GAMMASYSTEMS

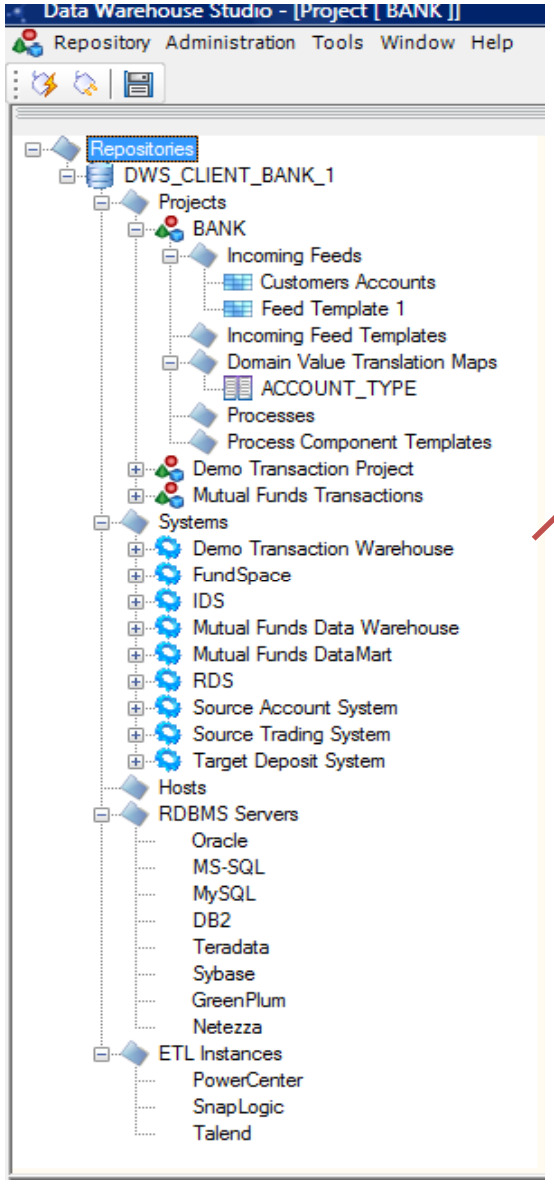
Gamma Data Warehouse Studio

Streamlined Implementation of

- Data Warehouses
- Data Marts
- Data Integration Projects

Gamma Data Warehouse Studio
Feature Highlights

DW Studio – Library of Reusable Objects, Systems, Feeds, Projects, Value Maps



DW Studio allows users to create or import table, view and file definitions, logically group them, and reuse the definitions as sources or targets in data mapping screens. Along with this, DWS facilitates creation of business and technical specifications and rules that can be easily reused across multiple projects.

GAMMASYSTEMS

DW Studio - Data Mapping Screen

Data Warehouse Studio - [Incoming Feed [Customers Accounts]]

Repository Administration Tools Window Help

- Repositories
  - DWS_CLIENT_BANK_1
    - Projects
      - BANK
        - Incoming Feeds
          - Customers Accounts
          - Feed Template 1
        - Incoming Feed Templates
        - Domain Value Translation Maps
        - Processes
        - Process Component Templates
      - Demo Transaction Project
      - Mutual Funds Transactions
    - Systems
      - Demo Transaction Warehouse
      - FundSpace
      - IDS
      - Mutual Funds Data Warehouse
      - Mutual Funds DataMart
      - RDS
      - Source Account System
        - Core Objects
          - CUSTOMER_ACCOUNTS
        - Mapping Layer Objects
        - Queries
      - Source Trading System
      - Target Deposit System
        - Core Objects
        - Mapping Layer Objects
          - IN_CUSTOMER_BALANCE
        - Queries
    - Hosts
    - RDBMS Servers
    - ETL Instances

General | Data Mapping | Data Quality Checks | Processing Specifications

Source Objects

Objects / Columns
CUSTOMER_ACCOUNTS
customer_id
customer_first_name
customer_last_name
account_nbr
account_balance
overdraft_code
protect_code
checking_acct_ind
account_type

Reusable Expressions

Target Objects — Hide Non-Mappable Columns

Objects / Columns	Expression	Default Value	Group By	Aggregation
IN_CUSTOMER_BALANCE				
CUST_FNAME	CUSTOMER_ACCOUNTS.customer_id	"N/A"		
CUST_ID	UPPER(CUSTOMER_ACCOUNTS.customer_id)			
CUST_LNAME	CUSTOMER_ACCOUNTS.customer_first_name			
ACCT_BALANCE	CUSTOMER_ACCOUNTS.account_balance			
ACCT_NUMBER	"ABC" + CUSTOMER_ACCOUNTS.account_nbr			
OVERDRAFT_PROTECT_CHK_SAV				
ACCT_TYPE_CODE	MAP ('ACCOUNT_TYPE', CUSTOMER_ACCOUNTS.ac			

Total mappable fields: 7

5/13/2010 8:13:01 PM REPO Connected to Repository DWS_CLIENT_BANK_1

DW Studio provides an easy way to manipulate data mapping specifications. Users can drag objects into the Source or the Target sections of the screen from the left hand-side pane. DWS hides all technical fields for targets, so the data mappers do not need to be concerned with technical activities such as defining the logic for creating primary and foreign keys or populating audit columns.

Besides the data mapping rules, DWS allows users to enter grouping specifications, data filters, default values, valid value lists and many other features that data mappers commonly use.

