**DataObjectException**

# The 7 milestones of a data migration with Dynamics 365

## How to migrate data from Dynamics CRM to Dynamics 365 on premise

**Whitepaper**
**Version 1.0, September 2019**

DataObjectException

Thierry Sinassamy

Software Architect & Developer

https://yuzuai.ca

## Disclaimer

The white paper describes the process of data migration from a MS SQL Server database to a MS SQL Server database and the process of data migration from Dynamics CRM (on-premises) to Dynamics 365 (Version 8.2 & 9.0, on-premises).

The process occurs within a banking institution with a data migration, and the white paper does not provide any confidential client data related to data migration.

The purpose is to describe the approach, the methodology and the orchestration making the data migration a success. I hope that this approach could help you to orchestrate a process related to a data migration.

# Table of Contents

## Summary

The data migration from Dynamics CRM 2011 (on premise) to Microsoft Dynamics 365 (on premise, Version 8.2) means to migrate the business data, system data and metadata from CRM 2011 to D365. This process requires a methodology adapted to the D365 platform because it is carried out in a "big bang" approach considering the technological context and technical constraints that continually put pressure on the CRM data and metadata transfer process.

The methodology is characterized by a process in terms of strategy and dynamics (stages), which is supported by .NET tools, Dynamics 365, and SQL Server technologies. So, these tools have helped us with the implementation of our strategies: organizing our data model, grouping, and ordering our entities, excluding our entities from the process, and adjusting our migration code. In addition, these tools help us in every step of the process that we had previously established: creating the essential requirements, preparing the basic data, transferring the data, and finalizing the data.

## Keywords

## Confidentiality

The white paper does not provide any confidential client data related to data migration. All the code in this white paper is only related to the CRM SDK365 code (downloadable for free on the Web).

# DataObjectException

Introduction: Questions to ask

Data Migration Options

## Introduction: What questions to ask before migrating?

### The selected data migration approach: big bang approach versus phased approach

In general, a data migration is a mass movement of data from a source to a destination. The data movement can be performed according to following approaches: the "big bang" approach and the "phase " approach. The "Big Bang" approach means that the migration takes place only once and we know when it starts and when it ends. This approach requires, in principle, a "shut down" of the system including the data source to be migrated. On the other hand, the phased approach, although it complicates the migration process because the data migrated to the destination has been modified in the original source, allows smaller pieces of data (and metadata) to be migrated.

Whatever approach is chosen, data migration takes place in a specific scenario, a scenario determined by its technological environment. Before describing our scenario, let us first establish the existing and potential list of options related to data migration.

### Data migration options

Without emphasizing the technical aspect of a case, in this case, whether it is a "Cloud" technology (cloud technology) or an "on Premise" technology (on-site technology), there are several execution contexts [1]:

| OPTION | DATA SOURCE | DATA DESTINATION | DESIGN OF SCENARIO |
|--------|-------------|------------------|--------------------|
| TYPE A | DATABASE | DATABASE | DB/DB |
| TYPE B | DATABASE | SERVICE | DB/SERVICE |
| TYPE C | FLAT FILE DATABASE | DATABASE | FFBD/DB |
| TYPE D | FLAT FILE DATABASE | FLAT FILE DATABASE | FFDB/FFDB |
| TYPE E | SERVICE | SERVICE | SERVICE/SERVICE |

From these options, could be extracted the cases of so-called figures specific to each execution context mentioned in the table above (types a, b, c, d, and e). So, what is a specific option?

### Data migration options in details

A specific scenario is characterized by the following three elements: the data and metadata structure, the technology or platform in which the transfer is constrained and the technological context, and finally, our last element, which is the version and build number of the platform.

### Description of option "type A": from database to database

| OPTION TYPE A (DB/DB) | DATA & METDADATA STRUCTURE | TECHNOLOGY/PLATFORM | VERSION/NUMBER |
|-----------------------|----------------------------|---------------------|----------------|
| TYPE A – 1 | SIMILAR STRUCTURE | SIMILAR PLATEFORM | SIMILAR VERSION |
| TYPE A – 2 | DIFFERENT STRUCTURE | SIMILAR PLATEFORM | DIFFERENT VERSION |

| TYPE A – 3 | DIFFERENT STRUCTURE | DIFFERENT PLATEFORM | DIFFERENT VERSION |
|---|---|---|---|

## Description of option "type B": from database to service

| OPTION TYPE B (DB/SERVICE) | DATA & METDADATA STRUCTURE | TECHNOLOGY/PLATFORM | VERSION/NUMBER |
|---|---|---|---|
| TYPE B – 1 | SIMILAR STRUCTURE | SIMILAR PLATEFORM | SIMILAR VERSION |
| TYPE B – 2 | DIFFERENT STRUCTURE | SIMILAR PLATEFORM | DIFFERENT VERSION |
| TYPE B – 3 | DIFFERENT STRUCTURE | DIFFERENT PLATEFORM | DIFFERENT VERSION |

## Description of option "type C": from « flat file » to database

| OPTION TYPE C (FLAT FILE/DB) | DATA & METDADATA STRUCTURE | TECHNOLOGY/PLATFORM | VERSION/NUMBER |
|---|---|---|---|
| TYPE C – 1 | SIMILAR STRUCTURE | N/A | N/A |
| TYPE B – 2 | DIFFERENT STRUCTURE | N/A | N/A |

## Description of option type D": from « flat file » to « flat File »

| OPTION TYPE D (FLAT FILE/FLAT FILE) | DATA & METDADATA STRUCTURE | TECHNOLOGY/PLATFORM | VERSION/NUMBER |
|---|---|---|---|
| TYPE D - 1 | SIMILAR STRUCTURE | SIMILAR PLATEFORM | SIMILAR VERSION |
| TYPE D – 2 | DIFFERENT STRUCTURE | SIMILAR PLATEFORM | DIFFERENT VERSION |

## Description of "type E": from service to service

| OPTION TYPE E (SERVICE/SERVICE) | DATA & METDADATA STRUCTURE | TECHNOLOGY/PLATFORM | VERSION/NUMBER |
|---|---|---|---|
| TYPE E – 1 | SIMILAR STRUCTURE | SIMILAR PLATEFORM | SIMILAR VERSION |
| TYPE E – 2 | DIFFERENT STRUCTURE | SIMILAR PLATEFORM | DIFFERENT VERSION |
| TYPE E – 3 | DIFFERENT STRUCTURE | DIFFERENT PLATEFORM | DIFFERENT VERSION |

## Selected option

Before the list of types mentioned above, it should be noted that our data migration takes place in TYPE A and more precisely in TYPE a-3.

Migration is the migration of data considering the three elements of our definition regarding our option:

- The data and metadata structure of the 2011 CRM differs from the data and metadata structure of the D365.
- The technology and the platform are different. From a .NET point of view, even if the platforms are based on the .NET Framework, their physical and application architectures differ from each other.
- Therefore, their versions and build numbers differ from each other.

| OPTION TYPE A (DB/DB) | DATA AND METADATA STRUCTURE | TECHNOLOGY/PLATEFORM | VERSION/PLATFORM BUILD NUMBER |
|---|---|---|---|
| TYPE A – 1 | SIMILAR STRUCTURE | SIMILAR PLATEFORM | SIMILAR VERSION |
| TYPE A – 2 | DIFFERENT STRUCTURE | SIMILAR PLATEFORM | DIFFERENT VERSION |
| TYPE A – 3 | DIFFERENT STRUCTURE | DIFFERENT PLATEFORM | DIFFERENT VERSION |

