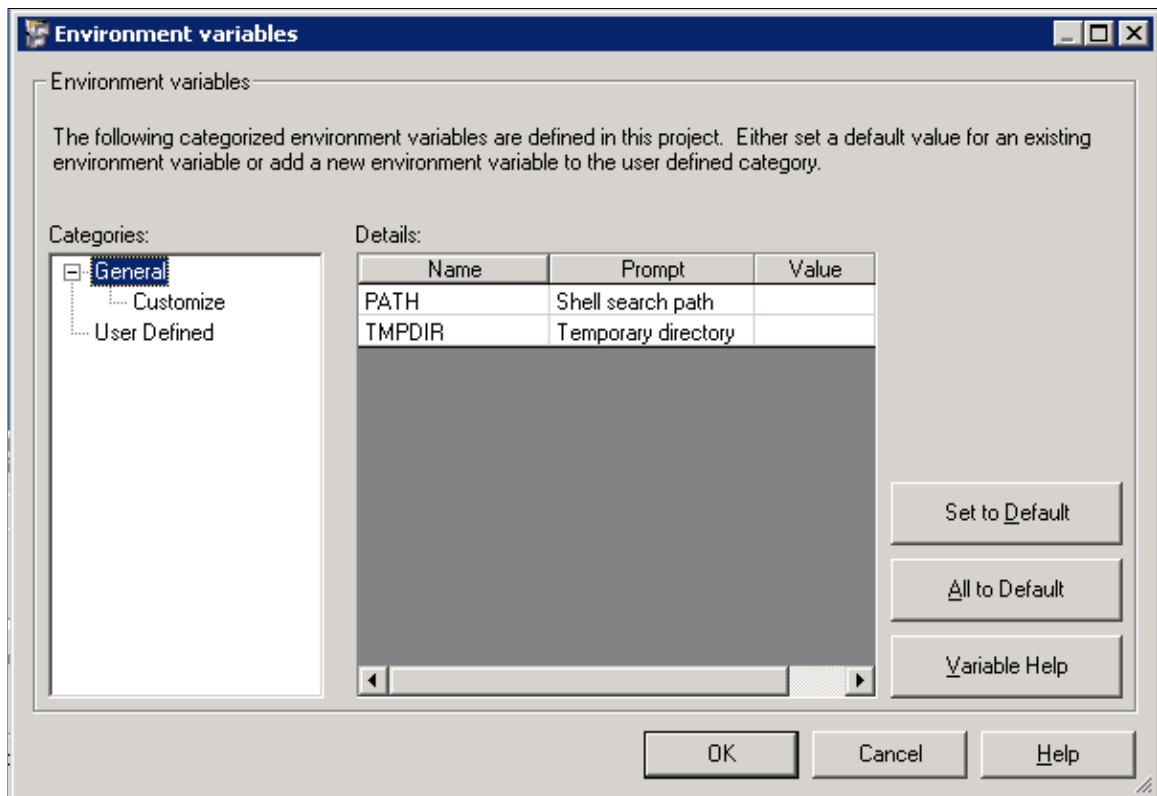


4. Click the Environment... button.

The Environment Variables box displays.

Image: Environment Variables window

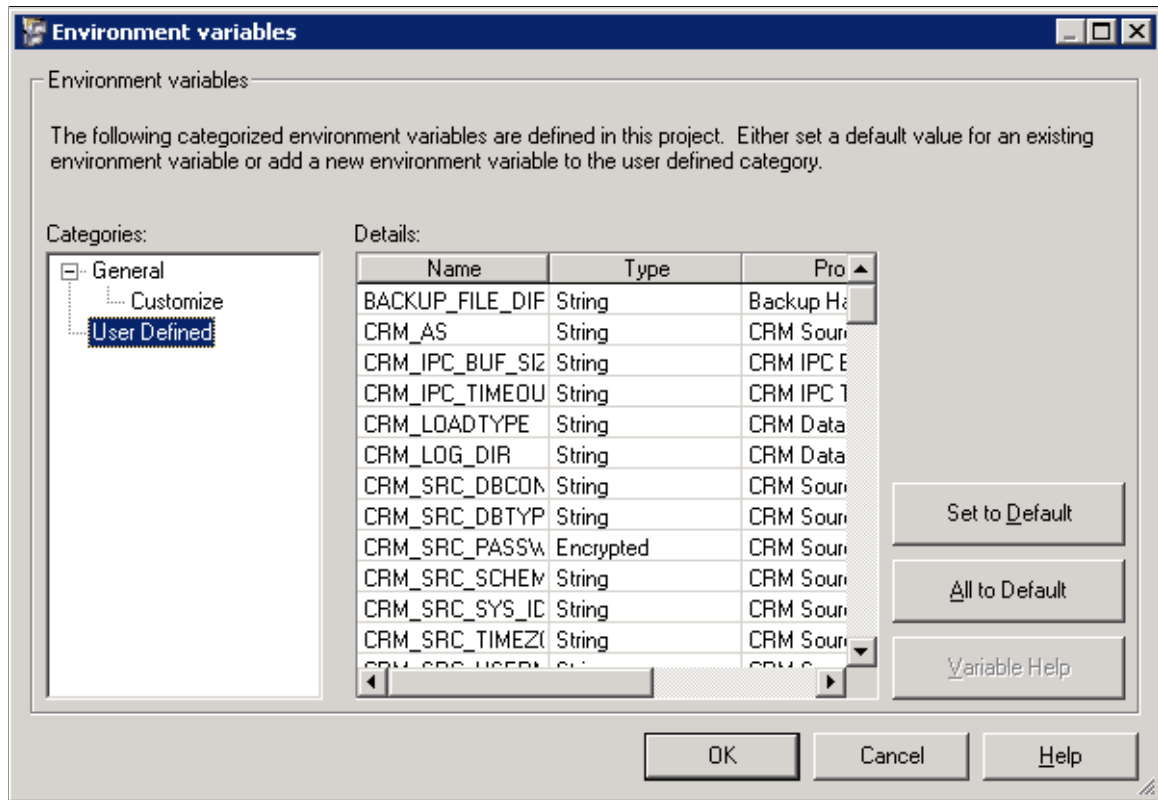
This example illustrates the Environment Variables window.



- In the Categories navigation frame, select *User Defined*.

Image: Environment Variables window, User Defined variables selected

This example illustrates the Environment Variables window, User Defined variables selected.



- Scroll to the bottom of the list and enter the default values based on the warehouse setup.

Ensure you do not include any trailing spaces in the values.

Note: Ensure that you perform a detailed analysis of the various environmental variables that impact your requirements with respect to the project created and the corresponding values. Use defaults wherever required. If you have not provided the correct values for the chosen environmental variables, the related jobs may abort or not work as required.

See [Environmental Parameters Information](#).

Importing Delivered .DSX Files

After you have created your projects and configured the project-level environmental parameters, you can import the .dsx files into the projects. PeopleSoft prepackaged .dsx files contain server jobs, job sequencers, and other metadata. The prepackaged jobs may belong to any of the following categories:

- Source to OWS
- Source to MDW (for CRM online marketing applications only)

- OWS to OWE
- OWS to MDW
- OWE to MDW

This topic discusses preparations for importing .dsx files and how to import each type of .dsx file.

See [Importing and Exporting Repository Components](#).

Preparing to Import *.DSX Files

Before you begin importing your *.dsx files, you should be thoroughly familiar with the [DSX File Import Descriptions](#) documentation, which contains the following information:

- The first column provides information on the order in which .dsx files should be imported.
- The second column indicates whether the .dsx file corresponds to a specific warehouse/SKU or generic file.
- The third column describes the type of jobs the .dsx file contains, such as OWS, OWE, and setup.
- The fourth column describes the source system associated with the .dsx file.
- The fifth column provides the name of the .dsx file.
- The sixth column provides the location of the .dsx file on the installation CD.
- The seventh column provides a detailed description of the .dsx file.

Recall that *.dsx files use the following naming convention:

- *<file_name>.dsx*, for DataStage 8.1 customers.
For example, Common_Uilities.dsx.
- *<file_name>_85.dsx*, for DataStage 8.5 customers.
For example, Common_Uilities_85.dsx.

In order to import the .dsx files, you must be able to link to the Windows File Server from the IBM WebSphere DataStage Client.

Note: The .dsx files are delivered as zip files and must be unzipped before importing.

Note: The .dsx files can only be installed on a Microsoft Windows client operating system. If you have installed EPM on UNIX, you must access the files from the Windows file server.

Steps to Import a .DSX File

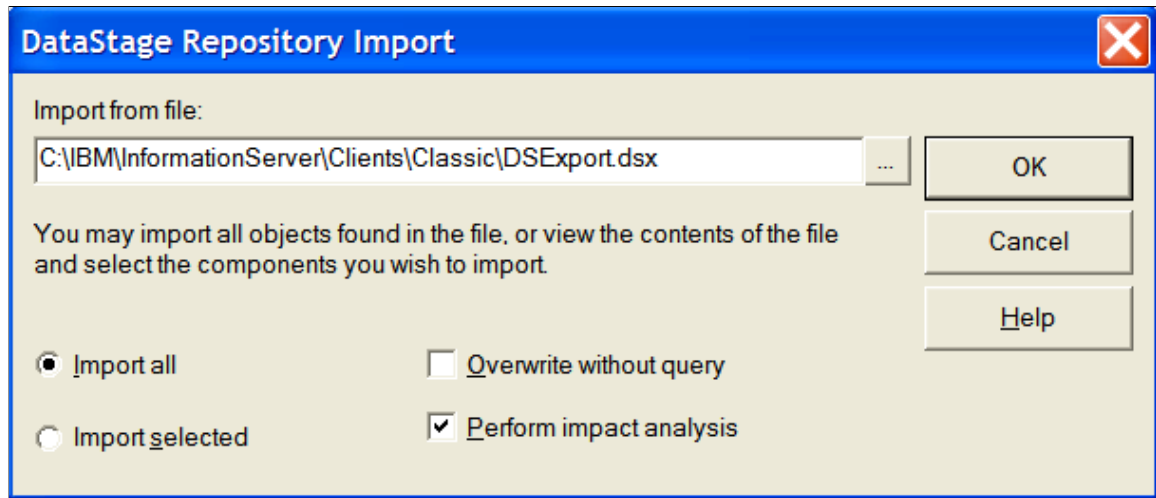
Use the following steps to import a .dsx file:

1. In DataStage Designer, select Import, DataStage Components from the menu.

The DataStage Repository Import window displays.

Image: DataStage Repository Import Window

This example illustrates the DataStage Repository Import Window.



2. Click the Import from file browse button to locate the .dsx file you want to import.

Do not select the Perform impact analysis check box unless you want to perform usage analysis. Selecting this check box increases the time it takes to import the .dsx files into a project.

3. Select Import All and click OK to import the file.

Once the import process is complete, the Import Progress window closes.

Importing .DSX Files for EPM Analytical Applications Only

If you are planning to implement only the EPM Analytical Applications, the following topics provide a guide to .dsx files you need.

Import the following common .dsx files:

- COMMON_UTILITIES.dsx
- COMMON.dsx
- OWE.dsx

Import any or all of the following .dsx files, depending upon the number of PeopleSoft source transaction systems that you may have:

- WCS_OWS.dsx (For a PeopleSoft Campus Solutions database)
- WCR_OWS.dsx (For a PeopleSoft CRM source database)
- WFN_OWS.dsx (For a PeopleSoft FMS source database)
- WHR_OWS.dsx (For PeopleSoft HRMS and Learning Management source database)

- WSC_OWS.dsx (For a PeopleSoft SCM source database)

See [DSX File Import Descriptions](#).

Importing .DSX Files for EPM Warehouses Only

If you are planning to implement only EPM Warehouses, import the following common .dsx files:

- COMMON_UTILITIES.dsx
- COMMON.dsx

Each EPM Warehouse (for example, Campus Solutions Warehouse or HCM Warehouse) requires a unique list of *.dsx files to import.

The *ETL Reference Documents* topic helps you determine which .dsx files you need to import for your warehouse. See [DSX File Import Descriptions](#).

Chapter 12

Defining ETL Parameters

Defining ETL Parameters

Certain EPM products require that you specify input parameters for ETL jobs prior to running the jobs. For example, some parameters specify row-selection criteria and load methodologies.

These topics discuss how to define ETL parameters for:

- Planning and Budgeting analytical application
- EPM Warehouses (FMS Warehouse and CRM Warehouse)

Defining ETL Parameters for the Planning and Budgeting Analytical Application

This topic discusses how to define financial asset item parameters for the Planning and Budgeting analytical application.

Page Used to Define Financial Asset Item Parameters

<i>Page Name</i>	<i>Definition Name</i>	<i>Navigation</i>	<i>Usage</i>
Financial Asset Item Setup	PF_FIN_AST_SETUP	EPM Foundation, EPM Setup, ETL Map Parameters, Financial Asset Item Setup, Financial Asset Item Setup	Enter the SetID for the rows to be deleted in the OWE table PS_BP_ASSET_ITEMS and reloaded from BD_ASSET_ITEMS.

Financial Asset Item Setup Page

Use the Financial Asset Item Setup page (PF_FIN_AST_SETUP) to enter the SetID for the rows to be deleted in the OWE table PS_BP_ASSET_ITEMS and reloaded from BD_ASSET_ITEMS.

Navigation

EPM Foundation, EPM Setup, ETL Map Parameters, Financial Asset Item Setup, Financial Asset Item Setup

Image: Financial Asset Item Setup page

This example illustrates the fields and controls on the Financial Asset Item Setup page. You can find definitions for the fields and controls later on this page.



SetID Enter the SetID associated with the asset data.

Del. Flag (Y/N) [Delete Flag (Y/N)] Check to indicate that the rows in the target table PS_BP_ASSET_ITEMS that match the SetID on this page are to be deleted.

Defining ETL Parameters for the EPM Warehouses

This topic describes how to set ETL parameters for data marts within the FMS and CRM warehouses.

Pages Used to Define ETL Parameters

<i>Page Name</i>	<i>Definition Name</i>	<i>Navigation</i>	<i>Usage</i>
Financial Mart Account Setup	PF_FIN_ACCT_SETUP	EPM Foundation, EPM Setup, ETL Map Parameters, Financial Mart Account Setup, Financial Mart Account Setup	Enter tree and node details that represent accounts on which Profitability analysis is performed.
Financial Mart GC Cled Setup	PF_FIN_CLED_SETUP	EPM Foundation, EPM Setup, ETL Map Parameters, Financial Mart GC Cled Setup, Financial Mart GC Cled Setup	Enter values such as Business Unit, Scenario, and Period that determine the rows to be deleted from Multidimensional Warehouse table PS_F_CLEDGER and reloaded from PS_GC_CLED_MGT_F00. in the MDW table F_CLEDGER.

Page Name	Definition Name	Navigation	Usage
Financial Mart GC Flow Setup	PF_FIN_FLOW_SETUP	EPM Foundation, EPM Setup, ETL Map Parameters, Financial Mart GC Flow Setup, Financial Mart GC Flow Setup	Enter values such as Business Unit, Scenario, and Period that determine the rows to be deleted from MDW table PS_F_FLOWS and reloaded from PS_GC_FLOW_MGT_F00
Financial Rolling Average Setup	PF_FIN_RAVG_SETUP	EPM Foundation, EPM Setup, ETL Map Parameters, Financial Rolling Avg. Setup, Financial Rolling Average Setup	Enter values that are used to determine rolling averages for loading PS_F_LEDGER.
AP Mart Aging Setup	PF_FIN_AP_SETUP	EPM Foundation, EPM Setup, ETL Map Parameters, AP Mart Aging Setup, AP Mart Aging Setup	Enter values that are used in calculating the AP Aging process.
AR Mart Aging Setup	PF_FIN_AR_SETUP	EPM Foundation, EPM Setup, ETL Map Parameters, AR Mart Aging Setup, AR Mart Aging Setup	Enter values that are used in calculating the AR Aging process.
Customer Segment Ranking Map	PF_CUSEG_RANK_MAP	EPM Foundation, EPM Setup, ETL Map Parameters, Customer Segment Rank Map, Customer Segment Ranking Map	Force rank customer segments.

Financial Mart Account Setup Page

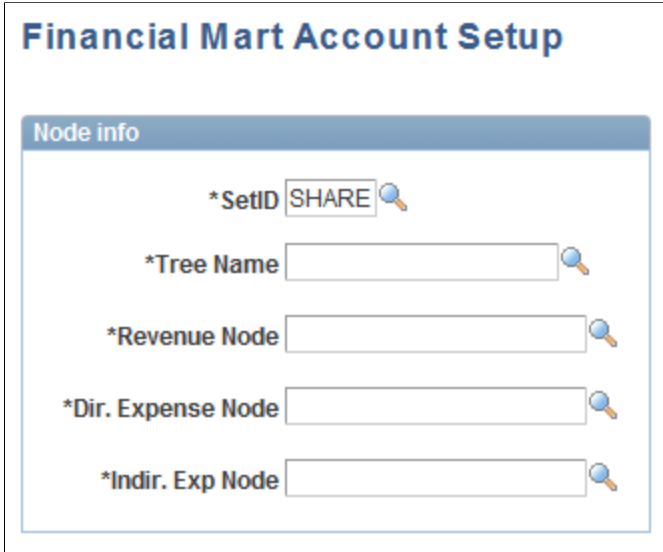
Use the Financial Mart Account Setup page (PF_FIN_ACCT_SETUP) to Enter tree and node details that represent accounts on which Profitability analysis is performed.

Navigation

EPM Foundation, EPM Setup, ETL Map Parameters, Financial Mart Account Setup, Financial Mart Account Setup

Image: Financial Mart Account Setup page

This example illustrates the fields and controls on the Financial Mart Account Setup page. You can find definitions for the fields and controls later on this page.



Use this page to enter details of nodes that belong to the Account tree. Only accounts created under these nodes are loaded by the ETL process into the PS_F_PROFITABILITY table.

SetID	Enter the SetID for the source data.
Tree Name	Enter the name of the Account tree.
Revenue Node	Enter the Account tree node that relates to revenue accounts.
Dir. Expense Node (Direct Expense Node)	Enter the Account tree node that relates to direct expense accounts.
Indir. Exp Node (Indirect Expense Node)	Enter the Account tree node that relates to indirect expense accounts.

Financial Mart GC Cled Setup Page

Use the Financial Mart GC Cled Setup page (PF_FIN_CLED_SETUP) to enter values such as Business Unit, Scenario, and Period that determine the rows to be deleted from Multidimensional Warehouse table PS_F_CLEDGER and reloaded from PS_GC_CLED_MGT_F00 in the MDW table F_CLEDGER.

Navigation

EPM Foundation, EPM Setup, ETL Map Parameters, Financial Mart GC Cled Setup, Financial Mart GC Cled Setup

Image: Financial Mart GC Cled Setup page

This example illustrates the fields and controls on the Financial Mart GC Cled Setup page. You can find definitions for the fields and controls later on this page.

The sequence number on this page indicates that the process accepts multiple sets of business unit, scenario, fiscal year, and accounting period as input. The ETL job processes each set of input data in the sequence indicated.

Load Sequence

Enter the sequence in which the ETL job processes the input parameters. The ETL process accepts multiple sets of business unit, scenario, fiscal year and accounting period as input.

The ETL job processes each set of input data in the sequence indicated.

Business Unit

Enter the performance business unit.

Scenario ID

Enter the scenario ID.

Fiscal Year

Enter the fiscal year.

Accounting Period

Enter the accounting period.

Financial Mart GC Flow Setup Page

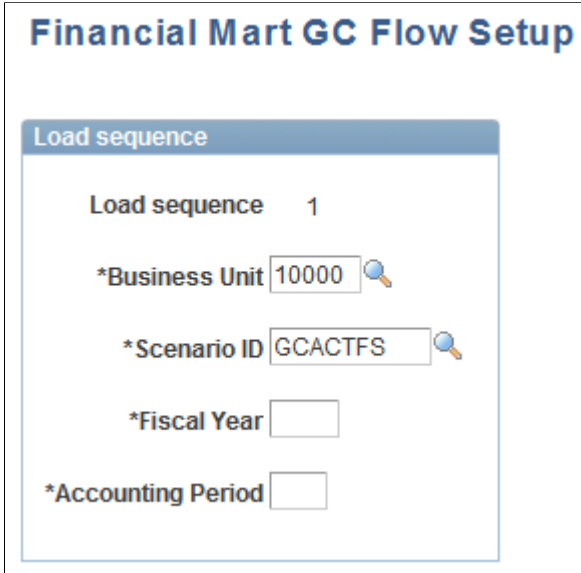
Use the Financial Mart GC Flow Setup page (PF_FIN_FLOW_SETUP) to .

Navigation

EPM Foundation, EPM Setup, ETL Map Parameters, Financial Mart GC Flow Setup, Financial Mart GC Flow Setup

Image: Financial Mart GC Flow Setup page

This example illustrates the fields and controls on the Financial Mart GC Flow Setup page. You can find definitions for the fields and controls later on this page.



Load Sequence

Enter the sequence in which the ETL job processes the input parameters. The ETL process accepts multiple sets of business unit, scenario, fiscal year and accounting period as input. The ETL job processes each set of input data in the sequence indicated.

Business Unit

Enter the performance business unit.

Scenario ID

Enter the scenario ID.

Fiscal Year

Enter the fiscal year.

Accounting Period

Enter the accounting period.

Financial Rolling Average Setup Page

Use the Financial Rolling Average Setup page (PF_FIN_RAVG_SETUP) to enter values that are used to determine rolling averages for loading PS_F_LEDGER.

Navigation

EPM Foundation, EPM Setup, ETL Map Parameters, Financial Rolling Avg. Setup, Financial Rolling Average Setup

Image: Financial Rolling Average Setup page

This example illustrates the fields and controls on the Financial Rolling Average Setup page. You can find definitions for the fields and controls later on this page.

Basis for Rolling Avg. Fact (Basis for Rolling Average Fact) Select the basis for the Rolling Average fact. Values are:

Beginning Period Amount: The ETL process uses the Beginning Amount from the PS_F_LEDGER table to calculate the rolling average.

Ending Period Amount: The ETL process uses the End Amount from the PS_F_LEDGER table to calculate the rolling average.

Net Period Amount: The ETL process uses the Net Amount from the PS_F_LEDGER table to calculate the rolling average.

Rolling Average Periods

Enter the number of periods used in calculating the average value.

AP Mart Aging Setup Page

Use the AP Mart Aging Setup page (PF_FIN_AP_SETUP) to enter values that are used in calculating the AP Aging process.

Navigation

EPM Foundation, EPM Setup, ETL Map Parameters, AP Mart Aging Setup, AP Mart Aging Setup

SetID	Enter the SetID associated with data to be used for the Aging process
Aging ID	Enter the aging ID, for example <i>MONTH</i> .
Calendar ID	Enter the calendar ID.
Accounting Period	Enter the accounting period.
Incl Draft flag (Include Draft flag)	Check to include drafts in AP Aging process.
Aging Date Type	Enter the Aging Date Type. Values are: <i>A</i> : The application uses the Accounting Date. <i>T</i> : The applications uses Invoice Date.
Fiscal Year	Enter the fiscal year.
Incl PrePay flag (Include PrePay flag)	Enter the value to indicate whether to include prepay in the AP Aging process. Values are: <i>Y</i> : Include prepay. <i>N</i> : Do not include prepay.
Unrecord Liab Ind (Unrecorded Liability Indicator)	Enter the value to indicate whether to include unrecorded liability in the AP Aging process Values are: <i>Y</i> : Include unrecorded liability. <i>N</i> : Do not include unrecorded liability.
Effective Date	Enter the effective date of the record.
Source System Identification	Enter the name of the source system.

AR Mart Aging Setup Page

Use the AR Mart Aging Setup page (PF_FIN_AR_SETUP) to enter values that are used in calculating the AR Aging process.

Navigation

EPM Foundation, EPM Setup, ETL Map Parameters, AR Mart Aging Setup, AR Mart Aging Setup

SetID	Enter the SetID for the data used for the AR Aging process.
Aging ID	Enter the number of days used for days sales outstanding (DSO) in the form of a band, for example <i>30-60</i> .
Fiscal Year	Enter the fiscal year.
Incl Grace Days (Include Grace Days)	Enter the value to indicate whether to include grace days in the AR Aging process Values are:

	<i>Y</i> : Include grace days.
	<i>N</i> : Do not include grace days.
DSO Fiscal Year (Days Sales Outstanding Fiscal Year)	Enter the fiscal year used as input to DSO.
Bad Debt Reason	Enter the reason code for bad debt.
Tot AR Specialist (Total Accounts Receivable Specialist)	Enter the number of AR specialists.
Effective Date	Enter the effective date of the record.
Accounting Period	Enter the accounting period.
Sales History ID	Enter the customer sales history ID.
DSO Acct Period (Days Sales Outstanding Accounting Period)	Enter the accounting period used as input to DSO.
Tot Credit Analyst (Total Credit Analyst)	Enter the number of credit analysts.
Tot Collect Analyst (Total Collect Analyst)	Enter the number of collection analysts.
Source System Identification	Enter the name of the source system.

Defining Customer Segment Ranking Parameters for the Customer Segment Data Mart

If you perform customer segment analysis, you may need to define parameters to set up customer segment ranking prior to moving data into the MDW. You must do this if you plan to populate the Segment dimension and perform customer segment ranking. The Segment dimension source the marketing tables RA_LIST and RA_LIST_REC, which come from the OWS.

If the OWS tables RA_LIST and RA_LIST_REC are populated and you plan to analyze customers by segment, you must rank customers on the Customer Segment Ranking Map page before you populate the Segment dimension in the MDW. If you have not populated these source marketing tables, you do not need to set up the customer ranking parameters.

Accessing the Customer Segment Ranking Map page

Use the Customer Segment Ranking Map page (PF_CUSEG_RANK_MAP) to force rank customer segments.

Navigation

EPM Foundation, EPM Setup, ETL Map Parameters, Customer Segment Rank Map, Customer Segment Ranking Map

Use this page to enter the numeric ranking for each customer segment. You must use one unique ranking for each segment. If a customer belongs to more than one segment within a segment group, the ETL

process puts that customer in the highest ranking (lowest number) segment within the selected segment group.

If you do not find any values on the search page for the Customer Segment Ranking Map page, you do not need to use this page; that is, there are no customer segments for you to rank.

To retrieve data to populate this page, run the OWS ETL jobs J_STAGE_PS_RA_LIST and J_STAGE_PS_RA_LIST_REC.

SetID	Displays the SetID for the customer segment group.
Source ID	Displays the source ID in the source database.
Segment Group	Displays the identifier for the segment group.
Segment	Displays the identifier for the segment in the segment group.

Note: The first time you populate the Segment dimension, you are prompted to enter a segment group value. This segment group value remains, unless you must change the value for subsequent runs of the ETL job to populate the Segment dimension. To change the value of the segment group, in DataStage Designer, click the job property J_DIM_PS_P_SEGMENT, click the Parameter tab, and enter the new value for Segment_Group.

Name	Displays the segment name.
Rank	Enter the unique numeric ranking of the segment within the segment group.

Running Initial Setup Jobs

Running Initial Setup Jobs

Initial setup jobs are a group of jobs that you run when you initially populate target OWS tables with data and need to set up common ETL components. These jobs set up your hashed files, shared lookups, and common dimensions, and bring PeopleSoft operational source data into the OWS tables. These jobs are common to all EPM products.

These topics discuss how to:

- Verify ETL components have been imported properly
- Compile ETL jobs
- Run ETL setup jobs to bring source data into EPM
- Run Shared Lookup Jobs
- Run Setup - OWE Jobs
- Run Common Dimension Jobs

Note: Running initial setup jobs are required for both the EPM Warehouses and the Analytical Applications. However, additional implementation jobs are required to set up the EPM Warehouses and the Analytical Applications:

See the topic entitled 'Importing Source Business Units into EPM to Create Warehouse Business Units' in this documentation, for both the EPM Warehouses and the Analytical Applications.

See the topic entitled 'Running the [*product name*] Warehouse Implementation Jobs' in your specific EPM warehouse documentation (for example, Running the HCM Warehouse Implementation Jobs in the HCM Warehouse).

Verifying ETL Components Have Imported Properly

After you have finished configuring DataStage for EPM and imported all of the appropriate *.dsx files (which include different ETL components) you must verify that all the necessary components have been imported properly. This must be performed prior to running any ETL setup jobs.

Verifying Routines

Perform the following steps to verify that your ETL routines are present:

1. In DataStage Designer, attach to your project and expand the Routines node in the left navigation panel of the window.

2. Verify that the object, EPM_Routines, is present in the list of routines.

If this object does not exist in the list, your import of the Common_Utilities.dsx file was unsuccessful. You must re-import the *.dsx file.

Verifying Shared Containers

Perform the following steps to verify that your shared containers are present:

1. In DataStage Designer, attach to your project and expand the Shared Containers node in the left navigation panel of the window.
2. Verify that the objects, Incremental_Logic and Language_Swap, are present in the list of shared containers. The Incremental_Logic object should also contain six components and Language_Swap should contain one.

If these objects do not exist in the list, your import of the Common_Utilities.dsx file was unsuccessful. You must re-import the *.dsx file.

Verifying ETL Jobs

Perform the following steps to verify that your ETL jobs are present:

1. In DataStage Designer, attach to your project and expand the Jobs node in the left navigation panel of the window.
2. Expand each of the sub-folders in the Jobs node, such as Common_Dimensions, Global_Dimensions_E, and Shared_Lookups, and verify that each folder has the requisite ETL jobs in it.

The number of jobs present in each sub-folder vary depending on the product you are implementing.

3. Repeat the first two steps for each product and related project (for example HCM Warehouse).

Compiling ETL Jobs

Before you run any ETL setup jobs, you must compile all jobs first. The jobs should be compiled after you imported the related *.dsx file. The following topics discuss how to verify if your jobs are compiled, and compile those that might not have been.

Verifying ETL Job Compilation

Perform the following steps to verify that your ETL jobs have been properly compiled:

1. In DataStage Director attach to your project and select *View, Status* from the menu.
2. In the left navigation panel of the DataStage Director window, expand the Jobs node.

Verify that the status of all jobs are equal to *compiled*.

If any of the jobs are not compiled, compile them using the steps outlined in the following topics.

Compiling Individual ETL Jobs

Perform the following steps to compile individual ETL jobs:

1. In DataStage Designer, navigate to the job you want to compile, open it, and click on the Compile button.

Image: Compile Button

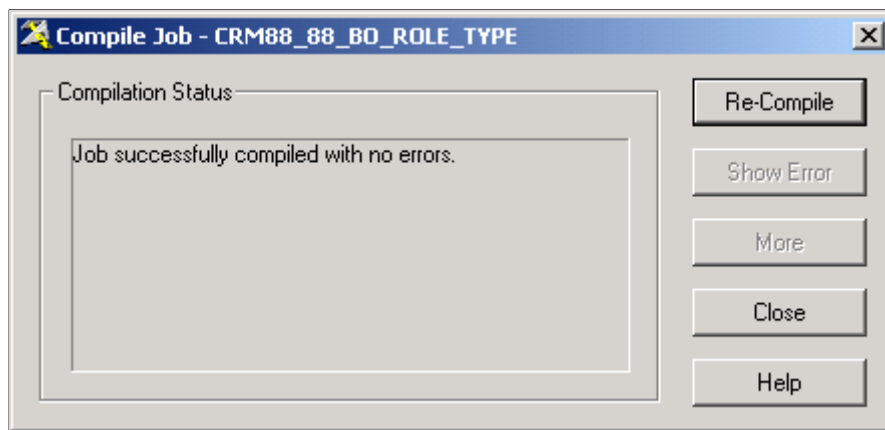
This example illustrates the Compile Button.



After compiling the job you receive a message informing you of the outcome in the Compilation Status window.

Image: Compilation Status Window

This example illustrates the Compilation Status Window.



2. If the job compiled with no errors, click Close.
If the job compiled errors, click Re-Compile.
3. Repeat steps one and two for each job you wish to compile.

Compiling Multiple ETL Jobs

Perform the following steps to compile multiple ETL jobs:

1. In the DataStage Designer attach to your project and select *Tools, Run Multiple Job Compile* from the menu`.

The *DataStage Batch Job Compilation Wizard* opens.

2. In the wizard, select the *Server, Sequence, Only select uncompiled jobs, and Show job selection page* check boxes.
3. The right panel of the wizard window lists all uncompiled jobs.

Click Next.

- Click the Start Compile button.

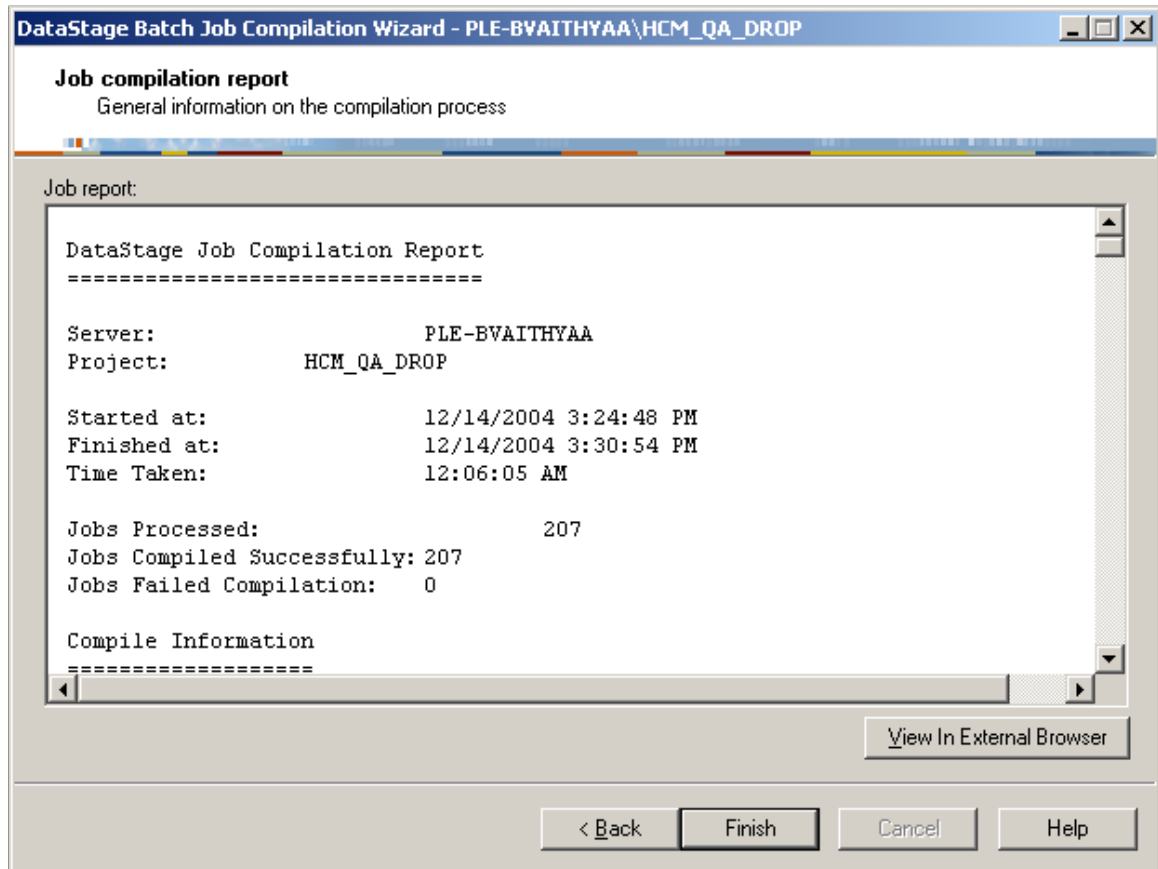
After job compilation is complete, the status for each job reads *Compiled OK*.

- Click Next, then Finish to complete the process.

The Job Compilation Report displays for you to review, should you wish to do so.

Image: Job Compilation Report

This example illustrates the Job Compilation Report.



Related Links

WebSphere DataStage Development: Designer Client Guide

Running ETL Setup Jobs to Bring Source Data Into EPM

After you verify that all ETL components have been successfully imported and all ETL jobs compiled, you are ready to run the jobs which bring your source data into the EPM database (the OWS Load_Hash_Files and Load_Tables jobs).

You have the option of running these jobs manually or using the Master Run Utility.

To run the jobs automatically with the Master Run Utility, follow the steps provided in the ETL Configurations topic of this documentation.

To run the jobs manually, follow the steps described below.

Running Hash Files Setup Jobs Manually

Perform the following steps to manually run hash files setup jobs:

1. In DataStage Director, navigate to the hash file jobs by expanding the nodes in the left navigation panel using the following path: *Setup_E, OWS, <Warehouse Code>, Base, Load_Hash_Files, Server*.

Note: *Warehouse Code* refers to each of the EPM Warehouse products (for example CS Warehouse or HCM Warehouse).

2. Select each hash file setup job in the Job Status view and select *Job, Run Now...* from the menu.

The Job Run Options box appears.

3. Update the job parameters if necessary and click Run.

The job is scheduled to run with the current date and time, and the job's status is updated to *Running*.

Running the Setup - OWS Jobs Manually

Perform the following steps to manually run setup - OWS jobs:

1. In DataStage Director, navigate to the setup jobs by expanding the nodes in the left navigation panel using the following path: *Setup_E, OWS, <Warehouse Code>, Base, Load_Tables, Sequence*.

Note: *Warehouse Code* refers to each of the EPM Warehouse products (for example, CS Warehouse or HCM Warehouse).

2. Select each setup - OWS job in the Job Status view and select *Job, Run Now...* from the menu.

The Job Run Options box appears.

3. Update the job parameters if necessary and click Run.

The job is scheduled to run with the current date and time, and the job's status is updated to *Running*.

Running Shared Lookup Jobs

Shared lookups function the same as hash file lookups—they act as views of specific EPM warehouse tables and contain only a subset of the data available in a warehouse table. These streamlined versions of warehouse tables are used to perform data validation (lookups) within an ETL job and select specific data from lookup tables (such as sourceID fields in dimensions). The only difference between a regular lookup and a shared lookup is that the shared lookups are used across all EPM products.

Because shared lookups are essential in the lookup process, jobs cannot function properly until all hash files are created and populated with data. Before you run any job that requires a hash file, you must first run all jobs that create and load the hash files—also called initial hash file load jobs.

Steps Required to Run Shared Lookup Jobs

Perform the following steps to run the shared lookup jobs:

1. In DataStage Designer, attach to your project and expand the Shared_Lookups node in the left navigation panel of the window.

The following sub-folders exist in the Shared_Lookups node:

- Control_Tables
- DimensionMapper_Lookups
- Language_Lookups
- System_Lookups

2. Select one of the sub-folders.
3. Select the lookup jobs in the Job Status view and select *Job, Run Now...* from the menu.

The Job Run Options box appears.

4. Update the job parameters if necessary and click Run.

The job is scheduled to run with the current date and time, and the job's status is updated to *Running*.

5. Repeat steps two and three for the remaining sub-folders.

Running Setup - OWE Jobs

Setup - OWE jobs load the setup tables used in standard OWE jobs (jobs that move your operational data from the OWS to the OWE). You can run these jobs manually or use the Master Run Utility. To run the jobs automatically with the Master Run Utility, follow the steps provided in the ETL Configurations topic of this documentation.

Perform the following steps to run the setup - OWE jobs manually:

1. In DataStage Director, navigate to the setup OWE jobs by expanding the nodes in the left navigation panel using the following path: *Setup_E, OWE, Base, Load_Tables, Sequence*.
2. Select each setup - OWE sequence job in the Job Status view and select *Job, Run Now...* from the menu.

The Job Run Options box appears.

3. Update the job parameters if necessary and click Run.

The job is scheduled to run with the current date and time, and the job's status is updated to *Running*.

See [Using the Master Run Utility to Automatically Run Your ETL Jobs](#).

Running Common Dimension Jobs

Common dimensions are dimensions that are shared across *all* EPM products. Not only do these dimensions play an important role in all reporting and analytical analysis, but they are particularly important to the Allocation Manager data enrichment tool, used by EPM Analytical Applications. In Allocation Manager, these dimensions are used to determine the divisor, therefore the ratio, for the spread even and prorata methods.

Common dimension jobs can be divided into the following five categories:

- Business_Unit
- Calendar
- Currency
- Language
- Unit_Of_Measure

The common dimension master sequence jobs can be found in the following DataStage Director paths:

- Common_Dimensions\E\Business_Unit\Master_Sequence
- Common_Dimensions\E\Calendar\Master_Sequence
- Common_Dimensions\E\Currency\Master_Sequence
- Common_Dimensions\E\Language\Master_Sequence
- Common_Dimensions\E\Unit_Of_Measure\Master_Sequence

Note: For all dimension load jobs (common dimension, global dimension, local dimension, OWE dimension, MDW dimension), users can customize the error validation by providing the environmental variable with the appropriate values. If you want to skip error validation, set \$ERR_VALIDATE to 'N.' If you want to perform error validation, set \$ERR_VALIDATE to 'Y.' Also, you can specify the threshold limit for the error validation. If you want the job to abort if the lookup fails more than 50 times, set \$ERR_VALIDATE to 'Y' and \$ERR_THRESHOLD to 50. This can all be done using DataStage Administrator.

Running Common Dimensions Jobs

Perform the following steps to run the common dimension jobs (the order reflects the master sequence order):

1. In DataStage Director, navigate to the MSEQ_E_Hash_Calendar (Calendar) master sequence by expanding the nodes in the left navigation panel using the path defined in the [Running Setup - OWE Jobs](#) topic.
2. Select the MSEQ_E_Hash_Calendar master sequence job in the Job Status view and select *Job, Run Now...* from the menu.

The Job Run Options box appears.

3. Update the job parameters if necessary and click Run.

The job is scheduled to run with the current date and time, and the job's status is updated to *Running*.

4. Repeat steps one through three for the remaining master sequence jobs, using the following order:
 - a. MSEQ_E_OWE_BaseDim_Calendar (Calendar)
 - b. MSEQ_E_OWS_BaseDim_Calendar (Calendar)
 - c. MSEQ_E_Hash_BU (Business Unit)
 - d. MSEQ_E_OWE_BaseDim_BU (Business Unit)
 - e. MSEQ_E_OWS_BaseDim_BU (Business Unit)
 - f. MSEQ_E_Hash_Currency (Currency)
 - g. MSEQ_E_OWE_BaseDim_Currency (Currency)

Importing Source Business Units into EPM to Create Warehouse Business Units

Understanding Warehouse Business Units, TableSet Sharing, and SetID Mapping

As a part of EPM setup, you must create *warehouse business units* (WBU), and establish a mapping between warehouse business units and the business units that exist in the source systems that you are bringing into EPM. The mapping between source business units and warehouse business units has implications for the appropriate SetID assignments for warehouse business units on warehouse *record groups*. Setting up warehouse business units is mandatory regardless of whether you plan to implement only data marts, only Analytical Applications, or some combination of the two.

SetIDs and TableSet Sharing

In PeopleSoft source systems (such as Campus Solutions or CRM), the rows in a control table (such as supplier, customer, and account) are divided into groups based on a key called the SetID. Each group of rows with the same SetID constitutes a *tableset*.

Image: Tablesets on a hypothetical supplier table

This diagram illustrates the tablesets on a hypothetical supplier table with two SetIDs represented in red and blue

Hypothetical Supplier Table	
SETID	SUPPLIER_NAME
RED	Pepsi Inc.
RED	Coca-cola Corp.
RED	Seven Up Bottling
BLUE	Frank's Bottling Co.
BLUE	Jane's Beverages, Inc.
BLUE	Seven Up Bottling
Etc.	Etc.

Every business unit associates with a particular SetID on every control table. For example, BU1 might use the red tableset on the supplier table. This assignment serves to limit the suppliers that appear in prompts for transactions that take place in the context of BU1. Several other business units could also use the red SetID on the supplier table, in which case these business units share the red tableset (hence the term *tableset sharing*). Since control tables in a PeopleSoft source system generally become dimensions in EPM, most EPM dimensions are SetID-based.

Record Groups

Control tables such as supplier are themselves grouped into record groups based on the commonality of business process. SetID assignments actually take place at the record group level. In other words, a business unit must have the same SetID on all control tables in a given record group. Each business unit in a PeopleSoft source system must have a SetID assignment on every record group. At the time a business unit is created, the system assigns a default SetID to all record groups for that business unit. The user can then manually modify the SetID assignments as desired. For example, the following table illustrates the SetID assignment for a few hypothetical business units on a few hypothetical record groups:

Image: SetID assignments for hypothetical record groups

This diagram provides an example SetID assignments for record groups.

A Hypothetical Set Control "Space"					
	RG1	RG2	RG3	RG4	Etc.
BU1	RED	RED	BLUE	YEL	
BU2	RED	RED	BLUE	YEL	
BU3	RED	BLUE	BLUE	YEL	
BU4	BLUE	RED	BLUE	YEL	
BU5	BLUE	RED	RED	YEL	
BU6	BLUE	RED	BLUE	PINK	
Etc.					

Assuming that the supplier table shown earlier is included in record group RG1, then (given the SetID assignments indicated in the figure above) BU1, BU2, and BU3 would have access to the first three suppliers, while BU4, BU5, and BU6 would have access to the last three.

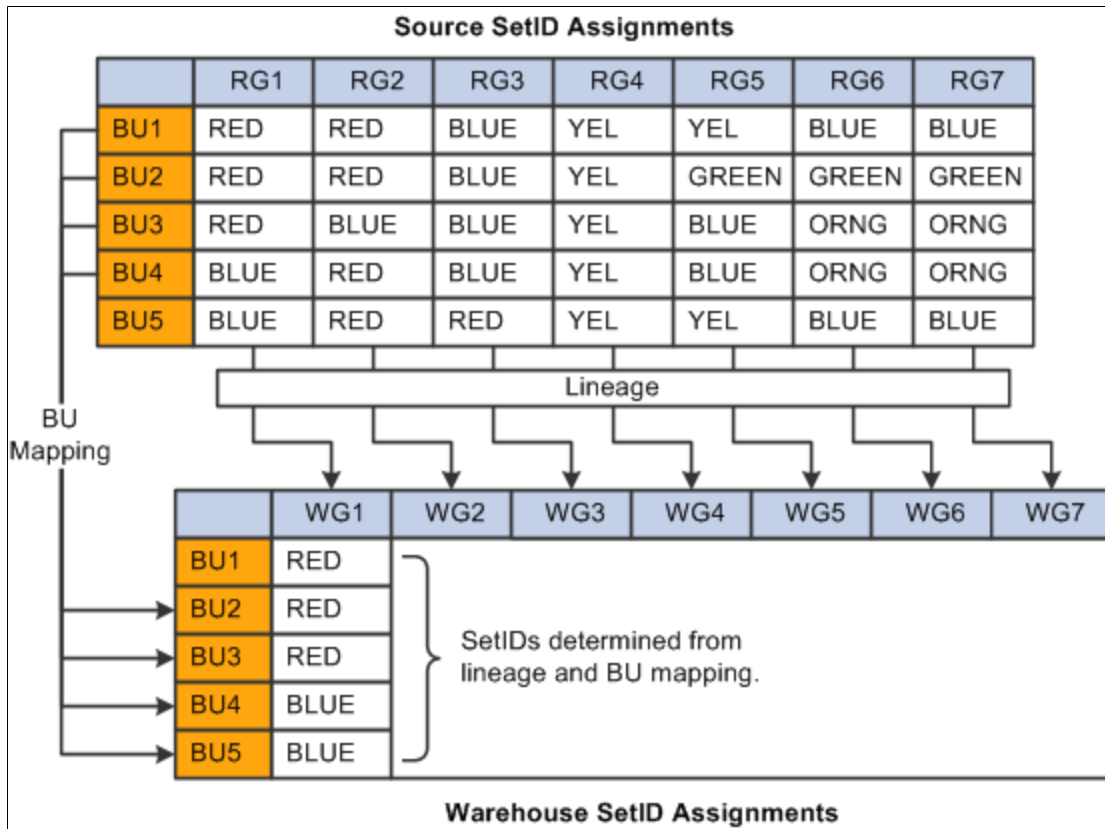
Mapping Two SetID Based Systems Together and Warehouse Lineage

EPM is a SetID-based database. Consequently, all warehouse business units must have SetID assignments on all warehouse record groups. However, because each PeopleSoft source system has its own tableset sharing (or *set control space*), the mapping between source business units and warehouse business units has implications for how SetIDs must relate between the source system and the warehouse. As illustrated below, in order to correspond two set control spaces, it is necessary to provide two distinct mappings: the business unit mapping (which is under user control) and the warehouse *lineage*. Warehouse lineage defines the correspondence between all PeopleSoft source tables and EPM warehouse tables.

See [Warehouse Lineage and Source Blueprints](#).

Image: Hypothetical source to warehouse set control space mapping

This diagram provides an example of source to warehouse set control space mapping.



The preceding figure illustrates two set control spaces and the mappings between them, with SetID values shown as colored squares. In this example, warehouse business units have been created and mapped to source business units in a one-to-one relationship. Based on the lineage, it is possible to determine which SetID values to assign to the corresponding warehouse record groups. Without the lineage information this determination would not be possible.

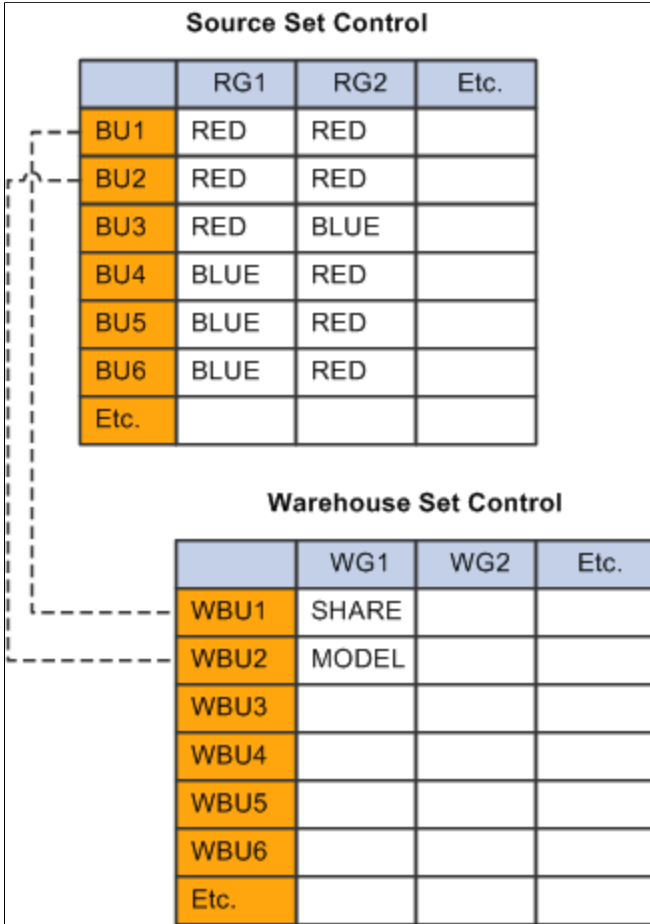
Potential SetID Conflicts

SetIDs from one or more sources can merge into a single warehouse SetID (in other words, "many-to-one" relationships are allowed). However, the system does not allow one-to-many relationships to exist between source SetIDs and warehouse SetIDs on the same warehouse record group. For example, in the illustration below, BU1 and BU2 shared SetID RED in the source. Assuming that RG1 is related by lineage to WG1, then the situation illustrated would force one SetID (RED) to map to two different values (SHARE and MODEL) on warehouse record group WG1. This situation is not allowed because it would create a multi-valued lookup that would cause errors in the ETL process. Consequently the system does not allow this configuration to pass its validation step. You must modify the warehouse SetID assignments

to remove the splitting. In the case illustrated below, simply assigning the same warehouse SetID to WBU1 and WBU2 on record group WG1 resolves the problem.

Image: SetID splitting

This diagram illustrates SetID splitting.

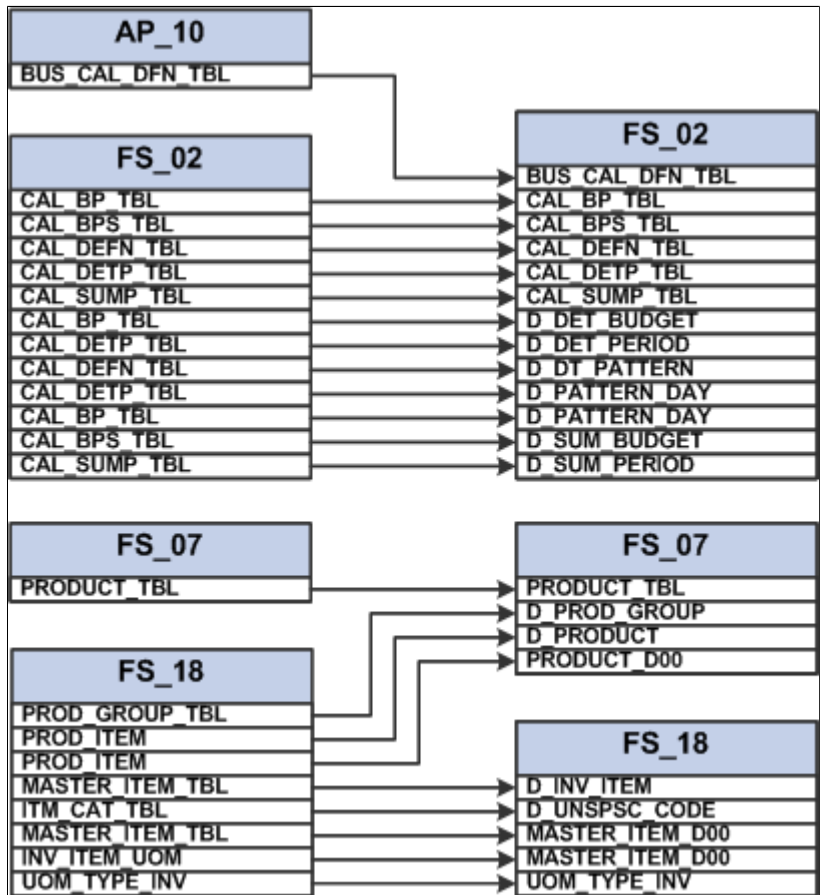


FSCM and SetID Conflicts

Business unit merging is not the only way that SetID conflicts can arise. There are two instances in the FSCM product that has a lineage that can introduce conflicts, as shown below:

Image: SetID conflicts in the FSCM to warehouse mapping

This diagram illustrates SetID conflicts when mapping FSCM to the EPM warehouse.



The problem occurs when record groups from a source merge in the warehouses (for example, AP_10 and FS_02 merging into FS_02, as shown in the diagram above). When you use the Business Unit Creation Wizard, it assigns an initial guess for the SetIDs of the warehouse business units that it creates for you. A problem can occur in the case where two or more business units have different SetIDs on AP_10 and FS_02. For example, if you had two business units in FSCM (BU1 and BU2), and assigned them both SetID RED on AP_10, and BLUE on FS_02, it is possible that the wizard could select RED as the SetID assignment for FS_02 for WBU1, and BLUE for the SetID of WBU2. In this case the wizard would detect splitting because it would see both RED and BLUE from the source being split to RED and BLUE on the target. The solution to this problem is straightforward: you should manually create the warehouse business units, and assign the correct SetID. Then you can map the business units and do the validation in the Business Unit Mapping page. If you have an FSCM source with a complex SetID configuration and you detect conflicts related to AP_10, FS_02, FS_07, or FS_18, it may be due to this lineage problem.

Warehouse Lineage and Source Blueprints

Warehouse lineage defines the relationship between PeopleSoft transaction tables and EPM target warehouse tables, as determined by the ETL process. Certain ETL jobs move data from PeopleSoft source tables to EPM Operational Warehouse - Staging (OWS), Operational Warehouse - Enriched (OWE), and

Multidimensional Warehouse (MDW) target tables, and warehouse lineage simply refers to the path the data takes between the two systems. Lineage information is used during the warehouse business unit creation process to determine the appropriate PeopleSoft source SetID for a given warehouse business unit, and map that SetID to the appropriate warehouse record group.

Source blueprints actually *record* the warehouse lineage, and ship with EPM as system data.

Warehouse lineage information is only required for SetID-based source systems. As such PeopleSoft delivers blueprints for PeopleSoft source systems only. Typically, third-party source systems do not use a SetID column, and thus have no impact on the blueprints. You need not create blueprints for non SetID-based third-party source systems.

Modifying a Source Blueprint Due to Customizations

If you customize your ETL jobs or add columns to an existing source or target table, you may need to modify the source blueprints to reflect these changes. It is only necessary to update the blueprints if you add a new SetID-based table to your PeopleSoft source system and map the table into new, SetID-based dimension in the warehouse. In that case, simply access the Source Blueprint page and add a row for each new dimension.

Remapping Business Units for an Updated Source Blueprint

If you receive an updated Source Blueprint from PeopleSoft, you must remap business units manually or with the business unit wizard after running the dms/dat. This creates the source to target mapping. All entries must be truncated and rerun.

Understanding Warehouse Business Unit Setup

Warehouse business units are created using several processes, including ETL jobs, defining parameters in PeopleSoft pages, and running a setup and validation wizard.

The following steps are required to establish warehouse business units:

1. Run prerequisite setup jobs.
2. Create SetIDs in the warehouse.
3. Define source systems.
4. Establish warehouse lineage.
5. Create warehouse business units.
6. Assign warehouse SetIDs to warehouse business units on all warehouse record groups.
7. Map source business units to warehouse business units.
8. Validate the business unit mapping.

PeopleSoft delivers the *Business Unit Creation Wizard* that automates many of these steps. However, you can manually perform these steps if you prefer.

Establishing Default SetIDs - Overview

You must define a default SetID for each source that you wish to bring into the warehouse. SetIDs are created in a PeopleTools page, where you can add a new value for each default SetID that you need to create. Generally you should use a different default SetID for each source to keep tablesets from all sources separate—unless you have a good reason why data from different tablesets should merge. The wizard uses the default SetID for all business units that it creates on all warehouse record groups for which the source blueprint supplies no lineage information. For example, if you bring in a CRM source, the CRM blueprint has no lineage information for many warehouse tables that relate to HCM or FSCM, simply because there are no source tables in CRM that relate to these tables. All unspecified warehouse tables receive the default SetID.

Establishing Warehouse Sources - Overview

For each PeopleSoft source that you are bringing into the warehouse, use the Define Warehouse Sources page to define the properties of your source. The Business Unit Creation Wizard uses the defaults specified on the page for warehouse business unit creation. You should define a warehouse source even for third-party systems that are not SetID-based.

The procedure for defining warehouse sources is discussed in the [Specifying Your EPM Sources](#) topic.

Note: To review the sources you currently have defined for your system, run the PS Query *DMRP_2_1_SRC_SYSTEM_TBL*.

Establishing Warehouse Lineage - Overview

Establishing lineage consists of two parts:

- Reviewing or updating source blueprints.
- Defining warehouse lineage.

Review or Update Source Blueprints

Source blueprints are used in establishing warehouse lineage. PeopleSoft delivers source blueprints for PeopleSoft sources. If you have added new SetID-based tables to your PeopleSoft system and you wish to bring these tables into the EPM database, you must update the source blueprints accordingly.

Define Warehouse Lineage

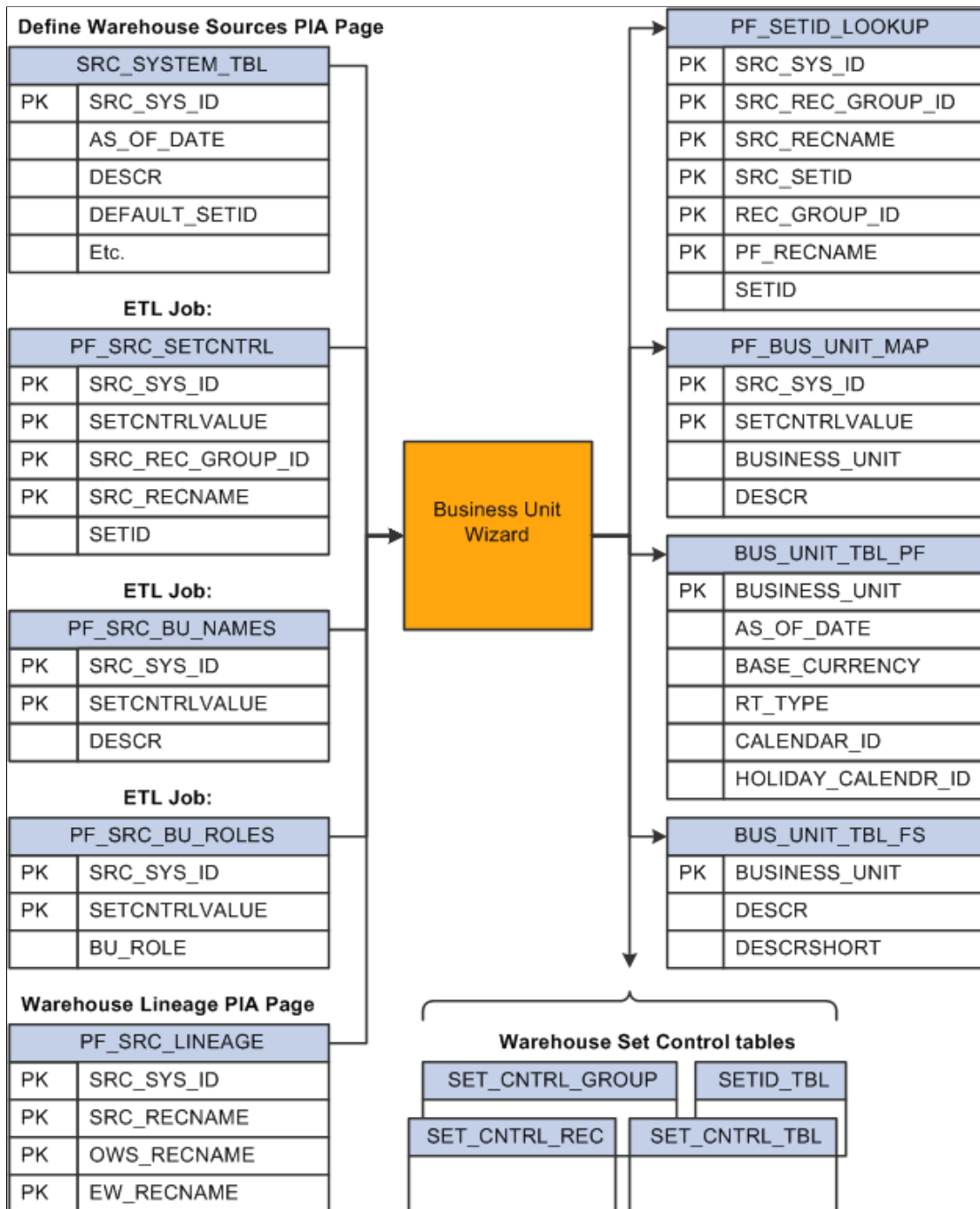
After you have defined your sources, you must associate a blueprint to each source by using the Warehouse Lineage page. The system ships with blueprints for all supported systems. Note that the FSCM Blueprint includes lineage information for ESA, SCM, and PeopleSoft Financials. Enterprise Learning Management (ELM) is not SetID-based, so you can use the NONSETID blueprint for ELM sources (you should also use the NONSETID blueprint for third-party systems that have no SetID). When you save the warehouse lineage page the system populates the PF_SRC_LINEAGE table with the combined lineage for all warehouse sources.

Creating Warehouse Business Units with the Business Unit Creation Wizard - Overview

The Business Unit Creation Wizard automates many of the steps required to set up warehouse business units and SetIDs. Before you can use the wizard, you should understand the input and output tables used by the wizard.

Image: Input and output tables used by the Business Unit Creation Wizard

This diagram illustrates the inputs that the wizard uses, and the output tables that it populates.



The Business Unit Creation Wizard uses the following input tables:

- **SRC_SYSTEM_TBL**: This table defines all the pillars that you are bringing into EPM. You populate this table using a PIA page
- **PF_SRC_SETCNTRL**: This table records the set controls (a combination of business units and SetIDs) from all of your sources, and the SetID assignments for all set controls on all source record groups. It is populated by ETL.
- **PF_SRC_BU_NAMES**: This table provides a distinct list of business units from all sources. It is populated by ETL
- **PF_SRC_BU_ROLES**: This table records all of the roles that associate to each source business unit, such as AP (accounts payable business unit), GL (general ledger business unit), and IN (inventory business unit).
- **PF_SRC_LINEAGE**: This table is a compilation of all of the Source Blueprints that you specify for all active sources that you create. It is populated by a PIA page when you save the Blueprint assignments (see below).

The Business Unit Creation Wizard populates the following output tables:

- **PF_SETID_LOOKUP**: The ETL process uses this table to map source SetIDs to warehouse equivalents.
- **PF_BUS_UNIT_MAP**: The ETL process uses this table to map source business units to warehouse equivalents.
- **BUS_TABLE_PF**: The Wizard makes entries in this table for all warehouse business units that it creates. These tables are used by the analytical applications.
- **BUS_TABLE_FS**: Same as **BUS_UNIT_TABLE_PF**
- **Warehouse Set Control Tables**: The wizard makes the necessary entries in the warehouse set control tables when it assigns SetIDs to each warehouse business unit on each warehouse record group.

The Business Unit Creation Wizard consists of three steps:

1. **Reviewing Incoming Set Controls, Business Unit Wizard Step 1**: Review the set controls from all of your sources and ensure you see the business units and SetIDs that you expect from each source.
2. **Addressing Collisions Between Set Controls, Business Unit Wizard Step 2**: Examine name collisions between and among incoming set controls and set controls that may already exist in the warehouse. You should resolve the name conflicts by renaming the incoming set control unless you have a good reason to merge them.
3. **Validating the Mapping, Business Unit Wizard Step 3**: After the wizard has analyzed the SetID assignments for all incoming set controls, you must review the mapping report produced by the wizard to see if any conflicts exist. If conflicts exist, you must correct the conflicts yourself or let the wizard correct the conflicts.

Note: Prior to running the Business Unit Creation Wizard, you must run the ETL jobs that populate the aforementioned input tables. These jobs are collectively described as dimension mapper ETL setup jobs. The Wizard populates the tables indicated above as output tables. For a new EPM installation, the output tables are empty. In the event that you make a mistake and need to re-run the wizard, you must truncate the output tables manually; no provision is made for the wizard to delete these entries. If you have already loaded data into the warehouse and then decide to re-run the wizard, the data you have already loaded may become invalid.

Creating Warehouse Business Units Manually - Overview

You can manually setup warehouse business units. The steps are outlined here:

Create Warehouse Business Units

Manually define all the warehouse business units that you require, using the Warehouse Business Unit page. You must define a default SetID, a default currency rate type, and a calendar for each warehouse business unit.

Assign Warehouse SetIDs to Warehouse Business Units

As you create warehouse business units, you are prompted to provide a default SetID. The default SetID is automatically assigned to the new business unit on all record groups. In order to modify these assignments, you must use the Tableset Control page. Select the business unit (set control) of interest and modify its SetID assignments on the record groups of your choice.

If several warehouse business units have exactly the same SetID assignments, it's possible to copy the SetIDs of an existing business unit. Define the SetID assignments for one warehouse business unit, and then as you create new ones, use the name of the first warehouse business unit as the default SetID of subsequent warehouse business units. The system assigns each record group the same SetID as the copied business unit.

Map Warehouse Business Units

Once warehouse business units have been created, you must map them to source business units using the Business Unit Mapping page. It's possible to associate several source business units with a single warehouse business unit, thus creating a many-to-one mapping. However this is not good practice since it promotes SetID conflicts, and it could cause fact table collisions as several business keys merge into one. PeopleSoft delivered ETL jobs do not support aggregating fact data during loading.

Validate Business Unit Mapping

Once you have created your mapping configuration, validate the mapping. The system analyzes the mapping configuration. If all of the set controls have been made unique as suggested above, the potential for conflicts in your business unit mapping is minimized. If conflicts are detected, you have two choices:

- Let the system correct the conflicts: The simplest choice is to click the "Accept Proposed SetIDs and Save" button. This causes the system to automatically reassign the SetIDs used by warehouse business units on various warehouse record groups in order to make the configuration valid. Generally this reduces the number of warehouse SetIDs to eliminate conflicts.

- **Modify business unit Mapping:** You can choose to resolve the conflicts manually. This requires that you analyze the report to determine which warehouse SetIDs must change in order to avoid SetID Splitting.

Regardless of how you fix the problem, if conflicts are detected, you must return to the business unit mapping page and try the validation again, until no conflicts exist. Continue to iterate the validation process until you see no more conflicts. Once the conflicts are resolved, the process of creating business units and assigning SetIDs is complete. You only need to inspect and modify the properties of the new warehouse business units.

Working with Invalid or Unused Source Business Units - Overview

The ETL logic that extracts source business unit data and brings it into the EPM database does not filter-out invalid or unused source business units. Hence, if you have invalid or unused business units in your PeopleSoft source, you can:

- Create a dummy Warehouse Business Unit (WBU) and map the undesired source business units to the dummy WBU.
- Reconfigure the Dimension Mapper setup jobs to filter out the source business units that you do not want to bring into the EPM database.

However, it is not recommended that you reconfigure the Dimension Mapper setup jobs or use SQL to delete business units from the internal Dimension Mapper tables (the unwanted business units will reappear the next time you run the Dimension Mapper setup jobs).

Working with PeopleSoft Human Capital Management (HCM) Source Business Units - Overview

As part of your configuration of the PeopleSoft HCM source system, you used the Company component to enter information about a single company or multiple companies in your organization. You assigned a three-character code for each company you defined in the HRMS source system. During the ETL process, those company codes are brought into the PF_SRC_SETCNTRL table, which records the source set controls, and is used as an input for the warehouse business unit mapping process.

The company codes reside in the same table (PF_SRC_SETCNTRL) as your source business units. Hence, when you map source business units to warehouse business units, be sure not to include the company codes in your mapping. If you have a large number of company codes, you may wish to filter-out these codes in the ETL job.

Filtering Company Codes in the ETL Job - Trans_Assign_Values Stage

If you are sure that all SETID and BUSINESS_UNIT values are five-characters in length (as is recommended), and all company codes are three-characters in length, you can access the ETL job

J_Stage_PS_S_SET_CNTRL_REC_HCM_HCM91_EPM91 using DataStage Designer and filter all rows with `Len(IPC_in.SETCNTRLVALUE) <= 3` in the *Trans_Assign_Values* Stage.

Image: Modifying the Stage Constraint, 1 of 2

This example illustrates the Modifying the Stage Constraint, 1 of 2.

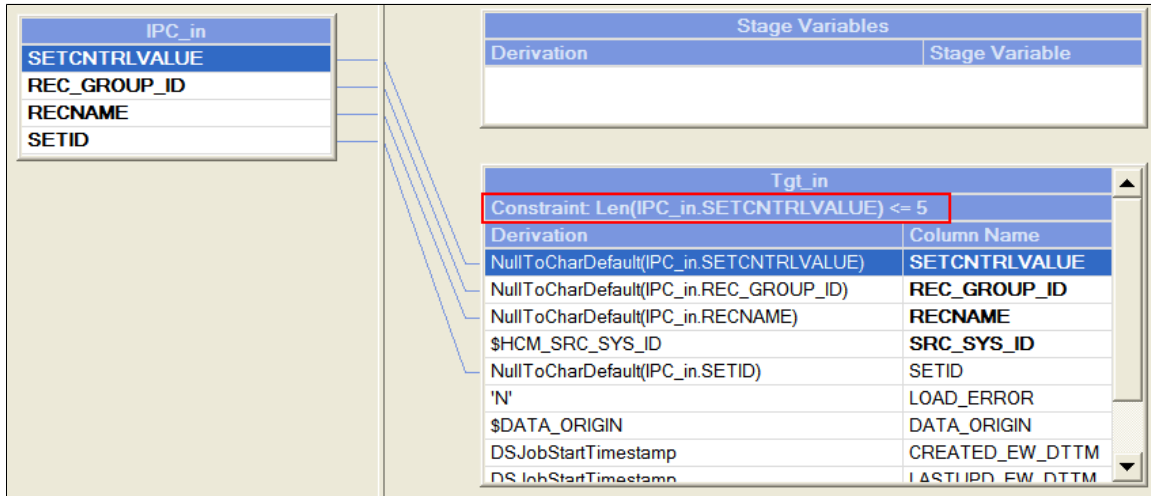
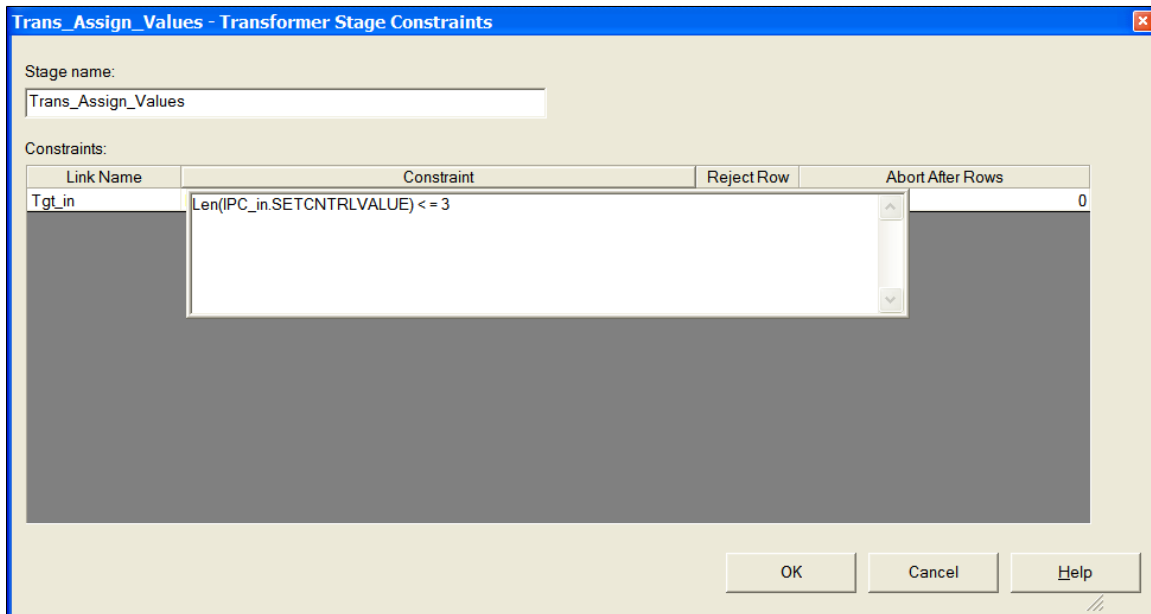


Image: Modifying the Stage Constraint, 2 of 2

This example illustrates the Modifying the Stage Constraint, 2 of 2.



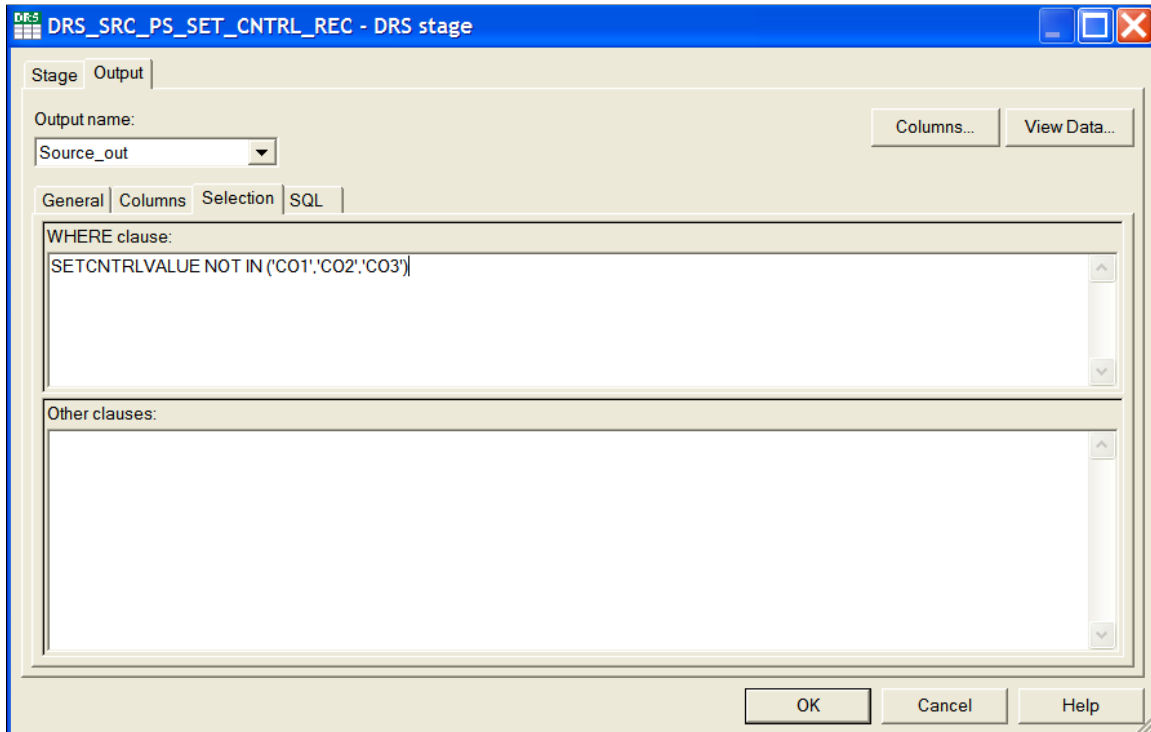
However, if you perform this configuration and you set up table-set sharing using company as the driving parameter along with business unit, the same table-set sharing information would be prevented from entering EPM and the business unit wizard would *not* be able to retain table-set sharing based on company codes in the HRMS source system.

Filtering Company Codes in the ETL Job - Source Stage

Alternatively you can create a WHERE filter in the selection-output of the source stage (DRS_SRC_PS_SET_CNTRL_REC) in the ETL job *J_Stage_PS_S_SET_CNTRL_REC_HCM_HCM91_EPM91*.

Image: Creating a WHERE filter in the selection-output of the source stage

This example illustrates the Creating a WHERE filter in the selection-output of the source stage.



The WHERE filter above is just an example, it is not meant to be the exact WHERE filter you should create for the source stage.

Reviewing Warehouse and General Ledger Business Unit Properties - Overview

After creating and mapping warehouse business units, you should review your warehouse business units and general ledger business units to ensure the properties (such as default calendar) meet your requirements.

Review Warehouse Business Unit Properties (Business Unit Creation Wizard Only)

After creating warehouse business units with the Business Unit Creation Wizard, access the Warehouse Business Unit page to review the detailed properties of each business unit. Some of your warehouse business units may have been created with a base currency or rate type that differs from the defaults that you defined for your source. If this is the case, you need to change these settings for the appropriate business units. In addition, the Business Unit Wizard does not associate calendars to business units. You must do this manually for all your warehouse business units.

Review General Ledger Business Unit Properties

PeopleSoft general ledger business units (GLBU) are extracted from your source system and populated in the EPM database using ETL jobs; you do not need to recreate them in the warehouse. You can view general ledger business units by accessing the General Ledger Business Unit page. You may, on some occasions, create general ledger business units manually in the warehouse for certain analytical applications. See your EPM application documentation for more details.

Creating Collision Maps - Overview

If you choose to allow two tablesets to merge in the EPM database, it is possible for collisions to occur between business keys (such as supplier ID). For example two suppliers from two different tablesets could both have the business ID "PEP", but could refer to very different suppliers (for example, Pepsi and Pep Boys). Collision maps provide a framework for resolving collisions between business IDs from two or more tablesets that merge in the warehouse. PeopleSoft provides three resolution methods for colliding business IDs: First In Wins, Error-Out Duplicates, and Use Mapping Table.

First In Wins

In this method, the first instance of a business ID is loaded into a tableset, and subsequent instances of the same ID from different tablesets is ignored. This approach is appropriate for large datasets, when the number of collisions is known to be small and the value of fixing errors is low. You can inspect the business keys that have been ignored by navigating to the Collision Map Error Report page.

Error Out Duplicates

In this method the system automatically loads the first instance of a business ID into a tableset, but subsequent instances of the same ID from different tablesets is sent to an error table where they can be inspected and remapped. This approach is appropriate for relatively small datasets, where the value of fixing errors is high. You must inspect and correct the errors manually by navigating to the Collision Map Error Reports page for the appropriate map.

Use Mapping Table

When using this mapping type, the system checks every incoming business key against a mapping table. If an entry does not exist in the mapping table, then the row errors-out. You can inspect the errors in the appropriate error report. For those entries that error out, you must update the mapping table and re-run the appropriate ETL map. This mapping method is relevant to the case where you have created a mapping table using an offline process or third-party tool.

Prerequisites to Creating Warehouse Business Units

This topic provides you with prerequisites to creating warehouse business units and discusses how to:

- Run prerequisite ETL setup jobs.
- Size tablespaces for input tables.
- Create backups of impacted output tables.
- Enable PS Queries

- Verify the state of output tables
- Validate business unit and set control data
- Verify source blueprints

Running Prerequisite ETL Setup Jobs

Before you can begin creating warehouse business units you must run specific ETL jobs that setup certain OWS and OWE tables, and bring your source business unit data into EPM tables. Some of these jobs include OWS hash file, setup OWS, shared lookup, and setup OWE jobs.

The general OWS and OWE ETL setup jobs are not discussed here because the information is covered in another topic of this documentation. However, the ETL jobs that are specific to business unit data are discussed below.

See [Understanding IBM WebSphere DataStage Setup and Configuration](#) and [Verifying ETL Components Have Imported Properly](#).

Running Hash File - Business Unit Data Jobs

To run the hash file jobs that pertain to business unit data:

- In DataStage Director, navigate to the hash file jobs for business unit data using the following navigation: *Setup_E, Dimension_Mapper, Base, Load_Hash_Files, Server*.
- Select the jobs in this category and run.

Running Setup - Business Unit Tables Jobs

To run the setup jobs that load business unit setup tables:

- In DataStage Director, navigate to the hash file jobs for business unit data using the following navigation: *Setup_E, Dimension_Mapper, Base, Load_Tables, Sequence*.
- Select the jobs in this category and run.

Running Shared Lookup - Business Unit Data Jobs

To run the shared lookup jobs that pertain to business unit data:

- In DataStage Director, navigate to the shared lookup jobs for business unit data using the following navigation: *Shared_Lookups, DimensionMapper_Lookups*
- Select the jobs in this category and run.

Sizing Tablespaces for the Input Tables

The Business Unit Creation Wizard input tables use the *EWLARGE* tablespace, while the *PSINDEX* tablespace is used for indexes. These table spaces should be large enough to store the data in input, output and intermediate tables related to warehouse business unit creation.

A good rule of thumb is to make sure that each of the two tablespaces are at least twice the size of the *PF_SRC_SETCNTRL* table. Determine the maximum size for a row of data in *PF_SRC_SETCNTRL*

by examining this table in your database. Estimate the number of rows expected in this table, and then multiply these factors together. Double the result to get a minimum tablespace size estimate: Minimum tablespace size = 2 * #Rows * Size of one row of PF_SRC_SETCNTRL.

Creating Backups of Impacted Output Tables (Optional)

Create a DAT file backup of the output tables that are impacted by the warehouse business unit creation process. That way, if you encounter problems during the creation process you can use the DAT file to roll back your system.

Run the prepackaged Data Mover script *DMBK_CREATE_DM_BACKUP* to produce the backup DAT file of the impacted output tables. The following are the output tables backed up by the DAT file:

- SETID_TBL
- SET_CNTRL_TBL
- SET_CNTRL_GROUP
- SET_CNTRL_REC
- BUS_UNIT_TBL_FS
- BUS_UNIT_TBL_PF
- BUS_UNIT_TBL_GL
- PF_BUS_UNIT_MAP
- PF_SETID_LOOKUP

To roll back your system, run the Data Mover script *DMBK_RESTORE_TABLES*.

Enabling PS Queries for Data Verification

The following topics provide instructions on how to verify different aspects of your business unit related data. To perform the verifications you must run certain prepackaged PS Queries. To obtain the specific queries you must first enable the queries and the query security by installing the *DMRP_QUERIES* Application Designer project.

Perform the following steps to install the *DMRP_QUERIES* project and enable prepackaged PS Queries:

1. Locate the *DMRP_QUERIES* Application Designer project on My Oracle Support and install it.
2. In PIA navigate to *QUERY_TREE_EW* using the following path: *PeopleTools, Security, Query Security, Query Access Manager*.
3. Enter *QUERY_TREE_EW* for the tree name and search.
4. Click the *QUERY_TREE_EW* link from the grid to access.
5. Click the Insert Child Group button.
6. Enter *DM_RED_PAPER_GROUP* for the access group and add.

7. Enter *Dimension Mapper Red Paper* for the description and click OK.
8. Select the *DM_RED_PAPER_GROUP - Dimension Mapper Red Paper* link and click the *Insert Child Record* button.
9. Enter *BUS_UNIT_SRC_PF* for the record and add.
10. Repeat the steps to insert the following records:
 - PF_BLUEPR_DFN
 - PF_BLUEPR_DTL
 - PF_BUS_UNIT_MAP
 - PF_SCR_TBL
 - PF_SETID_LOOKUP
 - PF_SRC_BU_NAMES
 - PF_SRC_BU_ROLES
 - PF_SRC_LINEAGE
 - PF_SRC_SETCNTRL
 - PF_SSCL_DFN
 - PF_SSCL_DTL
 - SRC_SYSTEM_TBL
11. Click Save.

Once the queries are installed you can access them as follows:

1. In PIA navigate to the queries using the following path: *Reporting Tools, Query, Query Viewer*.
2. Enter *DMRP* in the search box.
3. Click on the HTML link next to the query of your choice to view the results in a browser window.

Verifying the State of Output Tables (Optional)

The prepackaged PS Queries enable you to inspect the state of the output tables that Dimension Mapper populates. If you are populating a warehouse for the first time, the output tables are empty of data except for the set control tables *MODEL* and *SHARE*, which will contain SetID data.

If you are upgrading an existing warehouse, the tables are populated with data from all previously installed products. Existing warehouse business units can have an impact on the setup process if you map incoming business units to existing warehouse business units. The Business Unit Creation Wizard respects the SetID assignments of the existing warehouse business units and attempts to correlate the incoming SetIDs and the existing SetIDs on the appropriate record groups. If, for some reason, extraneous business units exist in the warehouses, unwanted SetID mappings may be created.

For example, if you manually create a business unit named US01 in the OWE, you must assign a SetID to US01 on all warehouse record groups. A default SetID is required at business unit creation time, although you can manually reassign the SetID for US01 on any warehouse record group by accessing the PeopleTools pages (should you wish).

If you bring in another source business unit (assume it is also named US01), and you map the two together, the system attempts to create mappings between the SetIDs used by US001 in the source, and the SetIDs used by US001 in the OWE. These mapping are then validated to ensure that no SetID conflicts exist. If the US001 entry in the warehouse is in fact erroneous (demo data, for example), then you may create unnecessary conflicts that will impact the setup process.

Likewise, if you bring in a SetID such as SHARE, and SHARE already exists in the warehouse, the Business Unit Creation Wizard prompts you to merge the two SetIDs. If you select yes, then any SetID assignments that SHARE itself might have in the source will be mapped to the existing SetID assignments that SHARE has in the warehouse. This "second-order" indirection is utilized by some applications to enable advanced prompting features. Consequently you should make sure that you do not have erroneous WBUs or SetIDs prior to running Dimension Mapper.

For these reasons, it is important to check the state of the output tables because existing business units and SetIDs impact how incoming business units and SetIDs from the source are mapped to their warehouse counterpart.

To inspect the state of the relevant input tables, you can use the following PS Queries:

<i>PS Query Object</i>	<i>Description</i>
DMRP_1_1_SET_CNTRL_REC	View the values for SET_CNTRL_REC.
DMRP_1_2_SET_CNTRL_GROUP	View the values for SET_CNTRL_GROUP.
DMRP_1_3_SET_CNTRL_TBL	View the values for SET_CNTRL_TBL.
DMRP_1_4_SETID_TBL	View the values for SETID_TBL.
DMRP_1_5_BUS_UNIT_TBL_PF	View the values for BUS_UNIT_TBL_PF.
DMRP_1_6_BUS_UNIT_TBL_FS	View the values for BUS_UNIT_TBL_FS.
DMRP_1_7_BUS_UNIT_MAP	View the values for BUS_UNIT_MAP.
DMRP_1_8_SETID_LOOKUP	View the values for SETID_LOOKUP.

Validating Business Unit and Set Control Data

Before you can begin creating warehouse business units or using the Business Unit Creation Wizard, you must run specific ETL job that setup the OWS, OWE, and common tables, and bring in your source business unit data. Use the PS Queries listed in the following sections to ensure that the appropriate setup jobs have been executed, the necessary data resides in your tables, and there is no corrupt data.

Source Business Unit Data

Run the queries listed below to ensure your source business unit data is present in EPM. All business units from each source should be present. If any are missing, check the ETL jobs and run again. If extraneous

business units are present (perhaps you imported the wrong data) then you will have to truncate the PF_SRC_BU_NAMES table manually.

PS Query Object	Description
DMRP_3_1_SRC_BU_NAMES	This query lists the source business units that are present in PF_SRC_BU_NAMES.
DMRP_3_2_SRC_BU_ROLES	<p>This query lists the roles present in PF_SRC_BU_ROLES.</p> <p>A role is associated with a business unit and relates to the type of business unit, such as accounts payable (AP), financial (FS), general ledger (GL), or inventory (INV). Each source BU can have one or more roles.</p> <hr/> <p>Note: All business units must have the FS role.</p>
DMRP_3_3_SETIDS	<p>This query identifies the source SetIDs. Any set control that is not present in the PF_SRC_BU_NAMES is interpreted as a SetID.</p> <hr/> <p>Note: If all of your SetIDs are also used as business units in the source, no rows display. This does not indicate a problem.</p>

Source Set Control Data

Run the query *DMRP_3_4_SRC_SETCNTRL* to ensure your source set control data is present in EPM. Set control information from all source systems should be present in PF_SRC_SETCNTRL and the query provides a count of the number of rows of set control information for each source that you have extracted into the system.

The number of rows returned for this query vary depending on the number of SetID based and set control tables in your source. You can determine how many rows should be present in the query results by multiplying the number of SetID-based tables in each source by the number of set controls that you defined in that source. The number of set controls for a given source is equal to the number of business units in that source plus the number of SetIDs in that source.

For example, if you have an FSCM source with 30 business units and one SetID, you should expect approximately 129,363 rows $[(30 \text{ BUs} + 1 \text{ SetID}) * (4,173) = 129,363]$. If you have more than one pillar then repeat this calculation for each pillar and add up the totals. Your estimate should come very close to the results of the query, with a 10-15 percent deviation at most. If the results are significantly different, check the ETL process for errors.

The following table is provided for you to estimate the number of rows in PF_SRC_SETCNTRL.

Source System	Number of SetID-Based Records		Number of Set Controls in each Source		Expected Number of Rows
CRM 9.0	1,350	*		=	

Source System	Number of SetID-Based Records		Number of Set Controls in each Source		Expected Number of Rows
ELM	98	*		=	
FSCM 9.0	4,173	*		=	
HCM 9.0	784	*		=	

Corrupt Source Data

Occasionally corrupt data can make its way into your source system (for example, someone enters data in the back-end rather than the using PIA pages, which control data quality). Corrupt data can affect your business units and set controls, and should not be allowed to enter EPM. As such, the following queries are provided to capture some business unit and set control error conditions:

PS Query Object	Description
DMRP_4_1_DANGLING_BU	<p>Identifies any occurrences of business units that are not found in the Source Set Control table.</p> <hr/> <p>Note: The Business Unit Creation Wizard does not create warehouse business units for source business units that do not appear in the Business Unit Names table. You must create these business units manually.</p>
DMRP_4_2_BAD_SETCNTRLS	<p>Identifies any business unit names that are greater than five character in length, null, or contain only a dash.</p> <p>These values must be removed or the validation step of the Business Unit Creation Wizard may hang indefinitely, causing the system to time-out.</p>

Verifying Source Blueprints

Source blueprints are vital in determining warehouse lineage and are delivered as system (SYS) data in EPM. Use the *DMRP_2_2_BLUEPR_DFN* query to confirm that the blueprints are present in the EPM system and populated with the correct data.

The table, PF_BLUEPR_DFN, should contain rows for all supported source systems. You should have the following number of rows for each source system:

- CRM90: 50
- FSCM90: 174
- HCM90: 69
- NONSETID: 1

Note: These numbers may change slightly due to updated bundle fixes.

Establishing Default Set IDs, TableSets, and Warehouse Lineage

Before you can create your warehouse business units, you must define SetIDs and warehouse lineage. This process identifies the source and warehouse tablesets, and the relationships (lineage) that exist between the two.

Pages Used to Define Default Set IDs, Warehouse Sources, and Lineage

<i>Page Name</i>	<i>Definition Name</i>	<i>Navigation</i>	<i>Usage</i>
TableSet ID	SETID_TABLE	PeopleTools, Utilities, Administration, TableSetIDs, TableSet ID	Create default SetIDs for each source that you wish to bring into EPM.
Source Blueprint	PF_BLUEPRINT	EPM Foundation, EPM Setup, Warehouse Sources and Bus. Units, View Source Blueprint, Source Blueprint	Review or update selected PeopleSoft source blueprints.
Warehouse Lineage	PF_WHOUSE_LINEAGE	EPM Foundation, EPM Setup, Warehouse Sources and Bus. Units, Warehouse Lineage	Associate a source blueprint to each PeopleSoft source to define the lineage between source tables and warehouse tables.

