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.]

J. Goldfedder, *Building a Data Integration Team*, https://doi.org/10.1007/978-1-4842-5653-4

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 include 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 populate 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), operations 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.]

APPENDIX B

Data Profiling Template

All data should be identified within the source systems to understand the shape, consistency, and variability among the individual records. The goal of the integration team is to gather the systems, tables, fields, and other source inputs and then analyze each of these records either from the source or through input staging files. Although every project migration is different, data usually has a specific pattern that is worth noting in a data profiling document. Don McMunn (www.ipcdesigns.com/data_profiling/) has provided a data profiling tool that can aid in this effort. See Table B-1 for several of the fields that I have found the most useful along with an Excel spreadsheet (located at *https://github.com/Apress/building-a-data-integration-team/tree/master/templates*) with several samples to get you started.

Table B-1. *Description of Profiling Fields*

Field	Description
TNAME	Name of table containing column being profiled
CNAME	Name of column in table being profiled
DATA_TYPE	Name of the data type of the column
DATA_LENGTH	Maximum length of the column in bytes
NUM_TABLE_ROWS	Total number of rows contained in the table at the time of the data profiling pass
NUM_DISTINCT_VALUES	Total number of unique values found in the column at the time of the data profiling pass
MIN_DATA_LENGTH	Shortest length of data value found in this column; mainly useful for string data types

(*continued*)

J. Goldfedder, *Building a Data Integration Team*, https://doi.org/10.1007/978-1-4842-5653-4

Table B-1. (*continued*)

Field	Description
MAX_DATA_LENGTH	Longest length of data value found in this column; mainly useful for string data types
NUM_NULLS	Total number of NULL values found in the column at the time of the data profiling pass
DENSITY	A ratio of non-null values to total rows in the table
ALPHANUM_COUNT	Number of rows containing ONLY an alphanumeric value that is not either a date or a numeric-only value
DATE_COUNT	Number of rows containing ONLY a valid date value in this column
NUMERIC_COUNT	Number of rows containing ONLY numeric values in this column regardless of data type
MIN_ALPHANUM_VALUE	Minimum alphanumeric value found in a string column; based on default page
MAX_ALPHANUM_VALUE	Maximum alphanumeric value found in a string column; based on default page
MIN_NUMERIC_VALUE	Smallest numeric value found in a numeric column
MAX_NUMERIC_VALUE	Largest numeric value found in a numeric column
MIN_DATE_VALUE	Oldest date value found in this column
MAX_DATE_VALUE	Most recent data found in this date column
NUM_EMAIL_INVALID	(If field is an EMAIL field) How many of these records are invalid?
NUM_CONTAINS_ NONASCII	How many records contain characters outside the range of normal ASCII values (i.e., foreign and non-printable values such as ä or line feeds ^013)?
PICKLIST_VALUE	(If field is a PICKLIST field) What are the values?

APPENDIX C

Issues Log Template

The Issues Log template is designed to record and track any issues or questions related to data quality and data migration follow-up. This document is similar in purpose to other software-based issues logs in that it anticipates requirement follow-ups or potential data corrections. In any case, it is convenient to have for documenting bugs and moving the project forward if data product owners are limited in availability.

I provide some basic Issues Log fields with associated definitions in Table C-1.

Table C-1. *Issues Log Tracking Fields and Definitions*

Field	Definition
Project Name	The name of the project under review
Date of Document	The current log publication date
Issue	This value should be a standard numbering system (e.g., 001, 002, 003)
Description	A detailed description of the issue
Priority	High, medium, or low priority
Category	Assign to a category
Reported By	Who reported the issue?
Assigned To	To whom is the issue assigned?
Status	What is the status of the issue?
Date Resolved	What date was the issue resolved?
Resolution/Comments	What was the resolution, or what is being done to resolve the issue?

J. Goldfedder, *Building a Data Integration Team*, https://doi.org/10.1007/978-1-4842-5653-4

APPENDIX D

Source-to-Target Mapping Template

As the name implies, the Source-to-Target Mapping template maps the association of data from the source to target systems. This spreadsheet is referenced and modified throughout the requirements gathering, build, testing, and monitoring phases, perhaps serving as the most important deliverable for the team. Consequently, the goal of the Source-to-Target Mapping spreadsheet should be to supply enough transformation rules and technical decisions to readers without overwhelming them with details.