**DataObjectException**

Generic Framework

# Generic Framework

Based on a generic framework, we asked ourselves questions related to the following concepts: From where, how, from what, which tools, what are the phases of migration, what …
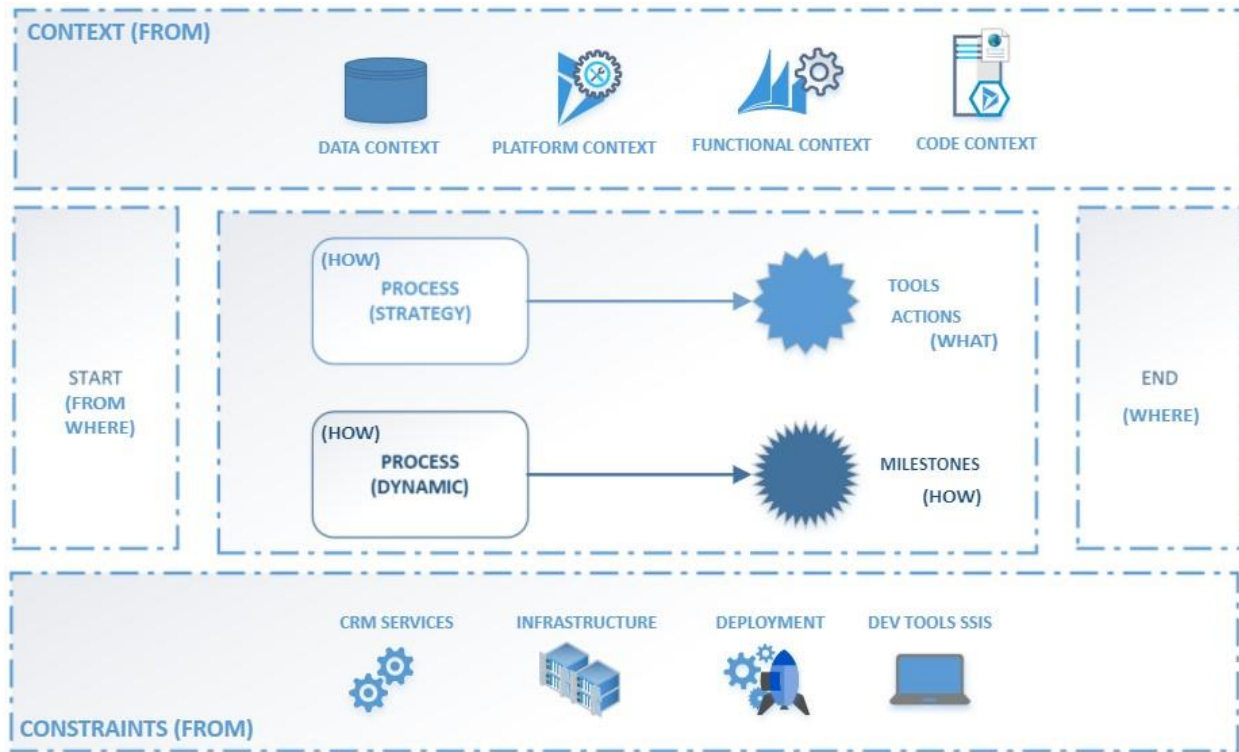


## Implementation assumptions

The fundamental problems we faced in a data migration process are:

- Where does the process of migration come from? This question is not limited to the original data source, but also to the dynamic context in which this data source continues to evolve. The main features of this changing context are:
  - The data volume and complexity of data integrity
  - The evolution of the data structure
  - The evolution of the code deployed in the application system
  - The upgrade of the platform or technology
  - The evolution of dependencies between the components of the system
  - The Hidden behaviour of a platform (black box)

- Where does the process of migration come from? This question also refers to the constraints which put pressure on the migration process, constraints which can play a determining role in the time allocated to migration:
  - The physical infrastructure or resources
  - The initial application architecture

-

- o The Deployment management
- o The Deployment Tools
- o The validation of intermediate results
- o  The time and performance constraints

- What does the process migrate? Of course, data or even metadata, although these may be commonly referred to as" entities". Roughly, how much data and metadata are being migrated?
  - o Which data and metadata from the source database should be migrated?
  - o Which data and metadata of the database need transformation in the process?
  - o Which data and metadata in the source database do not need to be migrated?

- How to execute the process of migration? So, which actions are required to structure and launch the data transfer to the destination? How can we support these actions?

- How do we organize our actions and orchestrate the process?
  - o To structure the migration process over time.
  - o To finalize the new data source.
  - o To control and validate the migration process itself? Also, validate intermediate and results?

# DataObjectException

Context of D365 data migration

# Describing the context of D365 data migration

## Starting point and goal of D365 data migration

The starting point of the D365 migration is a 2011 SQL Server CRM database with the SQL instance version 11.0 (MS SQL Server 2012). From this starting point, the objective is multiple :

- To migrate approximatively 1.5 terabytes of data in a new database with different data and metadata structure, in a SQL Server database (SQL instance version is 14.0, MS SQL Server 2016).
- Migrate knowing the fact: Asynchronous and synchronous CRM services do not behave in the same way as those of Dynamics CRM 2011.
- The migration of nearly 1.5 terabytes of data, including approximatively one terabyte of data related to the e-mail and mail merge templates in the CRM, is a constraint in terms of performance and time.
- To migrate within a short time interval of 10 to 15 hours of acceptable migration.
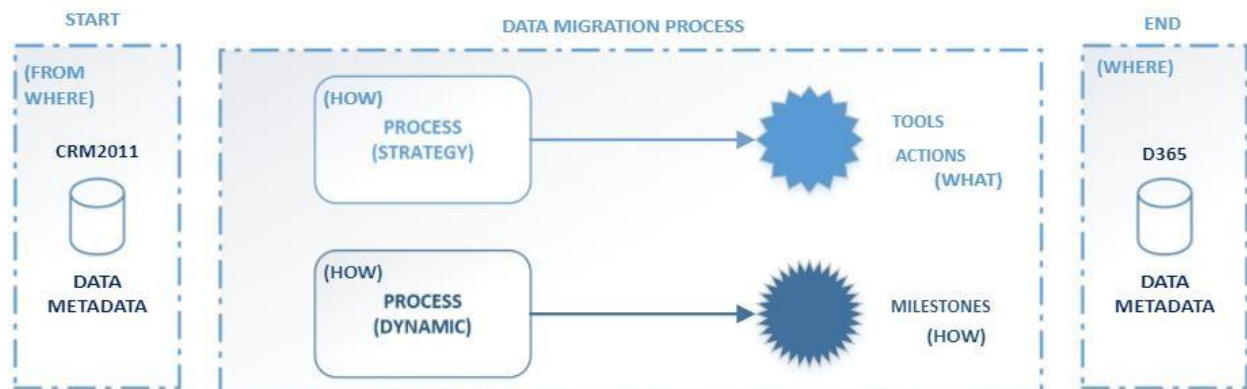


## Contextualization of D365: Two kinds of pressure

### Context of Dynamics CRM 2011 and Dynamics 365

- Data context :
  - Data volume context leading to a larger data integrity issue: 1.5 terabytes of data, data modified by external applications via an integration service (web services).
  - Context of the metadata and data structure via the CRM configuration, context of the native metadata of CRM 2011 different from the D365 metadata context (version 8.2 and 8.3).
  - Data normalization impacted by data denormalization: increase of the relationships (one to-one relationships, one-to-may relationships, many-to-many relationships).
  - Data integrity is a challenge because of the data volume.

- Platform context :
  - Background to the continuous update of the D365 platform: minor evolution of the "Build" numbers (Upgrade from version 8.2 to 8.3).
  - Black box context: CRM services deployed in full-server architecture mode could have unpredictable behaviors and CRM Services code could not be understood in its entirety.

- Functional context :
    - Background of the evolution of the native and custom functionalities : impact of the upgrade and hot fixes (minor and major evolution).
    - Use of non-supported functionalities.

