Exception Generation with Javautil

Jim Schmidt
jjs-javautil@dbexperts.com

June 27, 2020

Contents

Chapter 1

Introduction

Condition Identification supports the declarative identification of records that meet one or more criteria.

It works by running a set of SQL queries against the data and persisting the identification rule identifier and the primary key for the records that satisfy the query.

1.1 Features

1.2 Concepts

Part I

Installation

1.3 Terminology

1.3.1 Rule Group

A rule group is a set of rules that collectively define the conditions to be isolated.

1.3.2 Run Parms

Run Parms are bind variables used for the various rules.

1.4 Metrics

Metrics may be gathered for

1.5 UT_TABLE_RULE

This table identifies the database table that contains the primary key for table that is being identified.

Chapter 2

Schema

TODO list tables

Chapter 3

Processing

Chapter 4

Components

4.1 Rule Processor

4.2 Invocation

TODO dbms_pipe

4.3 Classes

4.3.1 ExceptionRuleService

TODO what it does

4.4 todo

4.5 Parallism

4.6 Data Setup

4.7 Output Review

4.8 Fatal and Warn

- fix schema tables with no surrogate key, no foreign keys etc. See exception processing.mer
- document the package
- document logging
- document metrics
- document ut_table_msg
- document ut_table_report_sum
- document exclusion rules
- need functional area
- document calling as a procedure
- what does ut_query do with anything
- no real support for ut_table_msg
- todo what is ds_table and why does ut_query reference it?
- document ut_table_rule and hist figure out what all of these columns are used for

  - is ut_query used

- Primary Keys
- Foreign Keys
- Not Null
- Check Constraints

However, there are many logical conditions which are beyond the scope of available functionality.

The Javautil Exception Generator allows you to set up simple rules to identify records or tables that fail to meet business requirements.

4.9 Benefits

Chapter 5

Pre-requisites

- Obtain javautil code
- configure machine
- configure datasources

Chapter 6

How it works

6.1 Overview

- getParms();
- getRun();
- getRules();
- getBinds();
- createProcessLog();
- processRules();
- updateRunStatus();
- acknowledge()

6.2 Creating the User

grant create sequence to user;

6.3 Parameters

6.3.1 Run Number

UT_RULE_GRP_RUN_NBR

6.4 Get UtRuleGrpRun

6.5 Get UT_TABLE_RULE

Get the rules for the run.

6.6 Get UT_RULE_GRP_RUN_PARMS

6.7 Process Rules

Connect to source - todo describe data Connect to destination binds Run the query insert into gtt_ut_table_row_msg merge into ut_table_row delete where they don't exist

Chapter 7

Database Objects

Also depends on the logging tables in Dbexperts3/ddl/oracle/logging

- huh

To generate the tables not only are the mapping files required, the associated beans are even though they are never used.

The dto's must be in the classpath.

http://docs.jboss.org/hibernate/core/3.3/reference/en/html/toolsetguide.htmltoolsetguide-s1-2

source and destination databases may be different

declarative rules

TODO need to support initialization procedure

7.1 SqlDeveloper

cd /opt sudo unzip /common/Downloads/sqldeveloper-4.2.0.16.260.1303-x64.zip

7.2 Installation

7.2.1 Create Database Objects

Oracle

cd /workspace-javautil/javautil-conditionidentification/src/main/ddl/oracle sqlplus sys/password as sysdba @create_user sqlplus condition/identification @