A quick Internet search will reveal many recommendations for the “best” Source-to-Target mapping template. I have identified in Table D-1 the fields and associated definitions that have served me well for the majority of my projects, but yours may differ. Feel free to modify, add, or subtract where necessary, keeping in mind that we want to balance flexibility with comprehension and not go too far in either direction.

J. Goldfedder, *Building a Data Integration Team*, https://doi.org/10.1007/978-1-4842-5653-4

Table D-1. *Source-to-Target Mapping Fields and Definitions*

Field	Definition
SourceDatasystem	Put the formal name for the system that is the data source, for example, “Mailing List,” “Recruiting”
SourceTable	The physical name of the source table, for example, “CONTACT,”“PRODUCT”
SourceField	The physical name of the source field/column, for example, “ROWID,” “FIRSTNAME”
SourceDataType	The data type for the source data, for example, “VARCHAR(30),” “DATETIME(mm/dd/yyyy),” “NUMBER”
TargetDatasystem	Put the formal name for the system that is the data target, for example, “Salesforce,” “DataWarehouse”
TargetTable	The physical name of the target table, for example, “MERCHANDISE__C,”“ACCOUNT”
TargetField	The physical name of the target field/column, for example, “ROWID,” “FIRSTNAME”
TargetDataType	The data type for the target data, for example, “STRING,” “REFERENCE,” “NUMBER”
TransformationRule	Enter the high-level business rules required for transforming the data from source to target, for example, “Join constituent.contact_id on constituent_add.rowid where address.primary_address = TRUE and address.valid=TRUE”
PicklistValues	Insert related picklist values, especially if customized, for example, “Freshman, Sophomore, Junior, Senior,” “Small, Medium, Large”
DesignNotes	Use this field to provide additional design notes that are not necessarily part of the transformation rules, for example, “Values for this field are not case sensitive and should be reviewed before import,” “Trim those values that exceed the maximum datatype for this field”
DateCreated	Insert the initial date this field was added to the model, for example, “12/1/2019”

(*continued*)

Table D-1. (*continued*)

Field	Definition
DateLastUpdated	Insert the date this field mapping was modified, for example, “12/22/2019”
OpenIssues	Insert any open issues encountered during development, for example, “We are missing several row identifiers in the join,” “What should we do with fields that go over 80 characters?”
Assumption	Describe the workaround/assumption for the open issues, for example, “Ignore all joins where the ID does not match,” “Trim fields at 80 characters”
AssumptionApproved	List whether the assumption was approved and who approved it. If NOT approved, list the workaround specified. For example: “Assumption approved by J. Smith,” “Workaround should be to omit all records where field > 80 characters--S. Brown”
ResolutionCloseDate	Insert the date the resolution was deemed as closed, for example, “12/08/2019”
FollowUpRequired	Indicate whether this data needs follow-up. During a filter, this will help guide conversations. For example: “YES,” “NO”

APPENDIX E

Data Migration Process Flow Design Template

The Data Migration Process Flow provides a detailed design for migration processes, including the order of operations. It can be considered a technical appendix to the Data Migration Strategy, intended to provide the step-by-step processes that fulfill the proposal offered in that document. This document tracks the migration at a granular level and, therefore, should be updated and reviewed after each change in scope to ensure the requirements still are in alignment with stakeholder expectations.

Order of Operation

[Describe the step-by-step operation for object migration. For example:]

This document defines the process flow for development, deployment, and cleanup required to migrate the stakeholder's list of data. These activities can vary in time, depending on the volume of data provided and the complexity of updates.

The integration team presents the breakdown for deployment to Development, Test, and Production environments in Figure E-1. These individual steps performed for each task are summarized later in this document.

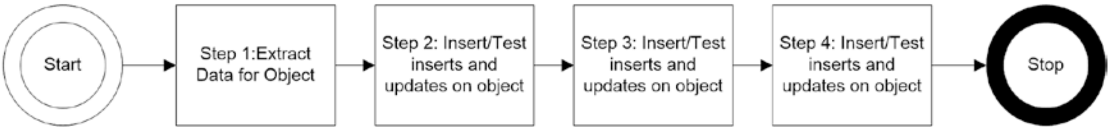

Start
Step 1:Extract Data for Object
Step 2: Insert/Test inserts and updates on object
Step 3: Insert/Test inserts and updates on object
Step 4: Insert/Test inserts and updates on object
Stop