TableSet ID Page

Use the TableSet ID page (SETID_TABLE) to create default SetIDs for each source that you wish to bring into EPM.

Navigation

PeopleTools, Utilities, Administration, TableSetIDs, TableSet ID

Image: TableSet ID page

This example illustrates the fields and controls on the TableSet ID page.

TableSet ID

SetID:

Description:

Short Description:

Comments:

Use this page to define a default SetID for each source that you wish to bring into the warehouse. You should use a different default SetID for each source to keep tablesets from all sources separate—unless you have a good reason why data from different tablesets should merge.

This page is discussed in detail in the PeopleTools documentation.

See *PeopleSoft PeopleTools: PeopleSoft Application Designer Developer's Guide*

Source Blueprint Page

Use the Source Blueprint page (PF_BLUEPRINT) to review or update selected PeopleSoft source blueprints.

Navigation

EPM Foundation, EPM Setup, Warehouse Sources and Bus. Units, View Source Blueprint, Source Blueprint

Note: Before you access this page, run the ETL job that populates the PS_PF_SRC_SETCNTRL table. Otherwise, the page cannot display all the source blueprint details correctly.

Image: Source Blueprint page

This example illustrates the fields and controls on the Source Blueprint page.

Source Blueprint

Blueprint: HCM910

*Description: HCM 9.1 source for EPM

Blueprint Details					
	*Source Table	*Staging Table	*Warehouse Table		
1	ABS_CODE_TBL	ABS_CD_TBL_LANG	ABS_CODE_D00	+	-
2	ABS_CODE_TBL	ABS_CODE_TBL	ABS_CODE_D00	+	-
3	ABS_TYPE_TBL	ABS_CLASS_TBL	ABS_CLASS_D00	+	-
4	ABS_TYPE_TBL	ABS_CL_TBL_LANG	ABS_CLASS_D00	+	-
5	ABS_TYPE_TBL	ABS_TYPE_TBL	ABS_TYPE_D00	+	-
6	ABS_TYPE_TBL	ABS_TYPE_TBL	D_AB_TYPE	+	-
7	ABS_TYPE_TBL	ABS_TY_TBL_LANG	ABS_TYPE_D00	+	-
8	ACCT_TYP_TBL_SF	ACCT_TYP_TBL_SF	D_ACCT_TYPE	+	-
9	ACTN_REASON_TBL	S_ACTN_RSN_TBL	ACTN_REASON_TBL	+	-
10	BUS_CAL_DFN_TBL	L_BUS_CALDFNTBL	BUS_CAL_DFN_TBL	+	-
11	CAL_BPS_TBL	L_CAL_BPS_TBL	CAL_BPS_TBL	+	-
12	CAL_BPS_TBL	S_CAL_BPS_TBL	CAL_BPS_TBL	+	-
13	CAL_BPS_TBL	S_CAL_BPS_TBL	D_SUM_BUDGET	+	-
14	CAL_BP_TBL	L_CAL_BP_TBL	CAL_BP_TBL	+	-
15	CAL_BP_TBL	S_CAL_BP_TBL	CAL_BP_TBL	+	-

Source Table

Displays the source table associated with the selected PeopleSoft source.

If you have customized your ETL jobs or added rows or columns to your tables, you may need to modify this field.

Staging Table

Displays the OWS staging table associated with the preceding source table.

If you have customized your ETL jobs or added rows or columns to your tables, you may need to modify this field.

Warehouse Table

Displays the warehouse table (OWE or MDW) associated with the preceding OWS staging table.

If you have customized your ETL jobs or added rows or columns to your tables, you may need to modify this field.

Warehouse Lineage Page





Use the Warehouse Lineage page (PF_WHOUSE_LINEAGE) to associate a source blueprint to each PeopleSoft source to define the lineage between source tables and warehouse tables.

Navigation

EPM Foundation, EPM Setup, Warehouse Sources and Bus. Units, Warehouse Lineage

Image: Warehouse Lineage page

This example illustrates the fields and controls on the Warehouse Lineage page. You can find definitions for the fields and controls later on this page.

Warehouse Lineage			
Warehouse Sources			
Find First 1-5 of 5 Last			
Warehouse Source ID:	CRM	Description:	CRM Source Database
Source Type:	Enterprise		
Source Blueprint:	<input type="text"/> 		
Warehouse Source ID:	ELM	Description:	Enterprise Learning Mgmt Db
Source Type:	Enterprise		
Source Blueprint:	<input type="text"/> 		
Warehouse Source ID:	EPM91	Description:	EPM 9.1 Database
Source Type:	Enterprise		
Source Blueprint:	<input type="text"/> 		
Warehouse Source ID:	FSCM	Description:	FSCM Source Database
Source Type:	Enterprise		
Source Blueprint:	<input type="text"/> 		

Use the Source Blueprint field to associate a blueprint with each PeopleSoft source you are using with EPM. This process define the lineage between source and warehouse tables for each PeopleSoft source.

Note: The blueprints you select for the *Source Blueprint* field are compatible with all supported source releases, regardless of the release number associated with the blueprint.

Establishing Warehouse Business Units Using the Business Unit Creation Wizard

The Business Unit Creation Wizard automates the creation of warehouse business units. Prior to running the Business Unit Creation Wizard, you must run the ETL jobs that populate the input tables used by the wizard. These jobs are collectively described as dimension mapper ETL setup jobs.

Pages Used to Run the Business Unit Creation Wizard

Page Name	Definition Name	Navigation	Usage
Business Unit Creation Wizard - Start	PF_BU_WIZ_START	EPM Foundation, EPM Setup, Warehouse Sources and Bus. Units, Business Unit Wizard, Business Unit Creation Wizard	Access the Business Unit Creation Wizard and start the warehouse business unit creation process.
Business Unit Creation Wizard - Review Set Controls from Source	PF_BU_WIZ_STEP_1	Click Start on the Business Unit Creation Wizard - Start page.	Review the set controls from your PeopleSoft source systems.
Business Unit Creation Wizard - Address Set Control Collisions	PF_BU_WIZ_STEP_2	Click Next on the Business Unit Creation Wizard - Review Set Controls from Source page.	Review and resolve name collisions between and among incoming and existing warehouse set controls.
Business Unit Creation Wizard - Validate Mapping	PF_BU_WIZ_STEP_3B	Click Next on the Business Unit Creation Wizard - Address Set Control Collisions page.	Review source to warehouse set control mappings and correct any set controls that collide in the warehouse.

Business Unit Creation Wizard - Start Page

Use the Business Unit Creation Wizard - Start page (PF_BU_WIZ_START) to access the Business Unit Creation Wizard and start the warehouse business unit creation process.

Navigation

EPM Foundation, EPM Setup, Warehouse Sources and Bus, Units, Business Unit Wizard, Business Unit Creation Wizard

Business Unit Creation Wizard - Review Set Controls from Source Page

Use the Business Unit Creation Wizard - Review Set Controls from Source page (PF_BU_WIZ_STEP_1) to review the set controls from your PeopleSoft source systems.

Navigation

Click Start on the Business Unit Creation Wizard - Start page.

This page enables you to review the set controls from all of your sources. Set control includes both source SetIDs and source business units. Review this list to make sure that you see the business units and SetIDs

that you expect from each source. If business units are missing, you may not have run the setup ETL jobs properly. Check the ETL error logs and the Business Unit Wizard input tables mentioned above for potential problems. Assuming that you see the business units and SetIDs you expect, click Next to proceed to the Business Unit Creation Wizard - Address Set Control Collisions page.

Business Unit Creation Wizard - Address Set Control Collisions Page

Use the Business Unit Creation Wizard - Address Set Control Collisions page (PF_BU_WIZ_STEP_2) to review and resolve name collisions between and among incoming and existing warehouse set controls.

Navigation

Click Next on the Business Unit Creation Wizard - Review Set Controls from Source page.

This page enables you to review name collisions between and among incoming and existing warehouse set controls. You should resolve the name conflicts by renaming the incoming set control unless you have a good reason to merge them. If you are adding a new database to an existing warehouse installation, this page gives you the opportunity to inspect name collisions between new, incoming set controls and those already in the warehouse.

You can allow colliding set controls to merge with existing set controls. However, note that an existing set control has SetID assignments on warehouse record groups at the time of its creation, and these assignments may not be consistent with those of the incoming set control. This increases the chances for SetID conflicts. The Business Unit Creation Wizard can fix these problems by reassigning warehouse SetIDs, but if data already exists in the warehouse, then some dimensions may have to be reloaded. It is preferable to rename conflicting set controls to avoid these problems. Once you have renamed conflicting set controls, click Next to proceed to the Business Unit Creation Wizard - Validate Mapping page. It may take some time to proceed to the next page as the system processes all the inputs and validates the mapping configuration.

Business Unit Creation Wizard - Validate Mapping Page

Use the Business Unit Creation Wizard - Validate Mapping page (PF_BU_WIZ_STEP_3B) to review source to warehouse set control mappings and correct any set controls that collide in the warehouse.

Navigation

Click Next on the Business Unit Creation Wizard - Address Set Control Collisions page.

This page displays a report based on your source to warehouse set control mappings and the analysis of SetID assignments for all incoming set controls. More specifically, the report displays any instance of SetID splitting (one-to-many SetID mappings) and total numbers of SetIDs created and merged. If all of the set controls have been made unique as suggested, the potential for conflicts should be minimized.

If conflicts exist, you have two choices:

- Let the system correct the conflicts: Click the Accept Proposed SetIDs and Save button to automatically reassign the SetIDs used by warehouse business units on various warehouse record groups in order to make the configuration valid. The system will attempt to eliminate conflicts by reducing the number of warehouse SetIDs in order to eliminate one-to-many mappings.

- **Modify business unit mapping or SetID assignments manually:** You can analyze the report to where SetID splitting is occurring, and remove the conflicts by either remapping business units or choosing different SetID assignments for warehouse business units.

Regardless of how you fix the problem, if conflicts are detected, you must return to the business unit mapping page and validate the configuration again, to make sure that all conflicts have been removed. Continue to iterate the validation process until you see no more conflicts.

See [Defining Collision Mappings \(Optional\)](#).

Once any conflicts are resolved, the process of creating warehouse business units and assigning SetIDs is complete. You only need to inspect the properties of the new warehouse and general ledger business units.

See [Reviewing Warehouse and General Ledger Business Unit Creation](#).

Establishing Warehouse Business Units Manually

You can manually create warehouse business units by defining warehouse business units, assigning warehouse SetIDs to warehouse business units, mapping source business units to warehouse business units, and then validating those mappings.

Pages Used to Create Warehouse Business Units Manually

<i>Page Name</i>	<i>Definition Name</i>	<i>Navigation</i>	<i>Usage</i>
Warehouse Business Unit	BUS_UNIT_TBL_PF1	EPM Foundation, Business Metadata, Business Framework, Warehouse Business Units, Warehouse Business Unit	Define or modify a warehouse business unit and its default properties.
Business Unit Mapping	PF_BU_MAPPER	EPM Foundation, EPM Setup, Warehouse Sources and Bus. Units, Business Unit Mapping	Map source set controls to warehouse set controls.
Validate Mapping	PF_BU_MAP_REPORT	Click Validate on the Business Unit Mapping page.	Validate your source to warehouse set control mappings and correct potential set control collisions.

Warehouse Business Unit Page

Use the Warehouse Business Unit page (BUS_UNIT_TBL_PF1) to define or modify a warehouse business unit and its default properties.

Navigation

EPM Foundation, Business Metadata, Business Framework, Warehouse Business Units, Warehouse Business Unit

Image: Warehouse Business Unit page

This example illustrates the fields and controls on the Warehouse Business Unit page. You can find definitions for the fields and controls later on this page.

Warehouse Business Unit

Business Unit: CORP1

Business Unit

*Description: Corporation 1 *As of Date: 01/01/1900

*Short Desc: CORP1

*Base Currency: US Dollar Create GLBU with this ID

*Rate Type: Current Rate

Default Properties

Calendar ID: Monthly Calendar - 01

Holiday Calendar:

Business Unit Type

Consolidated

Non-Processing

Last Batch Number: 31

Base Currency

Select the base, or primary, currency for the selected warehouse business unit.

A warehouse business unit can have *only one* base currency.

This is usually the local currency for the organization, but accounting rules or other circumstances might require a different base currency.

Rate Type

Select a rate type for the selected warehouse business unit.

Create GLBU with this ID

Select this check box to create a general ledger business unit with the same ID as the selected warehouse business unit.

Last Batch Number

enter the last batch number that was assigned.

This number automatically increments as you run batches of transactions and you should not normally need to edit it. For example, you might want to enter a batch number when you install the system for the first time; however, you only need to reset it to reuse or skip batch numbers.

Default Properties

Calendar ID

Specify the default calendar type for the selected warehouse business unit.

You can choose *12*, *2*, *DR*, or *Monthly*.

The calendar ID you select appears as the default for the business unit on subsequent pages.

Holiday Calendar

Specify the default holiday calendar type for the selected warehouse business unit.

You specify a holiday calendar type only if you use one of the applications for the financial services industry (PeopleSoft Risk-Weighted Capital, Funds Transfer Pricing, or Asset Liability Management).

Business Unit Type

Consolidated

Select this check box to indicate that data for this warehouse business unit should be rolled up to higher level units in a business unit tree.

Non-Processing

Select this check box to create a warehouse business unit without stored set control values.

Business Unit Mapping Page

Use the Business Unit Mapping page (PF_BU_MAPPER) to map source set controls to warehouse set controls.

Navigation

EPM Foundation, EPM Setup, Warehouse Sources and Bus. Units, Business Unit Mapping

Source ID

Displays the source from which the source set control originates.

Source Set Control

Displays the source set control.

Map To

Indicates the direction of the set control mapping (source to warehouse).

Warehouse Set Control

Select a warehouse set control that you want to map to your source set control.

Validate

Click to validate your source to warehouse set control mappings and access the Validate Mapping page.

Validate Mapping Page

Use the Validate Mapping page (PF_BU_MAP_REPORT) to validate your source to warehouse set control mappings and correct potential set control collisions.

Navigation

Click Validate on the Business Unit Mapping page.

This page displays a report based on your source to warehouse set control mappings and the analysis of SetID assignments for all incoming set controls. More specifically, the report displays any instance of SetID splitting (one-to-many SetID mappings) and total numbers of SetIDs created and merged. If all of the set controls have been made unique as suggested, the potential for conflicts should be minimized.

If conflicts exist, you have two choices:

- Let the system correct the conflicts: Click the **Accept Proposed SetIDs** button to automatically reassign the SetIDs used by warehouse business units on various warehouse record groups in order to make the configuration valid. The system will attempt to eliminate conflicts by reducing the number of warehouse SetIDs in order to eliminate one-to-many mappings.
- Modify business unit mapping or SetID assignments manually: You can analyze the report to where SetID splitting is occurring, and remove the conflicts by either remapping business units or choosing different SetID assignments for warehouse business units. Click the **Return to Business Unit Mapper** button to return to the Business Unit Mapping page.

Regardless of how you fix the problem, if conflicts are detected, you must return to the Business Unit Mapping page and validate the configuration again, to make sure that all conflicts have been removed. Continue to iterate the validation process until you see no more conflicts.

See [Defining Collision Mappings \(Optional\)](#).

Once any conflicts are resolved, the process of creating warehouse business units and assigning SetIDs is complete. You only need to inspect the properties of the new warehouse and general ledger business units.

See [Reviewing Warehouse and General Ledger Business Unit Creation](#).

Reviewing Warehouse and General Ledger Business Unit Creation

After you complete your warehouse business unit creation, you should review your warehouse business units and general ledger business units to ensure certain properties (such as base currency) meet your requirements.

If you created your warehouse business units automatically using the Business Unit Creation Wizard, some of your warehouse business units may have been created with a base currency or rate type that differs from the defaults that you defined for your source. If this is the case, you need to change these settings for the appropriate business units. In addition, the wizard does not associate calendars to business units. You must do this manually for all your warehouse business units.

Regardless of whether you used the Business Unit Creation Wizard to create your warehouse business units or did so manually, you should verify the properties of your general ledger business units (GLBU). PeopleSoft general ledger business units are extracted from your source system and populated in the EPM database using ETL jobs.

Pages Used to Review Your Warehouse and General Ledger Business Units

<i>Page Name</i>	<i>Definition Name</i>	<i>Navigation</i>	<i>Usage</i>
Warehouse Business Unit	BUS_UNIT_TBL_PF1	EPM Foundation, Business Metadata, Business Framework, Warehouse Business Units, Warehouse Business Unit	Review and modify warehouse business units and the default properties.
General Ledger	BUS_UNIT_TBL_GL1	EPM Foundation, Business Metadata, Business Framework, General Ledger Business Units, General Ledger	Review and modify general ledger business units and the default properties.

Warehouse Business Unit Page

Use the Warehouse Business Unit page (BUS_UNIT_TBL_PF1) to review and modify warehouse business units and the default properties.

Navigation

EPM Foundation, Business Metadata, Business Framework, Warehouse Business Units, Warehouse Business Unit

If you created your warehouse business units automatically using the Business Unit Creation Wizard, review the Base Currency and Rate Type properties to ensure that they match those defined in your source system. Also, define the calendar properties for the warehouse business unit.

General Ledger Page

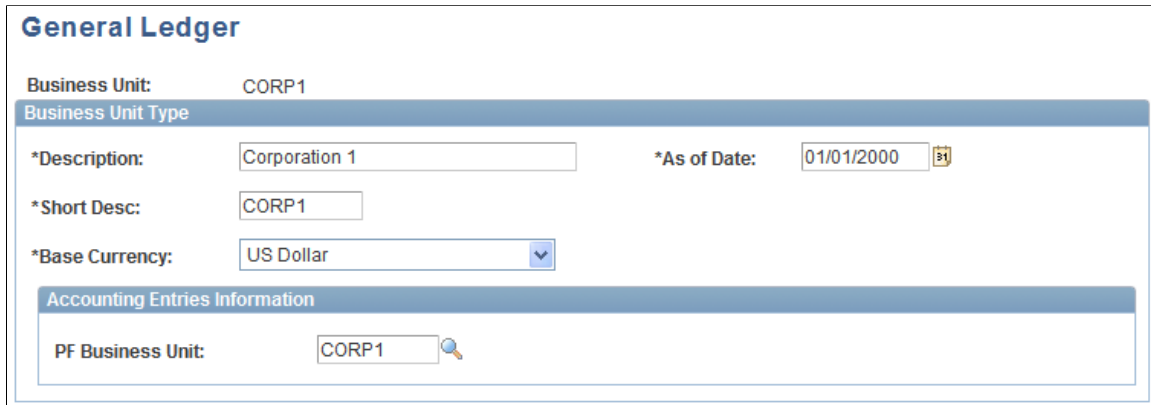
Use the General Ledger page (BUS_UNIT_TBL_GL1) to review and modify general ledger business units and the default properties.

Navigation

EPM Foundation, Business Metadata, Business Framework, General Ledger Business Units, General Ledger

Image: General Ledger page

This example illustrates the fields and controls on the General Ledger page. You can find definitions for the fields and controls later on this page.



General Ledger

Business Unit: CORP1

Business Unit Type

*Description: Corporation 1 *As of Date: 01/01/2000

*Short Desc: CORP1

*Base Currency: US Dollar

Accounting Entries Information

PF Business Unit: CORP1

Review the Base Currency and PF Business Unit properties to ensure that they match the properties defined in your source system.

Defining Collision Mappings (Optional)

When you validate your source to warehouse set control mappings, you can choose to allow two tablesets merge in the warehouse. However, if two tablesets merge it is possible for collisions to occur between business keys (such as supplier ID). If such collisions do occur, you must create a collision map to resolve the collision.

Pages Used to Define Collision Maps

Page Name	Definition Name	Navigation	Usage
Collision Map Definition	PF_COLLISION_MAP	EPM Foundation, EPM Setup, ETL Map Parameters, Collision Map Definition	Define collision maps to resolve business ID collisions.
Collision Map Error Report	PF_COLLISION_RPT	EPM Foundation, EPM Setup, ETL Map Parameters, Collision Map Error Reports, Collision Map Error Report	Displays the results of the error out duplicates collision map.

Collision Map Definition Page

Use the Collision Map Definition page (PF_COLLISION_MAP) to define collision maps to resolve business ID collisions.

Navigation

EPM Foundation, EPM Setup, ETL Map Parameters, Collision Map Definition

Image: Collision Map Definition page

This example illustrates the fields and controls on the Collision Map Definition page. You can find definitions for the fields and controls later on this page.

The screenshot shows a web form titled "Dimension Map Definition" with the following fields and values:

- Dimension Map Id:** FIRST_IN
- *Description:** First in Wins
- *Map Type:** First In Wins (dropdown menu)
- Map Category:** Resolution Method
- Dimension Name:** ACCOUNT
- *Mapping Table:** PF_DIMN_MAP
- Notes:** (empty text area)

Map Type

Select a collision map type to resolve any business ID conflict resulting from merging tablesets.

You can select:

First In Wins: The first instance of a business ID is loaded into a tableset and subsequent instances of the same ID from different tablesets are ignored.

Error Out Duplicates: The first instance of a business ID is loaded into a tableset, but subsequent instances of the same ID from different tablesets are sent to an error table where they can be inspected and remapped. You can examine these errors by accessing the Collision Map Error Reports page.

If you select this option, the Warehouse Record and Dimension ID fields appear for editing.

Use Mapping Table: Each incoming business ID is checked against a mapping table. If an entry does not exist in the mapping table, the row errors out.

Dimension Name

Select the target dimension that contains the merging tablesets.

The list is drawn from the set of dimensions defined in the Define Dimension page.

Mapping Table

Select a mapping table that stores resolved conflicts to verify future incoming business IDs.

You can use the default mapping table (PF_DIMN_MAP) or create your own mapping table in Application Designer.

Warehouse Record

Select the OWE or MDW table that associates with the dimension selected for the Dimension Name field.

This table is used for prompting in the Error-Out Duplicates map type.

Dimension ID Field

Select the ID column for the table you specified in the Warehouse Record field.

Description Fieldname

Select the column that stores the description for the table you specified in the Warehouse Record field.

Collision Map Error Report Page

Use the Collision Map Error Report page (PF_COLLISION_RPT) to displays the results of the error out duplicates collision map.

Navigation

EPM Foundation, EPM Setup, ETL Map Parameters, Collision Map Error Reports, Collision Map Error Report

This page displays the results of the error out duplicates collision map, where the first instance of a business ID is loaded into the relevant tableset and the subsequent instances of the same ID from different tablesets are sent to this page for review. For each row of data you can chose to:

- *Accept*: The mapping between the colliding business IDs is preserved as displayed.
- *Create*: You can create a new business ID that can be loaded into the applicable tableset.
- *Defer*: You can defer action to a later time.
- *Select*: You can select an existing member of the tableset as a mapping for this incoming business ID.

If you select *Create* or *Select*, the Target Business ID field becomes available for editing and you can create or select another business ID to load into the applicable tableset.

ETL Configurations

Using the Master Sequencer Utility to Create Master Sequencer Jobs

The master sequencer utility (`Create_MasterSequence`) enables you to create master sequencer jobs using delivered sequencer jobs as input. You will recall that sequencer jobs invoke and run other jobs. You can create a master sequencer job to invoke and run all the required sequencers that populate a specific fact table, or populate all fact tables for a given data mart, or all fact tables for an entire EPM warehouse.

For example, you can create a master sequencer job to automatically invoke and run 20 sequencer jobs from the Receivables mart. To do so, simply use the master sequencer utility to specify the 20 jobs you want to invoke, enter the applicable job parameters, and import the generated *.dsx file into your project.

The master sequencer utility is available in the following DataStage folder: `Utilities\Job_Utils\MasterSequence\Create_MasterSequence`.

Creating a Master Sequencer Job

To create a master sequencer job using the master sequencer utility:

1. Identify the individual sequencer jobs you want to include in the master sequencer job.

The ETL Lineage topic can help you determine which ETL sequencer jobs are required based on your needs. See [Understanding the EPM Lineage Spreadsheets](#).

Note: The master sequencer utility uses only sequencer jobs as input; please do not use server jobs as input. Also, the sequencer jobs you select should be compiled and ready for use.

2. Copy the list of sequencer jobs and paste the names into a text file.

Ensure that each job is entered on a separate line. For example,

```
SEQ_J_Dim_PS_D_DT_PATTERN_OWE  
SEQ_J_Dim_PS_D_BUS_UNIT_PF_OWE  
SEQ_J_Dim_PS_D_BUSINESS_UNIT_EX
```

Note: The master sequencer job uses the text file as an input and reads the list of sequencer jobs from the file.

3. Save the text file and specify the input file path (`SRC_JOBPATH`).

The file path you specify should be local to the server; do not specify a path to a client machine.

Note the file path of the text file for later use.

4. In DataStage Director, navigate to the Master Sequencer Utility using the following navigation: `Utilities\Job_Utils\MasterSequence\Create_MasterSequence`.

5. Select *Create_MasterSequence* and run.

The *Create_MasterSequence - Job Run Options* window appears where the job input parameters are displayed.

6. In the Parameters tab, use the *Enter Source text file Name* field to enter the file path for the text file you created in step three.
7. Use the *Enter Target File Path* field to enter a location where you want the new master sequencer job (the output *.dsx file) stored.

The file path you specify should be local to the server; do not specify a path to a client machine.

The file path must have the *.dsx extension appended to it, for example, *C:\MSEQ\Mseq01.dsx*.

8. Use the *Enter the Jobname* field to enter the name of the new master sequencer job.
9. Use the *Choose the Master Sequence Type* field to specify the mode in which you want the sequencer jobs to be run.

Valid values are *Parallel* or *Sequential*.

Note: Ensure you select the proper mode for the sequential jobs included in the master sequencer. Only staging sequencer jobs can be run in parallel. Other dimension and fact jobs may have dependencies and, unless those dependencies are identified and managed accordingly, we advise that you do not run them in parallel.

10. Click the Run button to generate the master sequencer job.

This process generates the master sequencer job and stores it in the target file path you specified in step seven.

11. Check the log file of the *Create_MasterSequence* job for warning messages

Hard-coded values in the sequencer job parameters are listed as warning messages in the log file. If you encounter a warning message, correct the master sequencer job parameters accordingly, and re-run the *Create_MasterSequence* job.

12. Import the generated master sequencer job into your DataStage project.

If a master sequencer job with the same name already exists in the project, it is overwritten and placed in the same category.

If the master sequencer job does not exist in the project, it is placed in the following default location: *Utilities\Job_Utils\MasterSequence*.

Once the master sequencer job is imported into your project, you can move it to another category.

The utility does not provide annotations for the master sequencer job, but you can provide annotations if you wish.

Note: You must compile the master sequencer job before you can use it.

See [Compiling ETL Jobs](#).

Using the Master Run Utility to Automatically Run Your ETL Jobs

This topic provides an overview of the Master Run Utility and discusses how to run the utility.

Understanding the Master Run Utility

PeopleSoft provides the Master Run Utility, which enables you to automatically run a set of jobs located in a flat file on the DataStage Server. When you use the Master Run Utility, it reads a list of jobs that are present in a specified flat file and triggers the jobs to run in serial mode, using the dependency logic specified in the Input flat file.

The following features are incorporated into the Master Run Utility:

- Run jobs from a specified flat file.
- Reset jobs and rerun when they are in an aborted stage.
- Active restart recovery.
- Run jobs in dependent or independent modes.
- Generate a job status report in the log file at the end of a run.

Location of the Master Run Utility

The Master Run Utility is available in the *Common_Utilities.dsx* file. After you import this .dsx file, you can locate the utility using the following navigation: *Jobs, Utilities, Job_Utils, Master_Run_UTILITY*.

Input File for Master Run Utility

The Input flat file contains the list of ETL jobs that are available in the DataStage Server and indicates whether a job is *independent* or *dependent*. Dependency information is used to determine job execution in the Master Run Utility. If a job labeled as independent fails to run properly, the Master Run Utility logs the information and proceeds to the next job. However, if a job labeled as dependent fails to run properly, the utility itself aborts. All jobs listed in the flat file contain the suffix *N* (Independent job) or *D* (Dependent Job) to indicate their dependency.

The Input flat file also contains comments that describe each set of ETL jobs. Comments are prefixed with an asterisk (*) to clearly indicate they are comments. The following screen shot provides an example flat file:

Image: Sample Flat File (HCM_OWS_E_Base_HashFiles.txt)

This example illustrates the Sample Flat File (HCM_OWS_E_Base_HashFiles.txt).

```

Sample_HCM_OWS_E_Base_HashFiles.txt - Notepad
File Edit Format View Help
*****
*** HCM - Load OWS Hash Files ***

J_HASH_CRC_PS_COMPETENCY_TBL_HCM91_EPM91, N
J_HASH_CRC_PS_INCIDENT_DATA_HCM91_EPM91, N
J_HASH_CRC_PS_MAJOR_TBL_HCM91_EPM91, N
J_HASH_CRC_PS_RATING_MDL_TBL_HCM91_EPM91, N
J_HASH_PS_ABS_CLASS_TBL_HCM91_EPM91, N
J_HASH_PS_ABS_CODE_TBL_HCM91_EPM91, N
J_HASH_PS_ABS_TYPE_TBL_HCM91_EPM91, N
J_HASH_PS_ABSENCE_HIST_HCM91_EPM91, N
J_HASH_PS_ABSV_ACCRUAL_HCM91_EPM91, N
J_HASH_PS_ABSV_PLAN_TBL_HCM91_EPM91, N
J_HASH_PS_ABSV_REQUEST_HCM91_EPM91, N
J_HASH_PS_ACCOMP_TBL_HCM91_EPM91, N
J_HASH_PS_ACCOMPLISHMENTS_HCM91_EPM91, N
J_HASH_PS_ACCT_CD_TBL_HCM91_EPM91, N
J_HASH_PS_ACCT_CD_TBL_GL_HCM91_EPM91, N
J_HASH_PS_ACT_RSN_TBL_AUS_HCM91_EPM91, N
J_HASH_PS_ACT_RSN_TBL_ESP_HCM91_EPM91, N
J_HASH_PS_ACT_RSN_TBL_GER_HCM91_EPM91, N
J_HASH_PS_ACT_RSN_TBL_MEX_HCM91_EPM91, N
J_HASH_PS_ACT_RSN_TBL_NA_HCM91_EPM91, N
J_HASH_PS_ACTION_TBL_HCM91_EPM91, N
J_HASH_PS_ADDRESSES_HCM91_EPM91, N
J_HASH_PS_BEN_DEFN_PGM_HCM91_EPM91, N
J_HASH_PS_BN_PERSON_HCM91_EPM91, N
J_HASH_PS_BUDGET_BUS_UNIT_HCM91_EPM91, N
J_HASH_PS_BUDGET_PERIOD_HCM91_EPM91, N
J_HASH_PS_CAR_PLAN_HCM91_EPM91, N
J_HASH_PS_CAREER_STRENGTH_HCM91_EPM91, N
J_HASH_PS_CITIZENSHIP_HCM91_EPM91, N
J_HASH_PS_CM_CLUSTER_TBL_HCM91_EPM91, N
J_HASH_PS_CM_EVALUATIONS_HCM91_EPM91, N
J_HASH_PS_CM_ROLE_HCM91_EPM91, N

```

The sample file above contains the list of HCM jobs used to load data into OWS target tables.

Restart Recovery Feature

When a dependent job fails to run properly, the Master Run Utility automatically aborts the job and all subsequent jobs. Assuming you fix the problem that caused the job to fail, you must rerun the Master Run Utility to complete the process of running those jobs that were aborted. Instead of running all the jobs in that particular folder, the restart recovery feature enables you to rerun only those jobs that were aborted.

For example, assume the file `SAMPLE_HCM_E_GLOBAL_DIMENSIONS_Base_Tables.txt` contains Global Dimension jobs that are all dependent to each other. If the `SEQ_J_Dim_PS_D_POS` job is aborted, the entire utility aborts. Later you fix the issue in the `SEQ_J_Dim_PS_D_POS` job. This time, you can run the `Master_Run_UTILITY` with the `Restart Recovery` option set to *Yes* so that it runs the jobs from `SEQ_J_Dim_PS_D_POS` only and ignores the jobs that have previously completed successfully.

Running the Master Run Utility

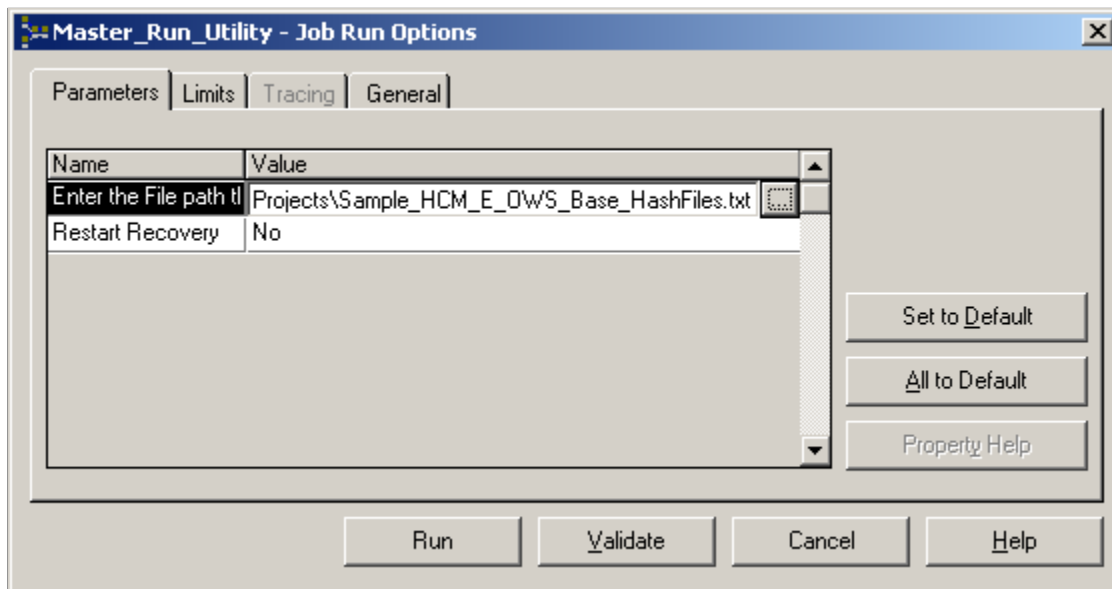
Perform the following steps to run the Master Run Utility:

1. In DataStage Director, navigate to the Master Run Utility using the following navigation: *Jobs, Utilities, Job_Utills*.
2. Select *Master_Run_UTILITY* and click Run.

The *Master_Run_UTILITY - Job Run Options* window appears.

Image: Master_Run_UTILITY - Job Run Options Window

This example illustrates the *Master_Run_UTILITY - Job Run Options* Window.



3. Enter the path to the location to the flat file and specify whether you want to use the restart recovery feature.

Click Run.

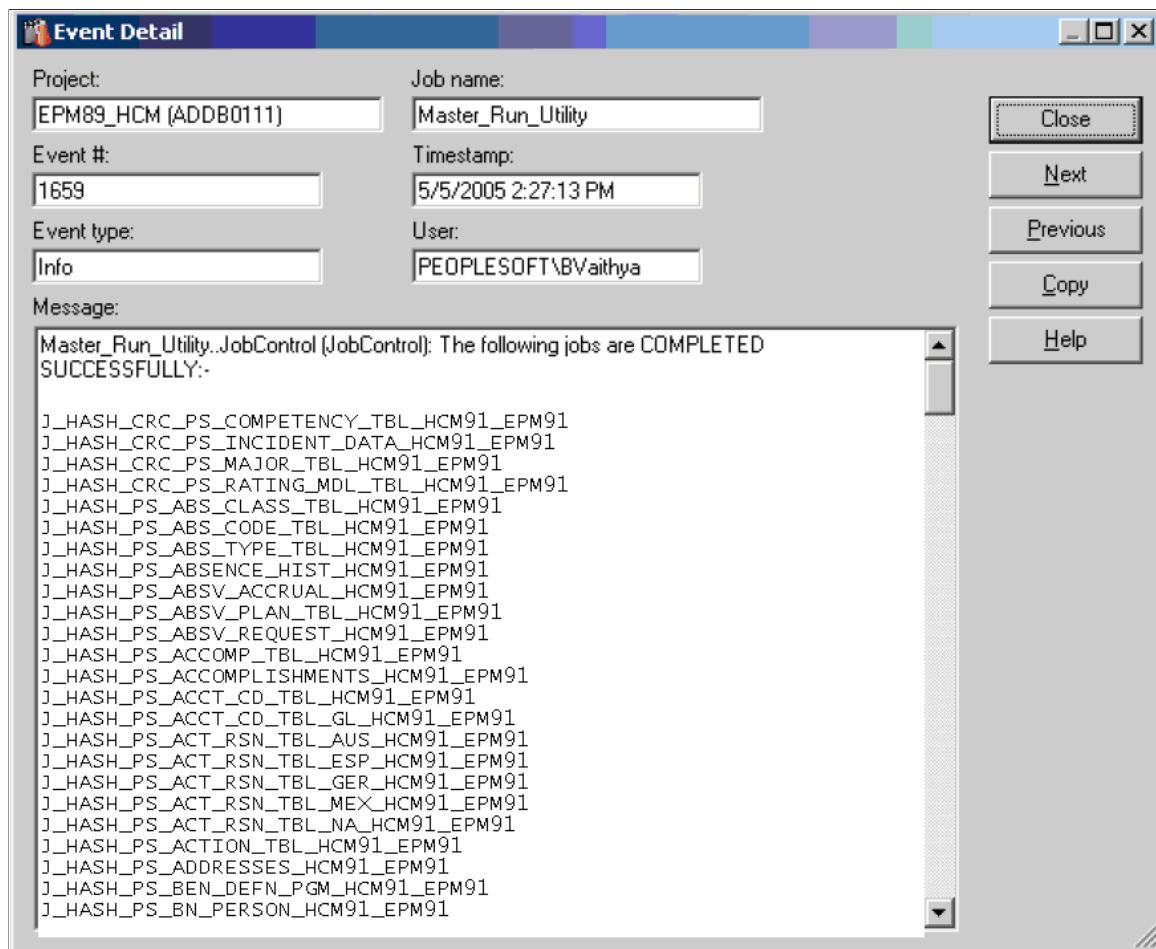
4. From the DataStage Director menu, select *View, Log*.

The Master Run Utility generates a report with the following information:

- Jobs that have completed successfully.
 - Jobs that have completed with warnings.
 - Jobs that have aborted
 - Jobs that have not compiled
 - Jobs that have incorrect names.
5. If you double-click log entries that contain the phrase *COMPLETED SUCCESSFULLY*, the Event Details window appears and displays all the jobs that have successfully completed.

Image: Event Details Window

This example illustrates the Event Details Window.



Likewise, if you double-click log entries that contain the key word *ABORTED*, the Event Details window appears and displays all the jobs that have aborted.

Supporting UniCode Data

To support Unicode data, you must have first installed DataStage Server with the *Install NLS for DataStage Server* check box selected. You must then configure NLS options in DataStage Administrator.

To configure NLS options:

1. In DataStage Administrator, select the Projects tab and then the project you want to configure NLS options.
2. Select the NLS button to select NLS options.

These options are available only if DataStage Server was installed with the NLS option.

3. In the Server Maps tab, select the correct NLS map for the project default.

This value is used in all the jobs to map Unicode data.

4. If you do not see the correct NLS map in the list, you can click the Install>> button to view all available maps and load the particular map you require.
5. Select the NLS tab to ensure that your selected map displays as the Project Default for individual jobs in the DRS stage.

Note: This value can be overwritten at the job level by changing this value in the DRS stage.

6. Click OK to return to the Projects tab.

Converting Incremental Load Jobs to Destructive Load Jobs

As part of the ETL configuration process, you can convert incremental load jobs to destructive load jobs. However, because server jobs that use CRC logic would require modification to at least 80% of the design, it is better not to alter the existing jobs and create a new destructive load job from scratch.

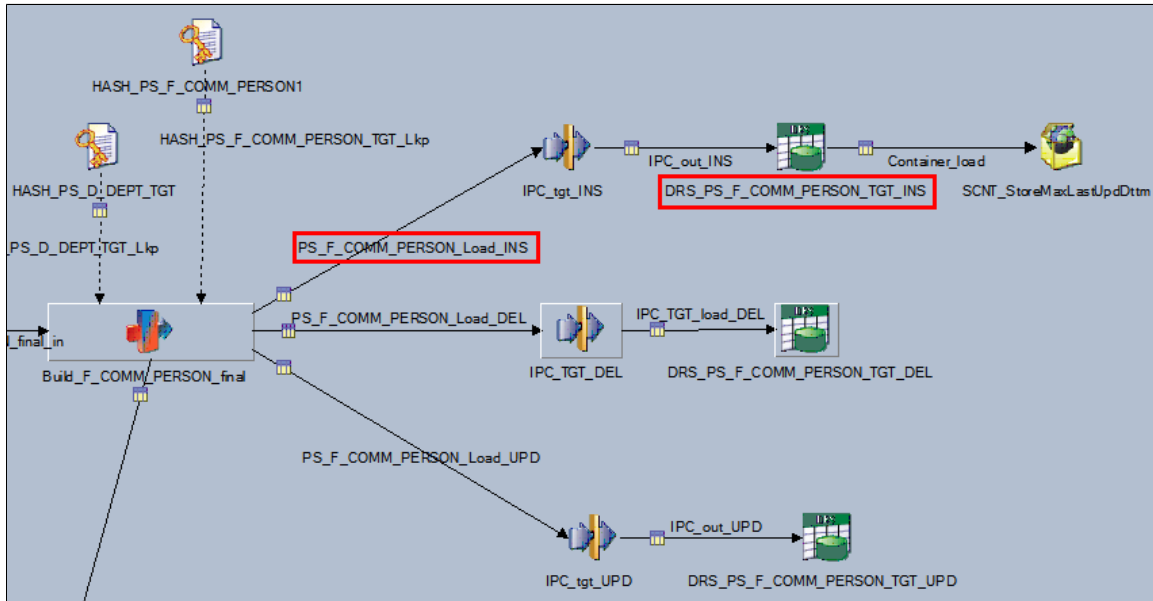
This topic discusses how to convert incremental load jobs that use the DateTime stamp.

Converting Jobs that Use the DateTime Stamp

The changes required to convert an incremental load job (that uses the DateTime stamp) to a destructive load job can be demonstrated using the J_Fact_PS_F_COMM_PERSON job as an example.

Image: Example of the incremental load in the J_Fact_PS_F_COMM_PERSON job

This example illustrates the Example of the incremental load in the J_Fact_PS_F_COMM_PERSON job.



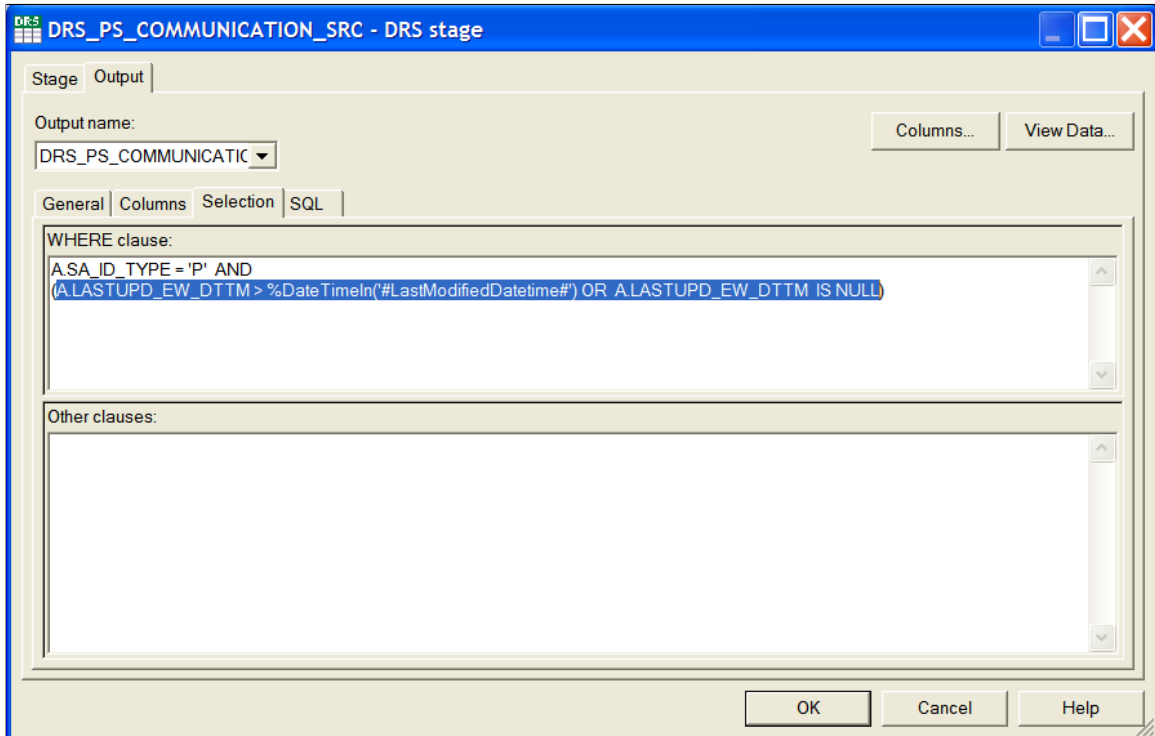
To convert an incremental load job (that uses the DateTime stamp) to a destructive load job:

1. In DataStage Designer, open the server job you want to convert.
2. Open the source DRS stage and select the Output tab.

3. In the Selection sub-tab, locate the WHERE clause and delete the last update date time portion (highlighted below).

Image: Deleting the WHERE clause

This example illustrates Deleting the WHERE clause.

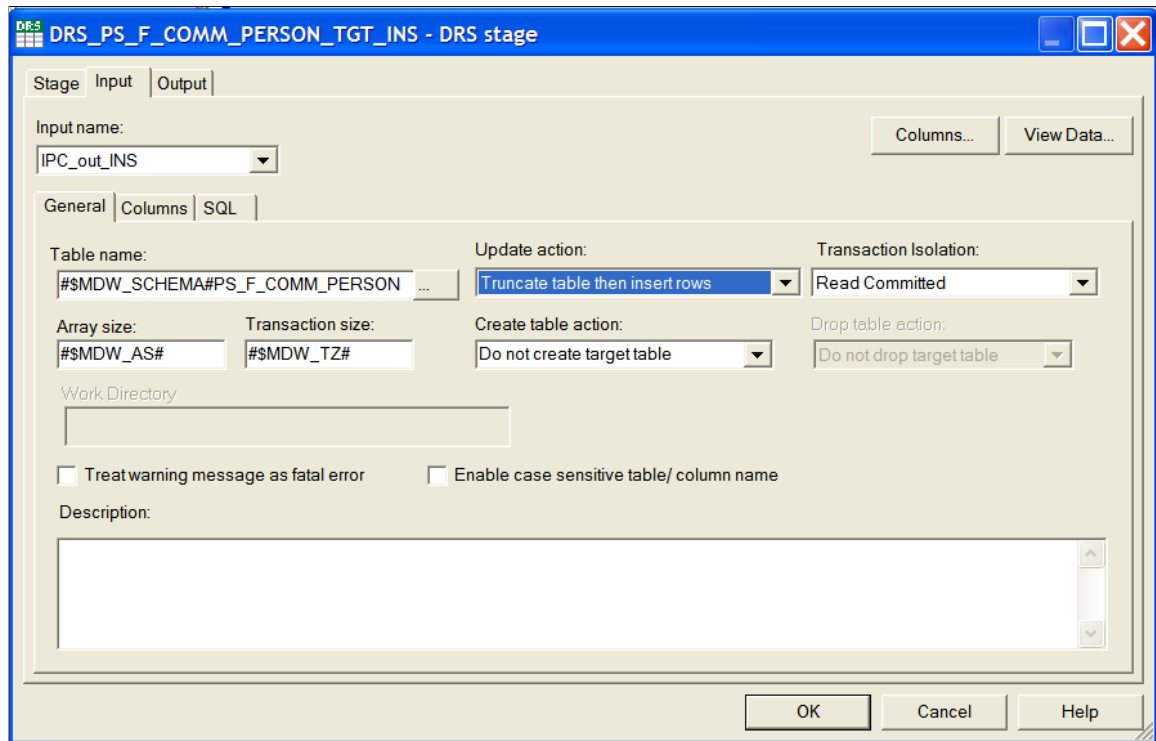


4. Click OK to save your changes.
5. Open the insert (*_INS) target DRS stage and select the Input tab.

- In the General sub-tab, select *Truncate table then insert rows* for the Update Action field.

Image: Update Action field

This example illustrates the Update Action field.

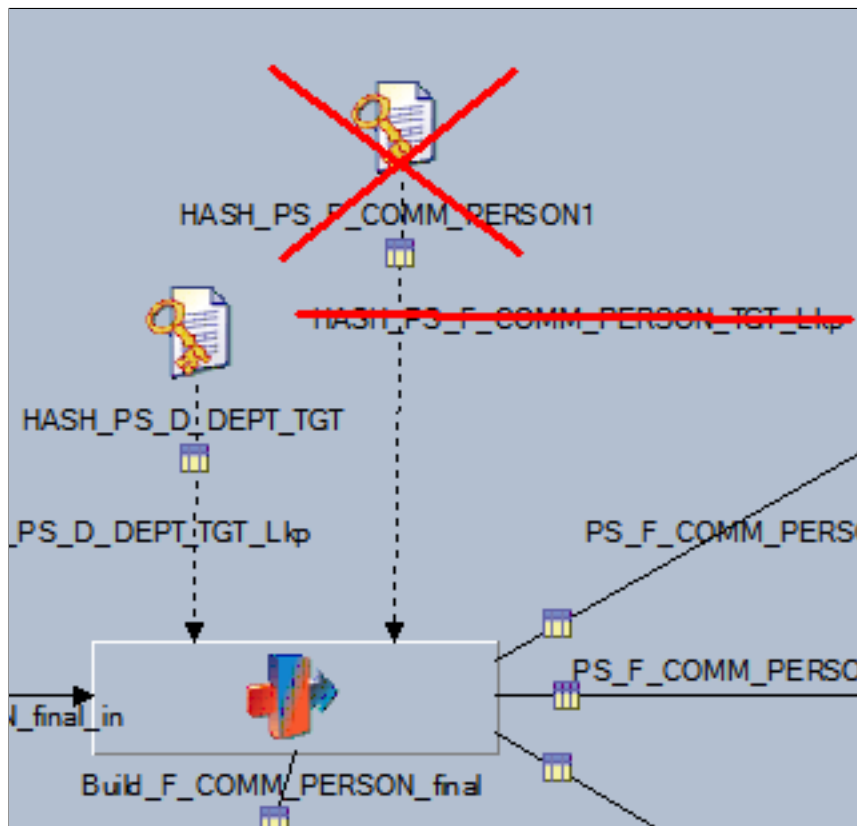


- Click OK to save your changes.
- Delete the StoreMaxLastUpdDttm container and link.
- Delete the delete (*_DEL) target DRS stage and link.
- Delete the update (*_UPD) target DRS stage and link.
- Delete the hash target table lookup (the hash lookup that is performed against target table data) and link.

Because this hash load is used to identify updated or new records and you are converting the job to destructive load, the hash load is no longer needed.

Image: Deleting the hash target table lookup

This example illustrates Deleting the hash target table lookup.



12. Open the last transformation stage in the job (it should immediately precede the insert target DRS stage).

New rows are identified in this stage and this is done to retain the `Created_EW_DTTM` of rows.

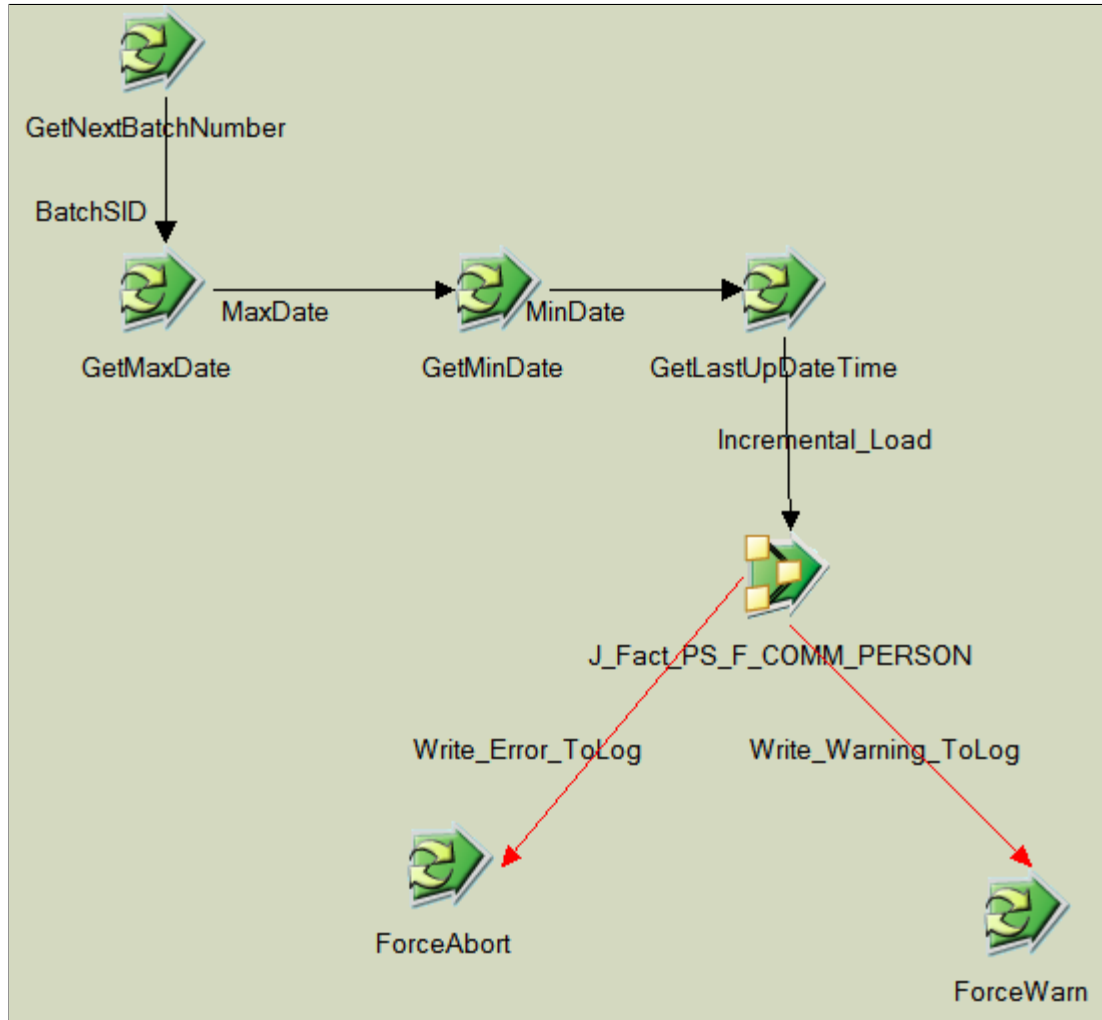
In the example job above, the last stage is called `Build_F_COMM_PERSON_final`.

13. Delete the *InsertFlag* stage variable and click OK to save and exit the window.
14. Select Edit, Job Properties from the menu and change the target column value for `CREATED_EW_DTTM` to *DSJobStartTimeStamp*, which is a DS Macro (and same as for the field `LASTUPD_EW_DTTM`).
15. Delete the *LastModifiedDateTime* job parameter and click OK to save and exit the window.

- Open the corresponding sequence job that calls the server job and delete the *GetLastUpDatetime* job activity stage (which calls the routine of the same name).

Image: Sequencer job

This example illustrates the Sequencer job.



- Select Edit, Job Properties from the menu and delete the *LastUpdDate Time* job parameter if it is present.

This parameter is not present in every job.

(I say "IF" because it may not be present in the sequence job. It is not needed in the sequencer anyway).

- Change the job annotations and descriptions to reflect the change.
- Save changes and exit.
- Save and recompile the job.

Adding New Environmental Variables

Environmental variables are project level parameters which are typically used across projects. The advantage to environmental variables is that they can be set at the project level and all associated jobs in that project will automatically use the value.

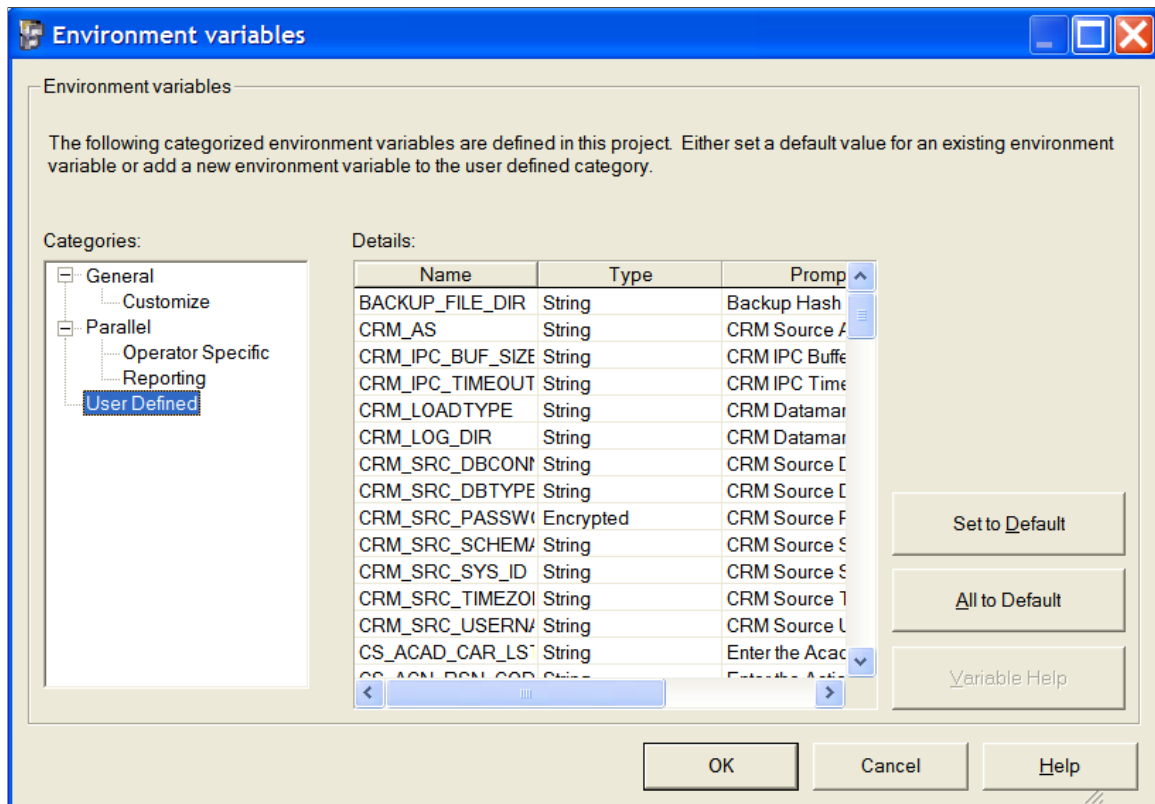
Creating a New Environmental Variable

To add a new environmental variable:

1. In DataStage Administrator, select the Projects tab.
2. Click the Properties button and select the General tab.
3. Click the Environment button and select User Defined from the Categories pane.

Image: Defining Environmental Variables

This example illustrates the Defining Environmental Variables.



4. Enter your new environmental variable parameters for the following fields:
 - a. *Name*
 - b. *Type*
 - c. *Prompt*

d. *Value*

5. Click OK to save your changes.

Adding an Environmental Variable to a Server Job

To add the new environmental variable to a server job:

1. In DataStage Designer, select Jobs from the project tree.
2. Select the job you want to add the environmental variable.
3. Select Edit, Job Properties from the menu and select the Parameters tab.
4. Click the Add Environment Variable... button.

The Choose Environment Variable window appears.

5. Select the new environmental variable from the Choose Environment Variable window.
6. Click OK to save your changes.

The new environmental parameter is now a part of the job parameters.

The Default value can be changed to \$PROJDEF in the job parameters to signify that the value shall be taken from the project default value unless overwritten.

7. Save and recompile the job.

Updating the Related Sequencer Job

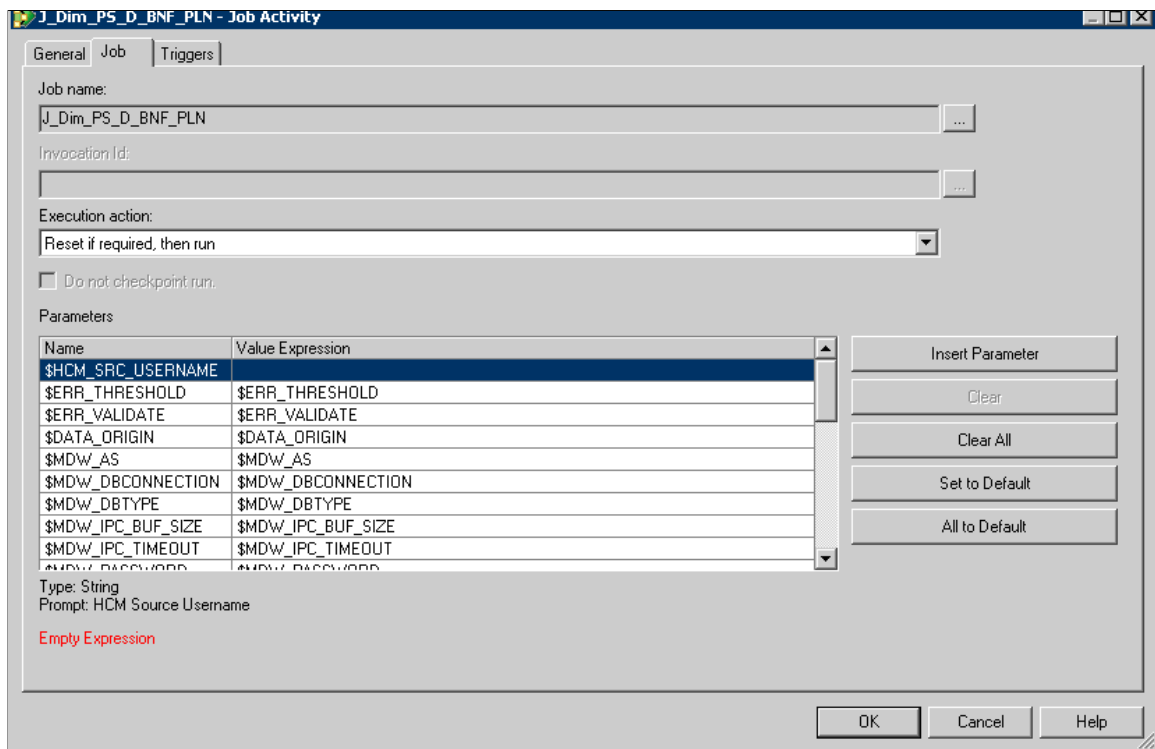
If there is a sequencer job that calls the server job you modified, you must update the sequencer job to add the parameter value to the job activity stage where the value is passed to the called job.

To modify the related sequencer job:

1. Select Edit, Job Properties from the menu and select the Job tab.

Image: Job Tab of Job Properties

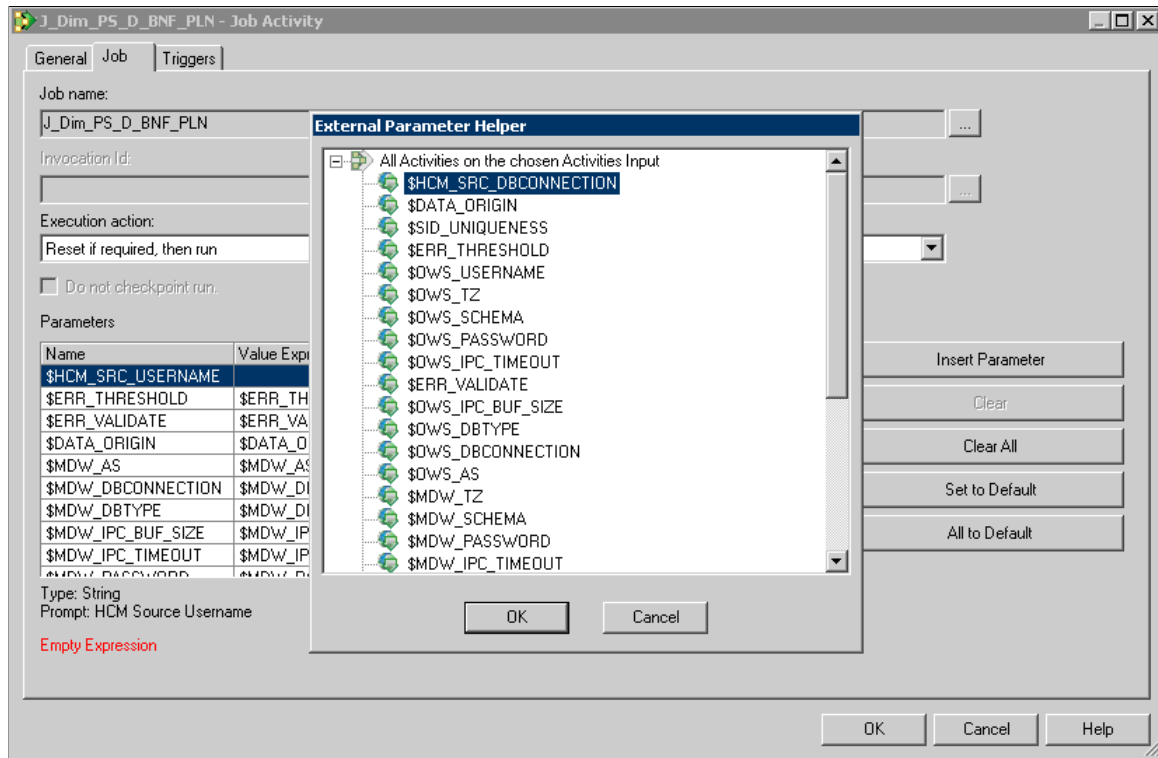
This example illustrates the Job Tab of Job Properties.



2. Select the parameter from the list, and click Insert Parameter.

Image: Inserting Your New Parameter

This example illustrates Inserting Your New Parameter.



3. Select your parameter, then click OK to save changes and exit.
4. Save and recompile the job.

Setting Up and Working with Metadata for the Operational Warehouse - Enriched

Understanding Metadata

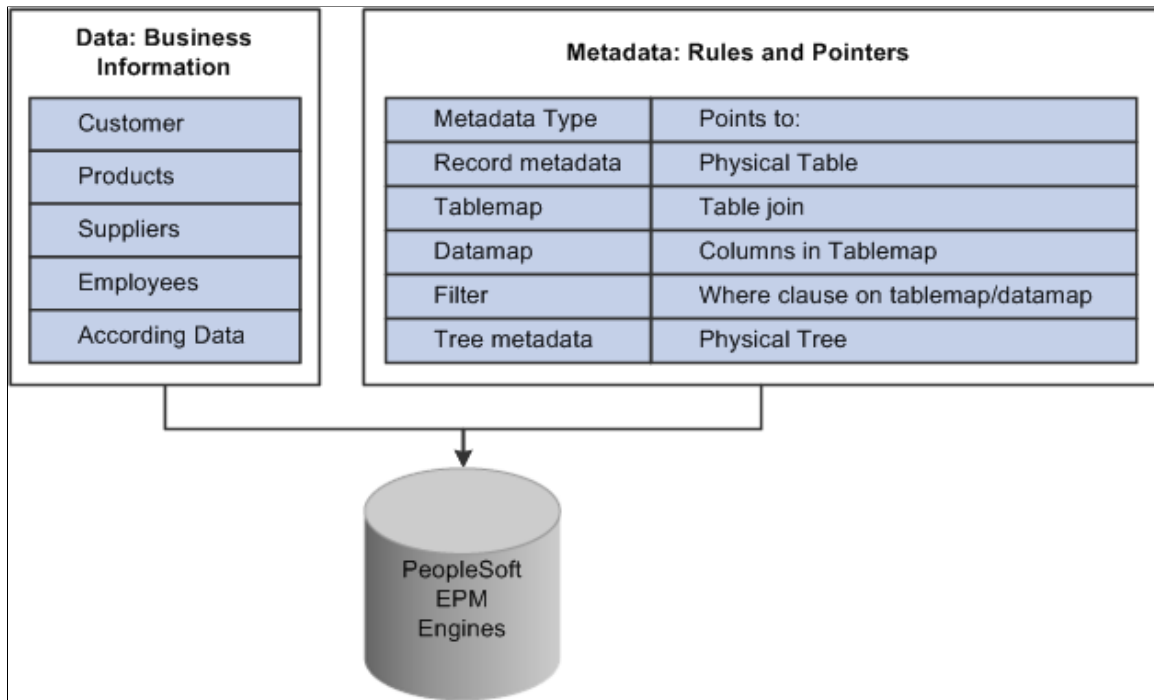
The PeopleSoft Analytical Applications are supported by a framework of metadata. Metadata defines everything from table and data structures to rules for running processes. Metadata is central to the entire Analytical Applications product suite.

When you run PeopleSoft EPM processes, you require two types of input: your business data and the appropriate EPM metadata. Metadata provides an abstraction layer that enables technical users to establish dynamic relationships between tables, business users to easily identify the data that interests them without having to know the database structure, and administrators to manage processes.

Metadata is information that is used by processes and application engines to define rules and physical objects such as tables or trees. For instance, you may want an application engine to process certain columns from certain tables. Instead of entering a long SQL statement, you can define a datamap.

Image: EPM data and metadata

This diagram illustrates the relationship between data and metadata in EPM.



This topic discusses:

- Metadata terms and objects.

- SQL object ID.
- Metadata utilities.

Metadata Terms and Objects

When we discuss metadata, we discuss:

- Technical metadata.

Technical metadata defines the physical structures of EPM and enables users to work with EPM Analytical Applications without the need to understand the underlying structures. Examples are record metadata, tablemaps, datamaps, filters, and constraints.

- Business intelligence metadata.

Business intelligence metadata defines how information is to be used to provide end users with the information they need to work most productively. An example of business intelligence metadata is the KPI Manager.

- Operational metadata.

Operational metadata provides information about the flow of data through EPM. An example of operational metadata is extract, transform, and load (ETL) jobs.

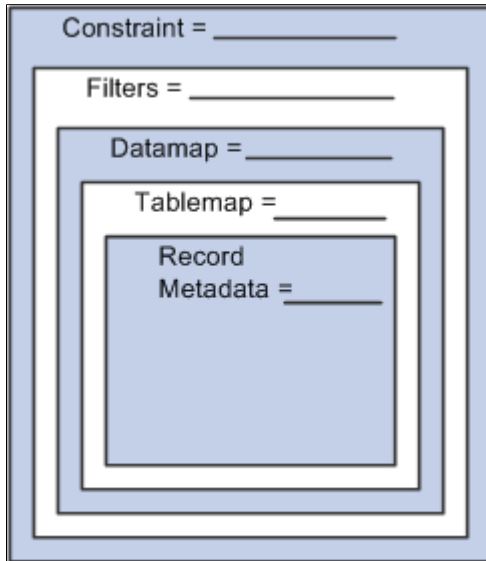
The metadata structure that is used by the PeopleSoft EPM engines is built in several layers or levels, each dependent on the layer below it. The following table lists the metadata layers that are found in PeopleSoft EPM in order from the lowest to the highest level:

<i>Metadata Object</i>	<i>Includes</i>
Record metadata	Defines the lowest level of EPM metadata table objects. This is the foundation on which all other metadata is defined. Record metadata defines and identifies EPM data tables.
Tablemaps	Define the physical relationships (joins) between tables and are foundations for datamaps.
Datamaps	Built on tablemaps, datamaps are similar to table definitions in that they describe a logical view of the tables themselves. They enable you to select information from different tables that are specified in a tablemap and define it as if it were one entity or table.
Constraints	Built on datamaps and can use one or more filters to define your business processing rules.

Metadata Object	Includes
Filters	Enable you to define what subset of data gets processed by or uses a specific business rule.

Image: EPM metadata, SQL

This diagram illustrates how PeopleSoft EPM metadata is nested to define a SQL statement.



The resulting SQL is SELECT (Datamap) FROM (Tablemap) WHERE (Constraints/Filters).

Other types of metadata and terms are:

ERP metadata

Examples are calendars, fiscal year, accounting period, business unit, and SetID.

See [Understanding PeopleSoft EPM Business Rule Setups](#).

Data sets

Used as input for various engines.

Expressions

Enable you to create virtual columns that are made up of mathematical calculations based on actual fields on a table.

Tree metadata

Captures information about the trees that you have set up in PeopleSoft EPM. It is mainly used for reporting.

Rule metadata

Gathers the rules for PeopleSoft EPM engines based on the standard keys of SetID, business unit, model ID, and effective date. Rule metadata is used by PeopleSoft EPM engines to recursively determine inheritance rules between models.

Balancing rules

Enable you to track before and after amounts in the system.

Job totals

Define flash totals for data to be used as input or output to the different source or target tables in the system.

PF_RECONCILIATION engine PF_RECONCILIATION uses the job totals and balancing rules metadata that you set up to validate balances.

Engine and job metadata Delivered metadata for running jobs and jobstreams.

Not all the metadata objects are described in this topic. Refer to the following topics for information about rule and engine metadata and balancing rules and job totals metadata.

Related Links

[Working with Engine Metadata](#)

[Setting Up Job Metadata](#)

[Using Balancing and Reconciliation Features](#)

SQL Object ID

The SQL object ID is a system-generated number that identifies a SQL object that is generated by a metadata component and is stored in the PeopleSoft SQL repository. The prefix identifies the PeopleSoft EPM product, metadata object, and sequential number for each metadata object respectively, for example PF\$_MR_140. Many setup pages within PeopleSoft EPM have a SQL object ID associated with the page definition.

Various types of SQL object ID prefixes exist, depending on the metadata object that you set up:

SQL Object ID Prefix	Metadata Object
PF\$_MR_#	Record Metadata
PF\$_RS_#	Record summary metadata
PF\$_TM_#	Tablemaps
PF\$_DM_#	Datamaps
PF\$_EX_#	Expressions
PF\$_CN_#	Constraints
PF\$_RL_#	Rule Metadata
PF\$_DS_#	Data sets
MD\$_xxx_#	Data Manager. Where xxx is the rule ID.
AB\$_yyy_xxx_#	ABM. Where yyy = SetID, and xxx = rule ID.

Metadata Utilities

Several utilities are available that you can use to search for and validate your metadata objects. These utilities include :

- Metadata Search engine: Enables you to search for metadata objects based on the description.

- Mass Validate: Enables you to check the validity of your metadata objects before running any PeopleSoft EPM engines or processes that depend on it.
- Impact Analysis tool: Enables you to determine the interdependencies of metadata before you change objects.

These utilities are described later in this documentation.

Warning! Only an experienced user should make changes to existing metadata.

Related Links

[Finding Metadata Objects](#)

[Running Mass Validate](#)

[Performing Impact Analysis](#)

Common Elements Used When Setting Up and Working with Metadata for the Operational Warehouse - Enriched

Hidden Object

This is a check box that enables power users to edit metadata objects while ensuring that everyday users can only view the objects. It enables another level of security to be applied to metadata.

Owner ID

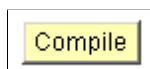
Assign an owner ID to a particular metadata object. The owner represents an EPM functional area (such as ABM or Budgeting) that is associated with a metadata object. Assigning an owner ID to the metadata organizes and groups the metadata, making it easier to locate and audit. After an owner ID is defined for record metadata, any tablemap, datamap, and constraint built on this record inherits the owner ID of the primary table.

Note: You can add owner ID values by updating the translate values in the Owner table (PF_OWNER). However, the added values represent a configuration and are not supported and must be migrated on upgrade.

Warning! When an owner ID is changed for parent metadata (for example, record metadata), the owner IDs for child metadata (for example tablemaps and datamaps) are not updated.

SQL Object ID Prefix

A system-generated number that identifies the prefix of the SQL that is built by this component in the SQL repository.



Click the Compile button on any of the metadata pages to build the metadata. Remember that you need to recompile any metadata that you change using this button.

Description

The Metadata search engine uses the description that you enter to find metadata objects.

Finding Metadata Objects

This topic describes how to search for metadata objects.

Page Used to Find Metadata Objects

<i>Page Name</i>	<i>Definition Name</i>	<i>Navigation</i>	<i>Usage</i>
Metadata Search	PF_SRCH_PANEL	EPM Foundation, Foundation Metadata, Other Metadata Operations, Metadata Search	Search for delivered and created metadata objects.

Metadata Search Page

Use the Metadata Search page (PF_SRCH_PANEL) to search for delivered and created metadata objects.

Navigation

EPM Foundation, Foundation Metadata, Other Metadata Operations, Metadata Search

Image: Metadata Search page

This example illustrates the fields and controls on the Metadata Search page. You can find definitions for the fields and controls later on this page.

Metadata Search

Search Keywords Search

Metadata Types

<input checked="" type="checkbox"/> Report	<input checked="" type="checkbox"/> TableMap	<input checked="" type="checkbox"/> Filter	<input checked="" type="checkbox"/> Record
<input checked="" type="checkbox"/> Engine	<input checked="" type="checkbox"/> DataMap	<input checked="" type="checkbox"/> Constraint	<input checked="" type="checkbox"/> Jobstream
<input checked="" type="checkbox"/> Tree	<input checked="" type="checkbox"/> Expression	<input checked="" type="checkbox"/> EW Data Set	<input checked="" type="checkbox"/> Job
<input checked="" type="checkbox"/> Metric			

Restrict SetID

Yes

No

Results Customize | Find | View 100 | First 1-25 of 4739 | Last

Metadata Name	Metadata Type	Effective Date	Description	i
BSC0001	Report Metadata	01/01/1900	Scorecard by Strategy Tree	i
BSC0002	Report Metadata	01/01/1900	Scorecard by KPI	i
BSC0003	Report Metadata	01/01/1900	Scorecards by Assessment Group	i
BSC0004	Report Metadata	01/01/1900	Strategy Trees, Non Term Node	i
BSC0005	Report Metadata	01/01/1900	Strategy Comp by Strat init	i
BSC0006	Report Metadata	01/01/1900	KPI's by Strategic Initiative	i
BSC0007	Report Metadata	01/01/1900	KPI and Calculation by Model	i
BSC0008	Report Metadata	01/01/1900	KPI's, Indicators, target Rule	i
BSC0009	Report Metadata	01/01/1900	Strategy Components by Perspec	i
BSC0010	Report Metadata	01/01/1900	KPI by Perspective	i
BSC0011	Report Metadata	01/01/1900	Strategy Components by KPI	i
BSC0012	Report Metadata	01/01/1900	Strategy Trees with Dup KPI's	i
BSC0013	Report Metadata	01/01/1900	Scorecard Assessment	i
BSC0014	Report Metadata	01/01/1900	KPI Detail	i
BSC0015	Report Metadata	01/01/1900	KPI Target Rule Value	i

Number of Rows Found: 4739

Search Keywords

Enter any keywords that you want to search by. You can use the AND, OR, and parentheses buttons to create a search string. Alternatively, you can leave this field blank and just search on a SetID.

Metadata Types

You can further narrow your search by specifying the metadata types to search on.



Click the All Metadata button to select all the metadata types that are listed.



Click the Clear all flags button to clear all the current selections in the Metadata Types group box.

Restrict SetID

Select Yes to restrict the search to a specific SetID. This applies only to data sets, constraints, and filters.

Search

After you have entered your criteria, click the Search button to activate the search. Depending on your criteria and the amount of metadata in your system, the search may take a while. The search returns metadata matching your criteria to the grid at the bottom of the page.



Click the Go to Setup Page button to go directly to the metadata setup page of any of the metadata objects that are returned.

Applying the Hidden Flag to Metadata Objects

To fully secure your metadata objects, it is recommended that you establish hidden flags. Hidden flags enable power users to edit metadata objects while ensuring that everyday users can only view the objects. Hidden flags provide you with an extra level of security for your metadata. Hidden flags use the PeopleTools My Personalizations feature to determine how to render the page for a user. If the user does not have access to the hidden flag through the personalization feature, then the page will be rendered as display only. This ensures that daily users are not able to modify or delete the metadata object.

Using this flag, system administrators can control access to metadata objects. System administrators may also assign power users access on an as-needed basis.

This flag acts as an attribute for the following metadata objects:

- Record metadata
- Tablemaps
- Datamaps
- Expressions
- Filters
- Constraints
- Data sets
- Dimension metadata

This topic discusses how to:

- Apply the hidden flag to metadata objects.
- Grant access to hidden metadata objects.

Applying the Hidden Flag to Metadata Objects

On any of the metadata setup pages for record metadata, tablemaps, datamaps, expressions, filters, constraints, or data sets, select the Hidden Flag check box to activate the hidden flag for that metadata object.

Granting Access to Hidden Metadata Objects

System administrators may grant access to hidden metadata objects for day-to-day users (power users) by enabling them to access the My Personalizations, EPM Expert User Settings page.

On this page, users can click the Personalize Option button for EPM Expert User Settings to access the Option Category: EPM Expert User Settings page. On this page, power users can set the override value for Display Hidden Objects to *Yes* to enable them to modify metadata objects that are hidden from day-to-day users.

Setting Up Record Metadata

This topic provides an overview of record metadata and discusses how to:

- Define record metadata.
- Review record column properties.
- Create and review related metadata objects.

Pages Used to Set Up Record Metadata

<i>Page Name</i>	<i>Definition Name</i>	<i>Navigation</i>	<i>Usage</i>
Record Metadata	PF_META_REC_TBL1	EPM Foundation, Foundation Metadata, Metadata Creation and Editing, Record Metadata, Record Metadata	Define record metadata for any new tables that you add to EPM.
Record Metadata - Field Properties	PF_META_REC_SEQ	EPM Foundation, Foundation Metadata, Metadata Creation and Editing, Record Metadata, Field Properties	Verify that your permanent and temporary tables are in sync.
Record Metadata - Table Description	PF_META_REC_NOTE	EPM Foundation, Foundation Metadata, Metadata Creation and Editing, Record Metadata, Table Description	Enter any notes that you want to associate with the record metadata.

Page Name	Definition Name	Navigation	Usage
Record Metadata - Related Metadata	PF_META_RELMD	EPM Foundation, Foundation Metadata, Metadata Creation and Editing, Record Metadata, Related Metadata	Create and review tablemaps, datamaps, and constraints that are related to specific record metadata.

Understanding Record Metadata

Record metadata defines the first level of metadata; that is, it defines the tables that are part of EPM . Your PeopleSoft database contains several types of tables:

- PeopleTools tables.
- PeopleSoft EPM business rules tables.
- PeopleSoft EPM data tables.

Record metadata defines and identifies the PeopleSoft EPM data tables only.

PeopleSoft delivers permanent data tables and the corresponding record metadata that identifies them as fact tables, fact reference tables, dimension tables, dimension reference tables, or transaction-dated tables.

If you add new tables, you must set up record metadata for each table that you add.

If you change a table, you must recompile record metadata for that table. If you add a non-key column to a table, you must recompile the record metadata. If you add a key column, you must recompile the record metadata and any tablemaps, datamaps, constraints, filters, or other metadata objects that are associated with it.

The Record Stub

Every permanent data table that is defined within EPM requires a shadow temporary table, known as the record stub. Shadow temporary tables have a similar record layout that generally matches the permanent tables. One exception to this is long varchar, long binary columns are removed. The temporary tables are defined as temporary tables on the PeopleTools record definition.

Record Metadata Page

Use the Record Metadata page (PF_META_REC_TBL1) to define record metadata for any new tables that you add to EPM.

Navigation

EPM Foundation, Foundation Metadata, Metadata Creation and Editing, Record Metadata, Record Metadata

Image: Record Metadata page

This example illustrates the fields and controls on the Record Metadata page. You can find definitions for the fields and controls later on this page.

Record Details

Temporary Table

Select a temporary table to use in parallel processing.

When the system uses temporary tables for parallel processing, it uses the temporary table in conjunction with the record suite table.

Note: Temporary tables need to be selected only for Operational Warehouse - Enriched (OWE) tables.

Table Type

Select the type of table for the selected record.

You can select *Dimension Reference Table*, *Dimension Table*, *Fact Reference Table*, *Fact Table*, *Security Join Table*, or *Transaction-Dated Fact Table*.

Merge Allowed

Select this check box if the table name needs to be merged from a temporary table to a final table by the Merge (PF_MERGE) application engine.

Generally, this check box is used for engine output tables only.

Warning! The following tables must never be marked for Merge: PF_LEDGER_F00, PF_JRNL_F00, PF_ADB_JRNL_F00, PF_LED_ADB_F00, LEDGER, or LEDGER_ADB.

Selective Merge

Select this check box to use in the selective merge delete clause.

Temporary tables that are created during the last step of a jobstream run are merged into permanent tables (F00 tables).

To avoid duplicates in the permanent tables, a delete must occur before the merge. Typically, the run control parameters are used to delete the necessary rows from the permanent tables and then the temporary tables are merged in. For some isolated cases, a more restrictive delete is necessary. The Selective Merge process performs a delete with even more criteria than just the run control parameters. Selective Merge bases its delete on the run control parameters plus the SQL Object ID that is entered in the record metadata settings for each specific table and settings within the Application Engine (AE).

Note: You generally do not use Selective Merge unless your application uses KPI Manager.

SQL Object ID

Select the ID of the SQL object for the selective merge.

Related Warehouse Tables

Error Table

Select the OWE error table related to the selected record.

The error table contains the error data that fails as part of the edit and modification process.

TSE Table Name

Select the TSE table related to the selected record.

The TSE (transaction editor set) table contains error message detail information.

Owner Details

Component

Select the component to which the record metadata belongs.

Select either *Multi-Dimensional Warehouse* or *Operational Warehouse - Enrich*.

Sub Component

Select the sub component, or type of data, associated with the selected record.

Select either *Enriched Data* or *Input Data*.

When you have completed your record metadata setup, click the Compile button to generate the SQL objects. You must also build your SQL anytime you make changes to the record metadata.

Record Metadata - Field Properties Page

Use the Record Metadata - Field Properties page (PF_META_REC_SEQ) to verify that your permanent and temporary tables are in sync.

Navigation

EPM Foundation, Foundation Metadata, Metadata Creation and Editing, Record Metadata, Field Properties

Image: Field Properties page

This example illustrates the fields and controls on the Field Properties page. You can find definitions for the fields and controls later on this page.

Record Metadata Field Properties Table Description Related Metadata					
Table Name: GC_ACCT_BAL_TBL Account Balancing Table			SQL Object: PF\$_MR_2301		
Record Columns					
	Key Field	Field Name	Field Type	Field on Temp Table	Resolve By
1	<input checked="" type="checkbox"/>	SETID	Char	<input checked="" type="checkbox"/>	SetID
2	<input checked="" type="checkbox"/>	ACCOUNT	Char	<input checked="" type="checkbox"/>	
3	<input checked="" type="checkbox"/>	EFFDT	Date	<input checked="" type="checkbox"/>	Effective Date
4	<input type="checkbox"/>	EFF_STATUS	Char	<input checked="" type="checkbox"/>	
5	<input type="checkbox"/>	ACCOUNT_TYPE	Char	<input checked="" type="checkbox"/>	
6	<input type="checkbox"/>	GC_BAL_FLAG	Char	<input checked="" type="checkbox"/>	

Key Field Select this check box if the field is a key on the permanent table.

Field Name Lists all the fields on the permanent table.

Field Type Lists the field type for each column.

Prompt Table Column in which you can enable the system to prompt for criteria and default value fields.

Field on Temp Table (field on temporary table) Select this check box for all fields that also appear on the temporary table. In general, this should be all fields, with the exception of transaction-dated tables.

Measure (Transaction-dated tables only). All numbers (DBFIELDTYPE = 2) and signed numbers (DBFIELDTYPE = 3) that are defined on the temporary table. When the SQL is generated for these fields in a list, they are enclosed in a sum construct, for example, sum (REPORTED_HRS).

Resolve By Select a key for use in the rule resolver process.

The Rule Resolver is an application engine program that is called by most PeopleSoft EPM engines to gather the rule sets that are used in processing. The main function of the Rule Resolver is to gather the rules for a given process run based on the standard keys of SetID, business unit, model ID, fiscal year, accounting period, and effective date.

Resolver fields are used for SQL object generation that enables applications to select data based on the run control parameters.

The Resolver reduces the amount of data that an application engine needs to process by populating tables with only the data necessary for the engine to run. Individual application engines call the Resolver as part of their run process.

Note: If an OWE table is selected, the resolve by field is automatically selected and you cannot edit this field. If a MDW table is selected, you can select a resolve by field.

Click the Table Description tab to enter a more detailed description of the record metadata.

Record Columns for Transaction-Dated Tables

At resolution time, data is selected from the permanent transaction-dated fact table by transaction date and stored in the temporary table in a fiscal year and period format so that it is processed in the same way as in any other table.

For example, if an engine runs for 2002 and is based on a monthly calendar, all transaction dates between January 1, 2002 and January 31, 2002 are selected from the transaction-dated fact table and inserted into the temporary table. Because more than one transaction with the same key information may exist, some aggregation must occur. A group by clause is generated that includes all fields on the temporary table that are not resolved or are not measure fields. You use the Record Columns page to see how the fields are defined.

Record Metadata - Related Metadata Page

Use the Record Metadata - Related Metadata page (PF_META_RELMD) to create and review tablemaps, datamaps, and constraints that are related to specific record metadata.

Navigation

EPM Foundation, Foundation Metadata, Metadata Creation and Editing, Record Metadata, Related Metadata

Image: Related Metadata page

This example illustrates the fields and controls on the Related Metadata page.

The screenshot shows the 'Related Metadata' page for the record 'GC_ACCT_BAL_TBL Account Balancing Table' (SQL Object: PF\$_MR_2301). It features two main sections: 'TableMaps' and 'DataMaps'. Each section contains a table with columns for ID, search icon, Name, Description, and Target Table Map. Below each table is a form to add new entries, with fields for Name, Description, and Target Table Map, and an 'Add' button.

The Related Metadata page enables you to create and review tablemaps and datamaps that are related to particular record metadata. The first time you define record metadata, you can use this page to create related tablemaps and datamaps at the same time that you create the record. Simply enter a name and description for the objects and click the add button—the objects are created automatically with the same name as the related record metadata. The SQL IDs for each object are also created automatically.

If you have already defined record metadata, you can use this page to review the tablemaps and datamaps that are built on top of the record metadata. Click any of the metadata objects to access their primary page (for example, the Tablemap page).

Setting Up Rule Metadata

This topic provides an overview of rule metadata and discusses how to define rule metadata.

Page Used to Set Up Rule Metadata

Page Name	Definition Name	Navigation	Usage
Rule Metadata	PF_METARULE_TBL	EPM Foundation, Foundation Metadata, Metadata Creation and Editing, Rule Metadata	Define rule metadata. Rule metadata is delivered with EPM and generates the SQL that is needed to resolve rules.

Understanding Rule Metadata

Rule metadata is used in conjunction with the Rule Resolver to specify the relationship between parent and child models. Unless you create a new PeopleSoft EPM engine, you will not likely need to create any rule metadata. PeopleSoft EPM engines use rule metadata to recursively determine inheritance rules between models.

Understanding the Rule Resolver

The Rule Resolver is an application engine program that is called by most PeopleSoft EPM engines to gather the rule sets that are used in processing. The main function of the Rule Resolver is to gather the rules for a given process run based on the standard keys of SetID, business unit, model ID, fiscal year, accounting period, and effective date. The system does this by passing the Rule Resolver a list of parent tables that need to be resolved. The related child tables (as defined in the Rule Metadata page) are also resolved as part of the Rule Resolver. In addition, if a table is keyed by model ID, the Rule Resolver determines what the parent models are and resolves rules for them.

Rule Metadata Page

Use the Rule Metadata page (PF_METARULE_TBL) to define rule metadata. Rule metadata is delivered with EPM and generates the SQL that is needed to resolve rules.

Navigation

EPM Foundation, Foundation Metadata, Metadata Creation and Editing, Rule Metadata

Image: Rule Metadata page

This example illustrates the fields and controls on the Rule Metadata page. You can find definitions for the fields and controls later on this page.

Rule Metadata

Record Name: CAL_DEFN_TBL Calendars SQL Object: PFS_RL_141

*Temporary Record: CAL_DEFN_T *Object Owner ID: Enterprise Warehouse Compile

Resolve

SetID Business Unit Model ID

Effective Date Fiscal Year Accounting Period

Child Tables Customize | Find | First 1 of 1 Last

Child Record Name	*Temporary Record
CAL_DETP_TBL	CAL_DETP_T

Resolve

The settings in this group box are set by default based on the table name that you select. You cannot update this information.

Child Record Name

Select the appropriate record name. This includes all child tables that must be resolved along with the parent table.

Record Stub

Select the appropriate record stub.

When you have completed your rule metadata setup, click the Compile button to generate the SQL objects. You must also build your SQL anytime you make changes to the rule metadata.

Setting Up Tablemaps

This topic provides an overview of tablemaps, lists common elements, and discusses how to:

- Define tablemaps.
- Define reference tables.
- View SQL IDs.

Pages Used to Set Up TableMaps

<i>Page Name</i>	<i>Definition Name</i>	<i>Navigation</i>	<i>Usage</i>
TableMap	PF_TBLMAP_DEFN	EPM Foundation, Foundation Metadata, Metadata Creation and Editing, TableMap, TableMap	Define tablemap and tablemap child tables.
Reference	PF_TBLMAP_REF	EPM Foundation, Foundation Metadata, Metadata Creation and Editing, TableMap, Reference	Define reference and reference child tables.
SQL IDs	PF_TBLMAP_REF	EPM Foundation, Foundation Metadata, Metadata Creation and Editing, TableMap, SQL IDs	View the generated SQL object ID prefixes for the tables that are defined in this tablemap.