DW Studio allows users to specify reusable expressions that can be used across many target fields

GAMMASYSTEMS

DW Studio - Data Mapping Screen – Multiple Source and Target Objects

General | Data Mapping | Data Quality Checks | Processing Specifications

Source Objects | Target Objects | Hide Non-Mappable Columns

Objects / Columns
CAMPAIGN_ACTIVITY
ACTIVITY_ID
CAMPAIGN_ID
DESCRIPTION
ACTIVITY_TYPE
START_DATE
END_DATE
PARAM1
PARAM2
PARAM3
EST_NUM
EST_COST
ACT_NUM
ACT_COST
EST_HRS
ACT_HRS
SALES_METHOD_ID
CREATED_DT
UPDATED_DT
CREATED_BY
UPDATED_BY
STATUS
ACT_COST_PER_UNIT
CAMPAIGN_RESPONSE_ACTIONS
RESPONSE_CODE
RESPONSE_DESC
FOLLOWUP_EXT_YN
FOLLOWUP_INT_YN

Reusable Expressions

Edit Header
Edit Columns
Remove Object
Set Filter

Objects / Columns	Expression	Default Value	Group By	Aggregation
API_DIM_PARTY_SCHED_FOR				
PARTY_TYPE			✓	
PARTY_PRIMARY_ROLE			✓	
PARTY_NAME	IFOR	"N/A"	✓	
PARTY_FIRST_NAME				Max
PARTY_LAST_NAME				Max
PARTY_DESC		"N/A"		Max
PARTY_SHORT_NAME	CALSCHED.SCHFOR		✓	
ADDRESS_LINE_1				Max
ADDRESS_LINE_2				Max
ADDRESS_LINE_3				Max
CITY				Max
STATE_CODE				Max
POSTAL_CODE				Max
COUNTRY_CODE				Max
CRM_STATUS_CODE				Max
MAIL_STATUS_FLAG				First
OFFICE_CODE				Max
PHONE_NUMBER				Max
EMAIL_ADDRESS				Max
API_FACT_ACTIVITY				
PRODUCT_GROUP_IDENTIFIER				
ACTIVITY_PARTY_GROUP_IDENTIFIER	Q_CALSCHED_CONTACT.CONTACT_LIST_GROU	"N/A"		
ACTIVITY_SRC_ID	CALSCHED.SCHUNQUE			
CAMPAIGN_ACTIVITY_ID_SRC	CAMPAIGN_ACTIVITY.ACTIVITY_ID	"N/A"		
OPPORTUNITY_SRC_ID	ISNULL(CALSCHED.OPPORTUNITY_ID, "N/A", CA			
LIT_ORDER_LIST_KEY_SRC	CALSCHED.LIT_ORD_LIST_KEY	"N/A"		
LIT_ORDER_PARTY_SHORT_NAME	"N/A"			
LIT_ORDER_PARTY_PRIMARY_ROLE	"FS_USER"			
ACTIVITY_START_FULL_DATE	CALSCHED.SCHSTART			

d fields: 28
unmapped fields: 0

DW Studio is capable of handling multiple sources or targets within a single data map. Filtering rules can be specified if needed for each source or target individually.

Multiple source objects can be linked together by specifying a linkage condition

GAMMASYSTEMS

DW Studio - Field Mapping – Straightforward Source-Target Drag

General | Data Mapping | Data Quality Checks | Processing Specifications

Source Objects

Objects / Columns
CUSTOMER_ACCOUNTS
customer_id
customer_first_name
customer_last_name
account_nbr
account_balance
overdraft_code
protect_code
checking_acct_ind
account_type

Target Objects — Hide Non-Mappable Columns

Objects / Columns	Data Type	Expression	Default Value	Group By	Aggregation
IN_CUSTOMER_BALANCE					
CUST_FNAME	varchar(50)	CUSTOMER_ACCOUNTS.cu	"N/A"		
CUST_ID	varchar(30)	UPPER(CUSTOMER_ACCC			
CUST_LNAME	varchar(80)	CUSTOMER_ACCOUNTS.cu			
ACCT_BALANCE	decimal(10,2)	CUSTOMER_ACCOUNTS.ac			
ACCT_NUMBER	varchar(40)	"ABC" + CUSTOMER_ACCI			
OVERDRAFT_PROTECT_CHK_SAV	varchar(10)				
ACCT_TYPE_CODE	STRING(10)	MAP ('ACCOUNT_TYPE', C			

There are different ways to map source fields to target fields. The simplest way is to drag a source column and drop it into a target field

Reusable Expressions

Total mappable fields: 7 — Unmapped fields: 1 — Mandatory unmapped fields: 0

GAMMASYSTEMS

DW Studio - Field Mapping – Expression

General | Data Mapping | Data Quality Checks | Processing Specifications

Source Objects | Target Objects | Hide Non-Mappable Columns

Objects / Columns	Objects / Columns	Expression	Default Value	Group By	Aggregation
CUSTOMER_ACCOUNTS	IN_CUSTOMER_BALANCE				
customer_id	CUST_FNAME	CUSTOMER_ACCOUNTS.customer_id	"N/A"		
customer_first_name	CUST_ID	UPPER(CUSTOMER_ACCOUNTS.customer_id)			
customer_last_name	CUST_LNAME	CUSTOMER_ACCOUNTS.customer_first_name			

Enter Expression for IN_CUSTOMER_BALANCE.CUST_ID

Columns | Functions | Reusable Expressions

- Date
- Numeric
  - ABS
  - CEIL
  - EXP
  - FLOOR
  - LOG
  - MOD
  - POWER
  - ROUND
  - SQRT
- Special
- String
  - ASCII
  - CHAR
  - INITCAP
  - LEFT
  - LENGTH
  - LOWER
  - POS
  - RIGHT

LENGTH(string), Returns NUMBER

Returns the length of the string.