- Code context :
    - Dependency context or dependency nodes between native code and native SQL requests D365: dependency between JavaScript (CRM forms code), C# (Plugins custom code), .NET Framework and finally, native SQL requests of CRM database.
    - Background of the evolution of the code responding to business needs (code-client JS and code-Server Plugin).
    - Deployment of custom code and non-supported code.
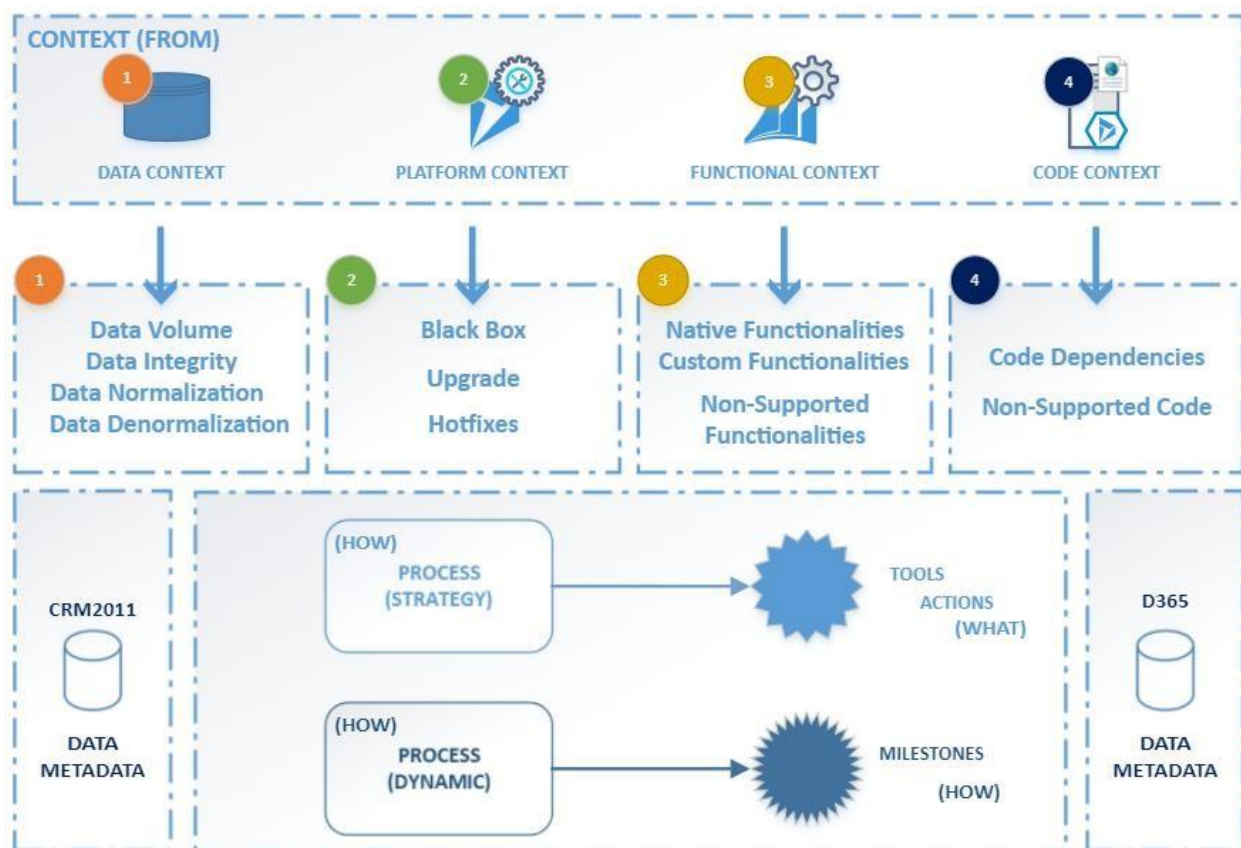


## Technical constraints regarding D365

- D365 Native Architecture constraints:
    - Change of the security model with "Mailbox "concept, change of the logging model by the addition of new "log" tables, change of the basic metadata 1-either of their object: entity, attribute, relation, set option ... - or of their categorisation ("entitytypecode").
    - Behaviour constraints of synchronous CRM (IIS) and asynchronous CRM (Async CRM) services: increasing interaction in contrast to CRM 2011. Asynchronous behavior is much more present in D365-CRM Async Services behave according to the data entered in certain tables (AsyncOperation table...) ... also, constraints related to SQL queries native

to the platform (accessible), queries that are called by CRM processes from the client interface or from backend CRM services.

- ☐ Infrastructure constraints :
  - ☦ Physical resources available to validate the migration process: disk space, processor, transfer of backups …

- ☐ Deployment Constraints :
  - ☦ CRM deployment process from each database backup (groups / zones). The Dynamics 365 Deployment Manager tool allows to create a D365 CRM organization while using the user mapping process, an internal process that will implicitly validate part of the D365 security model (user, business unit and team) and validates the user synchronization process with the "Active Directory" (server-type authentication).

- ☐ Development Tools :
  - ☦ Constraints related to the optimization of the SQL code due to the time interval allowed (12-15 h) for total data migration.
  - ☦ Integrated Development Environment (IDE) used for the development of the code (SSIS) and the deployment of the code generated from the IDE in the database server.
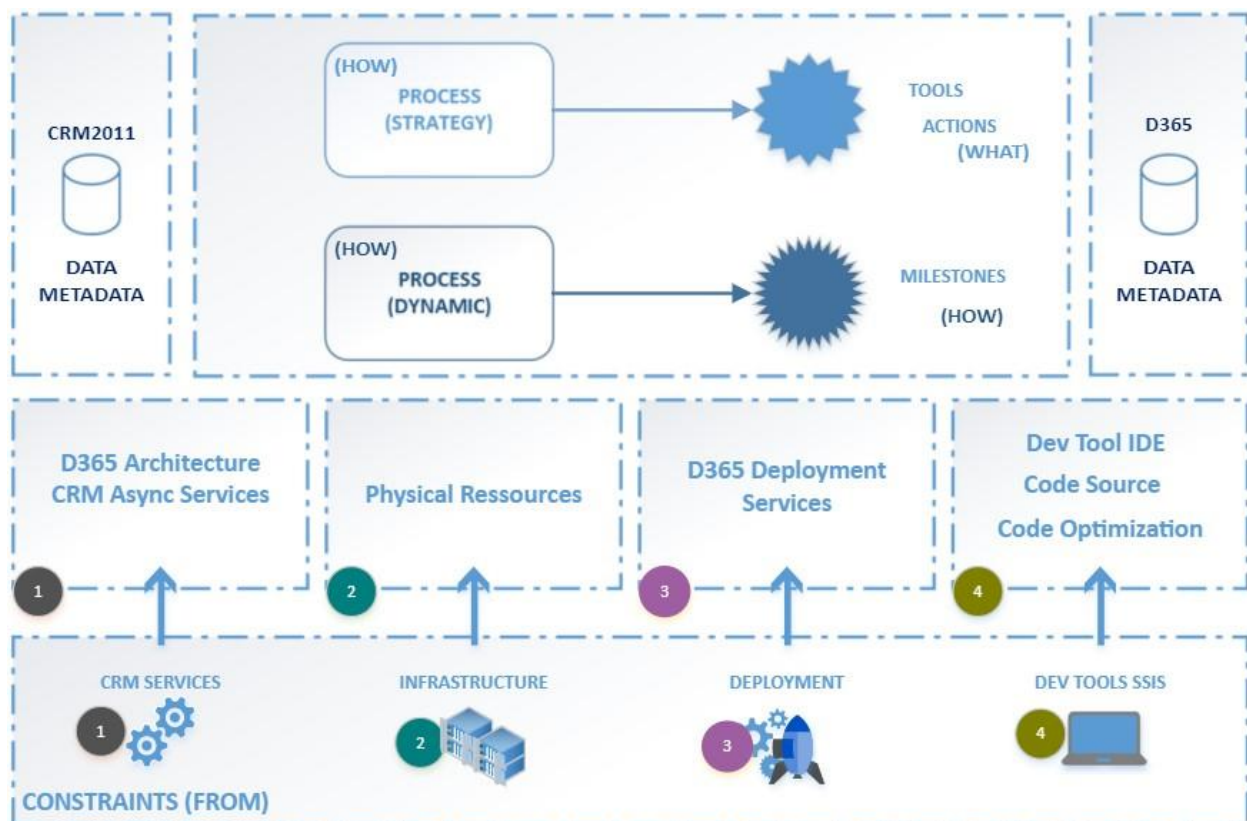


Once the context and the technical constraints have been established, it is time to present the processes of data migration organization: grouping and ordering of entities. These processes or strategies were created using tools specific to the "Dynamics 365 " technology and the ".NET" technology, tools that enabled us to extract relevant and detailed information.