Understanding Tablemaps

Tablemaps define the physical relationships between your PeopleSoft data warehouse tables. Tablemaps enable you to define families of related data warehouse tables and the columns that define the key relationships (or joins) between the tables. Tablemaps describe the parent-child relationships between tables by defining the common fields that join them together.

A tablemap can be defined as identifying a primary table only, or you may specify any combination of child tables, reference tables, and reference child tables:

The types of tables are:

Primary table

Often a fact table, the primary table, from a logical point of view, to which the other tables are related.

Child table

A hierarchical child of the primary table (all the same keys, plus one). Child tables enable you to have a one-to-many relationship between a primary table and its child tables and to define the relationship between these tables through the tablemaps. You specify a child table only if you need fields from that table in your tablemap.

Reference table

A table that defines the properties for one or more keys in the primary table. Reference tables contain additional attributes, or properties, of the transactional fact information. Some examples of reference tables are Customers, Products, Channels, Geography, and Time. A reference table always has one or more key fields that it shares with the primary table. These fields are defined on this page and matched with the corresponding fact field to enable engines to access data through the join. Each reference table may be the parent table to one or more reference child tables.

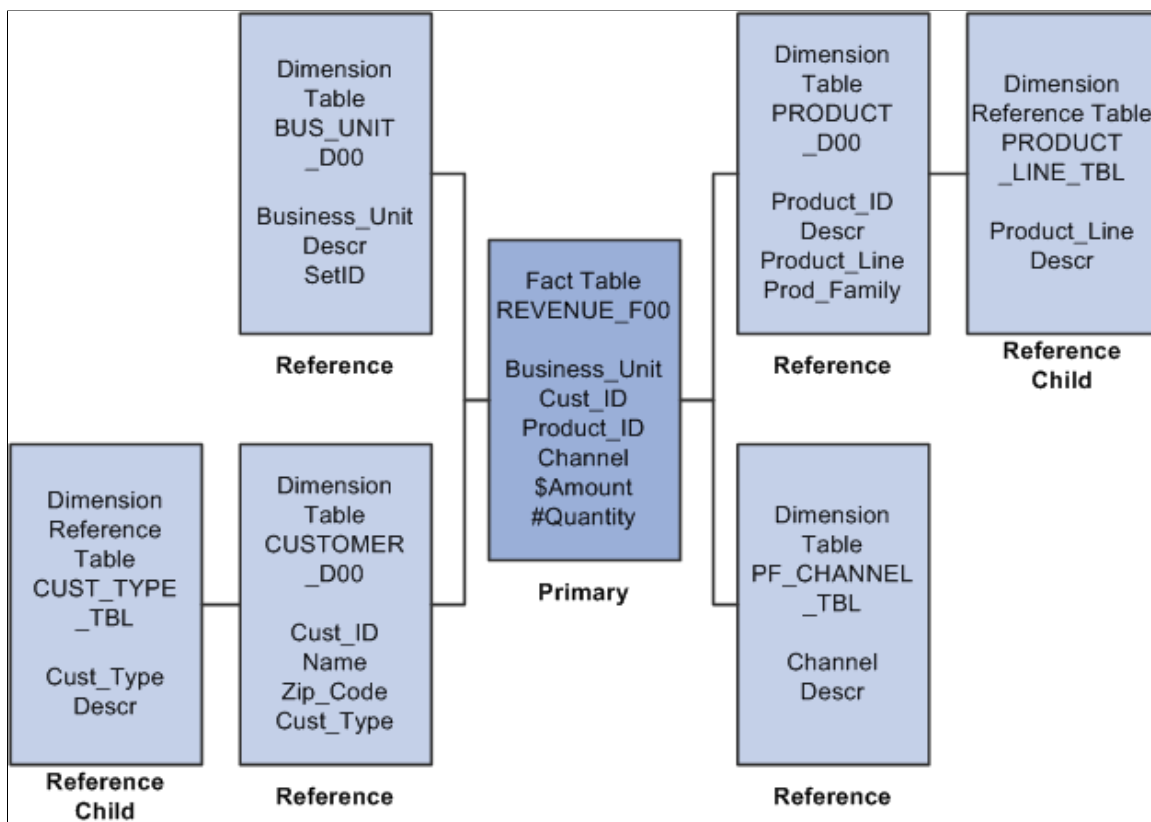
Reference child table

A hierarchical child of the reference table (all the same keys, plus one). Reference child tables are similar to child tables in that they allow for multiple types or rows of data per related dimension table. An example of a reference child and its related dimension is a customer demographic table's relationship to a customer table. That is, you may want to track various demographic values for each customer, such as salary range and geographic codes

The relationship between these tables is often referred to as a star schema. A star schema is a group of tables that are related to a central fact table. A single fact table can have numerous fact references and dimensions and each dimension can have numerous dimension references.

Image: Example of relationship between tables (star schema)

This diagram illustrates an example of table relationships.



Tablemaps are used as a basis for defining datamaps, filters, and ultimately constraints. After a tablemap is set up, you will likely keep it as it is. You therefore have many datamaps defined for each tablemap.

We deliver a set of tablemaps with EPM.

Warning! If you plan at the leaf level instead of the node level for a dimension, you must remove the reference table for that dimension. When you set up your data set that uses this dimension, you must select to group by dimension ID and *not* dimension node.

Common Elements Used to Set Up TableMaps

Join Mapping

Depending on the key field type, you may or may not be able to edit this field. The system automatically resolves certain fields.

Fields that you can edit are in white. If you change the method to *Map One to One =*, then you can specify the primary field.

Map One to One = is the default if the reference key field has the same name as on the primary table.

Primary Field

If you change the join mapping to *Map One to One =*, then you can specify the primary field using the valid values that are provided by the system.

Additional Field Lookup Table

If join mapping is set to *Additional Key*, you can specify that key value in this field using the valid values that are provided by the system. *Additional Key* is the default when the key cannot be resolved and if it does not have the same name on the primary table. In a valid tablemap, every child has only one additional key.

TableMap Page

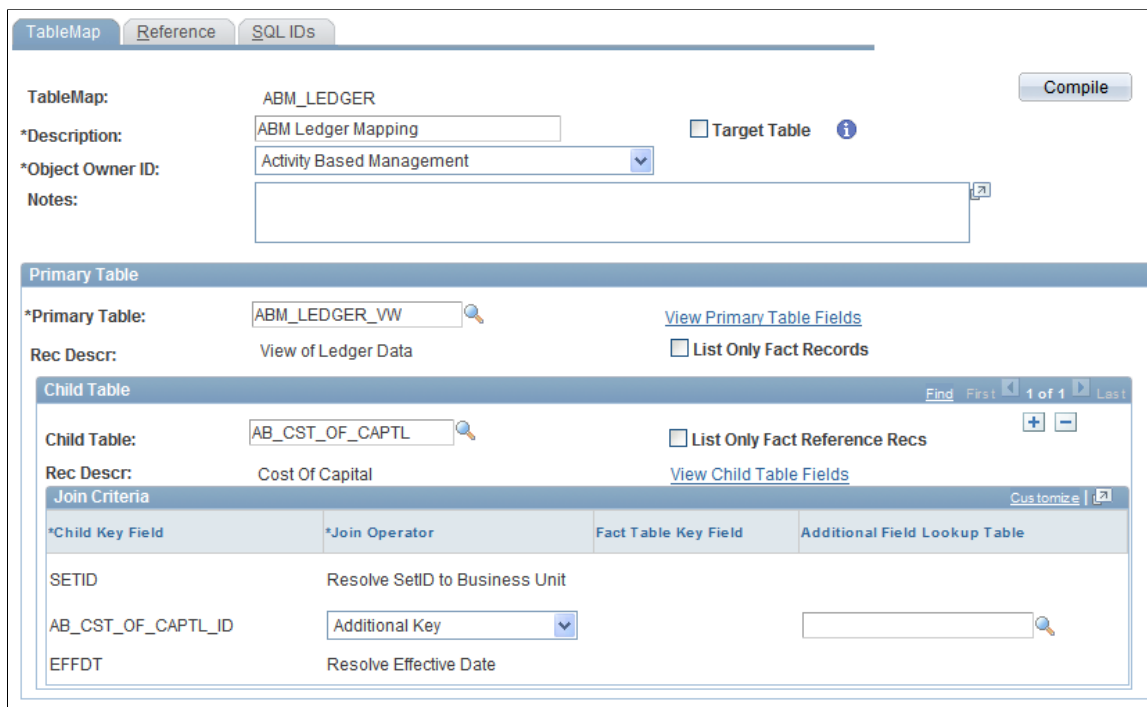
Use the TableMap page (PF_TBLMAP_DEFN) to define tablemap and tablemap child tables.

Navigation

EPM Foundation, Foundation Metadata, Metadata Creation and Editing, TableMap, TableMap

Image: TableMap page

This example illustrates the fields and controls on the TableMap page. You can find definitions for the fields and controls later on this page.



Target Table

Select this check box to define the primary table as a target table for Data Manager, Allocation Manager, or another application engine.

You must select this check box if you are using this table as a target for Data Manager or Allocation. If you select this check box, the Reference page is disabled.

Compile

Note: This check box is unavailable for editing when input MDW tables are selected.

Click to generate the SQL objects for the tablemap metadata.

You must rebuild your SQL anytime you make changes to the tablemap, such as when you delete or add a key field.

Note: If you need to recompile a tablemap, you must also recompile any datamaps or other metadata objects that are associated with the tablemap.

Primary Table**Primary Table**

Select the table that you want to use as the primary table for this tablemap.

Only tables that have been defined in record metadata appear as valid values.

A primary table is the center of the tablemap. Generally, a primary table will be a fact table. Fact tables contain the data that is relevant to a single business transaction, and they also have a unique key structure that can identify other related fact reference or child tables. A tablemap can contain only one primary table, but it may be related to many child and reference tables.

View Primary Table Fields

Click to go to the Field Properties page and review the fields that are associated with the underlying record metadata.

List Only Fact Records

Select this check box if you want to view only the tables that are defined in record metadata as fact table type.

Child Fact Tables**Child Table**

Select the appropriate child table to join with the primary table for this tablemap.

Only tables that have been defined in record metadata appear as valid values.

List Only Fact Reference Recs (list only fact reference records)

Select this check box if you want to view only the tables that are defined in record metadata as fact table type.

Child Key Field

Displays the key field of the child table used to join to the fact table key field.

This key field relates the child table to the primary table.

Join Operator

Select a method to join the child key field and the fact table key field.

If you select *Additional Key* to use in the join, you must specify the record in which the additional key resides.

Fact Table Key Field

Select the key field of the primary fact table used to join to the child key field.

This field relates the fact table to the child table. In some cases, this field is display only.

Additional Field Lookup Table

Select the record in which the additional key for the join resides.

This field is available only when you select *Additional Key* in the Join Operator field.

Reference Page

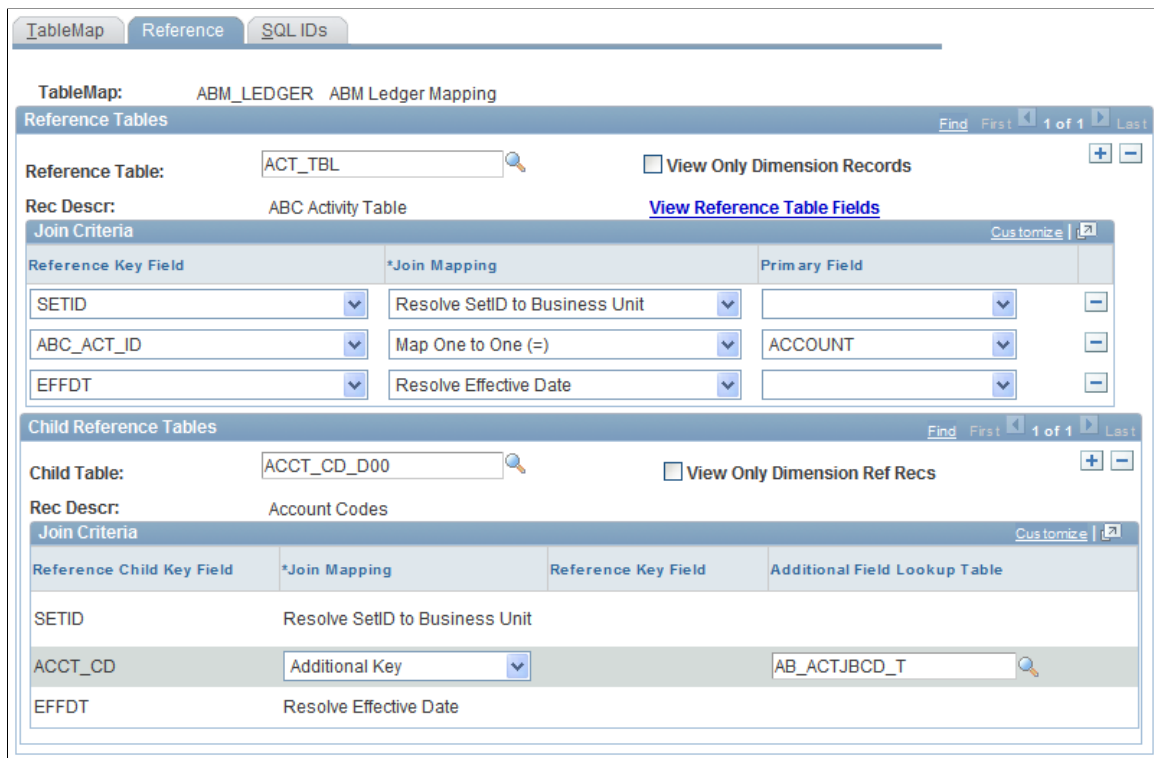
Use the Reference page (PF_TBLMAP_REF) to define reference and reference child tables.

Navigation

EPM Foundation, Foundation Metadata, Metadata Creation and Editing, TableMap, Reference

Image: Reference page

This example illustrates the fields and controls on the Reference page. You can find definitions for the fields and controls later on this page.



Reference Tables

Reference Table

Select the appropriate reference table that you want to join to the primary table.

Only tables that have been defined in record metadata appear as valid values.

View Only Dimension Records

Select this check box to view only the tables that are defined in record metadata as dimension table type.

View Reference Table Fields

Click to access the Field Properties page and view the record metadata fields related to the selected reference table.

Reference Key Field

Displays the key field of the reference table used to join to the primary table key field.

Join Mapping

Select a method to join the reference key field and the primary table key field.

Primary Field

Select the key field of the primary table used to join to the reference key field.

Child Reference Tables

Child Table

Select the appropriate reference child table that you want to join to the primary table.

Only tables that have been defined in record metadata appear as valid values.

View Only Dimension Ref Recs (view only dimension reference records) Select this check box to view only the tables that are defined in record metadata as dimension table type.

Reference Child Key Field

Displays the key field of the child reference table used to join to the reference table key field.

This key field relates the child reference table to the primary reference table.

Join Mapping

Select a method to join the reference child key field and the reference table key field.

If you select *Additional Key* to use in the join, you must specify the record in which the additional key resides.

Reference Key Field

Select the key field of the reference table used to join to the reference child key field.

Additional Field Lookup Table

Select the record in which the additional key for the join resides.

This field is available only when you select *Additional Key* in the Join Mapping field.

SQL IDs Page

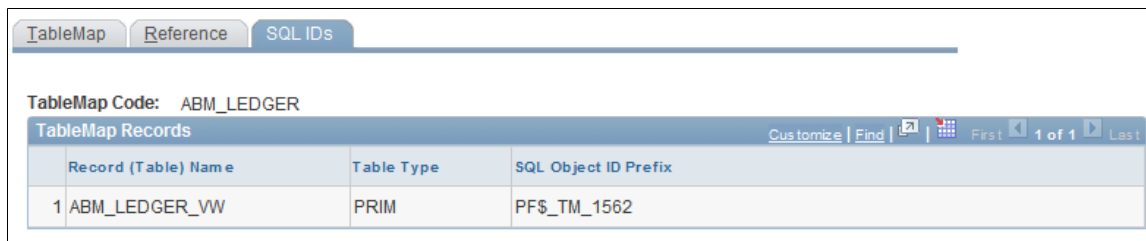
Use the SQL IDs page (PF_TBLMAP_REF) to view the generated SQL object ID prefixes for the tables that are defined in this tablemap.

Navigation

EPM Foundation, Foundation Metadata, Metadata Creation and Editing, TableMap, SQL IDs

Image: SQL IDs page

This example illustrates the fields and controls on the SQL IDs page.



After the SQL has been compiled, you can view the generated SQL object ID prefixes for the tables that are defined in this tablemap.

Setting Up Datamaps

This topic provides an overview of datamaps and discusses how to:

- Define datamaps.
- View datamap fields.
- View or add constraints for a datamap.
- Use the DataMap Wizard to create datamaps.

Pages Used to Set Up Datamaps

<i>Page Name</i>	<i>Definition Name</i>	<i>Navigation</i>	<i>Usage</i>
General Properties	PF_DATAMAP_DEFN	EPM Foundation, Foundation Metadata, Metadata Creation and Editing, DataMap, General Properties	Define datamaps.
DataMap Fields	PF_DATAMAP_DEFN2	Click DataMap Fields on the General Properties page.	View DataMap fields.
Constraints	PF_DMAP_CONSTRNTS	EPM Foundation, Foundation Metadata, Metadata Creation and Editing, DataMap, Constraints	View or add constraints for a specific datamap.

Page Name	Definition Name	Navigation	Usage
Datamap Wizard	PF_DATAMAP_WIZ, PF_DATAMAP_SUM	EPM Foundation, Foundation Metadata, Metadata Wizards, Datamap Wizard	Simplify datamap setup using the Datamap Wizard.

Understanding Datamaps

A datamap is the third level of metadata that builds upon the information that you captured in the tablemap and enables you to define a logical view of the physical EPM tables. Datamaps bring together information from the different tables that can be specified in a tablemap and defines it as if it were one entity or table. Not every column of every table that is defined in a tablemap is necessary; datamaps enable you to select only those columns that you want to use.

You can group fields that are related to common processes so that processing can be done on one data set. In addition, datamaps enable you to give columns more intuitive, meaningful names, making data retrieval and review easier for business users.

Warning! If you plan at the leaf level instead of the node level for a dimension, you must remove the row that contains this dimension. When you set up your data set that uses this dimension, you must select to group by dimension ID and *not* dimension node.

Understanding Datamap Setup

PeopleSoft provides you with the option of setting up datamaps using the *DataMap Wizard*, which is intended to simplify the datamap creation process. You can still use the Record Metadata, TableMap, and DataMap components to manually set up datamaps.

See [Setting Up Record Metadata](#), [Setting Up Tablemaps](#).

General Properties Page

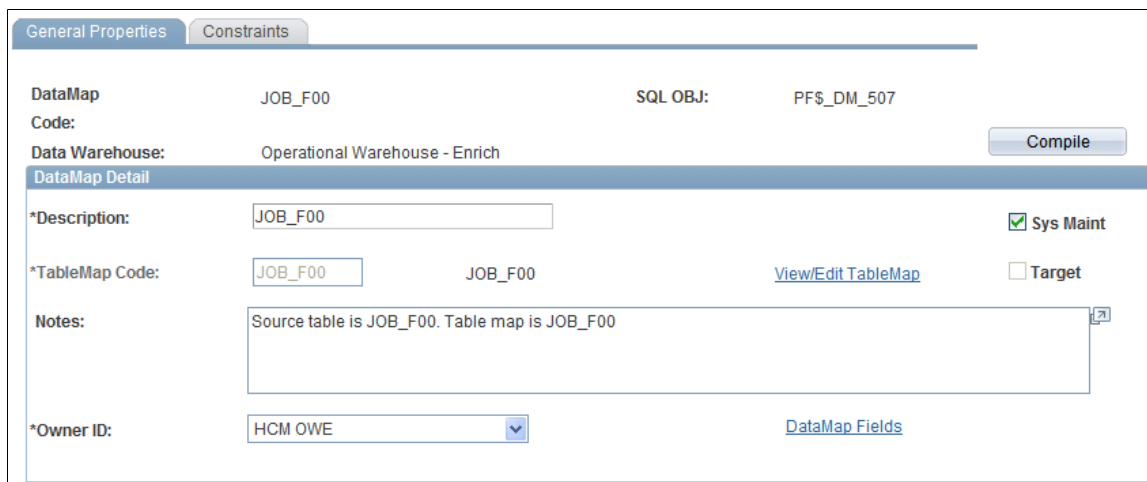
Use the General Properties page (PF_DATAMAP_DEFN) to define datamaps.

Navigation

EPM Foundation, Foundation Metadata, Metadata Creation and Editing, DataMap, General Properties

Image: DataMap - General Properties page

This example illustrates the fields and controls on the DataMap - General Properties page. You can find definitions for the fields and controls later on this page.



TableMap Code

Select a tablemap code. The code selected here limits the records that you can include in your datamap to the ones that are defined in the tablemap. When you add or change this value, the page automatically inserts each field from the tablemap's primary table into the grid. Generally, you include all fields from the primary table and add only a few from the reference tables. You can add or remove rows from the grid below. If your tablemap is a target table, you cannot insert or delete rows.

View/Edit TableMap

Click to access the TableMap page and view the tablemap associated with the selected datamap.

Target

This check box is controlled by the tablemap that the datamap is based on. If the tablemap that you select is a target, the check box is selected.

DataMap Fields

Click to access the DataMap - DataMap Fields page and view or edit the fields that are associated with this datamap.

When you have completed your datamap setup, click the Compile button to generate the SQL objects. You must also build your SQL anytime you make changes to the datamap, such as when you delete or add a key field.

Note: If you need to recompile a datamap, you must also recompile any tablemaps or other metadata objects that are associated with the datamap.

DataMap Fields Page

Use the DataMap Fields page (PF_DATAMAP_DEFN2) to view DataMap fields.

Navigation

Click DataMap Fields on the General Properties page.

Image: DataMap Fields page

This example illustrates the fields and controls on the DataMap Fields page. You can find definitions for the fields and controls later on this page.

Expr	Record	Field Name	Description	Type	Key Value	Lookup Table
<input type="checkbox"/>	JOB_F00	BUSINESS_UNIT	Business Unit	Attribute		SP_BU_FF_NONV Char
<input type="checkbox"/>	JOB_F00	EMPLID	EmplID	Attribute		PERSONAL_SRCH Char
<input type="checkbox"/>	JOB_F00	EMPL_RCD_NBR	Employment Rcd Nbr	Attribute		Number
<input type="checkbox"/>	JOB_F00	ASOF_DT	As Of Date	Attribute		Date
<input type="checkbox"/>	JOB_F00	EFFSEQ	Effective Sequence	Attribute		Number
<input type="checkbox"/>	JOB_F00	ASOF_STATUS	Status of As of Date	Attribute		Char
<input type="checkbox"/>	JOB_F00	ACTION	Action	Attribute		Char
<input type="checkbox"/>	JOB_F00	ACTION_REASON	Reason Code	Attribute		WA_ACTN_RSN_D Char
<input type="checkbox"/>	JOB_F00	ANNL_BENEF_BASE_RT	Annual Benefits Base Rate	Attribute		Rate
<input type="checkbox"/>	JOB_F00	ANNL_BBASE_BCE_AMT	Annual Benefit Base Cur Equiv	Attribute		Amount
<input type="checkbox"/>	JOB_F00	ANNUAL_RT	Annual Rate	Attribute		Rate
<input type="checkbox"/>	JOB_F00	ANNUAL_BCE_AMT	Annual Rate (BCE)	Attribute		Rate
<input type="checkbox"/>	JOB_F00	BAS_ACTION	BAS Action	Attribute		Char
<input type="checkbox"/>	JOB_F00	BAS_GROUP_ID	BAS Group ID	Attribute		Char
<input type="checkbox"/>	JOB_F00	BEN_STATUS	Benefits Employee Status	Attribute		Char

Expr (expression)

To create a virtual column, add a new blank row, select the Expr check box, and click the Expression button. This accesses the Expressions page.

After you have defined and saved your expression, all the changes made to the DataMap page are also saved. You cannot change an existing row into an expression.

See [Setting Up Expressions](#).

Record

Contains a list of all the records that are defined in the tablemap on which this datamap is based.

Field Name

Lists all the possible fields for the selected record. When you add or change a field, the description is automatically populated with the column field name that you select. You may then edit the description to be anything you want under 30 characters long.

Description

Change the name of a field to enable you to create more intuitive names that provide more meaning than the original column name when creating your datamap. The names that you select here will show up on reports and are used in filters, constraints, and by the Data Manager.

Type

Select *Attribute*, *Dimension*, or *Measure*. The system's default is *Attribute*.

The type field is important when you are defining datamaps for the Data Manager to verify rule definitions.

Note: The types mentioned here are different from the types that are used by PeopleSoft EPM metadata and OLAP objects.

Key Value

Required for fields from child tables. Enables you to define more than one column from a single column based on different values in the lookup table code on the child table.

Lookup Table

This field is available for character fields and is used when you are defining filters to limit the field to valid values that are contained in the Lookup table.

Field Type

Displays the field type that is associated with the field. Its value is based on the field type that is defined in PSDBFIELD. Number fields consist of the following types: 1 (Basis Points), 2 (Rate), 3 (Amount), and 5 (Number). These types represent ways in which the field will be summed.

Constraints Page

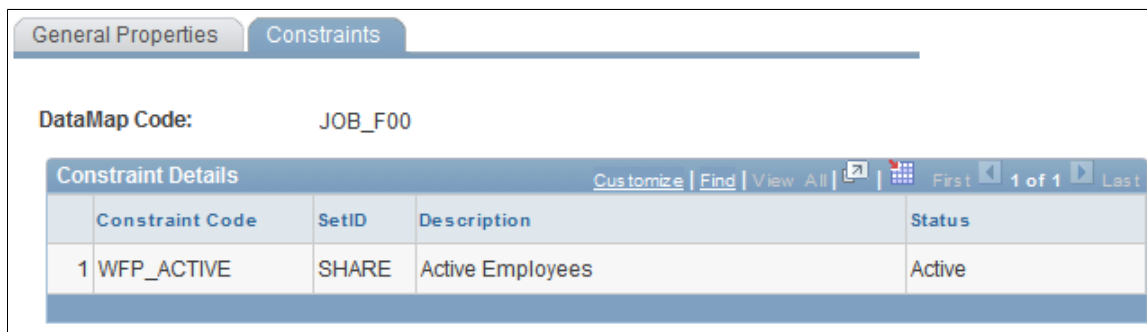
Use the Constraints page (PF_DMAP_CONSTRNTS) to view or add constraints for a specific datamap.

Navigation

EPM Foundation, Foundation Metadata, Metadata Creation and Editing, DataMap, Constraints

Image: DataMap - Constraints page

This example illustrates the fields and controls on the DataMap - Constraints page.



You can use this page to view constraints associated with a datamap. You can also add new constraints to the datamap.

Datamap Wizard Page

Use the Datamap Wizard page (PF_DATAMAP_WIZ, PF_DATAMAP_SUM) to simplify datamap setup using the Datamap Wizard.

Navigation

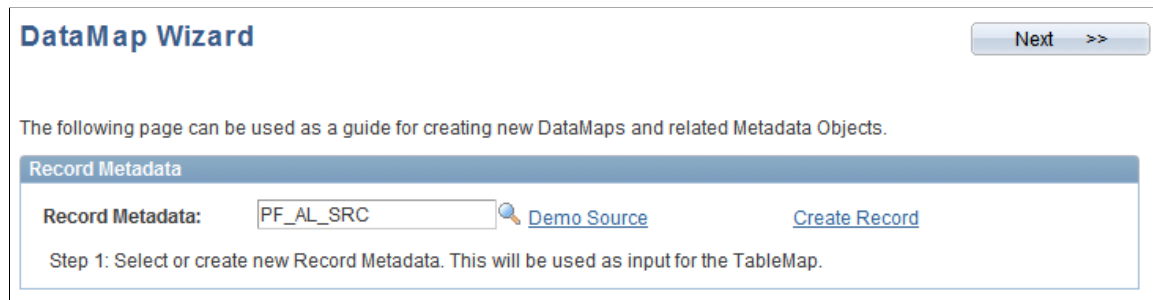
EPM Foundation, Foundation Metadata, Metadata Wizards, DataMap, Wizard

DataMap Wizard - Record Metadata (Step 1 of 6)

The first view of the DataMap Wizard enables you to define Record metadata for your DataMap.

Image: Datamap Wizard - Record Metadata input

This example illustrates the fields and controls on the Datamap Wizard - Record Metadata input. You can find definitions for the fields and controls later on this page.



Next

Click to advance to the next step of the DataMap Wizard and confirm your record metadata selection.

Record Metadata

Select the record metadata (primary table) to associate with the tablemap (which is used as the basis for your datamap)

Generally, a primary table will be a fact table.

Only tables that have been defined in record metadata appear as valid values.

Create Record

Click to access the Record Metadata page and create a new primary table to associate with the tablemap.

Note: Once you select record metadata for the wizard, a new link appears that can access the Record Metadata page for that specific record. The link name is determined by the description provided on the Record Metadata page.

In the example above the link is named *Demo Source* because that is the description given for the PF_AL_SRC record on the Record Metadata page.

DataMap Wizard - TableMap Metadata (Step 2 of 6)

The second view of the DataMap Wizard enables you to define TableMap metadata for your DataMap.

Image: Datamap Wizard - TableMap Metadata input

This example illustrates the fields and controls on the Datamap Wizard - TableMap Metadata input. You can find definitions for the fields and controls later on this page.

The screenshot shows the 'DataMap Wizard' interface. At the top right, there are two buttons: '<< Previous' and 'Next >>'. Below the title, a message states: 'The following page can be used as a guide for creating new DataMaps and related Metadata Objects.' The main content area is divided into two sections:

- Record Metadata:** A text input field contains 'PF_AL_SRC' and a link 'Demo Source' is visible. Below it, the instruction reads: 'Step 1: Select or create new Record Metadata. This will be used as input for the TableMap.'
- TableMap:** A text input field contains 'ALLOC_SRC' and a magnifying glass icon is next to it. To the right, there are two links: 'Demo Source For Allocations' and 'Create Tablemap'. Below it, the instruction reads: 'Step 2: Select or create a new TableMap. This will be used as input for the DataMap.'

Previous

Click to return to the previous step of the DataMap Wizard.

Next

Click to advance to the next step of the DataMap Wizard and confirm your tablemap selection.

TableMap

Select a tablemap to associate with the datamap.

Create TableMap

Click to access the TableMap page and create a new tablemap to associate with the datamap.

Once you select a tablemap for the wizard, a new link appears that can access the TableMap page for that specific tablemap. The link name is determined by the description provided on the TableMap page.

Note that as you progress through the DataMap Wizard steps, the previous section becomes unavailable for input. You can click Previous at any time to return to the previous step.

DataMap Wizard - DataMap Metadata (Step 3 of 6)

The third view of the DataMap Wizard enables you to define your DataMap.

Image: Datamap Wizard - DataMap Metadata input

This example illustrates the fields and controls on the Datamap Wizard - DataMap Metadata input. You can find definitions for the fields and controls later on this page.

DataMap Wizard

<< Previous
Next >>

The following page can be used as a guide for creating new DataMaps and related Metadata Objects.

Record Metadata

Record Metadata: [Demo Source](#)

Step 1: Select or create new Record Metadata. This will be used as input for the TableMap.

TableMap

TableMap: [Demo Source For Allocations](#)

Step 2: Select or create a new TableMap. This will be used as input for the DataMap.

DataMap

DataMap: [Demo Source For Allocations](#) [Create Datamap](#)

Step 3: Select or create a new DataMap.

Previous

Click to return to the previous step of the DataMap Wizard.

Next

Click to advance to the next step of the DataMap Wizard and confirm your datamap selection.

DataMap

Select a datamap.

Create DataMap

Click to access the DataMap page and create a new datamap.

Once you select a datamap for the wizard, a new link appears that can access the DataMap page for that specific datamap. The link name is determined by the description provided on the DataMap page.

Note that as you progress through the DataMap Wizard steps, the previous section becomes unavailable for input. You can click Previous at any time to return to the previous step.

DataMap Wizard - Filter Metadata (Step 4 of 6)

The fourth view of the DataMap Wizard enables you to define filter metadata for your DataMap.

Image: Datamap Wizard - Filter Metadata input

This example illustrates the fields and controls on the Datamap Wizard - Filter Metadata input. You can find definitions for the fields and controls later on this page.

DataMap Wizard << Previous Next >>

The following page can be used as a guide for creating new DataMaps and related Metadata Objects.

Record Metadata

Record Metadata: [Demo Source](#)

Step 1: Select or create new Record Metadata. This will be used as input for the TableMap.

TableMap

TableMap: [Demo Source For Allocations](#)

Step 2: Select or create a new TableMap. This will be used as input for the DataMap.

DataMap

DataMap: [Demo Source For Allocations](#)

Step 3: Select or create a new DataMap.

Filter

Filter: [Demo Source for Allocations](#) [Create Filter](#)

Step 4: Select or create a new Filter. This step is optional.

Previous

Click to return to the previous step of the DataMap Wizard.

Next

Click to advance to the next step of the DataMap Wizard and confirm your filter selection.

Filter

Select a filter to associate with the datamap.

Create Filter

Click to access the Filter page and create a new filter.

Once you select a filter for the wizard, a new link appears that can access the Filter page for that specific filter. The link name is determined by the description provided on the Filter page.

Note that as you progress through the DataMap Wizard steps, the previous section becomes unavailable for input. You can click Previous at any time to return to the previous step.

DataMap Wizard - Constraint Metadata (Step 5 of 6)

The fifth view of the DataMap Wizard enables you to define constraint metadata for your DataMap.

Image: Datamap Wizard - Constraint Metadata input

This example illustrates the fields and controls on the Datamap Wizard - Constraint Metadata input. You can find definitions for the fields and controls later on this page.

DataMap Wizard << Previous Next >>

The following page can be used as a guide for creating new DataMaps and related Metadata Objects.

Record Metadata

Record Metadata: [Demo Source](#)

Step 1: Select or create new Record Metadata. This will be used as input for the TableMap.

TableMap

TableMap: [Demo Source For Allocations](#)

Step 2: Select or create a new TableMap. This will be used as input for the DataMap.

DataMap

DataMap: [Demo Source For Allocations](#)

Step 3: Select or create a new DataMap.

Filter

Filter: [Demo Source for Allocations](#)

Step 4: Select or create a new Filter. This step is optional.

Constraint

Constraint: [Demo Source For Allocations](#) [Create Constraint](#)

Step 5: Select or create a new Constraint. This step is optional.

Previous

Click to return to the previous step of the DataMap Wizard.

Next

Click to advance to the next step of the DataMap Wizard and confirm your constraint selection.

Constraint

Select a constraint to associate with the datamap.

Create Constraint

Click to access the Constraint page and create a new constraint.

Once you select a constraint for the wizard, a new link appears that can access the Constraint page for that specific constraint. The link name is determined by the description provided on the Constraint page.

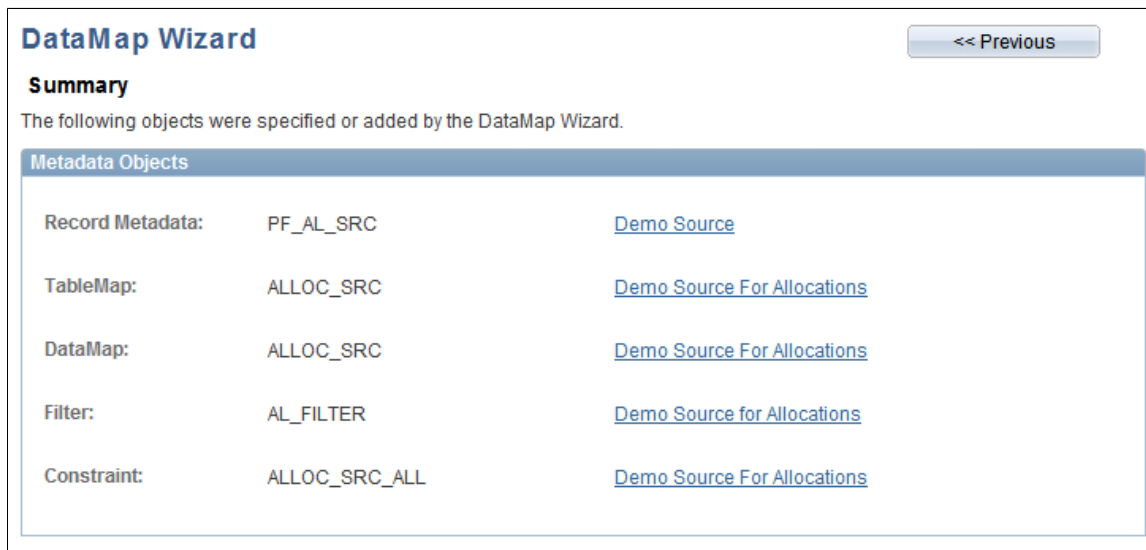
Note that as you progress through the DataMap Wizard steps, the previous section becomes unavailable for input. You can click Previous at any time to return to the previous step.

DataMap Wizard - DataMap Summary (Step 6 of 6)

The sixth and final view of the DataMap Wizard provides you with a summary of all the metadata defined for your DataMap.

Image: Datamap Wizard - DataMap summary

This example illustrates the fields and controls on the Datamap Wizard - DataMap summary.



You can click on the links to the right of each metadata object to return to their main criteria pages and make adjustments to the rules.

You can also click Previous to return to the previous DataMap Wizard steps.

Setting Up Expressions

Expressions enable you to create virtual columns that are made up of mathematical calculations based on actual fields on a table. Because expressions are resolved at runtime, duplicate information is not stored in the database. Expressions are user-defined columns that you add to a datamap. After you add them as columns to the underlying datamap, you can use them in filters just like record fields. An expression can be either numeric or a string. Numeric expressions can combine any number of record fields, value objects, and math operators. String expressions can be concatenations of any number of character record fields and value objects.

Note: On DB2 UDB for OS/390 and z/OS, you should compose your expressions so that multiplication occurs before division; otherwise, decimal precision may be affected as values may be truncated. Use parentheses where necessary to control the order of calculation to ensure correct decimal precision.

This topic discusses how to define expressions.

Page Used to Set Up Expressions

Page Name	Definition Name	Navigation	Usage
Expression	PF_EXPR_DEFN	EPM Foundation, Business Metadata, Constraint and Expressions, Expression	Define an expression.

Expression Page

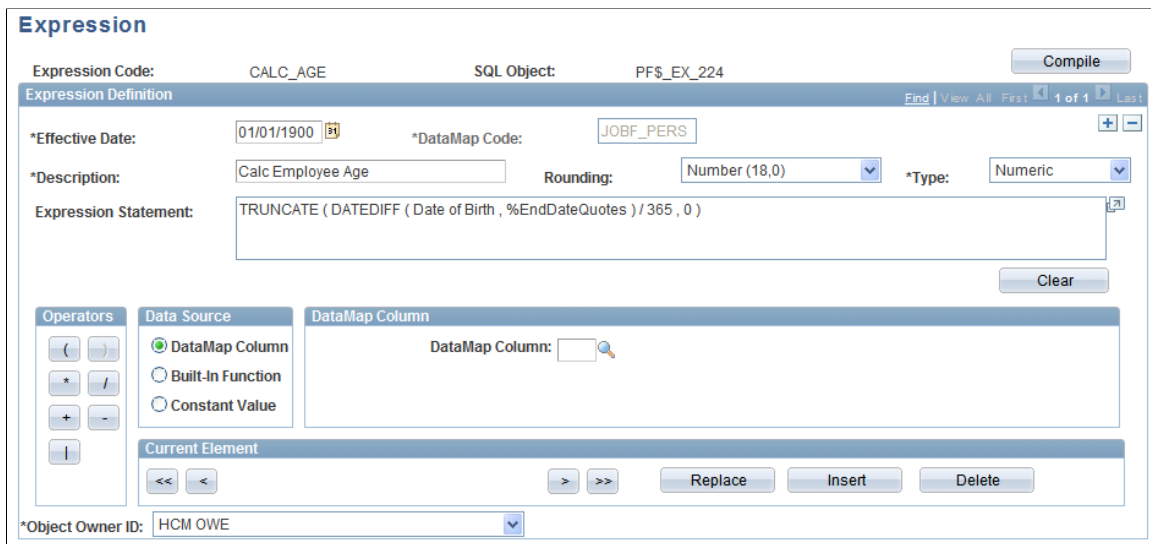
Use the Expression page (PF_EXPR_DEFN) to define an expression.

Navigation

EPM Foundation, Business Metadata, Constraint and Expressions, Expression

Image: Expression page

This example illustrates the fields and controls on the Expression page. You can find definitions for the fields and controls later on this page.



DataMap Code

Select the datamap that you want to build your expression on. You cannot change the datamap code after you have associated a datamap with an expression.

Rounding

For numeric expressions, the result of the expression is rounded based on the precision that is selected. For all the other types of expressions, the rounding factor is not applicable.

Type

Controls the set of operators, record fields, and value objects that can be selected. Values are:

Numeric: These expressions can combine any number of record fields, value objects, and math operators.

String: These expressions can be concatenations of any number of character record fields and value objects.

Date: These expressions can be a constant data or a date field that is derived from the datamap or built-in functions.

Expression Statement

Displays the expression that you build by selecting operators, fields, or value objects in the group boxes below. This is where you build your expression logic. You cannot type directly in the Expression Statement field.

First, select a data source. The data source that you select presents you with different options below it. To add an element to the expression statement, select it from the data source, and then click the Insert button. The system inserts your selection into the text area. Use the arrow buttons at the bottom to move in the text area. You can see what element is being specified by the arrows surrounding it, for example `>>Effective Date<<`.

Note: `>>Effective Date<<` is also displayed in the Current Element group box.

Operators

Select from the operators buttons to add the indicated value to the expression.

DataMap Column

Select to display a datamap column drop-down list box. Select the appropriate datamap column from the list and use the Insert button to place your selection in the text area of the page.

Built-In Function

Select to display a built-in function drop-down list box. Select the appropriate function from the list and use the Insert button to place your selection in the text area of the page. Built-in functions enable you to define numeric calculations, for example sum, average, or end of month, and minimum and maximum values.

Constant Value

Select to make available the constants options and, depending on your choice, a drop-down list box. The Ad Hoc option enables you to enter your own values for the character, numeric, and date field. The From List drop-down list box contains valid value objects. Use the Insert button to place your selection in the text area of the page.

Note: It is important to consider how an expression will be used when you want to specify date-related value objects. If the expression is to be used in a WHERE clause, then the `%CurrentDateIn` value object must be used. If the expression is to be used in a SELECT clause, then the `%CurrentDateOut` value object must be used.

Clear

Click to erase all of the text in the Expression Statement box.

First, Previous, Next, Last

Use these arrow buttons to move in the Expression Statement area of the page and identify the text that you want to replace, insert, or delete. When you click either `>` or `<`, you move one element forward or backwards. If you use the double arrows `>>` or `<<`, you move either to the beginning of the expression text or

to the end. You can see what element is being specified by the arrows surrounding it, for example >>*Effective Date*<<.

Replace

Use the arrow buttons to specify the element that you want to replace, select another element, and then click Replace.

Insert

After you have selected a data source, use this button to insert it into the Expression Statement text area. The system inserts the element at the position designated by the work area, pushing everything else to the back.

Delete

Use the arrow buttons to specify the element that you want to delete, and then click Delete.

When you have completed your expression, click the Compile button to compile the expression.

SQL Functions That Are Available to the Expression Builder

The following table lists the SQL functions that are available to build expressions.

SQL Object ID	Expression Function	Return Value	Description
PF_FUNCLIB_DATE_ DAYADD_UPD	AddtoDays (Date, Integer)	Date	Increase date by adding days (Integer).
PF_FUNCLIB_DATE_ DAYDIFF_UPD	DiffDates (Date, Date)	Integer	Calculate difference between two dates.
PF_FUNCLIB_DATE_DAY_ UPD	GetDay(Date)	Integer	Returns numeric day of the month from date.
PF_FUNCLIB_DATE_ MONADD_UPD	AddtoMonth(Date, Integer)	Date	Increase date by adding months (Integer).
PF_FUNCLIB_DATE_ MONBEG_UPD	BOM(Date)	Date	Returns the date value for the beginning of the month.
PF_FUNCLIB_DATE_ MONTH_UPD	GetMonth(Date)	Integer	Returns numeric month from date.
PF_FUNCLIB_DATE_ YEARBEG_UPD	BOY(Date)	Date	Returns date for the beginning of the year.
PF_FUNCLIB_DATE_ YEAREND_UPD	EOY(Date)	Date	Returns date for the end of the year.
PF_FUNCLIB_DATE_YEAR_ _UPD	GetYear(Date)	Integer	Returns numeric year from date.
PF_FUNCLIB_DATE_ YYMMDD_UPD	YYMMDD(Date)	Character	Formats date YYMMDD.

SQL Object ID	Expression Function	Return Value	Description
PF_FUNCLIB_DIFF_H(DTTM,DTTM)	DTTMDIFF_H(Date, Date)	Integer	Date time difference in hours.
PF_FUNCLIB_DTTM_DTTMDIFF_M_UPD	DTTMDIFF_M(Date, Date)	Integer	Date time difference in minutes.
PF_FUNCLIB_DIFF_S(DTTM,DTTM)	DTTMDIFF_S(Date, Date)	Integer	Date time difference in seconds.
PF_FUNCLIB_MATH_ABS_UPD	ABS(Integer)	Integer	Absolute value.
PF_FUNCLIB_MATH_MOD_UPD	MOD(Integer, Integer)	Integer	Modulus.
PF_FUNCLIB_MATH_TO_NUMB_UPD	TO_NUM(Character)	Integer	Convert to number.
PF_FUNCLIB_RTRIM	RTRIM(Character)	Character	RTrim blanks.
PF_FUNCLIB_TO_CHAR_UPD	TO_CHAR(Integer)	Character	Convert to character.

Note: All expression functions are operating system independent and database dependent.

Using Data Sets

Data sets are used as input for various engines, for instance, the Forecasting engine, user-defined functions, drivers in PeopleSoft Activity-Based Management (PeopleSoft ABM), and data elements in PeopleSoft KPI Manager. Data sets provide a user-defined set of information to the engines. Data sets restrict used columns and returned rows using constraints.

When you create a data element in PeopleSoft KPI Manager, for example, you are actually creating a data set. Though each data set is created by a process-specific setup, the underlying logic is the same, enabling you to more easily understand the functional aspects of the process.

Note: You will most likely not need to create a data set using the DataSet page because data sets are created behind the scenes. However, you may have to rebuild or recompile a data set if you change an underlying table. Data sets that are created in PeopleSoft EPM analytical applications do not appear on this page.

This topic provides an overview of data sets and discusses how to recompile or rebuild data sets.

Page Used to Use Data Sets

Page Name	Definition Name	Navigation	Usage
DataSet	PF_DATASET_DEFN	EPM Foundation, Foundation Metadata, Metadata Creation and Editing, DataSet	Recompile or rebuild a data set.

DataSet Page

Use the DataSet page (PF_DATASET_DEFN) to recompile or rebuild a data set.

Navigation

EPM Foundation, Foundation Metadata, Metadata Creation and Editing, DataSet

Image: DataSet page

This example illustrates the fields and controls on the DataSet page. You can find definitions for the fields and controls later on this page.

DataSet

SetID: SHARE DataSet Code: FIPRODRATINGS SQL OBJ: PFS_DS_47 Compile

Data Set End | View All | First | 1 of 1 | Last

*Effective Date: 01/01/1900 BT *Status: Active ▼ + -

*Description: Product Ratings

*Constraint Code: FIPRODRATINGS 🔍 [View Constraint](#) Reload Defn.

Used By: Risk Weighted Capital ▼ *Object Owner ID: FSI Profitability ▼

Data Set Fields Customize | End | 📄 | 📄 | First | 1-11 of 11 | Last

Description	Select	Aggregate Type
SetID	<input type="checkbox"/>	
Product ID	<input checked="" type="checkbox"/>	▼
New Model ID	<input type="checkbox"/>	
Effective Date	<input type="checkbox"/>	
Status as of Effective Date	<input type="checkbox"/>	
Description	<input type="checkbox"/>	
Currency Code	<input type="checkbox"/>	
Workout Costs	<input checked="" type="checkbox"/>	▼
Recovery Rate	<input checked="" type="checkbox"/>	▼
Severity Rating	<input checked="" type="checkbox"/>	▼
Correlation Coefficient	<input checked="" type="checkbox"/>	▼

Constraint Code

The constraint code for this data set. The selected constraint code populates the grid at the bottom of the page.

Select

Select this check box to indicate that this datamap column is included in the data set's select clause.

Aggregate Type

The aggregate type choices correspond to the standard SQL aggregate functions. If you select an aggregate type, all the other selected columns are populated with the group by default. Values are:

Avg (average): Returns the average of all the records retrieved.

Avg Distinct: Returns the average of all the unique records retrieved. For example, average (5, 8, 9, 12, 9, 7, 5) = 55/7 but average distinct (5, 8, 9, 12, 9, 7, 5) = 41/5.

Count: Returns the count of all the records retrieved.

Count Distinct: Returns the count of all the unique records retrieved. For example, count (A, B, A, C) = 4 but count distinct (A, B, A, C) = 3.

Group By: Groups the records retrieved by the measure field.

Min (minimum): Returns the minimum of all the records retrieved.

Max (maximum): Returns the maximum of all the records retrieved.

Sum: Returns the sum of all the records retrieved.

Sum Distinct: Returns the sum of all the unique records retrieved. For example, sum (1,1) = 2 but sum distinct (1,1) = 1.

Select the Compile button to compile the data set. This compilation sets the as of date of the data set to the effective date. If you want to change the as of date, you must recompile after updating the Effective Date field.

Use the Mass Validate utility to validate against as of dates.

Note: If a data set is sent to Resolver and has an inactive constraint, the system returns an error message.

PF_FETCH

PF_FETCH is an application engine process that is used by PeopleSoft EPM analytical applications to retrieve and display data by running the SQL behind the data set for a set of run control parameters after the data set is built (SQL is generated). PF_FETCH selects and inserts data into an output table that is specified by the application. Output table data is automatically deleted if it is more than two days old based on the date and time stamp.

Setting Up Filters

This topic provides an overview of filters and discusses how to:

- Define filters.
- Specify filter selection criteria.
- Use the tree viewer.

Pages Used to Set Up Filters

<i>Page Name</i>	<i>Definition Name</i>	<i>Navigation</i>	<i>Usage</i>
Filter	PF_FILTER_DEFN	EPM Foundation, Business Metadata, Constraint and Expressions, Filter, Filter	Define a filter and specify the datamap.
Filter - Selection Criteria	PF_FILTER_SEQ	EPM Foundation, Business Metadata, Constraint and Expressions, Filter, Selection Criteria	Define selection criteria for business rules.
Tree Viewer	PSTREEVIEWER	Click the Tree button on the Filter - Selection Criteria page. This appears only if you select an operation involving trees.	View all tree nodes. Select a node to be displayed in the value field on the Selection Criteria page by double-clicking the node.

Understanding Filters

Filters are used extensively by the PeopleSoft EPM engines to define subsets of data to perform operations or calculations on. In your datamap, you define which columns from the tablemap to use in processing. Filters enable you to specify which rows to use from those columns, similar to a WHERE clause in a query.

In general, the PeopleSoft EPM engines use set-based processing to process large amounts of data as efficiently as possible. Filters enable you to define what subset of data gets processed by or uses a specific business rule. Not every row of data may be necessary to process your data. Filters enable you to select only those rows you want.

You can create multiple filters based on a single datamap. In addition, your filters, and thus your business rules, can be different for each SetID, making the enrichment engines behave differently for the same rules in a different business unit.

Filter Page

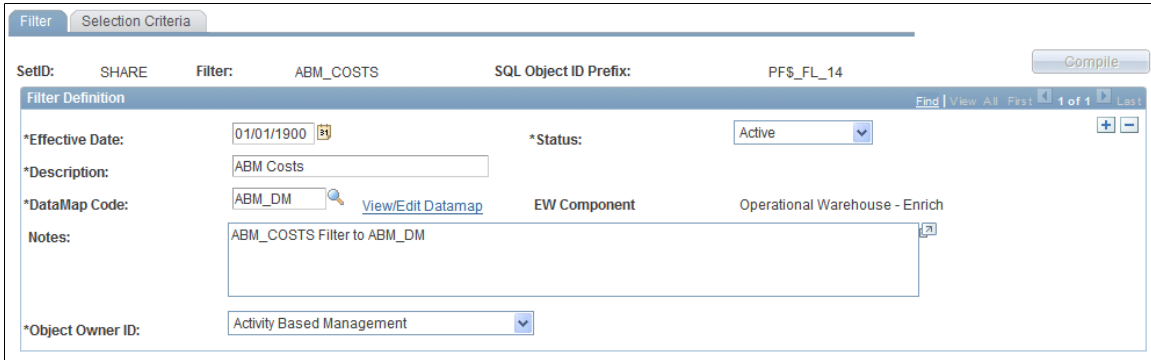
Use the Filter page (PF_FILTER_DEFN) to define a filter and specify the datamap.

Navigation

EPM Foundation, Business Metadata, Constraint and Expressions, Filter, Filter

Image: Filter page

This example illustrates the fields and controls on the Filter page. You can find definitions for the fields and controls later on this page.



DataMap Code

Select the datamap for which you want to define filters.

View/Edit Datamap

Click to transfer directly to the DataMap page for the selected DataMap to review it or make changes.

Notes

Enter any notes to further describe the filter.

When you have finished defining your filter, click the Compile button to compile the filter. This compilation sets the as of date of the filter equal to the effective date. If you want to change the as of date, you must recompile after updating the Effective Date field. Use the Mass Validate utility to validate against as of dates

Filter - Selection Criteria Page

Use the Filter - Selection Criteria page (PF_FILTER_SEQ) to define selection criteria for business rules.

Navigation

EPM Foundation, Business Metadata, Constraint and Expressions, Filter, Selection Criteria

Image: Selection Criteria page

This example illustrates the fields and controls on the Selection Criteria page. You can find definitions for the fields and controls later on this page.

Open	DataMap Column	*Oper	Value	Close	And/Or	Aggr
<input type="checkbox"/>	Cost Object Use	=	1	<input type="checkbox"/>	AND	<input type="checkbox"/>
<input type="checkbox"/>	Drill Level	=	1	<input type="checkbox"/>	AND	<input type="checkbox"/>
<input type="checkbox"/>	Cost Object Use	=	1	<input type="checkbox"/>	AND	<input type="checkbox"/>
<input type="checkbox"/>	Object Type To	=	%CostObjectType	<input type="checkbox"/>	AND	<input type="checkbox"/>
<input type="checkbox"/>	Tree Target Object	InTree		<input type="checkbox"/>		<input type="checkbox"/>

On this page, you can add or delete actual data values that make up your filter. You can have multiple filter rules.

Open

Select the number of left parentheses needed for the selection criteria. You may have multiple parentheses in a case such as (X or Y) and Z).

DataMap Column

Select the columns from the datamap for which you want to define filter criteria.

Oper (operation)

Select the operation to be used as selection criteria. The current operations available are *equal to*, *greater than*, *less than*, *greater than or equal to*, *less than or equal to*, *not equal to*, *In Tree*, *Not in Tree*, *In MetaTree*, and *NotInMetaTree*.

Like is valid only for character type columns that have no associated lookup tables in the datamap. The value column for this operator must have a pattern search wild card, for example % or _.

If you specify an operation of *In Tree*, click the Tree button that appears alongside the Oper field to specify the tree name and tree node that you want to use as selection criteria. You can search in the Lookup Tree ID page for your tree if it is not displayed. If you still can't find the tree, verify that when you added the tree in the Tree Manager, you also created a record in the Tree Metadata page. This record identifies the engines where the flattened tree data is stored. On selecting the tree name, other information that is related to the tree is retrieved and then sent to the tree viewer (PSTREEVIEWERWRK). All the nodes

can be seen here. The node you select here will be displayed in the Value field.

Obj (object)

Select this check box to select from a list of predefined value objects for the value setting. The Obj check box appears if you select any operand other than the tree choices.

Filter Tree

Select this to access the Tree Viewer page and view the tree nodes associated with the tree.

This button only appears if you select one of the tree options for the Operation field.

Value

Select the value that the column is to be evaluated against. If you've defined a lookup table for the column in the datamap, you are prompted to select from a list of valid values.

Close

Select the number of right parentheses that are needed for the selection criteria.

And/Or

Select either an *AND* or an *OR* join for the filter criteria.

Aggr (aggregate)

Select to aggregate the values.

Note: Remember that the Resolver resolves the following fields, so you do not have to include them as selection criteria: SetID, business unit, scenario ID, effective date, as of date, fiscal year, and period.

Tree Viewer Page

Use the Tree Viewer page (PSTREEVIEWER) to view all tree nodes. Select a node to be displayed in the value field on the Selection Criteria page by double-clicking the node.

Navigation

Click the Tree button on the Filter - Selection Criteria page. This appears only if you select an operation involving trees.

Image: Tree Viewer page

This example illustrates the fields and controls on the Tree Viewer page.



You can expand all levels of the tree to view detailed information. Double-click the folder button to expand specific levels. Select the node of the tree that you want to use and click the Select button to return to the Filter - Selection Criteria page. The value field is populated with the node that you selected.

Setting Up Constraints

Constraints can be made up of one or more filters linked together using AND, OR, and NOT logic. This means that you can keep your filters simple and dedicated to a single purpose, and link them together in constraints to form complex business logic.

Constraints enable you to define business rules for processing and also enable you to create and reuse filters.

This topic discusses how to:

- Define constraints.
- Specify constraint criteria.
- Specify constraint details.

Pages Used to Set Up Constraints

Page Name	Definition Name	Navigation	Usage
Constraint	PF_CONSTRAINT_DEFN	EPM Foundation, Business Metadata, Constraint and Expressions, Constraint, Constraint	Define a constraint.
Constraint - Criteria	PF_CONSTRAINT_SEQ	EPM Foundation, Business Metadata, Constraint and Expressions, Constraint, Criteria	Specify your constraint criteria. This is where you can combine filters to create complex business logic.
Constraint - Details	PF_FILTER_EXPLODE	EPM Foundation, Business Metadata, Constraint and Expressions, Constraint, Details	Display the filter contents of a given constraint. Modify the filter, as needed.

Constraint Page

Use the Constraint page (PF_CONSTRAINT_DEFN) to define a constraint.

Navigation

EPM Foundation, Business Metadata, Constraint and Expressions, Constraint, Constraint

Image: Constraint page

This example illustrates the fields and controls on the Constraint page. You can find definitions for the fields and controls later on this page.

Save As

Click to replicate existing constraint metadata. You are prompted to enter the new constraint code name and effective date. This is available only in correction mode.

DataMap Code

Select the datamap for the constraint.

View/Edit Datamap

Click to transfer directly to the DataMap setup page where, you can view and edit the selected datamap.

No Criteria

You are not required to link filters to your constraint. By selecting this check box, you indicate to the system that you want all the values from the datamap. The system adds a 0=0 WHERE clause to any other join criteria that the datamap needs (thus hiding the criteria page). The FROM clause is the same as the from field on the datamap.

Notes

Enter any notes to further describe the constraint.

When you have completed setup of the constraint, click the Build Constraint SQL button to compile the constraint. This compilation sets the as of date of the constraint equal to the effective date. If you want to change the as of date, you must recompile after updating the Effective Date field. Use the Mass Validate utility to validate against as of dates.

Constraint - Criteria Page

Use the Constraint - Criteria page (PF_CONSTRAINT_SEQ) to specify your constraint criteria. This is where you can combine filters to create complex business logic.

Navigation

EPM Foundation, Business Metadata, Constraint and Expressions, Constraint, Criteria

Image: Criteria page

This example illustrates the fields and controls on the Criteria page. You can find definitions for the fields and controls later on this page.

Use this page to add or delete the filters that you want to make up the constraint. You can link one or more filter codes. This page is not accessible if you select the No Criteria check box on the Constraint page.

Not

Select to signify the negative of the condition.

Open

Select the number of left parentheses that are needed for the selection criteria. You may have multiple parentheses in a case such as ((X or Y) and Z).

Filter Code

Select the filter to use. You define filters using the Filter component.

Close Select the number of right parentheses needed.

And/Or Select either *AND* or *OR* to relate one line to the next of the filter criteria.

Three filter types are available: *WHERE*, *HAVING*, and a combination of the two. Filter types relate lines of constraint rules.

Note: After a filter is created, the filter type cannot be changed.

Constraint - Details Page

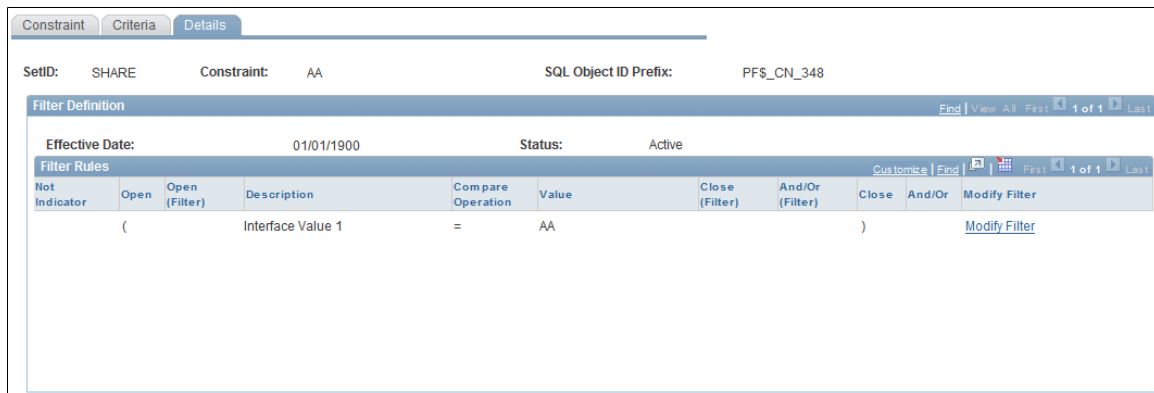
Use the Constraint - Details page (PF_FILTER_EXPLODE) to display the filter contents of a given constraint. Modify the filter, as needed.

Navigation

EPM Foundation, Business Metadata, Constraint and Expressions, Constraint, Details

Image: Details page

This example illustrates the fields and controls on the Details page.



Here you can view the data elements that are targeted based on the filter selection criteria.

Modify Filter Click to access the Filter setup page, where you can modify the filter.

Setting Up Metric Metadata

You can create metric metadata to process specific groups of your transactional data, based on columns in your record tables. These metrics can help you track essential measures for your organization, such as total sales and revenue. PeopleSoft does not deliver metrics. You can, however, create your own metrics using the Define Metric page. The page also enables you to define security for your metric.

See [Defining Dimension and Metric Security](#).

Page Used to Set Up Metric Metadata

Page Name	Definition Name	Navigation	Usage
Define Metric	PF_SY_METR_DEFN	EPM Foundation, EPM Security, Metrics and Dimensions, Create/Edit Secured Metrics, Define Metric	Define and secure metrics.

Define Metric Page

Use the Define Metric page (PF_SY_METR_DEFN) to define and secure metrics.

Navigation

EPM Foundation, EPM Security, Metrics and Dimensions, Create/Edit Secured Metrics, Define Metric

Image: Define Metric page

This example illustrates the fields and controls on the Define Metric page. You can find definitions for the fields and controls later on this page.

The screenshot shows the 'Define Metric' page with the following fields and values:

- Metric ID:** TEST
- Metric Definition:** (Section header)
- *Description:** tt
- *Record Name:** AB_ACTASGN_F00
- *Column Name:** ABC_RES_ID
- Is Secured:** Is Secured

Record Name Enter the name of the record that is associated with the metric that you are defining.

Column Name Select a column that is associated with the metric that you are defining.

Is Secured Select this check box to indicate that the metric is secured.

Working with Record Summary Metadata

Record summary metadata specifies the TSE views and pages, as well as the flash total fields that are associated with a table. Record summary metadata is delivered for fact tables. If you change the record structure of a table that uses record summary metadata, or change the totals to be summarized, you must click the Rebuild button to regenerate the SQL. This process is associated with Profit Manager.

See [Understanding Profit Manager Tools](#).

Page Used to Work With Record Summary Metadata

Page Name	Definition Name	Navigation	Usage
Record Summary	PF_SUMM_REC_TBL1	EPM Foundation, Foundation Metadata, Metadata Creation and Editing, Record Summary	Create new record summary metadata or rebuild delivered record summary metadata.
PF Record Summary	RUN_PF_SUMM	EPM Foundation, Data Enrichment Tools, Profit Manager, Record Summary, Summarize Error Statistics, PF Record Summary	Run the PF Record Summary engine before you process any fact table modifications.

Record Summary Page

Use the Record Summary page (PF_SUMM_REC_TBL1) to create new record summary metadata or rebuild delivered record summary metadata.

Navigation

EPM Foundation, Foundation Metadata, Metadata Creation and Editing, Record Summary

Image: Record Summary page

This example illustrates the fields and controls on the Record Summary page. You can find definitions for the fields and controls later on this page.

Record Summary

Table Name: BP_LED_F00 General Ledger SQL Object: PF\$_RS_BLED Compile

Record Summary Information

*Record Abbreviation: Apply Fact Error Correction

Error Table: BP_LED_E00 General Ledger

TSE Table: TSE_BLED_FLD Ledger Table Error

*TSE Table's View1:

*TSE Table's View2:

*Flash Total Field 1:

Flash Total Field 2:

Flash Total Field 3:

SQL Object Prefix ID

A system-generated number that identifies the location of the SQL that is built by this component in the SQL Repository.

Record Abbreviation

A naming convention that populates all other fields on the page. Enter up to four characters. By defining the record abbreviation,

the system makes a best guess and automatically populates all the fields on the rest of the page.

Apply Fact Error Correction

Select to apply fact error correction to this table.

Error Table and TSE Table

Are display-only and based on the error and TSE table definitions that were made in the record metadata for the selected table.

The following fields are automatically populated based on the record abbreviation. These objects are used for PF Modification:

TSE Table's View 1

The work record that is used in the PF Summary process.

TSE Table's View 2

The work record that is used by PF Edi.

Flash Total Field 1, 2, 3

Select the totals that you want to summarize. These totals are defined in the job totals metadata. These are monetary amounts that you want to track.

You can now run the PF Record Summary engine.

Note: Record Summary metadata must be defined before you run the PF Record Summary engine. You must run the PF Record Summary engine to run PF Modification.

Related Links

[Job Total Metadata Page](#)

PF Record Summary Page

Use the PF Record Summary page (RUN_PF_SUMM) to run the PF Record Summary engine before you process any fact table modifications.

Navigation

EPM Foundation, Data Enrichment Tools, Profit Manager, Record Summary, Summarize Error Statistics, PF Record Summary

Image: PF Record Summary page

This example illustrates the fields and controls on the PF Record Summary page. You can find definitions for the fields and controls later on this page.

The screenshot shows the 'PF Record Summary' configuration page. At the top, there are links for 'Report Manager' and 'Process Monitor', and a 'Run' button. The form contains the following fields and controls:

- User ID: VP1
- Run Control ID: DVP1
- Program Name: PF_SUMM
- When: Once (dropdown menu)
- As Of Dated Jobstream
- *Description: (text input field)
- *Business Unit: CORP1 (lookup field)
- Scenario ID: ACTUAL01 (lookup field)
- Fiscal Year: 2009 (text input field)
- Period: 1 (text input field)
- *Job ID: SUMM (lookup field) with label 'PF Summary'
- *Record: LEDGER_F00 (lookup field)

Description

The description is important because it is used by the Metadata Search engine to locate your metadata.

As Of Dated Jobstream

Select to replace the fiscal year and period with the as of date field for the engine run.

Business Unit

Select the appropriate business unit.

Scenario ID

Select the appropriate scenario.

Fiscal Year

Select the appropriate fiscal year. This field does not display for an as of dated jobstream.

Period

Select the appropriate period. This field does not display for an as of dated jobstream.

As Of Date

If you selected the As of Dated Jobstream check box, enter the date.

Job ID

Select the job ID of *Summ*. Job IDs are set up in job metadata.

Record

Select the record.

PF_SUMM summarizes error statistics by field and value for the PF Modification process. The record summary metadata uses record metadata to determine which records can be summarized.

Setting Up Report Metadata

Report metadata is delivered for PeopleSoft Analytical Application reports. Each delivered report or cube has a record within the Report Metadata page. You can change the metadata as reports change, or as additional parameters are needed. If you create your own reports, you can use report metadata to document them.

This topic discusses how to define report details.

Pages Used to Set Up Report Metadata

<i>Page Name</i>	<i>Definition Name</i>	<i>Navigation</i>	<i>Usage</i>
Report Details	PF_META_RPT_TBL1	EPM Foundation, Foundation Metadata, Metadata Creation and Editing, Report Metadata, Report Details	Define report details, including the report name and type. Report metadata is not required. Rather, it is provided for your reference.
Tables Used	PF_META_RPT_TBL2	EPM Foundation, Foundation Metadata, Metadata Creation and Editing, Report Metadata, Tables Used	Display the tables that are used to generate the report.

Report Details Page

Use the Report Details page (PF_META_RPT_TBL1) to define report details, including the report name and type. Report metadata is not required. Rather, it is provided for your reference.

Navigation

EPM Foundation, Foundation Metadata, Metadata Creation and Editing, Report Metadata, Report Details

Image: Report Details page

This example illustrates the fields and controls on the Report Details page. You can find definitions for the fields and controls later on this page.

The screenshot shows the 'Report Details' page with the following fields and values:

- Report ID:** BIALM22
- *Effective Date:** 01/01/1900
- *Status:** Active
- *Description:** Measures Summary
- *Report Type:** ROLAP Report
- *Product:** Asset Liability Management
- URL:** (empty)
- Cube Instance ID:** (empty)
- Notes:** Provides a summary of the various measures calculated by the Cash Flow engine.

Report Type

The type of report.

Product

The associated product for the report.

URL

Use to specify either a URL or the directory where the generated reports are stored.

Cube Instance ID

Select the ID assigned by PeopleTools Cube Builder that is used for reporting.

This field is active when you select *Cognos Cube* as the report type.

Notes

Enter any notes to further describe the report.

Click the Tables Used tab to review the tables that are used to create the selected report.

Cloning Metadata

Cloning metadata is an efficient and quick way to create a duplicate copy of existing or already defined metadata.

This topic discusses how to define metadata clone criteria.

Page Used to Clone Metadata

Page Name	Definition Name	Navigation	Usage
Metadata Clone	PF_MD_CLONE	EPM Foundation, Foundation Metadata, Other Metadata Operations, Metadata Clone	Define metadata clone criteria for copying metadata from one SetID to another.

Metadata Clone Page

Use the Metadata Clone page (PF_MD_CLONE) to define metadata clone criteria for copying metadata from one SetID to another.

Navigation

EPM Foundation, Foundation Metadata, Other Metadata Operations, Metadata Clone

Image: Metadata Clone page

This example illustrates the fields and controls on the Metadata Clone page. You can find definitions for the fields and controls later on this page.

SetID

The SetID from which you want to clone the metadata.

Clone SetID

The SetID for which you want to create the clone.

Clone Metadata Type

Select the type of metadata that you want to clone. Values are: *AL Rule* (allocation manager rule), *DM Rule* (data manager rule), *Constraint*, *DataSet*, or *Filter*.

Select on of the following operations: =, <>, *Like*, or *Not Like*.

Enter a metadata value to search on.



Click the Get Selected Metadata button. Your results will appear in the Metadata Selection List field.

Clone

Select this check box to return all metadata objects that are based on your selection criteria. Deselect the check box for any metadata that you do not want to clone.

Note: Remember that you must compile all cloned metadata. You can do so using the Mass Compile utility.

Related Links

[Running Mass Compile](#)

Using the Metadata Mover Utility

This topic provides an overview of the Metadata Mover utility and discusses how to:

- Specify metadata export criteria.
- Export metadata from a source to a target database.
- Import metadata into the target database.
- Build imported metadata in the target database.

Pages Used to Migrate Metadata

<i>Page Name</i>	<i>Definition Name</i>	<i>Navigation</i>	<i>Usage</i>
Metadata Export - Export Definition	PF_MD_EXPORT	EPM Foundation, Foundation Metadata, Other Metadata Operations, Metadata Export, Export Definition	Specify metadata export criteria.
Metadata Export - Export List	PF_MD_EXPLST	EPM Foundation, Foundation Metadata, Other Metadata Operations, Metadata Export, Export List Click the Generate Export List button on the Export Definition page.	View a list of metadata objects that you selected on the Export Definition page. These objects will be exported to your target database. The list includes the following information for each metadata object: metadata type, SetID value, metadata object code, and related objects.

Understanding the Metadata Mover Utility

The Metadata Mover utility enables you to migrate EPM metadata from one database to another. Specifically, the Metadata Mover utility moves record metadata, tablemaps, datamaps, expressions, filters, constraints, data sets, engine metadata, jobs, jobstreams, report metadata, metric metadata, tree metadata, value objects rule metadata, Data Manager rules, Data Manager rule sets, Allocation Manager rules, Allocation Manager rule sets, Data Mapper rule groups, and Data Mapper rule sets.

The Metadata Mover utility enables metadata to be moved between PeopleSoft 8.8x environments. For example, you cannot move metadata from a PeopleSoft 8.8x environment to a PeopleSoft 8.0x environment. Additionally, to successfully move metadata between PeopleSoft 8.8x EPM environments, the source and target databases must be the same type. For example, if you have a Microsoft database,

you must migrate to another Microsoft database. You can't migrate from a Microsoft database to an Oracle database.

The Metadata Mover utility maintains SQL counters for the target database. The source SQL counter will be lost when the move is completed so that uniqueness is ensured within the target database.

Follow these steps to move EPM metadata from your source database to your target database:

1. Sign in to your source database.
2. Select the metadata that you plan to export from your source database using the Metadata Export page.
3. Use the Data Mover script to export the objects from the source database.
4. Use the Data Mover script to import the objects into the target database.
5. Use the Mass Compile and Audit processes in the target database to build and validate the migrated metadata objects.

Metadata Export - Export Definition Page

Use the Metadata Export - Export Definition page (PF_MD_EXPORT) to specify metadata export criteria.

Navigation

EPM Foundation, Foundation Metadata, Other Metadata Operations, Metadata Export, Export Definition

Image: Export Definition page

This example illustrates the fields and controls on the Export Definition page. You can find definitions for the fields and controls later on this page.

*Metadata Object Type	*MetaData Object Code	Fetch Related Objects
1 Record	ABPS_LEDGER_F00	<input checked="" type="checkbox"/>

Note: You can maintain many export IDs, although you can have only one metadata export list. This means that when you generate the metadata export list, the system replaces the existing list with a new list.

In the Metadata Selection List group box, select:

Metadata Object Type and Metadata Object Code Select each metadata object type and code that you want to export to your target database. Remember that you must select metadata dependencies (if the metadata has dependencies).

Fetch Related Objects Select this check box to have the system fetch all objects that are related to or dependent on the object that you have selected and add them to the metadata export list.

Generate Export List Click to have the system generate the list of objects to be exported. You can view them on the Metadata Export - Export List page.

For example, if you select a datamap as the object type and select the Fetch Related Objects check box, the system adds all related expressions, filters, constraints, data sets, Data Manger rules, Allocation Manager rules, and value objects to the metadata export list.

The following table shows which metadata objects are dependent upon each other.

Metadata Type	Related Objects
Allocation Manager rules	Meta Values.
Allocation Manager rule set	Allocation Manager Rules and Meta Values.
Data Manager rules	Meta Values.
Data set	Option is not available.
Data Manager rule set	Data Manager rules and meta values.
Data Mapper rule set	Set of data mapper rules.
Data Mapper rule group	Group of Data Mapper rules.
Constraint	Data sets, Allocation Manager rules, Allocation Manager meta values, Data Manager rules, and Data Mart meta values.
Filter	Constraints and related objects.
Datamap	Expression, filter, and related objects.
Tablemap	Datamap and related objects.
Record	Tablemap and related objects.
Jobstream	Option is not available.
Job	Jobstream.
Engine	Job and jobstream.
Tree, metric, report, and rule metadata, expressions, Data Loader and data mart data	Option is not available.

Review the metadata export list and if you are sure that this is the data that you want to export, proceed to the next step.

Note: Data Mapper is documented in the *PeopleSoft EPM: Global Consolidations*.

See "Mapping to a Common Chart of Accounts (Data Mapper) (*PeopleSoft EPM 9.1: Global Consolidations*)".

Exporting Metadata from a Source Database to a Target Database

After selecting the metadata that you want to export from your source database to your target database (using the Export Definition page), perform the export. To perform the export, follow these steps:

1. Sign in to Data Mover in the metadata source database.
2. Open PFMMEXP.dms in the Data Mover tool.
3. Edit the path name for the output files (you can select which path you want).
4. Run the script.

The script exports the metadata tables into a DAT file for the export list that is generated through the Export Definition page.

5. Sign out of the source database.

You now need to import the metadata that you just exported.

Importing Metadata into the Target Database

Now that you have exported the metadata from the source database, you must import the metadata into the target database in the following way:

1. Sign in to the target database.
2. Open PFMMIMP.dms in the Data Mover tool.
3. Edit the path of both the input files to point to the output directory of the exported files.
4. Run the script.

The script exports the metadata tables from the .DAT files.

Building Imported Metadata in the Target Database

After you've finished importing the metadata into your target database, you must build the new metadata. To build the new imported metadata, you must first recompile it. If you imported a small number of new metadata objects, you can go to each metadata object's page and click the Compile button.

Note: Now there is no SQL object counter on the pages. Clicking the Compile button generates the counter.

If you have imported a large quantity of metadata objects, run the Mass Compile process followed by the Audit process.

Correct any errors that were identified during the Audit process.

Rerun the audit until it is clean.

Related Links

[Auditing PeopleSoft EPM Objects](#)

[Running Mass Compile](#)

Deleting Metadata

The Metadata Delete process enables you to delete records, tablemaps, datamaps, and rule metadata.

Page Used to Delete Metadata

Page Name	Definition Name	Navigation	Usage
Metadata Delete	PF_METADATA_DEL	EPM Foundation, Foundation Metadata, Other Metadata Operations, Delete Metadata, Metadata Delete	Delete metadata.

Metadata Delete Page

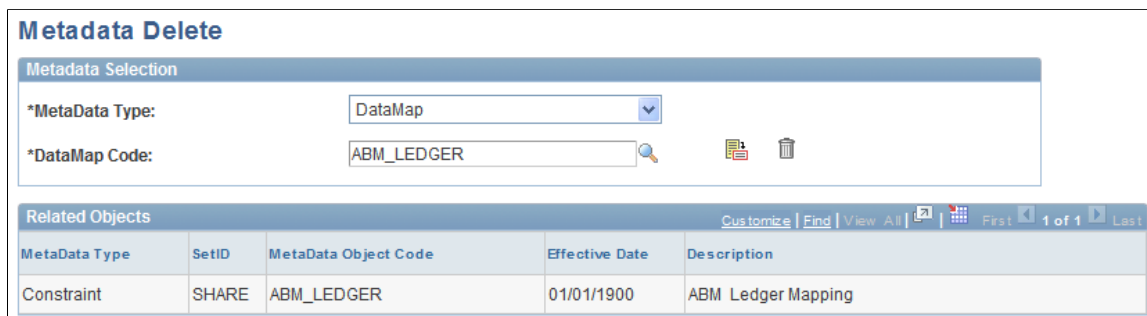
Use the Metadata Delete page (PF_METADATA_DEL) to delete metadata.

Navigation

EPM Foundation, Foundation Metadata, Other Metadata Operations, Delete Metadata, Metadata Delete

Image: Metadata Delete page

This example illustrates the fields and controls on the Metadata Delete page. You can find definitions for the fields and controls later on this page.



Metadata Type Select the type of metadata that you want to delete.

Metadata Object Code Based on the metadata type that you have selected, select the metadata object from the valid values that are available. For

example, if you select datamap as the metadata type, you will be able to select from the available datamap objects.



Click the Fetch Related Objects button to populate the grid with the related metadata objects.



Click the Delete All Objects button to delete the object code and all the related objects.

Creating User-Defined Functions

User-defined functions enable you to define functions one time through a common interface, then use them throughout many of the Analytical Applications. The options that are available to you when defining functions are based on predefined modules that are provided with your system.

Refer to the *PeopleSoft Application Fundamentals for Financial Services Industry Applications* for details on working with user-defined functions.

Related Links

"Understanding User-Defined Functions (*PeopleSoft EPM 9.1: Applications Fundamentals for Financial Services Industry*)"

OWE Dimension Correction Pages

Organizational change is both necessary and inevitable. Your organization will restructure departments, hire new department managers, develop new products, and obtain new customers through a variety of sales channels. In each case, you must capture these business rules in EPM.

Although the prepackaged ETL jobs are designed to automatically capture these changes to your organization, you might encounter times when you are required to add or update dimensions directly in the PeopleSoft pages. You can use the OWE Dimension Maintenance menu items for this purpose (EPM Foundation, Business Metadata, OWE Dimension Maintenance).

This menu includes a separate menu item for each functional area that corresponds to one or more dimensions from your PeopleSoft source data and contains the following categories:

- Common.
- CRM.
- Supply Chain Management.
- Human Resource Management System.

For example, under the Common menu there is a specific menu item to add or update common dimensions. The following is an example of one of the maintain dimension pages:

Image: Maintain Dimension page

This example illustrates the fields and controls on the Maintain Dimension page.

The screenshot shows a web-based form titled "Maintain Dimension". At the top, it displays "SetID: SHARE" and "Position Number: 00010101". Below this is a "Details" header with navigation options like "Find", "View All", "First", "1 of 1", and "Last". The form contains numerous fields:

- *Effective Date:** 12/31/1991
- *Status:** Active
- *Description:** General Manager
- *Position Status:** Approved
- Business Unit:** CORP1
- Reports To:** 00010101
- Dot-Line:** (empty)
- Action:** Posn Chg
- Company:** CCB
- Reg Region:** USA
- *Employee Type:** Salaried
- *Reg/Temp:** Regular
- *Full/Part:** Full-Time
- *Regular Shift:** N/A
- Pay Group:** (empty)
- Job Code:** 0101
- Holiday Schedule:** (empty)
- Max Head:** 1
- Union Code:** (empty)
- Department:** 00001
- Cnt:** (empty)
- Location:** 001
- Reason Code:** (empty)
- Default Hr:** 40.00
- Key Position
- Job Sharing Permitted
- Long Description:** General Manager

For each dimension, enter the relevant information for your dimension and save.

Warning! It is recommended that dimensions be accessible only to the EPM Administrator.

Note: Critical errors, such as missing SetIDs or business units, must be fixed on their respective setup pages. Typically, the types of errors you correct in the dimension pages are the non-key fields of an error record.

Chapter 17

Working with Metadata Utilities

Viewing Error Messages

You can view error messages generated by the engines described in this topic using the Engine Messages component. This component is described in the "Streamlining Processing with Jobstreams" topic of this documentation.

See [Viewing Engine Messages](#).

Auditing PeopleSoft EPM Objects

This topic provides an overview of the Audit utility and discusses how to:

- Run the Audit utility.
- Review metadata audit log error messages.

Page Used to Audit PeopleSoft EPM Objects

<i>Page Name</i>	<i>Definition Name</i>	<i>Navigation</i>	<i>Usage</i>
Audit EPM Objects	RUN_PF_AUDIT	EPM Foundation, Foundation Metadata, Other Metadata Operations, Audit EPM Objects	Run the Audit utility to identify setup errors.

Understanding the Audit Utility

The PeopleSoft EPM Audit utility is a common process used to identify setup errors in PeopleSoft EPM modules. For instance, you can use the Audit utility to check the validity and existence of metadata objects in PeopleSoft EPM application tables. The Audit utility verifies that your original metadata objects (tablemaps, datamaps, constraints, filters, trees, and so forth) are still intact. The Audit utility also provides audit functionality for row-level security group objects. To remove constraints, value objects, SQL object IDs, and counter objects from the row level group tables, you must fix the security group or the missing object manually.

The Audit utility process does not need to be set up in a jobstream and can be run at any time. The processing time depends on how many selections you make. This utility can affect database performance times. To ensure a successful engine run, PeopleSoft suggests that you run an audit, review results using the Process Monitor and Messages inquiry page, and fix any problems before running a jobstream. Any problems identified by the Audit utility may require assistance from a programmer or warehouse administrator.

Audit EPM Objects Page

Use the Audit EPM Objects page (RUN_PF_AUDIT) to run the Audit utility to identify setup errors.

Navigation

EPM Foundation, Foundation Metadata, Other Metadata Operations, Audit EPM Objects

Image: Audit EPM Objects page

This example illustrates the fields and controls on the Audit EPM Objects page.



Click the Select All button to run all of the engine options available.



Click the Clear All button to deselect all check boxes.

Performance Mgmt Warehouse (Performance Management Warehouse)

In this group box the audit options are: Metadata, KPI Manager, Allocation Manager Rule, Row Level Security, Data Manager Rule, and Technical Scenarios.

The metadata objects that are examined by the Audit utility include record metadata, tablemaps, datamaps, constraints, filters, expressions, data sets, engines, jobs, and jobstream rules.

If you select the Data Manager Rule check box, the Audit utility checks for inconsistencies between data manager rules and the metadata.

KPI Manager objects that are examined by the Audit include datamaps, data elements, dimensions, constraints, filters, calculation expressions, calculation cubes, and calculation IDs.

Financial Analytics

Here you can select to run the audit on Risk-Weighted Capital, Funds Transfer Pricing, or Support Modules.

Running the Audit Utility

Click the Run button to run this request.

After the audit has run, you can check the Process Monitor to verify the process ran smoothly. Use the Messages inquiry pages to review any errors found by the audit.

Note: You can run the Audit utility at any time. The processing time depends on how many selections you make and depending on your selections, the utility can affect database performance times.

Reviewing Metadata Audit Log Error Messages

All messages can be found and viewed by navigating to PeopleTools, Utilities, Administration, Message Catalog. EPM messages are numbered 9000-9999.

Performing Impact Analysis

Impact Analysis enables you to determine the effects that a change to one PeopleSoft EPM object may have on the other objects related to it. For instance, depending on the type of change you make to a warehouse table, you may have to change the tablemaps and datamaps built on top of the table. Impact Analysis provides a list of the objects that are related to the object that you want to change so that you can determine the impact of the change.

This topic discusses how to use Impact Analysis on PeopleSoft EPM objects.

Page Used to Perform Impact Analysis

<i>Page Name</i>	<i>Definition Name</i>	<i>Navigation</i>	<i>Usage</i>
Impact Analysis	PF_IMPACT_REQUES2	EPM Foundation, Foundation Metadata, Other Metadata Operations, Impact Analysis	Determine the relationships between data warehouse objects.

Impact Analysis Page

Use the Impact Analysis page (PF_IMPACT_REQUES2) to determine the relationships between data warehouse objects.

Navigation

EPM Foundation, Foundation Metadata, Other Metadata Operations, Impact Analysis

Image: Impact Analysis page

This example illustrates the fields and controls on the Impact Analysis page.

Impact Analysis

Parameters

*Object Type: ▼

TableMap: 🔍 📄

Long Name	Object Name	Description
Constraint	ABM_LEDGER	ABM Ledger Mapping
Data Manager Job Association	ABM_LEDG1	ABM Ledger DM
Data Manager Rule	ABMLEDG	ABM Ledger Data
Data Manager Rule Set	ABMLEDGRS	ABM Ledger
Data Warehouse Table	ABM_LEDGER_VW	View of Ledger Data
DataMap	ABM_LEDGER	ABM Ledger Mapping
Record	ABM_LEDGER_VW	View of Ledger Data

Object Type

Select the type of object you want to review. The field below changes based on the object type you select. For example, if you select *DataMap*, the field label changes to DataMap.

Enter the name of the object.



Click the Related Objects button to populate the grid with all the objects related to the object you specified.

You can download the list of objects using the download button at the top of the grid. Using this list, you decide whether a change is feasible. You can also use the list to update team members when something changes.

Note: This is a display-only page that provides a list of all related objects.

Running Mass Validate

The Mass Validate metadata utility enables you to validate, but not compile, metadata objects. Mass Validate will certify all as of dates created for filters, constraints, and data sets for the specified run date. This utility helps ensure your metadata is valid at run time and increases your chance of a successful engine run.

This topic discusses how to validate metadata objects.

Page Used to Run Mass Validate

Page Name	Definition Name	Navigation	Usage
Mass Validate	RUN_PF_VALIDATE	EPM Foundation, Foundation Metadata, Other Metadata Operations, Mass Validate	Validate metadata objects.

Mass Validate Page

Use the Mass Validate page (RUN_PF_VALIDATE) to validate metadata objects.

Navigation

EPM Foundation, Foundation Metadata, Other Metadata Operations, Mass Validate

Image: Mass Validate page

This example illustrates the fields and controls on the Mass Validate page. You can find definitions for the fields and controls later on this page.



Click the Select All button to run all of the engine options available.



Click the Clear All button to deselect all check boxes.

SetID

Select the SetID you want to run validate on.

As Of Date

Specify as of date for this run.

Only Imported Metadata

Select this check box to limit Mass Validate to only those objects which have their SQL counter field set to zero. Use this to validate any metadata imported into the database using the Metadata Migration utility. This metadata will not have a SQL ID when first imported. Also use it to validate any new metadata

that has not been compiled. First validate the metadata and then run Mass Compile to compile it.

You can either select all objects, or individually select the objects you want to validate in the Metadata Type, Performance Management Warehouse, KPI Manager, and Workforce Analytics group boxes.

Click Run to run the Mass Validate process.

Running Mass Compile

The Mass Compile metadata utility enables you to either compile individual metadata objects, or all metadata objects.

As discussed in the "Setting Up and Working with Metadata" topic of this documentation, if you change a table, you must recompile record metadata for that table. For instance, if you add a non-key column to a table, you must recompile the record metadata. If you add a key column, you must recompile both the record metadata and any tablemaps, datamaps, constraints, or other metadata objects associated with it. The advantage of Mass Compile is that you can opt to compile all metadata objects at once. In addition, if you have imported metadata into the database and validated it, you can compile it using Mass Compile.

Note: Mass Compile will not compile allocation manager rules that are period-based or have the multiple business unit option selected. These rules will be skipped and must be compiled from the Allocation Manager Rules component.

This topic discusses how to compile metadata objects using Mass Compile.

Page Used to Run Mass Compile

<i>Page Name</i>	<i>Definition Name</i>	<i>Navigation</i>	<i>Usage</i>
Compile Metadata Changes	RUN_PF_COMPILE	EPM Foundation, Foundation Metadata, Other Metadata Operations, Compile Metadata Changes	Compile metadata objects by running Mass Compile. To view errors, see select the Process Monitor or Report Manager link.

Related Links

[Defining Allocation Manager Rules](#)

Compile Metadata Changes Page

Use the Compile Metadata Changes page (RUN_PF_COMPILE) to compile metadata objects by running Mass Compile. To view errors, see select the Process Monitor or Report Manager link.

Navigation

EPM Foundation, Foundation Metadata, Other Metadata Operations, Compile Metadata Changes

Image: Compile Metadata Changes page

This example illustrates the fields and controls on the Compile Metadata Changes page. You can find definitions for the fields and controls later on this page.



Click the Select All button to select all the metadata types listed.



Click the Clear All button to deselect all check boxes.

SetID

Select the SetID to limit the compile to the specified input SetID.

Only Imported Metadata

Select this check box to limit the Mass Compile utility to only those objects with zero SQL counter that have been imported into the database and need compiling. You should validate this metadata beforehand using the Mass Validate process.

PATHNAME

Enter the directory location where you would like the output log to be placed.

You can either select all objects, or individually select the objects you want to validate in the Metadata Type, Performance Management Warehouse, Activity-Based Management, Scorecard, and Workforce Analytics group boxes.

Working With Tree Utilities

This topic provides an overview of tree utilities and discusses:

- Set up Tree Compare and run the Tree Compare job.
- Review Tree Compare results.
- Set up and run Super Tree.
- Review Super Tree results.

Pages Used to Work With Tree Utilities

<i>Page Name</i>	<i>Definition Name</i>	<i>Navigation</i>	<i>Usage</i>
Tree Compare	PF_TREE_CMP_TBL1	EPM Foundation, Business Metadata, Tree Metadata, Setup Tree Compare, Tree Compare	Set up and run Tree Compare by selecting trees to be compared for a business unit and scenario.
Tree Compare inquire page	PF_TREECOMP1	EPM Foundation, Business Metadata, Tree Metadata, Setup Tree Compare, Review Tree Compare, Tree Compare	Inquire on the results of a tree compare.
Super Tree	RUN_PF_SUP_TREE	EPM Foundation, Business Metadata, Tree Metadata, Update Super Tree, Super Tree	Merge multiple trees into one tree.
Super Tree inquire page	PF_SUP_TREE1	EPM Foundation, Business Metadata, Tree Metadata, Review Super Tree, Super Tree	Review results of the Super Tree process.

Understanding Tree Utilities

There are two types of tree utilities provided with EPM , Tree Compare and Super Tree.

Tree Compare Utility

The Tree Compare utility enables you to compare effective dates for trees. The results page shows nodes that have been added, deleted, or moved from one parent to another. You may also view the detail objects that have changed. To perform a tree compare, complete the Tree Compare setup page, run the Tree Compare job in a jobstream, and then inquire on the Tree Compare output using the Tree Compare inquiry page.