Business Rule | Overriding Engine Expression

UPPER(CUSTOMER_ACCOUNTS.customer_id) + "ABC"

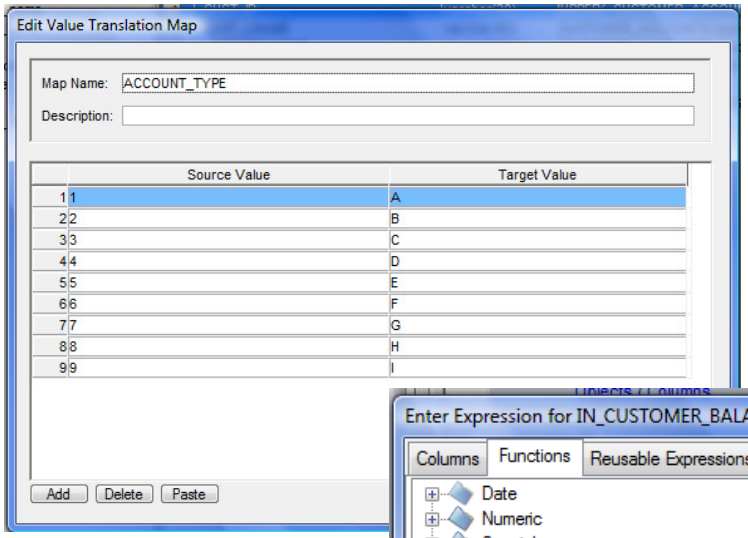
1 2 3 4 5 6 7 8 9 0 .

AND OR NOT + - < > = != * / <= >= || ^

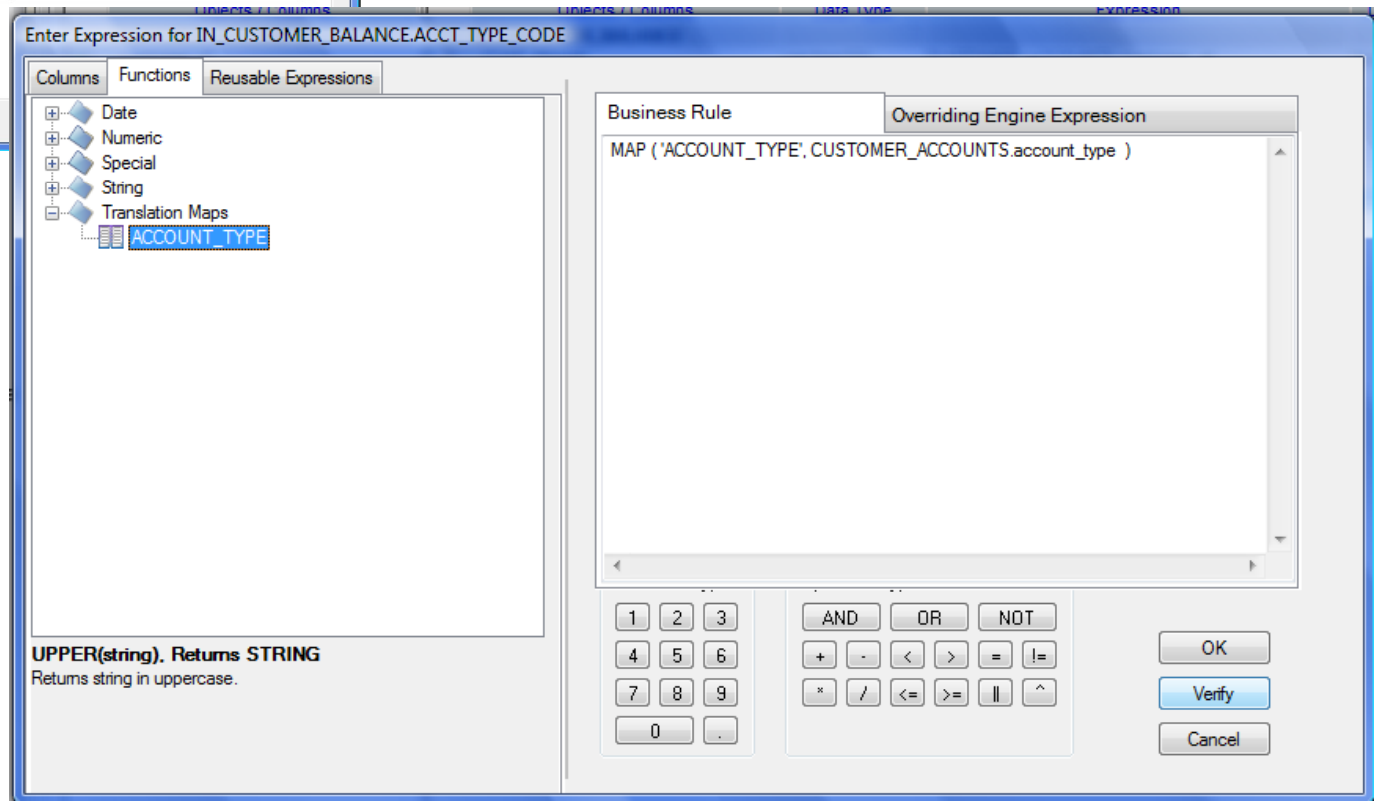
OK | Verify | Cancel

To enter more complex mapping specifications, users can use the expression editor. Excel-like functions with online help are available for usage in expressions Users can also utilize pre-defined custom functions that may be available for use.

