|  |  |
| --- | --- |
| PGE Logo | |
| Component Specification | |
|  |  |

Pre Canned Tracing

Prepared by:

IBM, Kris Foster

April 2014

Table of Contents

[Security Classification 3](#_Toc384830040)

[Retention Requirements 3](#_Toc384830041)

[Document Audience 3](#_Toc384830042)

[1 Introduction 4](#_Toc384830043)

[1.1 Goals and objectives 4](#_Toc384830044)

[1.2 Referenced documents 4](#_Toc384830045)

[1.3 Business Requirements 4](#_Toc384830046)

[1.4 Statement of scope 4](#_Toc384830047)

[1.4.1 In – Scope 4](#_Toc384830048)

[1.4.2 Out of scope 4](#_Toc384830049)

[1.5 Software context 4](#_Toc384830050)

[1.6 Major assumptions 5](#_Toc384830051)

[1.7 Process Flow 5](#_Toc384830052)

[2 Component Design 6](#_Toc384830053)

[2.1 Pre Process PL/SQL Scripts 6](#_Toc384830054)

[2.2 Tracing 6](#_Toc384830055)

[2.3 Post Process PL/SQL Scripts 6](#_Toc384830056)

[2.4 Restrictions / limitations 6](#_Toc384830057)

[2.5 Assumptions 7](#_Toc384830058)

[3 Data Design 8](#_Toc384830059)

[3.1 Data model Changes 8](#_Toc384830060)

[3.2 Error Handling and logging 9](#_Toc384830061)

[3.3 Unit Test 9](#_Toc384830062)

[4 Configuration 10](#_Toc384830063)

[4.1 Configuration file 10](#_Toc384830064)

[4.2 Required Sql Scripts 11](#_Toc384830065)

[5 Security 12](#_Toc384830066)

[5.1 Security Matrix 12](#_Toc384830067)

|  |  |  |  |
| --- | --- | --- | --- |
| Revision History | | | |
| Document # | Date | Author | Summary of Changes |
| 1.0 | 4/9/2014 | Kris Foster | Initial Draft |
|  |  |  |  |

## Security Classification

## Retention Requirements

## Document Audience

# Introduction

## Goals and objectives

The purpose of this document is to detail the architecture, configuration, and implementation of custom functionality to be developed for PG&E ED GIS Pre Canned Tracing. This document covers the components that will be developed to generate tables with the pre canned electrical tracing information in database tables from GIS on a nightly basis.

Specifications of each component of the proposed solution are described here and could be used by technical staff to develop the extensions components.

The intended audience includes project leads, technical leads and technical staff such as programmers/developers.

## Referenced documents

| Title | Source | Type | Date | Revision |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |

## Business Requirements

Implements the following requirements:

| TFS ID | Requirement | ID | Source |
| --- | --- | --- | --- |
|  |  |  |  |

## Statement of scope

### In – Scope

This component specification defines the components that are required for the pre canned tracing functionality.

### Out of scope

## Software context

This section defines the software requirements for implementing the custom extensions with specific version numbers. The software requirements are

ArcFM 10.0.3

ArcGIS 10 SP5

.Net 3.5

Oracle 11g

EDER (current version)

The development of custom components will be carried out using C# as the programming language and and will also utilize oracle PL/SQL procedures for additional processing.

## Major assumptions

* ArcFM Feeder Manager is configured properly.
  + All feeder manager information on in the GIS database is accurate.

## Process Flow



# Component Design

The pre canned tracing process will populate the necessary information into tables for the associated feeder manager geometric networks. The information populated in the new tables will allow for a simple sql query to determine an upstream or downstream trace for any starting feature in the network.

The process will populate the table by first executing any configured pre process PL/SQL scripts. Once finished, the process will execute an ArcFM feeder manager trace for every circuit in the network for all geometric networks configured. Once all tracing has been completed for all networks, any configured post process PL/SQL scripts will be executed.

## System Configuration Requirements

It is assumed that the database is unversioned for the logic supplied.

For performance the following indexes are needed in the system.

create index look\_up\_subtrc\_result\_net on edgis.n\_3\_desc(elementtype,userclassid,userid,eid);

create index look\_up\_trace\_result\_net on edgis.n\_4\_desc(elementtype,userclassid,userid,eid);

create index oid\_physical\_gdbitems on gdb\_items(OBJECTID,PHYSICALNAME) ;

## Pre Process PL/SQL Scripts

As of this document the following are the PL/SQL scripts that should be configured to run with the execution of this tool.

1. EDGIS.Drop\_Sub\_Trace\_Indices – This procedure will drop any indices that can hinder the performance of inserting rows during the tracing process.
2. EDGIS.Drop\_Elec\_Trace\_Indices – This procedure will drop any indeices that can hinder the performance of inserting rows during the tracing process.