Super Tree Utility

The Super Tree utility enables you to combine multiple effective dates of a tree into one. This super tree contains all tree changes for a certain period of time enabling you to analyze effective-dated trees more easily.

Tree Compare Page

Use the Tree Compare page (PF_TREE_CMP_TBL1) to set up and run Tree Compare by selecting trees to be compared for a business unit and scenario.

Navigation

EPM Foundation, Business Metadata, Tree Metadata, Setup Tree Compare, Tree Compare

Image: Tree Compare page

This example illustrates the fields and controls on the Tree Compare page. You can find definitions for the fields and controls later on this page.

The screenshot shows the 'Tree Compare' page with the following details:

- Business Unit:** CORP1
- Scenario ID:** ACTUAL01
- Tree Selection Table:**

*Tree Name	Description	Compare Tree
1 DEPARTMENT	Department Tree for WB-OLD	<input checked="" type="checkbox"/>

Tree Name Select the tree you want to compare. A description displays.

Compare Tree Select this check box to compare the tree. If this check box is not selected, Tree Compare does not include this tree.

You can now run the Tree Compare.

To run the Tree Compare job, include the job PF_TREECOMP job in a jobstream.

Once the job has run, review the results on the Tree Compare inquire page.

Tree Compare inquire Page

Use the Tree Compare inquire page (PF_TREECOMP1) to inquire on the results of a tree compare.

Navigation

EPM Foundation, Business Metadata, Tree Metadata, Setup Tree Compare, Review Tree Compare, Tree Compare

Image: Tree Compare page, 1 of 2

This example illustrates the Tree Compare page, 1 of 2.

Tree Compare					
Details					
Business Unit:	CORP1	Scenario ID:	ACTUAL01	Fiscal Year:	2000
Accounting Period:	12	SetID:	SHARE	Tree Name:	DEPARTMENT
As Of Date:	01/01/1900				
Details					
Object ID	Status	Parent Node	Old Parent		
11001	Moved	IS_GRP	CUSTSER_GRP		
16510	Deleted	SALES_GRP			
17310	Deleted	ENGR_GRP			
18000	Deleted	ENGR_GRP			
18110	Deleted	ENGR_GRP			
20900	Moved	ADMIN_GRP	HR_GRP		
21101	Moved	MRKTADM	DESIGN		
21200	Moved	SALES_GRP	DESIGN		
21210	Moved	SALES_GRP	DESIGN		
21220	Moved	SALES_GRP	DESIGN		
21300	Moved	SALES_GRP	DESIGN		
21310	Moved	SALES_GRP	DESIGN		
21401	Deleted	DESIGN			
21410	Moved	SALES_GRP	DESIGN		
21420	Moved	SALES_GRP	DESIGN		

Image: Tree Compare page, 2 of 2

This example illustrates the Tree Compare page, 2 of 2.

Node				
Tree Node	Status	Parent Node	Old Parent	
DESIGN	Deleted	UNITED STATES		
HR_GRP	Moved	ADMIN	CUSTSERV	
INVMGMT	Added	MANUF		
MANUF	Added	UNITED STATES		
MANUF_GRP	Added	MANUF		
NO_DEPTID	Added	UNITED STATES		
NO_DEPT_GRP	Added	NO_DEPTID		
PLANNING	Added	MANUF		
PURCH	Added	MANUF		
QA	Added	MANUF		
SHIP/REC	Added	MANUF		

Affected Filters				
Filter	Object ID	Tree Node	Status	Parent Node
Filter				

Use this page to view nodes and details (leaves) of trees that have been deleted, added, or moved from one parent to another. You may also view all metadata filters affected by the deleted nodes and details.

Filter

You can change any filter by clicking on the link for that filter. The system transfers you to the Filter page on which you can modify the filter to reflect tree changes.

Note: If you determine that the tree nodes are being used in ledger to resource mappings, run the ABM Model Validation job to find the impact on the mapping rules.

Super Tree Page

Use the Super Tree page (RUN_PF_SUP_TREE) to merge multiple trees into one tree.

Navigation

EPM Foundation, Business Metadata, Tree Metadata, Update Super Tree, Super Tree

Image: Super Tree page

This example illustrates the fields and controls on the Super Tree page. You can find definitions for the fields and controls later on this page.

The screenshot shows the 'Super Tree' configuration page. At the top, it displays 'User ID: VP1' and 'Run Control ID: SUTREE01'. There are links for 'Report Manager' and 'Process Monitor', and a 'Run' button. The main form contains the following fields:

- Program Name:** PF_SUP_TREE
- When:** Once (dropdown menu)
- *Description:** (empty text box)
- *SetID:** SHARE (with a search icon)
- *Tree Name:** 2010BUDGET (with a search icon) and 'Time Tree' (text)
- *Dominant Effective Date:** 01/01/1900 (with a search icon)
- *Time Span:** (empty dropdown menu)
- *Super Tree Name:** (empty text box)

SetID

SetID for the tree you want to merge.

Tree Name

Name of the tree to be merged to create a super tree.

Dominant Effective Date

Trees may have the same name and different effective dates. The Super Tree is created using the dominant effective date you enter in this field.

TimeSpan

The timespan determines the begin and end date of the period for which the merge is performed. Trees with effective dates that are between the begin and end dates are included in the super tree.

Super Tree Name

You must select a valid tree name for the super tree.

Warning! If a tree that is part of the super tree has a dead node, the dead node must be removed manually. Otherwise the super tree will not build properly. Dead nodes are created when a leaf or node is deleted. You may add a deleted node or leaf elsewhere on the tree.

Note: Any nodes or details deleted before the dominant effective date and any nodes or details added after the dominant effective date will not be included in the super tree.

Super Tree inquire Page

Use the Super Tree inquire page (PF_SUP_TREE1) to review results of the Super Tree process.

Navigation

EPM Foundation, Business Metadata, Tree Metadata, Review Super Tree, Super Tree

Get Nodes Click this link to populate the page with the nodes of the super tree.

Get Details Click to view the related object IDs.

Setting Up and Flattening Tree Metadata

This topic provides overviews of trees and tree flattening, and discusses how to:

- Define trees.
- Flatten tree metadata.

Pages Used to Set Up and Flatten Tree Metadata

<i>Page Name</i>	<i>Definition Name</i>	<i>Navigation</i>	<i>Usage</i>
Tree Metadata	PF_METATREE_TBL1	EPM Foundation, Business Metadata, Tree Metadata, Tree Metadata	Define tree metadata.
Run Tree Flattener	PF_RUN_TREEFLAT	EPM Foundation, Business Metadata, Tree Metadata, Tree Flattener	Run the tree flattener process.
Jobstream Email Notification	PF_EMAIL_MSG	EPM Foundation, Business Metadata, Tree Metadata, Tree Flattener, Specify Email Parameters	Specify email parameters for the tree flattener notification.

Understanding Trees

When you add trees in PeopleTools Tree Manager, you should also create a record on the Tree Metadata page to identify it to the PeopleSoft EPM system. The Tree Manager does not assign a level number to a node unless you specify that the tree either loosely or strictly enforces levels. Levels are assigned by creating a code for each level. In PeopleSoft EPM, for trees that use rules and are processed through the tree level, you must set up your trees using levels. PeopleSoft recommends specifying *strictly enforced levels* for all your trees.

PeopleSoft trees add a visual layer to show how detail items such as departments, accounts, products, channels, geography, and security fit into your organizational structure.

Trees depict hierarchical structures that represent a group of summarization rules for a particular database field. For example, a tree can specify how your manufacturing locations should be summarized, or rolled up, for reporting purposes. A tree can also represent the reporting relationships within an organization by specifying how the individual department should be summarized into territories, territories into regions, and regions into countries. Similarly, a tree can categorize items in a catalog.

The summarization rules depicted in a tree apply to the detail values of a particular field: vendors, departments, customers, or other values that you define. These detail values are summarized into nodes on a tree. The nodes may also be organized into levels to logically group nodes that represent the same type of information or level of summarization.

By building trees, you give the system a single place to look for summarization rules. Trees enable you to define rules once and then use them throughout the system. Different reports, ledgers, and security profiles might refer to parts of your company's organizational chart; therefore, all of these objects can be referenced in the same predefined tree.

For example, the values of the DEPTID field identify individual departments in your organization. You use Tree Manager to define the organizational hierarchy that specifies how each department relates to the others—departments 10700 and 10800 report to the same manager, department 20200 is part of a different division, and so on.

You create trees using the PeopleSoft Tree Manager. You can use the PeopleSoft Tree Mover to move trees between different versions of PeopleSoft databases, move tree node data, or move tree level data.

For more information about PeopleSoft Tree Manager, see the *PeopleTools: PeopleSoft Tree Manager* documentation.

Understanding Tree Flattening in the OWE

Trees are used in PeopleSoft EPM to represent hierarchies. The trees are stored in a proprietary format that is optimized for the PeopleSoft Tree Manager module. However, this format is not optimized for EPM batch processing. To convert tree structures to a format that is optimized for batch processing, PeopleSoft delivers a tree flattener process.

The Tree Flattener (PF_TREEFLT) application engine processes the hierarchy tree and converts it to a flattened format, which simplifies the structure to retrieve information such as all descendents and immediate descendents from a node with a simple join to the flattened table. The Application engine can then use very simple SQL to access the tree data that it needs. The flattened output from the Tree Flattener (PF_TREEFLT) process is loaded to a temporary table, and you can choose to load the flattened data to a permanent OWE table if you wish. The tree flattener process can flatten detail, node-oriented, and dynamic detail trees.

To flatten dynamic detail trees, the tree flattener process retrieves the dynamic detail value table name from the Tree Structure ID and inserts these values into the flattened table. During this process a static view of the tree is created as of the date and time the flattening occurs. This means that the flattener process gets the current values of the tree detail from the dynamic detail value table and uses these in the flattener table. Because of this fact, you should not use the Persist Permanently check box with dynamic detail trees unless you are sure that the dynamic table will not change.

Note: The OWE uses a different process for flattening trees than the MDW. For more information on the MDW tree flattening process see the *Processing Trees and Recursive Hierarchies* topic in your warehouse specific documentation (for example, the *PeopleSoft Campus Solutions Warehouse*).

Tree Metadata Page

Use the Tree Metadata page (PF_METATREE_TBL1) to define tree metadata.

Navigation

EPM Foundation, Business Metadata, Tree Metadata, Tree Metadata

Image: Tree Metadata page

This example illustrates the fields and controls on the Tree Metadata page. You can find definitions for the fields and controls later on this page.

The screenshot shows the 'Tree Metadata' page with the following details:

- SetID:** SHARE
- Tree ID:** EMPL_TREE
- Tree Metadata Information:**
 - *Effective Date: 01/01/1997
 - *Status: Active
 - *Description: Employee Tree
 - *Tree Name: EMPLOYEE
 - Buttons: Display Tree
- TreeTables:**
 - *Flattened Table: PF_TRFL_KPI_TBL
 - Persist Permanently
 - Permanent Table: PF_TREEFLAT_TBL
 - Tree Type:**
 - Summer Tree
 - Winter Tree

Tree Metadata Information

Tree Name Select the tree for which you are creating tree metadata.

Display Tree Click to access the Tree Viewer page and view your tree.

See [Tree Viewer Page](#).

Tree Tables

Flattened Table When you run the tree flattener (PF_TREEFLT) process, the preprocessor output is stored in a temporary table, which is associated with the table that you select here.

To move the preprocessor output from the temporary table to a permanent OWE table, you must select the Persist Permanently check box located on this page.

Note: You should not assign the same tree flattener table to more than one tree metadata definition.

Persist Permanently

Select this check box to move the preprocessor (flattened) output to a permanent OWE table from the temporary table.

The Permanent Table field appears when you select this check box.

Note: You should not select this check box for dynamic detail trees unless you are sure that the dynamic table will not change.

Permanent Table

Select a permanent OWE table you want to store the flattened output.

Tree Type

Specify whether the tree is a summer or winter tree.

If you override a summer tree by selecting *Winter*, the tree nodes are used, and the summer tree leaf values are ignored.

You cannot override a winter tree.

Run Tree Flattener Page

Use the Run Tree Flattener page (PF_RUN_TREEFLAT) to run the tree flattener process.

Navigation

EPM Foundation, Business Metadata, Tree Metadata, Tree Flattener

Image: Tree Flattener page

This example illustrates the fields and controls on the Tree Flattener page. You can find definitions for the fields and controls later on this page.

The screenshot shows the 'Tree Flattener' configuration page. At the top, it displays 'User ID: VP1' and navigation links for 'Report Manager' and 'Process Monitor', along with a 'Run' button. Below this, the 'Run Control ID' is 'TREE_FLAT01'. The main configuration area includes:

- Program Name:** PF_TREEFLT
- When:** Always (dropdown menu)
- *Description:** (empty text field)
- *SetID:** SHARE (with a search icon)
- *As Of Date:** (empty date field with a calendar icon)
- Job ID:** PF_TREEFLT
- Options:** Rerun Option (checkbox), Send Email Notification (checkbox), and a link for Specify Email Parameters.

 A table titled 'Trees to Flatten' is shown below, with columns for 'Tree ID' and 'Description'. It contains one entry:

Tree ID	Description
1 ABM_ACT_TREE	ABM ACT TREE

 At the bottom, there are fields for 'Last Run On:' and 'As Of Date:'.

SetID	Select the SetID that is associated with the tree being flattened.
Job ID	<p>Displays the job ID that is associated with the run control ID.</p> <p>Because the tree flattener process is not delivered as a standalone process but as a routine that is called within application engine, you must associate the run control ID with a job ID.</p>
Specify Email Parameters	Click to access the Jobstream Email Notification page and specify email details for the email notification.
Tree ID	<p>Enter the tree ID for the tree that you want flattened</p> <p>You can add rows to have multiple trees flattened.</p>

Setting Up Business Rules for the Operational Warehouse - Enriched

Setting Up Account Information

This topic provides an overview of account information setup and discusses how to:

- Define account types.
- Define accounts.
- Define account nodes.

Pages Used to Set Up Account Information

<i>Page Name</i>	<i>Definition Name</i>	<i>Navigation</i>	<i>Usage</i>
Account Types	ACCT_TYPE	EPM Foundation, Business Metadata, Business Framework, Account Types	Define the account types to which you will later assign accounts. Account types describe the various categories of accounts that correspond to those on your balance sheet or income statement.
Accounts	ACCOUNT_TBL1	EPM Foundation, Business Metadata, Business Framework, Accounts	Define new PeopleSoft EPM accounts.
Account Nodes	PF_ACCT_NODE_DFN	EPM Foundation, Business Metadata, Business Framework, Account Nodes	Define the accounts nodes to which you will later assign accounts.

Understanding Accounts

Like general ledger accounts, accounts in PeopleSoft EPM hold the key to monetary values stored in ledgers. These values are held in certain buckets—accounts—according to their function.

If you have PeopleSoft General Ledger you can duplicate your general ledger account structure from your transaction system using PeopleTools Data Mover to move your accounts to your PeopleSoft EPM database. If you use another online transaction processing (OLTP) system or a legacy system, you can migrate your structure using data migration tools.

PeopleSoft EPM contains a couple of important account tables:

GL_ACCOUNT_TBL	Contains all currently active general ledger and performance (PF) accounts, keyed by SetID.
PF_ACCOUNT_VW	View of the account table that filters for performance (PF) accounts only.

Account Types Page

Use the Account Types page (ACCT_TYPE) to define the account types to which you will later assign accounts. Account types describe the various categories of accounts that correspond to those on your balance sheet or income statement.

Navigation

EPM Foundation, Business Metadata, Business Framework, Account Types

Image: Account Types page

This example illustrates the fields and controls on the Account Types page. You can find definitions for the fields and controls later on this page.

The screenshot shows the 'Account Types' page with the following fields and controls:

- SetID:** SHARE
- Account Type:** R
- Description:** REVENUE
- Short Desc:** REVENUE
- Balance Forward

Account Type

Displays a letter-code that identifies each account type. You assign account codes to new accounts (those not migrated from other systems) that you enter on the Account page.

Values include:

A: Asset

E: Expense

L: Liability

Q: Equity

R: Revenue

F: Force Balancing

D: Detail

Balance Forward

Select if you use this account type primarily to determine whether to store balance forward amounts. How you track balance forward amounts is completely up to you. For example, in your regular accounting system, you may want asset, liability,

and equity accounts specified as balance forward accounts, but not revenue or expense accounts. The accounts you define later on the Accounts page will take on the balance forward attribute of the account type you assign to them.

Warning! For accounts that you use with PeopleSoft financial services industry applications it is important that you not deviate from the account type codes mentioned here. All asset accounts must be tagged with an *A*, expense accounts with an *E*, and so on.

Accounts Page

Use the Accounts page (ACCOUNT_TBL1) to define new PeopleSoft EPM accounts.

Navigation

EPM Foundation, Business Metadata, Business Framework, Accounts

Image: Accounts page

This example illustrates the fields and controls on the Accounts page. You can find definitions for the fields and controls later on this page.

The screenshot displays the 'Accounts' page in a web browser. At the top, it shows 'SetID: SHARE' and 'Account: 000000'. Below this is a 'Details' section with a search bar and navigation controls. The main form contains the following fields and controls:

- *Effective Date:** 01/01/2002 (with a calendar icon)
- *Status:** Active (dropdown menu)
- *Description:** All Accounts (text input)
- *Short Description:** All Accts (text input)
- Monetary Account Type:** ASSET (dropdown menu)
- Account** section:
 - GL Account
 - PF Account
 - ABM Account
 - Statistical Account
- Unit of Measure:** (dropdown menu)

Monetary Account Type

Select from the account types defined on the Account Types page.

Note: To obtain account information on reports, you need to add those accounts to the system. In PeopleSoft EPM, you set up a base of general ledger accounts by migrating them from your transaction accounting system. However, when your data migration is complete, you may find that you need to add new accounts to your system.

Warning! For accounts that you will use with financial services applications—PeopleSoft Funds Transfer Pricing and Risk-Weighted Capital—it is important that you not deviate from the standard account type codes. All asset accounts must be tagged with an *A*, expense accounts with an *E*, and so on.

GL Account	Select if the account is a transaction-based account.
PF Account	Select if the account is performance (PF) ledger.
ABM Account	An account tagged as an ABM account creates more detailed information within the performance (PF) ledger. The ABM-specific fields on this page enable you to specify that the performance ledger contain both the appropriate profitability dimension (cost object), and the activity that contributed costs to the cost object.
Object Type	When you select the ABM Account check box, a drop-down list box for object type appears. Select a default object type for the account.
Activity ID	When you select the ABM Account check box, a drop-down list box for activity ID appears. Select a default activity for the account.
Statistical Account	If you are establishing a statistical account, select the check box and select a unit of measure. Statistical accounts store only statistical amounts, not monetary amounts.
Unit of Measure	If you want to track statistical amounts, select a unit measure.

Warning! Although monetary amounts are assigned to performance (PF) accounts, there is *no* reconciliation *per se*, as there is in PeopleSoft General Ledger. The performance (PF) account and source dimensions are used to store and reconcile amount facts to the same amounts stored by a general ledger account. Performance (PF) accounts can differ significantly from general ledger accounts.

Account Nodes Page

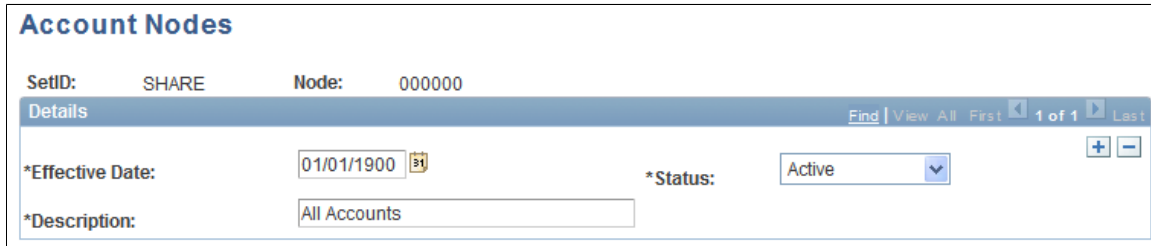
Use the Account Nodes page (PF_ACCT_NODE_DFN) to define the accounts nodes to which you will later assign accounts.

Navigation

EPM Foundation, Business Metadata, Business Framework, Account Nodes

Image: Account Nodes page

This example illustrates the fields and controls on the Account Nodes page.



The screenshot shows the 'Account Nodes' page. At the top, it displays 'SetID: SHARE' and 'Node: 000000'. Below this is a 'Details' section with a search bar and navigation controls. The main form contains the following fields:

- *Effective Date: 01/01/1900 (with a calendar icon)
- *Status: Active (dropdown menu)
- *Description: All Accounts (text input field)

You can define accounts nodes to which you will later assign accounts. The page displays an effective date, status, and description for each node.

Specifying Ledger Mapping Defaults

As a next step you need to specify your ledger and ChartField mapping.

On the Ledger Mapper Defaults pages, you specify the defaults and behavior for the Ledger Mapper page in EPM. The defaults you specify here determine how a new row in the Ledger Mapper page is presented and edited. You can override the defaults later on the Ledger Mapper page if you want to set up your mappings differently.

You use the Ledger Mapper to map data, such as assets and liabilities, that does not come into the system through one of the optional analytical applications. The Ledger Mapper is described later in this documentation.

This topic discusses how to:

- Specify tree views.
- Specify trees, default values, and validation.
- Set ledger mapping tree usage.

Pages Used to Set Up Ledger Mapper Defaults

Page Name	Definition Name	Navigation	Usage
Ledger Mapper Defaults	PF_LED_TMPL_TBL1	EPM Foundation, EPM Setup, Ledger Setup, Ledgers, Ledger Mapper Defaults, Ledger Mapper Defaults	Specify tree views to filter the tree information you've set up for your ledger and ChartField mappings.

Page Name	Definition Name	Navigation	Usage
Ledger Mapper Defaults II	PF_LED_TMPL_TBL2	EPM Foundation, EPM Setup, Ledger Setup, Ledgers, Ledger Mapper Defaults, Ledger Mapper Defaults II	Specify the trees, default values, and validations that define how your ledgers and ChartFields roll up, which default values to draw from, as well as whether or not to allow blank values.
Map Ledger to Tree	PF_LM_TOPT_TBL1	EPM Foundation, EPM Setup, Ledger Setup, Ledgers, Map Ledger to Tree	Set ledger mapping tree usage.

Related Links

[Setting Up Ledger Mapper](#)

Common Elements Used to Specify Ledger Mapping Defaults

ChartField

The ChartFields that appear on the list come from the LEDMAP_CHART subrecord.

Ledger Mapper Defaults Page

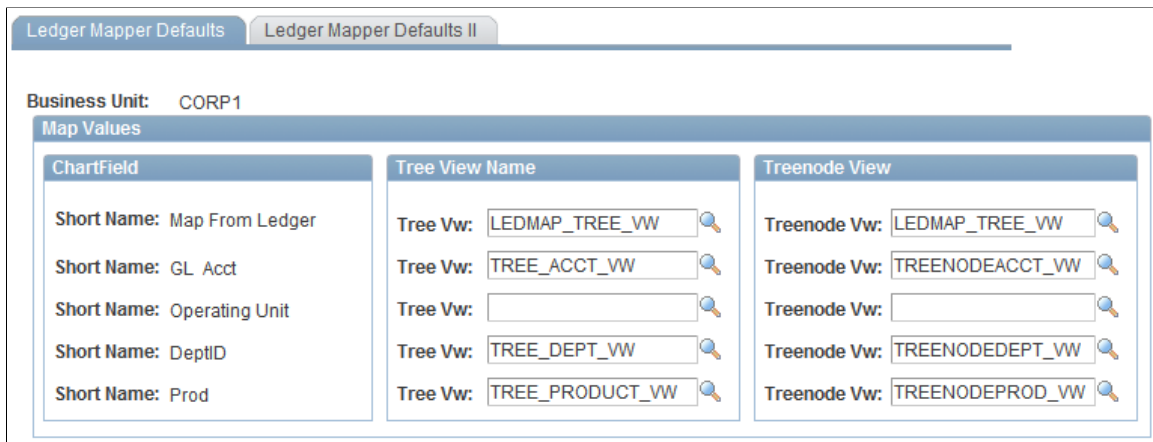
Use the Ledger Mapper Defaults page (PF_LED_TMPL_TBL1) to specify tree views to filter the tree information you've set up for your ledger and ChartField mappings.

Navigation

EPM Foundation, EPM Setup, Ledger Setup, Ledgers, Ledger Mapper Defaults, Ledger Mapper Defaults

Image: Ledger Mapper Defaults page

This example illustrates the fields and controls on the Ledger Mapper Defaults page. You can find definitions for the fields and controls later on this page.



- ChartField** The page displays the ChartFields: map from ledger, general ledger account, operating unit, department ID, and product.
- Tree View Name** For each ChartField, specify the appropriate tree view name. The prompt displays all views that have Set Control Field as BUSINESS_UNIT and TREE in the name of the view.
- Treenode View** For each ChartField, specify the appropriate treenode view name.

Note: If you configure ChartFields of your own to add to the system, you will need to duplicate the tree view, treenode view, and prompts. The system uses the specified views to select members from the trees you specify on the Ledger Mapper Defaults II page.

Ledger Mapper Defaults II Page

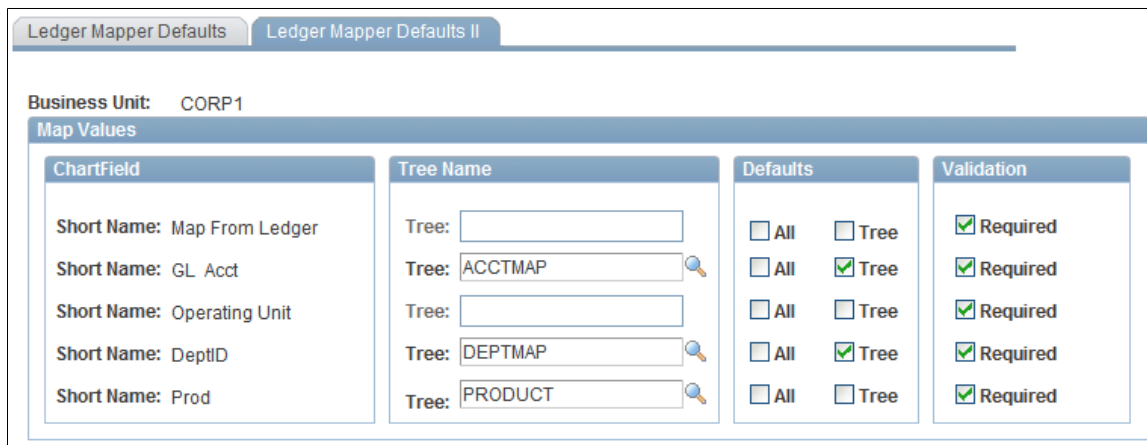
Use the Ledger Mapper Defaults II page (PF_LED_TMPL_TBL2) to specify the trees, default values, and validations that define how your ledgers and ChartFields roll up, which default values to draw from, as well as whether or not to allow blank values.

Navigation

EPM Foundation, EPM Setup, Ledger Setup, Ledgers, Ledger Mapper Defaults, Ledger Mapper Defaults II

Image: Ledger Mapper Defaults II page

This example illustrates the fields and controls on the Ledger Mapper Defaults II page. You can find definitions for the fields and controls later on this page.



Tree Name For each ChartField including your ledger, select the appropriate tree name from which you'll map values. The tree you select should also represent how your ChartFields roll up for reporting purposes. The tree name is required if you select the Tree check box.

Defaults Select whether to map all values in the system for the ChartField row or only to those values that are filtered out through the

tree and tree views you selected. If you select Tree, the system requires you to enter the tree name.

Validation

To disallow the selection of blank values from prompts and drop-down list boxes for a specific ChartField or for your ledger, select the appropriate validation check box. This validation occurs on saving the Ledger Mapper Defaults component.

Map Ledger to Tree Page

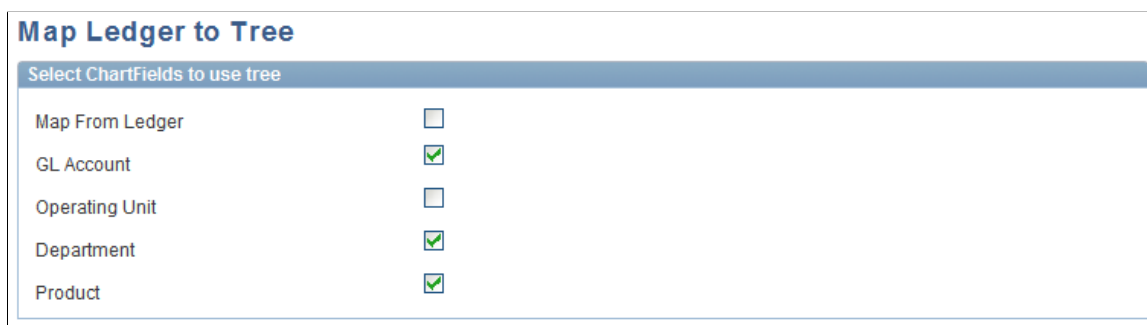
Use the Map Ledger to Tree page (PF_LM_TOPT_TBL1) to set ledger mapping tree usage.

Navigation

EPM Foundation, EPM Setup, Ledger Setup, Ledgers, Map Ledger to Tree

Image: Map Ledger to Tree page

This example illustrates the fields and controls on the Map Ledger to Tree page.



To specify whether a tree will be used while mapping the ChartField select the appropriate check box in the Select ChartFields to use trees group box.

Note: This page is used for general ledger mapping in addition to ABM ledger mapping. This page and the Ledger Mapping Defaults pages should be defined before you create ledger mapping rules.

Defining Ledger Event Codes

Ledger event codes relate performance ledger data so that you can report on the data or compare data between and within engines and models. Ledger event codes also make it possible for the Data Manager and Allocation Manager to assign an account to the target (as defined in the Data Manager or Allocation Manager) for reporting purposes and for posting.

This topic discusses how to establish ledger event codes.

Page Used to Set Up Ledger Event Codes

Page Name	Definition Name	Navigation	Usage
Ledger Event Codes	PF_SOURCE_TBL	EPM Foundation, Business Metadata, Business Framework, Ledger Event Codes	Establish ledger event codes.

Ledger Event Codes Page

Use the Ledger Event Codes page (PF_SOURCE_TBL) to establish ledger event codes.

Navigation

EPM Foundation, Business Metadata, Business Framework, Ledger Event Codes

Image: Ledger Event Codes page

This example illustrates the fields and controls on the Ledger Event Codes page. You can find definitions for the fields and controls later on this page.

The screenshot shows the 'Ledger Event Codes' page with the following details:

- SetID:** SHARE **Ledger Event Code:** 01
- Details:** Find | View All | First | 1 of 1 | Last
- *Effective Date:** 01/01/1900
- *Status:** Active
- *Description:** Product ABM Costs
- Short Description:** ABM Prod
- *Source:** Activity-Based Mgmt Engine
- Account:** PRODU
- Account Offset:** (empty)
- Model ID:** (Invalid Value)
- Record:** CALC_DETAIL_F00

Source

Use to specify the data that you will tie to the code. This helps you further filter where the amounts for this code originate. You can select from output that originates in your general ledger or transaction system, or in the various PeopleSoft EPM engines.

Account

Specify the account that you want to assign to the ledger event code.

Account Offset

You can also specify the account offset that you want to assign to the ledger event code.

Model ID

(Optional) Specify a model to associate with the code.

Record

(Optional) Specify a record to associate with the code.

Note: If you use PeopleSoft Funds Transfer Pricing or Risk-Weighted Capital, you assign ledger event codes to rules through the Rules pages in those applications. For more information, see the documentation for those applications.

To use ledger event codes appropriately through the Data Manager or Allocation Manager, you need to designate the PF_LEDGER_EVENT_CD field on the TableMap and DataMap system pages.

Related Links

[Setting Up Tablemaps](#)

[Setting Up Datamaps](#)

Defining Performance Ledger Templates

This topic provides an overview of performance ledger templates and discusses how to:

- Set up a performance ledger template.
- Specify ledger template ChartFields.
- Assign temporary tables to records.

Pages Used to Define Performance Ledger Templates

<i>Page Name</i>	<i>Definition Name</i>	<i>Navigation</i>	<i>Usage</i>
Ledger Template - Ledger Template	LEDGER_TEMPLATE1	EPM Foundation, EPM Setup, Ledger Setup, Ledger Template, Ledger Template	Set up a performance ledger template.
Ledger Template - ChartFields	LEDGER_TMPLT_CF	EPM Foundation, EPM Setup, Ledger Setup, Ledger Template, ChartFields	Specify ledger template ChartFields.
Ledger Template - Edit and Post Variables	LEDGER_TMPLT_EDPST	EPM Foundation, EPM Setup, Ledger Setup, Ledger Template, Edit and Post Variables	Assign temporary tables to records.

Understanding Performance Ledger Templates

A ledger template defines the physical attributes of a ledger and streamlines the ledger definition process. You define the template once and use it for all your ledgers.

When you create a ledger template, the system automatically identifies the list of ChartFields based on the ledger table selected. The ChartFields are identified based on the following criteria:

ChartField is equal to any field between the ACCOUNTING_PERIOD and CURRENCY_CD fields based on the selected ledger table's structure in the database

The currency code field is also always treated as a ChartField.

PeopleSoft provides a number of predefined ledger templates, which include default records and fields that support the ChartFields and other field and record definitions delivered with the system. (SetID does not key the template). If you want to create a new template or modify a template's records or fields, you need to follow the instructions in this topic.

You define performance ledger templates using the Ledger Template component.

Note: The last page in the Ledger Template component, Budget Variables, is used by PeopleSoft Planning and Budgeting and is described in *PeopleSoft EPM: Planning and Budgeting* documentation.

Ledger Template - Ledger Template Page

Use the Ledger Template - Ledger Template page (LEDGER_TEMPLATE1) to set up a performance ledger template.

Navigation

EPM Foundation, EPM Setup, Ledger Setup, Ledger Template, Ledger Template

Image: Ledger Template page

This example illustrates the fields and controls on the Ledger Template page. You can find definitions for the fields and controls later on this page.

Detail

Select this ledger type for ledgers containing detail ledger ChartFields which serve as keys.

Summary

Select this ledger type for ledgers to be used for reporting purposes.

EPM Ledger Type

Unless you're working with PeopleSoft Global Consolidations or Planning and Budgeting, you'll select *Performance Measurement Ledger*. If you change the EPM Ledger Type field, the system resets the values for the ledger records and ChartFields. Your selection here also determines what fields will be displayed on the Detail Ledger page so that the ledger data in the ledger table is uniquely identified.

Note: If you select an EPM ledger type of *Consolidation Source Ledger*, do not select a journal line table name as journal entries for these tables are not supported.

Used in Consolidations

Select this check box if this ledger template is used for consolidate information for reporting.

Ledger Record

Represents the ledger record that you use to store the balances for each ChartField combination that has posted activity.

Journal Line

Stores detail lines containing the monetary and statistical amounts for each journal entry.

Note: The ledger record and journal line prompts are based on PeopleSoft EPM record metadata. If your desired table is not returned in the prompt, define it in your PeopleSoft EPM record metadata first. You can access the Record Metadata page for the ledger record or journal line currently selected directly from the Ledger Template page by selecting the link beside the Ledger Record and Journal Line fields. The link text varies based on the ledger record or journal line selected. In the example above, in the Ledger Record field, select the Performance Ledger link to access the Record Metadata page for the ALM_LEDGER_F00 performance ledger.

Related Links

[Setting Up Record Metadata](#)

Ledger Template - ChartFields Page

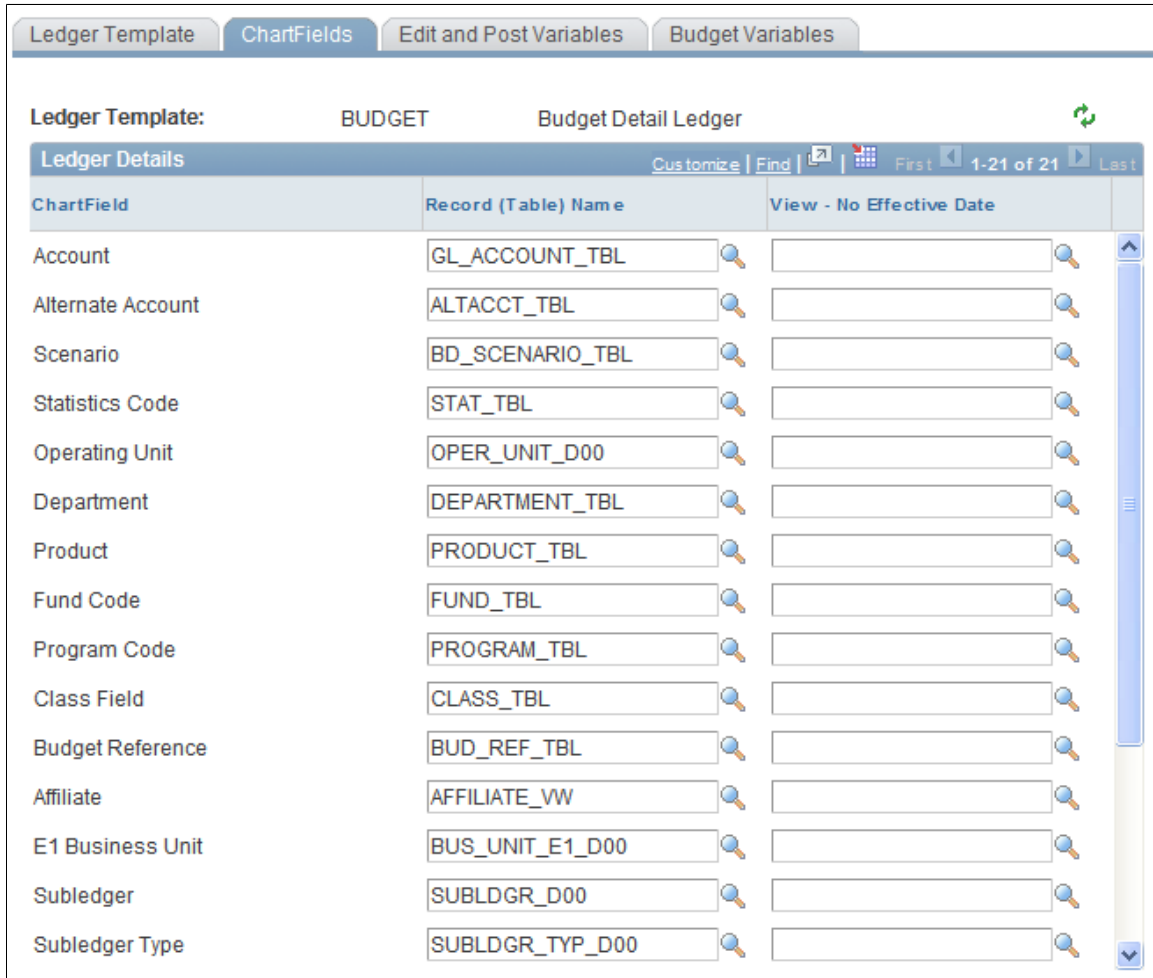
Use the Ledger Template - ChartFields page (LEDGER_TMPLT_CF) to specify ledger template ChartFields.

Navigation

EPM Foundation, EPM Setup, Ledger Setup, Ledger Template, ChartFields

Image: Ledger Template - ChartFields page

This example illustrates the fields and controls on the Ledger Template - ChartFields page.



ChartField	Record (Table) Name	View - No Effective Date
Account	GL_ACCOUNT_TBL	
Alternate Account	ALTACCT_TBL	
Scenario	BD_SCENARIO_TBL	
Statistics Code	STAT_TBL	
Operating Unit	OPER_UNIT_D00	
Department	DEPARTMENT_TBL	
Product	PRODUCT_TBL	
Fund Code	FUND_TBL	
Program Code	PROGRAM_TBL	
Class Field	CLASS_TBL	
Budget Reference	BUD_REF_TBL	
Affiliate	AFFILIATE_VW	
E1 Business Unit	BUS_UNIT_E1_D00	
Subledger	SUBLDGR_D00	
Subledger Type	SUBLDGR_TYP_D00	

Select Edit Table and View - No Effective Date records to use for this template.

The page accesses a set of default edit table names and reporting views for the ChartFields in the ledger. The edit table value comes from the edit table specified in the Application Designer for this field on the ledger table. The View - No Effective Date field is used for reporting prompts when you do not want to limit the selection by effective date (because historical rows might contain ChartFields that are no longer active). You can change these values if you are modifying your system.

Ledger Template - Edit and Post Variables Page

Use the Ledger Template - Edit and Post Variables page (LEDGER_TMPLT_EDPST) to assign temporary tables to records.

Navigation

EPM Foundation, EPM Setup, Ledger Setup, Ledger Template, Edit and Post Variables

Image: Ledger Template - Edit and Post Variables page

This example illustrates the fields and controls on the Ledger Template - Edit and Post Variables page. You can find definitions for the fields and controls later on this page.

Temp Table 1 Name, Temp Table 2 Name, Temp Table 3 Name, and Temp Table 4 Name

Use to assign up to four temporary tables to the ledger template. Your temporary table selection must match those tables in your ledger template. Each of the four temporary tables represent the journal and ledger records that Profit Manger uses to process your ledgers and journals.

Run Edit for Profit Manager

Select this check box to run edit for the Profit Manager.

Note: The journal and ledger tables have ChartField subrecords. If you add ChartFields, you must modify these table subrecords

Related Links

[Processing and Posting Journals](#)

Defining Detail Ledgers

Use the Detail Ledger component to link a ledger template to a detail ledger and specify the balancing options for the journals generated to a detail ledger.

Note: The second page in the Detail Ledger component, Consolidations, is used by PeopleSoft Global Consolidations and is described in *PeopleSoft EPM: Global Consolidations*.

Page Used to Define Detail Ledgers

Page Name	Definition Name	Navigation	Usage
Detail Ledger - Definition	LEDGER_DETAIL1	EPM Foundation, EPM Setup, Ledger Setup, Ledgers, Detail Ledger, Definition	Identify a unique set of ledger data within the ledger table.

Detail Ledger - Definition Page

Use the Detail Ledger - Definition page (LEDGER_DETAIL1) to identify a unique set of ledger data within the ledger table.

Navigation

EPM Foundation, EPM Setup, Ledger Setup, Ledgers, Detail Ledger, Definition

Image: Detail Ledger - Definition page

This example illustrates the fields and controls on the Detail Ledger - Definition page. You can find definitions for the fields and controls later on this page.

Ledger Template

Select the ledger template to link to this detail ledger.

Click the link that displays alongside the ledger template you select to access the Ledger Template page for that template.

EPM Ledger Type

Displays the type of ledger selected in the Ledger Template field.

Ledger

Displays for a ledger type of standard general ledger, budget project ledger, commitment budget ledger.

Budgeting Scenario

Displays for a ledger type of budget project ledger and commitment budget ledger.

Scenario ID

Displays for a ledger type of performance measurement ledger and consolidation ledger. Select the scenario ID to use.

Defining Ledger Groups

Ledger groups enable you to group detail ledgers by ledger template and ledger type.

This topic discusses how to set up ledger groups.

Page Used to Define Ledger Groups

Page Name	Definition Name	Navigation	Usage
EPM Ledger Group	LEDGER_GROUP	EPM Foundation, EPM Setup, Ledger Setup, Ledgers, Ledger Group	Set up a ledger group and assign detail ledgers to the group and identify the unique attributes of each ledger within the group.

EPM Ledger Group Page

Use the EPM Ledger Group page (LEDGER_GROUP) to set up a ledger group and assign detail ledgers to the group and identify the unique attributes of each ledger within the group.

Navigation

EPM Foundation, EPM Setup, Ledger Setup, Ledgers, Ledger Group

Image: EPM Ledger Group page

This example illustrates the fields and controls on the EPM Ledger Group page. You can find definitions for the fields and controls later on this page.

EPM Ledger Group

SetID: SHARE

Ledger Group: PROJECTS

*Description:

Ledger Template: [BD Projects Detail Ledger](#)

Ledger Group Type:

Ledger Details Customize |

Types Attributes

*Ledger ID	Description	Primary Ledger	Balanced Ledger	Currency Translation Ledger		
<input type="text" value="PROJ_ACT"/>	Project Actuals	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="text" value="PROJ_BUDG"/>	Project Budgets	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="text" value="PROJ_FCST"/>	Project Forecasts	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Ledger Template

Select a ledger template from the drop-down list box. Ledger templates are defined on the Ledger Template pages. Click the link alongside the Ledger Template field to transfer to the Ledger Template Definition page for the selected template.

Ledger Group Type

Select a group type from the following:

Average Daily Balance: Allows only a primary ledger.

Budget: Allows only a primary ledger.

Commitment Control Expense: Used by commitment control budgeting.

Commitment Control Revenue: Used by commitment control budgeting.

Standard: Allows one primary and up to nine secondary ledgers.

Translation: Allows only a primary ledger.

Entering Ledger Details

The Ledger Details grid at the bottom of the page displays two tabs: Types and Attributes.

On the Types tab select the following:

Ledger ID

Select the detail ledger you want to link to this ledger group. The detail ledgers listed are those associated with the ledger template defined on the Detail Ledger page.

Primary Ledger

Select this check box for the primary ledger.

You can specify one primary ledger and up to nine secondary ledgers for ledger groups that have a ledger group type of standard. For average daily balance, budget, and translation ledger group types, you can select only one ledger.

Balanced Ledger

Select if this ledger contains balanced ChartFields or you want to balance the ChartFields.

Currency Translation Ledger

Select if this is a currency translation ledger. Any non-primary ledger may be defined as a translation ledger. Currency translation ledgers are handled differently from normal secondary ledgers with respect to journal processing and multicurrency processing.

On the Attributes tab select the following:

Inherit Base Currency

Select to inherit the base currency of the business unit. The system uses the business unit base currency in the ledger.

Base Currency

If you do not elect to inherit the base currency, enter the base currency to use for the ledger.

Exchange Rate Type Default

Enter a rate type to use as a default.

This field influences which rate type will be used for the secondary lines. The calculation hierarchy is this: The system first checks for a value in rate type for the ledger (see the next field). It will always use this value for the secondary lines, regardless of what you enter in the Exchange Rate Type Default field. The system then checks for a rate type on the primary journal line. If there is one, it uses that rate type for the secondary line. If there is no value in rate type or in the primary journal line, the system uses the value entered in the Exchange Rate Type Default field.

Rate Type

Select the currency exchange rate type to use in the autogenerated journal lines for the secondary ledgers.

Note: If you're setting up a commitment control ledger group, two additional fields display: the Commitment Control Ledger Type field and the Affect Spending Authority field. These fields are specifically for commitment control purposes.

Processing Roll-Ups

This topic provides an overview of EPM roll-up engines, lists common elements, and discusses how to:

- Roll up scenarios.
- Roll up calendars.
- Roll up business units.

Pages Used to Run Roll-Up Engines

<i>Page Name</i>	<i>Definition Name</i>	<i>Navigation</i>	<i>Usage</i>
Roll-Up Scenario	RUN_PF_SCN_ROLL	EPM Foundation, Data Enrichment Tools, Roll-up Accounting Info, Scenarios, Roll-Up Scenario	Roll up scenarios to produce an aggregated scenario set of data based on the scenario tree. Generate your reports after you run the Scenario Roll-Up engine and the Calendar Roll-Up engine.
Roll-Up Calendar	RUN_PF_CAL_RU	EPM Foundation, Data Enrichment Tools, Roll-up Accounting Info, Calendar, Roll-Up Calendar	Roll up calendars to produce an aggregate set of product data based on calendars.
Roll-Up Business Unit	RUN_PF_BU_ROLL	EPM Foundation, Data Enrichment Tools, Roll-up Accounting Info, Business Units, Roll-Up Business Unit	Roll up business units to produce an aggregate business unit set of data based on the business unit tree.

Understanding EPM Roll-Up Engines

In EPM you can run calendar roll-ups on the application data, the performance ledger table (PF_LEDGER_F00), and roll up ledgers on the GL ledger table. In addition you can roll up scenario IDs and business units. Running the roll-up engines enables you to present the same information in a different (aggregated) way.

You can use the calendar roll-ups to roll up accounting periods based on the summary and detail calendars that you defined for your scenario. Most likely, you will use a calendar roll-up before you report on your scenario. For example, you may have monthly values in your current scenario data, but want to see quarterly values.

You can use the scenario ID and business unit roll-up engines to consolidate values. For example, when you define warehouse business units, you may specify that several business units roll up under a consolidated business unit. When you then run the business unit roll-up, the individual business unit data is aggregated into the consolidated business unit and can be reported in that way.

Note: Although the roll-up engines are usually the only job in a jobstream, they must still be set up in a jobstream. No merge is required on this engine.

Common Elements Used to Process Roll-Ups

Rerun

If you are rerunning this job with the same run control parameters, deselect the Rerun check box. This way, if you are assigned the same record suite and you were the last one to run in this record suite, the system does not re-resolve the tables prior to running the job. If you want to re-resolve the permanent tables, select the Rerun check box.

Re-resolving means that data is selected from the permanent table. If re-resolving is not needed (no new data in the permanent table for the given run control parameters), performance could be enhanced. Note that if the Rerun check box is selected, it will flatten (or reflaten) the business unit or the scenario tree.

Roll-Up Scenario Page

Use the Roll-Up Scenario page (RUN_PF_SCN_ROLL) to roll up scenarios to produce an aggregated scenario set of data based on the scenario tree. Generate your reports after you run the Scenario Roll-Up engine and the Calendar Roll-Up engine.

Navigation

EPM Foundation, Data Enrichment Tools, Roll-up Accounting Info, Scenarios, Roll-Up Scenario

Image: Roll-Up Scenario page

This example illustrates the fields and controls on the Roll-Up Scenario page. You can find definitions for the fields and controls later on this page.

Business Unit, Scenario ID, Fiscal Year, and Accounting Period

Enter the business unit, scenario ID, fiscal year, and accounting period for the roll-up.

Tree ID

You can select a tree. Ensure that the tree you select contains a SetID.

Job ID

Select *SCN_ROLL*.

ABM

Select this check box to roll up the scenarios in the tables associated with PeopleSoft Activity-Based Management AB_JOB_CD_F00, CALC_DETAIL_F00, and CALC_OBJ_F00 tables.

BSC

Select this check box to roll up the scenarios in the tables associated with PeopleSoft Scorecard KP_DATAVALS_F00 table.

PF Ledger

Select this check box to roll up to the PF_LEDGER_F00 table.

Click Run to run this request.

Roll-Up Calendar Page

Use the Roll-Up Calendar page (RUN_PF_CAL_RU) to roll up calendars to produce an aggregate set of product data based on calendars.

Navigation

EPM Foundation, Data Enrichment Tools, Roll-up Accounting Info, Calendar, Roll-Up Calendar

Image: Roll-Up Calendar page

This example illustrates the fields and controls on the Roll-Up Calendar page. You can find definitions for the fields and controls later on this page.

Business Unit, From Scenario ID, Scenario ID, Fiscal Year and Period Enter the business unit, from scenario ID, scenario ID, fiscal year, and accounting period for the roll-up.

Job ID Select *LED_ROLL*.

ABM Select this check box to roll up the calendars for PeopleSoft ABM.

PF Ledger Select this check box to roll up performance ledger calendars.

Click Run to run this request.

Roll-Up Business Unit Page

Use the Roll-Up Business Unit page (RUN_PF_BU_ROLL) to roll up business units to produce an aggregate business unit set of data based on the business unit tree.

Navigation

EPM Foundation, Data Enrichment Tools, Roll-up Accounting Info, Business Units, Roll-Up Business Unit

Image: Roll-Up Business Unit page

This example illustrates the fields and controls on the Roll-Up Business Unit page. You can find definitions for the fields and controls later on this page.

Roll-Up Business Unit

User ID: VP1 [Report Manager](#) [Process Monitor](#)

Run Control ID: BUROLL01

Program Name: PF_BU_ROLL When: Once

*Description:

*Business Unit: CORP1

*Scenario ID: ACTUAL

*Fiscal Year:

*Accounting Period:

Tree ID: ACT_TREE

*Job ID: BU_ROLL

Products

ABM BSC PF Ledger

Rerun

Business Unit, Scenario ID, Fiscal Year and Accounting Period

Enter the business unit, scenario ID, fiscal year, and accounting period for the roll-up.

Tree ID

You can select a tree. Ensure that the tree you select contains a SetID

Job ID

Select *BU_ROLL*.

ABM

Select this check box to roll up the business units for PeopleSoft ABM.

BSC

Select this check box to roll up the business units associated with PeopleSoft Scorecard.

PF Ledger

Select this check box to roll up business units for PF ledger.

Click Run to run this request.

Setting Up and Running Currency Conversion

This topic provides an overview of currency conversion process for the OWE and discusses how to:

- Define currency conversion rules.
- Create currency conversion rule sets.

- Associate a conversion rule set with a job.
- Run currency conversion.
- Review multicurrency conversion messages.

See [Understanding EPM Multiple Currency Processing Concepts](#).

Pages Used to Set Up and Run Currency Conversion

<i>Page Name</i>	<i>Definition Name</i>	<i>Navigation</i>	<i>Usage</i>
Currency Conversion Rule	PF_MC_RULE_DFN1	EPM Foundation, Data Enrichment Tools, Currency Conversion, Identify Rules, Currency Conversion Rule	Describe currency conversion rules.
Currency Conversion Rule Set	PF_MC_RULESET_TBL1	EPM Foundation, Data Enrichment Tools, Currency Conversion, Create Set of Rules, Currency Conversion Rule Set	Create currency conversion rule sets by grouping one or more conversion rules , creating different rules, and, if different constraints and currency code columns are used, combining rules in a set. You must create a rule set for each rule that you want to run.
Job Conversion Rule Set	PF_MC_JOB_TBL1	EPM Foundation, Data Enrichment Tools, Currency Conversion, Associate Rule Set to Job, Job Conversion Rule Set	Associate a conversion rule set with a job. A currency conversion rule set can be assigned to more than one job, but each job can contain only one currency conversion rule set.
Message Header	PF_ENGMSG_HEAD	EPM Foundation, Data Enrichment Tools, Currency Conversion, Review Conversion Messages, Message Header	Review process information and run control parameters. This component is used to review any engine messages that result from running a job or jobstream.
Message Detail	PF_ENGMSG_LOG	EPM Foundation, Data Enrichment Tools, Currency Conversion, Review Conversion Messages, Message Detail	Review engine message details.

Understanding Currency Conversion for the Analytical Applications

In PeopleSoft EPM, you can obtain profitability figures that include the effects of multiple currencies. The economic effects of transactions can be reconciled with accounting effects with greater accuracy. In addition, the effect of currency fluctuations (when analyzing similar business units or processes in different countries) is eliminated.

While data may be brought into EPM in many different currencies, the amounts must be converted to a single currency for each business unit in order for proper engine processing to occur. PeopleSoft EPM engines use these converted base amounts as input and output base amounts from their processes.

The Currency Conversion application engine can be used on any fact table containing the *from* and *to* currency code fields and the *from* and *to* amount fields (for example the FI_INSTR_F00 or REVENUE_F00 tables).

The Currency Conversion engine enables you to perform conversion on any fact table. However, only temporary tables defined for the fact tables are updated by the conversion engine. The Currency Conversion process should be run between two other jobs. For instance, run a job that populates the temporary table for the fact table, followed by the Currency Conversion engine process to update the temporary table, and then a job following the conversion that updates the actual fact table using the results from the temporary table.

Reports specific to currency processing are not included with PeopleSoft EPM. To view reports in a reporting currency other than your base currency, build a data mart from the PF_LEDGER_F00 table, migrate the selected data to the data mart, and run currency conversion on the data mart.

The Currency Conversion engine can be run either as a job within a jobstream, or it can be called as a function library from another PeopleSoft EPM engine.

Note: Because all ledger accounts, regardless of the engine processes, must go through the PF_JRNL_F00 before the edit and post process in order to populate to the PF_LEDGER_F00, all conversions for ledger accounts must be done on the PF_JRNL_F00 table (rather than directly on the LEDGER or PF_LEDGER tables).

Note: Any desired conversion on the LEDGER table (prior to the PeopleSoft Activity-Based Management engine using base amounts as input) must be done by the delivered ETL tool or by a configured solution created at implementation time.

Delivered Multicurrency Metadata

This table lists delivered multicurrency tablemaps, datamaps, and associated primary record names:

Tablemap/Datamap Name	Primary Record
MC_BALANCE	FI_IBAL_R00
MC_INSTR	FI_INSTR_F00
MC_EVENT	FI_IEVENT_R00
MC_POOLINS	FI_POOLINST_F00
MC_OPTION	FI_IOPTION_R00
MC_TRPOS	FI_TRPOS_F00
MC_PFJRNL	PF_JRNL_F00

Prerequisites

Before creating currency conversion rules and rule sets, and associating them with jobs, you must:

- Set up your currency metadata.
- Set up jobs with which you want to run OWE currency conversion.

Related Links

[Understanding EPM Multiple Currency Processing Concepts](#)

[Setting Up Job Metadata](#)

Currency Conversion Rule Page

Use the Currency Conversion Rule page (PF_MC_RULE_DFN1) to describe currency conversion rules.

Navigation

EPM Foundation, Data Enrichment Tools, Currency Conversion, Identify Rules, Currency Conversion Rule

Image: Currency Conversion Rule page

This example illustrates the fields and controls on the Currency Conversion Rule page. You can find definitions for the fields and controls later on this page.

Currency Conversion Rule

SetID: SHARE Currency Rule Code: KP_CONVERT

Currency Conversion Rule Find | View All First 1 of 1 Last

*Effective Date: 01/01/1900 *Status: Active

*Description: KPI Currency Conversion

Rate Type: As Of Dated

*Constraint Code: KP_CONVERT

*From Currency Code Column: FROM_CURRENCY

*To Currency Code Column: BASE_CURRENCY

DataMap Measures Customize | First 1-3 of 3 Last

*From Amount Column	*To Amount Column		
(Invalid Value)	(Invalid Value)	+	-
RESOLVED_VALUE	CNVRT_VALUE	+	-
(Invalid Value)	(Invalid Value)	+	-

Rate Type

Select the exchange rate type to use for this rule. The type that you specify overrides any rate type specified by the business unit. If you don't select a rate type, the system uses the rate type specified for the business unit.

As of Dated

Select to use the as of date. If you do not select this check box, the system uses the period end date (based on the jobstream run parameters) as the effective date for the exchange rate. If you select this check box, any job running with this currency

	conversion rule uses the as of date to determine the exchange rate.
Constraint Code	Specify a constraint code. Constraints are defined using the Constraint component.
From Currency Code Column	Select the appropriate column from your constraint code. Any field that is <i>not</i> defined in the constraint as a measure is a valid value.
To Currency Code Column	Select the appropriate column from your constraint code. Any field that is <i>not</i> defined in the constraint as a measure is a valid value.
From Amount Column	Select the appropriate column. Any field defined in the constraint as a measure, and that has 1 ? 3 decimal places, is a valid value.
To Amount Column	Select the appropriate column. Any field defined in the constraint as a measure, and that has 1 ? 3 decimal places, is a valid value.

Note: If your amount columns have more than four decimal places, you must define a view for this table to redefine your amount fields. You then use this view to build your metadata and ultimately the constraint that is used as input to the multicurrency rule.

You can add more rows. You must enter field names for at least one *from* and one *to* currency code and at least one *from* and one *to* amount column. You can enter multiple *from* and *to* amount column field names for constraints that contain multiple amount fields, such as the Instrument table (FI_INSTR_F00). After you have set up currency conversion rules, you can create a conversion rule set.

Note: For each rule that you create, you must establish multicurrency tablemaps (identified as target tables), multicurrency datamaps (with amounts specified as measures), and multicurrency constraints. PeopleSoft EPM is delivered with standard table and datamaps for multicurrency. They can be identified by the MC_ prefix.

Currency Conversion Rule Set Page

Use the Currency Conversion Rule Set page (PF_MC_RULESET_TBL1) to create currency conversion rule sets by grouping one or more conversion rules , creating different rules, and, if different constraints and currency code columns are used, combining rules in a set. You must create a rule set for each rule that you want to run.

Navigation

EPM Foundation, Data Enrichment Tools, Currency Conversion, Create Set of Rules, Currency Conversion Rule Set

Image: Currency Conversion Rule Set page

This example illustrates the fields and controls on the Currency Conversion Rule Set page. You can find definitions for the fields and controls later on this page.

The screenshot shows the 'Currency Conversion Rule Set' page. At the top, it displays 'SetID: SHARE' and 'Rule Set: KP_CONVERT'. Below this is a header for 'Currency Conversion Rule Set' with navigation options like 'Find', 'View All', 'First', '1 of 1', and 'Last'. The main form contains fields for '*Effective Date:' (01/01/1900), '*Status:' (Active), and '*Description:' (KPI Currency Conversion). Below the form is a table for 'Currency Conversion Rule' with columns for '*Expense Sequence Number' (10) and '*Currency Rule Code' (KPI Currency Conversion).

Expense Sequence Number

Enter a number to control the order in which the Currency Conversion engine runs your rules. To accommodate the addition of future currency rules, use numbers such as 100, 200, 300.

Currency Rule Code

Select the name of the rule that you created.

Note: Each currency rule must be in a rule set. You can also use the rule set to combine multiple currency rules.

After you have set up the currency conversion rule set, you need to associate the rule set with a job.

Job Conversion Rule Set Page

Use the Job Conversion Rule Set page (PF_MC_JOB_TBL1) to associate a conversion rule set with a job. A currency conversion rule set can be assigned to more than one job, but each job can contain only one currency conversion rule set.

Navigation

EPM Foundation, Data Enrichment Tools, Currency Conversion, Associate Rule Set to Job, Job Conversion Rule Set

Image: Job Conversion Rule Set page

This example illustrates the fields and controls on the Job Conversion Rule Set page. You can find definitions for the fields and controls later on this page.

The screenshot shows a web form titled "Job Conversion Rule Set". At the top, there are two buttons: "SetID: SHARE" and "Job ID: EDIT". Below these is a search bar with "Job Conversion Rule Set" entered. The main form contains the following fields:

- *Effective Date: 01/01/1900 (with a calendar icon)
- *Status: Active (dropdown menu)
- *Description: Currency Conversion (text input)
- *Currency Rule Set: KP_CONVERT - KPI Currency Conversion (dropdown menu)

Job ID

Enter the job with which you want to associate this rule set. You create jobs on the Job Metadata page. When you run this job in a jobstream, the system processes the currency conversion rules based on the rule set associated with the job.

Currency Rule Set

Select the rule set that you created. After you have created a job rule set, you can run the Currency Conversion engine using the rules you have set up.

Note: A currency conversion rule set can be assigned to more than one job, but each job can contain only one currency conversion rule set.

Running the Currency Conversion Process

After you have completed this setup and set up your jobstreams, the Currency Conversion engine runs as part of the jobs that you defined. The engine ensures that all transactions are coded with the information that the system needs to manage multiple currency transactions. The timing of running the Currency Conversion process varies based on your business processes and the PeopleSoft EPM engines involved in your processes. Because most engines use the base currency amounts in their calculations, you must run the Currency Conversion process prior to the other engine processes for the proper base currency amounts to be available for engine processing.

The Currency Conversion engine affects any record in the warehouse that contains currency code and amount fields.

Related Links

[Delivered Multicurrency Metadata](#)

Message Header Page

Use the Message Header page (PF_ENGMSG_HEAD) to review process information and run control parameters.

Navigation

EPM Foundation, Data Enrichment Tools, Currency Conversion, Review Conversion Messages, Message Header

Consider the following when you review dependencies:

- All amount fields for a record must be migrated to the warehouse in a single currency.
For example, you cannot have instrument balance in USD and payment amounts in MXP.
- Multicurrency tablemaps must be designated as *target* tablemaps.
- Multicurrency datamaps must use the multicurrency tablemaps.
- Amount fields on multicurrency datamaps must be designated as *measures*, because the prompt table for these amount fields on the rule page uses a view that filters out anything that is not designated as a measure.
- Any constraints used on the rules must be specific to the multicurrency datamaps.
- Jobs that are to be run using the Currency Conversion engine must have one conversion rule set defined, and each rule set must contain at least one rule.

Note: These pages are described in the [Understanding Jobstreams](#) topic.

Setting Up Models and Scenarios

Understanding Models and Scenarios

Models enable you to define how to measure performance within your organization and the types of data that you want to review through various reports. For example, you may want to review revenue information on a region-by-region basis—a very high-level scope. Or, you may want to review only those activities that relate to a certain product line for certain types of resources—a very narrow scope. Each analytical application uses models and scenarios differently.

Regardless of the size or scope of your model, you must *initially* proceed with the same set of steps. Likewise, when you set up models you perform setup steps that are common to all products in the PeopleSoft EPM product line. However, how you complete the process depends on which EPM products you license.

Note: You should refer to your application-specific documentation for more information on models and scenarios in your product.

Modeling Terminology

When using models, you should familiarize yourself with the following terms:

- Parent model: The master model that is used as the basis for an impact study.
- Child model: A child model stores the changes to the master or parent model.

The child model inherits all the properties from the parent model.

- Scenario: You create a scenario ID for each parent and child model that you want to study.

This scenario ID is used on all run controls.

Object-Based Modeling

Object-based modeling enables you to simulate various changes in an organizational model and study the impact on costs, revenue, and profitability. Instead of copying the whole model, you can copy only the data that needs to be changed. This reduces the volume of data copied and helps to identify the changes made and any effects the changes may have. Object-based modeling helps to define just the changes for the child model. All the unchanged data for the child model is inherited from the parent model. Enhanced modeling features improve the maintenance, reusability, and flexibility of a model, enabling you to create parent and child models easily.

Object-based modeling can be used not only in PeopleSoft Activity-Based Management but also in other applications in PeopleSoft EPM so that you can use your system to do planning and simulation. Scenarios for planning and simulation can be defined as child models. Child models represent the business decisions and assumptions for the scenarios. Scenarios can be simulated for each child model using various forecast

distributions for a given timespan. Results for various scenarios can be compared to select the best case scenario.

Scenarios

Once you establish warehouse business units and SetIDs, you create model IDs (where you define the parent and child relationships) and then point to those model ID by means of a scenario ID. Scenarios:

- Serve as a wrapper to run all analytic models.

When you run the analytical application engines, the engines tie the models together by means of the scenario ID.

- Enable you to work with what-if scenarios by creating more than one scenario with different models attached for what-if comparisons.

There are two types of scenarios:

- Historical.
- Forecast.

To set up scenarios:

1. Complete the Scenario Definition page.
2. Select the type of scenario (historical or forecast) and complete the appropriate pages.
3. Specify the rates that you want to use on the Economic Assumptions page.
4. Assign the scenarios to a business unit using the PF Unit Scenario Definition page.

Defining Models and Scenarios

This topic discusses how to:

- Create model definitions.
- Set up historical scenarios.
- Enter historical scenario business rules.
- Enter historical scenario economic assumptions.
- Set up forecast scenarios.
- Enter forecast scenario business rules.
- Enter forecast scenario economic assumptions.
- Assign scenarios to business units.

Pages Used to Define Models and Scenarios

Page Name	Definition Name	Navigation	Usage
Models	PF_MODEL_TBL1	EPM Foundation, Business Metadata, Business Framework, Models	Create a model definition.
Scenarios - Definition	PF_SCENARIO_DFN1	EPM Foundation, Business Metadata, Business Framework, Scenarios, Definition	Create a scenario definition.
Scenarios - Historical Business Rules	PF_SCENARIO_DFN2	EPM Foundation, Business Metadata, Business Framework, Scenarios, Historical Business Rules	For an historical scenario, link to a default model ID for the scenario.
Scenarios - Forecast Business Rules	PF_SCENARIO_DFN5	EPM Foundation, Business Metadata, Business Framework, Scenarios, Forecast Business Rules	For a forecast scenario, specify a fiscal year and accounting period for all model IDs that you want to use.
Scenarios - Economic Assumptions	PF_SCENARIO_DFN3	EPM Foundation, Business Metadata, Business Framework, Scenarios, Economic Assumptions	Specify the rate type for your scenario.
Scenarios - Notes	PF_SCENARIO_DFN6	EPM Foundation, Business Metadata, Business Framework, Scenarios, Notes	Describe the scenario's purpose in greater detail.
Warehouse Business Unit Scenario Definition	PF_BU_SCENARIO_DFN	EPM Foundation, Business Metadata, Business Framework, WBU Scenario Definition, Warehouse Business Unit Scenario Definition	Assign the scenarios you have defined to the appropriate business units.

Models Page

Use the Models page (PF_MODEL_TBL1) to create a model definition.

Navigation

EPM Foundation, Business Metadata, Business Framework, Models

Image: Models page

This example illustrates the fields and controls on the Models page. You can find definitions for the fields and controls later on this page.

The screenshot shows the Oracle EPM 'Models' page. At the top, it displays 'SetID: SHARE' and 'Model ID: HR'. Below this is a table with one row. The table has columns for 'Effective Date' (01/01/1900), 'Status' (Active), 'Description' (HR), 'Notes' (HR Model), and 'Parent Model ID' (HHC01). The table is titled 'Models' and has a search bar and navigation controls (First, 1 of 1, Last).

Effective Date	Status	Description	Notes	Parent Model ID
01/01/1900	Active	HR	HR Model	HHC01

Parent Model ID

For a child model that rolls up to another model, specify that ID of the parent model in this field.

Scenarios - Definition Page (for Historical Scenarios)

Use the Scenarios - Definition page (PF_SCENARIO_DFN1) to create a historical scenario definition.

Navigation

EPM Foundation, Business Metadata, Business Framework, Scenarios, Definition

Image: Scenario - Definition page

This example illustrates the fields and controls on the Scenario - Definition page.

The screenshot shows the 'Scenario - Definition' page with the following fields and controls:

- Tabs: Definition (selected), Historical Business Rules, Economic Assumptions, Notes
- SetID: SHARE, Scenario ID: ACTUAL
- Details section with search and navigation controls: Find, View All, First, 1 of 1, Last
- *Effective Date: 01/01/1900 (calendar icon)
- *Status: Active (dropdown menu)
- *Description: Business Planning Actual (text input)
- *Input Scenario ID: ACTUAL (text input with search icon)
- *Scenario Type: Historical (dropdown menu)
- Calendar ID: (dropdown menu)
- Prior Scenario ID: (text input)
- Forecast Group ID: (dropdown menu)
- Start Date: (text input)
- Number of periods: (text input)
- Consolidated

Input Scenario ID

By default, the input scenario ID is the same as the scenario ID. When you run a jobstream, Resolver uses the input scenario ID associated with the run control scenario ID to select data from a scenario-keyed table. This way, you can access data from a scenario other than the one you provide at run time.

Scenario Type

Select the *Historical* scenario type.

The component displays the Historical Business Rules tab.

Note: Depending on the scenario type you select, certain fields appear and others are hidden on all tabs within this component.

Consolidated

Select this check box to identify this scenario as one to which other scenarios will roll up.

Scenarios - Definition Page (for Forecast Scenarios)

Use the Scenarios - Definition page (PF_SCENARIO_DFN1) to create a forecast scenario definition.

Navigation

EPM Foundation, Business Metadata, Business Framework, Scenarios, Definition

Input Scenario ID

By default, the input scenario ID is the same as the scenario ID. When you run a jobstream, Resolver uses the input scenario ID associated with the run control scenario ID to select data from

a scenario-keyed table. This way, you can access data from a scenario other than the one you provide at run time.

Scenario Type

Select the *Forecast* scenario type. The component displays the Forecast Business Rules tab.

Note: Depending on the scenario type you select, certain fields appear and others are hidden on all tabs within this component.

Calendar ID

Select the appropriate calendar ID. The calendar ID is used as a basis for the Prior Scenario ID and Forecast Group ID fields.

Note: There is a relationship between the calendar ID on this page and the calendar ID on the PF Unit Scenario Definition page. If an entry has been made in the PF Unit Scenario Definition page that associates a calendar ID to a scenario and business unit combination, you cannot update the Scenario Definition page calendar ID (it is disabled). If the Scenario Definition Calendar ID needs to be changed, the corresponding row for the scenario must be deleted from the PF Unit Scenario Definition page first. This will enable the Calendar ID field on the Scenario Definition page.

Prior Scenario ID

Select the appropriate prior scenario ID. Only previously created scenario IDs with the same calendar ID are available for selection. The system uses this scenario ID as input to PeopleSoft Activity-Based Management. A prior scenario ID is required if you are specifying a model for the Financial Statement Simulator on the Forecast Business Rules page.

Forecast Group ID

Select the appropriate forecast group ID for the scenario. Only previously created forecast group IDs with the same calendar ID are available. Forecast group IDs are defined in the PeopleSoft Analytic Forecasting module.

Start Date

Enter the start date for the scenario. For example, if you specify a monthly calendar with a number of periods equal to 2 and specify a start date of July 12, 2002, then the system populates the next page with periods 7 and 8.

Number of Periods

Enter the number of periods related to your start date. This defines the dates in the grid on the Forecast Business Rules page.

Consolidated

Select this check box to identify this scenario as one that other scenarios drill up to.

Scenarios - Historical Business Rules Page

Use the Scenarios - Historical Business Rules page (PF_SCENARIO_DFN2) for an historical scenario, link to a default model ID for the scenario.

Navigation

EPM Foundation, Business Metadata, Business Framework, Scenarios, Historical Business Rules

Image: Scenarios - Historical Business Rules page

This example illustrates the fields and controls on the Scenarios - Historical Business Rules page. You can find definitions for the fields and controls later on this page.

Model Name	Value
Effective Date:	01/01/1900
Status:	Active
Default Model ID:	PROD1
ABM Model:	PROD1
FIN Model:	
Workforce Analytics:	PROD1
KPI Model:	PROD1
GLMP Model:	PROD1
CRM Model:	PROD1
Consol Model:	PROD1

Default Model ID

If you specify a default model ID, then the rest of the models listed default to that ID. Otherwise you can select specific models for the various applications

ABM Model

Model ID to use for your PeopleSoft Activity-Based Management application.

FIN Model

Model ID to use for financial analytics.

Workforce Analytics

Model ID to use for your Workforce Analytics applications.

KPI Model

Model ID to use for Key Performance Indications and scorecards.

GLMP Model

Model ID for the ledger mapper.

CRM Model

Model ID to use for your CRM analytics applications.

Console Model

Model ID to use for consolidations.

Scenarios - Economic Assumptions Page (for Historical Scenario)

Use the Scenarios - Economic Assumptions page (PF_SCENARIO_DFN3) to specify the rate type for your scenario.

Navigation

EPM Foundation, Business Metadata, Business Framework, Scenarios, Economic Assumptions

Image: Scenarios - Economic Assumptions page

This example illustrates the fields and controls on the Scenarios - Economic Assumptions page. You can find definitions for the fields and controls later on this page.

Rate Type

Select from the following:

Market Rates: Use current interest rate information.

Deterministic Rates: Enables you to select a hypothetical interest rate environment previously created within PeopleSoft Asset Liability Management. Deterministic rates play an integral role in modeling a financial services institution's exposure to interest rate risk. They enable you to model dynamic interest rate environments explicitly. If you define a scenario with either a deterministic rate assumption or a stochastic one, then you must select the term structure model ID to accompany the scenario.

Stochastic Rates: Scenarios with this rate type are used to randomly generate interest rate scenarios, and they are used in risk management. If define a scenario with either a deterministic rate assumption or a stochastic one, then you must select the term structure ID to accompany the scenario.

Age-Graded Table: Used for PeopleSoft Workforce Analytics.

The age-graded table includes effective dates of rate and rating factors (such as age ranges, sex, and smoking status) by employer and employee. The arrays of values used to calculate the cost of a plan to an employee. Rates can be age-graded, service-related, or general, depending upon the benefit plan type.

Flat Rate Table: Contains information on where you define rates to be charged per selected frequency for a particular benefit program or plan. The Flat Rate table represents a constant value.

For all subsequent years, as well as the first time it is done, flat rate is calculated simply by applying the flat rate percentage

to the average balance and allocating this amount among the periods.

Salary Percent Table: Used in PeopleSoft Planning and Budgeting, this table contains information related to earnings as a percentage of the base or gross salary amount.

Service Rate Table: Used in PeopleSoft Workforce Analytics, this table stores information such as service rate ID, effective date, pay frequency, rate per unit, service intervals, total rate, employer portion, and employee portion. Service rates are employee related rates. For example, a service rate would be an employee's days of vacation accrued per year or per month for a given vacation plan.

Note: You create term structure model IDs as part of your setup for using PeopleSoft FSI applications.

Scenarios - Forecast Business Rules Page

Use the Scenarios - Forecast Business Rules page (PF_SCENARIO_DFN5) for a forecast scenario, specify a fiscal year and accounting period for all model IDs that you want to use.

Navigation

EPM Foundation, Business Metadata, Business Framework, Scenarios, Forecast Business Rules

Enable Calendar Fill In

This is selected by default. The forecast scenario creates a start date on the exact date specified within the first period.

Depending on your selections, there could be time overlap or gaps. Calendar fill in helps avoid these gaps. Calendar fill in adds model IDs to blank rows for the grid lines not selected.

Specify the model IDs you want to use for each year and period. The selections you make depend on the analytical applications you have installed.

Note: You must specify a prior scenario ID on the Scenario Definition page to select a model of Financial Statement Simulator.

Scenarios - Economic Assumptions Page (for Forecast Scenario)

Use the Scenarios - Economic Assumptions page (PF_SCENARIO_DFN3) to specify the rate type for your scenario. The fields on this page are the same as those discussed for the Historical Forecast - Economic Assumptions page.

Navigation

EPM Foundation, Business Metadata, Business Framework, Scenarios, Economic Assumptions

Warehouse Business Unit Scenario Definition Page

Use the Warehouse Business Unit Scenario Definition page (PF_BU_SCENARIO_DFN) to assign the scenarios you have defined to the appropriate business units.

Navigation

EPM Foundation, Business Metadata, Business Framework, WBU Scenario Definition, Warehouse Business Unit Scenario Definition

Image: Warehouse Business Unit Scenario Definition page

This example illustrates the fields and controls on the Warehouse Business Unit Scenario Definition page. You can find definitions for the fields and controls later on this page.

Warehouse Business Unit Scenario Definition

Business Unit: CORP1

Business Unit - Scenario Definition Find | View All | First 1 of 20 Last

*Scenario ID: ACTUAL01

*Calendar ID: 01

Calendar Type: Detail Calendar

Manual Journal Approvals

*Manual Journal Approval Option: Pre-Approved

Business Process Name: BC_PRJ_APPROVAL

Approval Rule Set:

Scenario ID For each business unit, select the appropriate scenario ID.

Calendar ID For each business unit, select the appropriate calendar ID.

Note: There is a relationship between the calendar ID on this page and the calendar ID on the Scenario Definition page. If you make an entry on this page that associates a calendar ID to a scenario and business unit combination, you can no longer change the calendar ID on the Scenario Definition page (the field is disabled).

Manual Journal Approval

Journal approvals allow specific requirements to be met before the approval is finalized. The system defaults to *Pre-Approved* for the Manual Journal Approval option. The *Approval Required - Security* and *Approval Required - Workflow* options are used by PeopleSoft Global Consolidations.

Business Process Name

If you are using the approval workflow option in PeopleSoft Global Consolidations, specify the business process name for the approval workflow.

Approval Rule Set

If you are using the approval workflow option in the PeopleSoft Global Consolidations analytical application, specify the approval rule set to be used for the selected business process.

See PeopleSoft EPM: Global Consolidations

Streamlining Processing with Jobstreams

Understanding Jobstreams

This topic discusses:

- Jobstreams.
- Jobstream terminology
- Jobstream processing.
- Jobstream record suites.
- Jobstream chunking.
- Spawn process.
- Resolver engine.
- Resolver and chunking.
- Process monitor.

Jobstreams

To help streamline your processing, PeopleSoft provide jobstreams which use temporary tables for intermediate processing. Jobstreams enable different users to run their own jobs using instances of the same processing engines at the same time. Jobstreams enhance performance by sharing temporary tables passed between jobs.

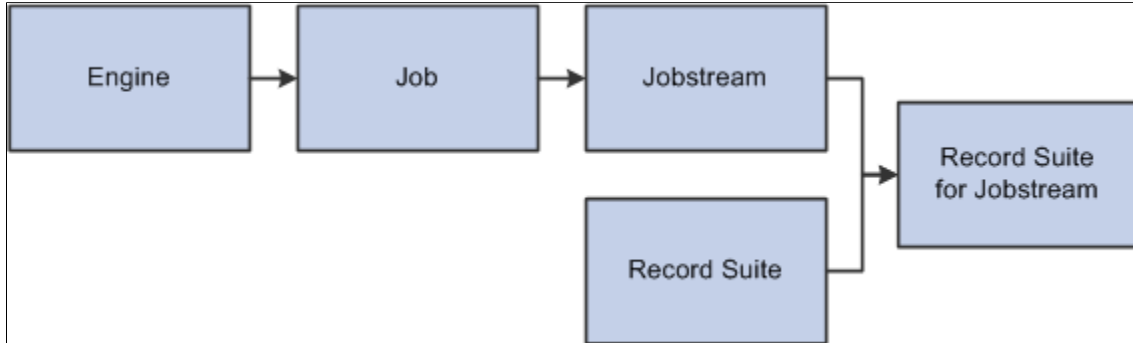
Instead of locking up the fact (primary input) tables, jobstreams use temporary tables for intermediate processing. A set of delivered temporary tables, referred to as a record suite, is assigned when the first job of a jobstream is run, and then the tables are released when the last job of a jobstream is completed. The use of record suites frees up the fact tables so that another user can access them and run a concurrent job. Each job then has its own record suite for a jobstream.

There are several steps involved in setting up a jobstream. PeopleSoft delivers predefined processing engines and engine metadata, jobs and job metadata, jobstreams, and record suites. If you use the

predefined metadata, the only item that you have to specify before you run an engine is one or more record suites for each jobstream for a given SetID.

Image: Jobstream overview

This diagram illustrates the components that make up a jobstream:



Jobstreams work by creating a copy of the processing engine. When you run a jobstream, you can:

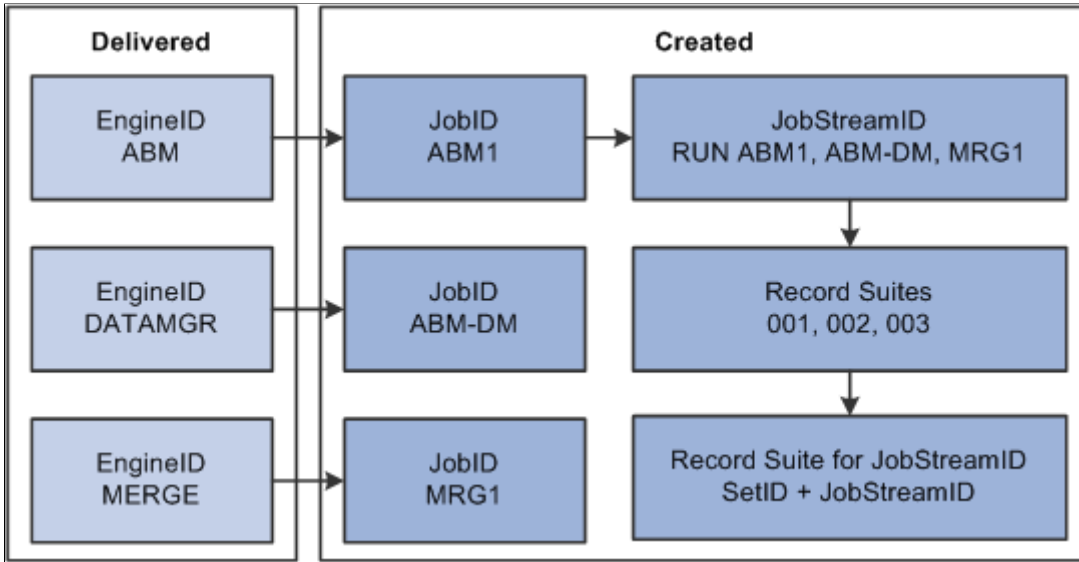
- Run multiple engines sequentially in one jobstream.
- Run each individual engine in its own jobstream.
- Run one sequential jobstream for multiple fiscal years or accounting periods.

Suppose you want to run the Activity-Based Management engine, Data Manager engine, and Merge engine at the end of the business day. You can select a jobstream that runs all three engines automatically. Each engine runs sequentially and populates specific temporary tables, with the Merge engine transferring the data from the temporary tables back to the fact tables. You only need to initiate the jobstream, no further action is required.

The Merge (PF_MERGE) engine merges the output temporary tables into the final tables for use as input for other processes. PF_MERGE is the last job in all jobstreams except when the POST job is run at the end of a jobstream.

Image: Jobstream setup

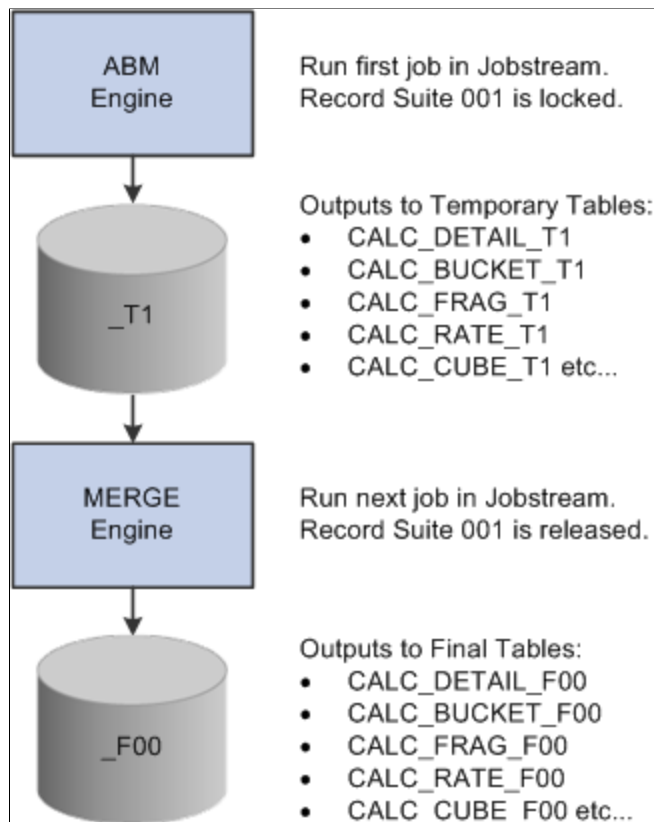
This diagram illustrates how jobstream setup works:



In the above diagram, note that the engine IDs on the left side of the illustration are delivered with PeopleSoft EPM. You create the job and jobstream IDs, and then assign record suites to the jobstream.

Image: Merge engine process

This diagram illustrates how the Merge engine moves output from the Activity-Based Management (ABM) engine to the into final fact tables:



Jobstream Terminology

The following terms apply to jobstreams in PeopleSoft EPM applications:

- | | |
|------------------------|--|
| Engine Metadata | Identifies the PeopleSoft application engine process that you want to run by engine ID. |
| Job Metadata | Enables you to create an instance or copy of a PeopleSoft application engine program to use in your jobstream. |
| Jobstream | Enables you to combine job IDs into a jobstream ID to pass data from one job to the next. Every job you run must be in a jobstream, whether it is a combination of sequential jobs or a single job. All jobs in the jobstream must be run for the same parameters (business unit, scenario, fiscal year, and so on). |
| Record Suites | Consist of a group of temporary tables with the same temp table append for processing instances of an application engine. Temporary tables are used during engine processing to free up the original fact tables so other users can access them and |

provide faster run times. A jobstream has exclusive use of the record suite during its execution, so there is no table locking or sharing.

For example, you might create a jobstream to first run the ABM engine, then the Data Manager engine, and finally run the Merge process. Each engine runs sequentially and populates specific temporary tables, with the Merge engine transferring the data from the temporary tables back to the fact tables.

Jobstream Record Suites

Assign record suites to the jobstream ID you create (using the appropriate SetID). The SetID used must be the same SetID assigned to the run control parameter business unit under record group PF_03.

Jobstream Processing

When you run a jobstream, the following occurs:

- Record suite 001 is locked and the first engine runs placing its output into the appropriate temporary table in record suite 001.
- The next engine runs getting its input from the temporary tables generated by the first job and putting its output into other temporary tables in record suite 001.
- The last job in the jobstream (either PF_MERGE or PF_POST) reads the data in the temporary tables, merges it, and then writes the output to the final target tables. The system releases the record suite.

Note: Not all processes use jobstreams. For example, PF_SUMM does, but PF_MODIFICATION does not. Exceptions are noted in the documentation as appropriate.

Jobstream Record Suites

When you initially process a jobstream, the system checks which record suites are assigned to the specified SetID and jobstream ID.

This process enables you to reserve a set of record suites for a specific type of engine. For example, if record suites 001 and 002 are assigned to PeopleSoft Activity-Based Management (ABM) jobstreams, and 003 is authorized for PeopleSoft Asset Liability Management (ALM), then PeopleSoft ALM jobstreams never compete for record suite availability with ABM jobstreams.

Jobstream Chunking

Chunking is a mechanism that enables you to select a smaller chunk of data for further processing and to parallel process data in multiple chunks. It enables you to horizontally partition source data so that only a subset of data is processed by an engine. This enables users to run multiple engines with different criteria and to run them in parallel to reduce the processing time.

During a jobstream run, chunking occurs when technical scenario is associated with the run scenario based on the scenario selected on the Technical Scenarios page. After a jobstream identifies that chunking has been requested, the jobstream initiates the PF_CHUNK application engine program to process each

chunking selection. The jobstream then invokes a parallel application engine PF_SPAWN to process each chunking selection. This program spawns a job for each chunking definition.

The number of jobs that can be spawned in parallel is restricted to the number of available record suites. You require one record suite for the jobstream process and one for each of the spawned processes.

Because all jobs that use the same technical scenario may not require chunking, the decision to chunk is based on the chunking selection in the engine metadata and chunking criteria specified on the Technical Scenarios page.

Spawn Process

The *spawn* application engine process (PF_SPAWN) provides greater control over jobstream processing by enabling jobs to be launched as needed.

Spawn Process Tables

PF_SPAWN creates the following tables to store data while the jobstream runs:

- The PF_SPWN_JOB_TBL table stores all required information about spawned jobs.
Entries in this table are deleted once all spawned jobs are complete.
- The PF_SPWN_CTL_T temporary table passes the run control parameters for each spawned job.
Use the sequence number field to control the order of the spawned jobs.
- The PF_SPWN_CRIT_T table stores all of the required spawn criteria for each of the spawned jobs.

Resolver Engine

The Resolver engine further enhances and increases application performance by reducing the amount of data an engine needs to process. You do not invoke the Resolver (PF_ENG_PROC.RESOLVE) engine. The system invokes it automatically as part of startup processing to resolve all records and tablemaps specified in the engine metadata for an engine. An application needs to explicitly invoke the Resolver engine to resolve datamaps, filters, constraints, and data sets referenced in business rules.

Resolution occurs on the SetID, business unit, scenario ID, effective date, as of date, fiscal year, and accounting period. The resolution process only moves the data that matches the run control values from the table to the associated temporary table as defined in the record metadata. The engine works only on the data in the temporary table.

Individual engines call the Resolver engine as part of their run processes.

Note: The Resolver engine is limited to resolving tables for only one value at a time. For example, it cannot resolve for multiple business units.

Resolver and Chunking

As part of chunking, the Resolver engine applies chunk criteria to the record that is being chunked based on the criteria defined on the Technical Scenarios page. When the system invokes the Resolver engine, it checks all the records that it needs to resolve to see if the record requires chunking. If this is the case,

the Resolver engine checks the record to see if the chunk field exists in the record. If the field exists, the system appends chunk criteria to the resolver query for this record.

Below is an example of chunk criteria:

```
(CUST_ID IN (SELECT CUST_ID FROM PS_CUSTiINTFC_F00 WHERE CUST_ID BETWEEN ('1000','10000')))
```

In this case the CUST_ID is the chunk field that exists in the record being resolved.

Process Monitor

During job processing, use Process Monitor to review the status of reports and processes. You can monitor process requests, server status, and the status of any job in the queue. If there are messages related to a process, you can view them from Process Monitor, as well. For example, if a process encounters an error, or if a server is down, you can find out almost immediately.

See *PeopleSoft PeopleTools: PeopleSoft Process Scheduler*

Failed Jobstreams and the Process Monitor

When a jobstream fails, one of the following status messages appear:

- *Error*: Indicates that the program that is associated with the process request encountered an error while processing transactions within the program. In this case, delivered programs are coded to update the run status to Error before terminating.
- *No Success*: Indicates that the program encountered an error within the transaction. No Success is different from Error because the process is marked as restartable.
- *Success With Application Error*: Indicates that a jobstream has completed, but with an application error. For example, a jobstream may result in an application error due to unavailable record suites.

Setting Up Chunking

This topic discusses how to:

- Establish chunking in the engine metadata.
- Set up chunking criteria.

Pages Used to Set Up Chunking

Page Name	Definition Name	Navigation	Usage
Engine Metadata - State Variables	PF_META_ENG_TBL2	EPM Foundation, Job Processing, Setup Engines and Jobstreams, Engine Metadata, State Variables	Specify records to be chunked during the resolve process.

Page Name	Definition Name	Navigation	Usage
Engine Metadata - Source TableMaps	PF_META_ENG_TBL4	EPM Foundation, Job Processing, Setup Engines and Jobstreams, Engine Metadata, Source TableMaps	Specify source tablemaps to be chunked during the resolve process.
Technical Scenario	PF_CHUNK_DFN	EPM Foundation, Business Metadata, Business Framework, Technical Scenarios	Set up the object type values that the Resolver uses to chunk the record or tablemap you are resolving.