DW Studio – Field Mapping - Source-Target Value Mapping Functionality



To map values from source systems to target fields, DW Studio's value mapping functionality could be utilized. Users can define reusable value maps and plug them into expressions.



GAMMASYSTEMS

DW Studio - Field Mapping – Conditional Expressions

Repository Administration Tools Window Help

Repositories
DWS_CLIENT_BANK_1
Projects
BANK
Incoming Feeds
Customers A
Feed Templa
Incoming Feed T
Domain Value Tr
ACCOUNT_
Processes
Process Compor
Demo Transaction P
Mutual Funds Transa
Systems
Demo Transaction W
FundSpace
IDS
Mutual Funds Data W
Mutual Funds DataM
RDS
Source Account Sys
Source Trading Syst
Target Deposit Syste
Hosts
RDBMS Servers
Oracle
MS-SQL
MySQL
DB2
Teradata
Sybase
GreenPlum
Netezza
ETL Instances
PowerCenter
SnapLogic
Talend

General | Data Mapping | Data Quality Checks | Processing Specifications

Source Objects | Target Objects | Hide Non-Mappable Columns

Objects / Columns | Objects / Columns | Data Type | Expression | Default Value | Group By | Aggrega

Enter Expression for IN_CUSTOMER_BALANCE.OVERDRAFT_PROTECT_CHK_SAV

Columns | Functions | Reusable Expressions

Sources
CUSTOMER_ACCOUNTS
customer_id
customer_first_name
customer_last_name
account_nbr
account_balance
overdraft_code
protect_code
checking_acct_ind
account_type

Business Rule | Overriding Engine Expression

```
IF (
CUSTOMER_ACCOUNTS.overdraft_code = 8
 AND
CUSTOMER_ACCOUNTS.protect_code = 9
 AND
CUSTOMER_ACCOUNTS.checking_acct_ind = 1,

1,
0
)
```

1 2 3 4 5 6 7 8 9 0 . AND + - * /

Reusable Expressions

Total mappable fields: 7 | Unmapped fields: 1 | Mandatory unmapped fields: 0

DW Studio supports conditional expressions, where the resulting value for each target field is determined by evaluating other fields

GAMMASYSTEMS

DW Studio – Exception Checks

General | Data Mapping | Data Quality Checks | Processing Specifications

	Exception Condition	Description	Exception	Severity Level
1	checking_acct_ind is null		Overdraft Code is Blank	High severity

Add | Delete

Along with supporting the data mapping activities, DW Studio allows users to define exception conditions and assign severity level to these conditions

GAMMASYSTEMS

DW Studio – Main Project Definition Screen

Repository Administration Tools Window Help

- Repositories
  - DWS_CLIENT_BANK_1
    - Projects
      - BANK
      - Demo Transaction Proje
      - Mutual Funds Transactic
    - Systems
    - Hosts
    - RDBMS Servers
    - ETL Instances

General | Admin | Subject Areas | Exception Management | Technical Specifications | Environments | Custom Transformation Functions

Project Information

Project Name: BANK

Project Code: B1

Description: This is a demo project for BANK1.

The project contains feeds that demonstrate the functionality of DW Studio.

Central System: Target Deposit System

DW Studio can group multiple data mapping specifications under a single project and reuse these collections of maps from project to project

GAMMASYSTEMS

DW Studio – Use Role Assignment Screen

Data Warehouse Studio - [Project [BANK]]

Repository Administration Tools Window Help

Repositories
- DWS_CLIENT_BANK_1
  - Projects
    - BANK
    - Demo Transaction Proje
    - Mutual Funds Transactic
  - Systems
  - Hosts
  - RDBMS Servers
  - ETL Instances

General | Admin | Subject Areas | Exception Management | Technical Specifications | Environments | Custom Transformation Functions

User Roles
- Project Administrators
- Subject Area Definitions
- Exceptions Definitions
- Data Map
- Architectu
- Environm
- Operation
- Deployme
- Custom F

Assign Privileges for Project Administrators

Available Users & Groups

Users & Groups
Administrator
Administrators
AGIR_DW
Nag
Public
Sriram
Venu
Vipul

>> <<

Selected Users & Groups

Users & Groups	Privileges
Simon E.	Read/Write Read Only
Warren	Read/Write Read Only

Save Close

DW Studio can help manage access to project and data mapping definitions through its user management functionality

GAMMASYSTEMS

DW Studio – Exception Definition Screens

General | Admin | Subject Areas | Exception Management | Technical Specifications | Environments | Custom Transformation Functions

Exception Severity | Exception Definitions

Severity Level	Severity Code	Description	Action
1	HIGH	High severity	Reject the Row
2	LOW	Low severity	Allow Row to Pass Through

General | Admin | Subject Areas | Exception Management | Technical Specifications | Environments | Custom Transformation Functions

Exception Severity | Exception Definitions

	Exception Code	Exception Name	Exception Description
1	EX01	Blank Last Name	
2	EX02	Overdraft Code is Blank	