Figure E-1. *Sample Project Flow*

J. Goldfedder, *Building a Data Integration Team*, https://doi.org/10.1007/978-1-4842-5653-4

Mapping Logic

[This section defines the high-level rules for migrating source-to-target data, mostly used to confirm with the client that these migrations are accurate. You will be referencing your Source-to-Target Mapping spreadsheet at some point in this section. As indicated throughout this book, the spreadsheet should be available in a separate spreadsheet or database, and you can include a summary table for easy reference. For example:]

Due to currently existing duplications as well as priority field determination, the integration team has included business logic as part of the ETL process. In collaboration with stakeholder requirements, the integration team has summarized the business rules for populating the target data store. The following source-to-target maps identify these transformation rules (last column) as well as the source system, the source location (i.e., the database or CSV file containing all deduplicated records), the source field name, and the target object and field. Table E-1 shows the field mapping for the MYCONTACTDB source system to the CONTACT target.

Table E-1. *Field Mappings for the MYCONTACTDB Source System*

Source System	Source Location	Source Field	Target Object	Target Field	Transformation Rule
MYCONTACTDB	Contact_Users	Full_Name	CONTACT	FIRSTNAME	Extract first name from source with following rule: (EXTRACT,Full_Name, “ ”)
MYCONTACTDB	Contact_Users	Full_Name	CONTACT	MIDDLENAME	Extract first name from source with following rule: (EXTRACT,Full_Name, “ ”, “ ”)

(*continued*)

Table E-1. *(continued)*

Source System	Source Location	Source Field	Target Object	Target Field	Transformation Rule
MYCONTACTDB	Contact_Users	Full_Name	CONTACT	LASTNAME	Extract first name from source with following rule: (EXTRACT FROM RIGHT,Full_Name, “ ”)
MYCONTACTDB	EMAIL spreadsheet	email	CONTACT	EMAIL	Direct transform action (note: only the first EMAIL value from the EMAIL table will be used; the second value (EMAIL) will be added to CONTACT. Description field)

Deployment Steps

[Define each step of the process that will occur during migration, along with a detailed description. The resulting flowchart should provide context for stakeholders and developers as well as a guide for deployment. Each step should then be summarized in a single sentence. For example, Figure E-2 illustrates the flowchart for analyzing, testing, and deploying Contact records into two distinct environments.]