Engine Metadata Pages

Use the Engine Metadata - State Variables page (PF_META_ENG_TBL2) to specify records to be chunked during the resolve process.

Navigation

EPM Foundation, Job Processing, Setup Engines and Jobstreams, Engine Metadata, State Variables

Use the Engine Metadata - Source TableMaps page (PF_META_ENG_TBL4) to specify source tablemaps to be chunked during the resolve process.

Navigation

EPM Foundation, Job Processing, Setup Engines and Jobstreams, Engine Metadata, Source TableMaps

To set up chunking you access the Engine Metadata - State Variables or Engine Metadata - Source TableMaps pages to set up application engine parameters. On the State Variables page, you specify the records to be chunked during the resolve process. Alternatively, on the Source TableMaps page you can specify the tablemaps to be chunked. The [Working with Engine Metadata](#) topic details the pages in the Engine Metadata component on which you define this setup.

As the next step, go to the Technical Scenarios page to set up the chunking criteria. Technical scenarios enable you to set up the object type values that the Resolver uses to chunk the records and tablemaps you are resolving. You establish the records or tablemaps to resolve on the Engine Metadata - State Variables and Engine Metadata - Source TableMaps pages.

Technical Scenario Page

Use the Technical Scenario page (PF_CHUNK_DFN) to set up the object type values that the Resolver uses to chunk the record or tablemap you are resolving.

Navigation

EPM Foundation, Business Metadata, Business Framework, Technical Scenarios

Image: Technical Scenarios page

This example illustrates the fields and controls on the Technical Scenarios page. You can find definitions for the fields and controls later on this page.

Scenario ID

Select the scenario ID to which to link this technical scenario. When a jobstream runs for this scenario, the chunking is invoked.

Technical Scenario ID

Enter the technical scenario ID to identify the chunking selection. You may enter as many chunk codes as you like for each scenario. Each chunk code may have different chunk criteria.

Chunk By

Select the object type to use for chunking. You must select the object type you selected in the Engine Metadata component.

Note: You can apply only one chunking criteria to a tablemap. The system does not support duplicate object types.

Constraint Code

Select the constraint that filters the values for your objects at run time.

DataMap Column

Select one column from the datamap of the constraint. This should be, but is not restricted to, the column that matches the object type you have selected. For example, if your chunk object type is *PRODUCT ID*, then your datamap column from the constraint may be *PRODUCT_ID* or *PRODUCT_TYPE*. You may copy chunking criteria to another effective date for the same scenario only. You need to be very careful about defining chunk criteria. You need to make sure to define your chunks to cover a complete set of data without any duplicates.

Note: If you delete a scenario with chunking criteria, the chunking criteria are automatically deleted. You can enter multiple technical scenarios to process data in multiple chunks. Remember, that all technical scenarios should select a mutually exclusive data set. The data is also not necessarily processed in the order defined.

Warning! If you enter an invalid or duplicate constraint code, the jobstream abends at run time.

You have established the chunk objects and the chunking criteria. Chunking initiates when there is a technical scenario associated with the run scenario.

For chunking to be successful, you must ensure that:

- The object type selected in the Engine Metadata component matches the one in the technical scenario used by the engine. This initiates the chunking process.
 - The column you select for the chunking criteria on the Technical Scenarios page must be in the record or any record of a tablemap you have selected for chunking. This completes the chunking process.
-

Note: You can only resolve (chunk) a record once in an engine. The system resolves tablemaps first. Any records resolved as part of the tablemap are not resolved again in the state record.

Working with Engine Metadata

PeopleSoft EPM delivers predefined engine metadata. Unless you create your own application engine processes as part of your implementation, you do not need to create any engine metadata. However, you do need to create different instances of the delivered engines to enable parallel processing, described in the [Setting Up Job Metadata](#) topic.

In most cases, you only use the pages described in this topic to review and modify the delivered engine metadata. You may also use them to define chunking selections.

If you do add an application engine process, use the pages described to add the engine metadata for the new application engine.

This topic discusses how to:

- Review, modify, or add engine metadata.
- Review, modify, or add state variables.
- Define rule state variables.
- Specify source tablemaps.
- Specify source trees.

Pages Used to Work with Engine Metadata

Page Name	Definition Name	Navigation	Usage
Engine Metadata	PF_META_ENG_TBL1	EPM Foundation, Job Processing, Setup Engines and Jobstreams, Engine Metadata	Review delivered engine metadata and modify if required. Add new metadata for a new application engine process. Unless you create your own application engine process, you do not need to create any engine metadata.
Engine Metadata - State Variables	PF_META_ENG_TBL2	EPM Foundation, Job Processing, Setup Engines and Jobstreams, State Variables	Review state variables modify if required. These state variables enable dynamic changes to application engine inputs.
Engine Metadata - Rule State Variables	PF_META_ENG_TBL3	EPM Foundation, Job Processing, Setup Engines and Jobstreams, Rule State Variables	Define rule state variables. This enables dynamic changes in the application engine.
Engine Metadata - Source TableMaps	PF_META_ENG_TBL4	EPM Foundation, Job Processing, Setup Engines and Jobstreams, Source TableMaps	Specify source tablemaps if the engine needs to resolve a set of tables before running.
Engine Metadata - Source Trees	PF_META_ENG_TBL5	EPM Foundation, Job Processing, Setup Engines and Jobstreams, Source Trees	Specify source trees to be flattened during the application engine initialization phase.

Engine Metadata Page

Use the Engine Metadata page (PF_META_ENG_TBL1) to review delivered engine metadata and modify if required. Add new metadata for a new application engine process. Unless you create your own application engine process, you do not need to create any engine metadata.

Navigation

EPM Foundation, Job Processing, Setup Engines and Jobstreams, Engine Metadata

Image: Engine Metadata page

This example illustrates the fields and controls on the Engine Metadata page. You can find definitions for the fields and controls later on this page.

Engine ID

Displays the unique identifier for the application engine.

Program Name

Select a delivered application engine program.

Engine Group

Select an engine group to tie the engine to a column on the Scenario definition page. When an engine is run, it picks up the model ID from the Scenario Definition page and uses it to process the rules and data.

State Record

Displays the name of the state record. The state record defines which record is used to define state variables on the next page in this component. It is a prompt for the state variable column on that grid.

Default Chunk Merge Method

Select a value to determine the method for the reloads of a chunked job merge to the main jobstream.

Values are:

Last in: Merges any duplicate data last, replacing the old data.

First in: Merges any duplicate data so that the original data remains.

Aggregate: Merges any duplicate data and aggregates it.

Append: Appends any duplicate data.

Process Wait Time (Seconds)

Displays the seconds of lag time before the next process runs. The default for this field is set on the Installation Options -

Web Services page. You can override the default setting here if required.

See [Web Services Page](#).

Balancing Rules

Enter any balancing rules that you want to run with this engine.

The program name appears.

Run Sequence

Displays the sequence number for the run. This number must be unique.

Section

This is not a required field. It is used to help you focus on the problem area when there is an out-of-balance situation. Before a section can be considered valid on the job totals page, it must be defined in the engine metadata. Valid section codes to be entered on the Engine Metadata page are the actual application engine section within the application engine program.

Engine Metadata - State Variables Page

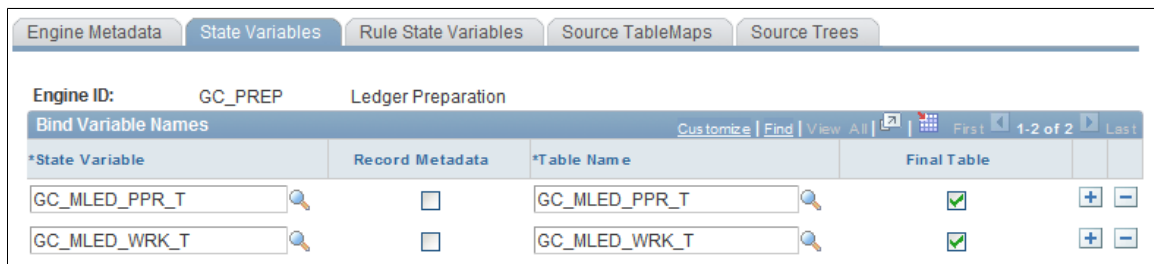
Use the Engine Metadata - State Variables page (PF_META_ENG_TBL2) to review state variables modify if required. These state variables enable dynamic changes to application engine inputs.

Navigation

EPM Foundation, Job Processing, Setup Engines and Jobstreams, State Variables

Image: State Variables page

This example illustrates the fields and controls on the State Variables page. You can find definitions for the fields and controls later on this page.



PeopleSoft delivers predefined state variables that enable dynamic changes to application engine inputs. State variables can include the input and output tables that an engine requires, so that the table name does not have to be hard-coded in the application engine program. You rarely need to change state variables. However, you can select records for chunking.

In the Bind Variables Names group box specify the following:

State Variable

Specify the record field name. The prompt list only shows those state record fields for the state record specified on the Engine Metadata page.

When the engine runs, the record stub is stored in this field on the state record. The page must be saved before the prompt works.

Record Metadata

Select this option if this is record metadata.

Table Name

Specify the table that is used in the application engine to populate the state variables with the record stub.

Final Table

Select this option if this is the final output table.

Chunk

Select this option to enable chunking for this record.

The Chunk By and Merge Method fields display.

Chunk By

Select the object type for the chunking for example by *Group ID*, *Job Code*, *Model ID*, and so on. This field defaults to *Group ID*. The object type you select here must match that selected on the Technical Scenarios page.

Merge Method

Select a value to determine the method for the reloads of a chunked job merge to the main jobstream.

Values are:

Last in: Merges any duplicate data last, replacing the old data.

First in: Merges any duplicate data so that the original data remains.

Aggregate: Merges any duplicate data and aggregates it.

Append: Appends any duplicate data.

You set the default on the Engine Metadata page.

Warning! If at least one chunk check box is selected, the Merge Method list box appears for the tables that are marked as *Final*. You may override the default merge method.

Record Stub and Table Status

The following table defines the relationship between record metadata and table status:

Record Metadata Selected?	Final Table Selected?	Chunk Selected?	Information
Yes	No	Yes	Typical input. Record stub is put into state record and table is resolved.

Record Metadata Selected?	Final Table Selected?	Chunk Selected?	Information
Yes	Yes	Yes	Typical output. Record stub is put into state record, temporary table is truncated, and the table is marked so that it is not resolved in the jobstream.
No	Yes	No	Record stub is put into state record and temporary table is truncated.
No	No	No	Record stub is put into state record.

Engine Metadata - Rule State Variables Page

Use the Engine Metadata - Rule State Variables page (PF_META_ENG_TBL3) to define rule state variables. This enables dynamic changes in the application engine.

Navigation

EPM Foundation, Job Processing, Setup Engines and Jobstreams, Rule State Variables

Image: Rule State Variables page

This example illustrates the fields and controls on the Rule State Variables page. You can find definitions for the fields and controls later on this page.

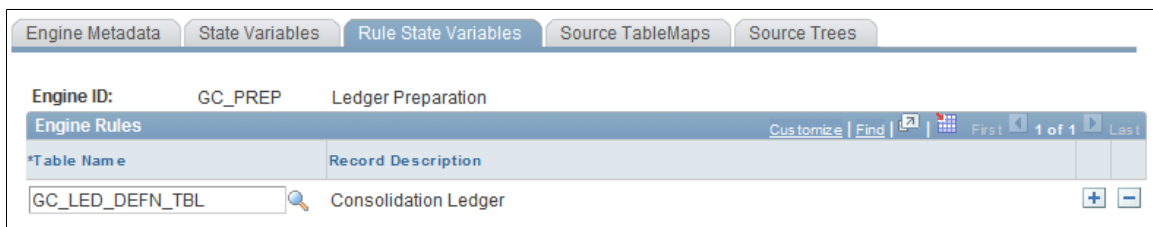


Table Name

Select the parent table of the rule to be resolved. You identify only the parent table because both parent and child tables are resolved. This rule table is resolved automatically as part of running the application engine.

The description for the table appears.

Engine Metadata - Source TableMaps Page

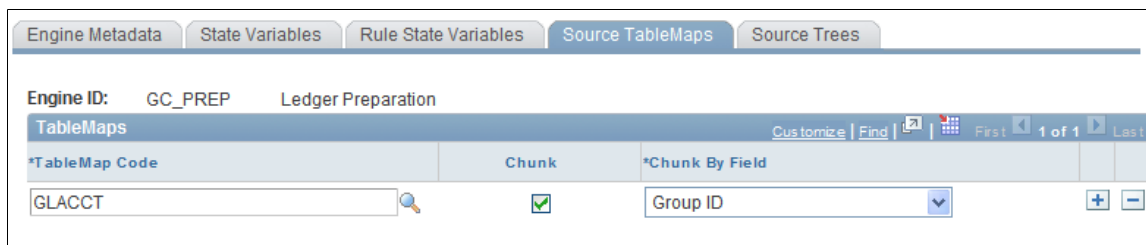
Use the Engine Metadata - Source TableMaps page (PF_META_ENG_TBL4) to specify source tablemaps if the engine needs to resolve a set of tables before running.

Navigation

EPM Foundation, Job Processing, Setup Engines and Jobstreams, Source TableMaps

Image: Engine Metadata - Source TableMaps page

This example illustrates the fields and controls on the Engine Metadata - Source TableMaps page. You can find definitions for the fields and controls later on this page.



TableMap Code

Displays only predefined tablemaps that are available for selection. The selected tablemap is automatically resolved as part of the application engine execution.

Chunk

If you select this check box, you enable the chunking of a tablemap within the Resolver. The Chunk By Field automatically displays a default value of *GROUP_ID*

Chunk By Field

Select an object type for chunking for example by *Group ID*, *Job Code*, *Model ID*, and so on. This field defaults to *Group ID*.

This object type must match the type selected on the Technical Scenarios page.

Engine Metadata - Source Trees Page

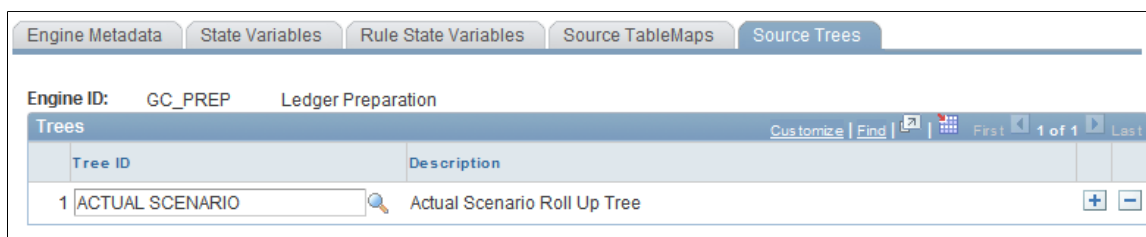
Use the Engine Metadata - Source Trees page (PF_META_ENG_TBL5) to specify source trees to be flattened during the application engine initialization phase.

Navigation

EPM Foundation, Job Processing, Setup Engines and Jobstreams, Source Trees

Image: Source Trees page

This example illustrates the fields and controls on the Source Trees page. You can find definitions for the fields and controls later on this page.



Tree ID

Select the trees to be resolved during the application engine initialization phase. If the run control parameters do not match

the tree parameters, the application engine logs a message that the tree is not resolved.

Setting Up Job Metadata

Job metadata (in other words, job ID) represents an instance, or copy, of a PeopleSoft application engine program and is used in the creation of a Jobstream. A job ID can be reused multiple times in the same jobstream or across multiple jobstreams. This enables you to use the same application engine more than once without having to define multiple job IDs. PeopleSoft deliver predefined job IDs for many of the jobs you need to run, but you can create additional job IDs if necessary.

After you complete the creation of a job ID, you can create a jobstream that runs just one engine or a combination of engines sequentially.

Page Used to Set Up Job Metadata

<i>Page Name</i>	<i>Definition Name</i>	<i>Navigation</i>	<i>Usage</i>
Job Metadata	PF_META_JOB_TBL1	EPM Foundation, Job Processing, Setup Engines and Jobstreams, Job Metadata	Create a job ID to define an instance an engine.

Job Metadata Page

Use the Job Metadata page (PF_META_JOB_TBL1) to create a job ID to define an instance an engine.

Navigation

EPM Foundation, Job Processing, Setup Engines and Jobstreams, Job Metadata

Image: Job Metadata page

This example illustrates the fields and controls on the Job Metadata page. You can find definitions for the fields and controls later on this page.

The screenshot shows the 'Job Metadata' page with the following fields and controls:

- Job ID:** ABM
- Job Metadata Information** (Section Header)
- *Description:** Run ABM
- *Engine ID:** ABM (with a search icon) Activity Based Management
- Limit Use to One Instance**

Job ID Displays the unique identifier for the instance of the engine.

Engine ID Select the correct engine ID for the instance you are defining.

Limit Use to One Instance Select this check box to make a job ID unique.

If this check box is selected, the job ID specified cannot be reused. You should deselect this check box for Job IDs that you create.

Setting Up Jobstreams

All jobs must be placed in a jobstream where you can add a single job or combine multiple jobs and pass data from one job to the next. All engines in a jobstream must run with the same run parameters (business unit, scenario ID, fiscal year, and so on). Additionally, jobstreams use the table appends defined on the Record Suites page to enable parallel processing of the same engines and tables by multiple users.

PeopleSoft EPM is delivered with a number of predefined jobstreams. Refer to your application-specific documentation for details on these jobstreams.

If a job in a jobstream cannot access a locked record suite during processing, the jobstream is sent to queue and waits for reprocessing once the record suite becomes available. You specify the queuing method on the Jobstream page.

Page Used to Define a Jobstream

<i>Page Name</i>	<i>Definition Name</i>	<i>Navigation</i>	<i>Usage</i>
Jobstream	PF_JOBSTRM_DFN1	EPM Foundation, Job Processing, Setup Engines and Jobstreams, Processes in Jobstream, Jobstream	Define a jobstream by specifying which engines (job IDs) to run and the order in which to run them.

Jobstreams Page

Use the Jobstream page (PF_JOBSTRM_DFN1) to define a jobstream by specifying which engines (job IDs) to run and the order in which to run them.

Navigation

EPM Foundation, Job Processing, Setup Engines and Jobstreams, Processes in Jobstream, Jobstream

Image: Jobstream page

This example illustrates the fields and controls on the Jobstream page. You can find definitions for the fields and controls later on this page.

Jobstream

Jobstream ID: ABM

Jobstream Definition Find | View All First 1 of 1 Last

*Effective Date: 01/01/1900 *Status: Active

*Description: Run ABM *Jobstream Type: Warehouse Jobstream

Retry Enabled:

Number of Attempts:

Unlimited Attempts

Maximum Number of Attempts:

Retry Interval (seconds):

Jobstream Information Customize | Find First 1-5 of 5 Last

Jobstream Sequence	Job ID	Description
100	AB_MDLGEN	ABM Model Generator
150	AB_LDMPGEN	ABM Ledger Maps Generator
200	ABM	Run ABM
250	AB_DRILL2	ABM Drillback
300	MERGEABM	Merge ABM

Jobstream ID

Displays the unique identifier for the series of jobs to be run.

Jobstream Type

Select the type of jobstream you are defining.

Retry Enabled?

Select this check box if you want a jobstream to queue if its related record suite is locked.

If this check box is selected the jobstream queues automatically when its record suite is locked and reruns when the record suite becomes available.

Number of Attempts

Unlimited Attempts

Select this option to have a queued jobstream attempt reprocessing indefinitely.

Maximum Number of Attempts

Select this option if you want a queued jobstream to attempt reprocessing a limited number of times.

In addition, specify the number of times you wish to have a queued jobstream attempt reprocessing. Once this option is selected, the Retry Interval (seconds) field becomes available for input.

Note: After the queued jobstream reaches the maximum number of reprocessing attempts specified in this field, the jobstream fails and a Warning status is set in the Process Monitor.

Retry Interval (seconds)

Specify the amount of time (in seconds) that should elapse between each reprocessing attempt.

Jobstream Information**Jobstream Sequence**

Enter a unique number, such as *100*, for the first job ID this jobstream should run. The next job ID to run is *200*.

Note: The actual sequence number is not important; it represents the sequence in which you want to run jobs. The sequence number must be unique.

Job ID

Select the job ID from the drop-down list box. The Job ID is created on the Job Metadata page and represents a unique instance of an engine for this jobstream.

After you create a jobstream, you can go back to the Job Metadata page and verify the Jobstream ID and Job Use fields. They are now populated.

Linking Jobstreams Sequentially

You can link multiple jobstreams sequentially using PeopleTools *JobSet* functionality. JobSets enable you to schedule any application engine process using a schedule JobSet definition. Because a jobstream is an application engine process, you can use the jobset to sequentially link multiple jobstreams together. Using the JobSet functionality you can:

- Use different run control IDs for each process within a jobset.
- Run processes from different operating systems or servers.
- Monitor processes from the process scheduler.

See *PeopleSoft PeopleTools: PeopleSoft Process Scheduler*

Working with Record Suites

PeopleSoft delivers three predefined record suites: 001, 002, and 003. Record suites are a group of temporary tables with the same temp table append for processing instances of an application engine. In order for your jobstream to run, you must associate record suites with the jobstream ID. You can also create new record suites and add them.

This topic discusses how to:

- Create new record suites.
- Set up record suites.

- Associate record suites with a jobstream.

Pages Used to Work with Record Suites

<i>Page Name</i>	<i>Definition Name</i>	<i>Navigation</i>	<i>Usage</i>
Record Suite	PF_RECSUITE_DFN1	EPM Foundation, Job Processing, Setup Record Suites, Define Record Suite, Record Suite	Set up the record suites that are delivered with PeopleSoft EPM. You might need to create more temporary tables.
Jobstream Record Suites	PF_REC_JOB_TBL1	EPM Foundation, Job Processing, Setup Record Suites, Jobstream Record Suites, Jobstream Record Suites	Associate record suites with a jobstream. You usually assign all three record suites to each jobstream ID.

Creating Record Suites

You can create record suites in addition to those delivered with PeopleSoft EPM.

To create new record suites:

1. Select PeopleTools, Utilities, Administration, PeopleTools Options.
2. Increase the number of temp table instances to the desired number.
3. Add new record suites on the Record Suite page described below.
4. Build the EPM_TEMP_RECORDS project.

Note: Be aware that this also builds 001, 002, and 003 again.

Record Suite Page

Use the Record Suite page (PF_RECSUITE_DFN1) to set up the record suites that are delivered with PeopleSoft EPM. You might need to create more temporary tables.

Navigation

EPM Foundation, Job Processing, Setup Record Suites, Define Record Suite, Record Suite

Image: Record Suite page

This example illustrates the fields and controls on the Record Suite page. You can find definitions for the fields and controls later on this page.

Record Suite

Record Suite ID: 001

*Description:

*Table Append:

Record Suite ID

Displays the unique identifier for the record suite.

Table Append

Enter the number to be used as the table append for the record suite. The number must be less than or equal to the number that is defined in the temp table instances field on the PeopleTools, Utilities, Administration PeopleTools Options page. You may define chunking record suites as well.

Jobstream Record Suites Page

Use the Jobstream Record Suites page (PF_REC_JOB_TBL1) to associate record suites with a jobstream. You usually assign all three record suites to each jobstream ID.

Navigation

EPM Foundation, Job Processing, Setup Record Suites, Jobstream Record Suites, Jobstream Record Suites

Image: Jobstream Record Suites page

This example illustrates the fields and controls on the Jobstream Record Suites page. You can find definitions for the fields and controls later on this page.

Jobstream Record Suites	
SetID:	SHARE
Jobstream ID:	ABMLEDG ABM Ledger
Find First 1-3 of 3 Last	
*Record Suite ID	
1	001 - Record Suite 1
2	002 - Record Suite 2
3	003 - Record Suite 3

SetID

Displays the SetID that you pointed to for your record group.

For example, if the run control is for business unit CORP1 and it points to SetID MODEL, you must create a jobstream and record suite combination for MODEL.

Jobstream ID

Displays the jobstream ID to which you are assigning the record suites.

Record Suite ID

Select the record suite to be used by the jobstream. You can add more than one record suite.

Creating Additional Instances of Temporary Tables for Record Suites

You can add instances of temporary tables to a record suite per your business requirements. To create additional instances of the temporary tables:

1. In Application Designer copy all tables in the relevant project and change the last two characters of the table name.
2. Rebuild the project.

Your new temporary table suite is ready. Repeat this process to create additional temporary table suites.

Project Name	Description
EPM_SECURITY	All delivered views that give secure access to EPM objects.
EPM_TEMP_RECORDS	One instance of the temporary tables needed to run all the EPM engines.
EPM_TEMP_RECORDS_INC	Incremental project that contains one instance of newly added temporary tables only.
EPM_TOOLS	All EPM-specific changes to the PPLTOOLS project.

Note: If you are altering the number of temporary table instances, change the default setting of 3 to the desired number on the PeopleTools Option page (located at PeopleTools, Utilities, PeopleTools Option) and rebuild EPM_TEMP_RECORDS project in PeopleSoft Application Designer.

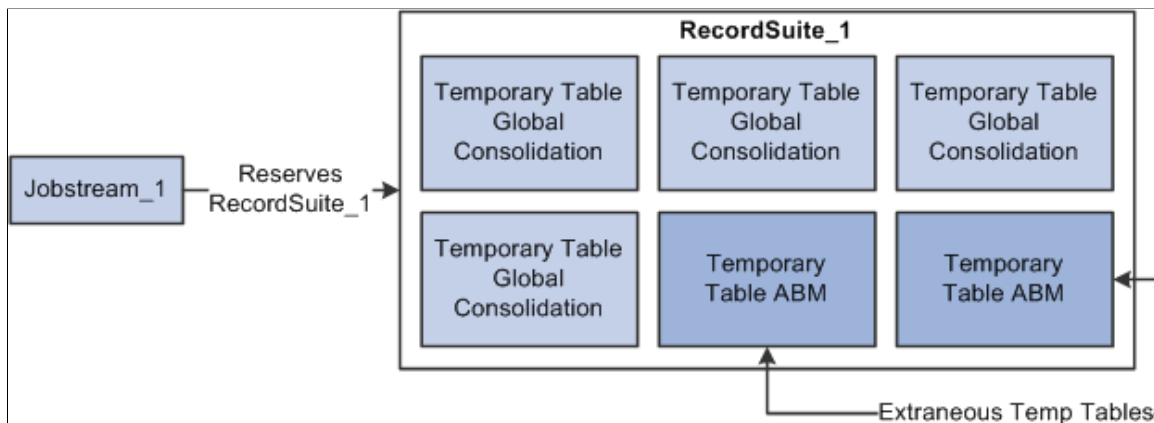
See *PeopleSoft PeopleTools: PeopleSoft Application Designer Developer's Guide*

Removing Extraneous Temporary Tables from Record Suites

Jobstreams use record suites and their corresponding temporary tables to process data. PeopleSoft delivers record suites with an entire set of EPM temporary tables, regardless of the products you license. For example, if you only purchase the Global Consolidations analytical application, the delivered record suites still contain temporary tables for all other EPM products—such as ABM and Budgeting. The following diagram depicts this scenario:

Image: Record suite with extraneous temporary tables

This diagram depicts this scenario:



Each delivered record suite can potentially contain hundreds or thousands of extraneous temporary tables—due to the fact that the total number of EPM temporary tables exceed 3,300. Each time a record suite is processed with these superfluous temporary tables, processing efficiency is severely degraded. However, PeopleSoft provide functionality that enables you to remove unnecessary temporary tables from the EPM database

Pages Used to Remove Temporary Tables from EPM

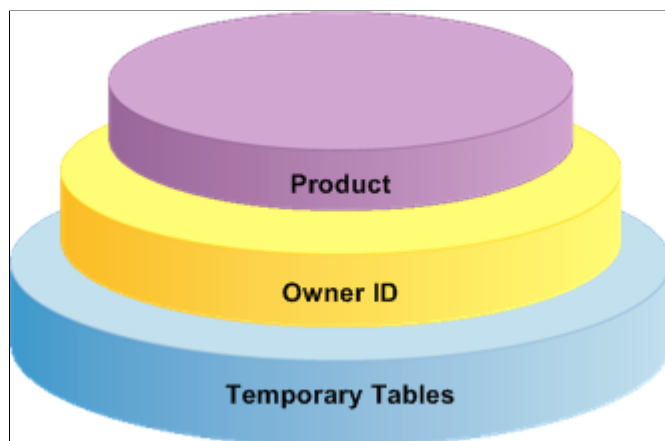
<i>Page Name</i>	<i>Definition Name</i>	<i>Navigation</i>	<i>Usage</i>
Clean Up Temporary Tables	PF_RUN_CLEAN_TMP	EPM Foundation, Job Processing, Temporary Tables, Cleanup Temp Tables, Clean Up Temporary Tables	Run the Clean Temporary Tables application engine.
Cleanup Log	PF_TMPTBL_VW	EPM Foundation, Job Processing, Temporary Tables, Cleanup Log	View temporary tables dropped from the EPM database.

Understanding the Temporary Table Removal Process

Extra temporary tables are removed from the EPM database by running the Clean Temporary Tables application engine (PF_CLEAN_TMP). To identify the temporary tables that need to be removed from EPM, all temporary tables are assigned an ownerID. Each owner ID is assigned to a specific EPM product. While temporary tables can only be assigned to one owner ID, you can assign several owner IDs to a single product.

Image: Temporary table to product hierarchy

This diagram depicts the hierarchical relationship between temporary tables and the products.



The Clean Temporary Tables process uses the product, ownerID, and customer licensing information (plus the relationships between these objects), to identify the tables that should be removed from the EPM database. The information is stored in the following EPM tables:

- PS_PF_PROD_TO_OWNR: Contains product code, ownerID, and record type, and identifies the relationship between EPM product and ownerID.
- PSRECDEFN: Identifies the relationship between EPM temporary tables and ownerID.

- **PSINSTALLATION:** Identifies which EPM products you have licensed.

At runtime the Clean Temporary Tables application engine uses the information stored in the aforementioned tables to identify and delete all instances of temporary tables that are not required by your licensed product(s). It then regenerates the EPM_TEMP_RECORDS and EPM_TEMP_RECORDS_INC application designer projects with the new temporary tables.

All temporary tables being shared among applications belong to a special ownerID named *AppCommon* and are not deleted. There are also fundamental temporary tables associated with the ownerID *EPM Foundation* and are not deleted.

After running the Clean Temporary Tables process, all temporary tables that have been dropped from the database are logged in the PF_TMPTBL_LOG table. You can view the results of this process using the Cleanup Log page.

The Clean Temporary Tables application engine should be run after every install, upgrade or patch of PeopleSoft EPM products. After it runs, open the EPM_TEMP_RECORDS_INC project in Application Designer, and if not empty, rebuild the project to ensure that newly added temporary tables are built.

Delivered EPM_TEMP_RECORDS_INC Project

The EPM_TEMP_RECORDS_INC project is provided to help you integrate new temporary tables into EPM when you have added (licensed) a new EPM product but already run the Clean Temporary Tables process. Keep in mind that when you run the Clean Temporary Tables process, you delete temporary tables associated with any uninstalled EPM product. Therefore, when you introduce a new EPM product you also add new temporary tables back into the related projects. As described in this topic, when you add new temporary tables to a project, you have to rebuild the project. The EPM_TEMP_RECORDS_INC project is an incremental project which contains only newly added temporary tables. You can use the EPM_TEMP_RECORDS_INC project to rebuild your temporary tables, instead of using the EPM_TEMP_RECORDS project with the entire set of your temporary tables. Because of its smaller size, rebuilding the EPM_TEMP_RECORDS_INC project saves you processing time.

See [Creating Additional Instances of Temporary Tables for Record Suites](#).

Clean Up Temporary Tables Page

Use the Clean Up Temporary Tables page (PF_RUN_CLEAN_TMP) to run the Clean Temporary Tables application engine.

Navigation

EPM Foundation, Job Processing, Temporary Tables, Cleanup Temp Tables, Clean Up Temporary Tables

Image: Clean Up Temporary Tables page

This example illustrates the fields and controls on the Clean Up Temporary Tables page. You can find definitions for the fields and controls later on this page.

When Select the frequency in which you would like the PF_CLEAN_TMP process to run.

Run Click to run the PF_CLEAN_TMP process.

Cleanup Log Page

Use the Cleanup Log page (PF_TMPTBL_VW) to view temporary tables dropped from the EPM database.

Navigation

EPM Foundation, Job Processing, Temporary Tables, Cleanup Log

Product Name Displays the product associated with the deleted temporary tables.

DateTime Displays the date and time the process was run.

Number of temporary tables Displays the total number of temporary tables associated with the selected product.

Installed Product Indicates whether this is an installed product.

	Only temporary tables from non-installed products should be dropped.
Tables Dropped	Displays the total number of temporary tables dropped from the EPM database.
Table Name	Displays the name of the temporary table deleted from the EPM database.
Object Owner ID	Displays the Owner ID associated with a particular temporary table.

Running Jobstreams

This topic discusses how to:

- Run jobstreams.
- Run multiple jobstreams.
- Set up email notification.

Pages Used to Run a Jobstream and Multiple Jobstreams

<i>Page Name</i>	<i>Definition Name</i>	<i>Navigation</i>	<i>Usage</i>
Run Jobstream	RUN_PF_JOBSTREAM	EPM Foundation, Job Processing, Update/Run Jobstreams, Run Jobstream	Run a jobstream.
Run Multiple Jobstream	RUN_PF_MULTIPERIOD	EPM Foundation, Job Processing, Update/Run Jobstreams, Run Multiple Jobstream	Run a jobstream for multiple fiscal years and accounting periods.
Jobstream Email Notification	PF_EMAIL_MSG	Click Specify Email Parameters on the Run Jobstream or Run Multiple Jobstreams page.	Set up email parameters for automatically notifying users when a jobstream is complete or abended.

Run Jobstream Page

Use the Run Jobstream page (RUN_PF_JOBSTREAM) to run a jobstream.

Navigation

EPM Foundation, Job Processing, Update/Run Jobstreams, Run Jobstream

Image: Run Jobstream page

This example illustrates the fields and controls on the Run Jobstream page. You can find definitions for the fields and controls later on this page.

The screenshot shows the 'Run Jobstream' page with the following details:

- User ID:** VP1. Links: [Report Manager](#), [Process Monitor](#), [View Messages](#).
- Run Control ID:** BC_ANALYZE_MFG_2003_1. Links: [Clear All Suites](#), [Clear Last Suite](#)
- Process Information** (Section Header)
- Program Name:** PF_JOBSTREAM. **When:** Always (dropdown). **Send Email Notification**. [Specify Email Parameters](#)
- As Of Dated Jobstream**
- *Description:** MFG Analyze process for Perio
- *Unit:** CORP1 Corporation 1
- *Scenario ID:** MFG_SC Manufacturing Scorecard
- Fiscal Year:** 2003
- Period:** 1
- *Jobstream ID:** BC_ANALYZE Calc KPIs and Assess Scorecard
- Rerun**
- Last Run On:** 03/26/2010 3:23:33PM PDT. **As Of Date:** 01/31/2003

As Of Dated Jobstream

Select this check box to disable the Fiscal Year and Period fields. Enter an as of date for the jobstream run.

Send Email Notification

Select this check box to send an email notification to all of the email addresses that you define by clicking the Specify Email Parameters link. The email notification informs the recipients that the jobstream is complete or has abended.

Description

Enter a description for the jobstream run. The Metadata Search engine uses this description to find the data later.

Unit and Scenario ID

Select the business unit and scenario ID combination.

Fiscal Year and Period

Enter the fiscal year and period for this jobstream run. This field does not appear if you select the As Of Dated Jobstream check box.

Jobstream ID

Select the jobstream you want to run.

Rerun

Select this check box if you are processing the same job an additional time using identical parameters and want the system to re-resolve the tables.

Re-resolving means that data is re-selected from the permanent table and moved to temporary tables of the assigned record suite.

Note: This option may slow down processing if you are assigned to the same record suite assigned the previous time that the engine was run.

Last Run On

Displays the date and time this jobstream was last run.

As Of Date

Displays the as of date for an as-of-dated jobstream. If you are using the Fiscal Year and Period, this field displays the last day of the fiscal year and period combination based on the calendars you defined.

View Messages

Once a jobstream has run, click to view the engine messages generated by the jobstream. This page is described later in this topic.

Clear Last Suite

Select this option to release the last record suite used by this jobstream.

Clear All Suites

Select this option to release all record suites. All record suites are now available to jobstreams.

Warning! Before clearing all record suites, make sure that no jobs are running.

Run Multiple Jobstream Page

Use the Run Multiple Jobstream page (RUN_PF_MULTIPERIOD) to run a jobstream for multiple fiscal years and accounting periods.

Navigation

EPM Foundation, Job Processing, Update/Run Jobstreams, Run Multiple Jobstream

Image: Run Multiple Jobstream page

This example illustrates the fields and controls on the Run Multiple Jobstream page. You can find definitions for the fields and controls later on this page.

The screenshot shows the 'Run Multiple Jobstream' page with the following fields and controls:

- User ID: VP1
- Run Control ID: RUNMULTI
- Program Name: PF_MULTI_PER
- When: Always (dropdown menu)
- Send Email Notification: (checkbox)
- *Description: [Empty text box]
- *Business Unit: CORP1 (with search icon)
- *Scenario ID: ACTUAL (with search icon)
- From Year: 2007
- From Period: 1
- To Year: 2009
- To Period: 4
- *Jobstream ID: ABM (with search icon)
- Rerun: (checkbox)
- Last Run On: [Empty text box]

Program Name

Displays the name of the jobstream program.

Send Email Notification

Select this check box to send an email notification to all of the email addresses that you define by clicking the Specify Email

	Parameters link. The email notification informs the recipients that the jobstream is complete or has abended.
Description	Enter a description for the jobstream run. The Metadata Search engine uses this description to find the data later.
Business Unit and Scenario ID	Select the business unit and scenario ID combination.
From Year and From Period , To Year and To Period	Enter the fiscal years and periods to include in this jobstream. Unlike the Run Jobstream page, on which you can only specify one fiscal year and period combination, you can specify a range of years and periods.
Jobstream ID	Select the jobstream you want to run.
Rerun	Select this check box if you are processing the same job an additional time using identical parameters and want the system to re-resolve the tables. Re-resolving means that data is re-selected from the permanent table and moved to temporary tables of the assigned record suite. <hr/> Note: This option may slow down processing if you are assigned to the same record suite assigned the previous time that the engine was run. <hr/>
Last Run On	Displays the date and time this jobstream was last run.
Run	Click this button to access the Process Scheduler Request page on which you define the parameters for running the jobstream.
Process Monitor	Click this button to access Process Monitor pages to check process scheduler results. Process Monitor provides updated information on the progress of reports and processes. From a Web browser, you can monitor process requests and the status of different servers that run your reports. If there are messages related to a process, you can view them from Process Monitor, as well.

Jobstream Email Notification Page

Use the Jobstream Email Notification page (PF_EMAIL_MSG) to set up email parameters for automatically notifying users when a jobstream is complete or abended.

Navigation

Click Specify Email Parameters on the Run Jobstream or Run Multiple Jobstreams page.

Use this page to list the email addresses of those recipients who should receive a notification when the jobstream completes or abends. You can enter a subject for the email and any text you would like to send.

Tracking Jobs

There are a number of pages enabling you to track the progress of your jobs. This topic discusses how to:

- Review record suites.
- Review record suite history.
- Review records in a jobstream.
- Review jobstream history.
- Review temporary tables.
- Review temporary table history.

Pages Used to Track Jobs

<i>Page Name</i>	<i>Definition Name</i>	<i>Navigation</i>	<i>Usage</i>
Record Suites	PF_RECSUITE_TBL1	EPM Foundation, Job Processing, Review Jobstream Content, Review Record Suites, Record Suites	View all of the defined record suites.
Jobstream Job Detail	PF_JOBSTRM_TBL2S	Click the button on the Record Suites page.	View runtime parameters to determine whether a record suite is in use.
Record Suite History	PF_RECSUITE_HIS1	EPM Foundation, Job Processing, Review Jobstream Content, Record Suite History	View the process instances, job description, and run control IDs that were run in the specified record suite.
Jobstream	PF_JOBSTRM_TBL1	EPM Foundation, Job Processing, Review Jobstream Content, Review Jobstream	View the current status of a jobstream for all defined record suites.
Jobstream History	PF_JOBSTRM_HIS1	EPM Foundation, Job Processing, Review Jobstream Content, Jobstream History	View the job ID, record suites, and run control parameters that have been run for a selected jobstream.
Temporary Table	PF_TEMP_REC_TBL1	EPM Foundation, Job Processing, Temporary Tables, Temporary Table	View, for each record suite, the temporary tables that have been populated. This page also displays the run controls that were used to populate them.
Temporary Table History	PF_TEMP_REC_HIS1	EPM Foundation, Job Processing, Temporary Tables, Temp Table History	Review table usage for a record suite.

Record Suites Page




Use the Record Suites page (PF_RECSUITE_TBL1) to view all of the defined record suites.

Navigation

EPM Foundation, Job Processing, Review Jobstream Content, Review Record Suites, Record Suites

Image: Record Suites page

This example illustrates the fields and controls on the Record Suites page. You can find definitions for the fields and controls later on this page.

Record Suites						
Record Suite ID	Date/Time Stamp	in use sw	Run Control ID	Instance	Chunk Lock Flag	
001	03/30/2010 10:03:13PM	<input type="checkbox"/>	 FTP_RATE01	421	<input type="checkbox"/>	
002	01/01/1990 12:00:00AM	<input checked="" type="checkbox"/>			<input type="checkbox"/>	
003	01/01/1990 12:00:00AM	<input type="checkbox"/>			<input checked="" type="checkbox"/>	

Record Suite ID

Lists all the record suites.

Date/Time Stamp

Displays the date and time for the last or current use of this record suite.

in use sw

If this check box is selected, it indicates that a record suite is being used. Deselect this check box and its related page to make the corresponding record suite available to waiting jobstreams.

Note: PeopleSoft recommends that only experienced users or the Warehouse Administrator make such a change.



Click the View Jobstream Job Details button to access the Jobstream Job Detail secondary page on which you can see all the jobstreams to which the record suite is assigned.

Run Control ID

Displays the identifier of the run control set up to run the jobstream.

Instance

Displays the process instance of the last run or current run of the run control ID.

Chunk Lock Flag

If this option is selected, it indicates the record suite is in use for chunking. This check box works very much like the in use sw check box except that it is used for jobs with chunking.

Record Suite History Page

Use the Record Suite History page (PF_RECSUITE_HIS1) to view the process instances, job description, and run control IDs that were run in the specified record suite.

Navigation

EPM Foundation, Job Processing, Review Jobstream Content, Record Suite History

Image: Record Suite History page

This example illustrates the fields and controls on the Record Suite History page.

Record Suite History				
Record Suite ID:		001	Record Suite 1	
Record Suite Details				
Date/Time Stamp	in use sw	Run Control ID	Description	Instance
03/30/2010 10:03:04PM	<input checked="" type="checkbox"/>	FTP_RATE01	Merge Process for FTP_RATE	421
03/30/2010 9:57:05PM	<input checked="" type="checkbox"/>	FTP_RATE01	FTP Rate Setting Process	421
03/26/2010 4:07:07PM	<input checked="" type="checkbox"/>	BC_ANALYZE_MFG_2003_11	Merge for BC Analyze stream	420
03/26/2010 4:06:59PM	<input checked="" type="checkbox"/>	BC_ANALYZE_MFG_2003_11	Publish Alert Notifications	420
03/26/2010 4:06:16PM	<input checked="" type="checkbox"/>	BC_ANALYZE_MFG_2003_11	Assess Balanced Scorecard	420
03/26/2010 4:03:29PM	<input checked="" type="checkbox"/>	BC_ANALYZE_MFG_2003_11	Calc and Assess KPIs	420
03/26/2010 4:02:07PM	<input checked="" type="checkbox"/>	BC_ANALYZE_MFG_2003_10	Merge for BC Analyze stream	419
03/26/2010 4:01:58PM	<input checked="" type="checkbox"/>	BC_ANALYZE_MFG_2003_10	Publish Alert Notifications	419
03/26/2010 4:01:15PM	<input checked="" type="checkbox"/>	BC_ANALYZE_MFG_2003_10	Assess Balanced Scorecard	419
03/26/2010 3:58:27PM	<input checked="" type="checkbox"/>	BC_ANALYZE_MFG_2003_10	Calc and Assess KPIs	419
03/26/2010 3:57:14PM	<input checked="" type="checkbox"/>	BC_ANALYZE_MFG_2003_9	Merge for BC Analyze stream	418
03/26/2010 3:57:06PM	<input checked="" type="checkbox"/>	BC_ANALYZE_MFG_2003_9	Publish Alert Notifications	418
03/26/2010 3:56:24PM	<input checked="" type="checkbox"/>	BC_ANALYZE_MFG_2003_9	Assess Balanced Scorecard	418
03/26/2010 3:53:40PM	<input checked="" type="checkbox"/>	BC_ANALYZE_MFG_2003_9	Calc and Assess KPIs	418
03/26/2010 3:52:17PM	<input checked="" type="checkbox"/>	BC_ANALYZE_MFG_2003_8	Merge for BC Analyze stream	417

Use this page to review the process instances, job description, and run control IDs that were run in the selected record suite. A start and end time also display as well as an in use sw flag.

Jobstream Page

Use the Jobstream page (PF_JOBSTRM_TBL1) to view the current status of a jobstream for all defined record suites.

Navigation

EPM Foundation, Job Processing, Review Jobstream Content, Review Jobstream

Image: Jobstream page

This example illustrates the fields and controls on the Jobstream page.

Jobstream

Jobstream ID: ABM Run ABM

Jobstream Information Customize | Find | View All | First 1-3 of 3 Last

Run Details Job Details

Record Suite ID	Status	in use sw	Date/Time Stamp	Run Control ID	Job ID	Instance
001		<input type="checkbox"/>	01/01/1900 12:00:00AM			
002		<input type="checkbox"/>	01/01/1900 12:00:00AM			
003		<input type="checkbox"/>	01/01/1900 12:00:00AM			

Use this page to view the current status of a jobstream for all the defined record suites.

Jobstream History Page

Use the Jobstream History page (PF_JOBSTRM_HIS1) to view the job ID, record suites, and run control parameters that have been run for a selected jobstream.

Navigation

EPM Foundation, Job Processing, Review Jobstream Content, Jobstream History

Image: Jobstream History page

This example illustrates the fields and controls on the Jobstream History page.

Jobstream History

Jobstream ID: ABMLEDG ABM Ledger

Jobstream Information Customize | Find | View All | First 1 of 1 Last

Run Details Job Details

Record Suite ID	Status	in use sw	Date/Time Stamp	Run Control ID	Job ID	Instance
		<input type="checkbox"/>				

Use this page to view the job ID, record suites, and run control parameters that have been run for a jobstream.

Temporary Table Page

Use the Temporary Table page (PF_TEMP_REC_TBL1) to view, for each record suite, the temporary tables that have been populated. This page also displays the run controls that were used to populate them.

Navigation

EPM Foundation, Job Processing, Temporary Tables, Temporary Table

Image: Temporary Table page

This example illustrates the fields and controls on the Temporary Table page.

Temporary Table

Record Suite ID: 002 Record Suite 2

Table Information					
Table Name	Date/Time Stamp	Business Unit	Scenario ID	Fiscal Year	Period
ABC_INTERFACE	01/01/1990 12:00:00AM				
ABC_INTF_LM	09/30/2009 9:57:27AM				
ABC_INTF_MG	10/01/2009 2:58:57PM				
ABM_LEDGER_VW	01/01/1990 12:00:00AM				
ABM_LEDMAP_F00	01/01/1990 12:00:00AM				
ABM_LEDMAP_INT	01/01/1990 12:00:00AM				
ABM_LEDTMPL_TBL	01/01/1990 12:00:00AM				
ABPS_LEDGER_F00	01/01/1990 12:00:00AM				
ABSV_ACCR_F00	01/01/1990 12:00:00AM				
ABSV_PLAN_D00	01/01/1990 12:00:00AM				
ABS_CLASS_D00	01/01/1990 12:00:00AM				
ABS_CODE_D00	01/01/1990 12:00:00AM				
ABS_HIST_F00	01/01/1990 12:00:00AM				
ABS_TYPE_D00	01/01/1990 12:00:00AM				
AB_ACTASGN_F00	01/01/1990 12:00:00AM				

Use this page to view for a given record suite the temporary tables that have been populated. This page also shows the run control parameters.

Temporary Table History Page

Use the Temporary Table History page (PF_TEMP_REC_HIS1) to review table usage for a record suite.

Navigation

EPM Foundation, Job Processing, Temporary Tables, Temp Table History

Image: Temporary Table History page

This example illustrates the fields and controls on the Temporary Table History page.

Temporary Table History					
Record Suite ID: 002		Record Suite 2			
Table Information					
Table Name	Date/Time Stamp	Business Unit	Scenario ID	Fiscal Year	Period

Use this page to review table usage for a record suite.

Viewing Engine Messages

After you run a job or jobstream, view the process information and run control parameters for the engine that you just ran using the Messages component. You can access this component directly from the Run Jobstream page by clicking the View Engine Messages link.

This topic discusses how to view engine messages

Pages Used to View Engine Messages

<i>Page Name</i>	<i>Definition Name</i>	<i>Navigation</i>	<i>Usage</i>
Engine Messages - Message Header	PF_ENGMSG_HEAD	EPM Foundation, Job Processing, Review Jobstream Content, Engine Messages, Message Header	View display-only process information such as record suite ID and engine ID, as well as the run control parameters for this process instance.
Engine Messages - Message Detail	PF_ENGMSG_LOG	EPM Foundation, Job Processing, Review Jobstream Content, Engine Messages, Message Detail	View display-only process information such as source name, field name, and field value.
Message Detail	PF_ENGMSG_MSG	Click the on the Message Detail page.	View the detailed error message.

Engine Messages - Message Header Page

Use the Engine Messages - Message Header page (PF_ENGMSG_HEAD) to view display-only process information such as record suite ID and engine ID, as well as the run control parameters for this process instance.

Navigation

EPM Foundation, Job Processing, Review Jobstream Content, Engine Messages, Message Header

Image: Message Header page

This example illustrates the fields and controls on the Message Header page. You can find definitions for the fields and controls later on this page.

Message Header		Message Detail	
Process Information			
Process Instance:	381	Record Suite ID:	001
Engine ID:	ALLOCATION Allocation Engine	Table Append:	1
Jobstream ID:	GC_ALLO	Start Date/Time:	03/02/2010 12:46PM
Run Control ID:	ALLOCATION	End Date/Time:	03/02/2010 12:47:02PM
Run Control Parameters			
Business Unit:	10000		
Scenario ID:	GCACTTB		
Fiscal Year:	2005		
Accounting Period:	1		
As Of Date:	01/31/2005		

Process Information

View details such as the process instance, record suite, engine ID, jobstream ID, run control, table append, as well as the start and end date and time.

Run Control Parameters

Displays the defined run control parameters for the instance (business unit, scenario ID, fiscal year and accounting period, and as of date if applicable).

Go to the next page in the component to view engine message details.

Engine Messages - Message Detail Page

Use the Engine Messages - Message Detail page (PF_ENGMSG_MSG) to view the detailed error message.

Navigation

EPM Foundation, Job Processing, Review Jobstream Content, Engine Messages, Message Detail

For a given process instance, engine ID, jobstream ID and run control you can view the message details.

Search

Click this button to retrieve engine message details. The message details appear in the grid at the bottom of the page.

Engine Message Details

Displays the source of the error. In the message box, you can view the message set to which the message belongs, the message number, and a brief description of the error.



Click the Explain First button to access the Message Detail secondary page on which you can view a detailed error message for the error.

Setting Up and Using Profit Manager

Understanding Profit Manager Tools

The profit manager is a set of integrated tools that enable true multidimensional profitability reporting. To obtain true and meaningful profitability reports, you need a central repository as well as reliable and consistent data, and you need to consolidate and enrich the data from your general ledger and other sources.

PeopleSoft EPM infrastructure is the underlying framework that provides reliable and consistent data and consolidates data sources such as your general ledger and the EPM Analytical Applications such as PeopleSoft Activity-Based Management, Scorecard, Funds Transfer Pricing, Risk-Weighted Capital, Global Consolidations, and Workforce Analytics, which are application engines that enrich and transform data.

The features that are described in this topic are tightly integrated with the PeopleSoft analytical applications and provide you with ways to:

- Move data from your PeopleSoft general ledger using the Ledger Mapper.
- Verify the accuracy of your data before you post it to the performance ledger table using the PF Edit engine.
- Track data movement to and from the performance ledger table (PF_LEDGER_F00) using ledger drill down.
- Control batch processing using the PF Post and PF Unpost engines.
- Keep the contents of the journal table (PF_JRNL_F00) clean using the PF Journal Cleanup engine.
- Reconcile final table amounts using the balancing and reconciliation features.

After you set up and run a source engine or map ledger balances using the Ledger Mapper, the enriched data is moved to the performance journal table (PF_JRNL_F00). You can run the PF Journal Edit engine to check data integrity at any time. Use PF Journal Modification to revise errors. The PF Ledger Post process moves your data from the performance journal table to the performance ledger table (PF_LEDGER_F00) for reporting.

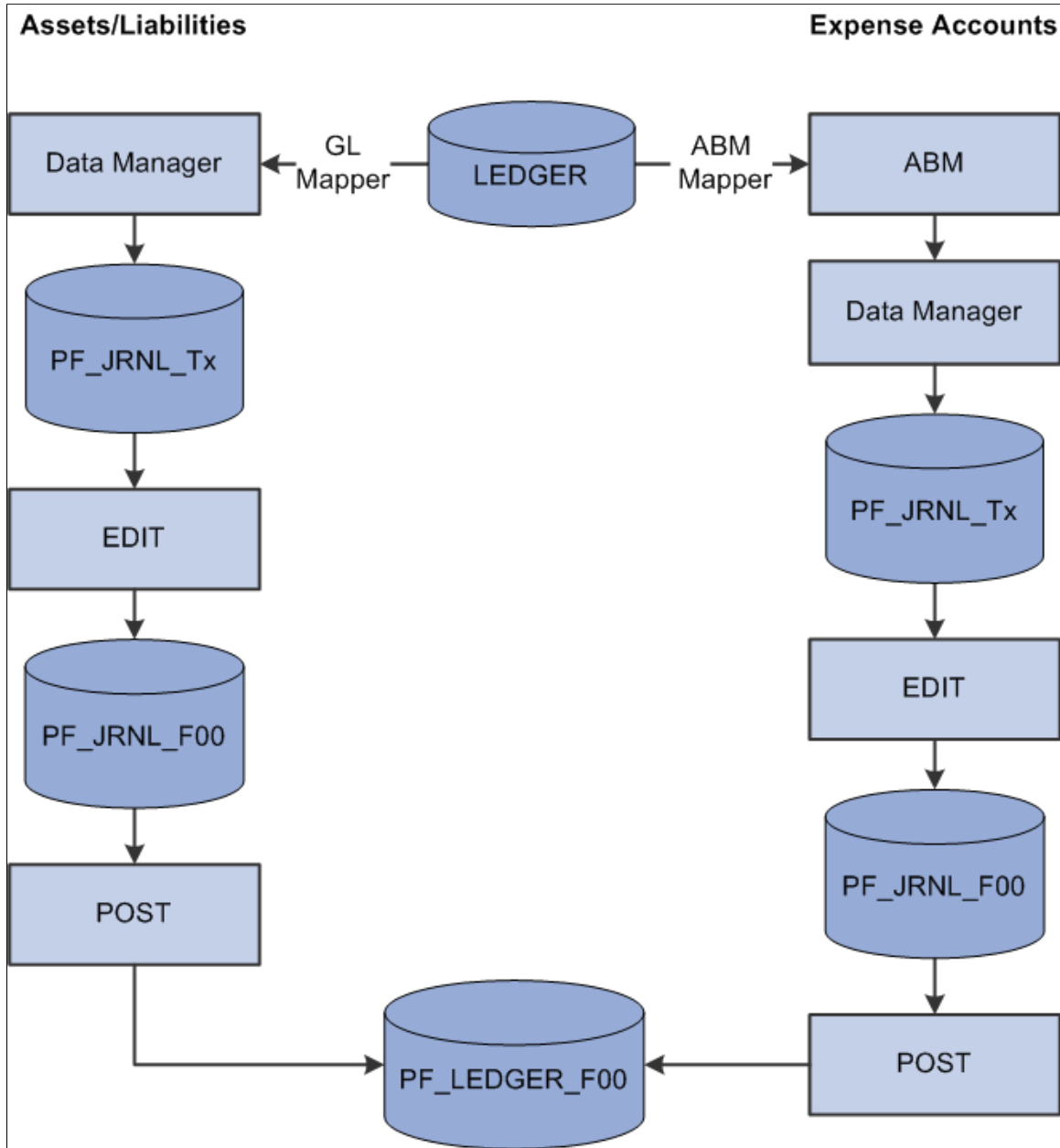
To check that data migration and enrichment is accurate:

1. Use the PF ledger drill down feature to track the source of general ledger data for a particular performance ledger after you populate the performance ledger table.

- Use the Reconciliation utility to check balances between tables such as GL_LEDGER and PF_LEDGER.

Image: Profit Manager loading the performance ledger table

This diagram illustrates how the Profit Manager loads the performance ledger table.



Ledger Mapper

Use the Ledger Mapper to map data, such as assets and liabilities, that does not enter the system through one of the optional analytical applications. After you define the ledger mapping rules, you must set up Data Manager rules using the GL mapper method and then run the Data Manager or Allocation Manager engine to populate the performance journal table (PF_JRNL_F00).

For example, PeopleSoft uses the Ledger Mapper to map expense data from the general ledger to Activity-Based Management resources.

To map ledger amounts, you:

1. Load the GL_LEDGER table.
2. Use the general ledger mapper method in Data Manager to move the general ledger data into the GLSTG temporary table that you identify as the source table for further data movement.

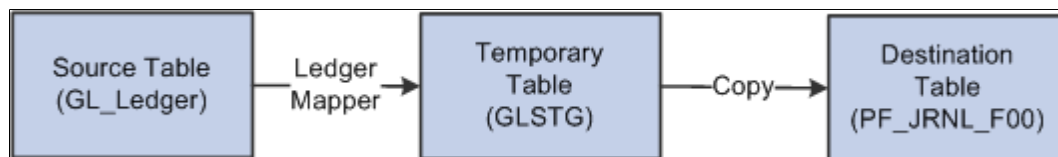
Note: You might decide to move data directly into the performance journal table (PF_JRNL_F00) using the copy method. You can also create multidimensional data using the prorata or spread even methods.

Note: A number of steps in this process use the Data Manager or the Allocation Manager engine. You must define tablemaps, datamaps, constraints, and filters, and then set up the Data Manager or Allocation Manager rules that use this metadata to produce multidimensional results. In addition, you must set up job metadata and jobstreams.

The following two diagrams illustrate how Ledger Mapper moves the data.

Image: Using the Ledger Mapper and copy methods to load the performance journal table (PF_JRNL_F00)

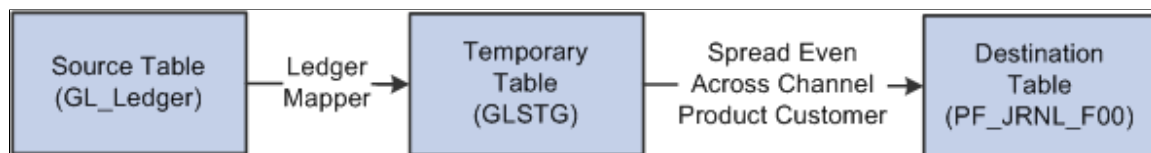
This diagram illustrates Ledger Mapper and the copy method.



The second diagram illustrates a method in which multidimensional data is created:

Image: Using the Ledger Mapper to move ledger data, create multidimensional data, and load the performance journal table (PF_JRNL_F00)

This example illustrates Using the Ledger Mapper to move ledger data, create multidimensional data, and load the performance journal table (PF_JRNL_F00).



The Ledger Mapper uses the business unit relationships that you established in EPM to map amounts. To map multiple general ledger accounts to one performance account, set up ledger mapping rules on the Ledger Mapper page.

Note: You must define your general ledger and warehouse business units and the relationship between them before you set up and run the Ledger Mapper. In addition, before you map ledgers, you must define the tablemaps, datamaps, and constraints to use when you run the Data Manager or Allocation Manager engines. EPM is delivered with predefined tablemaps, datamaps, constraints, and Ledger Mapper Data Manager rules for the SHARE SetID.

See [Understanding Warehouse Business Unit Setup](#), [Understanding Warehouse Business Units](#), [TableSet Sharing](#), and [SetID Mapping](#), and [Understanding Metadata](#).

Setting Up Ledger Mapper

This topic lists prerequisites and discusses how to:

- Map ledger data.
- Create data manager rules for the general ledger (GL) mapper method.
- Configure ledger mappings for new or configured ChartFields.

Page Used to Set Up the Ledger Mapper

<i>Page Name</i>	<i>Definition Name</i>	<i>Navigation</i>	<i>Usage</i>
Ledger to Resource Mapping	LEDMAP_TBL1	EPM Foundation, Data Enrichment Tools, Performance Ledger, Map Ledger Data, Ledger to Resource Mapping	Map ledger data, such as assets and liabilities, that do not enter the system through one of the analytic applications.

Prerequisites

Before you can set up your Ledger Mapper, you must:

- Complete your warehouse business unit setup.
- Specify your ledger mapping defaults.
- Set up your tablemaps, datamaps, filters, and constraints.

See [Understanding Warehouse Business Unit Setup](#), [Understanding Warehouse Business Units](#), [TableSet Sharing](#), and [SetID Mapping](#), [Specifying Ledger Mapping Defaults](#), and [Understanding Metadata](#).

Ledger to Resource Mapping Page

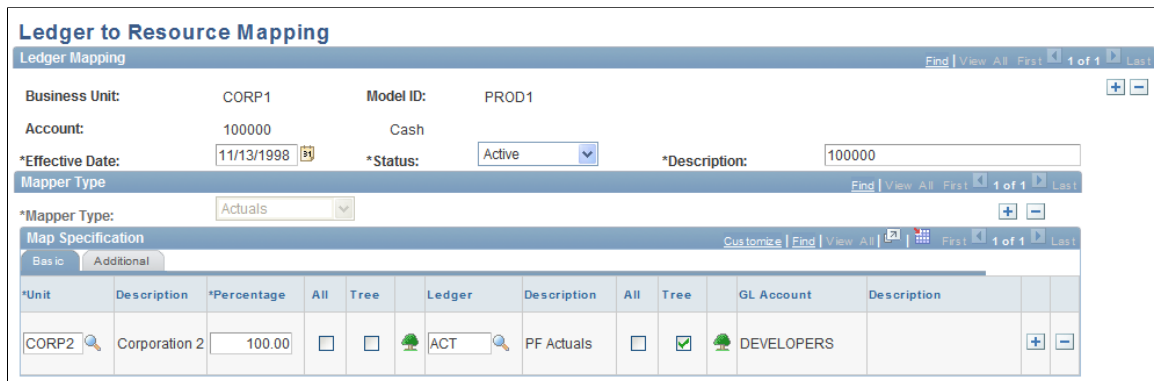
Use the Ledger to Resource Mapping page (LEDMAP_TBL1) to map ledger data, such as assets and liabilities, that do not enter the system through one of the analytic applications.

Navigation

EPM Foundation, Data Enrichment Tools, Performance Ledger, Map Ledger Data, Ledger to Resource Mapping

Image: Ledger to Resource Mapping page

This example illustrates the fields and controls on the Ledger to Resource Mapping page. You can find definitions for the fields and controls later on this page.



Mapper Type

Specify *Actuals* or *Budgeted* for mapper type.

This field is for the Activity-Based Management analytical application only.

Unit

Select the general ledger business unit to which you are mapping from the prompt list. Depending on your ledger mapping default setup, you might not have to enter information for the rest of the delivered ChartFields.

Description

Displays the description of the general ledger business unit.

Percentage

Use this field to split the ledger data from one general ledger business unit into multiple performance accounts or resources in Activity-Based Management. Do this by specifying that only a certain percentage from a ledger gets mapped.

For each delivered ChartField (ledger, general ledger account, operating unit, department, and product), enter the following information:

Description

Displays the description of each delivered ChartField value or tree node to include in the mapping.

All

Select this check box to bring all valid values for the specified ChartField (for example, to bring all department data to the performance account).

Tree

Select this check box to use values from a tree. If the *Tree Name not Defined* error appears, you must specify the tree name on the Ledger Mapping Defaults page. To create a new tree, use the PeopleTools Tree Manager.



Click the Tree button to specify the tree name and the tree node.

Related Links

[Specifying Ledger Mapping Defaults](#)

Creating Data Manager Rules for the GL Mapper Method

You can now set up the Data Manager rules. You must set up a Data Manager rule for each SetID for which you want to map ledger data. The data manager rule creates the SQL code for that SetID.

Related Links

[Defining Data Manager Rules](#)

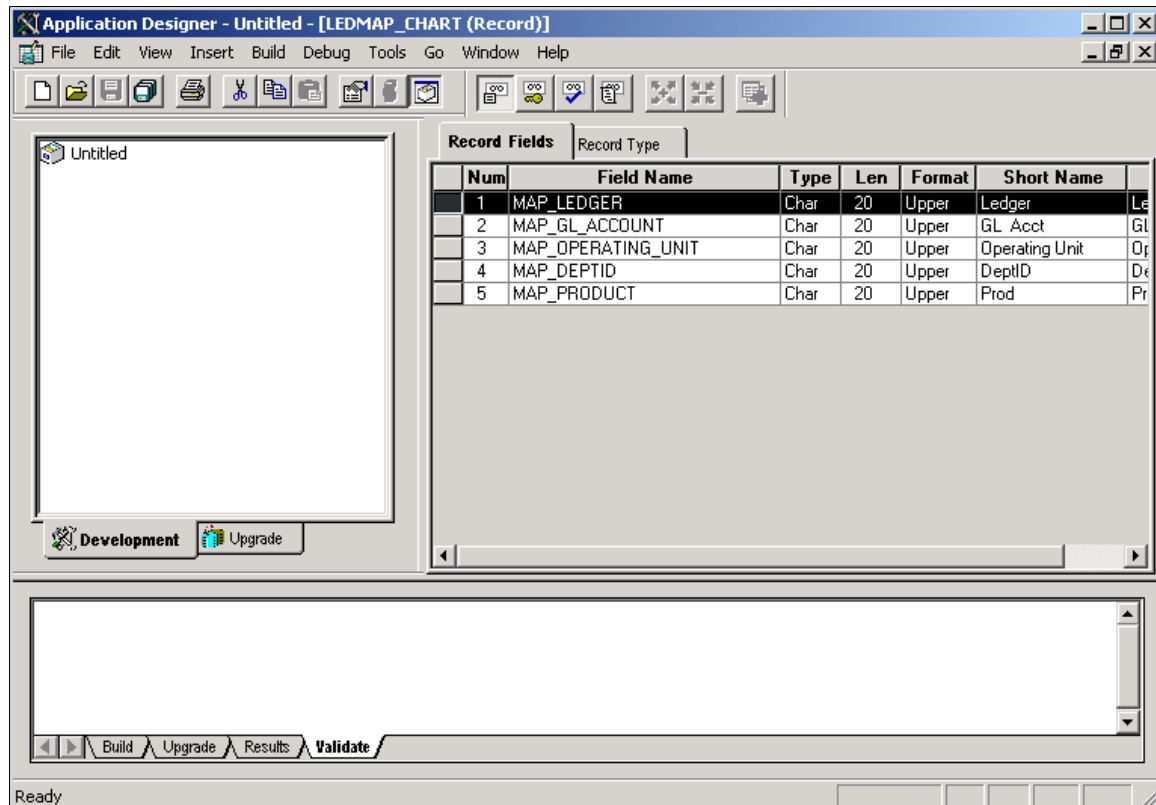
Configuring Ledger Mappings for New or Configured ChartFields

This topic describes how to configure ledger mapping with any configured or new ChartFields that you want to use with EPM. This process ensures that the ChartFields appear on the Ledger Mapping Defaults pages and other Ledger Mapper pages.

Warning! Be sure to save your original settings before configuring. Before you attempt any configurations, you should be familiar with the PeopleTools Application Designer and Application Engine, and your ChartField structure.

Image: Adding ChartFields to the LEDMAP_CHART subrecord

This example illustrates the Adding ChartFields to the LEDMAP_CHART subrecord.



To configure ledger mappings:

1. Add the ChartFields to the subrecord LEDMAP_CHART.

To open the record, launch the PeopleTools Application Designer and select File, Open. Open the LEDMAP_CHART subrecord and add the new ChartFields. Fields should be MAP_[name]. Make the new fields key fields.

Add ChartFields depending on whether a prompt table exists. When you are finished, save the record. Add the new ChartField to the LED_KEY2_SBR record and make it a key field. Also add the new ChartField to the LED_KNK_SBR record, but do not designate it as a key.

2. Add the ChartFields to the SQL object LEDMAP_CHART.

To open the SQL object, select File, Open in the Application Designer. Open the LEDMAP_CHART SQL object (object type: SQL), and add the new ChartFields. Fields should be MAP_[name].

3. Alter the corresponding tables and views for each of the modified subrecords.

You should alter them in the following order: LED_KNK_SBR, LED_KEY2_SBR, LEDMAP_CHART.

Note: You must use Find Object References to determine which tables and views need to be altered.

4. Alter the LEDMAP_SEQ grid on the page LEDMAP_TBL1.

In the Application Designer, access the page LEDMAP_TBL1. Go to the LEDMAP_SEQ grid and add or update the following fields:

Detail	Description
Check box CHARTFIELD_X_OPT*	Record DERIVED_LEDMAP (Short description) (Yes/No)
Edit box CHARTFIELD_X_OPT	Record LEDMAP_WORK (Long description, Invisible)
Check box CHARTFIELD_X_TOPT	Record LEDMAP_SEQ (Short description) (Yes/No)
Edit box CHARTFIELD_X_TREE	Record DERIVED_LEDMAP (Long description, Invisible)
Push Button CHARTFIELD_X_TSEL	Record DERIVED_LEDMAP (copy image from previous ChartField)
Edit box MAP_[name]	Record LEDMAP_SEQ (Long description)

Note: *X* refers to the ChartField number. If you insert the new ChartField between existing ChartFields on the LEDMAP_CHART subrecord, you must add the new fields and renumber the existing fields. The subrecord sequence number must match the CHARTFIELD_X_OPT.

Note: Before saving the grid and page, adjust the size of all boxes for All/Tree/Tree Name for ChartField.

5. Add the following fields to the PF_LED_TMPL_TBL1 page:

Detail	Description
Edit box FIELDNAMEX	Record DERIVED_LEDMAP (None, Invisible, Display Control) Label Text = "FIELDNAMEX"
Edit box CHARTFLD_X_TREEVW	Record PF_LED_TMPL_TBL (None)
Edit box CHARTFLD_X_NODEVW	Record PF_LED_TMPL_TBL (None)
Edit box SHORTNAME	Record PSDBFIELD (None, Display-Only, Related Display) Label Text = " FIELDNAMEX descr"

Note: *X* refers to the ChartField number. When you save the data, you can ignore the warning message *PSDBFLDLABL. SHORTNAME occurs more than once*.

If you receive any errors, check the layout order and reorder the fields as they should appear on the page.

6. Add the GL mapper rule for the current SetID on the Data Manager Rules Define Rule page if it is not there already.

Warning! If the ABMP rule already exists for the current SetID, you *must* rebuild the SQL by resolving the page.

7. To use trees, specify the appropriate tree view, tree node view, and tree name on the Ledger Mapping Defaults page.

Set up tree view and tree node views for the new ChartFields. Also, be sure to add the views under a record group control for the ChartField (add a new record group if one does not exist). Access the Ledger Mapper Defaults II page and set up a tree name if needed.

8. Correct existing ledger to resource mappings to reflect the new ChartField.

Use the All Values option as a default or specify a value or tree.

Creating, Editing, and Approving Manual Journals

This topic provides overviews of manual journals, the journal copy feature, and reverse journal entries, and discusses how to:

- Enter manual journals.
- Copy journals.
- Create reverse journal entries.
- Attach supporting documentation for manual journal entries.
- Approve journals.

Pages Used to Create, Edit, and Approve Manual Journals

<i>Page Name</i>	<i>Definition Name</i>	<i>Navigation</i>	<i>Usage</i>
Journal Entries	PF_JOURNAL_TBL1	EPM Foundation, Data Enrichment Tools, Profit Manager, Performance Journals, Journal Entries, Journal Entries	Enter or copy journals.
Journal Copy Options	PF_JOURNAL_CPY	Click the Copy button on the Journal Entries page.	Create journal copies.
Journal Reversal	PF_JOURNAL_TBL3	EPM Foundation, Data Enrichment Tools, Profit Manager, Performance Journals, Journal Entries, Journal Reversal	Select the option to create a reversal batch and the date of the reversal journal entries.

Page Name	Definition Name	Navigation	Usage
Journal Attachments	PF_JRNL_ATTACHMENT	EPM Foundation, Data Enrichment Tools, Profit Manager, Performance Journals, Journal Entries, Journal Attachments	Attach documentation to support journal entries.
Journal Approval	PF_JOURNAL_TBL2	EPM Foundation, Data Enrichment Tools, Profit Manager, Performance Journals, Journal Entries, Journal Approval	Set the approval status for a journal.

Understanding Manual Journals

Before you post your journals, you can use the Journal Entries page to manually record transactions and create journal debit and credit entries. Journals are written to a journal fact table (such as GC_JRNL_MGT_F00), which is the source record for the various ledger posting processes).

Warning! Before you post a journal, ensure that you have properly mapped the Foreign Currency Code in the Data Manager - Define Target page.

If you do not populate the Foreign Currency Code, unexpected results may occur when the PF_POST application engine runs.

See [Data Manager Rules - Define Target Page](#).

Understanding the Journal Copy Feature

You can copy a manual journal to:

- Record the same journal entry to multiple scenarios.

The journal must use the common consolidation business unit.

- Duplicate a journal for another fiscal year or period.
- Create reversal journal entries.

You can copy any existing valid or posted journal that was created with or copied from the Manual Journal Entries page.

If the base currency of the target journal's scenario is different from that of the source journal, the system calculates the new base amounts by:

1. Retrieving the currency exchange rate from the transaction currency and converting it to the new base currency, using the rate type that you specify on the line and the journal date as the currency effective date.

Rate type is required on all manual journals.

2. Computing the new base amount.

The new base currency amount is equal to the original transaction amount divided by the product of the rate divisor and rate multiplier that is applicable for currency conversion, rounded to the number of decimals that are specified for that base currency.

When copied, if the journal is out of balance because of rounding from any required currency conversion, the system adjusts the base amount of the first journal line.

Regardless of the source journal system source, the target journal system source is set to SCG, which means that the journal was system-generated and originated from the Journal Copy process.

Understanding Reverse Journal Entries

The Journal Reversal page enables you to create a reverse journal entry or reversal batch. The reversal batch is created in a valid (edited) status that does not require journal edits. Alternatively, you can copy your original journal, select the reverse amount check box, and designate the journal date to create a reversal journal.

Reverse journals are associated with your original journal by the journal ID. If you try to post a reversal journal to the system before posting the original, you receive an error message that the original journal has not yet been posted and the reversal does not post. If you should attempt to unpost the original journal that is associated with the reverse journal, you receive an error message specifying that this action cannot be performed. Similarly, if you attempt to unpost a reversal journal entry, you receive a warning saying that you must also unpost the original after unposting the reversal.

Journal Entries Page

Use the Journal Entries page (PF_JOURNAL_TBL1) to enter or copy journals.

Navigation

EPM Foundation, Data Enrichment Tools, Profit Manager, Performance Journals, Journal Entries, Journal Entries

Image: Journal Entries page

This example illustrates the fields and controls on the Journal Entries page. You can find definitions for the fields and controls later on this page.

The screenshot displays the 'Journal Entries' page with the following details:

- Journal ID:** 00283
- Journal Date:** 04/02/2010
- Status:** New
- Unit:** CORP1 Corporation 1
- Ledger:** ACTUALS
- Scenario:** ACTUAL Business Planning Actual
- Controls:** Calculate, Copy, Edit Journal, Post Journal, Draft, Hold Posting
- Journal Lines Table:**

Line	Unit	*Account	Alternate Account	Affiliate	Oper Unit	Department	Product	Project
1	CORP1	000000		CORP2	BERKELEY	1	ALLPRD	007
- Journal Totals Table:**

Unit	Total Lines	Base Currency	Total Debits	Total Credits
			0.000	0.000

Ledger ID

Select a ledger ID which is used to post the journal to the ledger.

Draft	Select this check box to create a draft journal. This option excludes the journal from the manual journal edit process.
Hold Posting	Select this check box to prevent the journal entry from posting during ledger post processing. This option enables you to save the journal entry and continue to add information to it at a later time.
Calculate	Click to initiate the calculate process and update the totals in the Journal Information region and Journal Totals grid. For each journal line on the journal line grid, the system compares all field values with the prior values that are stored in the buffer for any changes. If changes exist on currency, rate type, exchange rate, or base currency, the system determines the new exchange rate. This process also initiates exchange rate checking before the new exchange rate can be accepted. The new exchange rate is then converted to rate multiplier and rate divisor values. If any changes occur in amount, exchange rate, or base amount, the system calculates the unchanged amount based on the system rule or the rule that you specify in the calculate field. The system then adjusts the journal totals and balancing ChartField totals to reflect the new totals. Even if no amount field change occurs but a ChartField value change occurs, the system adjusts the balancing ChartFields totals accordingly.
Copy	Click to access the Journal Copy Options page and create a copy of this journal. <hr/> Note: This option is not available until you save the journal. <hr/>
Edit Journal	Click to edit (validate) your journal. Your journals cannot be posted until they are valid. If errors occurred on the journal, you can correct the errors with the Ledger Edit component.
Post Journal	Click this link to access the Post Ledger run control page to post the journal. <hr/> Note: This option is not available until the journal is valid (edited). <hr/>

Journal Information

Expand the Journal Information region to complete or review these fields:

Journal Source Code Select the source code for the journal.

Journal source codes enable you to track the source of your journal data. Source codes are defined with the Journal Source Code page.

Note: This is an optional field.

The system derives values in the Fiscal Year and Period fields from the scenario's calendar, based on the journal date. The system derives the value in the Base Currency field from the business unit definition. The amounts for total debits and total credits for the entire journal appear in the Journal Totals grid.

Journal Lines

Add rows to the Journal Lines grid to enter additional journal lines. For each line, you must complete the Account and Amount fields. If the transaction currency is not the same as the base currency, you must also specify values for the Currency and Rate Type fields. The system uses the associated exchange rate to populate the base amount. The consolidation dimension field label varies depending on the consolidation dimension. For example, if the consolidation dimension is business unit, the field label is Ledger Unit. The ledger template that is associated with the business unit for the journal entry determines the ChartFields that are available for the journal entry. You can use multiple business units within the same ledger ID.

Note: You can customize the ChartFields that appear by clicking the Customize link on the grid.

Journal Totals

Use the Journal Totals grid to view a summary of the journal entry. The information is grouped by dimension value (such as ledger unit) and lists the total number of lines and total debits and credits.

Journal Copy Options Page

Use the Journal Copy Options page (PF_JOURNAL_CPY) to create journal copies.

Navigation

Click the Copy button on the Journal Entries page.

Image: Journal Copy Options page

This example illustrates the fields and controls on the Journal Copy Options page. You can find definitions for the fields and controls later on this page.

Journal Copy Options

Copy Journal

*Journal ID: 00283 *Journal Date: 04/02/2010

Business Unit: CORP1 Scenario ID: ACTUAL

Create Reversal:

Journal Reversal

Next Period

Specific Period

Fiscal Year: Reversal Period:

To copy a journal:

1. Access the Manual Journal Entries page, select the journal that you want to copy, and then click Copy.

The Journal Copy Options page appears. The Copy Journal region lists the values that you are copying from the source journal.

2. Complete the fields in the Copy Journal group box:

Journal ID

Enter a journal ID for the new journal.

The default value in this field is the same value as the journal that you are copying; however, you can override this value.

Journal Date

Enter a date for the new journal.

If the target journal date is different from the source journal date, the system uses the target journal date to derive the fiscal year and accounting period values.

Create Reversal

Select this check box to create a reversal batch for this journal.

If you select this option, specify the period in which you want the reversing journal entry to occur. You can specify the next period based on your original journal date or by specifying a specific period in time.

3. Click OK to copy the journal.

The Journal Copied page summarizes the scenarios to which the journal was copied. Any copied journals that require approval (through workflow or security) have an initial approval status of *None*.

4. Review the journal and click Submit for Approval to initiate the approval process.

For preapproved journals, the approval status is immediately set to *Approved*, which copies the journal lines to the journal fact table that you specified on the ledger template.

Journal Reversal Page

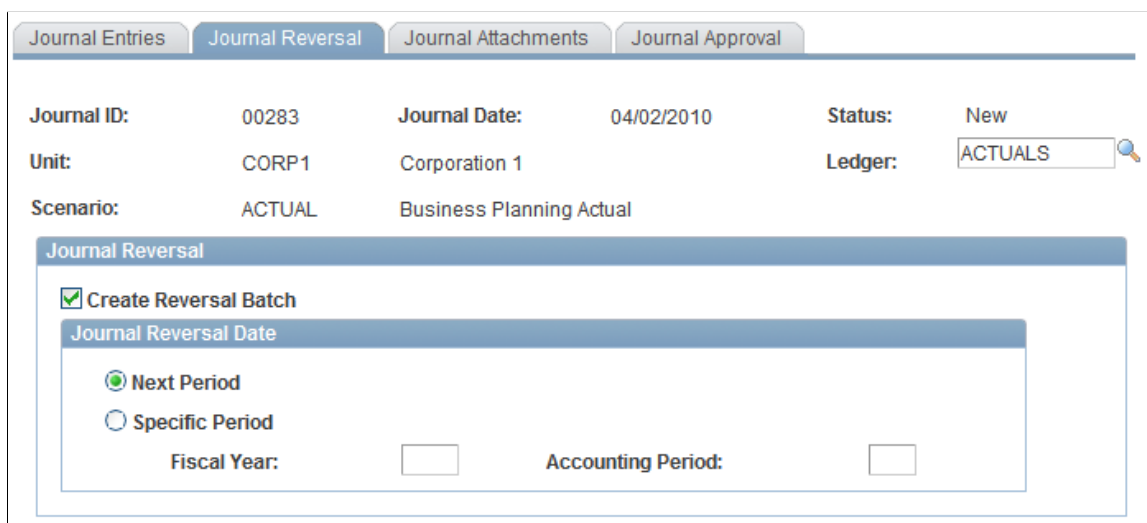
Use the Journal Reversal page (PF_JOURNAL_TBL3) to select the option to create a reversal batch and the date of the reversal journal entries.

Navigation

EPM Foundation, Data Enrichment Tools, Profit Manager, Performance Journals, Journal Entries, Journal Reversal

Image: Journal Reversal page

This example illustrates the fields and controls on the Journal Reversal page.



Select the Create Reversal Batch check box and specify the period in which you want the reversing journal entry to occur.

You can specify the next period based on your original journal date or by specifying a specific period in time.

Journal Attachments Page

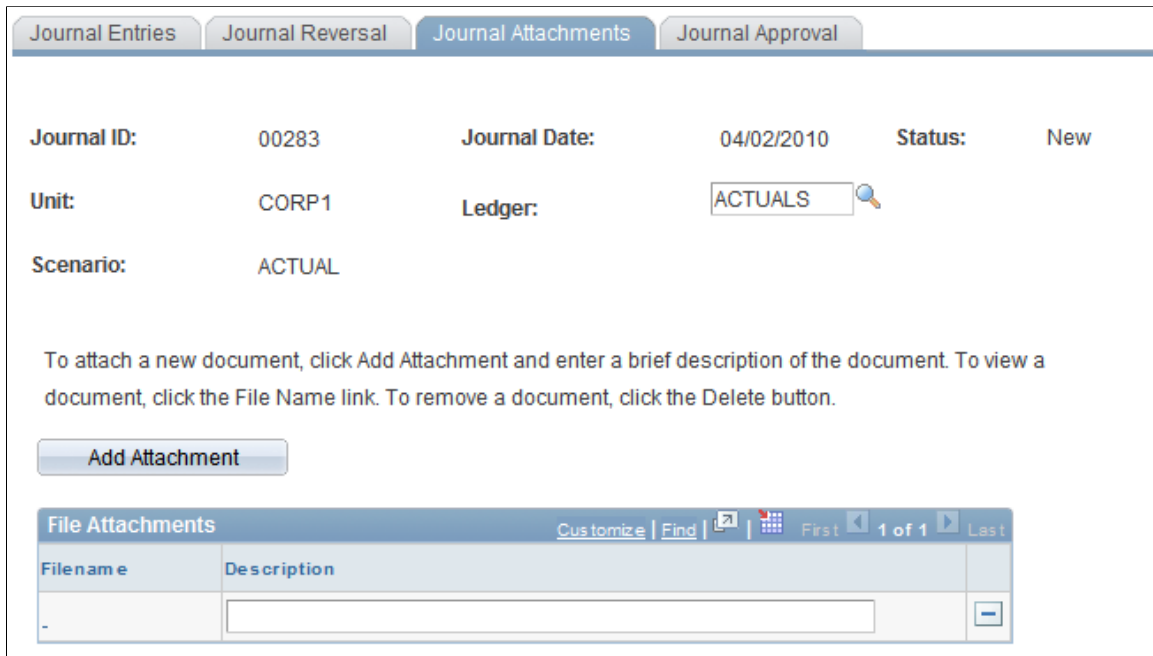
Use the Journal Attachments page (PF_JRNL_ATTACHMENT) to attach documentation to support journal entries.

Navigation

EPM Foundation, Data Enrichment Tools, Profit Manager, Performance Journals, Journal Entries, Journal Attachments

Image: Journal Attachments page

This example illustrates the Journal Attachments page.



This page enables you to add supporting documentation for your manual journal.

Click the Add Attachment button to locate the desired supporting documentation, select the file, and attach it to the page.

To view a document, click the File Name link. To remove a document, click the Delete button.

Journal Approval Page

Use the Journal Approval page (PF_JOURNAL_TBL2) to set the approval status for a journal.

Navigation

EPM Foundation, Data Enrichment Tools, Profit Manager, Performance Journals, Journal Entries, Journal Approval

Image: Journal Approval page

This example illustrates the fields and controls on the Journal Approval page. You can find definitions for the fields and controls later on this page.

The screenshot displays the 'Journal Approval' page with the following details:

- Journal ID:** 00283
- Journal Date:** 04/02/2010
- Status:** New
- Unit:** CORP1 Corporation 1
- Ledger:** ACTUALS
- Scenario:** ACTUAL Business Planning Actual

The 'Approval Status' section includes:

- Approval Status:** Approved
- *Approval Action:** Approve (dropdown menu)
- Comments for Denial Email:** (text area)
- Submit** button

Submit for Approval

Click to submit this journal entry for approval.

You set up approval rules on the PF Unit Scenario Definition page.

Approval Action

Select an approval status for the journal entry.

You can select from the following values:

None: Indicates a new, unsaved, or saved journal, requiring approval.

Approved: Indicates a journal that is approved (or was preapproved) and is eligible for posting.

The system does not copy journal lines into the journal fact table until the journal status is set to *Approved*. As a result, the journal is ineligible for posting until approved.

Pending: Indicates that the journal is submitted and awaiting approval.

When multiple approval steps are in place, the journal remains in pending approval status until all necessary parties approve the journal.

Denied: Indicates that the journal is denied approval.

Comments for Denial Email

Enter any comments that you want to include in the denial email as a result of the journal being rejected.

To update the approval status for a journal, select an approval action from the drop-down list box, and then click the Submit for Approval button.

Processing and Posting Journals

This topic provides overviews of journal processing and posting and the PF Journal Modification component and discusses how to:

- Run PF Journal Edit and PF Journal Re-Edit.
- View journal error statistics.
- Correct journal errors.
- Run PF Journal Cleanup.
- Run PF Ledger Post.
- Run PF Ledger Unpost.
- Unpost and repost transactions.

Pages Used to Process and Post Journals

<i>Page Name</i>	<i>Definition Name</i>	<i>Navigation</i>	<i>Usage</i>
PF Journal Modification - Journal Statistic	PF_JRNL_STATS	EPM Foundation, Data Enrichment Tools, Profit Manager, Performance Journals, Correct Errors, PF Journal Modification, Journal Statistic	View journal error statistics by viewing batches of data that are flagged as errors and moved to the PF_JRNL_E00 error table.
PF Journal Modification - Journal Correction	PF_JRNL_CORRECTION	EPM Foundation, Data Enrichment Tools, Profit Manager, Performance Journals, Correct Errors, PF Journal Modification, Journal Correction	View error fields and the number of errors.
PF Journal Modification - Error Details	PF_JRNL_SUMM	Click the Display Record Errors button on the Journal Correction page.	Correct errors as appropriate.
PF Journal Modification - Error Description	PF_JRNL_TSE	Click the Display Error Messages button on the Error Details page.	View error message details for an error.
Journal Cleanup	RUN_PF_DELB	EPM Foundation, Data Enrichment Tools, Profit Manager, Performance Journals, Journal Cleanup Report, Journal Cleanup	Run PF Journal Cleanup to delete journal data by batch ID.

Page Name	Definition Name	Navigation	Usage
Journal Cleanup - Batch Selection	PF_DELETE_BATCH	Click the Get Batch Data link on the Journal Clean Up page.	Specify whether to delete all displayed batches or use the check box to delete one batch at a time.
Ledger Post	RUN_PF_POST	EPM Foundation, Data Enrichment Tools, Profit Manager, Performance Journals, Post Ledger, Ledger Post	Run the PF Post engine to post data in the journal table that you have determined is valid.
Ledger Post - Batch Selection	PF_POST_BATCH	<ul style="list-style-type: none"> EPM Foundation, Data Enrichment Tools, Profit Manager, Performance Journals, Post Ledger, Batch Selection Click the Get Batch Data link on the Ledger Post page. 	Specify the batches that you want to post to the performance ledger.
PF Ledger Unpost	RUN_PF_UNP	EPM Foundation, Data Enrichment Tools, Profit Manager, Performance Journals, Unpost Ledger, PF Ledger Unpost	Run the PF Unpost to unpost any posted data by batch ID that you determine is invalid.
PF Ledger Unpost - Batch Selection	PF_UNPOST_BATCH	<ul style="list-style-type: none"> EPM Foundation, Data Enrichment Tools, Profit Manager, Performance Journals, Unpost Ledger, Batch Selection. Click the Get Batch Data link on the Ledger Unpost page. 	Specify the batches that you want to unpost from the performance ledger.

Understanding Journal Processing and Posting

After you set up and run engines such as Data Manager or Ledger Mapper, the enriched data is posted to a temporary table as specified on the Ledger Template page. You then run the PF Journal Edit engine to identify any balance errors and check for valid dimension values in the ChartFields that are specified on the journal entry. Valid data is written to the PF_JRNL_F00, while any errors that are found are written to the error table, PF_JRNL_E00. Use the PF Journal Modification engine to revise errors, and then run PF Journal Re-Edit to repopulate the PF_JRNL_F00 table. Finally, run the PF Ledger Post engine to populate the PF_LEDGER_F00 table with valid and accurate data from PF_JRNL_F00.

The PF Journal Edit process checks for TSE field errors. This process also checks the balancing rules for the jobs in the jobstream to verify whether the flash totals from these jobs are in balance based on the balancing rules that you defined for the jobs.

Input to the PF Journal Edit engine is always from the performance journal temporary table that is populated by other application engines. For this reason, you must always run the PF Journal Edit engine as a job after the application engine that populates the temporary journal table, but within the same jobstream. PF Journal Edit can never be the first or the only job in a jobstream.

Note: The edit process validates against any prompt tables within the record definition.

The purpose of the PF Journal Edit engine is to validate journals and ensure data integrity before you post to PF_LEDGER_F00. After you run PF Journal Edit for any journal, correct any TSE errors using PF Journal Modification, and then run a PF Re-Edit job. If you have missing dimensions in your first edit, use EPM Foundation dimension pages to add the missing dimensions.

The following table lists the PF Journal Edit batch statuses. These statuses appear on the Ledger Post - Batch Selection page. The system creates one record for each batch status change in the PF_Batch table to allow tracking of the batch status history:

Batch Status	Description
E	Batch has TSE errors and balance errors.
T	Batch has TSE errors but no balancing errors.
B	Batch has no TSE Errors but has balancing errors. A batch status is set to <i>B</i> if any of the balancing rules that are defined for the jobs in the jobstream with a balance rule type of control or force balance and the balancing rule total has an amount difference. The amount difference for the balancing rule with a force balance type is written to the PF_JRNL_F00 for the account that is related to the PF ledger event code and is defined for the balancing rule.
V	This is a valid batch. No TSE or balance errors exist. Batch is ready to post.
P	Batch has been posted to the ledger. Status is set by PF_POST engine.
S	Valid data for the batch has been force-posted to the PF Ledger table. Invalid data (TSE errors) still exists in PF_JRNL_E00 that can be modified and posted later. Status is set by PF_POST engine.
F	Batch has been force-posted to the PF Ledger table even though balance errors occurred for the batch. You can address the balancing error by posting the difference to the PF Ledger (force balancing). Status is set by PF_POST engine.
O	Batch has been force-posted to PF Ledger table even though balance and TSE errors occurred. Status is set by PF_POST engine.
U	Batch has been unposted from PF Ledger table. Status is set by PF_UNPOST engine.