In other words, before approaching the data migration process as such, it is necessary to present the organisational process and its sub-processes as well as the tools that have enabled the organisational processes to reach the desired level of maturity or required for data migration

**DataObjectException**

Strategies and actions in the

7 milestones

# Applying strategies and actions in the 7 milestones

## Describing strategies and actions

### Presenting a generic framework for strategy and actions

Regarding the starting point (page 11), I intend to focus on the strategy as process and the actions helping the process move forward.

### Digging into strategies and actions

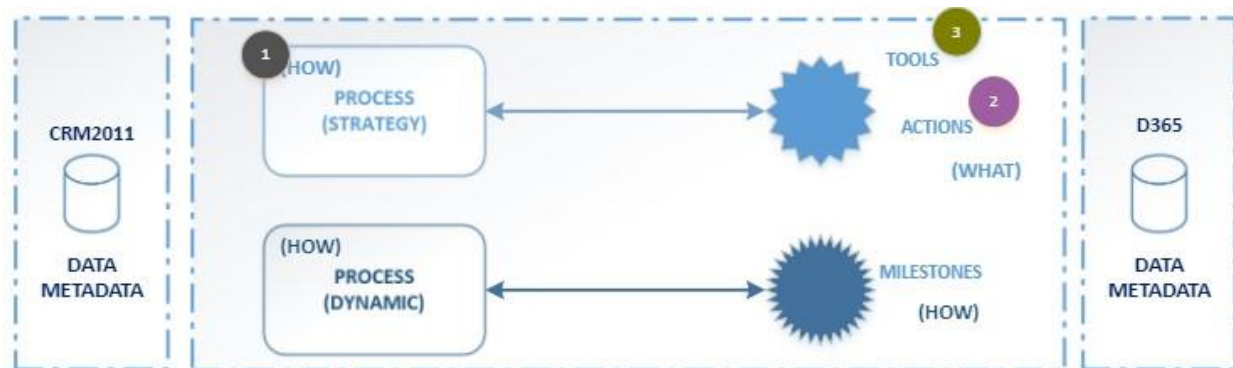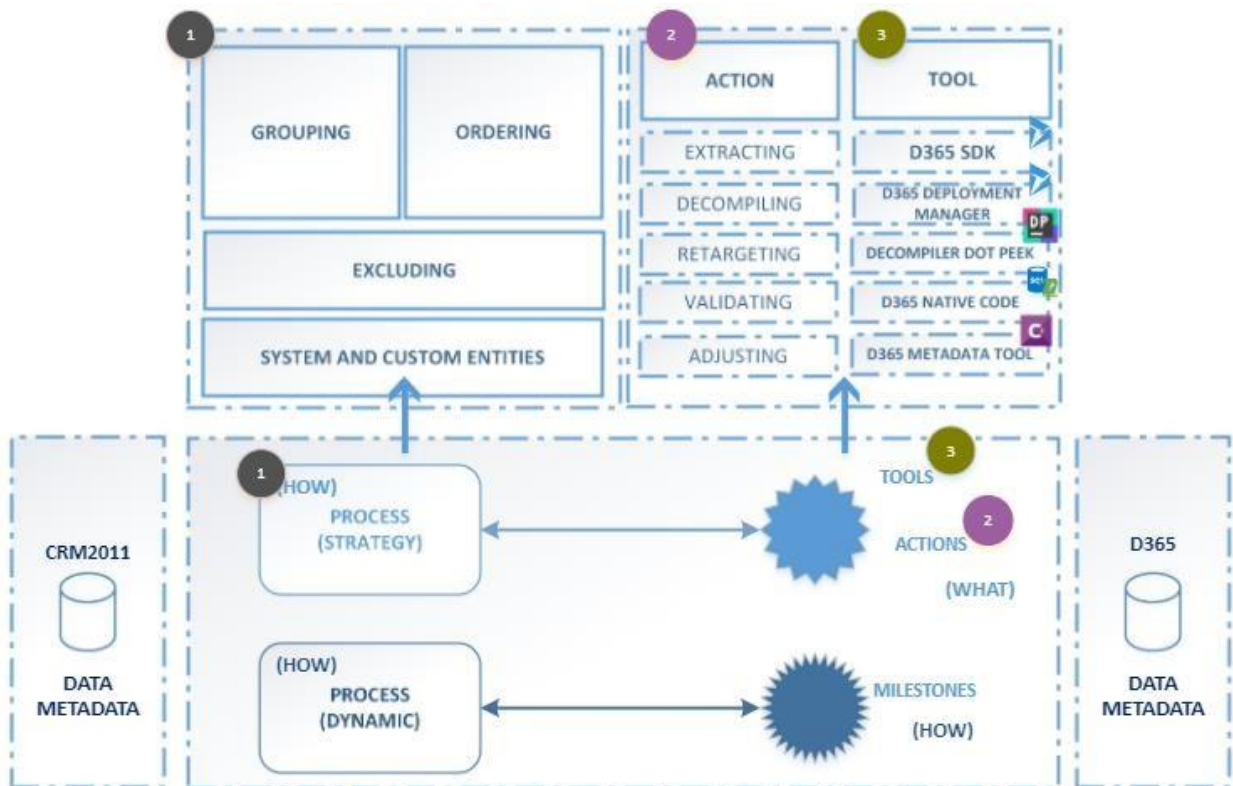| STRATEGIES | ACTION | TOOLS | ICONS |
|---|---|---|---|
| GO (Grouping /Ordering) | Extracting : Metadata, Relationships, Data Model | D365 Metadata Diagram Tool & Custom Console Apps. |  |
| | Extracting : Native and Custom Data Model | SQL Management Studio Object Explorer (DB Diagram functionality) |  |
| | Extracting : D365 Native Code SQL | SQL Management Studio (Views & Stored Procedures) |  |
| | Decompiling D365 .NET Code (C#): Object Model (Business Entities, Relationships), Internal Services, etc. | DotPeek Decompiler |  |
| | Validating: CRM Security Model, CRM Deployment Service. | D365 Deployment Manager |  |
| | Validating: Execution of functionalities (CRM Security Model) | MS SQL Profiler |  |
| | Validating: CRM Security Model, native functionalities | D365 Platform |  |
| GOE (Grouping /Prioritizing /Excluding) | Retargeting: CRM Security Model, Data Model, Relationships | N/A | |
| | Cleaning CRM Data | D365 SDK |  |
| | Adjusting: data migration custom code | N/A | |

Defining strategy through actions: from grouping to prioritizing strategy

It is about creating groups that contain each of the subgroups we have given the name "ZONE". Concretely, in Group 1, there could be several "ZONES" or zone 1, zone 2 ... Each of the zones that may include entities.

In each zone, we can find as many entities as possible of type "SYSTEM "(native to the D365 platform) and entities of type "CUSTOM" (which have been added to the platform to meet business needs). For example, in Group 1 called "Group 1-BASE", we could have 3 zones or the following zones with a variable number of entities : zone 1 including the entities "Currency" and "Owner", zone 2 including "User", "Business Unit", "Relationship", etc.. zone 3 includes "Website", "Goal Metric", "Territory", etc.
So, this involves ordering groups and associated zones. Of course, quantifying the group and the areas established the order of the migration process.