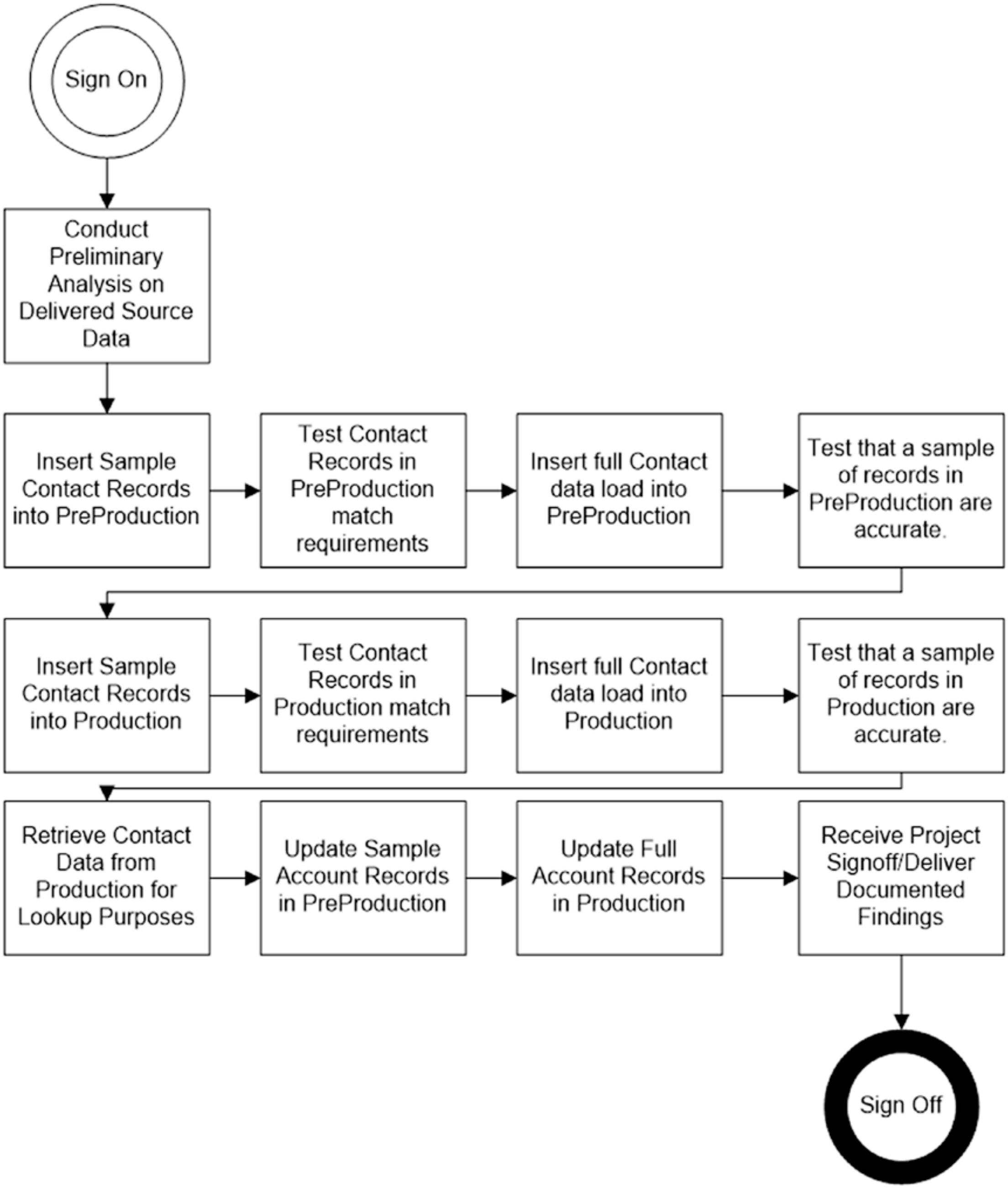

Sign On
Conduct Preliminary Analysis on Delivered Source Data
Insert Sample Contact Records into PreProduction
Test Contact Records in PreProduction match requirements
Insert full Contact data load into PreProduction
Test that a sample of records in PreProduction are accurate.
Insert Sample Contact Records into Production
Test Contact Records in Production match requirements
Insert full Contact data load into Production
Test that a sample of records in Production are accurate.
Retrieve Contact Data from Production for Lookup Purposes
Update Sample Account Records in PreProduction
Update Full Account Records in Production
Receive Project Signoff/Deliver Documented Findings
Sign Off

Figure E-2. *Step-By-Step Deployment*

Description of Prerequisites

- Step 0: Receive records for review; evaluate if existing IDs link with existing contact data (based on LEGACY_ID keys). Create lists of updates versus inserts based on found values.

Contact

- Step 1: Insert sample contact records into Preproduction—five to ten sample contact records will bulk load via the Oracle SQL∗Loader tool.
- Step 2: QA assessment of five to ten records with feedback followed by reiteration/deletion of sample records/reload (if needed).
- Step 3: Migrate remaining contacts through the Oracle SQL∗Loader tool.
- Step 4: Final check of records loaded/send out log files.
- Steps 5–8: Repeat process in the Production environment.

Account

- Step 9: Perform export of CONTACT object stored in Production and import into Staging database for reference lookup.
- Step 10: Update account record sample, followed by QA assessment with feedback (should take no longer than 1 hour).
- Step 11: Update full account load in the Production environment.

Project Signoff/Document Delivery

- During this final phase, the project will receive authorized signoff, and a final document will be placed into Salesforce Content as a reference package.

Log File Delivery

[Indicates where the deployment success and error logs will be stored. For example:]

Individual Delivery

At the end of each stage, log files identifying the individual SUCCESS and ERRORS (i.e., fallout) will be available for review. Oracle SQL*Loader will provide this file in a log report which team members can reach from the corporate Q:\ drive.

Success and errors counts, as well as load time, will also be tracked separately in spreadsheets to ensure appropriate benchmarking.

Portal Storage

Data logs for all deployments, both test and production, are archived in SharePoint and available for review from `https://corporatesharepointwebsite.com`.

Migration Results

[Once the migration is complete, record the high-level success and error results and any additional notes for each object. For example:]

Contact

Inserts

- # Records Inserted: 200000
- # Successes: 145158
- # Failures: 54842
- Due to field names being incorrect, the majority of errors were for empty Last Name (53,811) and First Name (1,031) fields.