DW Studio allows definitions of project-wide exception rules

GAMMASYSTEMS

DW Studio – Project-Level Technical Configuration Screen

Data Warehouse Studio - [Project [BANK]]

Repository Administration Tools Window Help

Repositories

General | Admin | Subject Areas | Exception Management | Technical Specifications | Environments | Custom Transformation Functions

Inbound Feed Architecture | Processing Standards | Stats & Logging | Talend

Inbound Feed Process Flow

Use Landing Area

Use Staging Tables — Populate With: ETL — STEP_1_STAGE_%FEED_NAME%

Perform Data Quality Checks — Using: Stored Procedure — STEP_2_QUALITY_CHECK_%FEED_NAME%

Materialize Mapping Layer — Populate With: ETL — STEP_3_CONFORM_%FEED_NAME%

Target Structures Populated by: — Populate With: Stored Procedures — STEP_4_TARGET_LOAD_%FEED_NAME%

ETL Tool: Talend (PowerCenter, SnapLogic, Talend)

Wrap Stored Procedures into ETL Jobs

Pre-Delete Instance Data from Intermediary Tables

Data Staging

Staging Table Prefix: STG_

Domain Value Translation Table

Table Schema: API

Table Name: BANK_TARGET_TRANSL

Apply Specifications

If an ETL technology has been selected, DW Studio further prompts for ETL technology-specific options.
Note: ETL is an optional component for DW Studio – DW Studio can generate shell and SQL scripts if the customer prefers not to involve an ETL tool.

DW Studio allows architects to enter project-level technical specs that will be applied at the code generation phase to each feed. DW Studio eliminates the need to re-code these specs for each feed. DWS ensures full adherence to all architectural preferences, technology standards, and naming conventions set for the project

GAMMASYSTEMS

DW Studio – Project-Level Talend Configuration Screen

Data Warehouse Studio - [Project [BANK]]

Repository Administration Tools Window Help

- Repositories
  - DWS_CLIENT_BANK_1
    - Projects
      - BANK
        - Incoming Feeds
          - Customer Accounts 1
        - Incoming Feed Templates
        - Domain Value Translation Maps
          - ACCOUNT_TYPE
        - Processes
        - Process Component Templates
      - Demo Transaction Project
      - Mutual Funds Transactions
    - Systems
    - Hosts
    - RDBMS Servers
    - ETL Instances

General | Admin | Subject Areas | Exception Management | Technical Specifications | Environments | Custom Transformation Functions

Inbound Feed Architecture | Processing Standards | Stats & Logging | Talend

Talend Project Level Settings

Project Name: PROJECT_BANK

Code Generation Language: Java

Talend Version: 3.2.0+

Talend Repository Settings

Root Folder Name: BANK_ROOT

Job Folder Names: Subject Area/Feed Name

Job Folder Name Pattern:

Job Folder Name Case: Upper

Context Group Settings

Param Context Group Name: JOB_PARAMS

Connections Name Pattern: CONN_%SYSTEM_CODE%

Context Name: Execution

Stats/Logging Context Group: STATS_INFO

☑ Load Context Values from File

Object Connections

○ Defined Individually (Built-In)

○ Inherited From DB Connection Metadata (Repository)

◉ Connection Context (Suitable for multiple schemas)

Process Flow Management

☑ Generate Container Job

MASTER_%FEED_NAME%

☐ Generate Reusable Joblets for Populating Target Tables

☐ Use Bulk-Load Inserts by Default

Object Definitions

○ Defined Individually (Built-In)

◉ Centralized Definition (Repository)

Flat File & Bulk Load Options

☑ Directory Structure Context Group

5/8/2010 9:51:08 PM | REPO | Connected to Repository DWS_CLIENT_BANK_1

DW Studio prompts for technical specs that are relevant for Talend, which will be used at the code generation phase of the project

GAMMASYSTEMS

DW Studio – Project-Level Informatica Configuration Screen

Data Warehouse Studio - [Project [BANK]]

Repository Administration Tools Window Help

- Repositories
  - DWS_CLIENT_BANK_1
    - Projects
      - BANK
        - Incoming Feeds
        - Incoming Feed Templates
        - Domain Value Translation Maps
        - Processes
        - Process Component Templates
      - Demo Transaction Project
      - Mutual Funds Transactions
    - Systems
    - Hosts
    - RDBMS Servers
    - ETL Instances

General | Admin | Subject Areas | Exception Management | Technical Specifications | Environments | Cu

Inbound Feed Architecture | Processing Standards | Stats & Logging | PowerCenter

PowerCenter Folder Organization

Source / Target Definition Folders:	System-Specific
Landing Process Folders:	System-Specific
Staging Population Folders:	Subject Area
API Process Folders:	Subject Area
Target Population Folders:	System-Specific
System-Specific Folder Naming:	Shared_%SYSTEM%
Subject Area Folder Naming:	SA_%SUBJECT_AREA%
Common Definitions Folder:	INFA_COMMON

Object Naming Standards

Mapping Prefix:	m_
Session Prefix:	s_
Workflow Prefix:	wk_
Transformation Name Case:	Upper Case
Database Connection Profile Naming:	%LOGICAL_NAME%
Source-to-Target Conn. Profile Name:	CONN_LOOKUP

DW Studio prompts for technical specs that are relevant for Informatica, which will be used at the code generation phase of the project

GAMMASYSTEMS

DW Studio – Code Generation

Data Warehouse Studio - [Code Deployment [Mutual Funds Transactions]]

