# 

ETL Specification Table of Contents

What follows is a table of contents for the ETL Specification document. This is targeted at organizations that do not have rigid specification / development procedures in place. Those who already follow clear development methodologies will find this specification document to be weak. Those who fly by the seat of their pants, will find this specification to be insanely detailed.

Much of this information should exist already. Examples include the logical dimensional model, the physical database model, the source-to-target map, and the data profiling reports. It’s very helpful to pull everything together into a single document. Make the specification document readable as a standalone document, and include links to the detailed external documents (it’s easy to embed links in a Word doc).

The “mandatory” column represents what we consider the bare minimum of information that you should have pulled together, and issues to have thought through, before you do any real development. Anytime you touch SSIS before that point should be considered throwaway / learning / prototyping.

# Change Log

|  |  |  |
| --- | --- | --- |
| What | Approx page count | Mandatory |
| Overview of changes to this document (who, what, when) | 1 | ✓ |

# Summary

|  |  |  |
| --- | --- | --- |
| What | Approx page count | Mandatory |
| What is this ETL specification for? What system/subsystem/phase? Historical load or incremental load? | 2 | ✓ |

# Detailed Dimensional Model

|  |  |  |
| --- | --- | --- |
| What | Approx page count | Mandatory |
| Provide the detailed dimensional model (such as Excel spreadsheet or Erwin diagram)   * Most details are available as a linked document * Include enough discussion within the text of this document to help the reader understand what’s going on * List and brief description of each target table | 3 (overview + link to spreadsheet and/or data model overview documentation) | ✓ |
| Source to target map | 2 (overview + link to document) | ✓ |
| Data profiling reports | 2 (overview + link to document) | ✓ |
| Database physical design | 1 (overview + link to DDL script or database definition) | ✓ |

# Architecture and Default Strategies

|  |  |  |
| --- | --- | --- |
| What | Approx page count | Mandatory |
| ETL Architecture: software, hardware, and placement of software on hardware   * Most details are available as a linked document * Include enough discussion within the text of this document to help the reader understand what’s going on * Location of staging areas (file v. database, target location) | 1 (overview + link to architecture document) | ✓ |
| Data model of the ETL process metadata | 2 |  |
| Requirements for system availability, and the basic approach to meeting those requirements | 2 | ✓ |
| Overview of generic error handling. | 2 |  |
| For each source system, the default strategy for extract   * Change Data Capture v. Push from source to flat files v. Separate set of extract packages, staging to tables v. whatever else you think of | 0.5 / source system | ✓ |
| Default slowly changing dimension handling strategy (eg use SSIS SCD transform) | 1 |  |
| Preconditions (like checking for disk space; closing DW to user queries; dropping or creating indexes; truncating staging tables) | 1 |  |
| Postconditions (updating statistics and indexes, cleaning up staging files or database; running a backup) |  |  |

# High level map and Table-level details

|  |  |  |
| --- | --- | --- |
| What | Approx page count | Mandatory |
| High level map picture and brief discussion. Draw a picture that explains where data is sourced from and where it’s going. Annotate the major changes that need to happen along the way. What we’re talking about is something similar to Figure 7-2 in the book, but at a much higher level so you can fit an entire subject area on 1-2 pages | 2-4 per subject area (1-2 pages for pictures, 1-2 pages for text) | ✓ |
| For each table, an estimate of complexity: Is this table’s package hard, medium, easy? | 0.5 | ✓ |
| For each table, a high level map as in Figure 7-2. Accompanied by discussion of the issues, pseudocode any really complicated transformations.   * Table design (basically the DDL to create the table, can be a link) * For each attribute, Type1 v. Type 2 handling * Incremental data volumes, measured as new and updated rows / load cycle * How to handle late arriving data for facts and dimensions * Load frequency (eg daily) * Table partitioning strategy * Overview of data source(s), focusing on any unusual characteristics (unusually short access window; data lives in Excel; etc) * Detailed source-to-target mapping (link to location in existing document) * Detailed source data profiling (link to location in existing document) | 2-8 per table |  |
| Deviations from default strategies, if any:   * SCD management (eg default is use SSIS wizard but for this table we’re going to do XYZ for reason ABC. * Extract strategy * Startup * Cleanup * Error handling | 0-2 |  |
| Dependencies: Which other tables need to be loaded before this table is processed? | 1 |  |

# Summary of flow

|  |  |  |
| --- | --- | --- |
| What | Approx page count | Mandatory |
| Describe master packages, and provide a first cut at job sequencing. Create a dependency tree that specifies which tables must be processed before others. Whether or not you choose to parallelize your processing, it’s important to know the logical dependencies that cannot be broken. | 2-4 per subject area (1-2 pages for pictures, 1-2 pages for text) | ✓ |