The two processes were constructed and finalized by using tools to generate specific processes or actions or generating processes.

Throughout these processes of grouping and ordering, a process of entity exclusion was built at the same time. Certain entities should not be considered in the migration process.
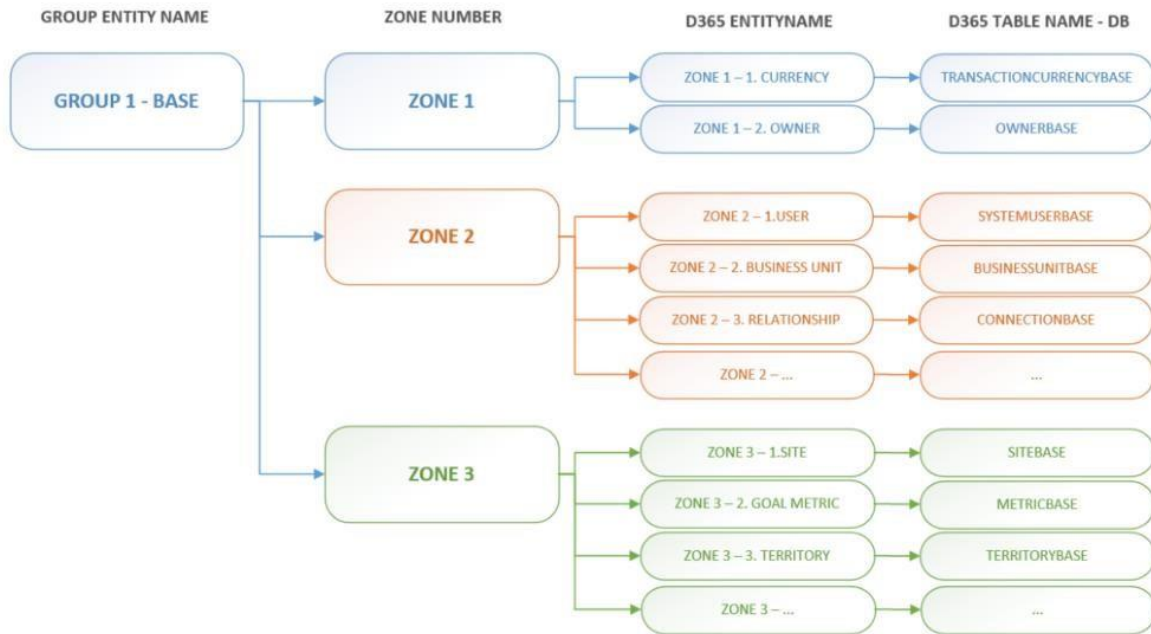
| GROUP ENTITY NAME | ZONE NUMBER | D365 ENTITYNAME | D365 TABLE NAME - DB |
|---|---|---|---|
| GROUP 1 - BASE | ZONE 1 | ZONE 1 – 1. CURRENCY | TRANSACTIONCURRENCYBASE |
| | | ZONE 1 – 2. OWNER | OWNERBASE |
| | ZONE 2 | ZONE 2 – 1.USER | SYSTEMUSERBASE |
| | | ZONE 2 – 2. BUSINESS UNIT | BUSINESSUNITBASE |
| | | ZONE 2 – 3. RELATIONSHIP | CONNECTIONBASE |
| | | ZONE 2 – … | … |
| | ZONE 3 | ZONE 3 – 1.SITE | SITEBASE |
| | | ZONE 3 – 2. GOAL METRIC | METRICBASE |
| | | ZONE 3 – 3. TERRITORY | TERRITORYBASE |
| | | ZONE 3 – … | … |

Defining strategy through actions: from grouping and prioritizing to excluding strategy

As result, it has become possible to establish two groups of entities: those that are an integral part of the data migration and those that are excluded from the same migration process.

There are many reasons why some entities and therefore some data migration has been set aside. The following is a list of the reasons that led to the exclusion of the said entities or tables:

- Entities or tables that are "populated" by importing our additional configuration (forms, attributes, views, security roles, etc.) and custom code (client code like JavaScript, CRM solution files ("zip")).
- Entities or tables that are "populated" by the system in the background (by the synchronous and asynchronous services of the CRM): some are used as logging and others as the source of operation for the CRM services.
- Entities or tables that are not" populated" because they were not in the original system (CRM 2011) and therefore, no need to migrate data in the D365 version.
- Entities or tables that are not "populated" because they are new in D365 and did not exist in 2011 version.

The table below presents the list of tables (entities), organized according to the four reasons mentioned above. However, the list is not exhaustive for a clear reason.

| EXCLUSION REASONS | ENTITIES EXCLUDED FROM MIGRATION |
|---|---|
| CUSTOMIZATION AND CUSTOM CODE | TABLES RELATED TO METADATA… |
| CRM SYSTEM/INTERNAL PROCESS | TABLES RELATED TO SYSTEM/INTERNAL PROCESS (ASYNCOPERATION…) |
| CRM 2011 DATABASE SOURCE WE DON'T WANT OR DON'T NEED ANYMORE | TABLES RELATED TO QUOTE… |
| NEW ENTITIES IN NEW D365 (VERSION 8 & 9) | TABLES RELATED TO MAILBOX…. |

# How to process data migration through 7 milestones

## Presenting a generic framework for 7 milestones

Regarding the starting point (page 11), I intend to focus on the dynamics as process and on the milestones as a part of the process.
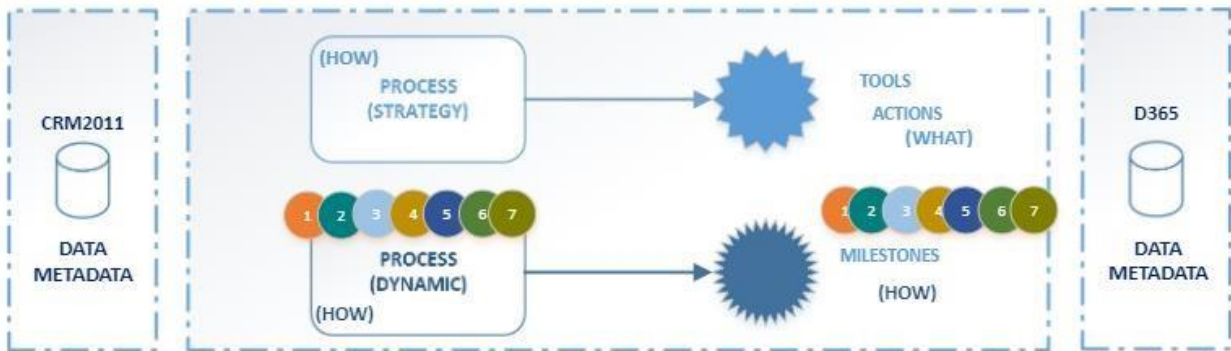


## Digging into the milestones

There are 7 milestones to consider: creating necessary requirements (number 1), preparing data related to CRM entities and metadata (number 2), transferring CRM security data (number 3), validating CRM security data (number 4), transferring CRM data (number 5), validating CRM data (number 6) and completing CRM data (number 7).



| 1 MILESTONE 1 CREATING REQUIREMENTS | 2 MILESTONE 2 PREPARING DATA | 3 MILESTONE 3 TRANSFERING DATA I | 4 MILESTONE 4 VALIDATING DATA | 5 MILESTONE 5 MIGRATING DATA II | 6 MILESTONE 6 VALIDATING DATA II | 7 MILESTONE 7 COMPLETING DATA |
|---|---|---|---|---|---|---|
| Database Template | Cleaning CRM Security Data | | CRM Deployment | | CRM Deployment | |
| | Cleaning CRM Data | CRM Security Data Migration | CRM Security Data Validation | CRM Data Migration | CRM Data Validation | CRM Deployment |
| Database Tool | Grouping CRM Security Entities | | Retargeting CRM Security Entities | | Retargeting CRM Security Entities | |
| Decompiling Native Code | Prioritizing CRM Security Entities | CRM Security Data Logging | Grouping CRM Security Entities | CRM Data Logging | Grouping CRM Security Entities | CRM Security Data Validation |
| Extracting Entities | | CRM Security Data Reporting | Prioritizing CRM Security Entities | CRM Data Reporting | Prioritizing CRM Security Entities | CRM Data Validation |
| | | | Adjusting Migration Code | | Adjusting Migration Code | |

## Preparing database as template

- Step 1: D365 "Deployment Manager" will help to create a CRM database "CRM_MSCRM" (CRM SQL Instance without custom configuration and code). Hence, it will create a CRM Organization (application side).

