

FRAMEWORK BASED APPROACH TO MIGRATING IBM WMB BASED SOLUTIONS TO SOFTWARE AG WEBMETHODS



ABSTRACT

This document outlines the approach to migrating an existing WMB based ESB layer to the SAG platform. It highlights the benefits of using the UST developed Integration Development Framework tool to fast track migration from IBM WMB to Software AG webMethods.

Pankaj Saxena (UST, IND)

UST Global Inc,
September 2017



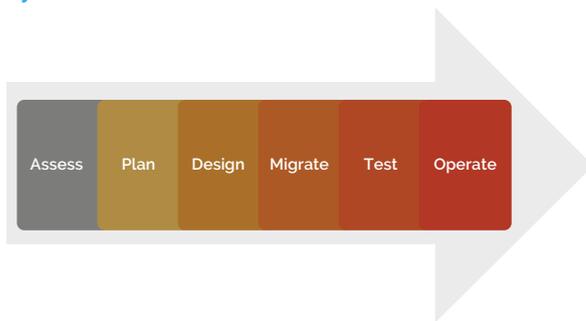
Table of Contents

● Introduction	3
● Conventional Approach to WMB to webMethods Migration	3
● UST Global Integration Development Framework (IDF) Based Migration	3
● Integrated Development Framework	4
● Components of the Framework	4
● Using the Framework	4
● Migrate Using IDF: Sample Use Case	5
● Summary	5

Introduction

The existing enterprise service bus is built on the IBM WebSphere Message Broker platform which is the contained middleware platform for clients. The platform ranks second as the system which influences most critical business processes. Since IBM WMB is coming to end of life and migrating to a higher version of IBM IIB would require substantial effort and time, there has been a strategic decision to move to Software AG's webMethods.

Given below is the representative migration life cycle:

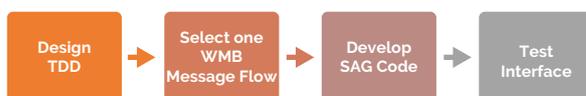


The migration engagement starts with the assessment of the existing WMB topology. It is an iterative process to identify and categorize the interfaces which qualify for migration. The criteria which could be considered are – business criticality, complexity, integration patterns and systems involved.

In the design phase, webMethods specific technical design document (TDD) has to be created for each interface to be migrated.

In the development phase, the TDD is implemented using the Software AG (SAG) product suite. In the testing phase, the migrated interfaces are tested against functional and non-functional requirements. It is also ensured that there is no change in business functionality due to migration.

Conventional Approach to WMB to webMethods Migration



Issues with the conventional approach are as follows:

- Migrates individual WMB message flows, resulting in:
 - Complexity and problems introduced over the lifetime of the flow will be migrated to the new environment
 - Ignores the best practices of webMethods (target) platform
 - No value addition to business
- All code components to be created manually using SAG Designer, resulting in:
 - Problems introduced by human errors; increases the cost of quality and cycle time.
 - Global Development teams will produce non-standard code; two interfaces which are exactly similar could result in very different implementations depending on the people.

UST Global Integration Development Framework (IDF) Based Migration

At UST, we recognize the need for standards and industry patterns based Integration. It is strongly recommended that the integration migration projects harvest the opportunity of migration to:

- Align the old integration interfaces to standard industry patterns for manageability
- Enforce Software Engineering standards across Global Development Teams consistently
- Support rapid business agility via reuse of services or interfaces

UST Global's Integrated Development Framework[®] (IDF) ensures that the above objectives are met. It is flexible enough to accommodate and extend any core integration frameworks defined by the enterprise. In cases where no best practices are defined, the framework is intelligent enough to pick the standard industry best practice.



IDF influences all phases of migration:

- **Assess/Plan/Design:** It ensures that all solutions are rethought to be implemented as a standard pattern.
- **Migrate:** IDF ensures that the skeleton code adhering to software engineering practices is automatically generated. It eliminates inconsistency of code in a global development team.
- **Test:** Reduces the human errors introduced during development.
- **Operate:** Consistency of interface implementation against particular patterns ensures that the Integration Application Support and Platform Support Teams have less number of unique problems to address.

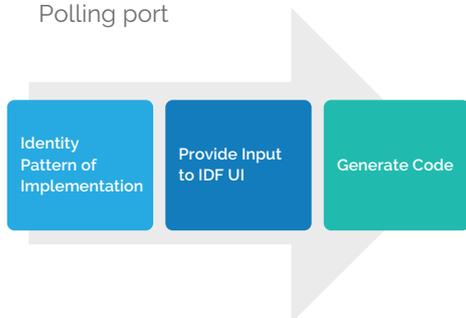
Integrated Development Framework

UST Global's Integrated Development Framework® (IDF) is a tool designed to enforce pattern based approach to development and drastically improve the cost of quality involved in SDLC phases.

The high level view of IDF usage is depicted in the following diagram.

Step 1: Identify the Pattern of Implementation
IDF supports the following patterns out of the box:

- Message Based Integration: Publish/Subscribe, Point to Point, Event using UM/JMS/Native Broker
- File Based Integration: Batch File using File Polling port



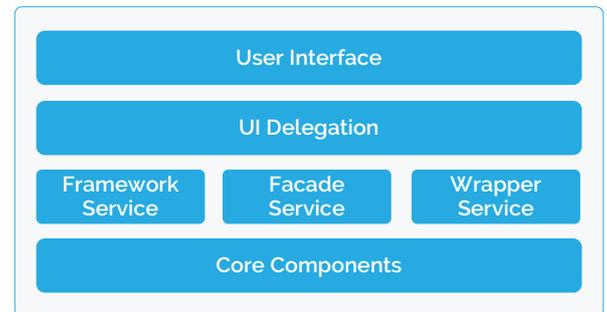
- Real-Time Integration: Request/Response using SOAP/REST protocols.