Note: If the batch is valid, the data is moved to the PF_JRNL_F00 table. If the batch has TSE errors, you must run the PF Journal Modification engine. After correcting errors in PF Journal Modification, use the Journal Re-Edit process to move the corrected data to the PF_JRNL_F00.

The difference between the Journal Edit and Journal Re-Edit processes is that Journal Edit must always be run as a job after the application engine that populates the temporary journal table, but within the same

jobstream. (The input file for Journal Edit engine is temporary journal table). You can run Journal Re-Edit as a standalone job because its input file is a permanent table (PF_JRNL_E00).

Journal Re-Edit performs the following functions for all batches: It checks all the batches in PF_JRNL_E00 that have the same keys as the run parameters (business unit, scenario ID, fiscal year, and period) for TSE errors.

Understanding the PF Journal Modification Component

Use the PF Journal Modification component to correct any data that was flagged as an error during the PF Journal Edit process and moved to the PF_JRNL_E00 error table. After correcting errors, run the Journal Re-Edit process to perform TSE validation on data in the PF_JRNL_E00 table and change the data error flag to *No* if the data is no longer erroneous. (The Journal Post process moves corrected data in PF_JRNL_E00 to PF_JRNL_F00).

The PF Journal Modification pages are designed to enable you to correct actual dimension *values* in the data, not the dimension tables. You must use EPM Foundation Dimension pages to correct the dimension tables.

Journal Re-Edit performs TSE validation against all the error records in PF_JRNL_E00 that have the *same* run control parameters (business unit, fiscal year, accounting period, scenario ID). This means that if you run multiple edits for the same parameters at different times, (thus creating multiple batches with the same runtime parameters), you can correct all the batches and dimensions first, and then run Journal Re-Edit once for all batches. Likewise, you can correct one batch at a time, and run Journal Re-Edit after correcting each batch.

Running PF Journal Edit and PF Journal Re-Edit

The PF Journal Edit process checks for TSE field errors. If the edit process does not find any errors, the data is moved to the PF_JRNL_F00 table. If the edit process does find errors, you must correct them before proceeding. After correcting the errors you must run the Journal Re-Edit process which performs TSE validation on data in the PF_JRNL_E00 table, changes the data error flag to *No* if the data is no longer erroneous, and moves corrected data in PF_JRNL_E00 to PF_JRNL_F00.

Note: In EPM, you can use the performance ledger and the Average Daily Balance (ADB) ledger. Use of the ADB ledger is optional and all of the processing in the ADB ledger is the same as in the performance ledger. The type of balance that is stored is the only difference between the two ledger tables.

PF Journal Edit

You must run PF Journal Edit as a job after running the engine that loads journal data to the performance journal temporary table.

PF Journal Re-Edit

To run PF Journal Re-Edit, follow these steps:

1. Add the jobstream *RE_EDIT* to the *Jobstream* page and select *RE_EDIT* for the job ID.

Note: You must run PF Journal Re-Edit as a stand-alone job in its own jobstream.

2. Using the *Jobstream Record Suites* page, add the appropriate record suites to the *RE_EDIT* jobstream.

- Using the *Journal Re-Edit Report* page, create the run control for the *RE_EDIT* jobstream.

The *Journal Re-Edit Report* page can be accessed using the following navigation: EPM Foundation, Data Enrichment Tools, Profit Manager, Performance Journals.

Warning! You should never run this jobstream from the *Run Jobstream* page! You cannot specify a ledger ID on the *Run Jobstream* page and the engine will fail without it.

When defining your run control parameters, remember that the difference between the Rerun check box and the PF Journal Re-Edit engine is that the PF Journal Re-Edit engine is used to re-edit previous batches of journal entries that had TSE errors. Rerun is used after another *batch with the same parameters* has already been run.

If the status of the batch is other than posted (P,S, F, O) or unposted (U), you can rerun the batch by selecting the Rerun check box. The batch ID is added to the PF_DELBATCH_TBL and is used to clean up the data in the journal for the batch using the PF Delete Batch utility.

Related Links

[Understanding Jobstreams](#)

PF Journal Modification - Journal Statistic Page

Use the PF Journal Modification - Journal Statistic page (PF_JRNL_STATS) to view journal error statistics by viewing batches of data that are flagged as errors and moved to the PF_JRNL_E00 error table.

Navigation

EPM Foundation, Data Enrichment Tools, Profit Manager, Performance Journals, Correct Errors, PF Journal Modification, Journal Statistic

Image: Journal Statistic page

This example illustrates the fields and controls on the Journal Statistic page. You can find definitions for the fields and controls later on this page.

Journal Statistic		Journal Correction	
PF Batch ID:	ABM_PRO26	Ledger ID:	PF_LEDGER
Business Unit:	CORP1	Scenario ID:	ACTUAL01
		As Of Date:	01/31/1999
Record Statistics			
	Number of valid records:	0	Number of errors records: 81
Field Names	Valid Totals	Error Totals	
MONETARY_AMOUNT	0.000	1,326,981.810	
FOREIGN_AMOUNT	0.000	1,326,981.810	
STATISTIC_AMOUNT	0.000	0.000	
Last Edited Operator Id:	SAMPLE	Last Edited Timestamp:	12/14/1999 6:15:49PM
Last Corrected Operator Id:		Last Corrected Timestamp:	12/14/1999 6:15:49PM

PF Batch ID

The PF Journal Edit process assigns a PF batch ID to a batch of data going into the journal table. It uses the PF_BATCH_NUMBER that is stored in the warehouse business unit table as the next batch number. The edit process increments this batch number by one every time it processes a new batch of data. The PF batch ID is a combination of the scenario ID, the fiscal year, and the accounting period.

Business Unit, Ledger ID, Scenario ID, and As Of Date

These parameters were specified in the run control that was used to run the jobstream that contained the PF Journal Edit engine. The business unit and PF batch ID uniquely identify a batch of data to be modified.

Record Statistics

Use the Number of valid records and Number of error records display fields for an overview perspective on the amount of errors.

Last Edited Operator ID, Last Corrected Operator ID, Last Edited Timestamp, and Last Corrected Timestamp

These fields display information about who ran the last journal edit and when it was run.

PF Journal Modification - Journal Correction Page

Use the PF Journal Modification - Journal Correction page (PF_JRNL_CORRECTION) to view error fields and the number of errors.

Navigation

EPM Foundation, Data Enrichment Tools, Profit Manager, Performance Journals, Correct Errors, PF Journal Modification, Journal Correction

Image: Journal Correction page

This example illustrates the fields and controls on the Journal Correction page. You can find definitions for the fields and controls later on this page.

The screenshot shows the 'Journal Correction' page with the following parameters:

- PF Batch ID: ABM_PRO26
- Ledger ID: PF_LEDGER
- Business Unit: CORP1
- Scenario ID: ACTUAL01
- As Of Date: 01/31/1999

Below the parameters is an 'Error Field Summary' table:

Err Dtl	Field Name	Field Value	Number of Errors
	ACCOUNT		81

Field Name Displays the name of the field in error.

Field Value Displays the value for the field in error.

Number of Errors Displays the total number of errors.



Click the Error Details button to access the Error Details page, where you can view the error details for the field and values in error. You can also correct the dimension value to correct the error.

PF Journal Modification - Error Details Page

Use the PF Journal Modification - Error Details page (PF_JRNL_SUMM) to correct errors as appropriate.

Navigation

Click the Error Details button on the Journal Correction page.

Image: Error Details page

This example illustrates the fields and controls on the Error Details page. You can find definitions for the fields and controls later on this page.

The screenshot shows the 'Error Details' page with the following fields and controls:

- PF Batch ID: ABM_PRO26
- Ledger ID: PF_LEDGER
- Field Name: ACCOUNT
- Field Value: [Text Input]
- All Values
- From: [Text Input]
- To: [Text Input]

The 'Error Record Summary' table is as follows:

Err Msg	Account	Operating Unit	Department	Customer ID	Product ID	Channel ID	Dimension
1	[Text Input]						

Field Value

Specify a field value (dimension) to apply to the lines in error.

All Values

Select this check box to apply this value to all the rows that are listed. Deselect this check box to apply the value to specific rows only. Specify the rows using the From and To fields.

From and To

Specify the row numbers to which the new field value should be applied.

Specify a dimension to apply to the values in error.



Click the Display Error Messages button on this page to access the Error Description page for that error and review the error message.

Save your changes.

Journal Cleanup Page

Use the Journal Cleanup page (RUN_PF_DELB) to run PF Journal Cleanup to delete journal data by batch ID.

Navigation

EPM Foundation, Data Enrichment Tools, Profit Manager, Performance Journals, Journal Cleanup Report, Journal Cleanup

Image: Journal Cleanup page

This example illustrates the fields and controls on the Journal Cleanup page. You can find definitions for the fields and controls later on this page.

The screenshot shows the 'Journal Cleanup - Batch Selection' interface. At the top, there are tabs for 'Journal Cleanup' and 'Batch Selection'. Below the tabs, the 'User ID' is set to 'VP1', and there are links for 'Report Manager' and 'Process Monitor', along with a 'Run' button. The 'Run Control ID' is 'JRNLCLN01'. A section titled 'Journal Cleanup Parameter' contains the following fields: 'Program Name' (PF_DELB), 'When' (Once), '*Description' (empty text box), '*Business Unit' (CORP1), '*Scenario ID' (ACTUAL), and '*Job ID' (PF Journal cleanup (re-run)). There is also a 'Get Batch Data' link.

Business Unit and Scenario ID Select the business unit and scenario ID for which to delete batches.

Job ID Select *PF Journal cleanup* for this standalone job.

Get Batch Data Click to retrieve all batch data for the specified business unit and scenario ID and access the Batch Selection page.

On the Journal Cleanup - Batch Selection page , select the batches to clean up by selecting the Delete Batch Flag check box. Save your changes and return to the Journal Cleanup page to run the PF_DELB process.

Ledger Post Page

Use the Ledger Post page (RUN_PF_POST) to run the PF Post engine to post data in the journal table that you have determined is valid.

Navigation

EPM Foundation, Data Enrichment Tools, Profit Manager, Performance Journals, Post Ledger, Ledger Post

Image: Ledger Post page

This example illustrates the fields and controls on the Ledger Post page. You can find definitions for the fields and controls later on this page.

The screenshot shows the 'Ledger Post' page with a 'Batch Selection' tab. At the top, there are links for 'Report Manager' and 'Process Monitor', and a 'Run' button. Below these are fields for 'User ID' (VP1) and 'Run Control ID' (LED0010). A section titled 'Ledger Post Parameters' contains several fields: 'Program Name' (PF_POST), 'When' (Once), '*Description' (empty), '*Business Unit' (CORP1), '*Scenario ID' (ACTUAL), '*Fiscal Year' (empty), '*Period' (empty), and '*Job ID' (PF Ledger Post). A 'Get Batch Data' link is located at the bottom right of the parameters section.

Business Unit, Scenario ID, Fiscal Year, and Period

Select the business unit , scenario ID, fiscal year, and accounting period for which to post batches.

Job ID

Select *PF Ledger Post* for this standalone job.

Get Batch Data

Click to access the Batch Selection page and retrieve all batch data for the specified parameters.

Note: PF Ledger Post is the last step in the data movement and migration process before you generate reports. The data that reaches the performance ledger table (PF_LEDGER_F00) must be accurate. You can post multiple batches at one time. The only way to post to the ledger is from the performance journal table (PF_JRNL_F00).

Ledger Post - Batch Selection Page

Use the Ledger Post - Batch Selection page (PF_POST_BATCH) to specify the batches that you want to post to the performance ledger.

Navigation

EPM Foundation, Data Enrichment Tools, Profit Manager, Performance Journals, Post Ledger, Batch Selection. Click the Get Batch Data link on the Ledger Post page.

Image: Batch Selection page

This example illustrates the fields and controls on the Batch Selection page. You can find definitions for the fields and controls later on this page.



Click the Get Batch Data button to retrieve the batch data. The data appears in the Batch Information grid.



Click the Display Journal Error Report button to access the PF Journal Modification component to view and correct any errors. If you choose not to correct the errors, select the Force Post check box to post this batch to the performance ledger table.



Click the Display Balance Results button to see any balance errors that might exist. The display-only Balance Error Statistics page appears. If you choose not to correct the balance errors, select the Force Post check box to post this batch to the performance ledger table.

Force Post

Select this check box to force post batches that are in error to the performance ledger table. If a balance rule with force balance is used, the out-of-balance amount is posted to the account that is related to the ledger event code for the force balance rule. Only one force balance rule is allowed per batch.

Post

If you do not have any errors for the batches, the Post check box is automatically selected.

Save your changes and return to the Ledger Post page to run the PF Ledger Post engine and post the specified batches to the performance ledger table.

If you believe that you have posted in error, run the PF Unpost engine.

PF Ledger Unpost Page

Use the PF Ledger Unpost page (RUN_PF_UNP) to run the PF Unpost to unpost any posted data by batch ID that you determine is invalid.

Navigation

EPM Foundation, Data Enrichment Tools, Profit Manager, Performance Journals, Unpost Ledger, PF Ledger Unpost

Image: PF Ledger Unpost page

This example illustrates the fields and controls on the PF Ledger Unpost page. You can find definitions for the fields and controls later on this page.

The screenshot shows the 'PF Ledger Unpost' page with a 'Batch Selection' tab. At the top, there are fields for 'User ID' (VP1) and 'Run Control ID' (UNPOSTABLMED). To the right of these fields are links for 'Report Manager' and 'Process Monitor', and a 'Run' button. Below this is a section titled 'Ledger Unpost Parameters' containing several input fields: 'Program Name' (PF_UNP), 'When' (Once), '*Description' (empty), '*Business Unit' (CORP1), '*Scenario ID' (ACTUAL), 'Fiscal Year' (empty), 'Period' (empty), and '*Job ID' (PF Ledger Unpost). A 'Get Batch Data' link is located at the bottom right of the parameters section.

Business Unit, Scenario ID, Fiscal Year, and Period

Select the business unit, scenario ID, fiscal year, and accounting period for which to unpost batches.

Job ID

Select *PF Ledger Unpost* for this standalone job.

Get Batch Data

Click to access the Batch Selection page and retrieve all batch data for the specified parameters.

Note: The functionality of the Batch Selection page is identical to Batch Selection page for Ledger Post; please refer to that documentation for reference.

Note: Running the PF Ledger Unpost engine does not necessarily zero-out all the posted data. If the keys are the same for PF batch data during the PF Post process, the data is aggregated in the performance ledger table for those batches. If you unpost aggregated data, you might not obtain a zero balance unless you unpost all the batches for the same parameters. Use the run control ID to find the batches that you want to unpost.

Unposting and Reposting Transactions

To ensure data integrity, PeopleSoft designed the system to prevent you from reposting a batch that has already been posted. Suppose that you post a batch of transactions called Batch A and subsequently find that the data is invalid. You unpost the batch and make the necessary changes. To repost the batch, you must complete the following steps:

To repost a batch previously posted:

1. Run the entire jobstream again to repopulate PF_JRNL_F00 and create a new batch ID (in our example Batch B).

Your jobstream might include running engines such as Activity-Based Management, File Transfer Protocol, Data Manager, Allocation Manager, or Ledger Mapper, and the PF Edit engine. On the PF Journal Edit run control, make sure that you select the Rerun check box to ensure that the system flags the previously posted batch (Batch A in our example) to be deleted.

2. Use PF Journal Cleanup to delete obsolete records in PF_JRNL_F00 from the previously posted batch (Batch A).
3. Post the transactions to PF_LEDGER_F00 using the new PF batch ID (Batch B).

Note: You can also copy the journal to a new journal and post the copy.

Using Balancing and Reconciliation Features

This topic provides overviews of PF Reconciliation, job total metadata, and balance rules metadata and discusses how to:

- Review and define job totals metadata.
- Review and define balance rules metadata.
- Run PF Reconciliation.
- Review reconciliation results.

Pages Used to Set Up Balancing and Reconciliation

<i>Page Name</i>	<i>Definition Name</i>	<i>Navigation</i>	<i>Usage</i>
Job Total Metadata	PF_META_TOT_TBL1	EPM Foundation, Foundation Metadata, Metadata Creation and Editing, Job Total Metadata	Review and define job totals for various data that is used as input or output to the different source or target tables in the system.
Balance Rules Metadata	PF_META_BAL_TBL1	EPM Foundation, Foundation Metadata, Metadata Creation and Editing, Balance Rules Metadata	Review and define balance rules.
PF Reconciliation	RUN_PF_JOB	EPM Foundation, Data Enrichment Tools, Profit Manager, Performance Journals, Reconcile Balances, PF Reconciliation	Run the PF Reconciliation engine to compare to and from amounts between tables.

Page Name	Definition Name	Navigation	Usage
Job Totals	PF_RECON_TOT_DATA	EPM Foundation, Data Enrichment Tools, Profit Manager, Performance Journals, Review Bal/Recon (Balance/Reconcile) Results, Job Totals	A display-only page that shows the results of a job totals reconciliation.
Balance Rule Data	PF_RECON_RULE_DATA	EPM Foundation, Data Enrichment Tools, Profit Manager, Performance Journals, Review Bal/Recon (Balance/Reconcile) Results, Balance Rule Data	A display-only page that shows the results of a balance reconciliation.

Understanding PF Reconciliation

Several features are available for you to use to check data integrity and to verify that input equals output between tables.

Run the PF Reconciliation engine after populating the performance ledger table (PF_LEDGER_F00) to verify that the data that you processed through the system is accurately reflected. The job totals and balance rules metadata pages enable you to specify the fields in the tables that you want to track. You define and use job totals metadata to compare totals between sources such as the GL ledger table or the revenue table versus the target table and the performance ledger table, and find any differences for an accounting period of a business unit and scenario ID.

The PF Reconciliation engine first resolves the sources and destinations into temporary tables using the constraints that are given for the job totals. The engine verifies all the job totals and balancing rules that are defined under the reconciliation job that is being run and calculates the totals and the differences for the balancing rules.

Typically, you run the PF Reconciliation engine after a period to compare to and from amounts between tables, such as the REVENUE_F00 and the PF_LEDGER_F00 or the GL_LEDGER and the PF_LEDGER_F00. The system delivers job totals and balance rules metadata for these types of tables. However, if you want to compare any other source and target tables, you can create your own job totals and balance rule metadata. After running your reconciliation, use the inquiry pages to track your reconciliation job.

Understanding Job Total Metadata

Job total metadata has two purposes:

1. To define totals for the PF Record Summary.

For PF Record Summary, define job totals using the Record Name field.

2. To define totals for balancing rules.

For balancing rules, you define job totals using constraints. Constraints are used to define the scope of the data that is being totaled. The data could come from a single table, but the scope of the data to be considered could depend on data in other tables.

Note: You can toggle the record and constraint fields on the Job Totals page, depending on the job ID that you select. Predefined job total metadata is delivered with EPM .

Understanding Balance Rules Metadata

Balance rules enable you to verify the data throughout the system. For example, you can verify that a monetary amount that you entered into the system reached the performance ledger table (PF_LEDGER_F00). The three balancing rules include:

Informational

Use this balancing rule to review how many rows or how many monetary amounts are in the specified table, or to see information that is not critical but can be used to validate or identify a model. For example, for PeopleSoft Activity-Based Management, you can check that data reaches cost objects directly from resources instead of going through activities.

Informational balance rule differences enable you to post to the ledger.

Control

Use this balancing rule to determine whether an out-of-balance amount exists. The batch is set to balance error status and can be analyzed in a report. Control balance rule differences will not allow you to post to the ledger.

Force Balancing

An out-of-balance amount will not stop the post process and can be added to the performance ledger table to force balance. Force balancing can be defined on a rule that compares totals between a source and a destination. You can define only one balance rule for a source and a destination. The SetID MODEL has a predefined force balance account (FBAL). If you need to define a force balance account, do so on the Ledger Event Codes page.

Predefined balance rules metadata is delivered with EPM.

Related Links

[Defining Ledger Event Codes](#)

Job Total Metadata Page

Use the Job Total Metadata page (PF_META_TOT_TBL1) to review and define job totals for various data that is used as input or output to the different source or target tables in the system.

Navigation

EPM Foundation, Foundation Metadata, Metadata Creation and Editing, Job Total Metadata

Image: Job Total Metadata page

This example illustrates the fields and controls on the Job Total Metadata page. You can find definitions for the fields and controls later on this page.

The screenshot shows the 'Job Total Metadata' page with the following fields and values:

- SetID:** SHARE
- Job ID:** ABM
- Section:** ACT-DRV
- Total ID:** ACT-DRV
- *Effective Date:** 01/01/1900
- *Status:** Active
- *Description:** Total Base amt into Activities
- Constraint Code:** ACT-IN
- *Field Name:** CALC_AMT
- *Aggregate Function:** Summation

Description

The description that you enter here is used by the Metadata Search engine to find your metadata.

Constraint Code

Select a constraint code. Constraints are used in this case to define the scope of the data that is being totaled. This data could come from a single table, but the scope of the data to be considered could depend on data in other tables.

This field is not available if you are setting up job totals for the SUMM Job ID (PF Record Summary engine). For PF Record Summary, job totals are defined using the Record field.

Field Name

Select a field name. This field displays the primary record for the field name that you select. For record summary totals, select the record name. The field name identifies the field that you want to total. If the list box does not list the field that you expected, check your datamap definition.

Aggregate Function

For aggregate functions, you usually select *Summation* to ensure that all the amounts in the specified field are summed.

Select *Count* to count total records in the data (for example, to count the number of employee IDs).

Select *Average* to average the specified field name totals.

Balance Rules Metadata Page

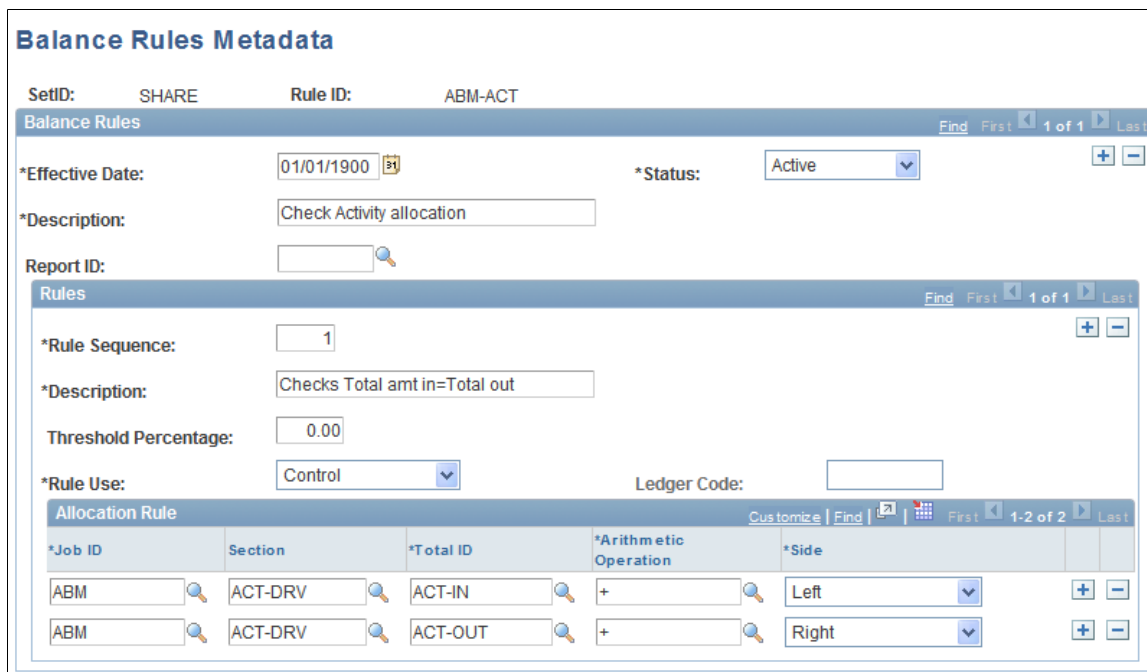
Use the Balance Rules Metadata page (PF_META_BAL_TBL1) to review and define balance rules.

Navigation

EPM Foundation, Foundation Metadata, Metadata Creation and Editing, Balance Rules Metadata

Image: Balance Rules Metadata page

This example illustrates the fields and controls on the Balance Rules Metadata page. You can find definitions for the fields and controls later on this page.



Description

The description that you enter is used by the Metadata Search engine to find your metadata.

Report ID

Select the ID of the report that you want to generate to help analyze any problems that caused the balance differences.

Rule Sequence

Enables you to group similar rules as subrules. To compare more than one balance field amount, click the add button in the Rule Sequence field to add a rule for a different balance amount field. For example, to compare posted total amounts, create rule sequence 1 and then click the Add button to add rule sequence 2 to compare base amounts.

Description

Enter a description that accurately defines the balance rule.

Threshold Percentage

Specify a percentage that reflects the highest percentage that you want to use for analysis. For instance, entering 5.00 in this field would indicate that if the difference between resources and activities in Activity-Based management is over five percent, the system should display the amounts. If the difference is not over five percent, post to the performance ledger table.

Rule Use

Select from the following rule uses:

Rule Use	Description
<i>Control</i>	Select <i>Control</i> if you are comparing tables for engine output amounts. For example, you would select <i>Control</i> and enter a threshold percentage if you were tracking whether the amount for resources equaled the amount that was driven to activities in Activity-Based Management. Any balance error is assigned a batch error status. Generate reports to analyze the balance error before posting.
<i>Force Balance</i>	Use <i>Force Balance</i> to post balance rule differences that should be posted to the PF Journal table. Then select the ledger event code that identifies the account to which the differences will be posted in the PF Ledger table.
<i>Info Only</i> (information only)	Select <i>Info Only</i> if you are comparing totals that are not critical (for example, the number of records). <i>Info Only</i> does not create a balance error batch. The batch will be valid.

Ledger Code

For a rule use of *Force Balance*, select the account to which the differences that are generated by force balancing should be posted in the performance ledger table.

Job ID

Select the job ID that is defined on the Job Metadata page.

Section

This is not a required field. Use this field to help you identify the problem area when an-out-of balance error occurs. The system uses the prompt table PS_PF_META_SEC_VW3. Before a section can be considered valid, it must first be defined in the engine metadata. Valid section codes that you enter on the engine metadata page are the actual application engine sections within the application engine program.

Total ID

Select the total ID that is defined in the Job Totals Metadata page.

Arithmetic Operation

Select an arithmetic operation (for example, + or -) to add or subtract multiple job totals. These operations are used with the Side field

Side

Select *Left* for the input total IDs, and *Right* for the output total IDs.

PF Reconciliation Page

Use the PF Reconciliation page (RUN_PF_JOB) to run the PF Reconciliation engine to compare to and from amounts between tables.

Navigation

EPM Foundation, Data Enrichment Tools, Profit Manager, Performance Journals, Reconcile Balances, PF Reconciliation

Image: PF Reconciliation page

This example illustrates the fields and controls on the PF Reconciliation page.

The screenshot shows the 'PF Reconciliation' page with the following fields and controls:

- User ID:** VP1
- Run Control ID:** PFREC01
- Program Name:** PF_RECON
- When:** Always (dropdown menu)
- As Of Dated Jobstream**
- *Description:** (text input field)
- *Unit:** CORP1 (text input field with search icon)
- *Scenario ID:** ACTUAL (text input field with search icon)
- Fiscal Year:** (text input field)
- Period:** (text input field)
- *Job ID:** RECN (text input field with search icon)
- Rerun**
- Last Run On:** (text input field)
- As Of Date:** (text input field)

Navigation links: [Report Manager](#), [Process Monitor](#), and a **Run** button.

The parameters on this page are described in the Streamlining Processing with Jobstreams topic earlier in this documentation.

After running PF Reconciliation, you can view your balance and reconciliation results using the Review Bal/Recon Results (review balance reconciliation results) component.

Note: You must include the PF Merge engine in a jobstream with the PF Reconciliation engine.

Related Links

[Running Jobstreams](#)

Job Totals Page

Use the Job Totals page (PF_RECON_TOT_DATA) to a display-only page that shows the results of a job totals reconciliation.

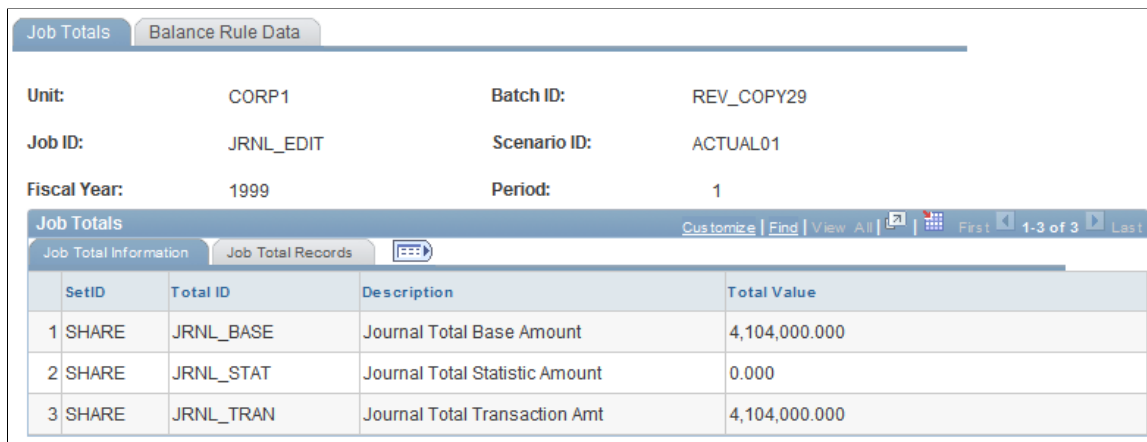
Navigation

EPM Foundation, Data Enrichment Tools, Profit Manager, Performance Journals, Review Bal/Recon (Balance/Reconcile) Results, Job Totals

Use the Job Totals page to review the job total information from your job. For example, you could use the total value on this page to verify that for Activity-Based Management, the total amount that was allocated to resources was transferred to the CALC_OBJ_F00 table.

Image: Job Totals page

This example illustrates the fields and controls on the Job Totals page.



This is a display-only page that shows the results of a balance reconciliation. If the Release Record Suite check box is selected, then this job has been released from a record suite.

Using Ledger Drill Down

After loading the performance ledger table (PF_LEDGER_F00), you can track data movement to and from the PF_LEDGER_F00 using the drill down feature. You can track performance ledger details and general ledger mapper details.

To build a search for ledger drill down, you must include the business unit, fiscal year, and accounting period when you specify your search keys.

This topic discusses how to:

- Specify drill-down criteria.
- View drill-down details.

Pages Used to Drill Down on Ledger Data

Page Name	Definition Name	Navigation	Usage
Drill Criteria	PF_DRILLSRCH_TBL1	EPM Foundation, Data Enrichment Tools, Profit Manager, Performance Journals, Ledger Details, Drill Criteria	Specify drill criteria for tracking data movement to and from PF_LEDGER_F00.

Page Name	Definition Name	Navigation	Usage
Drill Details	PF_DRILL_VW2	EPM Foundation, Data Enrichment Tools, Profit Manager, Performance Journals, Ledger Details, Drill Details	Review and drill down to the results in the PF_LEDGER_F00 table.

Drill Criteria Page

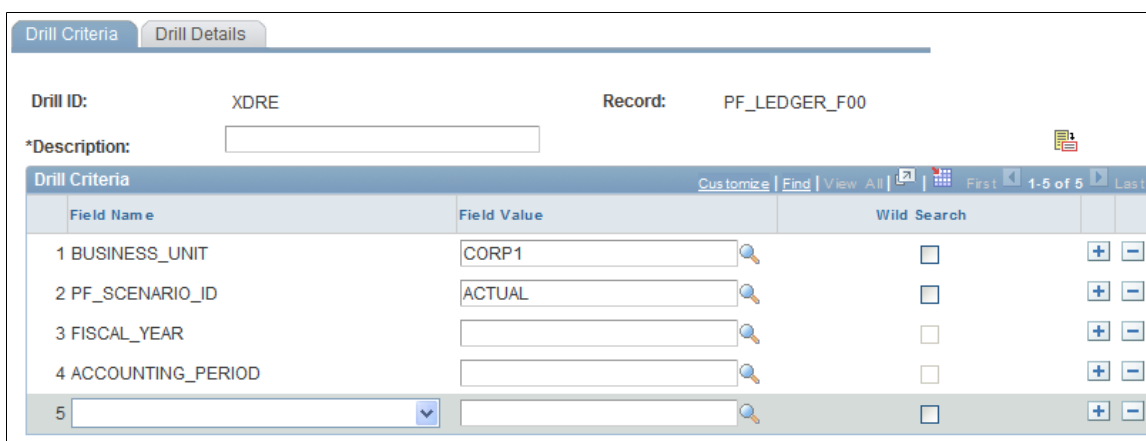
Use the Drill Criteria page (PF_DRILLSRCH_TBL1) to specify drill criteria for tracking data movement to and from PF_LEDGER_F00.

Navigation

EPM Foundation, Data Enrichment Tools, Profit Manager, Performance Journals, Ledger Details, Drill Criteria

Image: Drill Criteria page

This example illustrates the fields and controls on the Drill Criteria page.



Drill Criteria

Specify the drill criteria on which you want to perform the search. You can add further search criteria by adding a row and selecting the field on which to search. At a minimum, you must specify the business unit, scenario ID, fiscal year, and accounting period.

BUSINESS_UNIT Select the business unit.

PF_SCENARIO_ID Select the scenario ID.

FISCAL_YEAR Specify the fiscal year.

ACCOUNTING_PERIOD Specify the accounting period.

Field Value Specify your selections for any additional criteria that you add. The drill-down functions in the order in which you add the rows.

Wild Search

Alternately, you can select this check box to specify any values on which you want to search for the specified field.



When you have entered your criteria, click the Get Drill Data button to initiate your search.

The system retrieves the performance ledger data based on the entered search criteria. You can view the details on the Drill Details page.

Drill Details Page

Use the Drill Details page (PF_DRILL_VW2) to review and drill down to the results in the PF_LEDGER_F00 table.

Navigation

EPM Foundation, Data Enrichment Tools, Profit Manager, Performance Journals, Ledger Details, Drill Details

Image: Drill Details page

This example illustrates the fields and controls on the Drill Details page. You can find definitions for the fields and controls later on this page.

The screenshot shows the 'Drill Details' page with the following fields and controls:

- Drill ID:** XDRE
- Base Currency:** 0.000
- Description:**
- Total Base Amount:** 0.000
- Performance Ledger Data** section with tabs for 'Scenarios' and 'Amounts'.
- Table with columns: Drill, Unit, Year, Period, Scenario, Account, Source, Department, Dimension.
- Table content: Row 1 with a drill icon in the 'Drill' column.
- Navigation controls: Customize, Find, View All, First, 1 of 1, Last.

Use this page to view the performance ledger data that is based on your search criteria. You can view scenarios and amounts.



Click the Get More Drill Back Data button to access a detail page for a row of data. For example, if you are viewing data in Activity-Based Management, clicking the Get More Drill Back Data button accesses the ABM Ledger Data page (AB_DRILL_TBL2).

Reviewing Profit Manager Error Messages

When running any of the engines that are described in this topic, you can use the Engine Messages component, which is accessible from the Performance Ledger, Performance Journals menu, to review the progress of the engine and errors that were generated.

This component is described in detail in the Streamlining Processing with Jobstreams topic

Related Links

[Viewing Engine Messages](#)

Correcting Profit Manager Fact Table Errors Using PF Modification

This topic provides an overview of fact table error correction and discusses how to:

- View PF summary statistics.
- View and correct errors.

Understanding Profit Manager Fact Table Error Correction

After you have run the PF Summary process on a Profit Manager fact table, you can view and correct any errors that were found by opening the appropriate Profit Manager Fact Correction page. This process enables you to view and correct fact errors for your Profit Manager fact tables. For instance, you can use the Ledger Correction component to review and correct invalid data in the ledger fact table. Although there is a separate menu item and page for the Profit Manager Fact Correction process, the process is exactly the same for each.

There is an error table for each Profit Manager fact table. The error table contains all rows with errors that were found during data migration and is a mirror image of the target table, plus two extra columns called Error Flag and PF_EDIT_SEQ_NUM.

Note: Journal modifications are run separately using PF Journal Modification.

For each modification that you run, you must verify the delivered record metadata, record summary metadata, and job totals metadata are accurate.

PF Summary Statistics Page

Use the PF Summary Statistics page to view PF statistics.

Business Unit and PF Batch ID	Uniquely identifies a batch of data to be summarized or modified. The PF Batch ID is a combination of model ID, the fiscal year, and the accounting period.
Number of Valid Records and Number of Records with Errors	Review these display fields for an overview of the amount of errors. The field names on the left are defined in job totals metadata and are specified in record summary metadata.
Last Edited Time Stamp	These fields display information about who ran the last PF Record Summary and when the PF Record Summary was run.

Click on the Error Fields tab to continue with the modification process.

Error Fields Page

Use the Error Fields page to view and correct errors.

Field Name	Displays the field name of the error.
Field Value	Displays the value for the field name in error.
Number of Errors	Displays the number of records in error for the specified field.

You can correct fields by selecting the appropriate value to correct the specified column errors and clicking the Apply button. For other fields, you can click on the individual rows and correct those errors using the prompt table.

Using Data Enrichment Tools

Understanding Data Manager and Allocation Manager

This topic discusses:

- Data Manager.
- Allocation Manager.
- Mathematical model for Data Manager and Allocation Manager rules.

Data Manager

The Data Manager is a flexible tool that enables you to:

- Move data into the performance journal table (PF_JRNL_F00).
- Organize the output from any PeopleSoft EPM engine to create rows of data with multiple dimensions.
- Create reporting categories (dimensions) that represent groups of information for reports, such as business units, customers, products, channels, accounts (revenue, expense, and so on), and departments.

Note these key features of Data Manager:

- You can use Data Manager to consolidate or aggregate data and eliminate redundant rows.
- You can run Data Manager using different rules to organize and create multiple dimensions for direct costs and revenue data.

Data Manager supports the following methods for moving data:

Method	Description
Copy	Moves data from the source to the target. This method typically uses multiple-dimension data as the source.
Arithmetic	Applies an arithmetic operation using the amounts in both sources.
GL Mapper	Takes data from the general ledger tables and maps it to performance data. This method populates the temporary GLSTG table . From here, you can use the copy method to move data directly to the performance journal.

Method	Description
Prorata	Adds an additional dimension to your data and divides the amount across the dimension based on the percentage of the total amount.
Spread Even	Adds an additional dimension to your data and spreads the fact amount evenly across the dimension.
Tree Aggregation	Aggregates measures based on a tree hierarchy. The node names act as dimensions.

Data Manager and the Profit Manager

Data Manager works with the Profit Manager tools by moving data to the performance journal table (PF_JRNL_F00). The Profit Manager tools verify this data (PF_EDIT engine) and post the data to the performance ledger (PF_POST engine). They also unpost data, if necessary (PF_UNPOST engine) and can clean up the performance journal table (PF Journal Cleanup engine). The Profit Manager includes tools for balancing and reconciling your data.

The performance ledger, PF_LEDGER_F00, is a useful table for reporting because it contains all of the performance fact data. The facts have been allocated across several dimensions of your choice (typically channel, product, customer, and department). Having all of this information organized in a single table makes it easier for you to query and create reports.

The Profit Manager tool is described in the [Understanding Profit Manager Tools](#) topic.

Data Manager Setup

To set up the Data Manager, you need to:

1. Define the necessary metadata (including tablemaps, datamaps, constraints, and filters).

The system uses datamaps and constraints to point to the appropriate tables.

2. Define Data Manager rules, including the method by which you want to move, aggregate, or create multidimensional data, as well as the sources and target of the process.
3. Create a Data Manager rule set that contains one or more rules.
4. Set up job metadata and jobstreams.
5. Associate the Data Manager rule set with a job.
6. Run the Data Manager engine.

Allocation Manager

Allocation Manager is an EPM tool that enables you to distribute revenue, expense, and statistical amounts across business units, departments, and other dimensions. For example, you can allocate budget planning to detail levels to perform detailed budgeting.

The Allocation Manager tool can also be used to create offset and residual rows, capture source, basis, and target data, and create, process, and post journals to a ledger.

Allocation Manager consists of several PIA components and an application engine. You create the rules and rule sets that define your processing with PIA pages, while the application engine is executed using a jobstream and a run control PIA page.

Each allocation output is determined by the type of allocation method that you select. The following table lists the types of allocations that are supported by the Allocation Manager and describes each allocation type:

Allocation Type	Description
Arithmetic Operation	Defines a mathematical calculation using the source and basis, such as source + basis.
Prorata	Divides the source amount proportionately among the targets based on basis measures.
Spread Even	Distributes the source amounts equally by the specified basis fields. For instance, if expenses were spread evenly across four business units, each would have 25 percent of the expense.
Copy	Copies the source amounts to the targets.

Allocation Manager can use any source dimension within the OWE and provide output to any target. As a result, you can use sources from Global Consolidations and ABM results to further manipulate the output for analysis based on business rules specific to your organization.

Allocation Manager Dimensions

Due to platform limitation issues on DB2 UDB for OS / 390 and z/OS (the index size is limited to 255 characters) and Oracle (which requires 30 columns in an index) the predelivered index is on the first 10 dimensions. However, based on your requirements and the database platform, you can increase the maximum number of dimensions in the Allocation Manager to 28. The records that need to be modified for index changes include PF_AL_CALC_T, PF_AL_DIFF_T, PF_AL_DIV_T, PF_AL_SRC01_T, PF_AL_BAS02_T, and PF_AL_TOTAL_T.

To increase the number of dimensions:

1. Open each of the above records in the Application Designer.
2. Open the relevant subrecord.

The delivered unique index is shown on the fields PF_AL_DIM1 to PF_AL_DIM10.

3. Depending on the number of dimensions you are adding, modify the index by adding the extra dimensions as keys.

For example, if you want to use 15 dimensions then modify the index to include PF_AL_DIM11 through PF_AL_DIM15 and build the record. Alternatively, you can add a custom index on the table, using the Add Index feature.

Note: You can also apply these steps to Data Manager.

Mathematical Model for Data Manager and Allocation Manager Rules

The rules for Allocation Manager and Data Manager represent mathematical operations to be performed on the data you specify. Although Allocation Manager and Data Manager use different terminology, they perform similar calculations. For Allocation Manager, you must specify a source, basis, and target for the operation. For Data Manager you must specify two sources (source 1 and source 2) and the target.

The following table lists the Data Manager and Allocation Manager terms and how they relate.

Allocation Manager	Data Manager
Source	Source 1
Basis	Source 2
Target	Target

The *sources* of a rule use constraints to tell the Data Manager *what* is to be assigned and *what* basis to use. Use constraints to apply business rules to limit row selection. The *target* of a rule uses a datamap to tell the Data Manager *where* the rule should be located. The method that is applied to the rule determines the calculation that is performed.

Note: The explanation of the mathematical operations uses only Allocation Manager terms. Use the previous table to apply the Data Manager terms.

Understanding the Calculations Behind the Prorata and Spread Even Methods

The prorata and spread even methods perform the same calculation with the following exception: the prorata method uses measures from the basis table in the calculation, whereas the spread even method uses row counts from the basis to determine the ratio that is defined by $\text{basis measure} \div \text{basis total}$.

The prorata method performs the following calculation:

$$\text{Source Measure} * \text{Basis Measure} / \text{Basis Total}$$

Source Measure = Sum(Measures in source grouped by common dimensions and source mapped dimensions)

Basis Measure = Sum(Measures in basis grouped by common dimensions and basis mapped dimensions)

Basis Total = Sum(Measures in basis grouped by common dimensions only)

The spread even method performs the following calculation:

$$\text{Source Measure} * \text{Basis Measure} / \text{Basis Total}$$

Source Measure = Sum(Measures in source grouped by common dimensions and source mapped dimensions)

Basis Measure = Sum(Count of basis rows grouped by common dimensions → and basis mapped dimensions)

Basis Total = Sum(Count of basis rows grouped by common dimensions → only)

The following tables provide an example of the prorata method with one common dimension (unique dimension combinations).

Source:

Source Product	Source Amount
A	10
B	20

Basis:

Basis Product	Basis Channel	Basis Measure
A	X	10
A	Y	20
B	V	10
B	W	40

Target:

Target Product	Target Channel	Target Amount
A	X	3.33 (10 * 10/30)
A	Y	6.66 (10 * 20/30)
B	V	4 (10 * 10/50)
B	W	16 (20 * 40/50)

This table illustrates an example of the spread even method with one common dimension (unique dimension combinations).

Source

Source Product	Source Amount
A	10

Source Product	Source Amount
B	20

Basis:

Basis Product	Basis Channel
A	X
A	Y
B	V
B	W

Target:

Target Product	Target Channel	Target Amount
A	X	5 (10 * 1/2)
A	Y	5 (10* 1/2)
B	V	10 (20 * 1/2)
B	W	10 (20 * 1/2)

Understanding the Calculations Behind the Arithmetic Method

The arithmetic method performs the following calculation:

$$\text{Source Measure} [\text{Operator}] \text{Basis Measure}$$

Source Measure = Sum(Measures in source grouped by common dimensions and source mapped dimensions)

Basis Measure = Sum(Measures in basis grouped by common dimensions and basis mapped Dimensions)

Operator = [Addition (+), Subtraction (-), Multiplication (*), Division (/)]

Allocation Using Fixed Percentages

In Allocation Manager, you can divide a quantity by predetermined percentages and allocate those amounts by using the prorata method in combination with the fixed basis option.

In the calculation that is performed by the prorata method, the basis determines the ratio by which the source is divided. A ratio is another way to specify a percentage. By controlling the ratio, you can allocate the source amounts by specified percentages, or a fixed percentage.

$$\text{Source Measure} * (\text{Basis Measure} / \text{Basis Total})$$

or

$$\text{Source Measure} * (\% \text{Percentage})$$

Warning! The prorata method always allocates 100 percent of the source. You must use the correct percentages when defining the fixed basis.

Setting Up Allocations

To set up allocations:

1. Complete your metadata setup, including datamap setups for the source, basis, and target definitions.

This step is required.

Note: PeopleSoft provides the *Datamap Wizard* to greatly simplify the creation of datamaps. The Datamap Wizard can be found using either of these navigation paths:
 EPM Foundation, Foundation Metadata, Metadata Wizards, Datamap Wizard
 EPM Foundation, Data Enrichment Tools, Allocation Manager, Datamap Wizard

See [Datamap Wizard Page](#).

2. Define value objects.

This step is optional.

3. Create indices on Allocation Manager temporary tables.

This step is optional.

4. Define fixed dimensions.

This step is optional if you do not use a fixed source or basis in your allocations.

5. Define allocation rules, including the method, source, basis, and target definitions.

This step is required.

6. Create an Allocation Manager rule set that contains one or more allocation rules.

This step is required.

7. Associate Allocation Manager rule set with a job.

This step is required.

8. Process allocations using Run Allocations engine.

This step is required.

Setting Up Value Objects

A value object provides descriptive information about fields and values. Value objects enhance the power of filters and can use constant strings to fill in target fields. Used in a filter, value objects play a role in the constraint's WHERE clause of a SQL command to enforce selection rules. There are two value object classes:

Meta Value

A PeopleSoft variable that returns a field that is maintained by the system, such as the current system date, current system time, user ID, and so on, or a value that was entered as a parameter on the Data Manager run control prior to execution of Data Manager. Parameters include business unit, scenario ID, fiscal year, period, and so on.

Fixed Value

A user-defined constant value. Examples of fixed values that are useful in the Data Manager include 0 (zero), and N (for yes/no fields).

Note: PeopleSoft delivers predefined value objects with EPM. However, you might need to define your own using the Value Object page.

This topic discusses how to define value objects.

Page Used to Set Up Value Objects

<i>Page Name</i>	<i>Definition Name</i>	<i>Navigation</i>	<i>Usage</i>
Value Object	MD_VALOBJ_TBL1	EPM Foundation, Foundation Metadata, Metadata Creation and Editing, Value Object	Define or review value objects.

Value Object Page

Use the Value Object page (MD_VALOBJ_TBL1) to define or review value objects.

Navigation

EPM Foundation, Foundation Metadata, Metadata Creation and Editing, Value Object

Image: Value Object page

This example illustrates the fields and controls on the Value Object page. You can find definitions for the fields and controls later on this page.

Value Object ID

A unique identifier for the value object. All value objects are denoted in the system by a % at the beginning of the ID.

Field Type

Select the field type. Choices are: *Amount*, *Basis Pts* (basis points), *Char* (character), *Date*, *Datetime*, *Number*, and *Rate*.

Value Class

Select from the following:

Meta Value: Meta values are meta SQLs that retrieve system dates, user IDs, and engine run control parameters. If you select this option, you must specify the state record and state record field.

System Variables: System variables are values referencing system information (for example, the current date and time (%CurrentDateTime)).

Fixed Value: Fixed values are user-defined strings of text or numeric values that might be useful in specifying the definition of an assignment object or rule. These are usually constants.

If you select system variables or fixed values, the state record, state record field, quotes check box, and use tree check box are unavailable for entry. You must select the prompt table and enter a value.

State Record and State Record Field For *Meta Value*, enter the state record and state record field.

Quotes

For *Meta Value*, select this check box for character strings. In most cases, you will not select this check box for numeric values.

Use Tree	For <i>Meta Value</i> , select this check box if you want to use trees. The Prompt Table field changes to a Tree ID field and you can select your tree ID.
Prompt Table	Use to select a prompt table as needed.
Value	For <i>Meta Value</i> , the value field is automatically populated based on the state record and state record field selections. For example, if you select EPM_CORE_AET as your state record and BUSINESS_UNIT as your state record field, the value field is populated with : %BIND(EPM_CORE_AET.BUSINESS_UNIT). For <i>Fixed Value</i> , enter a character constant in quotes or a numeric constant without quotes. For <i>System Variables</i> , enter the variable to use.
Meta Value	Displays the meta value for any meta value objects.

Creating Indices on Allocation Manager Temporary Tables

Allocation Manager uses intermediate tables to temporarily store the results of an allocation calculation. When you perform allocations on large amounts of data, it is very important that indices are used correctly on all referenced temporary tables; this assures quality performance from the Allocation Manager. You should examine the indices on the temporary tables and ensure that they have been defined correctly, and are being used by the allocation SQL.

The intermediate tables used by Allocation Manager are:

- PF_AL_CALC_T
- PF_AL_DIFF_T
- PF_AL_DIV_T
- PF_AL_SRC01_T
- PF_AL_BAS02_T
- PF_AL_TOTAL_T

Due to platform limitation issues the prepackaged index is on the first 20 dimensions. Allocation Manager supports 30 dimensions. Depending on the number of dimensions you are using, modify the indices by adding the extra dimensions as keys, and rebuild the tables.

Defining Data Manager Rules

This topic provides an overview of Data Manager rule methods, lists prerequisites, and discusses how to:

- Define a Data Manager rule.
- Define sources.
- Define the target.
- Review and define the source 1 and source 2 columns.
- Review the mapped common dimensions.
- Review the generated SQL.

Pages Used to Set Up Data Manager Rules

Page Name	Definition Name	Navigation	Usage
Data Manager Rules - Define Rule	MD_RULE_TBL1	EPM Foundation, Data Enrichment Tools, Data Manager, Rules, Define Rule	Define a data manager rule, including the method you intend to use.
Data Manager Rules - Copy Rule	MD_RULE_COPY	Click on the Save As button on the Define Rule page.	Copy a Data Manager rule and save it under a new rule name.
Data Manager Rules - Define Source 1	MD_RULE_SCR1	EPM Foundation, Data Enrichment Tools, Data Manager, Rules, Define Source 1	Define source 1 information.
Data Manager Rules - Define Source 2	MD_RULE_SCR2	EPM Foundation, Data Enrichment Tools, Data Manager, Rules, Define Source 2	Define source 2 information.
Data Manager Rules - Define Target	MD_RULE_TBL3	EPM Foundation, Data Enrichment Tools, Data Manager, Rules, Define Target	Define the target for the rule and the target mapping.
Data Manager Rules - DataSet for Source 1	MD_DATASET1_DEFN	EPM Foundation, Data Enrichment Tools, Data Manager, Rules, DataSet for Source 1	Review and define the source 1 fields and measures that are mapped to target fields.
Data Manager Rules - DataSet for Source 2	MD_DATASET2_DEFN	EPM Foundation, Data Enrichment Tools, Data Manager, Rules, DataSet for Source 2	Review and define the source 2 fields and measures that are mapped to target fields. This page displays the prorata and spread even methods only.
Data Manager Rules - DataSet for Common Dimensions	MD_DATASET3_DEFN	EPM Foundation, Data Enrichment Tools, Data Manager, Rules, DataSet for Common Dimensions	Review the mapped common dimensions and prorata measure.

Page Name	Definition Name	Navigation	Usage
Data Manager Rules - SQL	MD_RULE_TBL4	EPM Foundation, Data Enrichment Tools, Data Manager, Rules, SQL	Review the generated SQL for the Data Manager rule.

Understanding Data Manager Rules and Methods

Data Manager rules use metadata to specify the source and the target tables for moving, aggregating, or creating multidimensional engine output.

Most rules have two sources: the measure to be assigned (for example Activity-Based Management indirect costs, revenue amounts, or inventory quantities) and the basis for the assignment (for example, per cent sales by region, product, or channel). Another datamap is selected as the target of the rule.

Data Manager rules define three things: source tables, targets, and the method that you want to apply to the data from the source tables as it moves to the target. These methods are:

- Copy method.
- Arithmetic method.
- GL Mapper method.
- Spread even method.
- Prorata method.
- Tree aggregation method.

Copy Method

The copy method moves data from the source to the target, but does nothing to the data. You would typically use multidimensional data as the source.

An example of this method would be the copy of REVENUE_F00, which is already attributed to the customer, product, and channel dimensions, to PF_JRNL_F00. In this move, no spreading of the revenue amount from the source to the target occurs.

In addition, this example shows that you can also use the copy method to generate aggregate amounts since more than one row of data might exist in source 1 for a customer/product/channel intersection. The copy method allows this type of aggregation by using the sum column on the Define Target page.

Source 1 datamap definition:

- DIMs: CUST_ID, PRODUCT_ID, CHANNEL_ID
- MSR: MONETARY_AMOUNT

Define Target page:

- Target fields of CUST_ID, PRODUCT_ID, CHANNEL_ID mapped to corresponding source 1 fields
- Sum check box selected for MONETARY_AMOUNT and mapped to source 1

Source 1 data:

<i>Row</i>	<i>CUST_ID</i>	<i>PRODUCT_ID</i>	<i>CHANNEL_ID</i>	<i>MONETARY_AMOUNT</i>
1	CU1111	PR111	CH11	1000.00
2	CU1111	PR111	CH22	2000.00
3	CU1111	PR222	CH11	1000.00
4	CU1111	PR222	CH11	2000.00
5	CU1111	PR222	CH22	3000.00
6	CU2222	PR111	CH11	1000.00
7	CU2222	PR111	CH22	3000.00
8	CU2222	PR222	CH11	3000.00
9	CU2222	PR111	CH22	1000.00

Target results:

<i>Row</i>	<i>CUST_ID</i>	<i>PRODUCT_ID</i>	<i>CHANNEL_ID</i>	<i>MONETARY_AMOUNT</i>
1	CU1111	PR111	CH11	1000.00
2	CU1111	PR111	CH22	2000.00
3	CU1111	PR222	CH11	3000.00 (Aggregated)
4	CU1111	PR222	CH22	3000.00
5	CU2222	PR111	CH11	1000.00
6	CU2222	PR111	CH22	4000.00 (Aggregated)
7	CU2222	PR222	CH11	3000.00

Arithmetic Method

The arithmetic method applies an arithmetic operation (for example, add, subtract, multiply, or divide) to combine amounts in the sources that are grouped by common dimensions and source (or basis) mapped dimensions.

GL Mapper Method

The GL Mapper takes data from the general ledger tables and maps it to performance data. It populates the temporary table GLSTG. From GLSTG, you can use the copy method to move data directly to the performance journal or apply other methods to enrich your data.

For example, let's say that source 1 looks like this:

<i>warehouse business unit</i>	<i>Account</i>	<i>Customer</i>	<i>Amount</i>
CORP1	110	C1	1500 USD
CORP1	111	C1	1500 USD
CORP1	110	C2	1000 USD
CORP1	111	C2	1000 USD

If you map 100 percent of the amounts from source accounts 110 and 111 to performance account 100000, the target would look like this:

<i>warehouse business unit</i>	<i>Account</i>	<i>Customer</i>	<i>Amount</i>
CORP1	100000	C1	3000 USD
CORP1	100000	C2	2000 USD

Spread Even Method

The spread even method adds a dimension to your data, and spreads the fact amount evenly across the dimension. For example, you have revenue for three products as follows:

Source 1:

<i>Product</i>	<i>Amount</i>
P1	3000 USD
P2	2000 USD
P3	1000 USD

Source 2:

<i>Customer</i>
C1
C2

The target looks like this:

Product	Customer	Amount
P1	C1	1500 USD
P1	C2	1500 USD
P2	C1	1000 USD
P2	C2	1000 USD
P3	C1	500 USD
P3	C2	500 USD

The amounts for product are spread evenly across the customers.

Prorata

The prorata method adds an additional dimension to your data and divides the amount across that dimension based on the percentage of the total amount. For example, you have revenue for three products as follows:

Source 1:

Product	Amount
P1	3000 USD
P2	2000 USD
P3	1000 USD

You also want to attribute that revenue to customers. The revenue by customers is:

Source 2:

Customer	Amount
C1	4000 USD
C2	2000 USD

The target looks like this:

Product	Customer	Amount
P1	C1	2000 USD

Product	Customer	Amount
P1	C2	1000 USD
P2	C1	1333.33 USD
P2	C2	666.67 USD
P3	C1	666.67 USD
P3	C2	333.33 USD

The amounts for the products are prorated based on the percentage of total for each customer (C1 has two-thirds, or 66.67 percent, and C2 has one-third or 33.33 percent).

Tree Aggregation

In the tree aggregation method, measures are aggregated based on a tree hierarchy. The node names act as the dimension. For example, assume you have the following data for individual customer IDs.

Source 1:

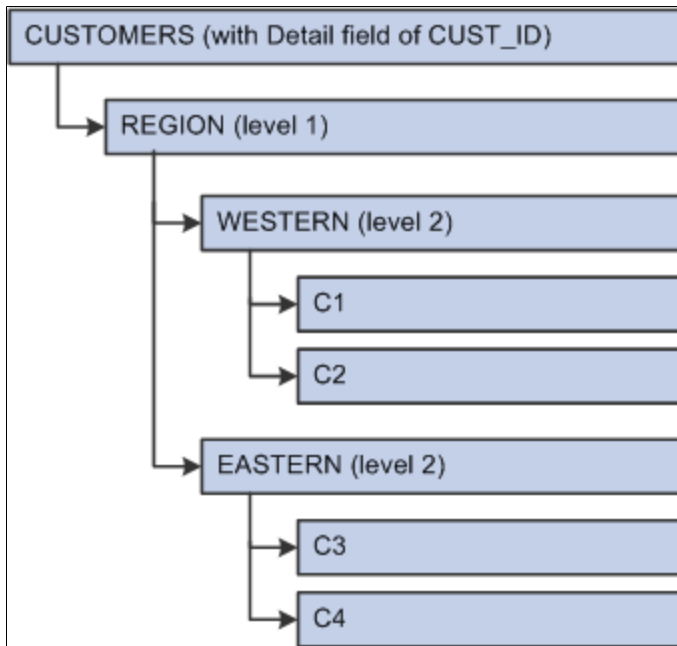
Customer	Product	Channel	Amount
C1	P1	CH1	1000
C1	P1	CH1	1000
C1	P2	CH2	2000
C2	P1	CH1	1000
C2	P2	CH2	2000
C3	P2	CH2	2000
C4	P1	CH1	1000
C4	P2	CH2	2000
C4	P2	CH2	2000

Assume you have the following customer location tree structure.

Source 2:

Image: Sample customer tree

This diagram illustrates an example of a customer tree.



The system rolls up the source data (customer IDs) to the next level up (regions) summarizes it, and reduces the number of rows. In the target table, the customer ID is now the tree node at the desired roll-up level.

This is how the target looks:

<i>Customer</i>	<i>Product</i>	<i>Channel</i>	<i>Amount</i>
WESTERN	P1	CH1	3000
WESTERN	P2	CH2	4000
EASTERN	P1	CH1	1000
EASTERN	P2	CH2	6000

Prerequisites

Before setting up your Data Manager rules, you must:

- Complete set up of your metadata (tablemaps, datamaps, and constraints).

See [Understanding Metadata](#)

- Define any value objects on the Value Object page.

See [Setting Up Value Objects](#)

Data Manager Rules - Define Rule Page

Use the Data Manager Rules - Define Rule page (MD_RULE_TBL1) to define a data manager rule, including the method you intend to use.

Navigation

EPM Foundation, Data Enrichment Tools, Data Manager, Rules, Define Rule

Image: Define Rule page

This example illustrates the fields and controls on the Define Rule page. You can find definitions for the fields and controls later on this page.

The screenshot shows the 'Define Rule' page with the following fields and values:

- SetID: SHARE
- Rule ID: ABMLEDG
- SQL Object ID Prefix: MD\$_ABMLEDG_
- Effective Date: 01/01/1900
- Status: Active
- Description: ABM Ledger Data
- Method: Copy
- Notes: Copy ABM Ledger with no filter to PF_Jrnl with 6 dimensions.

Method

Select the method for the rule you are defining. Options are: *Arithmetic*, *Copy*, *GL Mapper*, *Prorata*, *Spread Even*, and *Tree Aggregation*.

See [Understanding Data Manager Rules and Methods](#).

Note: If the method is changed after the target fields have been populated, some fields might be reset to blank. This occurs if the current method has a source 2 and the new method does not. All source 2 fields that were mapped in the Data Manager Rules component are reset. The system displays a warning message.

For a method of *Arithmetic*, select the operation to use: *Addition (+)*, *Division (/)*, *Multiplication (*)*, and *Subtraction (-)*.

For a method of *Prorata*, the system prorates measures in source 1 across all common dimensions based on source 2 numbers.

For a method of *Spread Even*, the system spreads source 1 measures evenly across all common dimensions based on source 2 numbers.

For a method of *Tree Aggregation*, the system copies data elements from the source to the target based on the source 2 tree structures.

Save As

In correction mode, click the Save As button to access the Copy Rule page on which you can enter a new rule ID to which you want to copy the rule information. Enter an effective date as today's date. Click OK to save the changes and return to the Define Rule page.



Click the Compile button on this page or other pages in the component to build the Data Manager rule. Review the generated SQL on the Data Manager Rules - SQL page.

Data Manager Rules - Define Source 1 Page

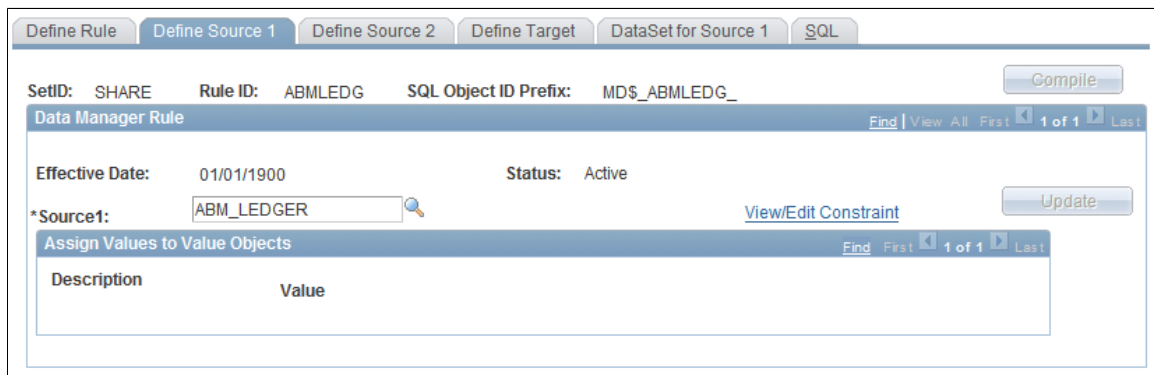
Use the Data Manager Rules - Define Source 1 page (MD_RULE_SCR1) to define source 1 information.

Navigation

EPM Foundation, Data Enrichment Tools, Data Manager, Rules, Define Source 1

Image: Define Source 1 page

This example illustrates the fields and controls on the Define Source 1 page.



Use the Data Manager Rules - Define Source 2 page (MD_RULE_SCR2) to define source 2 information.

Navigation

EPM Foundation, Data Enrichment Tools, Data Manager, Rules, Define Source 2

Image: Define Source 2 page

This example illustrates the fields and controls on the Define Source 2 page.

The screenshot shows the 'Define Source 2' page in the EPM Foundation interface. At the top, there are several tabs: 'Define Rule', 'Define Source 1', 'Define Source 2' (which is the active tab), 'Define Target', 'DataSet for Source 1', and 'SQL'. Below the tabs, there are several fields: 'SetID: SHARE', 'Rule ID: ABMLEDG', 'SQL Object ID Prefix: MD\$_ABMLEDG_', and 'Effective Date: 01/01/1900'. The 'Status' is 'Active'. There is a 'Source2:' field with a dropdown menu and a 'View/Edit Constraint' link. Below this is a table titled 'Assign Values to Value Objects' with columns 'Description' and 'Value'. The table is currently empty. There are also 'Compile' and 'Update' buttons.

Source 1

Select the constraint to tell the Data Manager what to assign.

The constraints you define can limit the row selection. You can also define constraints to have no criteria, in which case they pull all the data. Most rules have two sources, with the exception of the copy and GL Mapper methods, which only require one source. Source 1 contains the measure to be assigned (for example, the Activity-Based Management costs), while source 2 contains the basis for the assignment (for example, spreading those costs either evenly or prorating them over customers).

For the GL Mapper method, the source constraint must be *GLM_LEDGER_ALL*. This constraint must be built on the primary table of the LEDGER_VW tablemap.

Note: To review the primary table of the LEDGER_VW tablemap, click the View/Edit Constraint link to navigate to the Constraint page. On the Constraint page, click the View/Edit Datamap link.

For the tree aggregation method, a source 1 mapping field displays next to the Source 1 field. The Data Manager uses this field in conjunction with the tree node field to determine the join criteria in the SQL.

Source 2

Select the constraint to tell the Data Manager what basis to use.

For the tree aggregation method, select the tree you want to use for aggregation. When you select the tree, the Tree Viewer opens and enables you to drill down to the tree level you require. You can also review the tree node information. Click the Select button on the Tree Viewer page to select the tree level.

Note: You do not need to specify a source 2 for the copy or GL Mapper methods.

View/Edit Constraint

Click the link to view or edit the selected constraint. This option transfers you to the Constraint page for that constraint.

Update

Click the Update button to view an updated list of the source 1 or source 2 assignment values. You must be in correction mode to click this button.

Note: If you change a source constraint after the populating the target fields, the target fields that were mapped to source fields will not be lost so long as the datamap is the same.

Data Manager Rules - Define Target Page

Use the Data Manager Rules - Define Target page (MD_RULE_TBL3) to define the target for the rule and the target mapping.

Navigation

EPM Foundation, Data Enrichment Tools, Data Manager, Rules, Define Target

Image: Define Target page

This example illustrates the fields and controls on the Define Target page. You can find definitions for the fields and controls later on this page.

Target Description	Metadata Type	Rvrs Sign	Sum	From	Source Description
PF Batch ID	Dimension	<input type="checkbox"/>	<input type="checkbox"/>	Fixed Val	%Blank
Business Unit	Dimension	<input type="checkbox"/>	<input type="checkbox"/>	MetaValue	%BusinessUnit
Scenario ID	Dimension	<input type="checkbox"/>	<input type="checkbox"/>	MetaValue	%ScenarioId
Fiscal Year	Dimension	<input type="checkbox"/>	<input type="checkbox"/>	MetaValue	%FiscalYear
Accounting Period	Dimension	<input type="checkbox"/>	<input type="checkbox"/>	MetaValue	%AccountingPeriod
Account	Dimension	<input type="checkbox"/>	<input type="checkbox"/>	Source1	Account
Source	Dimension	<input type="checkbox"/>	<input type="checkbox"/>	Fixed Val	%Blank
Operating Unit	Dimension	<input type="checkbox"/>	<input type="checkbox"/>	Source1	Operating Unit
Department	Dimension	<input type="checkbox"/>	<input type="checkbox"/>	Source1	Department
Customer ID	Dimension	<input type="checkbox"/>	<input type="checkbox"/>	Fixed Val	%Blank
Product ID	Dimension	<input type="checkbox"/>	<input type="checkbox"/>	Source1	Product
Channel ID	Dimension	<input type="checkbox"/>	<input type="checkbox"/>	Fixed Val	%Blank
Dimension	Dimension	<input type="checkbox"/>	<input type="checkbox"/>	Fixed Val	%Blank
Project	Dimension	<input type="checkbox"/>	<input type="checkbox"/>	Source1	Project
Fund Code	Dimension	<input type="checkbox"/>	<input type="checkbox"/>	Fixed Val	%Blank

Target

Select the datamap to use as the target for the processed data.

The system populates the Target Object Field Map grid with target fields based on the datamap you select. The system handles the mapping of all the fields except for the department and product fields, which you can only map to the source 1 department or product, a fixed value, or a meta value.

For the GL Mapper method, the target is *GLSTG*.

Warning! For the prorata and spread even methods, each target field must map to a source 1 field, source 2 field, or value object. The system issues a warning message if any dimensions exist on the datamap that are not mapped to a target field.

Prorata Measure	For a method of <i>Prorata</i> , select the measure to use for the prorata calculation. Only source 2 fields that are defined as measures on the datamap display in the prompt list.
Measure	For a method of <i>Arithmetic</i> , select the measure to use. The values in the list box are based on the selected source 2 constraint.
Target Description and Metadata Type	The system populates these fields based on the datamap you select as the target. The target description comes from the datamap definition.
RvrsSign (reverse sign)	Select to reverse the sign of the amount when the data is copied to the target table. Check boxes are available in this column only for target fields that are defined as a measure on the target datamap.
Sum	For a method of <i>Copy</i> , select this check box to total the amounts when the source 1 data is copied to the target table. Check boxes are available in this column only for target fields that are defined as a measure on the target datamap.
From	<p>This field can be:</p> <p><i>Source 1</i>: Data is copied from the source 1 table to the target field.</p> <p><i>Fixed Value</i>: A constant value is inserted into the target field.</p> <p><i>Meta Value</i>: A meta value is inserted into the target field.</p> <p><i>Sys Var</i>: (system variable) A system variable is inserted into the target field.</p> <hr/> <p>Note: For the GL Mapper method, you can only specify the From field for the department and product ID.</p> <hr/>
Source Description	<p>If the value in the From field is <i>Source 1</i> or if the value is mapped from source 1, a list box of the fields that were defined in source 1 is available.</p> <p>If the value in the From field is <i>Source 2</i> or if the value is mapped from source 2, a list box of the fields that were defined in source 2 is available.</p> <p>If one of the value object types is specified for the value in the From field, the drop-down list box contains the objects that correspond to the specified value object class.</p>
Prorata	For a method of <i>Prorata</i> , select this check box for any measures you want to use for the proration. You must select at least one measure field for prorata.

Arithmetic

For a method of *Arithmetic*, select this check box for a field in which you want to use in the arithmetic operation that is specified on the Data Manager Rules - Define Rule page.

Target Definition for the Copy Method

When you are setting up the target field map for the copy method:

- If you select a target measure field that is mapped to a source 1 field for a sum operation, all other measures that are mapped to source 1 fields must also be selected for a sum operation.
- If you select the sum check box for any of the fields, you can map only key source 1 fields.
- If you do not select the sum check box for any of the fields, you can map any source 1 fields to any target fields.

This might result in duplicate data when you run the Data Manager engine. This is because you are not grouping the data by the key values.

- You can use expressions for your mappings.

Target Definition for the Prorata Method

When you are setting up the target field map for the prorata method:

- Only three mapped measures are allowed.
- The source 1 and source 2 datamaps must both have at least one key field that is marked as a dimension.
- All non-measure target fields can be mapped only to source 1 or source 2 fields that are marked as dimensions (keys) on the datamap.
- All measure target fields can only be mapped to source 1, fixed value, or meta value fields.
- At least one measure target field must be marked for prorata.
- If a target field is mapped to a source 1 field, it must be marked for prorata.

Target Definition for the Spread Even Method

When you are setting up the target field map for the spread even method:

- Only three mapped measures are allowed.
- The source 1 and source 2 datamaps must both have at least one key field that is marked as a dimension.
- All non-measure target fields can only be mapped to source 1 or source 2 fields that are marked as dimensions (keys) on the datamap.
- All measure target fields can only be mapped to source 1, fixed value, or meta value fields.

Data Manager uses all the source 1 mappings for the spread even calculation.

- At least one measure target field must be mapped to a source 1 field.

Target Definition for Tree Aggregation Method

When you are setting up the target field map for the tree aggregation method:

- Each target field must have a source 1 field, tree node, or value object to which it is mapped.
- Only one target field can be mapped to the source 2 tree node.

This tree node field is used in conjunction with the source 1 mapping field for join criteria in the SQL.

- You must have at least one non-measure target field mapped to source 1.
- If the target field is mapped to source 2, you can select from all the fields for that source.

If a target field is mapped to source 2, the only selection available is *Tree Node*.

Defining Foreign Currency

You must always provide values for the Foreign Currency Code in the target definition. Even if you are using United States dollars as your base currency, you must populate the Foreign Currency Code in the

target definition. For example, if you use United States dollars as your base currency, you must map *Source1* to *Base Currency Code* (as the following graphic demonstrates).

Image: Define Target page - Foreign Currency Code field

This example illustrates the fields and controls on the Define Target page - Foreign Currency Code field.

Define Rule | Define Source 1 | Define Source 2 | **Define Target** | DataSet for Source 1 | SQL

SetID: SHARE Rule ID: PF_REV1 SQL Object ID Prefix: MDS_PF_REV1_

Data Manager Rule First 1 of 1 Last

Effective Date: 01/01/1900 Status: Active

*Target: PF_JRNL

Target Object Field Map Customize Find First 1-24 of 24 Last

Target Description	Metadata Type	Rvrs Sign	Sum	From	Source Description
Product ID	Dimension	<input type="checkbox"/>	<input type="checkbox"/>	Source1	Product ID
Channel ID	Dimension	<input type="checkbox"/>	<input type="checkbox"/>	Source1	Channel ID
Dimension	Dimension	<input type="checkbox"/>	<input type="checkbox"/>	Fixed Val	%Blank
Project	Attribute	<input type="checkbox"/>	<input type="checkbox"/>	Fixed Val	%Blank
Fund Code	Attribute	<input type="checkbox"/>	<input type="checkbox"/>	Fixed Val	%Blank
Foreign Currency Code	Dimension	<input type="checkbox"/>	<input type="checkbox"/>	Source1	Base Currency Code
Last Edit Seq Number	Attribute	<input type="checkbox"/>	<input type="checkbox"/>	Fixed Val	%Zero
Currency Code	Attribute	<input type="checkbox"/>	<input type="checkbox"/>	Fixed Val	%CURRENCY_CD
Process Instance	Attribute	<input type="checkbox"/>	<input type="checkbox"/>	MetaValue	%ProcessInstance
Monetary Amount	Measure	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Source1	Monetary Amount
Foreign Amount	Measure	<input type="checkbox"/>	<input type="checkbox"/>	Fixed Val	%Zero
Statistic Amount	Measure	<input type="checkbox"/>	<input type="checkbox"/>	Fixed Val	%Zero
Load Date and Time	Attribute	<input type="checkbox"/>	<input type="checkbox"/>	Sys Var	%CurrentDateTimeln
Error Flag	Attribute	<input type="checkbox"/>	<input type="checkbox"/>	Fixed Val	%No

If you do not populate the Foreign Currency Code in the target definition, unexpected results may occur when the PF_POST application engine runs.

Related Links

[Setting Up Value Objects](#)

Data Manager Rules - DataSet for Source 1 Page

Use the Data Manager Rules - DataSet for Source 1 page (MD_DATASET1_DEFN) to review and define the source 1 fields and measures that are mapped to target fields.

Navigation

EPM Foundation, Data Enrichment Tools, Data Manager, Rules, DataSet for Source 1

Image: DataSet for Source 1 page

This example illustrates the fields and controls on the DataSet for Source 1 page. You can find definitions for the fields and controls later on this page.

Define Rule | Define Source 1 | Define Source 2 | Define Target | **DataSet for Source 1** | SQL

SetID: SHARE Rule ID: ABMLEDG

Data Manager Rule Find | View All | First | 1 of 1 | Last

Effective Date: 01/01/1900 Status: Active SQL Object ID Prefix: PF\$_DS_2566

Dataset Columns Customize Find First 1-7 of 7 Last		
Description	Select	Aggregate Type
Account	<input checked="" type="checkbox"/>	
Operating Unit	<input checked="" type="checkbox"/>	
Department	<input checked="" type="checkbox"/>	
Product	<input checked="" type="checkbox"/>	
Project	<input checked="" type="checkbox"/>	
Currency Code	<input checked="" type="checkbox"/>	
Posted Total Amount	<input checked="" type="checkbox"/>	

Select

Select this check box to include the dimension or measure.

Note: The DataSet for Source 2 page does not display for the copy or GL Mapper method.

Data Manager Rules - DataSet for Common Dimensions Page

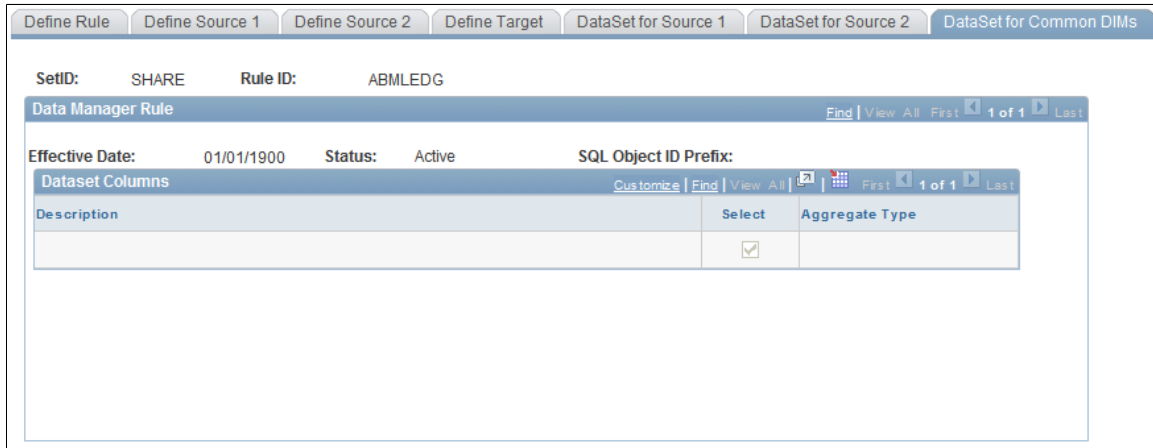
Use the Data Manager Rules - DataSet for Common Dimensions page (MD_DATASET3_DEFN) to review the mapped common dimensions and prorata measure.

Navigation

EPM Foundation, Data Enrichment Tools, Data Manager, Rules, DataSet for Common Dimensions

Image: DataSet for Common DIMs page

This example illustrates the fields and controls on the DataSet for Common DIMs page.



This page shows the common dimensions that exist between source 1 and source 2. A dimension is common if it has a matching description in the source 1 and source 2 datamap fields.

For the prorata method, the page also shows the prorata measure marked as *Sum*.

Note: Common dimensions are very important. They determine the join criteria for the method. The criteria are based on the datamap, so always review this page to ensure the results are calculating as you expect.

Data Manager Rules - SQL Page

Use the Data Manager Rules - SQL page (MD_RULE_TBL4) to review the generated SQL for the Data Manager rule.

Navigation

EPM Foundation, Data Enrichment Tools, Data Manager, Rules, SQL

Image: SQL page

This example illustrates the fields and controls on the SQL page.

The screenshot shows the 'SQL' tab of a Data Manager rule configuration. The rule is named 'SHARE' with ID 'ABMLEDG'. It is active and uses the 'Copy' method. The source is 'ABM_LEDGER' and the target is 'PF_JRNL'. The SQL statement is as follows:

```
Insert into PS_PF_JRNL_T%Bind(EPM_CORE_AET.TABLE_APPEND,NOQUOTES) (%LIST
(FIELD_LIST,PF_JRNL_T)Select '', A1.ACCOUNT, '', A1.OPERATING_UNIT, A1.DEPTID, '',
A1.PRODUCT, '', A1.PROJECT_ID, '', 0, A1.CURRENCY_CD, %Bind
(EPM_CORE_AET.PROCESS_INSTANCE, NOQUOTES), 0, A1.POSTED_TOTAL_AMT, 0, %
CURRENTDATETIMEIN, 'N', %Bind(EPM_CORE_AET.BUSINESS_UNIT), %Bind
(EPM_CORE_AET.PF_SCENARIO_ID), %Bind(EPM_CORE_AET.FISCAL_YEAR,
NOQUOTES) %Bind(EPM_CORE_AET.ACCOUNTING_PERIOD, NOQUOTES) From %SQL
```

Once you have compiled a rule using the Compile button, review the generated SQL statement.

All %Bind and %SQL objects are resolved at run-time. There is only one SQL statement that is generated for each method.

A copy of this SQL is stored in the SQL repository with the SQL object ID prefix that is shown in this component. A sequential number for each SQL statement completes the SQL object name for the repository.

Defining Data Manager Rule Sets

You must define a Data Manager rule set for any Data Manager rules that you want to process. You can also use rule sets to combine multiple rules in one run of the Data Manager engine. Rules in a rule set are run in the order shown on the Data Manager Rule Sets page. The first rule must complete successfully before the second rule starts. The system places the data from the first rule in a temporary table (assigned by the engine) that will be the source of the second rule, and so on.

Note: A rule must be in a rule set, even if only one rule is processed.

This topic discusses how to set up Data Manager rule sets.

Pages Used to Define Data Manager Rule Sets

Page Name	Definition Name	Navigation	Usage
Assignment Rule Sets	MD_RULESET_TBL1	EPM Foundation, Data Enrichment Tools, Data Manager, Rule Set, Assignment Rule Sets	Set up Data Manager rule sets by defining the rules that are to be included in the rule set.
Assignment Rule Sets - Notes	MD_RULESET_TBL2	EPM Foundation, Data Enrichment Tools, Data Manager, Rule Set, Assignment Rule Sets	Describe the purpose of this rule set.

Assignment Rule Sets Page

Use the Assignment Rule Sets page (MD_RULESET_TBL1) to set up Data Manager rule sets by defining the rules that are to be included in the rule set.

Navigation

EPM Foundation, Data Enrichment Tools, Data Manager, Rule Set, Assignment Rule Sets

Image: Assignment Rule Sets page

This example illustrates the fields and controls on the Assignment Rule Sets page.

The screenshot shows the 'Assignment Rule Sets' page with the following details:

- SetID:** SHARE
- Rule Set:** ABMLEDGRS
- Data Manager Rule Set:**
 - *Effective Date:** 01/01/2000
 - *Status:** Active
 - *Description:** ABM Ledger
- Rules in Rule Set:**

*Sequence	*Rule ID
1	ABMLEDG
2	ABMPROD

RuleSet

A unique identifier for this rule set definition.

Sequence

Enter a number, such as 100, for the first rule ID in the rule set. The next rule ID to run would have a sequence of 200.

The actual sequence number is not important; it represents the order in which you want to process rules. The sequence must be unique.

Rule ID

Select the Data Manager rule to include in the rule set. You create the rules using the Data Manager Rules component.

Click the Notes tab to enter a more detailed description of this rule set.

Renaming Data Manager Rules

This topic discusses how to rename a Data Manager rule.

Page Used to Rename Data Manager Rules

Page Name	Definition Name	Navigation	Usage
Rename Data Manager Rule	PF_MD_RULE_RENAME	EPM Foundation, Data Enrichment Tools, Data Manager, Rename Rules, Rename Data Manager Rule	Rename existing rules for modification.

Rename Data Manager Rule Page

Use the Rename Data Manager Rule page (PF_MD_RULE_RENAME) to rename existing rules for modification.

Navigation

EPM Foundation, Data Enrichment Tools, Data Manager, Rename Rules, Rename Data Manager Rule

Image: Rename Data Manager Rule page

This example illustrates the fields and controls on the Rename Data Manager Rule page. You can find definitions for the fields and controls later on this page.

SetID and Rule ID

Select the SetID and rule ID for the rule you want to rename.

Search

Click the Search button. The search populates the grid with the names of the Data Manager rule sets that is affected by this rename.

Rename

To proceed with the renaming process, click the Rename button and enter a new name for the rule. Every instance in which the original rule name is used is changed to the new name.

Associating Data Manager Rule Sets With a Job

The next step in the Data Manager process is associating the rule set with a Data Manager job. Each unique run of the Data Manager engine is given a job ID. You must establish an association to the rules and the actual job that will run them on the Job Association page. When the jobstream that contains the

defined Data Manager job ID runs, it uses the setup on the Job Association page to find the Data Manager rule set to execute.

Note: Each time that you run an EPM engine, it must have a unique job ID. These jobs must then be run in a jobstream.

This topic discusses how to define rule sets to be run for a given job.

Pages Used to Associate Data Manager Rule Sets With a Job

<i>Page Name</i>	<i>Definition Name</i>	<i>Navigation</i>	<i>Usage</i>
Job Association	MD_JOB_RULESET_TBL	EPM Foundation, Data Enrichment Tools, Data Manager, Job Association, Job Association	Define rule sets to be run for a given job ID. The order in which these rule sets are run is also defined.
Job Association - Notes	MD_JOB_RULESET_TB2	EPM Foundation, Data Enrichment Tools, Data Manager, Job Association, Notes	Describe the job association.

Prerequisites

Before you can associate rule sets with job, you must:

- Set up the Data Manager job ID.
- Set up a jobstream for the job ID.

See [Understanding Jobstreams](#).

Job Association Page

Use the Job Association page (MD_JOB_RULESET_TBL) to define rule sets to be run for a given job ID. The order in which these rule sets are run is also defined.

Navigation

EPM Foundation, Data Enrichment Tools, Data Manager, Job Association, Job Association

Image: Job Association page

This example illustrates the fields and controls on the Job Association page. You can find definitions for the fields and controls later on this page.

Job ID

Select the job ID. The job ID was created in the job metadata and represents a unique instance of a Data Manager engine in a jobstream.

Ledger ID

Select a ledger ID. This is required for the Profit Manager.

See [Processing and Posting Journals](#).

Sequence

Enter a number, such as 100, for the first rule set in the job association. The next rule set to run would have a sequence of 200. The actual sequence number is not important; it represents the order in which you want to process rule sets. The sequence must be unique.

Rule Set

Select the rule set or rule sets to include in the job. You create rule sets on the Data Manger Rule Set - Assignment Rule Sets page.

Click the Notes tab to enter a more detailed description of this job association.

Running the Data Manager Engine

As a last step, you run the Data Manager engine in a jobstream. As you run the jobstream:

- Use the Process Monitor to verify the status of your job.
- Review the results by querying the target table for the Data Manager rules that you have run.

- Review any error messages using the Error Messages component.

Related Links

[Running Jobstreams](#)

[Viewing Engine Messages](#)

Defining Allocation Manager Rules

This topic provides an overview of allocation rules setup, fixed source and fixed basis options in allocations, lists prerequisites, and discusses how to:

- Determine the list of dimensions to use.
- Create a list of sources.
- Define an Allocation Manager rule.
- Define the allocation rule source.
- Define the allocation rule basis.
- Review common dimensions.
- Define the target.
- Assign residual amounts.
- Specify offset target information.
- Define allocation template.

Pages Used to Define Allocation Manager Rules

<i>Page Name</i>	<i>Definition Name</i>	<i>Navigation</i>	<i>Usage</i>
Fixed Dimension Setup	PF_AL_DIM_TBL	EPM Foundation, Data Enrichment Tools, Allocation Manager, Fixed Basis Dimensions, Fixed Dimension Setup	Determine the list of dimensions that you can use with fixed source and basis.
Allocation DataMap Setup	PF_AL_DM_TBL	EPM Foundation, Data Enrichment Tools, Allocation Manager, DataMap Setup, Allocation DataMap Setup	Create a list of sources for the source, basis, or target and specify dimensions for your datamap. The purpose of this page is to reduce the of number datamaps from which to choose. In addition, you can give the datamaps more intuitive names. Add as many datamaps as necessary.

Page Name	Definition Name	Navigation	Usage
Select Dimensions	PF_AL_SELECT_DIM	Click the Select Dimensions button on the Allocation DataMap Setup page	Select the dimension metadata type.
Rules - Method	PF_AL_METHOD_DEFN	EPM Foundation, Data Enrichment Tools, Allocation Manager, Rules, Method	Define an Allocation Manager rule and select the method.
Rules - Copy Rule	PF_AL_RULE_COPY	Click on the Save As button on the Allocation Manager Rules - Method page.	Copy Allocation Manager rules.
Rules - Source	PF_AL_SOURCE_DEFN	EPM Foundation, Data Enrichment Tools, Allocation Manager, Rules, Source	Define the allocation rule sources.
Fixed Source Dimension Setup	PF_AL_USE_FSRC_DIM	Click on the Choose Fixed Source Dimension link on the Allocation Manager Rules - Source page.	Select which dimensions for a fixed source.
Rules - Basis	PF_AL_BASIS_DEFN	EPM Foundation, Data Enrichment Tools, Allocation Manager, Rules, Basis	Define the allocation rule basis.
Fixed Basis Dimension Setup	PF_AL_USE_FIX_DIM	Click on the Choose Fixed Basis Dimension link on the Allocation Manager Rules - Basis page.	Select which dimensions for a fixed basis.
Rules - Source to Basis Relationship	PF_AL_COM_DIM_DEFN	EPM Foundation, Data Enrichment Tools, Allocation Manager, Rules, Source to Basis Relationship	Review all common dimensions between the source and the basis.
Rules - Target	PF_AL_TGT_DEFN	EPM Foundation, Data Enrichment Tools, Allocation Manager, Rules, Target	Define the target, including the allocation mappings. The target is the destination to which the amounts that are defined by the source and basis are allocated.
Allocation Manager Rules - Residual	PF_AL_RESID_DEFN	EPM Foundation, Data Enrichment Tools, Allocation Manager, Rules, Residual	Assign residual amounts after the source amount has been allocated to dimensions in the target datamap. You can specify the residual account to use and other dimensions as applicable.
Rules - Offset	PF_AL_OFFSET_DEFN	EPM Foundation, Data Enrichment Tools, Allocation Manager, Rules, Offset	Specify offset target information based on dimensions in the target datamap.

Page Name	Definition Name	Navigation	Usage
Allocation Template	PF_AL_TMPLT_DEFN	EPM Foundation, Data Enrichment Tools, Allocation Manager, Allocation Template	Predefine a set of source, basis, and target datamaps that you can later associate with any allocation rule.

Understanding Allocation Rule Setup

The following list represents required and optional tasks necessary to set up allocation rules.

1. Define datamaps for your source, basis, and target.

This step is a required prerequisite.

Note: PeopleSoft provides the *Datamap Wizard* to greatly simplify the creation of datamaps. The Datamap Wizard can be found using either of these navigation paths:
 EPM Foundation, Foundation Metadata, Metadata Wizards, Datamap Wizard
 EPM Foundation, Data Enrichment Tools, Allocation Manager, Datamap Wizard

See [Datamap Wizard Page](#).

2. Select fixed dimensions to use with a fixed source or basis.
 This step is optional if you are not using a fixed source or basis.
3. Select DataMaps for the allocation source, basis, and target.
 This step is required.
4. Define an allocation rule and select the calculation method.
 This step is required.
5. Define the allocation rule source.
 This step is required.
6. Define the allocation rule basis.
 This step is required.
7. Review all common dimensions between the source and the basis.
 This step is required.
8. Define the allocation rule target, including the allocation mappings.
 This step is required.
9. Assign post-allocation residual amounts.
 This step is optional.
10. Specify offset target information.

This step is optional.

With the exception of the first list item above, you use the Allocation Manager Rules component to set up all remaining tasks.

Note: You must set up the DataMaps and constraints you want to use in the allocation prior to creating an allocation rule.

See [Defining Allocation Manager Rules](#), [Setting Up Datamaps](#).

Streamlining Allocation Rule Setup with the Allocation Template

PeopleSoft EPM provides an *allocation template* that enables you to predefine a set of source, basis, and target DataMaps that you can later associate with any allocation rule, thereby saving time and ensuring DataMap consistency among various users setting up allocation rules.

When you define an allocation template you can associate it with an allocation rule using the *Template* field on the Method page, which is a part of the Allocation Manager Rules component. Once you select a template on the Method page, the appropriate source, basis, and target DataMaps automatically populate the Source, Basis, and Target pages.

See [Allocation Template Page](#).

Understanding Allocation Manager Fixed Source and Fixed Basis Options

Allocation Manager gives you the flexibility of adding fixed source and basis values in an allocation rule without referencing a database table. For example, you may only have a few dimension values that you want to use in a rule, like a source amount of \$100,000 that you want to allocate to the accounts in the Basis table. Or, you may want to perform an arithmetic allocation and multiply all values by a fixed rate of 0.25. In these cases, it is much simpler to input the value in the rule, without having to create a database table.