Repository Administration Tools Window Help

Repositories
- DWS_CLIENT_BANK_1
  - Projects
    - BANK
    - Demo Transaction Project
    - Mutual Funds Transactions
  - Systems
  - Hosts
  - RDBMS Servers
  - ETL Instances

Edit Project
Delete Project
Documentation...
Code Deployment ▸ Into DEV
Process Monitor

Deployment Information
Environment: Development
Comment: Status: NOT STARTED
Code deployment on 05/20/2010
Start: End:

Deployment Components
- FundSpace Activity Fact
  - STEP_10_STAGING_FundSpace_Calsched_Fact
  - STEP_20_QUALITY_CHECK_FundSpace_Calsche
  - STEP_30_API_FundSpace_Calsched_Fact
  - STEP_40_TARGET_FundSpace_Calsched_Fact
- FundSpace Calsched Contact Group
- FundSpace Calsched Product Group
  - STEP_10_STAGING_FundSpace_Calsched_Produ
  - STEP_20_QUALITY_CHECK_FundSpace_Calsche
  - STEP_30_API_FundSpace_Calsched_Product_Gr
  - STEP_40_TARGET_FundSpace_Calsched_Produ
- FundSpace Campaign Contact Group
- FundSpace Campaign Dim
- FundSpace Campaign Response Action
- FundSpace Distribution Hierarchy
- FundSpace DSM
- FundSpace Lit Inventory
- FundSpace Lit Inventory Kit
- FundSpace Lit Order Fact
- FundSpace Lit Order Header
- FundSpace Master Firm
- FundSpace NSM
- FundSpace Opportunity Contact Group
- FundSpace Opportunity Fact

Process-Level Deployment Specifications
- (•) Incremental (Changed and New processes)
- ( ) All Processes
- ( ) Subject Area Specific

Component-Level Deployment Specifications
- [x] All Components
- [x] Landing Components
- [x] Quality Check Components
- [x] Staging Components
- [x] API Components
- [x] Target Loading Components

ETL Code Deployment
- [x] ETL Code
- [x] Reusable Objects

Intermediary Table Options
- [ ] Generate "Drop Table" Statements
- [x] Generate "Create Index" Statements
- [x] Generate "Grant Privileges" Statements

Mapping-Layer-to-Target Processes
- [x] Generate Reusable Components
  - (•) Deployed Processes Only
  - ( ) Entire API Layer

Retrieve Components | Deploy Components | DDL...

Generating code in DW Studio takes a click of a button – simply choose the target environment and components you want to generate, and then click the **Deploy Components** button.
DW Studio generates ETL code, Stored procedures, and all supporting and intermediary tables required by the generated jobs

View generated DDLs, SQL, and Stored Procedures

GAMMASYSTEMS

DW Studio – Generated SQL and DDL Statements

Gamma Data Warehouse Studio - [Code Deployment [Mutual Funds Datamart]]

Repository Administration Tools Window Help

- Repositories
  - DWS_CLIENT_BANK_1
    - Projects
      - Bank Merge
      - Demo Transaction Project
      - Mutual Funds Datamart
    - Systems
    - Hosts
    - RDBMS Servers
    - ETL Instances

