Appendix A

Data Migration Strategy Plan

The Data Migration Strategy Plan outlines the high-level approach for the migration (or integration) portion of the project. It includes the overall project background, required tasks, project assumptions and constraints, current and future state architectures, and other relevant details that will communicate the proposed approach to the reader. To create your first draft, populate each category using the provided guidelines and then add, modify, or delete where necessary.

# Overview

[This section introduces the background of the project, usually copied from the proposal or other project management plans. This section also should contain at least one or two paragraphs on the goals of the current migration/integration tasks and how it meets the overall business objectives.]

## Introduction

[State the overall vision of the project. You can reference any existing Statements of Work or other artifacts, keeping in mind that a broad definition is supported.]

## Purpose

[Describe the purpose of the integration part of the project]

## References

[Describe documents that have been used in previous systems or for referencing more project details.]

# Assumptions, Constraints, Risks, and Gaps

[This section sets the expectations for the limits of the current project. It defines assumptions, (what the team can and cannot do versus what other teams must do), constraints (limitations due to technology or time), risks (potential issues that might occur), and gaps (areas that remain open for debate).]

## Assumptions

[Describe any assumptions or dependencies regarding the data conversion effort. These may concern such issues as related software or hardware, operating systems, end-user characteristics, and the data that must be available for the conversion.]

## Constraints

[Describe any limitations or constraints that have a significant impact on the data conversion effort. Such constraints may be imposed by any of the following (the list is not exhaustive):

* Hardware or software environment
* .End-user environment (e.g., user work and delivery schedules, timeframes for reports, etc.)
* Availability of resources
* Interoperability requirements (e.g., the order that each system involved in the conversion processes data)
* Interface/protocol requirements
* Data repository and distribution requirements (e.g., volume considerations, such as the size of the database and amount of data to be converted; the number of reads and the time required for conversions)
* Referential data integrity
* The time allowed to complete the conversion process
* Security Requirements]

## Risks

[Describe any risks associated with the data conversion and proposed mitigation strategies. Include any risks that could affect conversion feasibility, technical performance of the converted system, the conversion schedule, costs, backup and recovery procedures, etc.]

## Gaps

[Describe those parts of the current implementation that are gaps.]

# Architecture

[This section contains two broad constructs: the current state and the future state. The current state represents the “as-is” system, that is, where the data currently rests, the data systems involved, the counts of records stored, and other details.

The future state is the “to-be” system—the final snapshot where the data will exist after the project is complete. These subcategories would include parallel topics to the current state, but with more assumptions—most of which is unknown at the outset but will be completed over time.]

## Current State

[This diagram will show the current state of the system with all feeders, nodes, and so on. Feel free to use the modeling tool of your choice to represent the way you want the system to look.]

### Current State Data Sources

[List the data sources, their description, and the tool/technology used to house and manipulate the data in the existing architecture.]

### Current State Record Counts

[Provide the record counts for each relevant entity currently in the source systems. Separate each system into individual tables.]

### Current State Data Model

[Describe the high-level data model for the current state (if one exists).

### Current State Integration Points

[List the “integration points”—how the current source to target systems are populated in daily loads—which includes the Source System name, Target System Name, description of the process, the entities touched (i.e., Subject Area), the type of integration, the frequency of the occurrence, and (optionally), the number of files required to produce this integration.]

## Future State

[Represent the proposed architecture with the data design model, generally produced following initial requirements gathering and consultation with other teams.]

### Future State Data Sources

[List the data sources and their description used to house and manipulate the data in future architectures.]

### Future State Data Model

[Describe the high-level data model for the future state (if one exists).]

### Future State Integration Points

[List how the future source to target systems populates in daily loads—which includes the Interface (target) name, the ETL direction (inbound outbound), the source system, the target system, the description of the process, the entities touched (i.e., Subject Area), and the data format.]

# Development Tools

[Describe the tools for development, including your ETL tools, scripting languages, or any other technology. We’ve discussed several of them in Chapter 7 Additional tools/technologies could include ETL tools such as Informatica, Pentaho, Data Loader, Cast Iron, Jitterbit, and so on.]

## Environment Usage

[Describe the different environments that support the implementation, such as Development, Test, and Production—see Chapter 7 for more details.]

## Data Migration Approach

[This section covers the overall approach to the integration, such as scope, roles and responsibilities of team members, what should take place pre, during and post-migration, and contingency plans. It also describes how data will move from the source to the target and supplemental systems.]

### Scope

[Defines the scope of data migration/integration such as quantity and history, as well as those items NOT included in migration such as data profiling or data remediation.]

### Approach

[Defines how the data migration will take place using the source systems, the ETL tool, and the environments. This section is the general, as opposed to the detailed plan covered in the Data Migration Process Flow Design.]

### Team Roles and Responsibilities

[Defines the names and titles of stakeholders involved with the overall migration strategy and their roles.]

## Migration Process

[Introduces the migration process where you describe what will happen during the steps of pre-migration, during migration, and post-migration. It is generally an introduction to the parts that come next.]

### Pre-Migration Activities

[Defines the major pre-migration activities such as preparing the data loads through profiling and remediation, acquiring login access to systems, and loading sample data]

### Migration Activities

[Define the major migration activities from preparation to test to deployment, including the high-level mapping of objects between source and target. The detailed step-by-step review is described in the Data Migration Process Flow plan.]

### Post-Migration Activities

[Defines the major post-migration activities such as monitoring (see Chapter 8), operation and maintenance, and future design reviews.]

# Contingency Plan

[Defines the contingency plan should the deploy fails, and rollback or reiteration is necessary.]

# Testing/Validation

[Describes the framework for the type of tests we plan to conduct, such as unit testing, joint integration, verification testing, and so on. It also describes the testing environment and which of our project teams will be in charge of the tests.]

## Testing Methods

[Describes the methods of testing we plan to conduct (unit testing, joint integration, blue-green), as well as the measurements provided to confirm the testing such as inputs, outputs, and definitions of success.]

## Migration

[Describes the tests conducted to confirm migration success/failure.]

## Integration

[Describes the tests conducted to confirm integration success/failure.]