You use datamap metadata to define the tables that contain the source, basis, and target for your allocation in the Allocation Manager rule. However, in some cases you might not have source or basis data stored in tables in your database. To accommodate this situation, we deliver a set of metadata (the FIX_BASIS datamap, filters and constraints PF_AL_FIX_SOURCE , PF_AL_FIX_BASIS) that point to the same fixed source and basis table (PS_FP_AL_FIX_BASIS). You can tailor this table by adding your organization's dimensions to it on the Fixed Dimensions page. Use the Fixed Dimension page to list the dimensions and assign prompt tables and trees and select values for each dimension.

Note: Fixed source and basis is used with the Allocation Manager only. It is available with all allocation methods except period-based allocation.

Note: Before you create an Allocation Manager rule to associate with a SetID, create a PF_AL_FIX_BASIS and PF_AL_FIX_SOURCE filter and constraint for that SetID. You can model the setup for the PF_AL_FIX_BASIS and PF_AL_FIX_SOURCE filter and constraint using the sample data under the SHARE SetID.

Calculating Offset Target Information (Optional)

Allocation Manager enables you to calculate offsets for the target balance. Offset accounts aid in balancing source transaction and budget entries. You can use offsets to balance ledgers. You can create either single or multiple offsets. A single offset allows a single target entry for the entire offset and multiple offsets create identical set of target entries with the target measure values negated. Also, a separate offset entry can be created for each of the residual amounts. You must ensure at least one key value is different from the residual entry in order to avoid collisions. The same is true if a single offset entry is selected.

The following is an example of an arithmetic method offset allocation account:

Source		Basis		Target	
Department	Amount	Department	Unit	Department	Amount
CORP	1000	DIV1	0.1	DIV1	100
		DIV2	0.1	DIV2	100
		DIV3	0.2	DIV3	200
				OFST (Offset Account)	=400

In this example, the offset account is equal to 400, which is the same total for Divisions (DIV) 1, 2, and 3.

Prerequisites

Before setting up your Allocation Manager rules, you must:

- Complete the setup of your metadata (tablemaps, datamaps, and constraints).

See [Understanding Metadata](#)

- Define any value objects on the Value Object page.

See [Setting Up Value Objects](#)

Fixed Dimension Setup Page

Use the Fixed Dimension Setup page (PF_AL_DIM_TBL) to determine the list of dimensions that you can use with fixed source and basis.

Navigation

EPM Foundation, Data Enrichment Tools, Allocation Manager, Fixed Basis Dimensions, Fixed Dimension Setup

Image: Fixed Dimension Setup page

This example illustrates the fields and controls on the Fixed Dimension Setup page. You can find definitions for the fields and controls later on this page.

Dimension Name	Look Up Table	Look Up Tree	Description
Account	GL_ACCOUNT_TBL	ACT_TREE	Account
Base Currency	CURRENCY_CD_TE		Base Currency
Channel ID	CHANNEL_D00		Channel ID
Currency Code	CURRENCY_CD_TE		Currency Code
Customer ID	CUSTOMER_D00		Customer ID
Department	DEPARTMENT_TBL	FNI_DEPARTMEN	Department
Product ID	PRODUCT_D00		Product ID
Operating Unit	OPER_UNIT_D00		Operating Unit
Ledger Business Unit	BUS_UNIT_TBL_PF		Ledger Business Unit

Dimension Name

Each dimension on this page is based on the FIX_BASIS datamap. This datamap is delivered with your system and consists of a generic table into which you can enter your organization-specific information. You can define a maximum of 12 dimensions for this datamap. PeopleSoft delivers this datamap with several dimensions; however, you can create more, if necessary.

Lookup Table and Lookup Tree

For each dimension, assign either prompt tables, trees, or both to select values.

Refresh

Changes to the FIX_BASIS datamap do not automatically display on this page. Click the Refresh button to update the page with any additions or deletions to the datamap.

Note: You can only change dimensions. Do not change attributes or the measure.

Allocation DataMap Setup Page

Use the Allocation DataMap Setup page (PF_AL_DM_TBL) to create a list of sources for the source, basis, or target and specify dimensions for your datamap. The purpose of this page is to reduce the of number datamaps from which to choose. In addition, you can give the datamaps more intuitive names. Add as many datamaps as necessary.

Navigation

EPM Foundation, Data Enrichment Tools, Allocation Manager, DataMap Setup, Allocation DataMap Setup

Image: Allocation DataMap Setup page

This example illustrates the fields and controls on the Allocation DataMap Setup page. You can find definitions for the fields and controls later on this page.

Allocation DataMap Setup				Customize Find View All First 1-15 of 15 Last	
*DataMap Code	Target	Description	View Dimensions		
ABM_PRDCST	<input type="checkbox"/>	Product Costs			
ALLOC_BAS	<input type="checkbox"/>	Demo Basis For Allocations			
ALLOC_SRC	<input type="checkbox"/>	Demo Source For Allocations			
ALLOC_TGT	<input checked="" type="checkbox"/>	Demo Target For Allocations			
BP_LED_BUD	<input checked="" type="checkbox"/>	BP Standard Budgeting Ledger			
FIX_BASIS	<input type="checkbox"/>	Fixed Basis			
GCALLOBAS1	<input type="checkbox"/>	GC Basis 1			
GCALLOSRC1	<input type="checkbox"/>	GC Allocation Source 1			
GCCLEDMGT	<input type="checkbox"/>	Consolidation Ledger			
GCJRNLMGT	<input checked="" type="checkbox"/>	Consolidation Journal			
GCMGTASRC	<input type="checkbox"/>	GC Allocation Source			
GCMGTJRNL	<input checked="" type="checkbox"/>	GC allocations jrnl target			
PF_JRNL	<input checked="" type="checkbox"/>	PF Journal Target			
REVENUE	<input checked="" type="checkbox"/>	PF Revenue Source			
	<input type="checkbox"/>				

DataMap Code

This code identifies the datamap.

Note: You must create the datamaps before using the Datamap component.

Target

If selected, the target indicates that the datamap can be used as a target. This check box is only an indicator. This is defined as part of the datamap definition.



Click the Select Dimensions button to access the Select Dimensions page and review the dimension fields in your datamaps. When you are done, click OK to return to the Allocation DataMap Setup page.

Use the Select Dimensions page (PF_AL_SELECT_DIM) to select the dimension metadata type.

Navigation

Click the Select Dimensions button on the Allocation DataMap Setup page

Image: Select Dimensions page

This example illustrates the fields and controls on the Select Dimensions page.

DataMap Code: ALLOC_SRC

Description: Demo Source For Allocations [View/Edit Datamap](#)

Dimensions	
Description	Metadata Type
Business Unit	Dimension
Fiscal Year	Dimension
Accounting Period	Dimension
Scenario ID	Dimension
Operating Unit	Dimension
Account	Dimension
Department	Dimension
Customer ID	Dimension
Product ID	Dimension
Channel ID	Dimension
Posted Base Currency Amount	Measure

OK Cancel

Related Links

[Setting Up Datamaps](#)

Rules - Method Page

Use the Rules - Method page (PF_AL_METHOD_DEFN) to define an Allocation Manager rule and select the method.

Navigation

EPM Foundation, Data Enrichment Tools, Allocation Manager, Rules, Method

Image: Method page

This example illustrates the fields and controls on the Method page.

The screenshot displays the 'Method' page for an Allocation Rule. At the top, there are tabs for 'Method', 'Source', 'Basis', 'Source to Basis Relationship', 'Target', 'Residual', and 'Offset'. The 'Method' tab is selected. Below the tabs, the 'SetID' is 'SHARE' and the 'Rule ID' is 'AL_PRO'. The page title is 'Allocation Rule' with search and navigation options. The main form includes:

- *Effective Date: 01/01/1900
- *Status: Active (dropdown menu)
- *Description: Demo Prorata Method
- *Method: Prorata (dropdown menu) with a 'Method Help' link.
- Period Based Allocation:
- Allocate rounding differences:
- Template: (empty field with search icon)
- Notes: Allocation Manager Rule that demonstrates the Prorata method. The source is Demo Source, the basis is Demo Basis, and the Target is Demo Target.

 At the bottom, the 'Compile Information' section shows the 'SQL Object ID Prefix' as 'PF\$_AL_AL_PRO_SHARE' and a 'Compile' button.

Method

Select the calculation method for the allocation. The method determines the calculation options, such as how to determine the source amounts going to the targets.

Method options are: *Arithmetic*, *Copy*, *Prorata*, and *Spread Even*.

For the *Arithmetic* method, the Arithmetic Operation field becomes available for input.

For the *Prorata* method, the system prorates measures in source 1 across all common dimensions based on source 2 numbers.

For the *Spread Even* method, the system spreads source 1 measures evenly across all common dimensions based on source 2 numbers.

Method Help

Click this link to view additional information about the chosen calculation type.

For example, if you select the prorata method and click this link, the following information appears:

Divides the source amount proportionately among the targets based on basis measures.

The prorata method performs the following calculation:

$Source\ Measure * Basis\ Measure / Basis\ Total$
 $Source\ Measure = Sum(measures\ in\ source\ grouped\ by\ common\ dimensions\ and\ source\ mapped\ dimensions)$

$Basis\ Measure = Sum(measures\ in\ basis\ grouped\ by\ common\ dimensions\ and\ basis\ mapped\ dimensions)$

$Basis\ Total = Sum(measures\ in\ basis\ grouped\ by\ common\ dimensions\ only)$

Arithmetic Operation

For a method of *Arithmetic*, select the operation to use from *Addition (+)*, *Division (/)*, *Multiplication (*)*, and *Subtraction (-)*.

Period Based Allocation

Select this check box to specify period-based allocation.

This check box is only available when you select the *Prorata* and *Spread Even* methods.

Period-based allocation enables you to allocate from one period to one or more additional periods. For example, you may allocate quarterly source data into monthly target data based upon monthly basis data. To do this, a scenario ID must be specified for the basis on the Basis page. The scenario ID determines from which calendar to select the basis data. The calendar that is identified by the scenario that you selected on the Basis page determines which periods from the basis are used to perform the prorata or spread even processes.

Note: The source period must include one or more periods from the basis calendar. For example, if your source period is quarterly, your basis period should be monthly, weekly or daily.

Note: For period-based allocations, after the allocation is complete, the Merge application engine will move data directly from the temporary target table to the permanent target table.

Allocate Rounding Differences

Select this check box to enable rounding.

If selected the amount is rounded to 3 decimal points, or 83.333 per period.

In some allocation rules, the totals of the source amounts may not match the total amounts allocated. This may be due to the rule definition itself, or rounding errors introduced by the allocation. This feature will add the difference generated by the rounding to one row so that the total amount allocated is the same as the source amount.

You may also deselect this check box to disable rounding and increase system performance. If rounding is disabled, the system rounds all amounts that are prorated or distributed to the closest whole number for each period. This takes considerably less

time than it would using rounding. For example, it is more time consuming to spread an amount of 1,0000 evenly across twelve periods because the results equals 83.333333333333 per period.

Template (optional)

Select a template if you want to associate a set of predefined source, basis, and target DataMaps with the allocation.

Selecting a template will automatically populate the Source, Basis, and Target pages with the DataMaps specified in the template.

See [Allocation Template Page](#).

Copy Rule

Click to access the Copy Rule page, on which you can enter a new rule ID and copy the rule information. Enter an effective date. The system default is today's date. Click OK to save the changes and return to the Method page.

Compile

Click to individually compile the Allocation Manager rule. You can also use Mass Compile.

Note: You cannot use Mass Compile for allocation rules that are period-based or have the inter-business unit option selected. The system will skip these . You must compile these rules directly from the Allocation Manager Rules component.

Rules - Source Page

Use the Rules - Source page (PF_AL_SOURCE_DEFN) to define the allocation rule sources.

Navigation

EPM Foundation, Data Enrichment Tools, Allocation Manager, Rules, Source

Image: Source page

This example illustrates the fields and controls on the Source page.

Method Source Basis Source to Basis Relationship Target Residual Offset

SetID: SHARE Rule ID: AL_PRO

Allocation Rule Find | View All First 1 of 1 Last

Effective Date: 01/01/1900 Status: Active Refresh

Description: Demo Prorata Method
 Use Fixed Source

*Source: ALLOC_SRC Demo Source For Allocations *Source Adjustment: 1.0000

*Constraint: ALLOC_SRC_ALL Demo Source For Allocations Create New Constraint

* Required Field

Image: Source page with fixed source option and Tree node button

This example illustrates the fields and controls on the Source page with fixed source option and Tree node button. You can find definitions for the fields and controls later on this page.

Method Source Basis Source to Basis Relationship Target Residual Offset

SetID: SHARE Rule ID: AL_FIXED

Allocation Rule Find | View All First 1 of 1 Last

Effective Date: 01/01/1900 Status: Active

Description: Demo Fixed Source and Basis
 Use Fixed Source Choose Fixed Source Dimension

*Source Adjustment: 1.0000

Value	Tree	Account	Tree	Department
1000000.00000000		<input type="text"/>		<input type="text"/>

* Required Field

Refresh

In correction mode, click Refresh to update the page with any additions or deletions to the DataMaps.

Use Fixed Source

Select this check box to enable the use of all available fixed dimensions.

See 'Using a Fixed Source' below.

Source Adjustment

Enter a source adjustment against the source datamap.

This field defaults to 1, but you can input a larger number for a multiplying factor against source amounts.

You can also input a negative number adjustment so that the source amounts are multiplied by a negative number.

Selecting a Source

If you are not using fixed source, you must specify the source to use in the allocation.

Source	Select a source from the available options. The options are based on the datamaps that are listed on the Allocation DataMap Setup page
Constraint	Select a constraint for the source DataMap. If the criteria that you selected requires meta-value objects, the Assign Values to Value Objects group box appears and you can assign values to the constraint at this time.
Demo Source for Allocations	For the Source field, click the link to access the General Properties (DataMap) page and view the selected DataMap. For the Constraint field, click the link to access the Constraint page and view the selected constraint.
Create New Constraint	Click to access to the Constraint setup page and create a new constraint.

Using a Fixed Source

Choose Fixed Source Dimension	For a fixed source, click this link to access the Fixed Source Dimension Setup page and select your dimensions. Click OK to return to the Source page.
Fixed Source Value	If you select a fixed source, populate the value field for each dimension that you select You must assign a value for each fixed dimension. Where applicable, click the Tree Node button alongside the Fixed Source Value field to access the Tree Viewer page. Navigate through the tree by collapsing and expanding the tree. When you have located the tree node you want to use, click the Select button to select this node and return to the Source page.



Rules - Basis Page

Use the Rules - Basis page (PF_AL_BASIS_DEFN) to define the allocation rule basis.

Navigation

EPM Foundation, Data Enrichment Tools, Allocation Manager, Rules, Basis

Image: Basis page

This example illustrates the fields and controls on the Basis page.

Method Source **Basis** Source to Basis Relationship Target Residual Offset

SetID: SHARE Rule ID: AL_PRO

Allocation Rule Find | View All First 1 of 1 Last

Effective Date: 01/01/1900 Status: Active Refresh

Description: Demo Prorata Method

Use Fixed Basis

*Basis: [Demo Basis For Allocations](#) *Basis Adjustment:

*Constraint:

*Arithmetic Measure:

Image: Basis page with fixed basis option

This example illustrates the fields and controls on the Basis page with fixed basis option. You can find definitions for the fields and controls later on this page.

Method Source **Basis** Source to Basis Relationship Target Residual Offset

SetID: SHARE Rule ID: AL_FIXED

Allocation Rule Find | View All First 1 of 1 Last

Effective Date: 01/01/1900 Status: Active

Description: Demo Fixed Source and Basis

Use Fixed Basis [Choose Fixed Basis Dimension](#)

*Tree Level: *Basis Adjustment:

Fixed Basis Customize Find First 1-3 of 3 Last

Account	Tree	Resolve to Details		
<input type="text" value="400000"/>		<input type="checkbox"/>	<input type="button" value="+"/>	<input type="button" value="-"/>
<input type="text" value="403000"/>		<input type="checkbox"/>	<input type="button" value="+"/>	<input type="button" value="-"/>
<input type="text" value="450900"/>		<input type="checkbox"/>	<input type="button" value="+"/>	<input type="button" value="-"/>

Refresh

In correction mode, click Refresh to update the page with any additions or deletions to the DataMaps.

Use Fixed Basis

Select to use a fixed basis for this rule.

Basis Adjustment

Enter a basis adjustment against the basis datamap.

This field defaults to 1, but you can input a larger number for a multiplying factor against basis amounts.

You can also input a negative number adjustment so that the basis amounts are multiplied by a negative number.

Note: The allocation application engine applies the basis adjustment to the basis before it is used within the allocation calculation.

Note: If you selected the copy method, only the Basis Adjustment field is available on this page.

Selecting a Basis

Basis

Select a basis from the available options. The options are based on the datamaps that are listed on the Allocation DataMap Setup page

Constraint

Select a constraint for the basis DataMap.

If the criteria that you selected requires meta-value objects, the Assign Values to Value Objects group box appears and you can assign values to the constraint at this time.

Demo Basis for Allocations

For the Basis field, click the link to access the General Properties (DataMap) page and view the selected DataMap.

For the Constraint field, click the link to access the Constraint page and view the selected constraint.

Create New Constraint

Click to access to the Constraint setup page and create a new constraint.

Base Scenario

If you selected the Period Based Allocation check box on the Method page, select a basis scenario. Allocation Manager uses the calendar for the basis scenario that you selected to determine from which periods to use to prorate or spread the allocations.

The target has the same fiscal year and accounting period as the basis.

Arithmetic Measure

For the arithmetic method, select a measure. The available measures are based on the datamap that you selected as the basis.

Using a Fixed Basis

Choose Fixed Basis Dimension

For a fixed basis, click this link to access the Fixed Basis Dimension Setup page and select your dimensions.

Click OK to return to the Basis page.

Prorata Measure

For the prorata method, select a prorata measure.

The available measures are based on the datamap that you selected as the basis.

Tree Level

Select the tree level to which you want to resolve your allocation. Your options are: *No Tree Allocation*, *Allocate to leaf level*, or *Allocate to next node*.

Select the values for each of the dimensions that you selected on the Fixed Basis Dimension Setup page.



Click the Use Tree Node next to any dimension field in which it displays to select the value using the Tree Viewer.

Resolve to Details

For tree node values, select this check box to sort your tree node information in ascending order.

Rules - Source to Basis Relationship Page

Use the Rules - Source to Basis Relationship page (PF_AL_COM_DIM_DEFN) to review all common dimensions between the source and the basis.

Navigation

EPM Foundation, Data Enrichment Tools, Allocation Manager, Rules, Source to Basis Relationship

Image: Source to Basis Relationship page

This example illustrates the fields and controls on the Source to Basis Relationship page.

The screenshot shows the 'Source to Basis Relationship' page with the following details:

- Tabs: Method, Source, Basis, **Source to Basis Relationship**, Target, Residual, Offset
- SetID: SHARE, Rule ID: AL_PRO
- Allocation Rule: Find | View All | First | 1 of 1 | Last
- Effective Date: 01/01/1900, Status: Active, Refresh button
- Description: Demo Prorata Method
- Text: "The Source to Basis relationship is used to determine which dimensions will be common between them, referred to as common dimensions. Common dimensions are very important. They determine how the rule is to be applied. Common dimensions determine the divisor, therefore the ratio, for Spread Even and ProRata. For all methods they determine how the Source and Basis are to be grouped before applying any mathematical formulas."
- Table: Common Dimensions (Customize | Find | First | 1-6 of 6 | Last)

Use	Source Field	Relationship	Basis Field
<input checked="" type="checkbox"/>	Operating Unit	=	Operating Unit
<input checked="" type="checkbox"/>	Account	=	Account
<input checked="" type="checkbox"/>	Department	=	Department
<input checked="" type="checkbox"/>	Customer ID	=	Customer ID
<input checked="" type="checkbox"/>	Product ID	=	Product ID
<input checked="" type="checkbox"/>	Channel ID	=	Channel ID

This page lists the common dimensions. Common dimensions are very important as they are used to determine the divisor, therefore the ratio, for the spread even and prorata methods. For all methods, they determine how the source and basis are to be grouped before applying any mathematical formulas. Select from which source to base dimensions by selecting the Use check box.

The system determines the common dimensions using the descriptions of the datamaps for the source and basis. It considers that any source and basis fields with the same description represent the same information and are therefore a common dimension if the fields are marked as dimensions on both the source and basis. When looking for common dimensions, the system ignores case and spaces at the beginning and end of the description.

Rules - Target Page

Use the Rules - Target page (PF_AL_TGT_DEFN) to define the target, including the allocation mappings. The target is the destination to which the amounts that are defined by the source and basis are allocated.

Navigation

EPM Foundation, Data Enrichment Tools, Allocation Manager, Rules, Target

Image: Target page

This example illustrates the fields and controls on the Target page. You can find definitions for the fields and controls later on this page.

The screenshot shows the 'Target' page with the following details:

- Method:** Target
- SetID:** SHARE
- Rule ID:** AL_PRO
- Allocation Rule:** Demo Prorata Method
- Effective Date:** 01/01/1900
- Status:** Active
- *Target:** ALLOC_TGT (with a link to 'Demo Target For Allocations')

The **Target Map** table is as follows:

Target Field:	Data Type	Negate	Map from			Map to
			Source	Basis	List	
Business Unit	Dimension	<input type="checkbox"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	%BusinessUnit
Fiscal Year	Dimension	<input type="checkbox"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	%FiscalYear
Accounting Period	Dimension	<input type="checkbox"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	%AccountingPeriod
Scenario ID	Dimension	<input type="checkbox"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	%ScenarioId
Operating Unit	Dimension	<input type="checkbox"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	Operating Unit
Account	Dimension	<input type="checkbox"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	Account
Department	Dimension	<input type="checkbox"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	Department
Customer ID	Dimension	<input type="checkbox"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	Customer ID
Product ID	Dimension	<input type="checkbox"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	Product ID
Channel ID	Dimension	<input type="checkbox"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	Channel ID
Posted Base Currency Amount	Measure	<input type="checkbox"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	Posted Base Currency Amou

Target Select the target datamap for the allocation.

Demo Target for Allocations Click the link to access the General Properties (DataMap) page and view the selected target DataMap.

Target Field and Data Type The system populates these fields based on the target you select.

Negate	Select the Negate check box to reverse the sign of the amount when the data is copied to the target table. The Negate check box applies only to target fields with a data type of <i>Measure</i> .
(Map from) Source	Select to use the field values from the source records.
(Map from) Basis	Select to use the field values from the basis records.
(Map from) List	Select to use predefined value objects.
Map To	Specify the mapping for your selection.

Allocation Manager Rules - Residual Page

Use the Allocation Manager Rules - Residual page (PF_AL_RESID_DEFN) to assign residual amounts after the source amount has been allocated to dimensions in the target datamap. You can specify the residual account to use and other dimensions as applicable.

Navigation

EPM Foundation, Data Enrichment Tools, Allocation Manager, Rules, Residual

Image: Residual page

This example illustrates the fields and controls on the Residual page. You can find definitions for the fields and controls later on this page.

Allocate Residual

Select this check box to assign residual amounts after the source amount has been allocated to dimensions in the target datamap.

The page displays the target fields for the residual map.

List

Select this check box to allocate residual amounts of a target field to a value object.

The Map To field displays the value object.

Map To

If you select the List check box, the system displays the value object. Otherwise, you can enter the appropriate dimension value to use in this field. For example, to allocate residual amounts to a specific account, do not select the List check box. Simply select an account from the list of departments. You must select a value for at least one of the dimensions.

The following table provides an example of an arithmetic allocation rule in which the residual source amount is set up with a target department value of RES (Residual). DIV1, 2, and 3 use a total of 400. The residual amount is 600, so that amount is placed in the RES account.

Source		Basis		Target	
Source Department	Amount	Basis Department	Unit	Department	Amount
CORP	1000	DIV1	0.1	DIV1	100
		DIV2	0.1	DIV2	100
		DIV3	0.2	DIV3	200
				RES (Residual Account)	600

Rules - Offset Page

Use the Rules - Offset page (PF_AL_OFFSET_DEFN) to specify offset target information based on dimensions in the target datamap.

Navigation

EPM Foundation, Data Enrichment Tools, Allocation Manager, Rules, Offset

Image: Offset page

This example illustrates the fields and controls on the Offset page.

The screenshot shows the 'Offset' tab selected in a navigation bar. The main content area is divided into several sections:

- SetID:** SHARE
- Rule ID:** AL_PRO
- Allocation Rule:** Includes a search bar with 'Find | View All | First | 1 of 1 | Last' and expand/collapse buttons.
- Effective Date:** 01/01/1900
- Status:** Active
- Description:** Demo Prorata Method
- *Target:** Demo Target For Allocations
- *Allocate Offset:** None (dropdown menu)
- Allocate Offset for Residual:**
- Offset Map:** Includes a search bar and a table with columns: Target Field, Same as Source, Same as Basis, List, and Map to. Each column has a checkbox.
- Offset for Residual Map:** Includes a search bar and a table with columns: Target Field, List, and Map to. Each column has a checkbox.

Image: Offset page with single row offset option

This example illustrates the fields and controls on the Offset page with single row offset option. You can find definitions for the fields and controls later on this page.

Method Source Basis Source to Basis Relationship Target Residual **Offset**

SetID: SHARE Rule ID: GC_PRORT

Allocation Rule Find | View All First 1 of 1 Last

Effective Date: 01/01/1900 Status: Active

Description: Demo GC Prorata *Target: GC allocations jrn target

*Allocate Offset: Single Row Offset Allocate Offset for Residual

Offset Map Find First 1-32 of 32 Last

Target Field:	Same as Source	Same as Basis	List	Map to
PF Batch ID	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	%Blank
Business Unit	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	%BusinessUnit
Scenario ID	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	%ScenarioId
Fiscal Year	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	%FiscalYear
Accounting Period	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	%AccountingPeriod
Ledger Business Unit	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	11000
Source Process	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	07
Account	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	670000
Affiliate	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	%Blank
Operating Unit	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	%Blank
Department	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	%Blank
Foreign Currency Code	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	USD
Last Edit Seq Number	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	%Zero
Currency Code	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	%Blank
Process Instance	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	%ProcessInstance
Monetary Amount				
Foreign Amount				
Statistic Amount				
Load Date and Time	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	%CurrentDateTime
Error Flag	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	%No

Offset for Residual Map Find First 1 of 1 Last

Target Field:	List	Map to
	<input type="checkbox"/>	

Allocate Offset

Select this check box to allocate offset target information based on dimensions in the target datamap.

The system displays the target fields for the offset map.

Allocate Offset for Residual

Select this check box to allocate offset amounts for a residual amount.

The system displays the target fields for the offset for residual map.

Same as Source

Instead of selecting a dimension value for the offset, select this check box to use the source dimension value you already specified.

Same as Basis

Instead of selecting a dimension value for the offset, select this check box to use the basis dimension value you already specified.

List

Select this check box to allocate the offset to a value object.

The Map To field displays the value object.

Map To

If the List check box is selected, this field displays the value object and is not available for input.

If the List check box is not selected, you can enter the appropriate dimension value to use in this field.

For example, to allocate offset amounts to a specific account, do not select the List check box. Simply select an account from the list of accounts.

Note: You must select a value for at least one of the dimensions.

Allocation Template Page

Use the Allocation Template page (PF_AL_TMPLT_DEFN) to predefine a set of source, basis, and target datamaps that you can later associate with any allocation rule.

Navigation

EPM Foundation, Data Enrichment Tools, Allocation Manager, Allocation Template

Image: Allocation Template page

This example illustrates the fields and controls on the Allocation Template page. You can find definitions for the fields and controls later on this page.

Allocation Template

SetID: SHARE Template ID: GC_TMPLT01

Effective Date: Find | View All First 1 of 1 Last

*Effective Date: 01/01/1900 *Status: Active

Description: Demo GC Allocations Template

Source

*Source DataMap: GCALLOSRC1 [GC Allocation Source 1](#)

Source Constraint: GCALLOCONST1 [GC Allocation Contrain 1](#)

Source Adjustment: 1.0000

Basis

*Basis DataMap: GCALLOBAS1 [GC Basis 1](#)

Basis Constraint: GCALLOCONST2 [GC Allocation Contrain 2](#)

Basis Adjustment: 1.0000

Target

*Target DataMap: GCJRNLMT [GC Journal Management](#)

Notes: Demo GC Allocations Template

Use the Allocation Template page to predefine a set of source, basis, and target DataMaps that you can later associate with any allocation rule.

Fields Common to this Page

Create New DataMap

Click the link to access the General Properties (DataMap) page and either create a new DataMap or select an existing DataMap.

Note: Once you select a DataMap and save the template, the link name changes to the DataMap description provided on the General Properties (DataMap) page. Also, the link will now access the General Properties (DataMap) page specific to the *selected* DataMap, where you can view or update details about the DataMap

Create New Constraint

Click this link to access the Constraints page and view details about the selected constraint.

Click the link to access the Constraints page and either create a new constraint or select an existing constraint.

Note: Once you select a constraint and save the template, the link name changes to the constraint description provided on the Constraint page. Also, the link will now access the Constraint page specific to the *selected* constraint, where you can view or update details about the constraint.

Source

Source DataMap

Select a source DataMap for the allocation template.

You can only select from DataMaps that are defined on the Allocation DataMap Setup page.

Source Constraint

Select a source DataMap constraint for the allocation template.

You can only select from constraints that are defined for this DataMap on the Constraints page.

Source Adjustment

Enter a source adjustment against the source datamap.

This field defaults to 1, but you can input a larger number for a multiplying factor against source amounts.

You can also input a negative number adjustment so that the source amounts are multiplied by a negative number.

Basis

Basis DataMap

Select a basis DataMap for the allocation template.

You can only select from DataMaps that are defined on the Allocation DataMap Setup page.

Basis Constraint

Select a basis DataMap constraint for the allocation template.

You can only select from constraints that are defined for this DataMap on the Constraints page.

Basis Adjustment

Enter a basis adjustment against the basis datamap.

This field defaults to 1, but you can input a larger number for a multiplying factor against basis amounts.

You can also input a negative number adjustment so that the basis amounts are multiplied by a negative number.

Note: The allocation application engine applies the basis adjustment to the basis before it is used within the allocation calculation.

Target

Target DataMap

Select a target DataMap for the allocation template.

You can only select from DataMaps that are defined on the Allocation DataMap Setup page.

Defining Allocation Manager Rule Sets

This topic provides an overview of Allocation Manager rule sets and discusses how to

- Set up Allocation Manager rule sets.
- Define post-processing routines for an allocation rule set.

Pages Used to Define Allocation Manager Rule Sets and Post-Processing Routines

<i>Page Name</i>	<i>Definition Name</i>	<i>Navigation</i>	<i>Usage</i>
Allocation Manager Rule Set	PF_AL_RULESET_DEFN	EPM Foundation, Data Enrichment Tools, Allocation Manager, Rule Set, Allocation Manager Rule Set	Set up an Allocation Manager rule set by defining the rules to include in the rule set.
Result Processing	PF_AL_PROC_OPTIONS	EPM Foundation, Data Enrichment Tools, Allocation Manager, Rule Set, Result Processing	Define post-processing routines that further process the results of the rule set.

Understanding Allocation Manager Rule Sets

You must define an Allocation Manager rule set for any Allocation Manager rules that you want to process. You can also use rule sets to combine multiple rules in one run of the Allocation Manager engine. Rules in a rule set run in the order that is shown on the Allocation Manager Rule Sets page. The first rule must complete successfully before the second rule starts. The system places the data from the first rule in a temporary table (which is assigned by the engine) that will be the source of the second rule, and so on.

Note: A rule must be included in a rule set, even if only one rule is processed.

After you define a rule set, you can create post-processing routines that further process the results of the rule set. You can also tie the routines to a particular rule set so that they automatically run when the rule set completes. The following topics summarize specific features of the post-processing routines.

Auditing Allocation Rules

This feature provides visibility into the inputs and rules that calculate your allocations. Runtime information, source data transformations, and basis data transformations are some of the data that is captured by the audit feature. The data is captured in logs and is stored in the corresponding Operational Warehouse - Enriched (OWE) tables. The following table contains a complete list of delivered audit logs and the data warehouse table in which the data reside:

Log Type	Corresponding OWE Table	Data Captured
Runtime Log	PF_AL_AUD_RUNLG	Captures all rule and runtime values. All the following table entries are tied together by this unique runtime log key.
Common Dimensions Log	PF_AL_AUD_CDIM	Captures common dimensions.
Source & Basis to Target Mapping Log	PF_AL_AUD_MAP	Mapping from source/basis columns to target column.
Source Log	PF_AL_AUD_SRC	Captures input source data.
Basis Log	PF_AL_AUD_BAS	Captures input basis data.
Target Log	PF_AL_AUD_TGT	Captures target data.
Mapping Logs	PF_AL_AUD_SRMAP PF_AL_AUD_BSMAP PF_AL_AUD_TGMAP	Captures source, basis, and target name mappings.

You can view the audit logs using any SQL Query Tool.

Note: If a Ledger ID is not entered in the post processing page, a batch ID cannot be generated for the audit trail.

Merging Allocation Results to Permanent Tables

After an allocation rule set completes processing, the results are stored in a temporary target data warehouse table. This feature automatically loads the results that are stored in the temporary target table to a permanent non-journal target table—for example, PS_PF_AL_TGT.

This feature should be used if the target is not a journal table. The target tables should have the *Merge Allowed* check box selected on the Record Metadata page.

Note: Merging occurs automatically for period-based allocations and post processing is not required.

Validating Journals

If you load your data to a journal target table, this feature ensures the journal dimensions have valid values. For example, you might create allocation rules at the beginning of the year. At the time of allocation rule creation, you specified account 100 as one of the target accounts for the allocation. Account 100 is a valid account. Suppose that during the year, account 100 is deactivated. If the allocation process is run and validate journal is checked, the process creates an error, as account 100 is no longer valid. Invalid journal dimensions are sent to the journal error table (PF_JRNL_E00).

Posting Journals

This feature calls the PF_POST application engine to the post a journal entry to the ledger indicated by the Ledger ID.

Reversing Journals

If you load your data to a journal target table, this feature automatically reverses the journal entry. This process creates an identical reversed set of journal entries and results in two journal entries: the original journal and another with all the amount signs reversed. In addition, you can specify whether you want to reverse a journal for the same or alternate accounting period.

If you select the validate and post options with the reverse journal option, the reversed journal entries will also be validated and posted. In the case of specific period, proper error checking is done by the Allocation Manager to determine if the entered fiscal year and accounting period are valid.

Allocation Manager Rule Set Page

Use the Allocation Manager Rule Set page (PF_AL_RULESET_DEFN) to set up an Allocation Manager rule set by defining the rules to include in the rule set.

Navigation

EPM Foundation, Data Enrichment Tools, Allocation Manager, Rule Set, Allocation Manager Rule Set

Image: Allocation Manager Rule Set page

This example illustrates the fields and controls on the Allocation Manager Rule Set page. You can find definitions for the fields and controls later on this page.

The screenshot shows the 'Allocation Manager Rule Set' page. At the top, there are two tabs: 'Allocation Manager Rule Set' and 'Result Processing'. The 'Allocation Manager Rule Set' tab is selected. Below the tabs, there are two sub-sections. The first sub-section is titled 'Allocation Manager Rule Set' and contains the following fields: SetID: SHARE, Ruleset Id: GC_PRORT, Effective Date: 01/01/1900, Status: Active, Description: GC Prorata, and Notes. The second sub-section is titled 'Rules in Rule Set' and contains a table with the following data:

*Sequence Number	*Allocation Rule ID
10	GC_PRORT

Ruleset ID Displays the unique identifier for this rule set definition.

Rules in Rule Set

Sequence Number Enter a number for the first rule ID in the rule set (for example 100).
This number represents the sequencing order in which the rules run.
The next rule ID to run might have a sequence number of 200.
The actual sequence number is not important; it represents the order in which you want to process rules. The sequence must be unique.

Allocation Rule ID Select the Allocation Manager rule to include in the rule set.
You create the rules using the Allocation Manager Rules component.

Result Processing Page

Use the Result Processing page (PF_AL_PROC_OPTIONS) to define post-processing routines that further process the results of the rule set.

Navigation

EPM Foundation, Data Enrichment Tools, Allocation Manager, Rule Set, Result Processing

Image: Result Processing page

This example illustrates the fields and controls on the Result Processing page. You can find definitions for the fields and controls later on this page.

Ruleset ID

Displays the unique identifier for this rule set definition.

Create Audit Trail

Select this check box to generate audit logs of the inputs and rules that calculate your allocations.

Runtime information, source data transformations, and basis data transformations are some of the data that is captured by the audit logs. You can view the audit logs using PeopleSoft Query.

Options

Create Journal

Select this option to automatically load the allocation output into a target Journal line table (for example, PS_PF_JRNL_F00).

At runtime, the PF_EDIT process is called and a batch header (PF_BATCH_TBL) is created, which moves the journal lines from a temporary table to a permanent journal table. You can view results of this process using any SQL query tool such as Toad, Query Analyzer, and so on. When this option is selected, the Journal Options group box becomes available.

Merge to Non Journal Target

Select this option to automatically load the allocation output into a non-journal permanent target table. At runtime, the PF_MERGE process is called and loads the allocation output from the temporary table to a permanent table that you specify in the

target setup page. You can view results of this process using any SQL query tool such as Toad, Query Analyzer, and so on.

None

Select this option if you do not want to load the allocation output into a permanent target table. When this option is selected, data from the allocation process remains in the temporary table. This option is the default for the upgrade process.

Journal Options

Validate Journal

Select this check box to have the PF_EDIT engine validate the journal dimensions that are specified during the rule setup.

When this option is selected, the system validates all journal dimensions. The system also rejects any values that do not appear on the corresponding prompt table. Invalid values are sent to the journal error table (PF_JRNL_E00).

If you do not select this option, the journal is created without validation and will be posted regardless of errors. It is recommended that you select this option. Note, however, that the validation process does increase processing time.

Ledger ID

Select the ledger ID to generate a corresponding batch ID, which is used to post the journal to the ledger.

Post Journal

Select this check box to store the data in a journal table and automatically post it to a permanent ledger table (for example, PF_LEDGER_F00). At runtime, the system calls PF_POST, which updates the ledger.

Reverse Journal

Select this check box to reverse a journal entry for the same or alternate accounting period. This process results in two journal entries: the original journal and another with all the amount signs reversed. When this option is selected, the Reverse Journal Options group box becomes available and enables you to further specify a particular accounting period for the reversed journal.

Next Period

Select this option to reverse a journal for the next accounting period.

Specific Period

Select this option to reverse a journal for a specific accounting period.

Fiscal Year

Enter the fiscal year for which you want to reverse the journal.

Accounting Period

Enter the accounting period for which you want to reverse the journal.

The following table provides examples of the edit, validate, and post options used in combination with one another and the resulting output of the combination:

Options Selected	Result
Edit in Journal and Validate Journal	Allocation output is automatically loaded into a target journal line table. A journal with dimensions is checked for balancing and batch header with a valid status.
Edit in Journal and Post Journal	The allocation output is automatically loaded into a target journal line table. The post process (PF_POST) is called to post data to a specified ledger.
Edit in Journal, Validate Journal, and Post Journal	The allocation output is automatically loaded into a target journal line table. A journal with dimensions is checked for balancing and batch header with a valid status. The post process (PF_POST) is called to post data to a specified ledger.

Please note that these are merely examples of edit, validate, and post option-combinations; you can choose a number of different combinations.

Renaming Allocation Manager Rules

This topic discusses how to rename an Allocation Manager rule.

Page Used to Rename Allocation Manager Rules

Page Name	Definition Name	Navigation	Usage
Rename Allocation Rule	PF_AL_RULE_RENAME	EPM Foundation, Data Enrichment Tools, Allocation Manager, Rename Rules, Rename Allocation Rule	Rename existing Allocation Manager rules.

Rename Allocation Rule Page

Use the Rename Allocation Rule page (PF_AL_RULE_RENAME) to rename existing Allocation Manager rules.

Navigation

EPM Foundation, Data Enrichment Tools, Allocation Manager, Rename Rules, Rename Allocation Rule

Image: Rename Allocation Rule page

This example illustrates the fields and controls on the Rename Allocation Rule page. You can find definitions for the fields and controls later on this page.

Ruleset Id	Effective Date	Description
1 AL_ARITH	01/01/1900	Demo Arithmetic

SetID and Rule ID

Select the SetID and rule to rename.

Search

Click the Search button. The system populates the Search Results grid with a list of the rule sets that affected by the renaming process.

Rename

To proceed with the renaming process, click the Rename button and enter a new name for the rule. Every instance of the original rule name is changed to the new name.

Associating Allocation Manager Rule Sets With a Job and Processing Allocations

The next steps required to process allocations include associating an allocation rule set with a job and running the Allocation Manager engine.

Each unique run of the Allocation Manager engine is assigned a job ID so you must associate the rules with the actual job that will run them using the Allocation Manager Rule Set Job Association page. When the jobstream that contains the defined Allocation Manager job ID runs, it uses the setup on the Allocation Manager Rule Set Job Association page to find the Allocation Manager rule set to execute.

Note: Each time that you run an EPM engine, it must have a unique job ID. These jobs must then be run in a jobstream.

Pages Used to Associate Allocation Manager Rule Sets with a Job and Process Allocations

<i>Page Name</i>	<i>Definition Name</i>	<i>Navigation</i>	<i>Usage</i>
Allocation Manager Rule Set Job Association	PF_AL_JOB_RSET_DEF	EPM Foundation, Data Enrichment Tools, Allocation Manager, Job Association, Allocation Manager Rule Set Job Association	Define rule sets to be run with a given job and the order in which these rule sets are run.
Run Allocations	PF_RUN_ALLOCATION	EPM Foundation, Data Enrichment Tools, Allocation Manager, Run Allocations, Run Allocations EPM Foundation, Data Enrichment Tools, Allocation Manager, Run Allocations, Run Allocations	Process allocation rules.

Prerequisites

Before you can associate rule sets with job, you must:

- Set up the Allocation Manager job ID.
- Set up a jobstream for the job ID.

See [Understanding Jobstreams](#)

Allocation Manager Rule Set Job Association Page

Use the Allocation Manager Rule Set Job Association page (PF_AL_JOB_RSET_DEF) to define rule sets to be run with a given job and the order in which these rule sets are run.

Navigation

EPM Foundation, Data Enrichment Tools, Allocation Manager, Run Allocations, Run Allocations

Image: Run Allocations page

This example illustrates the fields and controls on the Run Allocations page.

Run Allocations

User ID: VP1 [Report Manager](#) [Process Monitor](#)

Run Control ID: 001 [Clear Last Suite](#) [Clear All Suites](#)

Process Information

Program Name: PF_JOBSTREAM When: Always Send Email Notification

*Description: Fixed basis [Specify Email Parameters](#)

*Unit: 10000 World Wide Consolidation

*Scenario ID: FINAL Final Budget

Fiscal Year: 2005

Period: 1

*Jobstream ID: AL_FIXED Rerun

Last Run On: 02/02/2010 1:40:31PM PST As Of Date: 01/31/2005

As a last step to process your allocations, run the Allocation Manager engine.

As you run the jobstream:

- Use the Process Monitor to verify the status of your job.
- Review any error messages using the Error Messages component.

Related Links

[Viewing Engine Messages](#)

Querying Allocations

Allocation Manager enables you to output allocation results to target batches or journals, or merge data directly to a ledger target table. Additionally, Allocation Manager sometimes uses PeopleSoft Global Consolidations or Activity Based Management outputs as sources. Given these complex processes, it can be difficult to determine if the allocation output was calculated correctly based solely on viewing the results. As such, PeopleSoft EPM provides the new Allocation Manager Inquiry tool that enables you to review allocation audit data, from the target to source—specifically, you can review allocation targets, drill down to basis and source elements of an allocation, and view calculation rules.

The Allocation Manager Inquiry tool can also integrate with Global Consolidation and Activity Based Management allocation rules.

See [Allocation Manager Inquiry Results - Drill to Source Page](#), "Auditing Consolidation Data (*PeopleSoft EPM 9.1: Global Consolidations*)", and "Using Object Navigator and Model Analyzer (*PeopleSoft EPM 9.1: Activity-Based Management*)".

Allocation Manager Audit Tables

The Allocation Manager Inquiry tool pulls data from the following EPM audit tables:

- Audit Target table (PF_AL_AUD_TGT)
- Audit Target Map table (PF_AL_AUD_TGMAP)
- Audit Basis table (PF_AL_AUD_BAS)
- Audit Basis Map table (PF_AL_AUD_BSMAP)
- Audit Source table (PF_AL_AUD_SRC)
- Audit Source Map table (PF_AL_AUD_SRMAP)
- Audit Common Dimension table (PF_AL_AUD_CDIM)
- Audit Map table (PF_AL_AUD_MAP)
- Audit Run log table (PF_AL_AUD_RUNLG)

Pages Used to Query Target Allocations

<i>Page Name</i>	<i>Definition Name</i>	<i>Navigation</i>	<i>Usage</i>
Allocation Manager Inquiry - Audit Parameters	PF_AL_INQUIRY1	EPM Foundation, Data Enrichment Tools, Allocation Manager, Allocation Manager Inquiry	Specify audit criteria for a specific allocation rule.
Allocation Manager Inquiry Results	PF_AL_INQUIRY2	Click Go on the Allocation Manager Inquiry - Audit Parameters page	View allocation rule data that matched your criteria. This data is the starting point for the audit trail.
Allocation Manager Inquiry Results - Drill to Basis	PF_AL_INQUIRY3	Click the Drill to Basis link on the Drill Details tab of the Allocation Manager Inquiry Results page	View basis rule details for an allocation.
Allocation Manager Inquiry Results - Drill to Source	PF_AL_INQUIRY4	Click the Drill to Source link on the Drill Details tab of the Allocation Manager Inquiry Results page	View source rule details for an allocation.

Allocation Manager Inquiry - Audit Parameters Page

Use the Allocation Manager Inquiry - Audit Parameters page (PF_AL_INQUIRY1) to specify audit criteria for a specific allocation rule.

Navigation

EPM Foundation, Data Enrichment Tools, Allocation Manager, Allocation Manager Inquiry

Image: Allocation Manager Inquiry - Audit Parameters page

This example illustrates the fields and controls on the Allocation Manager Inquiry - Audit Parameters page. You can find definitions for the fields and controls later on this page.

Use the Allocation Manager Inquiry - Audit Parameters page to specify audit criteria for a specific allocation rule.

Select the business unit, scenario ID, and jobstream ID associated with allocation you want to audit. You must also enter the fiscal year and period for the allocation. You may select a batch ID associated with the allocation to audit, but this field is optional since you may want to inquire on multiple batches at one time or the target merged directly to the datamap and did not create a batch.

Suppress Blank Dimensions

Select this check box to prevent dimension columns without values from displaying in the results grid.

Max Rows to Scroll

Enter the maximum number of rows you want displayed in the results grid.

Go

Click to process the query parameters you provided and view the results grid.

Save Preference

Click to access the Save Inquiry Preference page and save the query parameters you input for future use.

You can input a name and description for your preference.

With the Default Preference check box you can set your parameters as the default values for the page, meaning the parameters automatically populate the parameters page when a user accesses it.

Get Preference

Click to access the Select Inquiry Preference page where you can select a preference that will automatically populate the parameters page with its values.

With the Default Preference check box you can set the preference as the default for the page, meaning the parameters

automatically populate the parameters page when a user accesses it.

Allocation Manager Inquiry Results Page

Use the Allocation Manager Inquiry Results page (PF_AL_INQUIRY2) to view allocation rule data that matched your criteria. This data is the starting point for the audit trail.

Navigation

Click Go on the Allocation Manager Inquiry - Audit Parameters page

Image: Allocation Manager Inquiry Results page

This example illustrates the fields and controls on the Allocation Manager Inquiry Results page. You can find definitions for the fields and controls later on this page.

Allocation Manager Inquiry																																																																																																																																																																																			
Main Criteria																																																																																																																																																																																			
Business Unit:	10000	World Wide Consolidation	Scenario ID:	GCACTTB	Actual Trial Balance																																																																																																																																																																														
Fiscal Year:	2005		Period:	1																																																																																																																																																																															
Jobstream ID:	GC_ALLO	GC Allocation																																																																																																																																																																																	
⏪ ⏩ 1 to 25 of 34 ⏪ ⏩																																																																																																																																																																																			
Go To: Selection Page																																																																																																																																																																																			
Target Details																																																																																																																																																																																			
Customize Find 1-25 of 25																																																																																																																																																																																			
<table border="1"> <thead> <tr> <th>PF Batch ID</th> <th>Monetary Amount</th> <th>Currency</th> <th>Row Description</th> <th>Ledger Business Unit</th> <th>Source Process</th> <th>Account</th> <th>Affiliate</th> <th>Operating Unit</th> <th>D</th> </tr> </thead> <tbody> <tr> <td>GC_ALL4039</td> <td>-656.13</td> <td>USD</td> <td>Offset</td> <td>11110</td> <td>07</td> <td>200200</td> <td>11000</td> <td></td> <td></td> </tr> <tr> <td>GC_ALL4039</td> <td>656.13</td> <td>USD</td> <td>Allocation</td> <td>11110</td> <td>07</td> <td>620000</td> <td></td> <td>EMEA</td> <td></td> </tr> <tr> <td>GC_ALL4039</td> <td>-52.38</td> <td>USD</td> <td>Offset</td> <td>11120</td> <td>07</td> <td>200200</td> <td>11000</td> <td></td> <td></td> </tr> <tr> <td>GC_ALL4039</td> <td>52.38</td> <td>USD</td> <td>Allocation</td> <td>11120</td> <td>07</td> <td>620000</td> <td></td> <td>EMEA</td> <td></td> </tr> <tr> <td>GC_ALL4039</td> <td>-87.07</td> <td>USD</td> <td>Offset</td> <td>11130</td> <td>07</td> <td>200200</td> <td>11000</td> <td></td> <td></td> </tr> <tr> <td>GC_ALL4039</td> <td>87.07</td> <td>USD</td> <td>Allocation</td> <td>11130</td> <td>07</td> <td>620000</td> <td></td> <td>EMEA</td> <td></td> </tr> <tr> <td>GC_ALL4039</td> <td>-16.84</td> <td>USD</td> <td>Offset</td> <td>11140</td> <td>07</td> <td>200200</td> <td>11000</td> <td></td> <td></td> </tr> <tr> <td>GC_ALL4039</td> <td>16.84</td> <td>USD</td> <td>Allocation</td> <td>11140</td> <td>07</td> <td>620000</td> <td></td> <td>EMEA</td> <td></td> </tr> <tr> <td>GC_ALL4039</td> <td>-728.57</td> <td>USD</td> <td>Offset</td> <td>11150</td> <td>07</td> <td>200200</td> <td>11000</td> <td></td> <td></td> </tr> <tr> <td>GC_ALL4039</td> <td>728.57</td> <td>USD</td> <td>Allocation</td> <td>11150</td> <td>07</td> <td>620000</td> <td></td> <td>EMEA</td> <td></td> </tr> <tr> <td>GC_ALL4039</td> <td>-7.12</td> <td>USD</td> <td>Offset</td> <td>11210</td> <td>07</td> <td>200200</td> <td>11000</td> <td></td> <td></td> </tr> <tr> <td>GC_ALL4039</td> <td>7.12</td> <td>USD</td> <td>Allocation</td> <td>11210</td> <td>07</td> <td>620000</td> <td></td> <td>APAC</td> <td></td> </tr> <tr> <td>GC_ALL4039</td> <td>-19.26</td> <td>USD</td> <td>Offset</td> <td>11220</td> <td>07</td> <td>200200</td> <td>11000</td> <td></td> <td></td> </tr> <tr> <td>GC_ALL4039</td> <td>19.26</td> <td>USD</td> <td>Allocation</td> <td>11220</td> <td>07</td> <td>620000</td> <td></td> <td>APAC</td> <td></td> </tr> <tr> <td>GC_ALL4039</td> <td>-14.00</td> <td>USD</td> <td>Offset</td> <td>11310</td> <td>07</td> <td>200200</td> <td>11000</td> <td></td> <td></td> </tr> <tr> <td colspan="10">Sum Total Amount: 0.00 USD</td> </tr> </tbody> </table>										PF Batch ID	Monetary Amount	Currency	Row Description	Ledger Business Unit	Source Process	Account	Affiliate	Operating Unit	D	GC_ALL4039	-656.13	USD	Offset	11110	07	200200	11000			GC_ALL4039	656.13	USD	Allocation	11110	07	620000		EMEA		GC_ALL4039	-52.38	USD	Offset	11120	07	200200	11000			GC_ALL4039	52.38	USD	Allocation	11120	07	620000		EMEA		GC_ALL4039	-87.07	USD	Offset	11130	07	200200	11000			GC_ALL4039	87.07	USD	Allocation	11130	07	620000		EMEA		GC_ALL4039	-16.84	USD	Offset	11140	07	200200	11000			GC_ALL4039	16.84	USD	Allocation	11140	07	620000		EMEA		GC_ALL4039	-728.57	USD	Offset	11150	07	200200	11000			GC_ALL4039	728.57	USD	Allocation	11150	07	620000		EMEA		GC_ALL4039	-7.12	USD	Offset	11210	07	200200	11000			GC_ALL4039	7.12	USD	Allocation	11210	07	620000		APAC		GC_ALL4039	-19.26	USD	Offset	11220	07	200200	11000			GC_ALL4039	19.26	USD	Allocation	11220	07	620000		APAC		GC_ALL4039	-14.00	USD	Offset	11310	07	200200	11000			Sum Total Amount: 0.00 USD									
PF Batch ID	Monetary Amount	Currency	Row Description	Ledger Business Unit	Source Process	Account	Affiliate	Operating Unit	D																																																																																																																																																																										
GC_ALL4039	-656.13	USD	Offset	11110	07	200200	11000																																																																																																																																																																												
GC_ALL4039	656.13	USD	Allocation	11110	07	620000		EMEA																																																																																																																																																																											
GC_ALL4039	-52.38	USD	Offset	11120	07	200200	11000																																																																																																																																																																												
GC_ALL4039	52.38	USD	Allocation	11120	07	620000		EMEA																																																																																																																																																																											
GC_ALL4039	-87.07	USD	Offset	11130	07	200200	11000																																																																																																																																																																												
GC_ALL4039	87.07	USD	Allocation	11130	07	620000		EMEA																																																																																																																																																																											
GC_ALL4039	-16.84	USD	Offset	11140	07	200200	11000																																																																																																																																																																												
GC_ALL4039	16.84	USD	Allocation	11140	07	620000		EMEA																																																																																																																																																																											
GC_ALL4039	-728.57	USD	Offset	11150	07	200200	11000																																																																																																																																																																												
GC_ALL4039	728.57	USD	Allocation	11150	07	620000		EMEA																																																																																																																																																																											
GC_ALL4039	-7.12	USD	Offset	11210	07	200200	11000																																																																																																																																																																												
GC_ALL4039	7.12	USD	Allocation	11210	07	620000		APAC																																																																																																																																																																											
GC_ALL4039	-19.26	USD	Offset	11220	07	200200	11000																																																																																																																																																																												
GC_ALL4039	19.26	USD	Allocation	11220	07	620000		APAC																																																																																																																																																																											
GC_ALL4039	-14.00	USD	Offset	11310	07	200200	11000																																																																																																																																																																												
Sum Total Amount: 0.00 USD																																																																																																																																																																																			

This page displays the target details associated with the allocation you selected for audit. Some of the columns displayed on this page are Monetary Amount, Row Description, and Ledger Business Unit. Values for the row description field include *Allocation*, *Offset*, and *Residual*.

If you did not specify a specific batch on the Allocation Manager Inquiry - Audit Parameters page, this page also displays the PF Batch ID.

If you did not select the Suppress Blank Dimensions check box on the Allocation Manager Inquiry - Audit Parameters page, this page will display columns without values.

Click the Selection Page link to return to the Allocation Manager Inquiry - Audit Parameters page.

Note: The Audit Target (PF_AL_AUD_TGT) table is the source for all the column values on this page.

Drill Details Tab

Click the Drill Details tab to drill down to the basis or source information for an allocation.

Image: Drill Details tab

This example illustrates the fields and controls on the Drill Details tab. You can find definitions for the fields and controls later on this page.

PF Batch ID	Monetary Amount	Currency	Row Description	Drill to Basis	Drill to Source
GC_ALL4039	-656.13	USD	Offset		
GC_ALL4039	656.13	USD	Allocation	Drill to Basis	Drill to Source
GC_ALL4039	-52.38	USD	Offset		
GC_ALL4039	52.38	USD	Allocation	Drill to Basis	Drill to Source
GC_ALL4039	-87.07	USD	Offset		
GC_ALL4039	87.07	USD	Allocation	Drill to Basis	Drill to Source
GC_ALL4039	-16.84	USD	Offset		
GC_ALL4039	16.84	USD	Allocation	Drill to Basis	Drill to Source
GC_ALL4039	-728.57	USD	Offset		
GC_ALL4039	728.57	USD	Allocation	Drill to Basis	Drill to Source
GC_ALL4039	-7.12	USD	Offset		
GC_ALL4039	7.12	USD	Allocation	Drill to Basis	Drill to Source
GC_ALL4039	-19.26	USD	Offset		
GC_ALL4039	19.26	USD	Allocation	Drill to Basis	Drill to Source
GC_ALL4039	-14.00	USD	Offset		
Sum Total Amount:		0.00	USD		

Click the Drill to Basis link to access the Allocation Manager Inquiry Results - Drill to Basis page and view basis details associated with the allocation.

Click the Drill to Source link to access the Allocation Manager Inquiry Results - Drill to Source page and view source details associated with the allocation.

Click the Selection Page link to return to the Allocation Manager Inquiry - Audit Parameters page.

Note: Only rows labeled with *Allocation* for the Row Description column have the option to drill to the source and basis.

Rule Detail Tab

Click the Rule Detail tab to drill down to the rule information for an allocation.

Image: Rule Detail tab on the Allocation Manager Inquiry Results page

This example illustrates the fields and controls on the Rule Detail tab on the Allocation Manager Inquiry Results page. You can find definitions for the fields and controls later on this page.

Go To: [Selection Page](#)

Target Details							Customize	Find	1-25 of 26	
Details							Drill Details	Rule Detail		
PF Batch ID	Monetary Amount	Currency	Row Description	Allocation Rule ID	Description	Allocation Method				
GC_ALL4039	-656.13	USD	Offset	GC ALLO1	Demo GC Allocation 1	Prorata				
GC_ALL4039	656.13	USD	Allocation	GC ALLO1	Demo GC Allocation 1	Prorata				
GC_ALL4039	-52.38	USD	Offset	GC ALLO1	Demo GC Allocation 1	Prorata				
GC_ALL4039	52.38	USD	Allocation	GC ALLO1	Demo GC Allocation 1	Prorata				
GC_ALL4039	-87.07	USD	Offset	GC ALLO1	Demo GC Allocation 1	Prorata				
GC_ALL4039	87.07	USD	Allocation	GC ALLO1	Demo GC Allocation 1	Prorata				
GC_ALL4039	-16.84	USD	Offset	GC ALLO1	Demo GC Allocation 1	Prorata				
GC_ALL4039	16.84	USD	Allocation	GC ALLO1	Demo GC Allocation 1	Prorata				
GC_ALL4039	-728.57	USD	Offset	GC ALLO1	Demo GC Allocation 1	Prorata				
GC_ALL4039	728.57	USD	Allocation	GC ALLO1	Demo GC Allocation 1	Prorata				
GC_ALL4039	-7.12	USD	Offset	GC ALLO1	Demo GC Allocation 1	Prorata				
GC_ALL4039	7.12	USD	Allocation	GC ALLO1	Demo GC Allocation 1	Prorata				
GC_ALL4039	-19.26	USD	Offset	GC ALLO1	Demo GC Allocation 1	Prorata				
GC_ALL4039	19.26	USD	Allocation	GC ALLO1	Demo GC Allocation 1	Prorata				
GC_ALL4039	-14.00	USD	Offset	GC ALLO1	Demo GC Allocation 1	Prorata				
Sum Total Amount:		0.00	USD							

Click the allocation rule link in the Allocation Rule ID column to access the Method page and view allocation method details, such as allocation method type and period based allocations.

Click the Selection Page link to return to the Allocation Manager Inquiry - Audit Parameters page.

Allocation Manager Inquiry Results - Drill to Basis Page

Use the Allocation Manager Inquiry Results - Drill to Basis page (PF_AL_INQUIRY3) to view basis rule details for an allocation.

Navigation

Click the Drill to Basis link on the Drill Details tab of the Allocation Manager Inquiry Results page

Image: Allocation Manager Inquiry Results - Drill to Basis page

This example illustrates the fields and controls on the Allocation Manager Inquiry Results - Drill to Basis page. You can find definitions for the fields and controls later on this page.

Allocation Manager Inquiry					
Main Criteria					
Business Unit:	10000	World Wide Consolidation	Scenario ID:	GCACTTB	Actual Trial Balance
Fiscal Year:	2005		Period:	1	
Jobstream ID:	GC_ALLO	GC Allocation			
Go To: Selection Page -> Target Details					
Basis					
Details Rule Detail <input type="text"/> Customize Find 1-11 of 11					
PF Batch ID	Basis Factor	Basis Amount	Basis Total	Ledger Business Unit	Operating Unit
GC_ALL4039	0.4011	-46,005.240	-114,682.660	11110	EMEA
GC_ALL4039	0.0320	-3,672.360	-114,682.660	11120	EMEA
GC_ALL4039	0.0532	-6,104.760	-114,682.660	11130	EMEA
GC_ALL4039	0.0102	-1,180.560	-114,682.660	11140	EMEA
GC_ALL4039	0.4454	-51,084.460	-114,682.660	11150	EMEA
GC_ALL4039	0.0043	-499.080	-114,682.660	11210	APAC
GC_ALL4039	0.0117	-1,350.720	-114,682.660	11220	APAC
GC_ALL4039	0.0085	-981.600	-114,682.660	11310	AMERICAS
GC_ALL4039	0.0002	-24.360	-114,682.660	11320	AMERICAS
GC_ALL4039	0.0165	-1,901.400	-114,682.660	11330	AMERICAS
GC_ALL4039	0.0163	-1,878.120	-114,682.660	11340	AMERICAS

This page displays the basis details associated with the allocation you selected for audit. Some of the columns displayed on this page are Basis Factor, Basis Amount, and Basis Total. The basis total is the sum of the basis amounts taken from the Audit Basis (PF_AL_AUD_BAS) table. The basis factor is calculated as the ratio between the basis amount and basis total. If an adjustment was applied to the basis (and basis adjustment is not equal 1) an additional column called Adjustment Applied will appear in the grid and display the adjustment factor that was applied.

Click the Selection Page link to return to the Allocation Manager Inquiry - Audit Parameters page.

Click the Target Details link to return to the Allocation Manager Inquiry Results page.

Rule Detail Tab

Click the Rule Detail tab to view the basis rule summary for an allocation.

Image: Rule Detail tab on the Drill to Basis page

This example illustrates the fields and controls on the Rule Detail tab on the Drill to Basis page. You can find definitions for the fields and controls later on this page.

Go To: [Selection Page](#) -> [Target Details](#)

Basis						
Customize Find 1-11 of 11						
Details Rule Detail [F5]						
PF Batch ID	Basis Factor	Basis Amount	Basis Total	Allocation Rule ID	Description	Allocation Method
GC_ALL4039	0.4011	-46,005.240	-114,682.660	GC_ALLO1	Demo GC Allocation 1	Prorata
GC_ALL4039	0.0320	-3,672.360	-114,682.660	GC_ALLO1	Demo GC Allocation 1	Prorata
GC_ALL4039	0.0532	-6,104.760	-114,682.660	GC_ALLO1	Demo GC Allocation 1	Prorata
GC_ALL4039	0.0102	-1,180.560	-114,682.660	GC_ALLO1	Demo GC Allocation 1	Prorata
GC_ALL4039	0.4454	-51,084.460	-114,682.660	GC_ALLO1	Demo GC Allocation 1	Prorata
GC_ALL4039	0.0043	-499.080	-114,682.660	GC_ALLO1	Demo GC Allocation 1	Prorata
GC_ALL4039	0.0117	-1,350.720	-114,682.660	GC_ALLO1	Demo GC Allocation 1	Prorata
GC_ALL4039	0.0085	-981.600	-114,682.660	GC_ALLO1	Demo GC Allocation 1	Prorata
GC_ALL4039	0.0002	-24.360	-114,682.660	GC_ALLO1	Demo GC Allocation 1	Prorata
GC_ALL4039	0.0165	-1,901.400	-114,682.660	GC_ALLO1	Demo GC Allocation 1	Prorata
GC_ALL4039	0.0163	-1,878.120	-114,682.660	GC_ALLO1	Demo GC Allocation 1	Prorata

Click the allocation rule link displayed in the Allocation Rule ID column to access the Basis page and view basis rule details, such as adjustments, constraints, or if it is a fixed basis.

Allocation Manager Inquiry Results - Drill to Source Page

Use the Allocation Manager Inquiry Results - Drill to Source page (PF_AL_INQUIRY4) to view source rule details for an allocation.

Navigation

Click the Drill to Source link on the Drill Details tab of the Allocation Manager Inquiry Results page

Image: Allocation Manager Inquiry Results - Drill to Source page

This example illustrates the fields and controls on the Allocation Manager Inquiry Results - Drill to Source page. You can find definitions for the fields and controls later on this page.

Allocation Manager Inquiry

Main Criteria					
Business Unit:	10000	World Wide Consolidation	Scenario ID:	GCACTTB	Actual Trial Balance
Fiscal Year:	2005		Period:	1	
Jobstream ID:	GC_ALLO	GC Allocation			

Go To: [Selection Page](#) -> [Target Details](#)

Source			
Customize Find 1 of 1			
Details Rule Detail [F5]			
PF Batch ID	Posted Total Amount	Account	Posted Transaction Amount
GC_ALL4039	1,635.600	620000	1,635.60000000

This page displays the source details associated with the allocation you selected for audit. Some of the columns displayed on this page are Posted Total Amount and Posted Transaction Amount. If an adjustment was applied to the source (and source adjustment is not equal 1) an additional column called Adjustment Applied will appear in the grid and display the adjustment factor that was applied.

Click the Selection Page link to return to the Allocation Manager Inquiry - Audit Parameters page.

Click the Target Details link to return to the Allocation Manager Inquiry Results page.

Rule Detail Tab

Click the Rule Detail tab to view the source rule summary for an allocation.

Image: Rule Detail tab on the Drill to Source page

This example illustrates the fields and controls on the Rule Detail tab on the Drill to Source page. You can find definitions for the fields and controls later on this page.

PF Batch ID	Posted Total Amount	Allocation Rule ID	Description	Allocation Method
GC_ALL4039	1,635.600	GC_ALLO1	Demo GC Allocation 1	Prorata

Click the link displayed in the Allocation Rule ID column to access the Source page and view source rule details, such as adjustments, constraints, or if it is a fixed source.

Drill to Source Page for an Activity Based Management Allocation

The Allocation Manager Inquiry Results - Drill to Source page changes slightly in appearance and functionality when you are working with an ABM allocation.

Image: Allocation Manager Inquiry Results - Drill to Source page for an ABM allocation

This example illustrates the fields and controls on the Allocation Manager Inquiry Results - Drill to Source page for an ABM allocation. You can find definitions for the fields and controls later on this page.

Drill to ABM	PF Batch ID	Source Amount	Object ID	Object Type	Actuals Amount	Budgeted Amount	Calculated Amount
Drill to ABM	ABM_COPY32	133,670.707	CHECKING	Cost Object	133,670.707053	217,721.212084	133,670.707052
Drill to ABM	ABM_COPY32	37,404.444	MONEY_MARKET	Cost Object	37,404.444422	59,573.333296	37,404.444422
Drill to ABM	ABM_COPY32	46,080.000	MUTUAL_FUND	Cost Object	46,080.000000	66,720.000000	46,080.000000
Drill to ABM	ABM_COPY32	21,022.222	PHONE	Cost Object	21,022.222211	35,546.666648	21,022.222211
Drill to ABM	ABM_COPY32	50,084.444	SAVING	Cost Object	50,084.444402	80,693.333256	50,084.444402

This page displays the source details associated with the ABM allocation you selected for audit. Some of the ABM-specific columns displayed on this page are Object ID, Object Type and Budgeted Amount.

Clicking the link displayed in the Drill to ABM column transfers you to the Object Navigator Inquiry tool found in ABM. Using the ABM tool you can further drill down to the model with its resources, activities, and cost objects, as well as view their sources and targets.

See "Using Object Navigator and Model Analyzer (*PeopleSoft EPM 9.1: Activity-Based Management*)".

Creating XBRL Instance Documents in EPM

Understanding XBRL Reporting Using PeopleSoft EPM

These topics discuss XBRL reporting functionality, which enables you to meet the XBRL financial reporting requirements and create XBRL Instance documents per the XBRL Specifications 2.1, and includes:

- Pages that enable you to define the components for an XBRL Instance document.
- Pages that enable you to set up and define taxonomy elements and the required sections of an XBRL Instance document.
- An import utility to upload taxonomy elements and financial report facts.
- An application engine process that generates the XBRL Instance report.

This document discusses how to set up the new XBRL Reporting functionality and create XBRL instance documents for submission to a specified reporting entity. Examples used are for US GAAP.

Note: The functionality deals only with XBRL encoding. This means that you may have generated your financial reports in any source system, (PeopleSoft General Ledger or PeopleSoft Global Consolidations, for example) or using any reporting tool (PS nVision, for example) or any other third party tool.

XBRL Reporting Requirements and Reference Resources

In May 2008, the United States Security Exchange Commission (SEC) issued rules to adopt eXtensible Business Reporting Language (XBRL), the financial reporting version of XML, as the standard to meet financial reporting requirements.

The SEC's proposed schedule would require companies using U.S. Generally Accepted Accounting Principles (US GAAP) with a worldwide public float over \$5 billion to make financial disclosures using interactive data formatted in eXtensible Business Reporting Language (XBRL) for fiscal periods ending in late 2008. If adopted, the first interactive data provided under the new rules would be made public in early 2009. The remaining companies using U.S. GAAP would provide this disclosure over the following two years. Companies using International Financial Reporting Standards (IFRS) would provide this disclosure for fiscal periods ending in late 2010. The disclosure would be provided as additional exhibits to annual and quarterly reports and registration statements. Companies also would be required to post this information on their websites.

The required tagged disclosures would include companies' primary financial statements, notes, and financial statement schedules. Initially, companies would tag notes and schedules as blocks of text, and later, they would provide tags for the details within the notes and schedules.

The following table lists references and resources for more information about the SEC rulings and XBRL requirements.

Documents	Location
XBRL Specifications 2.1	http://www.xbrl.org/Specification/XBRL-RECOMMENDATION-2003-12-31+Corrected-Errata-2008-07-02.htm/
Financial Reporting Instance Standards 1.0	http://www.xbrl.org/technical/guidance/FRIS-PWD-2004-11-14.htm
Tutorial and in-depth technical training on XBRL	http://www.xbrl.org/EducationAndTraining/
Education and Training	http://www.xbrl.org/EducationAndTraining/
XBRL US GAAP Taxonomy 1.0	http://www.xbrl.org/FRTaxonomies/
XBRL US GAAP Taxonomy Preparers Guide	http://xbrl.us/Documents/PreparersGuide.pdf

Note: This documentation assumes that you are familiar with the XBRL instance document requirements; it does not describe the XBRL specifications, reporting standards, or taxonomy structure. Please refer to the reference materials in the previous table for detailed information about XBRL.

Using the XBRL Reporting Features

The following list outlines the typical activities that you'll complete to generate XBRL financial statements:

1. Prepare Financial Statements and map elements to required taxonomy.

These steps are completed outside of the PeopleSoft EPM application, using financial applications, spreadsheets, and the XBRL websites to prepare and tag the data. Typically this phase includes the following tasks:

- a. Close the accounting period.
- b. Generate, review, and finalize financial reports (which includes formatting & rounding of amounts).
- c. Map the line items of your source financial reports to GAAP Taxonomy elements, using a spreadsheet program or similar tool, and using applicable XBRL taxonomy websites.

The number of elements that are applicable to your organization is likely to be a small subset of all taxonomy elements. For example, there are 12,000 elements defined for US GAAP. A US GAAP user will access the US XBRL website and choose their industry entry point to review and copy the taxonomy elements (and some of their attributes) applicable to their reporting.

Note: The US XBRL website provides a taxonomy viewer to search and look up taxonomy elements for financial report line items and standard footnotes.

2. Set up XBRL Definitions in EPM. This information is discussed in the [Setting Up XBRL Definitions in EPM](#) topic.
3. Upload XBRL taxonomy and financial element facts, and create footnotes.

These steps are discussed in the [Uploading and Maintaining XBRL Facts and Taxonomy Elements](#) topic.

4. Create an XBRL instance report.

This step is discussed in the [Creating XBRL Instance Documents](#) topic.

Setting Up XBRL Definitions in EPM

This topic provides an overview of XBRL instance document components, defines pages used, and discusses how to:

- Define a namespace set
- Define taxonomy
- Define an Instance Header
- Define a reporting entity
- Define an instance report.
- Define a report set.
- Define XBRL units of measure.
- Define context.

Pages Used to Set up Reporting Definitions

<i>Page Name</i>	<i>Definition Name</i>	<i>Navigation</i>	<i>Usage</i>
Define Namespace Set	XBRL_NS_SET	EPM Foundation, XBRL Reporting, Define XBRL, Taxonomy, Define Namespace Set	Define XML namespaces. These will be declared in the root element of the Instance. Some namespaces are specific to XML and some to a specific taxonomy. The sets are re-usable and are referenced in the Instance template
Define Taxonomy	XBRL_TAXONOMY	EPM Foundation, XBRL Reporting, Define XBRL, Taxonomy, Define Taxonomy	Define taxonomy attributes, including its name space and the physical location of its Schema. These attributes are used in the Instance document.

Page Name	Definition Name	Navigation	Usage
Define Instance Header	XBRL_INST_HDR	EPM Foundation, XBRL Reporting, Define XBRL, Taxonomy, Define Instance Header	Define an XBRL instance document header.
Define Reporting Entity	XBRL_ENTITY	EPM Foundation, XBRL Reporting, Define Reporting, Define Reporting Entity	Define a reporting entity.
Define Instance Report	XBRL_RPT_DEFN	EPM Foundation, XBRL Reporting, Define Reporting, Define Instance Report	Define a report ID.
Define Report Set	XBRL_RPT_SET	EPM Foundation, XBRL Reporting, Define Reporting, Define Report Set	Create Report Sets, which are a grouping of report IDs.
Define XBRL Unit of Measure	XBRL_UOM	EPM Foundation, XBRL Reporting, Define Reporting, Define XBRL Units of Measure	Create Unit of Measure (UOM) definitions for XBRL reporting.
Define Context	XBRL_CONTEXT	EPM Foundation, XBRL Reporting, Define Reporting, Define Context	Define context IDs.

Understanding XBRL Instance Document Components

XBRL instance documents contain facts reported by a specific entity, at a specific point in time, in a specific currency using GAAP guidelines like US GAAP, IFRS, and so on.

A GAAP taxonomy document contains definitions for the accounting concepts and their relationships for a specific GAAP like US GAAP. You can visualize it as GL account definitions and its reporting tree for US GAAP. Technically, it includes an xml schema definition and other supporting xml files.

There can be several instance documents for a taxonomy document. For example, you can have two balance sheet instances (reports) for a legal reporting entity – one for year 2009 and another for year 2010. Or you could have balance sheet instances for a parent corporation and all of its subsidiaries.

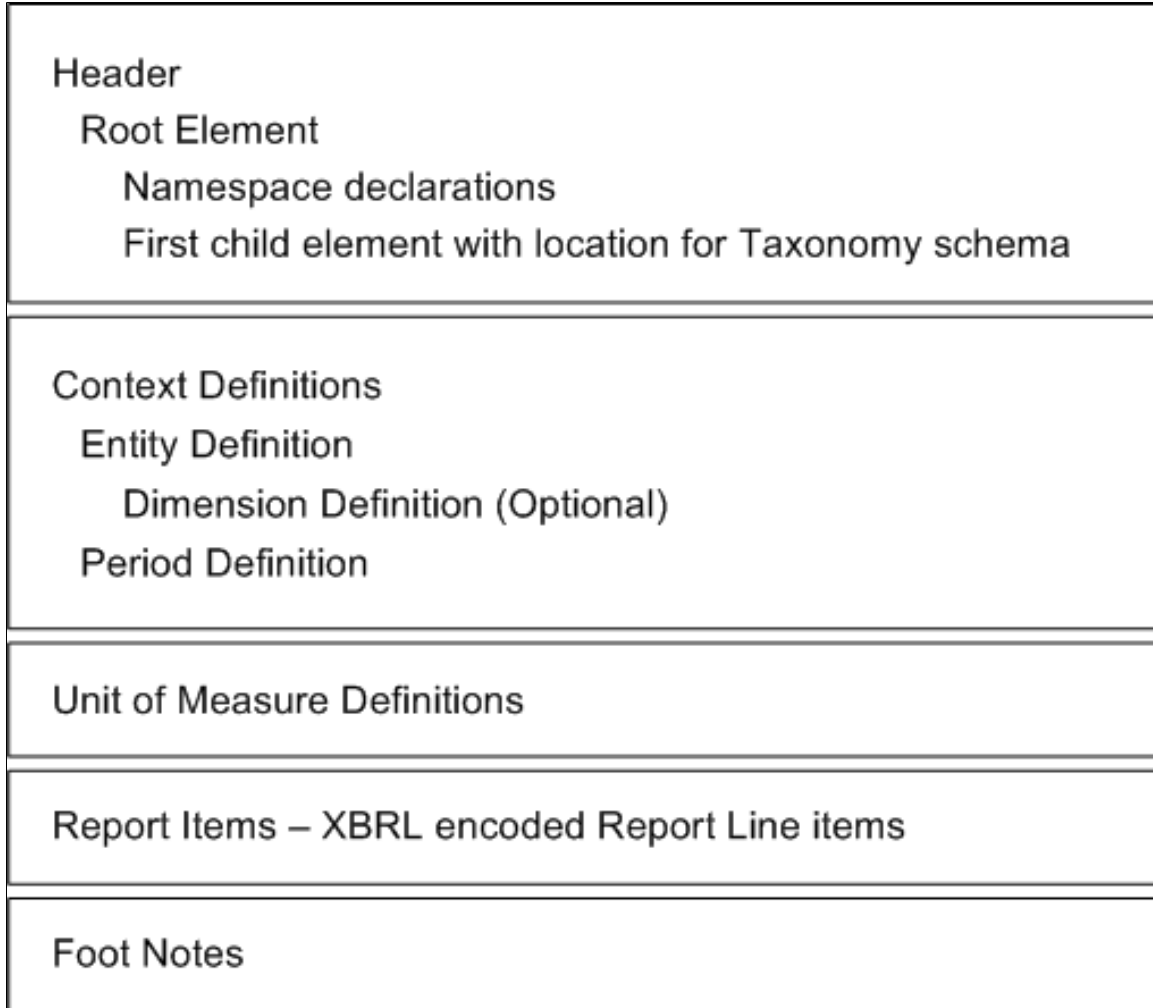
XBRL instance documents are XML documents and need to be well-formed (as defined by the XML schema) and valid (as defined by the taxonomy schema).

XBRL Shell and Sample Structure

The basic "shell" of an XBRL instance document includes the following structure:

Image: XBRL Shell and Sample Structure

This diagram illustrates the structure of the basic shell of an XBRL instance document.



The following table provides an example of an XBRL instance. Table breaks separate each "shell" section.

XBRL Tag	Description
<?xml version="1.0"?>	<p>All XML documents start with a prolog. This is an XML Identifier. This line is the required beginning line for every XML document, regardless of its type. Additional data can be inserted but this is the minimum required information.</p> <p>This line indicates that it is an XML document based on XML Recommendation 1.0. As a minimum, it must be included exactly as shown.</p>

XBRL Tag	Description
<xbrl	<p>The root element is <xbrl></p> <p>This is followed by namespace declarations for xml and for the taxonomy referenced in the instance document.</p>
xmlns="http://www.xbrl.org/2003/instance"	<p>This is the namespace for all XBRL instance documents. You can access the schema using this URL.</p> <p>(The schema defines the structure and format of their instances).</p>
xmlns:link="http://www.xbrl.org/2003/linkbase"	<p>This identifies the location of XBRL Link bases - Anything referring to the role of linkbases within XBRL is defined here and any reference to it will be prefixed with the alias, "link:"</p>
xmlns:xlink="http://www.w3.org/1999/xlink"	<p>This is the location of BASE xml link bases. It refers to the roles of link bases specified within the XLink specifications.</p>
xmlns:us-gaap="http://xbrl.us/us-gaap/2009-01-31" >	<p>This is the US GAAP Taxonomy Namespace Reference</p>
<link:schemaRef xlink:type="simple" xlink:href=" http://taxonomies.xbrl.us/us-gaap/2009/elts/us-gaap-std-2009-01-31.xsd"/>	<p>In an XBRL instance, the schemaRef element points to a taxonomy schema.</p> <p>Every XBRL instance must contain at least one schemaRef element.</p> <p>The schemaRef element must occur as a child element of an xbrl element and must occur before other children of the xbrl root element, in document order.</p>
<Context id="Asof_June30" >	<p>The context element contains information about the entity being described and the reporting period, all of which are necessary for understanding a business fact captured as an XBRL item.</p> <p>The context element has an attribute, ID = "Asof_June30" in this example</p>
<entity>	<p>The entity element.</p>
<identifier scheme="http://www.sec.gov/">1234567890</identifier>	<p>The entity is identified using 2 attributes:</p> <ul style="list-style-type: none"> • Scheme – the URL for SEC • ID – the 10 digit central index key (CIK) assigned by SEC <p>(The entity may also be identified using a URL for the stock exchange and the ticker symbol of the company).</p>

</entity>	Closing tag.
<period>	The period element.
<instant>2008-06-30</instant>	Period type is instant and is 'As of date', June 30, 2008.
</period>	Closing tag.
</Context>	Closing tag.
<unit ID="usd"	Unit Element with a ID attribute.
<measure> ISO4217:USD </measure>	Measure is the sub-element. The unit of measure is US dollars and takes its meaning from ISO4217.
</unit>	Closing tag.
<us-gaap:AssetsCurrent decimal="INF" unitRef="usd" contextRef=" Asof_June30">727</us-gaap:AssetsCurrent >	<p>The element details are comprised of the following:</p> <ul style="list-style-type: none"> us-gaap – the prefix(alias) refers to the US GAAP Taxonomy decimal="INF" – The fact value is exact (not rounded) AssetsCurrent – the element name <p>The element has 2 reference attributes:</p> <ul style="list-style-type: none"> unitRef="usd" – refers to the unit Id defined above contextRef=" Asof_June30" – refers to context Id defined above <p>727 – is the fact value for the element</p>
< us-gaap:liabilities precision="3" unitRef="usd" contextRef=" Asof_June30">635</ci:liabilities>	The element details for US GAAP liabilities.
</xbrl>	<p>End of instance document.</p> <p>Closing tag for the root element</p>

Define Namespace Set Page

Use the Define Namespace Set page (XBRL_NS_SET) to define XML namespaces. These will be declared in the root element of the Instance. Some namespaces are specific to XML and some to a specific taxonomy. The sets are re-usable and are referenced in the Instance template

Navigation

EPM Foundation, XBRL Reporting, Define XBRL, Taxonomy, Define Namespace Set

Image: Define Namespace Set page

This example illustrates the fields and controls on the Define Namespace Set page. You can find definitions for the fields and controls later on this page.

Use this page to define the namespaces to include in a namespace set. Namespaces are declared in the root element of the Instance document. Some namespaces are specific to XML and some to a specific taxonomy. You will reference a namespace set when you define an instance header, and when you define taxonomy, so typically, you will define at least two namespace sets.

Description Enter the description for the namespace set, for example, US GAAP Taxonomy.

Namespace Group Box

Complete the following fields for each namespace that you add to the Namespace grid.

Sequence Enter a value for each namespace within the namespace set. This must be unique; it functions as an identifier for each namespace.

Prefix Enter a shorthand sequence of letters for a namespace; us-gaap, for example, is a recommended prefix for the namespace http://xbrl.us/us-gaap/2008-01-31.

Namespace Enter the Universal Resource Identifier (URI) that identifies the namespace.

Add Button Click the Add button to specify additional namespaces within this namespace set.

Define Taxonomy Page

Use the Define Taxonomy page (XBRL_TAXONOMY) to define taxonomy attributes, including its name space and the physical location of its Schema. These attributes are used in the Instance document.

Navigation

EPM Foundation, XBRL Reporting, Define XBRL, Taxonomy, Define Taxonomy

Image: Define Taxonomy page

This example illustrates the fields and controls on the Define Taxonomy page. You can find definitions for the fields and controls later on this page.

Define Taxonomy

Taxonomy ID: USGAAP

Effective Date: 01/01/1900

*Effective Date: *Status: Active

*Description:

*Namespace Set: [Edit/View Namespace Set](#)

*Schema Location:

Taxonomy Home Page:

[Taxonomy Home Page](#)

Use this page to define a taxonomy. You reference the Taxonomy ID when you define an instance header.

Description

Enter a description for the taxonomy.

Namespace Set

Select the namespace set that is associated with this taxonomy.

Click the Edit/View Namespace Set link to access the Define Namespace Set page, where you can edit or view the namespace set.

Schema Location

Enter the Universal Resource Identifier (URI) that identifies the physical location of the taxonomy schema.

Taxonomy Home Page

Enter the URL for the taxonomy homepage.

Taxonomy Home Page link

Click to view the home page for the specified taxonomy.

Define Instance Header Page

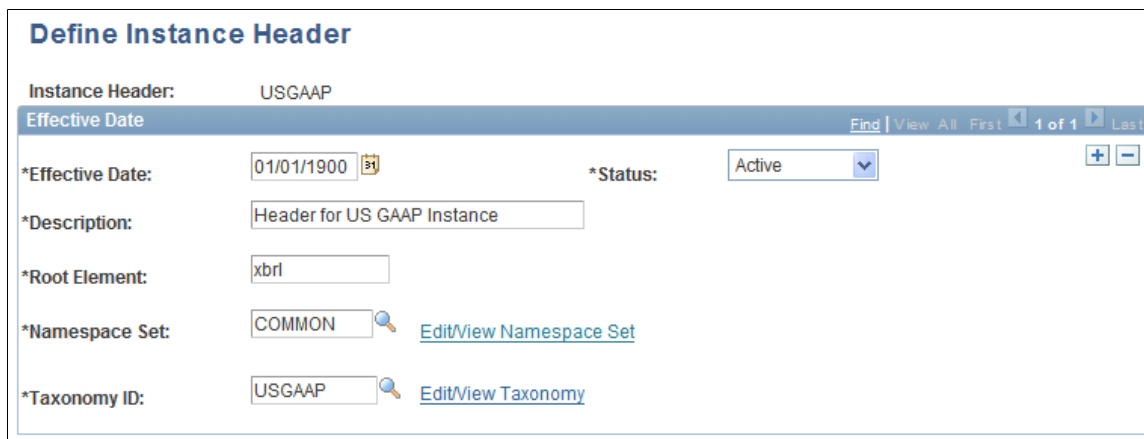
Use the Define Instance Header page (XBRL_INST_HDR) to define an XBRL instance document header.

Navigation

EPM Foundation, XBRL Reporting, Define XBRL, Taxonomy, Define Instance Header

Image: Define Instance Header page

This example illustrates the fields and controls on the Define Instance Header page. You can find definitions for the fields and controls later on this page.



This page captures the following attributes of an XBRL instance:

- The root element (currently "xbrl").
- The namespace set
- The taxonomy ID.

You reference an instance header when you create an XBRL instance.

Instance Header and Description	Enter a unique name for the XBRL instance header, and its description.
Root Element	Specify the root element for the instance document.
Namespace Set	Specify the namespace set to associate with the instance document. You establish namespace sets using the Define Namespace Set page.
Edit/View Namespace Set	Click to access the Define Namespace Set page for this namespace, where you can review the Namespace Set definition.
Taxonomy ID	Specify the taxonomy to associate with this instance header. You establish taxonomy IDs using the Define Taxonomy page.
Edit/View Taxonomy	Click to access the Define Taxonomy page for the specified taxonomy, where you can review or edit the taxonomy definition.

Define Reporting Entity Page

Use the Define Reporting Entity page (XBRL_ENTITY) to define a reporting entity.

Navigation

EPM Foundation, XBRL Reporting, Define Reporting, Define Reporting Entity

Image: Define Reporting Entity page

This example illustrates the fields and controls on the Define Reporting Entity page. You can find definitions for the fields and controls later on this page.

Define Reporting Entity

Entity Name: ABC_CORP_SEC

Effective Date: 01/01/1900

*Effective Date: 01/01/1900

*Status: Active

*XBRL Scheme: http://www.sec.gov

Specify URL of the organization being reported to. For Example, US companies reporting to SEC will have the value, www.sec.gov/ If reporting to Nasdaq stock exchange, the scheme url would be www.nasdaq.com/

*Entity Id: 1234567890

Specify reporting entity ID with respect to the scheme url provided. For Example, US companies reporting to SEC will use their 10 digit CIK code. If reporting to a stock exchange (example, Nasdaq), use the entity's stock ticker symbol (example, ORCL)

Entity Name

Enter a unique name for the reporting entity.

XBRL Scheme

Enter the URL of the organization you are reporting to.

Entity ID

Enter your organization's Entity ID for the organization to which you are reporting.

Define Instance Report Page

Use the Define Instance Report page (XBRL_RPT_DEFN) to define a report ID.

Navigation

EPM Foundation, XBRL Reporting, Define Reporting, Define Instance Report

Image: Define Instance Report page

This example illustrates the fields and controls on the Define Instance Report page. You can find definitions for the fields and controls later on this page.

Define Report

Report ID: DISC-DR

Effective Date: 01/01/1900 Find | View All | First | 1 of 1 | Last

*Effective Date: *Status:

*Description:

Report ID Enter a unique identifier for the instance report.

Description Enter the description to use for the instance report.

The system creates an instance document for either a single report ID or a report set.

Define Report Set Page

Use the Define Report Set page (XBRL_RPT_SET) to create Report Sets, which are a grouping of report IDs.

Navigation

EPM Foundation, XBRL Reporting, Define Reporting, Define Report Set

Image: Define Report Set page

This example illustrates the fields and controls on the Define Report Set page. You can find definitions for the fields and controls later on this page.

Define Report Set

Report Set: FIN_RPTS

Effective Date: 01/01/1900 Find | View All | First | 1 of 1 | Last

*Effective Date: *Status:

*Description:

Instance Reports			Customize Find <input type="button" value="B1"/> <input type="button" value="B2"/> First 1-5 of 5 Last	
*Report ID	Description	Edit/View Report Definition		
BAL_SHT	Balance Sheet	Edit/View Report Definition	<input type="button" value="+"/>	<input type="button" value="-"/>
CASHFLOW	Cash Flow	Edit/View Report Definition	<input type="button" value="+"/>	<input type="button" value="-"/>
DISC-DR	Disclosures - Deferred Revenue	Edit/View Report Definition	<input type="button" value="+"/>	<input type="button" value="-"/>
EPS	Earnings per share	Edit/View Report Definition	<input type="button" value="+"/>	<input type="button" value="-"/>
INCOME	Statement of Income	Edit/View Report Definition	<input type="button" value="+"/>	<input type="button" value="-"/>

Report Set	Enter a unique name for the report set.
Description	Enter a description for the report set.

Instance Reports Group Box

Specify the report IDs to include in this report set. You can add additional rows to include more than one report ID in the report set.

Report ID	Specify the report ID to include. You establish report IDs using the Define Instance Report page.
Edit/View Report Definition	Click this link to access the Define Instance Report page, where you can review or modify the report definition.

The system creates an instance document for either a single report ID or a report set.

Define XBRL Unit of Measure Page

Use the Define XBRL Unit of Measure page (XBRL_UOM) to create Unit of Measure (UOM) definitions for XBRL reporting.

Navigation

EPM Foundation, XBRL Reporting, Define Reporting, Define XBRL Units of Measure

Image: Define XBRL Unit of Measure page

This example illustrates the fields and controls on the Define XBRL Unit of Measure page. You can find definitions for the fields and controls later on this page.

Unit ID Enter a unique identifier for the XBRL unit of measure. The unit of measure IDs are referenced in the instance facts.

Description Enter a description for the unit of measure.

Type Specify the unit of measure type, either *Simple* or *Fraction*.

Currency is an example of a Simple type of unit of measure; earnings per share is an example of a Fraction type of unit of measure.

For a fraction type of unit of measure, enter the values for Numerator and Denominator. For a simple type of unit of measure, select *Simple* and enter the value for XBRL Unit.

Define Context Page

Use the Define Context page (XBRL_CONTEXT) to define context IDs.

Navigation

EPM Foundation, XBRL Reporting, Define Reporting, Define Context

Image: Define Context page

This example illustrates the fields and controls on the Define Context page. You can find definitions for the fields and controls later on this page.

The screenshot shows a web form titled "Define Context". At the top, the "Context ID" is set to "FISCAL_YEAR_2008". Below this is a section header "Context". The form contains the following fields and controls:

- *Description:** A text input field containing "Fiscal year, 2008".
- *Period Type:** A dropdown menu currently set to "Duration".
- From Date:** A date input field containing "01/01/2008" with a calendar icon.
- To Date:** A date input field containing "12/31/2008" with a calendar icon.