## Tracing

After the pre process PL/SQL scripts have finished executing the tracing will begin as follows.

1. Determine the list of geometric networks that are configured to be traced.
2. For each circuit in the configured geometric networks perform an ArcFM downstream trace from the circuit source feature populate the EDGIS.PGE\_{GeometriceNetworkName}\_Trace table with all features traced.

Note: At this point information in the tracing tables is very generic and does not contain any information about what feature classes or globalIDs they are associated with.

## Example diagram of depth first order, with branch tracking logic table values created during trace.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| CircuitID | Order Num | Tree Level | Min Branch | Max Branch | From EID | To\_EID |
| 123 | 6 | 0 | 0 | 3 | Null | 0 |
| 123 | 2 | 1 | 0 | 1 | 0 | 1 |
| 123 | 0 | 2 | 0 | 0 | 1 | 3 |
| 123 | 1 | 2 | 1 | 1 | 1 | 4 |
| 123 | 5 | 1 | 2 | 3 | 0 | 2 |
| 123 | 3 | 2 | 2 | 2 | 2 | 5 |
| 123 | 4 | 2 | 3 | 3 | 2 | 6 |

Branch

Level

## Post Process PL/SQL Scripts

As of this document the following are the PL/SQL scripts that should be configured to run with the execution of this tool (in the order provided).

1. EDGIS.UpdateFCIDs – This will update all of the feature class ID values for all entries in the tracing tables.
2. EDGIS.UpdateSchemFCIDs – This will update all of the schematics feature class ID values. This stored procedure relies on a mapping between the feature class IDs of the electric feature class IDs and the schematics feature class IDs. This mapping is specified inside the PL/SQL procedure.
3. EDGIS.UpdateGlobalIDsAndFeederInfo – This procedure will update the tracing table with all of the globalIDs and FeederInfo information for all rows. This provides a simple lookup to obtain the original feature.
4. EDGIS.Create\_Sub\_Trace\_Indices – This will create any necessary indices for performance sake with common sql queries.
5. EDGIS.Create\_Elec\_Trace\_Indices – This will create any necessary indices for performance sake with common sql queries.

### 2.4 Restrictions / limitations

1. This process requires an ArcFM and ArcGIS license to execute.
2. As the process must execute PL/SQL queries that drop / create indices and gather stats, the user executing the process must be the owner of the tracing tables. At the current writing of this document it is assumed that the tool will be run as the EDGIS user.
3. Currently the pre canned tracing does not work in a version database system. All edits must be compressed to the base tables or the tracing will be out of synch.

### 2.5 Assumptions

1. Database user that that will be configured to run the tool will be the EDGIS user.
2. All edits are compressed to the base tables. (i.e. Unversioned database).

# Data Design

## Data model Changes

**New Tables**

**Component Name:** Electric Distribution Network tracing table

**Table Name:** PGE\_ElecDistNetwork\_Trace

| **Field Name** | **Datatype** |
| --- | --- |
| FeederID | NVARCHAR2(9) |
| From\_Feature\_EID | NUMBER(38) |
| To\_Feature\_EID | NUMBER(38) |
| To\_Feature\_GlobalID | CHAR(38) |
| To\_Feature\_FCID | NUMBER(38) |
| To\_Feature\_Schem\_FCID | NUMBER(38) |
| To\_Feature\_FeederInfo | NUMBER(38) |
| To\_Feature\_Type | NUMBER(1) |
| Order\_Num | NUMBER(38) |
| Min\_Branch | NUMBER(38) |
| Max\_Branch | NUMBER(38) |
| TreeLevel | NUMBER(38) |

**Component Name:** Substation Network tracing table

**Table Name:** PGE\_SubGeomNetwork\_Trace

| **Field Name** | **Datatype** |
| --- | --- |
| FeederID | NVARCHAR2(9) |
| From\_Feature\_EID | NUMBER(38) |
| To\_Feature\_EID | NUMBER(38) |
| To\_Feature\_GlobalID | CHAR(38) |
| To\_Feature\_FCID | NUMBER(38) |
| To\_Feature\_Schem\_FCID | NUMBER(38) |
| To\_Feature\_FeederInfo | NUMBER(38) |
| To\_Feature\_Type | NUMBER(1) |
| Order\_Num | NUMBER(38) |
| Min\_Branch | NUMBER(38) |
| Max\_Branch | NUMBER(38) |
| TreeLevel | NUMBER(38) |

## Error Handling and logging

Logging in this tool utilizes Log4Net logging and can have it’s configuration file adjust as per Log4Net documentation. The process will return 0 for success and non-zero for failure.