API.SQL | SEQUENCES.SQL | STAGING.SQL | STORED_PROCS.SQL | TRANSLATION_MAP.SQL

```
CREATE PROCEDURE SP_PROCESS_API_DIM_ACTIVITY ( @LOAD_PROCESS_INSTANCE_ID INTEGER, @OBJECT_INSTANCE VARCHAR(100))
AS
BEGIN

BEGIN TRANSACTION
BEGIN TRY

/*
Resetting status on API.API_DIM_ACTIVITY table -- marking records as READY...
*/
UPDATE API.API_DIM_ACTIVITY SET
RECORD_STATUS = 'READY',
ERROR_REASON = NULL
WHERE LOAD_PROCESS_INSTANCE_ID = @LOAD_PROCESS_INSTANCE_ID
AND OBJECT_INSTANCE = @OBJECT_INSTANCE
AND RECORD_STATUS = 'INVALID_RI'

/*
Check if records with the same matching keys exist on the target table and mark API records appropriately
*/
UPDATE API.API_DIM_ACTIVITY
SET
RECORD_STATUS =  CASE  WHEN 20  =   CASE  WHEN ISNULL(API.API_DIM_ACTIVITY.ACTIVITY_TYPE,'')  =  ISNULL(DM.DIM_ACTIVITY.
CASE  WHEN ISNULL(API.API_DIM_ACTIVITY.ACTIVITY_SUB_TYPE,'')  =  ISNULL(DM.DIM_ACTIVITY.ACTIVITY_SUB_TYPE,'') THEN 1  EI
CASE  WHEN ISNULL(API.API_DIM_ACTIVITY.METHOD_ACT_ID,'')  =  ISNULL(DM.DIM_ACTIVITY.METHOD_ACT_ID,'') THEN 1  ELSE 0 END
CASE  WHEN ISNULL(API.API_DIM_ACTIVITY.DEPARTMENT,'')  =  ISNULL(DM.DIM_ACTIVITY.DEPARTMENT,'') THEN 1  ELSE 0 END +
CASE  WHEN ISNULL(API.API_DIM_ACTIVITY.ACTIVITY_OPTION_CODE,'')  =  ISNULL(DM.DIM_ACTIVITY.ACTIVITY_OPTION_CODE,'') THEN
CASE  WHEN ISNULL(API.API_DIM_ACTIVITY.ACTIVITY_REFERENCE_TX,'')  =  ISNULL(DM.DIM_ACTIVITY.ACTIVITY_REFERENCE_TX,'') TH
CASE  WHEN ISNULL(API.API_DIM_ACTIVITY.ACTIVITY_STATUS_CODE,'')  =  ISNULL(DM.DIM_ACTIVITY.ACTIVITY_STATUS_CODE,'') THEN
CASE  WHEN ISNULL(API.API_DIM_ACTIVITY.PRIORITY,'')  =  ISNULL(DM.DIM_ACTIVITY.PRIORITY,'') THEN 1  ELSE 0 END +
CASE  WHEN ISNULL(API.API_DIM_ACTIVITY.REFERENCE1,'')  =  ISNULL(DM.DIM_ACTIVITY.REFERENCE1,'') THEN 1  ELSE 0 END +
CASE  WHEN ISNULL(API.API_DIM_ACTIVITY.SCH_ALARM_FLAG,0)  =  ISNULL(DM.DIM_ACTIVITY.SCH_ALARM_FLAG,0) THEN 1  ELSE 0 END
API_DIM_ACTIVITY.RING_DATE,'1/1/1900')  =  ISNULL(DM.DIM_ACTIVITY.RING_DATE,'1/1/1900') THEN 1  EI
API_DIM_ACTIVITY.DONE_DATE,'1/1/1900')  =  ISNULL(DM.DIM_ACTIVITY.DONE_DATE,'1/1/1900') THEN 1  EI
API_DIM_ACTIVITY.NOTE,'')  =  ISNULL(DM.DIM_ACTIVITY.NOTE,'') THEN 1  ELSE 0 END +
API_DIM_ACTIVITY.PRIVATE_FLAG,0)  =  ISNULL(DM.DIM_ACTIVITY.PRIVATE_FLAG,0) THEN 1  ELSE 0 END +
API_DIM_ACTIVITY.ALARM_PAGE,'')  =  ISNULL(DM.DIM_ACTIVITY.ALARM_PAGE,'') THEN 1  ELSE 0 END +
API_DIM_ACTIVITY.ALARM_FLAG,0)  =  ISNULL(DM.DIM_ACTIVITY.ALARM_FLAG,0) THEN 1  ELSE 0 END +
```

6/24/2010 6:42:25
6/24/2010 6:42:25
6/24/2010 6:42:25

DW Studio generates SQL statements, stored procedures, and DDLs for intermediary tables used for processing feeds along with comments

GAMMASYSTEMS

DW Studio – Talend Code Import

Import items from an archive file or directory.

Select root directory: :\Work\Projects\DWS\Server_Deploy\Work\671

Select archive file:

Items List

type filter text

- Job Designs
  - AGIR_DM
    - CAMPAIGN
      - FundSpace_Campaign_Contact_Group
        - CONTAINER_FundSpace_Campaign_Contact_Group 0.1
        - STEP_10_STAGING_FundSpace_Campaign_Contact_Group 0.1
        - STEP_20_QUALITY_CHECK_FundSpace_Campaign_Contact_Group 0.1
        - STEP_30_API_FundSpace_Campaign_Contact_Group 0.1
        - STEP_40_TARGET_FundSpace_Campaign_Contact_Group 0.1
    - DISTRIBUTION_HIERARCHY
      - FundSpace_Distribution_Hierarchy
        - CONTAINER_FundSpace_Distribution_Hierarchy 0.1
        - STEP_10_STAGING_FundSpace_Territory 0.1

Errors and Warnings

overwrite existing items

DW Studio generates code components in compliance with the project-wide architectural specs – individual steps of a process along with a master (or container) job linking all the processing phases

DW Studio – Sample Talend Job

Talend Integration Suite (3.2.2.r33000) | se@gs.com | AGIR_DM_DEMO (Connection: Local)

File Edit View Window Help

Repository | Code Viewer | Outline

- Business Models
- Job Designs
  - AGIR_DM
    - ACTIVITY
    - CAMPAIGN
      - FundSpace_Campaign_Contact_Group
        - CONTAINER_FundSpace_Campaign
        - STEP_10_STAGING_FundSpace_Cam
        - STEP_20_QUALITY_CHECK_FundSpa
        - STEP_30_API_FundSpace_Campaign
        - STEP_40_TARGET_FundSpace_Camp
      - FundSpace_Campaign_Dim
      - FundSpace_Campaign_Response_Actio
    - DISTRIBUTION_HIERARCHY
  - HR_ETL_ROOT
  - test1 0.1
- Joblet Designs
- Contexts
- Code
- SQL Templates
- Metadata
  - Db Connections
    - AGIR_DataMart 0.1
    - FundSpace 0.1
    - Local_Sybase 0.1
    - Mutual_Funds_DataMart 0.1
      - CDC Foundation
      - Queries
      - Synonym schemas
      - Table schemas
        - AGIR_DM_SOURCE_TARGET_MAP
        - API_DIM_ACTIVITY_PARTY_GROUP
        - API_DIM_DISTRIBUTION_HIERARCH

