migFx – Bridge

Our generic approach to integration with any target system as part of a complex data migration.

migFx is a comprehensive solution for the mapping, execution and tracking of complex data migration. migFx applies a fully developed generic migration model to any data migration.

Explaining migFx in greater detail is outside the scope of this paper. For a deeper understanding, please refer to our migFx-Concepts document available here.

Introduction

As a generic solution, migFx can handle data migration to *any* Target System and delivers many valuable benefits right out of the box.

The migFx Solution spans the entire data migration from Source to Target. The architecture of migFx allows for the clean separation between Target System related elements and Source System related elements. A benefit of this separation is a large potential for reuse of the Target related elements from migration project to migration project.

In hopp we exploit this potential by offering ready-made bridges to commonly available Target Systems, such as Oracle eBusiness Suite, Oracle Fusion, Microsoft Dynamics, SAP etc.

Bespoke

In order to apply migFx to a *specific* Target System, *extensions* to migFx can be developed to achieve deeper integration of migFx with the given Target System. migFx cleanly separates the mapping in a *Target Map* and a *Source Map*. The Source Map is linked to the legacy Source System while the Target Map is linked to the Target System.

In the case of a bespoke Target System, extensions are typically developed in collaboration with hopp as part of the deployment of migFx, while the Target Map is typically developed as part of the migration project.

Ready-made

In the case of commonly available Target Systems hopp offers ready-made *migFx Bridges*. A migFx Bridge contains the necessary *extensions* to integrate with the given Target System as well as a *Target Map* ready to go.

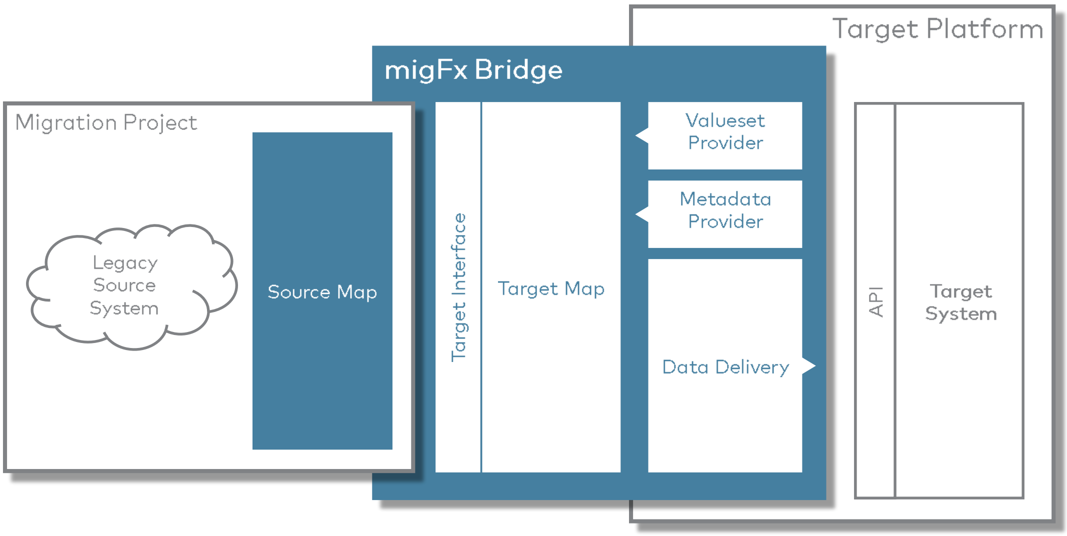
With a migFx Bridge, any data migration project into a commonly available Target System will gain a valuable head-start.

Regardless of Target System, a migFx Bridge contains a Target Map and a given set of extensions. This note highlights the elements of a migFx Bridge and serves as a reference document for the notes describing the specific migFx Bridges offered by hopp.

Please contact us to understand which specific migFx Bridges are already available from hopp or how to build your own bridge to your bespoke target system.

Bridge elements

A migFx Bridge extends the generic migFx Solution and fits in the overall path from Source- to Target System as illustrated below.



Target Platform

The Target System naturally resides on some sort of technological platform, for instance Linux/Oracle, Windows Server/Sql Server etc.

migFx Extensions delivered with a migFx Bridge may need to install part of the necessary functionality on the Target Platform. One example could be a Rest API or similar that can be called by the migFx Extension through the Extension Interfaces to retrieve either Metadata or Valuesets.

Target System

For migFx, the Target System is a black-box. migFx needs to now the metadata describing what must be delivered to the Target System as a result of the data migration. The actual delivery of the data must be handled by the migFx Bridge.

The Target System will invariably contain a setup of base data that the migrated data must conform to, i.e. many different sets of valid values for certain fields in the Target Data. The more migFx knows about these Valuesets, the better migFx can validate and ensure that no data escapes the migration that if it will not be accepted by the Target System.