- Step 2: Importing the custom "CRM solutions". Hence, it will complete the native data and metadata model of CRM that will later be associated with our data.

- Step 3: Validating the relevant foreign keys following the previous step 2.

- Step 4: Creating a database backup will create our basic "Template" for future identical migrations.

- Step 5: D365 "Deployment Manager" will help to delete dig the CRM Organization created at step 1.

## Preparing database as tool and as development support

- Step 1: Creating a database tool as a toolbox useful as much for the source database (CRM 2011) as for the destination database (D365).

- Step 2: Creating the first layer of abstraction in our toolbox including SQL objects as synonyms [2]. Hence, it will support the process of migration.

- Step 3: Completing the first layer creating a second layer of abstraction i.e. additional objects as stocked procedures and functions [3]. Hence, it will as well, support the process of migration.

- Step 4: Creating a database tool backup for future identical migrations.

## Extracting SQL native code

- Step 1: D365 Metadata Diagram Tool [4] will help to extract the custom entities model, including the native entities. Hence, the tool will dig into details :
    - logical relationships because the database could be highly denormalized: one-to-one relationships, one-to-many relationships, many-to-may relationships.
    - level of ownership: business-owned, organization-owned, user-owned or not owned.

- Step 2: Creating a file backup will be helpful for future migrations.

- Step 3: Extracting the CRM model from the "Object Explorer" [5] in the SQL instance will create an additional tool to visualize the relationships. However, it will provide only a high-level view.

## Extracting CRM entities

- Step 1: DotPeek Tool [6] will help to decompile the CRM native code (.Net library accessible in a specific directory in the server) [7]. Hence, the tool will dig into details:

✞   entity and business layer (i.e. business entity)
✞   entity and data access layer (i.e. entity internal service)
✞   entity and configuration layer (i.e. object model configuration)
✞   client layer (i.e. authentication for Active Directory)
✞   entity and metadata layer (i.e. managed properties)

Milestone 2: Preparing data related to the CRM Organization (database)

Step 1: Cleaning data (security, system and business data)

Step 2: Grouping and prioritizing CRM security data

Milestone is mainly composed of two strategies mentioned above and will be impacted by step 1:

☐   Implementing actions plan with specific tools
☐   Defining strategy through actions: from grouping and prioritizing to excluding strategy

The strategies focus only on CRM security entities: System User, Team and Business Unit. Moreover, they focus on all the entities related to CRM security model: logical relationships (i.e. relationship with "ownership") and physical (i.e. relationships with tables foreign key) [8].
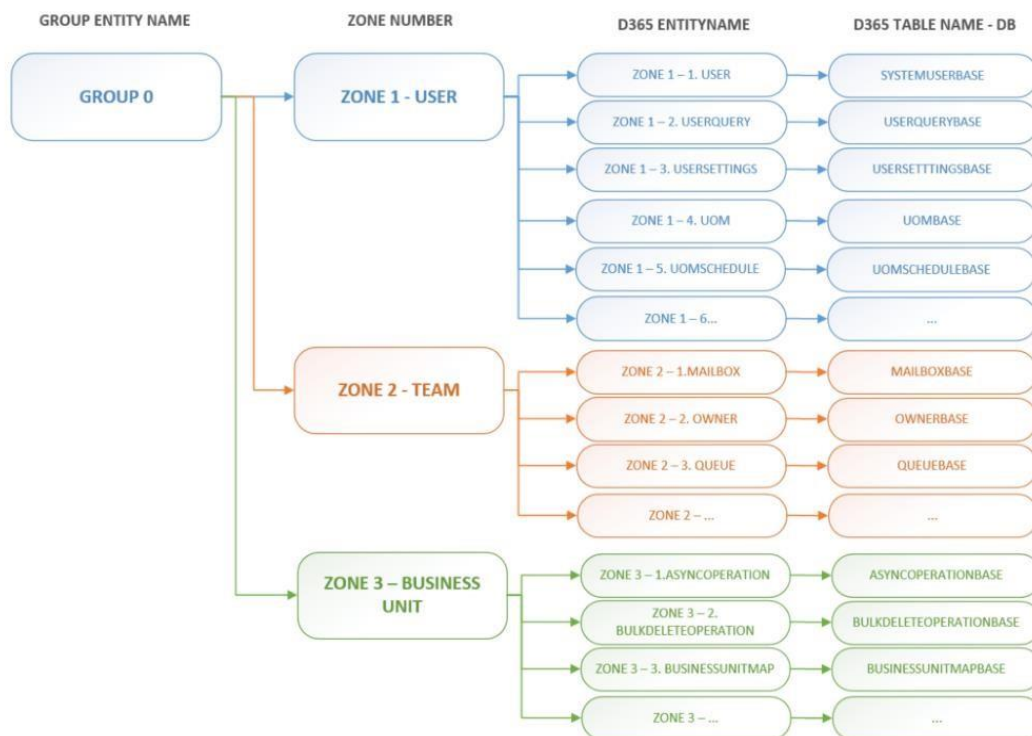


Milestone 3: Transferring CRM security data to D365

☐   Step 1: Transferring entities related to D365 security model and D365 metadata
First, the process involves preparing the main entities associated with the D365 metadata:

- ✞ "Solution" entity / "SolutionBase" table (name in database "CRM_MSCRM").
- ✞ "Solution Component" entity / "SolutionComponentBase" table (name in database "CRM_MSCRM")
- ✞ "Localized Label" entity / "LocalizedLabelBase" table (name in database "CRM_MSCRM")

In a way, it is a question of updating identifying him unique (Guid or Globally Unique Identifier) of entity "solution" in mentioned entities (solution, solution component localised and quality-label) to replace that of solution D365 (initially created in "Template" of base) with that of solution CRM 2011.

Also, the process consists in preparing the main entities related to the D365 model of security:

- ✞ Entity "Team"or table "TeamBase" (table name in database "CRM_MSCRM")
- ✞ Entity "User" or table "SystemUserBase" (table name in database "CRM_MSCRM")
- ✞ Entity "Business Unit" or table "BusinessUnitBase" (table name in database "CRM_MSCRM")
- ✞ Entity "Queue" or table "QueueBase" (table name in database "CRM_MSCRM")
- ✞ Entity "Mailbox"or table "MailboxBase" (table name in database "CRM_MSCRM")

- ⬜ Step 2: Logging data related to CRM security entities

During the data transfer, once the updated entity, could be logged, it is possible to show the result of changed data. Once the logging ended for every table, an electronic mail is sent to the desired addressees.

- ⬜ Step 3: Generating data related to CRM security entities

Once CRM security data have been logged, generating a report of results in a « Excel » file. The latter could introduce a more deepened explanation regarding the logged data.

## Milestone 4 : Validating Data

- ⬜ Step 1: CRM Deployment [9]

  Complete validation limits itself not only to a report of data containing logical explanation. It must go farther as much as SDK of CRM is capable of validating results got in the table of logging.
  D365 platform contains in a way a « SDK Deployment ». The latter helps us to import a CRM organisation from the database.  During the process, it is mandatory to undertake a mapping of users. Hence, it helps to synchronise the user with « Active Directory » and finally, it helps us to validate a part of CRM security Model.