For each context you define, complete the following fields.

Context ID Enter the context identifier. PeopleSoft recommends using something that is easily interpreted, such as DefRev_Jun30_2008.

Description Enter a description for the context.

Period Type Specify the timeframe for the context.

Select *Instant* to define a specific point in time, and select the date in the As of Date field. This option would apply to a balance sheet report, for example.

Select *Duration* to define a date range, and select the starting and ending dates in the From Date and To Date fields. This option would apply to income and cash flow reports, for example.

The context ID within an instance is for a unique combination of entity, segment and time dimensions. Context IDs are system generated for each segment (dimension member) value, as shown in the following example.

Report Type	Line Item	Entity	Segment	Period	Amount	Context ID
Balance sheet	Disposals	ABC Corp		As of Dec31, 2008	100,000	AsofDec31
Disclosure - Plant Property & Equipment	Additions	ABC Corp	Building	As of Dec31, 2008	15,000	AsofDec31_Building

Report Type	Line Item	Entity	Segment	Period	Amount	Context ID
Disclosure - Plant Property & Equipment	Disposals	ABC Corp	Building	As of Dec31, 2008	20,000	AsofDec31_Building
Disclosure - Plant Property & Equipment	Disposals	ABC Corp	Land	As of Dec31, 2008	23,000	AsofDec31_Land

Uploading and Maintaining XBRL Facts and Taxonomy Elements

This topic provides an overview of data upload requirements and discusses how to:

- Review file definitions.
- Upload instance facts and taxonomy elements.
- Maintain taxonomy elements.
- Maintain footnotes.
- Maintain instance facts.

Pages Used to Upload Facts and Taxonomy Elements

Page Name	Definition Name	Navigation	Usage
Define File Layout	XBRL_FILE_DFN	EPM Foundation, XBRL Reporting, Import Data, Define File Layout	Review the delivered file definitions and file layouts.
Load Data	XBRL_DATA_LD	EPM Foundation, XBRL Reporting, Import Data, Load Data	Upload taxonomy elements or instance facts.
Maintain Taxonomy Elements	XBRL_ELEMENTS	EPM Foundation, XBRL Reporting, Maintain Data, Maintain Taxonomy Elements	Review or update taxonomy elements that you have uploaded.
Maintain Footnotes	XBRL_FOOTNOTES	EPM Foundation, XBRL Reporting, Maintain Data, Maintain Footnotes	Review or update footnotes.
Maintain Footnotes – Fact References	XBRL_FNOTES_REF	EPM Foundation, XBRL Reporting, Maintain Data, Maintain Footnotes, Fact References	Review or update footnote fact references.
Maintain Instance Facts	XBRL_INST_FACT	EPM Foundation, XBRL Reporting, Maintain Data, Maintain Instance Facts	Review or update instance facts that you have uploaded.

Understanding Data Upload Requirements

We provide a generic data upload utility, the Load Data Application Engine process, to import your mapped taxonomy elements, financial instance facts, and disclosure facts into the system. File layouts are delivered for comma separated value (CSV) and tab separated value (TSV) text files. Typically, you will prepare the data using a spreadsheet program and save it as a CSV or TSV file. The following sections describe the required file layouts.

File Layout for Taxonomy Elements

The following table lists the required format for the file that contains the mapped taxonomy elements.

<i>Field Nbr</i>	<i>Field Name</i>	<i>Long Name</i>	<i>Field Type</i>	<i>Format</i>	<i>Length</i>
1	PS_ELEMENT	Element Id in PSFT	CHAR		30
2	EFFDT	Effective Date	DATE	YYYY/MM/DD	10
3	TAXONOMY_ID	Taxonomy ID	CHAR		10
4	NAMESPACE_ALIAS	Namespace Alias	CHAR		10
5	XBRL_ELEMENT_NAME	Element Name	CHAR		254
6	EFF_STATUS	Status as of Effective Date	CHAR		1
7	XBRL_ELEMENT_LBL	XBRL Element Label	CHAR		254
8	DRCR	XBRL Balance	CHAR		15
9	XBRL_DATATYPE	Data Type	CHAR	NUM/TXT	3

The following example shows a spreadsheet that contains taxonomy elements using the required format. PS Element is the user-defined ID you will use in PeopleSoft EPM for the element.

Image: Taxonomy Elements

This example illustrates the Taxonomy Elements.

PS Element	Effective Date	Status	Taxonomy Id	Namespace Alias	XBRL Element ID	XBRL Element Label	Balance	Data Type
PROVISION	2009/01/01	A	USGAAP	us-gaap	ProvisionForDoubtfulAccounts	Provision for Doubtful Accounts	debit	NUM
AR_CURR	2009/01/01	A	USGAAP	us-gaap	ReceivablesNetCurrent	Receivables, Net, Current	Debit	NUM
DEBT_LT_PAY	2009/01/01	A	USGAAP	us-gaap	RepaymentsOfLongTermDebt	Repayments of Long-term Debt	credit	NUM
EXP_R&D	2009/01/01	A	USGAAP	us-gaap	ResearchAndDevelopmentExpense	Research and Development Expense	Debit	NUM
RETAINED_EARNINGS	2009/01/01	A	USGAAP	us-gaap	RetainedEarningsAccumulatedDeficit	Retained Earnings (Accumulated Deficit)	Credit	NUM
REV_SALES	2009/01/01	A	USGAAP	us-gaap	SalesRevenueGoodsGross	Sales Revenue, Goods, Gross	Credit	NUM
REV_SERVICES	2009/01/01	A	USGAAP	us-gaap	SalesRevenueServicesNet	Sales Revenue, Services, Net	Credit	NUM
INVEST_SHORT	2009/01/01	A	USGAAP	us-gaap	ShortTermInvestments	Short-term Investments	Debit	NUM
REV_TECH	2009/01/01	A	USGAAP	us-gaap	TechnologyServicesRevenue	Technology Services Revenue	Credit	NUM
SEG_SUB_ARRANGE	2009/01/01	A	USGAAP	us-gaap	SubscriptionArrangementMember	Subscription Arrangement	none	TXT
T_REVREC_GP	2009/01/01	A	USGAAP	us-gaap	RevenueRecognitionGeneralPrinciples	Revenue Recognition, General Principles	none	TXT
T_REVREC_LP	2009/01/01	A	USGAAP	us-gaap	RevenueRecognitionLoyaltyPrograms	Revenue Recognition, Loyalty Programs	none	TXT
DIM_REV_DEF_ARRANGE	2009/01/01	A	USGAAP	us-gaap	DeferredRevenueArrangementTypeAxis	Deferred Revenue Arrangement Type [Axis]	none	TXT
SEG_LAYAWAY_SALE	2009/01/01	A	USGAAP	us-gaap	LayawaySaleMember	Layaway Sale	none	TXT

Note: The date field format is YYYY/MM/DD. For example, the date Jul 01, 2009 should be formatted as 2009/07/01.

File Layout for Financial Report Facts

The following table lists the required format for the file that contains the element facts for financial reports.

Field Nbr	Field Name	Edit Table	Long Name	Field Type	Length
1	TAXONOMY_ID	XBRL_TAXONOMY	Taxonomy ID	CHAR	10
2	ENTITY_NAME	XBRL_ENTITY	Entity Name	CHAR	50
3	REPORT_ID	XBRL_RPT_DEFN	Report ID	CHAR	8
4	XBRL_CONTEXT_ID	XBRL_CONTEXT	XBRL Context Identifier	CHAR	30
5	LINE_NBR		Line Number	NUM	5
6	PS_ELEMENT	XBRL_ELEMENTS	Element ID in PSFT	CHAR	30
7	XBRL_UOM_ID	XBRL_UOM	XBRL Unit of Measure ID	CHAR	30
8	XBRL_DECIMALS		Rounded to Decimals	CHAR	20
9	XBRL_AMOUNT		Amount	CHAR	30

Field Nbr	Field Name	Edit Table	Long Name	Field Type	Length
10	XBRL_ FOOTNOTE_ID	XBRL_ FOOTNOTES	Foot Note Id	CHAR	30

The following example shows a spreadsheet that contains element facts for financial reports using the required format. PS Element is the user-defined ID you will use in PeopleSoft EPM for the element.

Image: Element Facts for Financial Reports

This example illustrates the Element Facts for Financial Reports.

Taxonomy Id	Entity Name	Report	Context Id	Line#	PS Element	UOM ID	Decimals	Amount	Footnote ID
USGAAP	ABC_CORP_SEC	BAL_SHT	AS_OF_JUNE30_2008	1	CASH	USD	-6	20000000	
USGAAP	ABC_CORP_SEC	BAL_SHT	AS_OF_JUNE30_2008	2	INVEST_SHORT	USD	-6	10000000	
USGAAP	ABC_CORP_SEC	BAL_SHT	AS_OF_JUNE30_2008	3	AR_CURR	USD	-6	5000000	
USGAAP	ABC_CORP_SEC	BAL_SHT	AS_OF_JUNE30_2008	4	EXP_PREPAID	USD	-6	3000000	
USGAAP	ABC_CORP_SEC	BAL_SHT	AS_OF_JUNE30_2008	5	ASSETS_OTHER_CURR	USD	-6	15000000	
USGAAP	ABC_CORP_SEC	BAL_SHT	AS_OF_JUNE30_2008	6	INVENTORY_NET	USD	-6	4000000	
USGAAP	ABC_CORP_SEC	BAL_SHT	AS_OF_JUNE30_2008	7	ASSETS_CURR	USD	-6	30000000	
USGAAP	ABC_CORP_SEC	BAL_SHT	AS_OF_JUNE30_2008	8	PPE_NET	USD	-6	8000000	PPE_JUN30_2008
USGAAP	ABC_CORP_SEC	BAL_SHT	AS_OF_JUNE30_2008	9	DEBT_CURR	USD	-6	4000000	
USGAAP	ABC_CORP_SEC	BAL_SHT	AS_OF_JUNE30_2008	10	AP_CURR	USD	-6	2000000	
USGAAP	ABC_CORP_SEC	BAL_SHT	AS_OF_JUNE30_2008	11	COMMON_STOCK	USD	-6	40000000	
USGAAP	ABC_CORP_SEC	BAL_SHT	AS_OF_JUNE30_2008	12	CAPITAL_OTHER	USD	-6	5000000	
USGAAP	ABC_CORP_SEC	BAL_SHT	AS_OF_JUNE30_2008	13	RETAINED_EARNINGS	USD	-6	9000000	
USGAAP	ABC_CORP_SEC	BAL_SHT	AS_OF_JUNE30_2008	14	T_REVREC_GP	USD	-6	000000	STD_REVREC_GP
USGAAP	ABC_CORP_SEC	BAL_SHT	AS_OF_JUNE30_2008	15	T_REVREC_LP	USD	-6	000000	STD_REVREC_LP
USGAAP	ABC_CORP_SEC	INCOME	FISCAL_YEAR_2008	1	INCOME_OPERATING	USD	-6	5000000	
USGAAP	ABC_CORP_SEC	INCOME	FISCAL_YEAR_2008	2	INCOME_OTHER	USD	-6	4000000	
USGAAP	ABC_CORP_SEC	INCOME	FISCAL_YEAR_2008	3	REV_SALES	USD	-6	20000000	
USGAAP	ABC_CORP_SEC	INCOME	FISCAL_YEAR_2008	4	REV_SERVICES	USD	-6	10000000	
USGAAP	ABC_CORP_SEC	INCOME	FISCAL_YEAR_2008	5	REV_TECH	USD	-6	3000000	
USGAAP	ABC_CORP_SEC	INCOME	FISCAL_YEAR_2008	6	REV_LICENSE	USD	-6	6000000	
USGAAP	ABC_CORP_SEC	INCOME	FISCAL_YEAR_2008	7	EXP_R&D	USD	-6	1000000	
USGAAP	ABC_CORP_SEC	INCOME	FISCAL_YEAR_2008	8	EXP_DEPR	USD	-6	1000000	
USGAAP	ABC_CORP_SEC	INCOME	FISCAL_YEAR_2008	9	EXP_OTHER	USD	-6	1000000	
USGAAP	ABC_CORP_SEC	INCOME	FISCAL_YEAR_2008	10	EPS_BASIC	EPS	0	50	

The decimal column specifies rounding. If the amount is rounded to thousands, the decimal value is "-3". If the amount is rounded to millions, the decimal value is "-6". The Amount field should be populated as per the XBRL guidance.

You should ensure that these are not comma formatted. For Example, 2 million USD rounded to millions should be represented as:

Decimal = -6 and Amount = 2000000

File Layout for Disclosure Schedule

The disclosure schedule (table) has two additional columns for Dimension and Segment (dimension member). The following table lists the required format for the file that contains the disclosure facts.

Field Nbr	Field Name	Edit Table	Long Name	Field Type	Length
1	TAXONOMY_ID	XBRL_ TAXONOMY	Taxonomy ID	CHAR	10
2	ENTITY_NAME	XBRL_ENTITY	Entity Name	CHAR	50

Field Nbr	Field Name	Edit Table	Long Name	Field Type	Length
3	REPORT_ID	XBRL_RPT_DEFN	Report ID	CHAR	8
4	XBRL_CONTEXT_ID	XBRL_CONTEXT	XBRL Context Identifier	CHAR	30
5	LINE_NBR		Line Number	NUM	5
6	PS_ELEMENT	XBRL_ELEMENTS	Element ID in PSFT	CHAR	30
7	XBRL_DIMENSION	XBRL_ELEMENTS	Dimension	CHAR	30
8	XBRL_SEGMENT	XBRL_ELEMENTS	XBRL Segment	CHAR	100
9	XBRL_UOM_ID	XBRL_UOM	XBRL Unit of Measure ID	CHAR	30
10	XBRL_DECIMALS		Rounded to Decimals	CHAR	20
11	XBRL_AMOUNT		Amount	CHAR	30
12	XBRL_FOOTNOTE_ID	XBRL_FOOTNOTES	Foot Note ID	CHAR	30

The following example shows a spreadsheet that contains disclosure facts using the required format.

Image: Disclosure Facts Formatting

This example illustrates the Disclosure Facts Formatting.

A	B	C	D	E	F	G	H	I	J	K	L
Taxonomy Id	Entity Name	Report	Context Id	Line#	PS Element ID	Dimension	XBRL Segment	UOM ID	Decimals	Amount	Footnote ID
USGAAP	ABC_CORP_SEC	DISC-DR	SALE_JUNE30_2008	1	REV_DEF_CURR	DIM_REV_DEF_ARRANGE	SEG_LAYAWAY_SALE	USD	0	60000	
USGAAP	ABC_CORP_SEC	DISC-DR	SALE_JUNE30_2008	2	REV_DEF_ICURR	DIM_REV_DEF_ARRANGE	SEG_LAYAWAY_SALE	USD	0	45000	
USGAAP	ABC_CORP_SEC	DISC-DR	SALE_JUNE30_2008	3	REV_DEF_CURR	DIM_REV_DEF_ARRANGE	SEG_SUB_ARRANGE	USD	0	50000	

Define File Layout Page

Use the Define File Layout page (XBRL_FILE_DFN) to review the delivered file definitions and file layouts.

Navigation

EPM Foundation, XBRL Reporting, Import Data, Define File Layout

Image: Define File Layout page

This example illustrates the fields and controls on the Define File Layout page. You can find definitions for the fields and controls later on this page.

The screenshot shows a web interface titled "Define File Layout". At the top, it displays "File Definition: ELMTS_CSV". Below this is a section titled "Specify File Layout" with a blue header. Inside this section, there are three rows of input fields:

- *Description: XBRL Elements Layout
- *File Layout: XBRL_ELEMENTS_CSV (with a magnifying glass icon to its right)
- *Target Record: XBRL_ELEMENTS (with a magnifying glass icon to its right)

The following table lists the delivered file definitions. You should not modify these.

<i>File Definition</i>	<i>File Layout</i>	<i>Format</i>	<i>Description</i>
DISCL_CSV	XBRL_DISCL_FACTS_CSV	CSV	Use this file layout to upload disclosure schedule data in CSV format.
DISCL_TSV	XBRL_DISCL_FACTS_TSV	TSV	Use this file layout to upload disclosure schedule data in TSV format.
ELMTS_CSV	XBRL_ELEMENTS_CSV	CSV	Use this file layout to upload taxonomy elements in CSV format.
ELMTS_TSV	XBRL_ELEMENTS_TSV	TSV	Use this file layout to upload taxonomy elements in TSV format.
FACTS_CSV	XBRL_INST_FACT_CSV	CSV	Use this file layout to upload financial instance facts in CSV format.
FACTS_TSV	XBRL_INST_FACT_TSV	TSV	Use this file layout to upload financial instance facts in TSV format.

Load Data Page

Use the Load Data page (XBRL_DATA_LD) to upload taxonomy elements or instance facts.

Navigation

EPM Foundation, XBRL Reporting, Import Data, Load Data

Image: Load Data run control page

This example illustrates the fields and controls on the Load Data run control page. You can find definitions for the fields and controls later on this page.

Load Data

User ID: VP1 [Report Manager](#)

Run Control ID: ELEMENTS [Process Monitor](#)

Process Information

*Description: When:

Attached File:

First Data Line:

*File Definition: [Edit/View File Definition](#)

*Load Option:

File Errors Options:

Attach

Click this button to browse to and attach the file to import.

View

Click this button to view an attached file.

Delete

Click this button to delete an attached file.

First Data Line

Enter the line number where the actual data begins. For example, if the first row of your file contains headings, then you should enter 2, if not, enter 1.

File Definition

Specify the file definition to use for the import. The delivered file definitions that you use are:

- *DISCL_CSV*: Select to import disclosure facts in CSV format.
- *DISCL_TSV*: Select to import disclosure facts in TSV format.
- *ELMTS_CSV*: Select to import taxonomy elements in CSV format.
- *ELMTS_TSV*: Select to import taxonomy elements in TSV format.
- *FACTS_CSV*: Select to import financial element facts in CSV format.
- *FACTS_TSV*: Select to import financial element facts in TSV format.

Load Option

Specify how to handle duplicate rows. Values are:

- *Insert and/or Update*: Select to load all rows. If duplicate rows exist, they are replaced with the new data in the source file.
- *Ignore rows that already exist*: Select to only load new rows. If duplicate rows exist they are rejected. In other words, the existing data is retained, while new data is loaded.

File Errors Option

Specify how to handle errors during the import process. Value are:

- *Abort run without loading data*: Select to reject the file without loading any data. Error rows are written to the error log.
- *Skip errors rows and continue*: Select to load only valid rows. Rejected rows are written to the error log.

Run

Click to start the data upload process.

After running the process, review errors logs (if any) and take the necessary corrective actions.

The Load Data process does not use EPM job streams. Instead, the process uses online temporary table instances, and Peopletools manages the assignment of temporary tables. The instance count for the Load Data process has been set to 3. Make sure that the "Online Temp Table Instances" parameter in PeopleTools Options is configured to at least 3.

Maintain Taxonomy Elements Page

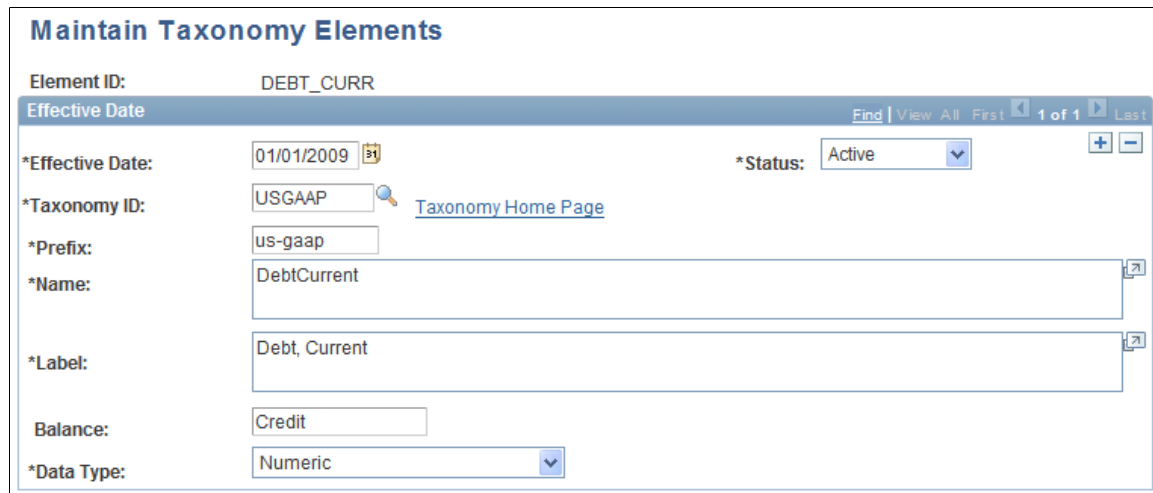
Use the Maintain Taxonomy Elements page (XBRL_ELEMENTS) to review or update taxonomy elements that you have uploaded.

Navigation

EPM Foundation, XBRL Reporting, Maintain Data, Maintain Taxonomy Elements

Image: Maintain Taxonomy Elements page

This example illustrates the fields and controls on the Maintain Taxonomy Elements page . You can find definitions for the fields and controls later on this page.



Maintain Taxonomy Elements

Element ID: DEBT_CURR

Effective Date: 01/01/2009

*Effective Date: 01/01/2009

*Taxonomy ID: USGAAP [Taxonomy Home Page](#)

*Prefix: us-gaap

*Name: DebtCurrent

*Label: Debt, Current

Balance: Credit

*Data Type: Numeric

*Status: Active

Use this page to maintain the taxonomy elements that you upload.

Element ID

Each element ID represents a combination of taxonomy element name and the namespace it belongs to. The element ID is referenced in the instance facts, and also when you create an XBRL instance. This is a user-defined field, in other words, you decide what ID to use.

Taxonomy ID

The taxonomy this element is associated with. You set up taxonomy IDs using the Define Taxonomy page.

Prefix

The namespace prefix (alias) for the element.

Name

The taxonomy element name, as defined in the taxonomy schema.

Label

The taxonomy element label, as defined in the taxonomy schema.

Balance

Balance type can be Debit or Credit. This field is blank for some elements that are used for footnotes and cash flow line items.

Data Type

Values are *Numeric* or *Text*.

Maintain Footnotes Page

Use the Maintain Footnotes page (XBRL_FOOTNOTES) to review or update footnotes.

Navigation

EPM Foundation, XBRL Reporting, Maintain Data, Maintain Footnotes

Image: Maintain Footnotes page

This example illustrates the fields and controls on the Maintain Footnotes page. You can find definitions for the fields and controls later on this page.

You can create or update footnotes using this page, and use the Fact References tab to review the instance facts they are associated with. The footnotes may be associated to an element representing a report line or it may have its own element. In the later case, the amount and UOM fields will be blank.

Copy

Click this button to copy the current footnote to a new Footnote ID.

Language ISO Locale

Enter the ISO language code for the footnote.

Type

Specify the type of footnote. Values are:

- *General Footnote*: Select for a footnote that does not reference a report line item.
- *Reference to Report Line Item*: Select for a footnote that references a line item on the report, and enter the report line number it is associated with in the Footnote# field.

Maintain Instance Facts Page

Use the Maintain Instance Facts page (XBRL_INST_FACT) to review or update instance facts that you have uploaded.

Navigation

EPM Foundation, XBRL Reporting, Maintain Data, Maintain Instance Facts

Image: Maintain Instance Facts page

This example illustrates the fields and controls on the Maintain Instance Facts page . You can find definitions for the fields and controls later on this page.

Maintain Instance Facts							
Taxonomy ID: USGAAP		Entity Name: ABC_CORP_SEC					
Report ID: BAL_SHT							
Fact Data							
*Context ID		*Line Number	*Element ID	Unit of Measure	Decimals	Amount	Footnote ID
AS_OF_JUNE30_2008	1	CASH	USD	-6	10000000		
AS_OF_JUNE30_2008	2	INVEST_SHORT	USD	-6	10000000		
AS_OF_JUNE30_2008	3	AR_CURR	USD	-6	5000000		
AS_OF_JUNE30_2008	4	EXP_PREPAID	USD	-6	3000000		
AS_OF_JUNE30_2008	5	ASSETS_OTHER_CU	USD	-6	15000000		
AS_OF_JUNE30_2008	6	INVENTORY_NET	USD	-6	4000000		
AS_OF_JUNE30_2008	7	ASSETS_CURR	USD	-6	30000000		
AS_OF_JUNE30_2008	8	PPE_NET	USD	-6	8000000		
AS_OF_JUNE30_2008	9	AP_CURR	USD	-6	2000000		
AS_OF_JUNE30_2008	10	CAPITAL_OTHER	USD	-6	5000000		
AS_OF_JUNE30_2008	11	RETAINED_EARNING	USD	-6	9000000		
AS_OF_JUNE30_2008	12	T_REVREC_GP	USD	-6	30000	STD_REVREC_GP	
AS_OF_JUNE30_2008	13	T_REVREC_LP	USD	-6	40000	STD_REVREC_LP	
AS_OF_JUNE30_2008	14	COMMON_STOCK	USD	-6	2000000		

Use this page to review and maintain the instance facts that you upload. You can associate footnotes with a fact by selecting a Footnote ID.

Creating XBRL Instance Documents

This topic discusses how to create an XBRL instance document. Page Used to Create Instance Documents.

Page Used to Create an XBRL Instance Document

Page Name	Definition Name	Navigation	Usage
Create Instance	XBRL_INST_RUN	EPM Foundation, XBRL Reporting, Generate Instance, Create Instance	Generate an XBRL instance report.

Create Instance Page

Use the Create Instance page (XBRL_INST_RUN) to generate an XBRL instance report. After all of your setup is complete, and you've imported your taxonomy and financial facts, you can create an XBRL instance document using the Create Instance application engine process.

Navigation

EPM Foundation, XBRL Reporting, Generate Instance, Create Instance

Image: Create Instance run control page

This example illustrates the fields and controls on the Create Instance run control page. You can find definitions for the fields and controls later on this page.

This page uses all of the set up information to create an XBRL instance report.

Instance Header

Specify the instance header to use for this XBRL instance report. You establish an instance header using the Define Instance Header page.

As of Date

Specify the date for the instance. This date is used to resolve the setup rules with effective dates.

Edit/View Header

Click this link to access the Define Instance Header page, where you can review or modify the instance header definition.

Entity Name

Select the entity that you will be submitting the XBRL instance document to. You establish XBRL entities using the Define Reporting Entity page.

Edit/View Entity

Click this link to access the Define Reporting Entity page, where you can review or modify the reporting entity definition.

Report Output

Specify the type of report output. Values are:

- *Report ID*: Select to generate an instance document from a single report ID, then select the report in the Report ID field.

- *Report Set*: Select to generate an instance document for a report set, then select the reports set in the Report Set field.

Preview

Select this check box to create an BI publisher report for all the line items including footnotes. Use this report to verify the results of the XBRL instance document.

Note: BI Publisher must be installed for this option to work.

See *PeopleSoft PeopleTools PeoplBook: BI Publisher for PeopleSoft*

Create

Select this check box to create the XBRL instance document.

Publish

If the Create option is selected, then you can also select this check box to create an XML message using Integration Broker. The message type is non-rowset based. You can view the Instance document as an XML message using the Service Operations Monitor of the Integration Broker.

Note: You need to set up Integration Broker for this option to work. Only the message object is delivered.

See *PeopleSoft PeopleTools: PeopleSoft Integration Broker Administration*

File Name

Enter the filename to assign to the generated XBRL instance report, without the file extension. The system automatically appends ".XBRL" and ".XML" to the file name.

The content of the XML and XBRL files are identical. You will need to submit the XBRL file per the XBRL requirements.

The Create Instance process does not use EPM job streams. Instead, the process uses online temporary table instances, and Peopletools manages the assignment of temporary tables. The instance count for the Create Instance process has been set to 3. Make sure that the "Online Temp Table Instances" parameter in PeopleTools Options is configured to at least 3.

To confirm the document is well formed, open it in an XML editor or an Internet browser, such as Internet Explorer or Mozilla Firefox. If it loads without errors, it is well-formed.

After the process is complete, review the process monitor status and the message logs for any logged exceptions. You can access the instance documents using the process monitor or the report manager if the report node is setup properly.

The context IDs are system generated for each segment (dimension member) value of a table schedule.

Appendix A

ETL Installation and Implementation Prerequisites and Considerations

Preparing to Install and Implement IBM WebSphere DataStage

Perform the following preparatory tasks before you begin implementing ETL jobs:

1. Create a detailed list of all the EPM products that have been purchased and the related license codes.
Identify and enumerate the products you are going to implement and in what order.
2. Create a detailed implementation schedule that accounts for the EPM data marts and business units you are going to implement.
3. Review the list of ETL application software components (such as .dsx, parameter, and DSParams files) and identify which are necessary for your requirements based on your implementation schedule.
See [DSX File Import Descriptions](#).
4. Identify the list of database tables that will be populated and the list of corresponding jobs that have to be executed to populate these tables.

Note: Apart from the jobs, which directly populate the relevant target tables, you must also identify all the dependent jobs, such as hash file load jobs.

5. Perform all non-ETL implementation tasks.

Sizing Your Database/DataStage Servers

EPM uses hash files extensively, which are stored in a directory which you specify on the server. It is important, therefore, to remember this when you are determining sizing requirements because the amount of data in the hash files will increase with time. Also the server directory should hold the flat files as well as XML file inputs that the ETL process requires. Generally, every staging table has a corresponding hash file, and every dimension table has a corresponding hash file, so the size of all the hash files is a function of the size of the data that is stored in staging tables and the dimension tables. However, it is also to remember that only relevant columns in a table are loaded into a hash file.

For sizing the space requirement for hash files, we suggest that you take a few sample hash files and compare them with the underlying tables to determine the size requirement. Also compare the structure of the table and the number of columns in that table that are actually loaded to the hash file. It is very important to keep sufficient buffer size for future incremental data, since as the data size increases with time the hash files also grow in size. Another way to do this is with the help of an unsupported tool

provided along with the IBM WebSphere DataStage CD. The tool is called HFC.exe, which is short for Hash File Calculator.

Perform the following server sizing tasks before you begin implementing ETL jobs:

1. Refer to all relevant database sizing documents delivered with EPM, and thoroughly familiarize yourself with it before implementation.
2. Perform database sizing, considering all the tables that are populated by the ETL process as well as those used for reporting.
3. Run the delivered script for inserting a *Not Available* row into all relevant tables.

This script will insert one Not Available row each into every table, which is a prerequisite for the ETL application.

Note: You can find the script on the installation CD in the following location: <PSHOME>\SRC\ETL.

4. To size the DataStage server, determine the number of hash files that will be created for the subset of the ETL application that you are going to implement.

You can use the list of jobs you have created in previous steps and the list of hash files that are supplied along with EPM.

5. Calculate the space required for storing all of these hash files.

You must consider hash file properties and structure, as well as the quantum of data that is associated to each hash file to perform hash file sizing.

Note: A buffer should be allocated for future incremental data (growth in the size of the hash file).

6. Decide where you will physically store hash files by setting the value in the environmental parameter.

Space is also required for Datastage server log files.

7. Allocate space for all the other input data files such as XML files, parameter files, and *.dat files.

DataStage Server Requirements

Please see the *IBM Information Server: Planning Installation and Configuration Guide* for the minimum requirements to install the DataStage Server on a specific platform.

DataStage Client Requirements

Please see the *IBM Information Server: Planning Installation and Configuration Guide* for the minimum requirements to install the DataStage Client.

Determining Server Configuration and Installing IBM WebSphere DataStage

Perform the following server configuration and installation tasks before you begin implementing ETL jobs:

1. Determine a suitable server configuration for your development, QA, and production environments.
2. Install the DataStage servers.

Create separate servers for development, QA, and production environments.

3. Perform all required steps to configure the database, depending on your source and target databases.
4. Install the DataStage client.
5. Apply the latest patches for DataStage server and client.

IBM WebSphere DataStage Implementation Considerations

The following considerations should be noted before you begin DataStage implementation:

1. Perform a detailed analysis of your project creation strategy.

You should decide whether you would like a single project for the whole EPM application or have separate projects for each data mart.

2. Create separate DataStage projects for development, QA, and production.

PeopleSoft recommends that the production project reside on a separate DataStage server.

3. Classify your jobs as high, medium, and low volume.

Provide project defaults for array size, transaction size, IPC buffer and other performance parameters. Any exceptions and special cases must be handled by changing the value at the job level.

4. Open a sample job from each category and familiarize yourself with the filter conditions in the source, update strategy, job design, job parameters and other transformations.
5. Review the master run utility and create appropriate sequential file inputs.

Analyze this feature and decide on the different categories that you want to run using this utility.

6. Review the master sequencers and familiarize yourself with them.

7. Open one of the business process and identify all the jobs that are required to run it.

Run this as an example to learn how the jobs are ordered, the interdependencies, the hash file usage, and so forth.

Related Links

[Understanding IBM WebSphere DataStage Setup and Configuration](#)

[Understanding ETL in EPM](#)

Defining a Job Execution Strategy

The following job execution strategies should be noted before you begin running jobs:

1. Plan a job scheduling strategy and use the DataStage Director scheduler or another third-party tool.

Do a sample run using the scheduling tool to test whether the tool meets all your requirements for scheduling the application.

2. Familiarize yourself with all the job execution utilities that are provided with DataStage.
3. Define the error validation strategy you wish to use in your job.

Appendix B

ETL Frequently Asked Questions

IBM WebSphere DataStage ETL Tools

This table provides answers to general questions about IBM WebSphere DataStage ETL tools.

Question	Answer
What versions of IBM WebSphere DataStage are supported in EPM?	EPM only supports IBM WebSphere DataStage version 8.1 server edition.
What if I am using an earlier version of DataStage either as part of a non-PeopleSoft installation or as part of EPM?	You must upgrade to IBM WebSphere DataStage version 8.1 server edition
What are the differences between ETL content for EPM release 9.1 and the previous EPM releases?	For 9.1 OWS staging tables contain all the fields of its related source tables.
What are the IBM WebSphere software components that are delivered with EPM?	EPM delivers IBM InfoSphere Information Server 8.1
What is QualityStage?	<p>IBM WebSphere QualityStage provides a powerful framework for developing and deploying data investigation, standardization, enrichment, probabilistic matching and survivorship operations. For use in transactional, operational, or analytical applications, in batch and real-time, the same services are seamlessly deployed to facilitate data validation, cleansing or master data entity consolidation for your, locations and products.</p> <p>For more information, please use the following link to the IBM WebSphere website: http://www-01.ibm.com/software/data/infosphere/qualitystage/</p>
What is InfoSphere Information Analyzer?	For more information, please use the following link to the IBM WebSphere website: http://www-01.ibm.com/software/data/infosphere/information-analyzer/
What is IBM Metadata Workbench?	IBM Metadata Workbench provides Web-based exploration of information assets that are generated and used by IBM Information Server applications. IBM Metadata Workbench gives you out-of-the-box reporting on data movement, data lineage, and the impact of changes and dependencies. With the workbench, you can trace the data lineage of business intelligence reports to provide a basis for compliance with regulations such as Sarbanes-Oxley and Basel II. IBM Information Server components generate design time and runtime metadata, automatically storing that metadata in the IBM Information

Question	Answer
What is Parallel Extender?	DataStage Parallel Extender (DS-PX) is a highly scalable parallel processing infrastructure package for the development and execution of data integration, data warehousing, business intelligence and analytical applications.
How does DataStage ensure application security?	Password and role based security can be effectively implemented in DataStage at a project level from the DataStage administrator.
How does DataStage ensure data security?	EPM accesses data on databases using the DRS stage. The user ID and password for accessing the databases are parameterized as environmental variables and the password parameter can be set as an encrypted field in the DataStage Administrator. This ensures data security by restricting the database access passwords.

DataStage Server Installation

This table provides answers to DataStage Server installation questions.

Question	Answer
What are the prerequisites to the DataStage server installation?	For information on prerequisites to DataStage server installation, please refer to the <i>PeopleSoft Enterprise Performance Management Installation Guide</i> . See <i>IBM Information Server: Planning Installation and Configuration Guide</i>
What steps must I perform to install IBM WebSphere DataStage Server?	For information on prerequisites to DataStage server installation, please refer to the <i>PeopleSoft Enterprise Performance Management Installation Guide</i> . See <i>IBM Information Server: Planning Installation and Configuration Guide</i>
How do I verify if I have successfully installed the IBM WebSphere DataStage Server?	After you install and configure IBM InfoSphere DataStage and QualityStage, test the installation by logging in to the Administrator and Designer clients. See <i>PeopleSoft Enterprise Performance Management Installation Guide</i> .

DataStage Client Installation

This table provides answers to DataStage Client installation questions.

Question	Answer
How do I verify whether I have successfully installed DataStage client?	Please refer to the <i>PeopleSoft Enterprise Performance Management Installation Guide</i> .

DataStage Patches

This table provides answers to questions about DataStage patches.

Question	Answer
Are there any other relevant patches that I have to apply other than DataStage patches?	<p>There are two patches for SQL Server and DB2 database users.</p> <p>These are DRS patches for these two databases.</p> <p>You can find these patches in the following resolutions:</p> <ul style="list-style-type: none"> • 829720 - DRS patch for SQL server database users • 829719 - DRS patch for DB2 database users
How do I verify that the patch has been installed correctly?	Verify that the files specified in the install guide for the patch are updated properly to verify that the patches were installed correctly. Also, test some jobs in the project and ensure that no abnormal termination or any other errors occurs right after applying the patch.

ETL Content

This table provides answers to ETL content packaging questions.

Question	Answer
How is the prepackaged ETL content delivered to me?	The DataStage application is delivered in the form of *.dsx files, which are text files that can contain export of a whole DataStage project or can contain DataStage design components and executables, like server jobs, sequencers, routines, containers and so forth.
Where can I find the list of delivered DSX files?	See DSX File Import Descriptions .
How are the delivered dsx files packaged in the CD (and future bundles)?	There are 38 dsx files that are delivered with EPM. You can see dsx files for five warehouses and 1 for Common where you have all the dsx files that are common across the warehouses.
What are the non-DSX file deliverables with the CD (and future bundles)?	<p>Other than DSX files, which contain DataStage job designs, the other deliverables are flat file inputs, xml file inputs and parameter files.</p> <p>See Parameter and Source Data Files Information.</p>

Question	Answer
What are the various EPM bundle files and how will they be packaged in the bundle1/bundle2/ICE Resolution?	Refer to the bundle posting related information from You Connection.
What are the various source application release versions that have been used for EPM 9.1 release?	<ul style="list-style-type: none"> • FSCM Source Release – FSCM 9.1 • CRM Source Release - CRM 9.1 • HRMS Source Release - HCM 9.1 • Campus Solutions Source Release -CS 9.0

Environmental Variables

This table provides answers to questions about environmental variables.

Question	Answer
What are environmental variables and how are they used?	Environment variables enable you to set the project defaults and these are used in jobs.
What is the DSParams file?	For any datastage project, all project levels, user defined environmental items are stored in a file called DSParams. It is located in <dshome>\DataStage\Projects\<projname>.

Question	Answer
<p>How do I copy the DSParams file the first time as opposed to copying from one project to another project?</p>	<p>The methodology described below provides a workaround for moving or sharing the global parameters without having to re-type them in the administrator. The work around consists of replacing and/or editing this file to add the parameters. Be sure to back up the original file before any other activity occurs.</p> <p>For a <i>new project</i> that has not yet defined any global parameters, just copy the existing DSParams file to the new project. Be sure to rename the existing DSParams file. Ensure all DS clients (Designers, etc.) are logged off, and stop and start the DataStage services to activate it. Then go into the DataStage Administrator and all the parameters should be visible in the user-defined section of the environment screen. At this point, edit the default values for each parameter.</p> <p>For an <i>existing project</i> that has already defined some global parameters; the DSParams file must be edited to add the desired parameters. The process below describes how to do this. The user-defined parameters are in two sections of the DSParams file: one section defines the parameters [EnvVarDefns] and the second section contains the default values [EnvVarValues]. The approach is to copy the correct lines from the original source project file into the target project DSParams file:</p> <ol style="list-style-type: none"> 1. Rename/Backup the DSParams file in the target project directory and backup the source project DSParams file as well. 2. Edit the source project DSParams. Go to the end of the [EnvVarDefns] section and find the user defined parameters, which are at the end of the section. Select the lines up to but not including the line which contains [PROJECT]". 3. Copy these lines and paste them into the target project DSParams file before the "[PROJECT]" section. 4. Go back and edit the source project DSParams file. Find the section starting with the line "[EnvVarValues]". This is usually at the end of the file. Copy all of the lines of that section, or select all the lines for the specific parameters to be moved. 5. Locate the end of the DSParams file in the target project directory. See if it has a section called "[EnvVarValues]". If it does not, add it. If it does, then go to the next step. 6. Paste the lines into the target project DSParams file at the end of the "[EnvVarValues]" section and before the end of file. 7. Save the DSParams file in the target project directory. 8. Ensure all DS clients (designers and so forth) are logged off, and stop and start the DataStage services to activate it. Then go into the DataStage Administrator and all the parameters should be visible in the user-defined section of the environment screen. At this point, change the default values for each parameter.

Question	Answer
<p>What are 'Array' and 'Transaction' sizes?</p>	<ul style="list-style-type: none"> • <i>Array Size:</i> Array size is a parameter to specify the number of rows written (to a database) at a time. In other words it refers to the number of rows that are transferred in one call between DataStage and the database before they are written. Generally, increasing the array size will increase performance since client memory is used to cache records resulting in lesser server hits. The maximum size for array size is 32767. But increasing the array size too much will result in strain on the client memory. Hence an optimal value must be arrived at considering the client memory. For flexibility this has been parameterized as a environmental variable. Separate environmental variables are available for each source as well as for OWS, OWE and MDW. For direct DRS lookups, the recommended array size is 1 since DS lookups generally expects 1 row of data to be returned. • <i>Transaction Size:</i> Transaction size refers to the number of rows that are written to the database before the data is committed. Giving a transaction size of zero will ensure that commit doesn't happen until all the records are written. The default value is 0. If the transaction size is set to 100 then the database table commits are performed every 100 rows. Here again an optimal value must be arrived at considering the strain on the Database server and the number of records. For flexibility this has also been parameterized as a environmental variable. Separate environmental variables are available for each source as well as for the OWS, OWE and MDW.
<p>How is the DATA_ORIGIN environment variable configured?</p>	<p>The value for DATA_ORIGIN is 'S' for following loads:</p> <ul style="list-style-type: none"> • OWS (data flow from PeopleSoft source system) • OWE (data flow from OWS) • MDW (data flow from OWS) <p>The DATA_ORIGIN must be toggled to <i>Enterprise</i> only when running those sets of jobs whose flow is from OWE to MDW.</p>

Project Creation, Import, and Export

This table provides answers to project creation, import, and export questions.

Question	Answer
What are the different methods of maintaining projects if I have jobs that source from more than one warehouse?	<p>The EPM ETL design enables jobs to distribute across multiple projects or only a single project. However, the following are some of the options to create projects based on your needs.</p> <ul style="list-style-type: none"> • <i>Option 1: One project per EPM warehouse</i> <p>If you have more than one EPM warehouse, you can create one project per warehouse and one project for all the jobs related to Common jobs and Global Dimensions. For example, if you have HCM and FMS warehouse then there are three projects created for HCM, FMS and Common. The J_Dim_PS_D_PERSON is a part of Global Dimensions and this job must be triggered in Common project whereas the corresponding Hash Load jobs have to be triggered in HCM and FMS projects as well.</p> <ul style="list-style-type: none"> • <i>Option 2: All EPM warehouses in a single project</i> <p>You can have one single project for all the jobs for all the EPM warehouses, Common jobs and Global Dimensions. If you maintain only one project for all the warehouses (HCM, FMS and Common jobs) you need to run the Global Dimensions, Local Dimensions (HCM, FMS), SKU based dimensions and there is no need to run the Hash Load jobs after the first run.</p> <p>For this option, there is a possibility of reaching a maximum number of files limit for directories in some OS platforms. Please verify with system administrators before proceeding with this option.</p>
What configurations must I perform after creating the projects?	<p>Projects must be created from the DataStage Administrator.</p> <p>See Setting Up DataStage Projects.</p>
How do I import a project? How do I verify a successful import?	<p>For more information about importing a project, see section 2-5 of the Configuring IBM WebSphereDataStage document.</p> <p>See Setting Up DataStage Projects.</p>
What are the DataStage categories (folders) and sub-categories that I will see after project import?	<p>For more information about the DataStage categories available, see the topic 'Understanding the Project Structure' in this document.</p>

Setting Up Development and Production Environments

This table provides answers to questions about development and production environment setup.

Question	Answer
Where can I find configuration information for different server platforms?	See <i>IBM Information Server: Planning Installation and Configuration Guide</i>

Question	Answer
<p>What are the supported platforms for Data stage?</p>	<p>The following platforms support the IBM InfoSphere Information Server:</p> <ul style="list-style-type: none"> • HP-UX 11i v2, v3 on PA-RISC • HP-UX 11i v2, v3 on Intel Itanium • IBM AIX 5.3 and 6.1 • Red Hat Enterprise Linux Advanced Server 4 on AMD or Intel processors • Red Hat Enterprise Linux 5 Advanced Platform on AMD or Intel processors • Red Hat Enterprise Linux 5 Advanced Platform on IBM System z • SUSE Linux Enterprise Server 10 on AMD or Intel processors • SUSE Linux Enterprise Server 10 on IBM System z • Sun Solaris 9 and Sun Solaris 10 • Microsoft Windows XP Service Pack 2, Microsoft Windows Vista, Microsoft Windows Server 2003 Service Pack 2 (Client only) • Microsoft Windows 2003 Service Pack 2 (Server only) <p>Data sources Supported by EPM 9.1 application:</p> <ul style="list-style-type: none"> • Oracle • MSSQL Server • IBM DB2 on OS/390 • IBM DB2 on Unix/Windows • Flat files (Sequential Files) • XML files <p>Refer to the PeopleSoft Enterprise Performance Management Hardware and Software Requirements Guide for more details.</p>
<p>Are there database-specific configuration steps?</p>	<p>For MSSQL Server, you must enable the options to support functional index (MSCONCATCOL) in the database:</p> <ul style="list-style-type: none"> • SET ANSI_NULLS • QUOTED_IDENTIFIER • CONCAT_NULL_YIELDS_NULL • ANSI_WARNINGS • ANSI_PADDING

ETL Jobs

This table provides answers to general questions about ETL jobs.

Question	Answer
What are the different categories of jobs in EPM?	<p>The following is a list of EPM job categories:</p> <ul style="list-style-type: none"> • Source to OWS (Staging area) • OWS to OWE • OWS to MDW • OWE to MDW • OWS to MDW (for CRM online marketing data mart) • OWE to source (for GL and Position Budgeting in HCM)
What types of staging loads does EPM support?	<p>EPM supports incremental staging loads along with a small number of destructive staging loads.</p> <p>See Understanding ETL Load Strategies in EPM.</p>
What is the load strategy for the MDW and OWE?	<p>Most of the MDW and OWE loads are provided with Incremental logic using LASTUPD_EW_DTTM field which is part of the subrecord LOAD_OWS_SBR.</p> <p>See Understanding ETL Load Strategies in EPM.</p>
How does EPM manage incremental loading if the source fields have Null values for the Datetime stamp?	<p>If the Datetime column is a nullable field on the source database, then source filter will include a condition to bring that data as well along with the incremental data.</p> <p>See Understanding ETL Load Strategies in EPM.</p>
Are there any special loading requirements between the first run and subsequent runs?	<p>No, there are no special loading requirements between the first run and subsequent runs, provided the prepackaged jobs are used without any customization.</p>
How does EPM manage deletes in the source tables?	<p>For the Campus Solutions Warehouse, EPM provides staging jobs that can identify source records that have been physically deleted from your PeopleSoft transaction system and flag those records for physical deletion from the Campus Solutions Warehouse schema.</p> <p>For more information see the topic <i>Managing Source System Deletes and Archiving</i> in the <i>PeopleSoft Campus Solutions Warehouse</i>.</p>
Does EPM have the functionality to accommodate late arriving dimensions?	<p>No, EPM does not have the functionality to accommodate late arriving dimensions.</p>
What types of data sources does EPM accommodate?	<p>EPM can accommodate XML Files, Flat Files, and RDBMS tables.</p>

Question	Answer
Does EPM use any control tables in its ETL design?	EPM no longer uses control tables and those that were used in past EPM releases (such as ETL_JOB_CTL, ETL_RUN_HST, ETL_TSE_ERR) have been deprecated.
Does EPM use surrogate keys? If so, how are Surrogate IDs generated and managed?	Datastage manages the universe file SDKSequences, which will hold the surrogate key sequences for a particular key. See Surrogate Keys .
Does EPM have a strategy for handling slowly changing dimensions?	In EPM, the dimension D_EMPL_JOB from HCM warehouse is designed as Type 2 slowly changing dimension and all the other dimension loads are Type 1. However, the lookup operation supports Type 2 slowly changing dimension (for example, whenever there is lookup on other dimension, it will have effective dated logic). For more information on configuring Type 2 slowly changing dimensions, see the <i>Configuring Slowly Changing Dimensions</i> topic in your warehouse specific documentation (for example, the <i>PeopleSoft Campus Solutions Warehouse</i>).
How to change my dimension load from Type 1 design to Type 2 design?	For more information on configuring Type 2 slowly changing dimensions, see the <i>Configuring Slowly Changing Dimensions</i> topic in your warehouse specific documentation (for example, the <i>PeopleSoft Campus Solutions Warehouse</i>).
Does EPM support ETL rollbacks? If so, how is this handled?	Rollback is possible through the Transaction Size parameter. If the transaction size is selected as zero and if the job aborts in the middle, then the job will rollback the transactions since it follows the principle of two-way commit. If the transaction size is anything other than zero and if the job fails in the middle, then the job will perform commits for the number of rows that processed till the error message.
What aggregation strategy does EPM ETL employ?	The aggregator stage is generally not used in job design since the aggregation functions are better left to the database since the database can perform aggregation functions more efficiently than Datastage. Whenever the aggregation must be performed on the source data, it is achieved within DRS source stage itself. In case of generated sql queries, aggregate functions are given in against columns in corresponding derivation columns and group by clause is given in 'Other clauses' text area. Wherever User Defined SQL option is selected the query is specified appropriately with the aggregate function. In specific instances where an aggregation function must be performed on data that is transformed and not directly read from the Database and in cases where the number of records is going to be large, temporary table is created where the data is temporarily written and then read out, when the aggregation functions can be performed.
What indexing strategy does EPM ETL employ?	EPM data models are delivered with indexes. Before loading the target tables, drop the indexes and then build them after load. This improves ETL performance.

Question	Answer
How are lookups used in the EPM ETL design?	<p>Lookups are usually used in a Hashed file stage, except for relational joins, when they are used in the DRS stage instead.</p> <p>See ETL Hashed Files, Understanding Data Validation and Error Handling in the ETL Process.</p>
What types of job parameters does EPM use to increase run time flexibility?	<p>Parameterization helps you enter run time parameters without resorting to changing jobs.</p> <p>Run time information, such as the Database type, the database connection parameters, and parameter file directories should be set as environmental variables, which are used in individual jobs.</p> <p>Parameter files are used for those jobs, which read from the user, input variable values or a list of values, which may change from run to run. The variables and their respective values are given in parameter files.</p> <p>See Parameter and Source Data Files Information.</p>
How is DataStage code re-used?	<p>PeopleSoft packages reusable DataStage code with Shared Containers, routines, and some server jobs (found in the Reusable Jobs folder).</p>
Are there any customizations required to handle Unicode data?	<p>To support Unicode databases, the DataStage Server must be installed with NLS enabled. Also, the proper character set should be selected based on the requirements by the user, in the DataStage Administrator.</p>

Survey Jobs

This table provides answers to questions about survey jobs.

Question	Answer
<p>Where are the Survey-related jobs in the dsx files?</p> <p>After import, where can I find the Survey jobs in the DataStage Project?</p>	<p>Survey jobs are present in OWE and MDW modules of HCM warehouse.</p> <p>In OWE module, there are some D00 jobs which reads the flat file data as source and loads the R00 tables. These jobs can be located in OWE.dsx and after the import, the jobs will be present under the OWE_E\HCM\D00\Base\Load_Tables\Server category.</p> <p>In the MDW module, the R00 tables are used as source and it load the Survey Dimension tables. These jobs can be located in WHR_WORKFORCE_PROFILE_MART_E.dsx file and after the import, the jobs will be present under the \HCM_E\WORKFORCE_PROFILE_MART\Survey\OWE_To_MDW\Dimensions\Base\Load_Tables\Server category.</p> <p>In EPM, the dimension D_EMPL_JOB from HCM warehouse is designed as a type 2 slowly changing dimension and all the other dimension loads are Type 1. However, the lookup operation supports Type 2 (for example, whenever there is lookup on other dimension, it will have effective dated logic).</p>
<p>What are the required steps in a Survey load?</p>	<ol style="list-style-type: none"> 1. <i>Run Survey Setup ETL:</i> These are the jobs that read the source flat files or the temp tables and loads the R00/D00 tables. These jobs can be located in OWE_E.dsx files and it will be present under the path \OWE_E\HCM\D00\Base\Load_Tables\Server category. 2. <i>Run Survey Load ETL Batch:</i> These are the jobs that read the data loaded in the above step and loads the F00 tables. These jobs can be located in OWE.dsx files and it will be present under the path \OWE_E\HCM\F00\Base\Load_Tables\Server category. 3. <i>Load Employee level Competencies:</i> These jobs loads the Competency details of the employee from the OWS tables. These jobs can be located in OWE.dsx files and it will be present under the path \OWE_E\HCM\F00\Base\Load_Tables\Server category.
<p>What are the prerequisites for loading the Survey module?</p>	<p>OWE Survey jobs are present under the HCM warehouse. These jobs use Flat files as sources. These source flat files should be present in the Project home directory and the environmental parameter \$SOURCE_FILE_DIR should have the directory path of these survey flat files.</p> <p>If the DataStage server is on Windows, then the survey jobs have to be modified by accessing the Sequential file stage and changing the Line Termination to <i>DOS Style (CR LF)</i>. And then save the job, Compile and run the same.</p>

ETL Hashed Files

This table provides answers to questions about ETL hashed files.

Question	Answer
How are hash files used and for what purpose?	<p>Hash Files are used to enhance the performance of the ETL job. Hash Files are typically used for lookups in an ETL job.</p> <p>In EPM, there are jobs to initialize Hash Files. These jobs create the hash files before the jobs requiring them for lookup are executed. These Hash Files are also updated once the target table is loaded in the ETL job. This method will enable multiple jobs to utilize the same hash file as long as the structures required are the same.</p> <p>Another method is to load the hash file within the same job using them as a lookup. This method requires the hash files to be reloaded every time the job executes.</p> <p>See ETL Hashed Files, Understanding Data Validation and Error Handling in the ETL Process.</p>
What should I keep in mind when managing my hash files?	<p>The default setting for Hashed Files are project specific and cannot be shared across projects. The validity of Hashed Files is dependent on the base table it is generated from. The base table should only be updated by the ETL jobs provided in EPM. If not, the hashed file and the table will be out of sync and may result in faulty data when used in an ETL job.</p> <p>There are several Hashed File utilities provided in EPM. These are located in the Utilities\Hash_Utils category.</p>
Can I customize the storage location for hash files?	<p>It is possible to customize the storage location for hash files by specifying the directory path.</p> <p>You can set the storage path of the hash files. The path location has to be set in the environmental parameter # \$HASHED_FILE_DIRECTORY# and this parameter is used across all the hash files.</p>
How to recover data from corrupted hash files?	<p>Generally, a corrupted hash file must be reloaded from the base table. EPM provides utilities to back up and recover DateTime and SurrogateKey hashed files.</p>

ETL Routines

This table provides answers to questions about ETL routines.

Question	Answer
How are routines used?	<p>Routines are used to make DataStage job code reusable. Routines are used in various parts of the job design. 95 routines are delivered as part of EPM ETL. These are present in a category called 'EPM_Routines'.</p>
Where can one find the details for all the EPM Routines?	<p>See Routine Descriptions.</p>

ETL Job Process Flow

This table provides answers to questions about the ETL job process flow.

Question	Answer
How does EPM manage the process flow for ETL jobs? How are job interdependencies managed?	DataStage Sequence job allows you to specify several jobs to run in controlled manner and can be used to specify different courses of action to take depending on whether a job in the Sequence succeeds or fails. Every ETL load has a Sequence job and each business process within a datamart is provided with a master sequence to trigger all the jobs belonging to it. See Using the Master Sequencer Utility to Create Master Sequencer Jobs .
What are the common triggers used in the process flow?	Triggers are used to control the flow of a Sequence job in triggering various other Sequence/Server child jobs. The most commonly used ones are <ul style="list-style-type: none"> • Failed – Conditional • Warning – Conditional • OK – Conditional • Unconditional

ETL Utilities

This table provides answers to questions about delivered ETL utilities.

Question	Answer
What is the Language Swap utility?	If the source database base language is different from the EPM database base language, you must ensure that the EPM base tables have descriptions in EPM base language and the related language table have descriptions in EPM installed foreign language. The Language Swap ETL utility provides this functionality. For more information on the language swap utility, see the <i>Setting Up Multilanguage Processing and Running the Language Swap Utility</i> topic in your warehouse specific documentation (for example, the <i>PeopleSoft Campus Solutions Warehouse</i>).
What are related language tables and how are they packaged?	In EPM, every table that requires language translation has a corresponding related language table. ETL jobs to populate these language tables are created. These jobs are packaged along with the base table jobs. Running these jobs is optional, since not all of them require the use of multi-language functionality.

Question	Answer
What is the Currency Conversion utility?	<p>This utility is used to populate the reporting amount and reporting currency code columns in fact tables in the Multidimensional Warehouse. This population is considered an ETL post process. Before running the ETL, the setup for MDW Currency Conversion definitions should be completed in the PIA pages.</p> <p>For more information on the MDW currency conversion utility, see the <i>Implementing Currency Conversion for Multiple Currencies</i> topic in your warehouse specific documentation (for example, the <i>PeopleSoft Campus Solutions Warehouse</i>).</p>
What are tree-processing jobs?	<p>If a warehouse or data mart has tree or recursive hierarchy data, the ETL utility to process this data must be triggered. The utility flattens and denormalizes the set of hierarchies. These hierarchy definitions needs to defined in PIA pages before running the ETL jobs.</p> <p>For more information on the MDW tree flattening process, see the <i>Processing Trees and Recursive Hierarchies</i> topic in your warehouse specific documentation (for example, the <i>PeopleSoft Campus Solutions Warehouse</i>).</p>
What are the Dimension Mapper jobs?	<p>Dimension mapper jobs refer to a set of jobs that enable users to set up Warehouse Business Units and SetIDs. This mapper tool utilizes data from several other tables such as PF_SRC_SETCNTRL, PF_SRC_BU_NAMES, and PF_SRC_BU_ROLES, which are loaded by the dimension mapper jobs.</p> <p>The output tables of the Dimension Mapper tool are PF_SETID_LOOKUP, PF_BUS_UNIT_MAP, BUS_UNIT_TBL_PF, BUS_UNIT_TBL_FS, SETID_TBL, SET_SNTRL_TBL, SET_CNTRL_GROUP, and SET_CNTRL_REC. These tables are used as lookups in ETL job design.</p> <p>See Understanding Warehouse Business Units, TableSet Sharing, and SetID Mapping and Understanding Warehouse Business Unit Setup.</p>

Running ETL Jobs

This table provides answers to questions about running ETL jobs.

Question	Answer
How do I identify ETL jobs that are needed based on the license code or my implementation plans?	See Understanding the EPM Lineage Spreadsheets .
What do the OWS to OWE (D00) jobs do?	The D00 job from OWS to OWE loads data from staging tables to _D00 tables in the OWE database.
What do the OWS to OWE (F00) jobs do?	The F00 job from OWS to OWE loads data from staging tables to _F00 tables in the OWE database.

Question	Answer
What do the OWS to MDW (Global Dimensions) jobs do?	The Global Dimension jobs loads data from staging tables to dimension tables. Global Dimensions are dimension tables that are shared across warehouses.
What do the OWS to MDW (Local Dimensions) jobs do?	The Local Dimension jobs loads data from staging tables to dimension tables. Local Dimensions are dimension tables that are shared across different marts in a warehouse.
What do the OWS to MDW (data mart dimensions and facts) jobs do?	The dimension and fact job loads data from staging tables to dimension or fact tables. A dimension contains a key (SID) value and attributes used for slicing and dicing measures located in a fact table.
What do the OWE to MDW (Dimensions and Facts) jobs do?	The dimension and fact job loads data from OWE tables (D00 or F00) to dimension or fact tables. A dimension contains a key (SID) value and attributes used for slicing and dicing measures located in a fact table.
How do uncompiled jobs affect EPM?	All server jobs/sequencers should be compiled before running. Uncompiled jobs will not run and have to be compiled using the Designer prior to running.
In what order must I run different categories of jobs?	See Understanding the EPM Lineage Spreadsheets .
How are Master Sequences used to load data marts?	<ul style="list-style-type: none"> <li data-bbox="865 947 1464 1058">• <i>Loading Sequence for OWS:</i> To run the OWS jobs, you can use the Master_Run_UTILITY, which reads the list of OWS jobs from a flat file and executes them in the specific order as mentioned in the input flat file. Using this Master_Run_UTILITY, you can run Hash Load jobs (first time to create the Hash Files), Base - Sequence Jobs, and Language Sequence jobs. <li data-bbox="865 1199 1464 1310">• <i>Loading Sequence for OWE:</i> To run the OWE jobs, you can use the Master_Run_UTILITY, which reads the list of OWS jobs from a flat file and executes them in the specific order as mentioned in the input flat file. Using this Master_Run_UTILITY, user can run Hash Load jobs (first time to create the Hash Files), D00, F00 and Language D00 jobs. <li data-bbox="865 1451 1464 1675">• <i>Loading Sequence for MDW:</i> To run the MDW Dimensions and Facts, user can run the Master_Sequence jobs that are present under the Master_Sequence category. Master_Sequence category will be present under all subject areas of the data mart node. Hence, there are four Master Sequence jobs for each subject area: Hash Load Jobs, Base - Dimensions, Facts and Language Dimensions. For Global Dimensions and Local Dimensions you can find 3 Master Sequence; Hash Load Jobs, Base – Dimensions and Language Dimensions. Alternately, you can also use Master_Run_UTILITY to run the MDW Dimensions and Facts, provided there is a loading sequence flat file present in the DataStage Server.

Question	Answer
Can I use the People Tools Process Scheduler to trigger jobs?	No, you cannot use the Process Scheduler to trigger jobs. To schedule jobs, you can use the DataStage Scheduler or any other third party scheduling tool.
When do I use the Master Run utility in the utilities folder?	The Master_Run_UTILITY can be used to run the set of jobs that are present in a flat file in the DataStage Server. This utility will read the list of jobs that are present in the file and trigger them in a serial mode, by taking care of dependency logic as mentioned in the Input Flat file. Master_Run_UTILITY can be used to run any jobs in a dependent or independent mode.

Error Handling with ETL Jobs

This table provides answers to questions about error handling with ETL jobs.

Question	Answer
What is the Error validation mechanism built into EPM ETL design?	See Understanding Data Validation and Error Handling in the ETL Process .
How is Error Validation performed using Error tables?	See Understanding Data Validation and Error Handling in the ETL Process .

ETL Job Validation

This table provides answers to questions about ETL job validation.

Question	Answer
How do I confirm whether a job has run successfully?	<p>Log onto Datastage director and check the status of the job. If the status of the job is "Finished" then the job completed Successfully with no errors or warnings. If the status is finished then you can go and query your target database and check if the number of rows populated in the target table are correct.</p> <p>If the status of the job is Finished (See Log) then there are some warnings in the job and you need to check the log file to see what the warning is about and fix that particular issue and run it till the job completes with status Finished. Finally if the status is Abort then the job has some Fatal errors. You need to fix those errors and run the job again to make sure the job is in "Finished" status.</p>

Question	Answer
How can I test OWS, OWE, and MDW jobs?	<ul style="list-style-type: none"> • <i>OWS</i>: Run the sequencer and check if the job status of the sequencer is in Finished. Then do a data compare between the source and the target table and make sure the data is matching exactly to the result set you have. • <i>OWE</i>: Ensure the OWS data is populated first as the source for OWE jobs is OWS. Run the OWE job and make sure the job is in Finished status. Do a data compare with the source and target database to make sure the data that is populated in the target tables is matching to your expected result set. • <i>MDW</i>: Ensure the OWS data is populated first as the source for MDW job is OWS. In some cases, the source can be OWE and in such cases the OWE jobs must be executed first before running the MDW jobs. Run the MDW job and make sure the job is in 'Finished' status. Do a data compare with source and target database and check if the data populated in the target database is matching to your expected results.
How do I validate a job execution by verifying the data?	Perform a Database compare. This can be done by handwritten scripts or by third party DBCompare tools.

Common ETL Issues

This table provides answers to questions about common ETL issues.

Question	Answer
How do you debug a job that aborted or completed with warnings?	See Managing Aborted and Failed Jobs .
What if I drop and recreate an index?	Dropping and recreating an index would increase the ETL performance. However the downside of it is when there are any integrity constraints defined via indexes at the DB level and they not being handled in the ETL application. In such cases, data that gets loaded might not be cleaner and this might produce errors during recreating the index.

Question	Answer
<p>How do I resolve the following issue I receive when running any job in DataStage:</p> <p>I receive the error message - <i>Could not load drsoci.so</i> when pointed to an Oracle database and the DataStage server is UNIX.</p>	<p>Verify the dsenv file, which is a centralized file for storing environmental variables in the DataStage Server. It resides in \$DSHOME, where \$DSHOME identifies the DataStage main directory (for example /u1/dsadm/IBM WebSphere/DataStage/DSEngine).</p> <p>The dsenv file is a series of Bourne shell arguments, which are referenced during DataStage server startup and can be referenced by interactive users or other programs or scripts. For a connection using a non-wire protocol driver, you generally need to specify the following in the dsenv file:</p> <ul style="list-style-type: none"> • Environment variables required by the database client software • Database home location • Database library directory <p>Certain Plug-ins require shared libraries to be loaded and you need to include the library path in an environment variable. The names of the library path environment variables is platform dependent:</p> <ul style="list-style-type: none"> • Solaris Platform = LD_LIBRARY_PATH • HP-UX Platform = SHLIB_PATH • AIX Platform = LIBPATH • Compaq Tru64 Platform = LD_LIBRARY_PATH • LINUX Platform = LD_LIBRARY_PATH <p>The following provides typical entries for commonly used databases:</p> <pre># Oracle 8i ORACLE_HOME=/space/oracle8i ORAHOME=/space/oracle8i LD_LIBRARY_PATH=\$LD_LIBRARY_PATH:\$ORACLE_HOME/lib:\$ORACLE_HOME/rdbms/lib;export LD_LIBRARY_PATH ORACLE_SID=WSMK5 ORASID=WSMK5 export ORACLE_HOME ORAHOME ORACLE_SID ORASID</pre> <p>See <i>IBM Information Server: Planning Installation and Configuration Guide</i></p>

Question	Answer
<p>How do I resolve the following issue I receive when running any job in DataStage:</p> <p>I receive the error message - <i>Could not load drsdb2.so</i> when pointed to a DB2 database and the DataStage server is UNIX.</p>	<p>Verify the dsenv file, which is a centralised file for storing environmental variables in the DataStage Server. It resides in \$DSHOME, where \$DSHOME identifies the DataStage main directory (for example /u1/dsadm/IBM WebSphere/DataStage/DSEngine).</p> <p>The dsenv file is a series of Bourne shell arguments, which are referenced during DataStage server startup and can be referenced by interactive users or other programs or scripts. For a connection using a non-wire protocol driver, you generally need to specify the following in the dsenv file:</p> <ul style="list-style-type: none"> • Environment variables required by the database client software • Database home location • Database library directory <p>Certain Plug-ins require shared libraries to be loaded and you need to include the library path in an environment variable. The names of the library path environment variables is platform dependent:</p> <ul style="list-style-type: none"> • Solaris Platform = LD_LIBRARY_PATH • HP-UX Platform = SHLIB_PATH • AIX Platform = LIBPATH • Compaq Tru64 Platform = LD_LIBRARY_PATH • LINUX Platform = LD_LIBRARY_PATH <p>The following provides typical entries for commonly used databases:</p> <pre>#DB2 6.1 DB2DIR=/opt/IBMDB2/V6.1;export DB2DIR DB2INSTANCE=DB2inst1; export DB2INSTANCE INSTHOME=/export/home/DB2inst1;export INSTHOME PATH=\$PATH:\$INSTHOME/sqlib/bin:\$INSTHOME/sqlib/adm:\$INSTHOME/sqlib/misc export PATH LD_LIBRARY_PATH=\$LD_LIBRARY_PATH: \$INSTHOME/sqlib/lib;export LD_LIBRARY_PATH THREADS_FLAG=native;export THREADS_FLAG</pre>

Question	Answer
How do I switch to a new project when the warehouse tables have already been loaded with data?	<p>There might be a need to switch to new project when the warehouse tables have already been loaded for some time. In such cases, there is some project specific control data that must be restored onto the new project. For this purpose, it is always a good idea to backup this control data at some regular intervals of time after significant chunk of ETL loading gets completed.</p> <p>PeopleSoft delivers utilities that the backup/recovery process.</p> <p>Refer the section 'Running Datastage Project Utilities' and the following subsections in the PeopleSoft EPM Red Paper: ETL Implementation Steps (found in My Oracle Support) procedure to run these utilities:</p> <ul style="list-style-type: none"> • Backup_SurrogateKey_HashFile – Utility • Backup_DateTime_HashFiles – Utility • Recovery_DateTime_HashFiles – Utility • Recovery_SurrogateKey_HashFile – Utility
If a job aborts after half of the one million rows are written to the tables, what will happen?	If the Transaction size is selected as zero and if the job aborts in the middle, then the job will rollback the transactions since it follows the principle of Two-way commit. If the Transaction size is anything other than zero and if the job fails in the middle, then the job will perform commits for the number of rows that processed till the error message.
How do I report an issue with a job log for a job that completed with warnings or errors?	Report an issue to Global Customer Support. To report an issue, you must include the job log of the last run. In DataStage Designer, view the log for the job in detailed view mode. Select Project, Print from the menu. In the Print dialog box, select the <i>All entries</i> , <i>Full details</i> , and <i>Print to file</i> options. Click OK, name the file, and send the log along with your issue description and other pertinent information.

Configuring Delivered ETL Content

This table provides answers to questions about configuring and enhancing delivered ETL content.

Question	Answer
What are some of the configurations I may want to perform?	<ul style="list-style-type: none"> • Convert an incremental load job to a destructive load job. • Parameterize the hash file directory for user defined directories that store hashed files (rather than the default project directory). • Remove lookups that do not pertain to your business. • Configure jobs for initial load.

Question	Answer
How do I convert an incremental load job to a destructive load job?	See Converting Incremental Load Jobs to Destructive Load Jobs .
How do I run ETL jobs in destructive mode after some have already run as incremental loads?	See Converting Incremental Load Jobs to Destructive Load Jobs .
How do I add a new environmental variable?	See Adding New Environmental Variables .
What if I make a change that requires a new attribute in a dimension table?	If there is change in the data model with respect to a new addition of attribute to the EPM database, then you have to update the corresponding dimension job to incorporate this new attribute, otherwise the job will fail. If there is no source for this new attribute, then in the dimension job you can have a Default value assigned to this new attribute by using a routine that is delivered.
What if I make a change that requires a new dimension in a fact table?	If a new dimension key is added to a fact table in the database, then this is a change to the data model. Since the database has an additional dimension key for the fact table, this will result in changes to the ETL job. If this is a new dimension, then a new job has to be developed for this new dimension. Fact job must be updated accordingly with the correct dimension key and Corresponding SID population in the Fact table.
What if I make a change that requires a new measure in a fact table?	If a new measure is added to a fact table in the database, then this is a change to the data model. Since the database has an additional measure for the fact table, this will result in changes to the ETL job. Fact job must be updated accordingly with the correct measure getting assigned to the value that is either coming from the source directly or applying any logic that is required for this measure to be populated as per your requirements.
What if I make a change that requires a new dimension table?	A new ETL job has to be developed for this new dimension table as per the requirements.
What if I make a change that requires a new fact table?	A new ETL job has to be developed for this new fact table as per the requirements.
What are some techniques I can use to enhance ETL performance?	<ul style="list-style-type: none"> • <i>InterProcess Stage:</i> The IPC stage is used to implement pipeline parallelism. The IPC stage can be used to explicitly de-link two passive stage activities to run in separate processes. As a good practice an IPC stage can be inserted before a database write stage. • <i>Link Partitioner/Link Collector Stages:</i> The link partitioner and link collector can be used in conjunction to implement partition parallelism. Usage of link partitioner and link collector can be envisaged where specific requirement exists to further enhance job performance. This can be used most effectively to run in parallel processes that take a lot of time. The functionality that is to be run in parallel is typically identical and can be made into a local container. This way any changes to the common parallel part of the job need to be effected only at a single place.

Question	Answer
<p>How can I enhance the parallel processing capabilities of ETL server jobs?</p>	<ul style="list-style-type: none"> • <i>InterProcess Stage:</i> The IPC stage is used to implement pipeline parallelism. The IPC stage can be used to explicitly de-link two passive stage activities to run in separate processes. As a good practice an IPC stage can be inserted before a database write stage. • <i>Link Partitioner/Link Collector Stages:</i> The link partitioner and link collector can be used in conjunction to implement partition parallelism. Usage of link partitioner and link collector can be envisaged where specific requirement exists to further enhance job performance. This can be used most effectively to run in parallel processes that take a lot of time. The functionality that is to be run in parallel is typically identical and can be made into a local container. This way any changes to the common parallel part of the job need to be effected only at a single place. <p>See <i>WebSphere DataStage Development: Designer Client Guide</i></p>

Appendix C

ETL Reference Documents

DSX File Import Descriptions

Click the link to view the [DSX Files Import Description.pdf](#).

Environmental Parameters Information

Click the link to view the [List of Environment Parameters.pdf](#).

Parameter and Source Data Files Information

Click the link to view the [Parameter and Source data Files Information.pdf](#).

Routine Descriptions

Click the link to view the [EPM Routine Details.pdf](#).

Using the PeopleSoft EPM Lineage Spreadsheets

Understanding the EPM Lineage Spreadsheets

The EPM lineage spreadsheets provide information about the ETL jobs that are delivered with the EPM warehouses. The spreadsheets act like a reverse-engineering tool or family tree; they enable you to view the ancestry of source, target, and lookup tables and their relevant ETL jobs. Each spreadsheet provides lineage information for a single warehouse. The following table lists the lineage spreadsheets that are currently available:

Lineage Spreadsheet Filename	Warehouse
ETL_CS_Lineage_Spreadsheet.xls	Campus Solutions Warehouse
ETL_CRM_Lineage_Spreadsheet.xls	CRM Warehouse
ETL_FMS_Lineage_Spreadsheet.xls	FMS Warehouse
ETL_HCM_Lineage_Spreadsheet.xls	HCM Warehouse
ETL_SCM_Lineage_Spreadsheet.xls	SCM Warehouse

By using the spreadsheets, you can:

- View lineage information for staging, dimension, and fact ETL jobs, or source, target, and lookup tables.
- Identify the sequence of jobs to run for a specific data mart.
- Identify inter-mart and cross-warehouse dependencies.
- Generate lineage information for a specific ETL job.

Spreadsheet Structure

Each EPM lineage spreadsheet includes several worksheets. The following table provides a description of each worksheet, by name, listed in the order in which it appears:

Worksheet	Description
Template	This worksheet contains overview information, a legend, and a definition of the columns used in the worksheets.

Worksheet	Description
Setup	This worksheet contains ETL lineage information for all of the setup and staging jobs required for the warehouse.
Com Dims	This worksheet contains ETL lineage information for the common dimension jobs required for the warehouse.
Utils	This worksheet contains ETL lineage information for the currency conversion jobs required for the warehouse.
Global Dims	This worksheet contains ETL lineage information for the global dimension jobs required for the warehouse.
Local Dims	This worksheet contains ETL lineage information for the local dimension jobs required for the warehouse.
<Data Mart> For example: GL & Profitability, ESA, Campus Community, and so on.	This worksheet contains ETL lineage information for the jobs required for a specific data mart. <hr/> Note: Each spreadsheet includes several data mart worksheets. <hr/>
Dynamic_Lineage_Generator	This worksheet provides a macro that enables you to enter the name of an ETL job and automatically generate a list of the complete lineage for that job.
JobOrder	This worksheet is an extension of the Dynamic_Lineage_Generator worksheet. It displays the order in which jobs need to be run.

Column Descriptions

The following table provides descriptions of the columns in the worksheets.

Column	Description
Sequencer Job	The name of the job sequencer, which is responsible for invoking and running other ETL server jobs.
Server Job	The name of the server job that is called by the job sequencer.
Server Job Category	The location of the server job in the IBM WebSphere DataStage project.
Target Table	The name of the target table used in the server job.
Target Update Action	The target load strategy for the server job.
Source Table	The name of the source table used in the server job.

Column	Description
Source Extraction Type	The type of extraction from the source table in the server job (for example, incremental date time or cyclical redundancy check).
Lookup Tables	The name of the lookup tables that are used in the server job. Lookups can be hashed files or direct DRS lookups. The lineage information captures the table names from which the hash files are populated and the table names for the direct DRS lookup.
Setup Jobs	The name of the setup job that populates the source and/or the lookup table.
Setup Sequencer Job	The name of the job sequencer that calls the setup server job.
MDW	The name of the MDW server job. This column has an entry if the source table or lookup table is populated from an MDW server job.
MDW Sequencer	The name of the MDW sequence job.
OWS	The name of the OWS server job. This column has an entry if the source table or lookup tables are populated from an OWS server job.
OWS Sequencer	The name of the OWS sequence job.
OWE	The name of the OWE server job. This column has an entry if the source table or lookup tables are populated from an OWE server job.
OWE Sequencer	The name of the OWE sequence job.
EPM Foundation	The application or EPM foundation setup page that populates the source table or the lookup table, such as Global Consolidations, Dimension Mapper, or setup PIA pages.
Category	The categories in which the setup jobs, MDW jobs, OWS jobs or OWE jobs are placed.
Comments	Any additional comments, if applicable.

Note: The spreadsheet does not contain lineage details for OWE jobs and Tree jobs, except for the GL&Profitability Mart of the FMS warehouse, which does include lineage information for OWE jobs.

Viewing Lineage Information

This topic discusses how to use the spreadsheet to:

- Find lineage information for a server job.
- Identify the list of Jobs to be run for a data mart.

Finding Lineage Information for a Server Job

To find lineage information for a server job:

1. Access the worksheet in which the job is categorized.
2. Use Excel's Find feature to find the server job name in column B.
 - a. Type Ctrl-F to access the Find and Replace Dialog box.
 - b. Enter the name of the server job in the Find what edit box.
 - c. Click Find Next until the job name is found in the Server Job column (column B).
 - d. Close the Find dialog box.
3. Review the lineage information in the adjacent columns.

The Sequencer Job column (column A) lists the sequencer which calls this job. The Server Job Category column (column C) lists the category this job is associated with. The Target Table, Target Update Action, Source Table, and Source Extraction Type for this server job are listed in columns D, E, F, and G respectively. The Lookup Tables column (Column H) lists all the lookups used by this job.

The source tables and the lookup tables are placed in separate rows. This enables you to find the lineage information for each of these tables by navigating through the other subsequent columns in the same row. Columns I through R list the dependent jobs that are required to populate the source and lookup tables, and entries in these columns indicate whether the table is populated by Setup jobs, (column I), MDW jobs (column K), OWS jobs (column M), OWE jobs (column O), or Foundation setup / Apps (column Q). The Category column (column R) lists the category that the dependent job is associated with.

Source tables that are from a different data mart (inter-mart) or different warehouse (cross-warehouse) are indicated by the colors specified in the legend on the Template worksheet page.

The spreadsheet lists the lineage of a source or lookup table to the level of the job that directly populates it. The lineage information does not extend to level of the last staging job. To get the complete lineage for a fact or dimension job fully extended through the lowest staging level, you can use the dynamic lineage generator tool, which generates a list of all the required dependent jobs that need to be run in order to load a particular fact or dimension.

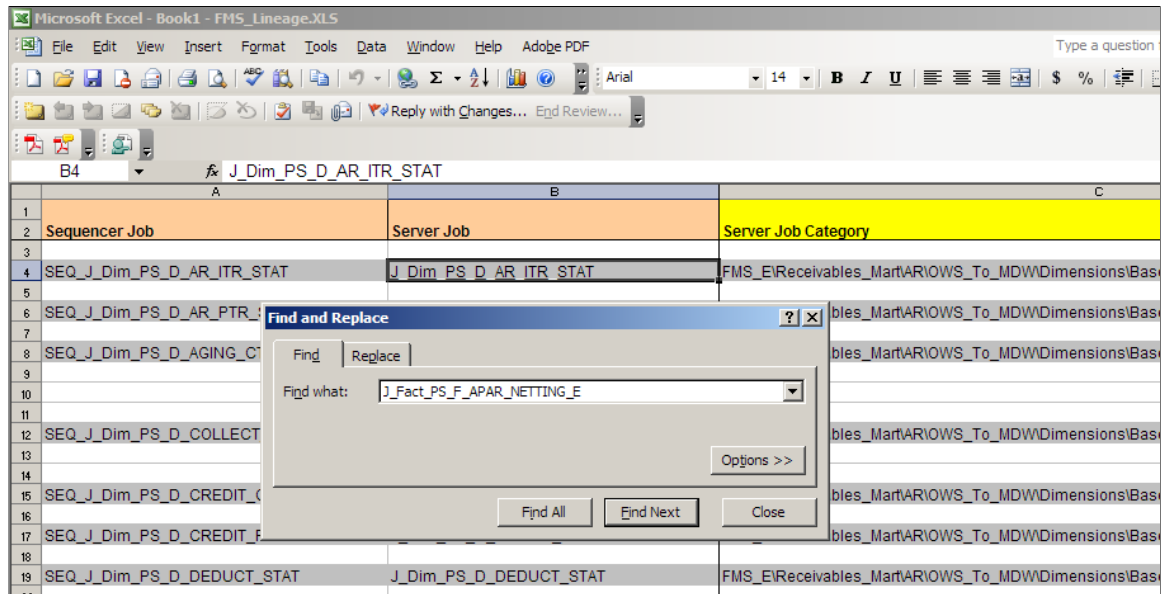
Example

This example, from the ETL FMS Lineage spreadsheet, takes you through the tasks you would complete to review the information for the fact job J_Fact_PS_F_APAR_NETTING_E, which is used for the AR Data Mart.

1. Navigate to the AR worksheet page.
2. Type Ctrl-F and type J_Fact_PS_F_APAR_NETTING_E into the Find and Replace dialog box.

Image: Find and Replace Dialog Box

This example illustrates the Find and Replace Dialog Box.



3. Type Ctrl-F and type J_Fact_PS_F_APAR_NETTING_E into the Find and Replace dialog box.
4. Click Find Next until you access the cell in the Server Job column that contains the J_Fact_PS_F_APAR_NETTING_E job.

- Close the Find and Replace dialog box. You should see the following information:

Image: J_Fact_PS_F_APAR_NETTING_E job displayed in spreadsheet

This example illustrates the J_Fact_PS_F_APAR_NETTING_E job displayed in spreadsheet. .

	A	B	
1			
2	Sequencer Job	Server Job	Server Job Category
90			
91	SEQ_Dims_L_O_CREDIT_CLASS	J_Dim_PS_O_CREDIT_CLASS	FMS_E\Receivables_MartAR\OWS_T
92			
93			
94	SEQ_Dims_L_O_CREDIT_RISK	J_Dim_PS_O_CREDIT_RISK	FMS_E\Receivables_MartAR\OWS_T
95			
96			
97	SEQ_Dims_L_O_DEDUCT_STAT	J_Dim_PS_O_DEDUCT_STAT	FMS_E\Receivables_MartAR\OWS_T
98			
99			
100	SEQ_Dims_L_O_DISPUTE_STAT	J_Dim_PS_O_DISPUTE_STAT	FMS_E\Receivables_MartAR\OWS_T
101			
102			
103	SEQ_Dims_L_O_ENTRY_RSTYP	J_Dim_PS_O_ENTRY_RSTYP	FMS_E\Receivables_MartAR\OWS_T
104			
105			
106	SEQ_J_Fact_PS_F_APAR_NETTING_E	J_Fact_PS_F_APAR_NETTING_E	FMS_E\Receivables_MartAR\OWS_T
107			
108			
109			
110			
111	SEQ_J_Fact_PS_F_AR_ACCOUNT_LN_E	J_Fact_PS_F_AR_ACCOUNT_LN_E_ITEM	FMS_E\Receivables_MartAR\OWS_T

- Scroll to the right to review the columns shown here:

Image: Reviewing data associated with the J_Fact_PS_F_APAR_NETTING_E job

This example illustrates the Reviewing data associated with the J_Fact_PS_F_APAR_NETTING_E job.

D	E	F	G
Target Table	Target Update Action	Source Table	Source Extraction Type
		PS_D_ENTRY_RSTYP	
PS_F_APAR_NETTING	Truncate table then insert rows	PS_D_CUST_ORG	
		PS_D_SUPPLIER	
		PS_F_AR_AGING	
PS_F_AR_ACCOUNT_LN	Insert new rows or update existing ones	PS_ITEM_DST	DateTime Incremental
		PS_ITEM	

The Target Table, Target Update Action, Source Table, and Source Extraction Type for the J_Fact_PS_F_APAR_NETTING_E server job are listed in columns D, E, F, and G, respectively.

- Continue to scroll to the right to view the remaining columns.

The Lookup Tables column (Column H) lists all the lookups used in J_Fact_PS_F_APAR_NETTING_E.

Image: Lookup Tables Column

This example illustrates the Lookup Tables Column.

F	G	H
Source Table	Source Extraction Type	Lookup Tables
PS_D_CUST_ORG		
PS_D_SUPPLIER		
PS_F_AR_AGING		PS_F_AP_AGING
PS_ITEM_DST	DateTime Incremental	
PS_ITEM		

In this example there are three source tables: PS_D_CUST_ORG, PS_D_SUPPLIER, PS_F_AR_AGING. The lookup table is PS_F_AP_AGING. The source tables and the lookup tables are each placed in a unique row one after the other. This enables you to view the lineage information for each of these tables by navigating through the succeeding columns within the same row.

Columns I through R list out the dependent jobs required to populate these source and lookup tables. In this example, the source table PS_D_CUST_ORG has an entry in the MDW column, which means that it is populated from the MDW dimension J_Dim_PS_D_CUST_ORG_SCM, which is placed in the category Global_Dimensions_E\OWS_To_MDW\Base\Load_Tables\Server.

As shown in the following screenshot, the source table PS_D_SUPPLIER is an SCM warehouse dimension. The cross-warehouse dependency is identified by the different color (the color legend is located on the first worksheet page).

Image: Cross-warehouse dependencies for PS_D_SUPPLIER

This example illustrates the Cross-warehouse dependencies for PS_D_SUPPLIER.

F	L	R
Source Table	MDW Sequencer	Category
PS_D_CUST_ORG	SEQ_J_Dim_PS_D_CUST_ORG_SCM	Global_Dimensions_E\OWS_To_MDW\Base\Load_Tables\Server
PS_D_SUPPLIER	SEQ_J_Dim_PS_D_SUPPLIER	Global_Dimensions_E\OWS_To_MDW\Base\Load_Tables\Server
PS_F_AR_AGING	SEQ_J_Fact_PS_F_AR_AGING_E	FMS_E\Receivables_Mart\OWS_To_MDW\Facts\Base\Load_Tables\Server
	SEQ_J_Fact_PS_F_AP_AGING_E	FMS_E\Payables_Mart\OWS_To_MDW\Facts\Base\Load_Tables\Server

Similarly, the lookup table PS_F_AP_AGING is populated from the fact job J_Fact_PS_F_AP_AGING placed in the category FMS_E\Payables_Mart\AP\OWS_To_MDW\Facts\Base\Load_Tables\Server. This fact job belongs to a different mart as indicated by the different color.

Image: Cross-warehouse dependencies for PS_F_AP_AGING

This example illustrates the Cross-warehouse dependencies for PS_F_AP_AGING.

DEPENDENCIES: ETL jobs which populate the source and the lookup			
Lookup Tables	MDW	MDW Sequencer	Category
	J_Dim_PS_D_CUST_ORG_SCM	SEQ_J_Dim_PS_D_CUST_ORG_SCM	Global_Dimensions_EIOWS_To_MDW\Base\Load_Tables
	J_Dim_PS_D_SUPPLIER	SEQ_J_Dim_PS_D_SUPPLIER	Global_Dimensions_EIOWS_To_MDW\Base\Load_Tables
	J_Fact_PS_F_AR_AGING_E	SEQ_J_Fact_PS_F_AR_AGING_E	FMS_E\Receivables_Mart\AP\OWS_To_MDW\Facts\Base
PS_F_AP_AGING	J_Fact_PS_F_AP_AGING	SEQ_J_Fact_PS_F_AP_AGING_E	FMS_E\Payables_Mart\AP\OWS_To_MDW\Facts\Base\Load

Identifying the List of Jobs to be Run for a Data Mart

You can use the information in the spreadsheet to identify the list of jobs that need to be run for a specific data mart. These include common jobs that are required for every data mart, which we refer to as prerequisite jobs, as well as jobs specific to the particular data mart.

If you prefer, you can create your own master sequencers based on the information provided in this topic.

Alternatively, you can generate the list of jobs by using the Dynamic Lineage Generator tool. For more information, see "Generating Lineage Information for a Job".

Note: All the server jobs relating to Hash files that are present within the Load_Hash_Files category need to be run first before running other Sequence jobs within the Load_Tables category since these hash files are being used in other server jobs.

Prerequisite Jobs

The prerequisite jobs include setup jobs, staging jobs, and dimension jobs.

The following sets of jobs need to be run for every mart, in the order that they are listed in the worksheets:

1. Run these setup jobs in the Setup worksheet:
 - a. All jobs within the Setup_E\OWS\<Warehouse> category.
(For example all jobs within the Setup_E\OWS\FSCM category for the FMS warehouse and all jobs within the Setup_E\OWS\CS category for the CS warehouse).
 - b. All jobs within the Setup_E\Dimension mapper category.

Note: Please ensure that you run the Business Unit Wizard before proceeding with the following steps.

See [Understanding Warehouse Business Units](#), [TableSet Sharing](#), and [SetID Mapping and Understanding Warehouse Business Unit Setup](#).

- c. All jobs within the Shared_Lookups\DimensionMapper_Lookups category.
- d. All jobs within the Shared_Lookups\Control_Tables category.

- e. All jobs within the Shared_Lookups\System_Lookups category.
 - f. All jobs within the Shared_Lookups\Language_Lookups category.
 - g. All jobs within the Setup_E\OWE category (this step does not apply to the Campus Solutions warehouse).
 - h. If you are implementing currency conversion, then run the jobs listed in the Utils worksheet.
2. Run the staging jobs listed in the OWS Sequencer column (column N) in the following worksheets:
 - a. Com Dims.
 - b. Global Dims.
 - c. Local Dims.
 - d. <Data Mart>, where <Data Mart> is the name of the data mart, for example AP, AR, Campus Community, Student Financials .
 3. Run the Common Dimension Jobs listed in the Com Dims worksheet.
 4. Run the Global Dimensions jobs listed in the Global Dims worksheet. (These jobs are required for running the FMS warehouse jobs.)
 5. Run the Local Dimension Jobs placed in the Local Dims worksheet.

Data Mart Specific Jobs

Run all the Server jobs listed in column B of the worksheet for the specific data mart, to populate the corresponding Dimension and Fact tables for that mart.

Note: Do not run the jobs that are listed within the Reusable Jobs category. These jobs are not used to load target tables. They are automatically triggered by various Sequence jobs.

Generating Lineage Information for a Job

The Dynamic_Lineage_Generator worksheet contains a macro that generates a list of all the dependent jobs that are required for any ETL job. This will easily help you identify all the list of jobs to be run for a specific fact or dimension job.

To use the Dynamic Lineage Generator:

1. Access the Dynamic_Lineage_Generator worksheet.
2. Enter the job name in cell B1.
3. Click the Get Job Lineage button.

The macro retrieves the lineage required for running this fact job from the setup, staging, and the dimension jobs and displays it in the cells below. The macro also copies the entire list of dependent jobs to the JobOrder worksheet, so you can identify the complete list to be run in sequence.

You must run the following prerequisite setup jobs before you run the jobs listed in the JobOrder worksheet:

- Setup_E\OWS*Warehouse Name*> Job Sequencer.

For example Setup_E\OWS\FSCM Job Sequencer or Setup_E\OWS\CS Job Sequencer.

- Setup_E\Dimension mapper Job Sequencer.
- Run the Business Unit Wizard to populate the Dimension mapper tables.

See [Understanding Warehouse Business Units, TableSet Sharing, and SetID Mapping and Understanding Warehouse Business Unit Setup](#).

- Shared_Lookups\DimensionMapper_Lookups
- Shared_Lookups\Control_Tables
- Shared_Lookups\System_Lookups
- Shared_Lookups\Language_Lookups
- Setup_E\OWE Job Sequencer (this step does not apply to the Campus Solutions warehouse).

After you run the prerequisite setup jobs, then run the jobs listed in the JobOrder worksheet.