API

In some (rare) cases, the data delivered from migFx as a result of the migration is inserted directly into the tables of a database inside the Target System. But it is more common that the data must be delivered through an API exposed by the Target System.

migFx must know the metadata describing the data that must be delivered to the API. Any API may reject data from the migration and the migFx Bridge must handle the housekeeping necessary in this respect.

migFx bridge

The migFx Bridge contains the elements that is added on to the generic migFx solution in order to integrate with the given Target System. A migFx Bridge will typically contain a Target Map and a set of Extensions.

Target Map

The Target Map is an important and valuable part of any specific migFx implementation. The Target Map is developed in the migFx Studio application and serves several purposes:

- It exposes a *Target Interface* specifying the data that must be received from any Source Map
- It *validates* the data from the Source Map and ensures a very high quality of the data sent on to the Target System.
- It *maps* how to create the Target Data from the data received through the Target Interface

The validation performed in the Target Map typically makes extensive use of sets of valid values extracted from the Target System.

It is a feature of migFx that the validation significantly improves over the lifetime of the migration projects and indeed between migration projects reusing the same Target Map.

An important part of a migFx Bridge is to deliver a Target Map as close to finished as possible. In some cases, it may be finished. But in most cases, it is only realistic to expect that customizations in the Target System may require a final touch up of the Target Map.

Extensions

migFx contains multiple extension points. An extension point is typically a mechanism where migFx calls out through a publicly exposed interface in order to collect or deliver data. A given extension is implemented by extending the relevant interface with functionality accessing the Target System some way or other.

You can read our detailed reference library for the migFx Extension Interfaces here.

Metadata Provider

The Target Map needs to know the Metadata describing the data to deliver to the Target System. The Metadata Provider Extension interface contains methods that can be called by the Studio in order to import these metadata.

Implementing a Metadata Provider Extension means extending these interfaces so the call from Studio will in fact be routed towards the Target System to retrieve the metadata.

Having an easy and fast way to import metadata into Studio is beneficial in situations where the Target System itself is being modified as part of the migration project ('moving target').

Valueset Provider

The Target Map that comes with a migFx Bridge will typically contain a number of so-called Valuesets. In the Target Map, a Valueset is simply a definition of a table of data – not the data themselves.

When migFx executes the migration, the Valuesets are populated with data by calling out through a Valueset Provider interface. Just as the Metadata Provider, implementing a Valueset Provider simply means extending these interfaces so the call from migFx is routed through to the Target System from where the data are retrieved.

Delivery

migFx executes the migration right through to the point when the Target Data has been produced and resides inside migFx. Of course, the story hardly ends there – the data must be delivered from migFx to the Target System.

This is clearly an extension point. migFx will launch a job calling out through an Extension Interface to off-load the Target Data. To implement a Delivery Extension, this off-load interface must be extended. The implementation of this extension is highly dependent on the Target System in question. In some cases, it may as simple as writing the data to a set of files that are handled in a following processing outside of the knowledge of migFx. In other cases, the extension may be more evolved.

Regardless of actual implementation, there are some common considerations and task that a Delivery Extension and migFx must handle in unison.

Table 1: Common considerations

Keys
migFx creates and maintains complete referential integrity in the Target Data by using numeric *Identity* values. In many cases however, the final keys of elements in the Target System are generated when data is delivered through the API. An important part of a delivery extension is to provide translation between the Identities created by migFx and the Keys assigned by the Target System.
Errors
Errors from the Target System API are unavoidable – especially in the first part of a migration project. As the Target Map matures with additional validation rules the occurrence of errors should decrease. It is the task of the delivery extension in collaboration with the API to handle possible partial deliveries, where some part of an object has been delivered successfully while other parts have failed As part of the Delivery extension interface, errors from the Target System API can be channelled back to migFx in order to be displayed as Events in the migFx Tracker application to be processed similarly to other errors of the migration.
Redelivery
The Delivery extension must handle the housekeeping in connection with redelivery of migrated data. migFx can notify if data has changed since last delivery. But it is the task of the Delivery extension to handle redelivery data that has already been handed off to the Target System in a previous version.

Migration Project

The migration project is outside the scope of any migFx Bridge and only included for completeness.

With the offering of a migFx Bridge handling the Target Map and all the extensions towards the Target System, the main scope of the migration project narrows down to the creation of the Source Map to deliver the data from the Source System in a format that conforms to the Target Interface exposed by the Target Map of the migFx Bridge.

The migFx Studio is used to create the Source Map and offers the usual benefits of a comprehensive application handling all the aspects of the mapping including validation, cross-referencing, collaborative support etc.

Further reading:

- migFx Concepts
- Interface reference