- ⬜ Step 2: CRM Security Data Validation

  - ✞ Visualizing logged CRM security data (previously, step 2 – Milestone 3)
  - ✞ Visualizing CRM security data reporting (previously, step 3 – Milestone 3)
  - ✞ Functional testing and CRM security entities: Once the ended importation, it is necessary to go further and so, to call CRM services CRM and test the data related to CRM security Model. Hence, we execute CRUD operations (Create, Read, Update and Delete) and we try to disable data related to the user [10].

- ⬜ Step 3: Retargeting, Grouping and Prioritizing

Retarget the entities will help to regroup and prioritize the CRM security entities. Retargeting means that we must focus as much as possible on the logical and physical relationships.

- Step 4: Adjusting Migration Code

If we need to retarget the security entities and their relationships, we need to adjust our migration code.

- Step 5: Backing up the database (CRM security Model)

If we do not need to apply steps 3 and 4, we could right away back up the database and access to the next milestone.

## Milestone 5: Completing CRM data transfer (business and system)

Milestone 4 is a requirement for the milestone because we cannot migrate all the CRM data if CRM security data migration is not a success.

- Step 1: Transferring entities related to D365 business and system data. This step should be impacted by the step 1 of milestone 2 (preparing data).

- Step 2: Logging data related to CRM entities.

During the data transfer, once the updated entity, could be logged, it is possible to show the result of changed data. Once the logging ended for every table, an electronic mail is sent to the desired addressees.

- Step 3: Generating data related to CRM entities.

Once CRM data have been logged, generating a report of results in a « Excel » file. The latter could introduce a more deepened explanation regarding the logged data.

## Milestone 6 : Validating migrated data

- Step 1: CRM Deployment [11]

Complete validation limits itself not only to a report of data containing logical explanation. It must go farther as much as SDK of CRM is capable of validating results got in the table of logging. Before validating the migration, we need to import a CRM organisation from the database ("SDK Deployment").

- Step 2: CRM business and system Data Validation

  - Visualizing logged CRM data (previously, step 2 – Milestone 5)
  - Visualizing CRM data reporting (previously, step 3 – Milestone 5)
  - Functional testing and CRM entities: Once the ended importation, it is necessary to go further and so, to call CRM services CRM and test the data related to CRM Model. Hence, we execute CRUD operations (Create, Read, Update and Delete) and we try to disable data [12].

- Step 3 : Retargeting, Grouping and Prioritizing

Retarget the entities will help to regroup and prioritize the CRM entities. Retargeting means that we must focus as much as possible on the logical and physical relationships.

- ☐ Step 4 : Adjusting Migration Code

  If we need to retarget the entities and their relationships, we need to adjust our migration code.
- ☐ Step 5: Backing up the database (CRM business and system Model)

  If we do not need to apply steps 3 and 4, we could right away back up the database and access to the next milestone.

### Milestone 7 : Finalization of data

- ☐ Step 1 : CRM Deployment : two subprocesses

  - ✝ Deploy the requirements related to the database: manage the indexes and the symmetrical keys at the security level.
  - ✝ Import the CRM Organization from the database ("SDK Deployment").
  - ✝ Import the CRM configuration (business and system).
  - ✝ Import the CRM custom code.

- ☐ Step 2: Primary validation - CRM Security Data Validation (apply step 2 and 3, milestone 3)

- ☐ Step 3: CRM Data Validation (apply step 2 and 3, milestone 6)

## Iterating between the milestones

There are 4 milestones to consider: milestone 3, milestone 4, milestone 5 and milestone 6.
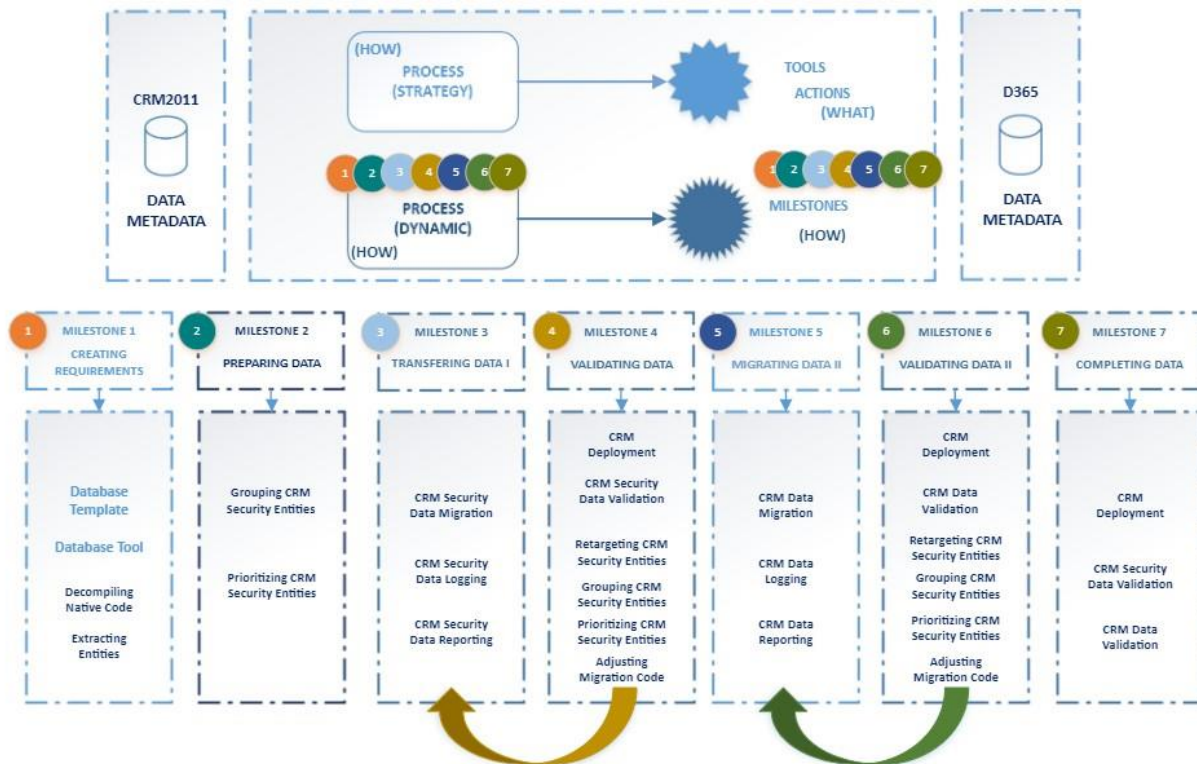
The first iteration occurs between the milestones 3 and 4: after validating CRM security data (User, Business unit, Team) and data related to CRM security data (milestone 4), a failed validation takes a step backward (milestone 3).

Once the milestone 4 is a success, the second iteration occurs between the milestones 5 and 6: after validating CRM data (business and system data), a failed validation takes a step backward (milestone 5).

**DataObjectException**

Conclusion: Overview

# CONCLUSION

**CRM**

**1** Creating Requirements
- DB Template
- DB Tool
- Decompiling native Code
- Extracting CRM/D365 Entities

**2** Preparing Data
- Grouping CRM/D365 Entities
- Prioritizing CRM/D365 Entities

**3** Transfering CRM Security data
- CRM security Data Migration
- CRM security Data Logging
- CRM security Data reporting

**4** Validating CRM Security data
- CRM Deployment
- CRM Security data validation
- Retargeting CRM Security Entities
- Grouping CRM Security data
- Prioritizing CRM Security data
- Adjusting Migration Code

**5** Transfering CRM data
- CRM Data Migration
- CRM Data Logging
- CRM Data reporting

**6** Validating CRM data
- CRM Deployment
- CRM data validation
- Retargeting CRM Entities
- Grouping CRM data
- Prioritizing CRM data
- Adjusting Migration Code

**7** Completing CRM data
- CRM Deployment
- CRM Security data validation
- CRM data validation

**D365** → Creating CRM Organization

# References

[1] The current list is not intended to be exhaustive, and the description is intended primarily to be conceptual and nontechnological.