Job STEP_30_API_FundSpace_Campaign_Contact_Group 0.1

DELETE_INSTANCE_RUN_DATA

OnSubjobOk

STG_Q_CAMPAIGN_CONTACT_SOURCE — SOURCE_TO_TMAP (Main) — BUSINESS_LOGIC — API_DIM_ACTIVITY_PARTY_GROUP (Main order:1) — AGGR_API_DIM_ACTIVITY_PARTY_GROUP — OUT_API_DIM_ACTIVITY_PARTY_GROUP (Main) — AUDIT_FIELDS_API_DIM_ACTIVITY_PARTY_GROUP — API_DIM_ACTIVITY_PARTY_GROUP

LOOKUP_FUNDSPACE_COUNTRY_CODE_1 (Lookup) — API_DIM_PARTY (Main order:2)

OnSubjobOk

LOOKUP_FUNDSPACE_COUNTRY_CODE_1 — AGGR_API_DIM_PARTY (Main) — AGGR_API_DIM_PARTY — OUT_API_DIM_PARTY (Main) — AUDIT_FIELDS_API_DIM_PARTY — API_DIM_PARTY

OUTPUT_STATS

Designer | Code

Job(STEP_30_ | Contexts(Job | Component | Run | Job Hierarch

0 errors, 0 warnings, 0 infos

Description
Errors (0 items)
Warnings (0 items)
Infos (0 items)

DW Studio generates jobs that comply with the business and technical rules, metadata objects for more efficient repository maintenance, and clear naming conventions.

GAMMASYSTEMS

DW Studio – Job Scheduling

General | File System | Target Database | Source Database Connections | Environment Variables | Operational

Batch Level Specifications | Process Schedule

Deployed Processes

Processes	Active	Deployed On
FundSpace Master Firm	☑	12/14/2009 14:47:5
FundSpace Trading Firm	☑	12/14/2009 14:48:0
FundSpace Trading Office	☑	12/14/2009 14:48:2
FundSpace NSM	☑	12/14/2009 14:48:4
FundSpace DSM	☑	12/14/2009 14:49:0
FundSpace Distribution Hierarchy	☑	12/14/2009 14:49:1
FundSpace WholeSaler	☑	12/14/2009 14:49:3
FundSpace Trading Rep	☑	12/14/2009 14:49:4
FundSpace Transaction	☑	1/5/2010 17:58:46
FundSpace Territory Target	☑	12/14/2009 14:50:2
FundSpace Lit Inventory	☑	12/14/2009 14:50:3
FundSpace Lit Inventory Kit	☑	12/14/2009 14:50:4
FundSpace Lit Order Header	☑	12/14/2009 14:51:0
FundSpace Lit Order Fact	☑	12/29/2009 09:35:5
FundSpace User	☑	12/14/2009 14:51:1

Days of Week

☐ Monday ☐ Thursday
☐ Tuesday ☐ Friday
☐ Wednesday ☐ Saturday
☐ Sunday

○ Time Based

Start Time 00 00 PM

◉ Dependency Based

Prerequisite Processes
FundSpace Trading Firm

Save Process Info

DW Studio can help define job execution order, scheduling and dependencies, and generate a master process that will drive all ETL jobs.

GAMMASYSTEMS

DW Studio – Generated Project Documentation

1

BANK
Project Documentation

Project Description
This is a demo project for BANK1.

The project contains feeds that demonstrate the functionality of DW Studio.

Subject Areas
The following subject areas are present in the system:

Subject Area Code	Subject Area Name	Description
ACCOUNTS	ACCOUNTS	
DEPOSITS	DEPOSITS	
WITHDRAWALS	WITHDRAWALS	

Feed Inventory
This section provides various views of feed related information.
There are 1 incoming feeds defined for the system.

Breakdown of incoming feeds by subject area:

Feed Name	Feed Code	Description
ACCOUNTS	ACCOUNTS	1 Incoming Feed
Customer Accounts 1	CA_1	

Incoming feeds deployment information by environment:

Feed Name	CODE_GEN
ACCOUNTS	
Customer Accounts 1	Deployed

Exceptions Definitions

The following exception severity levels have been defined for the system:

Severity Level Code	Severity Level Name	Description	Action
HIGH	High severity	N/A	REJECT
LOW	Low severity	N/A	PASS

The following generic exceptions have been defined for the system:

Exception Code	Exception Name	Description
EX01	Blank Last Name	
EX02	Overdraft Code is Blank	

Environment Information
This section provides information on environments registered for the project. Please note that it does not

2

contain sensitive information such as database and encryption passwords.

The following table lists the environments defined for the project:

Environment Name	Environment Code	Description
Code Generation	CODE_GEN	

Code Generation Environment Details

Host: , OS Type:

Directory	Path
Root Directory	/var/DWS/
Config Directory	$ROOT/config
Archive Directory	$ROOT/archive
Bin Directory	$ROOT/bin
Log File Directory	$ROOT/logs
Log File Directory	$ROOT/logs
Source File Root Directory	$ROOT/incoming
Output File Root Directory	$ROOT/outgoing
Temp Directory	$ROOT/temp

ETL Information

Parameter	Value
Instance Name	
Repository Host	
Repository Port	
Repository Server Port	

System Architecture
This section provides information on the architecture of the system.

Incoming Feed Architecture
Incoming feeds into B1 go through the following phases:

Step 1 – Staging Area
Data feeds get staged in staging tables that are replicated off of the source structures.
Staging area is populated using ETL
Each staging job has a naming pattern STEP_1_STAGE_%FEED_NAME%
Staging tables are identified by the prefix 'STG_'.

DW Studio can generate full documentation including business/data specifications and technical guidelines on a project or individual data mappings.

GAMMASYSTEMS