## Unit Test

Unit testing for this process is relatively simple as it is an impossible to verify the tracing electric order without a consumer. Most unit testing will be involved in any consumers of the tracing table.

**Test Case 1:** After configuration of the PGE\_Cached\_Tracing.exe.config file, execute the PGE\_Cached\_Tracing.exe by double clicking on it. Monitor the console window for completion. Upon completion review the log files to ensure there were no errors. Open ArcCatalog and browse the the tracing tables. Ensure they are populated with data in all columns. If any columns are null, this is an indication that something failed.

# Configuration

## Configuration file

<?xml version="1.0" encoding="utf-8" ?>

<configuration>

<appSettings>

<!-- Licenses Configuration. -->

<!-- ArcGIS License (refer to esriLicenseProductCode enumeration in ESRI documentation for valid values)-->

<add key="EsriLicense" value="esriLicenseProductCodeAdvanced"/>

<!-- ArcFM License (refer to mmLicensedProductCode enumeration in ArcFM documentation for valid values)-->

<add key="ArcFMLicense" value="mmLPArcFM"/>

<add key="OracleConnection" value="edgiswebdb01,edgisp2d,edgis,edgis" />

<add key="SDEConnectionFile" value="C:\Users\k1f8\AppData\Roaming\ESRI\Desktop10.0\ArcCatalog\EDGISP2D\_edgis.sde"/>

<!-- Comma separated Geometric network names -->

<add key="GeometricNetworks" value="EDGIS.ElectricDistNetwork,EDGIS.SubGeometricNetwork"/>

<!-- Comma separated trace results tables. A unique trace results table must be specified for each geometric network specified-->

<add key="TraceResultsTables" value="EDGIS.PGE\_ElecDistNetwork\_Trace,EDGIS.PGE\_SubGeomNetwork\_Trace"/>

<add key="FromFeatureEIDFieldName" value="FROM\_FEATURE\_EID"/>

<add key="CircuitIDFieldName" value="FEEDERID"/>

<add key="ToFeatureEIDFieldName" value="TO\_FEATURE\_EID"/>

<add key="OrderNumFieldName" value="ORDER\_NUM"/>

<add key="MinBranchFieldName" value="MIN\_BRANCH"/>

<add key="MaxBranchFieldName" value="MAX\_BRANCH"/>

<add key="TreeLevelFieldName" value="TREELEVEL"/>

<add key="ToFeatureTypeFieldName" value="TO\_FEATURE\_TYPE"/>

<!-- Specify any oracle stored procedures that should be run when before tracing has started and after it has finished. They will

be executed in the order they are listed-->

<add key="BeforeTracingStoredProcedures" value="Drop\_Sub\_Trace\_Indices,Drop\_Elec\_Trace\_Indices"/>

<add key="AfterTracingStoredProcedures" value="UpdateFCIDs,UpdateSchemFCIDs,UpdateGlobalIDsAndFeederInfo,Create\_Sub\_Trace\_Indices,Create\_Elec\_Trace\_Indices"/>

</appSettings>

</configuration>

## Required Sql Scripts

Current Releases Directory will contain the most recent sql script.

\\sfetgis-nas01\sfgispoc\_data\ApplicationDevelopment\IBM\_Delivery\Releases\PGECachedTracing

# Security

## Security Matrix

Data Sensitivity Levels

|  |  |  |  |
| --- | --- | --- | --- |
| **Sensitivity Level** | **Rating** | **Description** | **Example** |
| 4 | Critical | Any Information where an unauthorized disclosure to unintended recipients could result in financial loss, loss of life, damage to company reputation, or cause significant risks | Customer Information under SB1386, confidential databases, trade secrets, inventions, financial reports prior to release. |
| 3 | High | Any Information that, if disclosed to or modified by unauthorized individuals, might result in the risk of significant loss, significant productivity loss, or significant embarrassment to PG&E | Non-administrative passwords, customer Information, shareholder Information, labor and employee relations data |
| 2 | Medium | Information which does not meet the risk criteria for higher Sensitivity levels and, If disclosed or modified by unauthorized individuals, might result in the risk of some monetary loss, some productivity loss or some embarrassment to PG&E. Sensitivity Level 2 information is often collected for internal use. | Corporate Policies, directories of employee information, organizational charts, contracts (unless otherwise classified) |
| 1 | Low | Anything produced for Public review. At this level any disclosures could reasonably expect not to have an adverse affect. Unintentional alteration or destruction is the primary concern for Sensitivity Level 1 information. | Public information on [www.pge.com](http://www.pge.com) public news releases |

|  |  |
| --- | --- |
| **Security** | **Usage** |
| Data Privacy Level |  |
| Activity Performed |  |
| GIS Application Security |  |