[2] Synonyms in SQL Instance – Database Tool

```
Here is an example of creation of synonyms for the source database (CRM) :
USE CRM_TOOLS
GO
CREATE SYNONYM [dbo].[Syn_Mig_CRM2011_AccountTable] FOR [CRM2011DB_MSCRM].[dbo].[AccountBase]


Here is an example of creation of synonyms for the source database (D365) :
USE CRM_TOOLS
GO
CREATE SYNONYM [dbo].[Syn_Mig_D365_AccountTable] FOR [D365DB_MSCRM].[dbo].[AccountBase]
```

[3] Stored procedures and Functions in SQL Instance – Database Tool

```
 USE [CRM_TOOLS]
 GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[sp_Update_BusinessUnit]
     @ParamSet UNIQUEIDENTIFIER,
     @ParamExp UNIQUEIDENTIFIER,
     @BU_Name NVARCHAR(255)
AS
BEGIN
SET NOCOUNT ON ;
UPDATE [syn_Mig_D365_BusinessUnitBase_Table] set [Name] = @BU_Name   WHERE  [BusinessUnitId]   = @ParamExp;
UPDATE [syn_Mig_D365_BusinessUnitBase_Table] set [BusinessUnitId]=@ParamSet where  [BusinessUnitId]  = @ParamExp;
etc.
```

[4] "Metadata Diagram Tool" helps to generate the partial or complete CRM entities model ("system", "custom") in a "Visio" file.
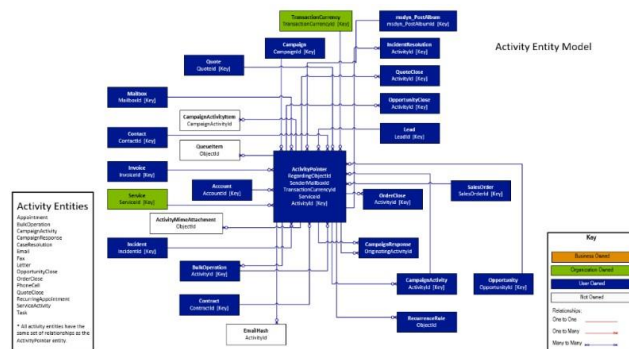


[5] "Object Explorer" View in the "SQL Management Studio" tool helps to centralize our visualization around the same concept extracted by the tool "Metadata Diagram Tool".

[6] Decompiling with DotPeek affords to glimpse the use of entities and the existing relations between entities.



[7] All the D365 .NET library is in the following directory:



| Name | Date modified | Type | Size |
|---|---|---|---|
| Microsoft.Crm.Asynchronous.System.dll | 2017-10-11 3:43 PM | Application extens... | 254 KB |
| Microsoft.Crm.Asynchronous.Workflow.dll | 2017-10-11 3:43 PM | Application extens... | 75 KB |
| Microsoft.Crm.Audit.dll | 2017-10-11 3:43 PM | Application extens... | 73 KB |
| Microsoft.Crm.Authentication.Azure.dll | 2017-10-11 3:43 PM | Application extens... | 26 KB |
| Microsoft.Crm.Authentication.dll | 2017-10-11 3:43 PM | Application extens... | 291 KB |
| Microsoft.Crm.Authentication.S2S.dll | 2017-10-11 3:43 PM | Application extens... | 91 KB |
| Microsoft.Crm.CalculatedField.dll | 2017-10-11 3:43 PM | Application extens... | 169 KB |
| Microsoft.Crm.Client.Shared.dll | 2017-10-11 3:43 PM | Application extens... | 119 KB |

[8] In a way, it is a question of updating unique identifier only (GUID) and to replace that of D365 (who comes from "Template" of base) with that of CRM 2011. This modification touches the list of entities listed above but also, entities which are linked of a point of "logical" life (for example by the concept of "OwnerId") and "physical appearance" (by the foreign key in the table). Here is below the non-exhaustive list of entities:

- Entity "Team" : Mailbox, Owner, …
- Entity "User" : AttributeMap, ApplicationFile, …
- Entity "Business Unit": AsyncOperation, BulkDeleteOperation, …

[9] [11]

[10] [12] D365 CRUD operations :

- For « User », following operations were possible: Read, Update and Create. Attention, naturally, in D365, it is not possible to abolish a user (Delete). We can only disable the user.
  Operation « disable » could be like operation « delete » : this operation was possible.
- For « Team », following operations were possible also: Create, Read, Update and Delete.
- For « Business Joins », following operations were possible also, Create, Read, Update and Delete.

## Legal Note

This document may not be circulated without prior written consent. You may quote or use the white paper or part of it for any type of document, research or presentation, citation and referencing rules apply to this white paper. For more information, please do not hesitate to contact the author.

This white paper cannot be used by you or one client of your client, in a single product without the author's consent.

## Glossary

Active Directory: AD is a directory service developed by Microsoft for Windows domain networks. It is included in most Windows Server operating systems as a set of processes and services (https://en.wikipedia.org/wiki/Active_Directory).

CRUD Operations: In computer programming, create, read, update, and delete (CRUD) are the four basic functions of persistent storage (https://en.wikipedia.org/wiki/Create,_read,_update_and_delete).

Deployment Manager: Dynamics 365 Deployment Manager is a Microsoft Management Console (MMC) snap-in that system administrators and value-added resellers use to manage Deployment Administrator accounts, organizations, servers, and licenses for Dynamics 365 for Customer Engagement deployments (https://docs.microsoft.com/en-us/dynamics365/customerengagement/on-premises/deploy/usedeployment-manager-to-manage-the-deployment).

DLL: Library that contains code and data that can be used by more than one program at the same time (https://docs.microsoft.com/en-us/troubleshoot/windows-client/deployment/dynamic-link-library).

Functions: SQL Server Functions are useful objects in SQL Server databases. A SQL Server function is a code snippet that can be executed on a SQL Server (https://www.c-sharpcorner.com/UploadFile/b926a6/function-operation-in-sql-database/).

GUID: A universally unique identifier (UUID) is a globally unique identifier (GUID). This number is created by running the Guidgen.exe command line program (https://docs.microsoft.com/en-us/previousversions/windows/desktop/automat/globally-unique-identifier-guid-).

Jet Brains dotPeek : Free .NET Decompiler tool (https://www.jetbrains.com/decompiler/) Metadata:

Data that provides information about other data.

SDK Deployment: Dynamics 365 Software Development Kit related to CRM Deployment. SDK helps to manage CRM organization (i.e. import a new organization from a database).

Synonyms: alias or alternative name for a database object such as a table, view, stored procedure, userdefined function, and sequence. A synonym provides you with many benefits if you use it properly (https://www.sqlservertutorial.net/sql-server-basics/sql-server-synonym/).

Stored procedures: SQL Server stored procedures are used to group one or more Transact-SQL statements into logical units. The stored procedures are stored as named objects in the SQL Server Database Server (https://www.sqlservertutorial.net/sql-server-stored-procedures/).

## About Me and philosophy

Thierry Sinassamy - I strongly advocate a methodology of data migration targeting D365 on-premises. The methodology describes 7 milestones already implemented during my experiences and the way of orchestrating a data migration.
Regardless of whether the target is on-premises or cloud, I am confident that the 7 milestones should always belong to our journey and could help you to anticipate the milestones.

## Contacts

DataObjectException Inc.
Thierry Sinassamy 1-514-883-6068
thierry.sinassamy@dataobjectexception.com
https://www.yuzuai.ca