Updates

- # Records Updated: 26,079
- 100% Success

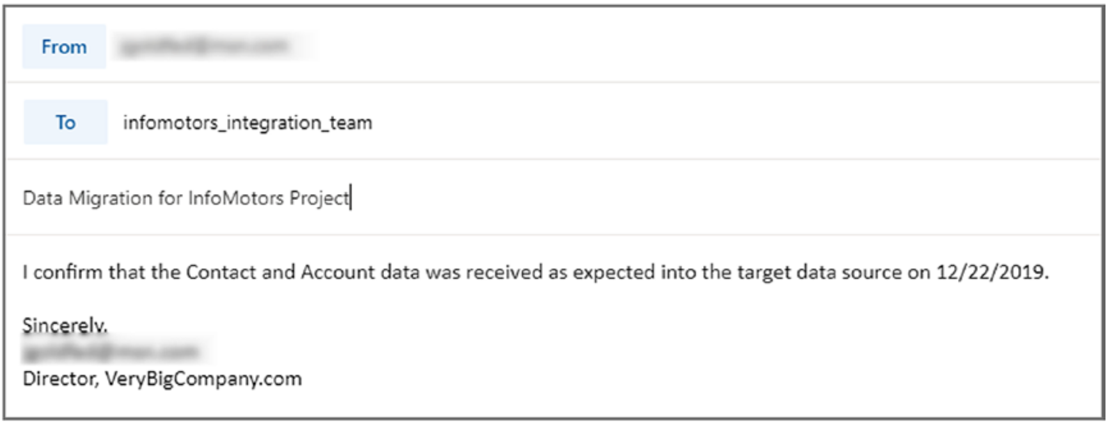
Account

- # New Records Updated: 92192
- # Existing Records Updated: 36
- 100% Success

Final Signoff

[As a final step, prepare an email screenshot from the project sponsor or product owner that the deployment was successful and the target system received the data per requirements. Generally, this signoff happens after deployment and establishes that the right people were satisfied with the deployment results. For example:]

The project sponsor has confirmed that the target data source on 12/22/2019 successfully received Production contact and account data (see Figure E-3), thereby concluding this migration.



From [redacted]

To infomotors_integration_team

Data Migration for InfoMotors Project

I confirm that the Contact and Account data was received as expected into the target data source on 12/22/2019.

Sincerely,
[redacted]
Director, VeryBigCompany.com

Figure E-3. *Sample Project Flow*

APPENDIX F

Migration Results Report Template

The Migration Results Report template provides the summary statistics for data transfer between source and target systems, grouping by table or object, success/error counts, and action items. The integration team should populate this record after each migration, from development to mock testing to eventual production. These results will enable you to predict deployment times as well as errors that may or may not need resolution. You may want to add your final results to the Migration Results section of the Data Migration Process Flow.

The basic layout is self-explanatory, containing nine fields in total (see Table F-1 for field names, definitions, and an example). As with the other templates available, feel free to make whatever modifications, additions, or subtractions you need to ensure your measurements are repeatable, consistent, and predictable.

J. Goldfedder, *Building a Data Integration Team*, https://doi.org/10.1007/978-1-4842-5653-4

***Table F-1.** Issues Log Tracking Fields and Definitions*

Field	Definition	Example
Table/Object	The source table or object you are migrating	ACCOUNT
Method	The type of migration you are performing. There are five types available—Query, Delete, Insert, Update, and Upsert (i.e., a combination of Insert and Update) Most of the time, your activity will be either Insert, Update, or Upsert	Insert
Start Date	The date and time that the migration begins execution. Note that this is a TIMESTAMP field that must include hours, minutes, and, if measurable, seconds	4/18/2019 2:19 PM
End Date	The date and time that the migration begins execution. Note that this is a TIMESTAMP field that must include hours, minutes, and, if measurable, seconds	4/18/2016 2:25 PM
Delta Load	A formula field (in minutes or seconds) that subtracts the end date from the start date	6 minutes (360 seconds)
Total Records	The total number of records available for migration	50
Success Count	The number of records successfully transferred from the source object to the target object	47
Error Count	The number of records unsuccessfully transferred from the source object to the target object	3
Error Message	The error messages (separated by semicolons) produced during the migration. If the same error message occurs multiple times, then list that message once followed by the total number of times it appears	“Value too large for column ‘MIDDLE INITIAL’ (actual: 4, maximum: 1), “Cannot insert NULL into (string)” (2)

Index

J. Goldfedder, *Building a Data Integration Team*, https://doi.org/10.1007/978-1-4842-5653-4