Step 2: Provide Input using IDF UI

Step 3: Click on 'Generate' to generate code

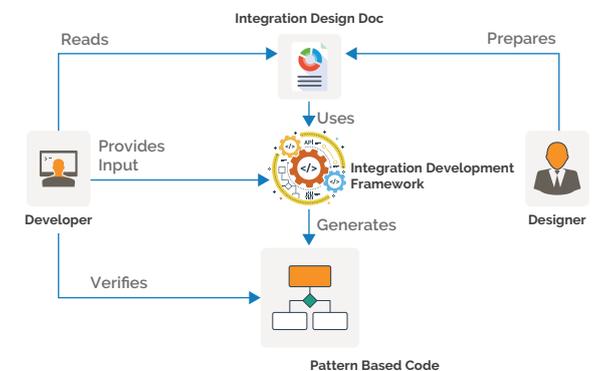
Components of the Framework

IDF has four logical layers as shown in the following diagram:



- **User Interface Layer:** SAG Designer or IDF provided UI can be used to provide inputs.
- **UI Delegation Layer:** Mediates between the UI layer and the service layer.
- **Service Layer:** Hosts functionality-specific services based on different patterns.
- **Core Component Layer:** Reusable component services orchestrated by the service layer.

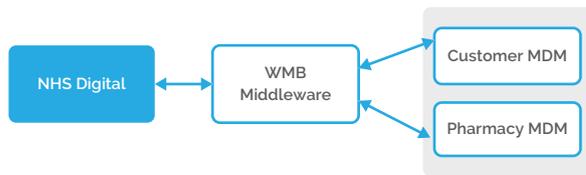
Using the Framework



- Developer needs to install two Integration Server (IS) packages which are bundled with IDF. If need, the developer can modify the configuration as per client development standards.
- Developer provides inputs received from the Design Doc for a given interface from the UI layer or via SAG Designer.
- IDF validates the input data against a given integration pattern.
- It generates the skeleton code for the selected integration pattern.
- It creates the service façade specific to the integration pattern.

Migrate Using IDF: Sample Use Case

To understand IDF better, let's take a fictitious use case of "Publishing Health Record" to National Health Scheme (NHS). Customer MDM (Master Data Management) stores customer records along with customer consent data which governs if the medical record (stored in Pharmacy MDM) can be published to external systems.



The above diagram shows the message flow among different systems. This interface is based on the request/response pattern. To migrate this interface to SAG platform, we need two integrations:

- 1) **Provider:** Expose customer service as web service to NHS (and any future system interested in this data).
- 2) **Consumer:** Fetch data from Customer MDM and Pharmacy MDM.

Let us migrate Provider integration using IDF and see how it impacts the following SDLC phases:

- **Design** – IDF mandates the designer to create the TDD in such a way that it clearly details out all technical details for a given interface. It ensures uniformity in the design phase itself. The following diagram shows a TDD snippet for the Provider integration explained in our use case.

Pattern Details

Protocol	SOAP
Direction	Provider
Messaging Type	Synchronous
WSDL Details	/media/sf_shared_vm/IDF/demo/customer_v10.wsdl
XSD Details	/media/sf_shared_vm/IDF/demo/customer_v30.xsd
Transport Protocol	HTTPS

SAG Components

Package Name	CoprWmPwhCerCenCustomers
Service Name	SearchCustomers
Service Namespace	corp.ini.wm.buk.cer.cen
Service Version	1.0
Mapping Service	NO
Validation Service	Yes

- **Development** – Developer provides the inputs from the TDD to IDF and clicks OK in the designer. The below diagram shows a sample output:



The IDF first creates the code skeleton wherein components are structured as per the selected integration pattern. It then creates a façade service containing a template for that integration pattern. Achieving this manually would take **30 mins** of development time for a skilled developer. IDF on the other hand does the same in **0.005 mins**. In addition, even a naïve SAG developer can complete this task using IDF. This point onwards, the developer can focus on core migration and mapping logic.

- **Testing** – IDF in conjunction with WmTestSuite ensures that unit test cases are written once and reused for all subsequent development.

Summary

Through qualitative analysis, it has been found that IDF provides **30% effort savings in the entire SDLC**. It enables the client to complete migration faster in a most cost efficient way, so that the impact of new technology can be experienced at the earliest possible chance, thereby meeting strategic goals.

ABOUT UST GLOBAL®

UST Global® is a fast-growing digital technology company that provides advanced computing and digital services to large private and public enterprises around the world. Driven by a larger purpose of Transforming Lives and the philosophy of "fewer Clients, more Attention", we bring in the entrepreneurial spirit that seeks the fastest path to value in today's digital economy. Our innovative technology services and pioneering social programs make us stand apart.

UST Global is headquartered in Aliso Viejo, California and operates in 21 countries. Our clients include Fortune 500 companies in Banking and Financial Services, Healthcare, Insurance, Retail, High Technology, Manufacturing, Shipping, and Telecom. UST Global believes in building long-lasting, strategic business relationships through agile and client-centric global engagement models that combines local experts and resources with cost, scale, and quality advantages of global operations.

For more information, please visit: www.ust-global.com



Corporate Office: UST Global®

5 Polaris Way, Second Floor, Aliso Viejo, CA 92656

Tel: (949) 716-8757 Fax: (949) 716-8396

www.ust-global.com

For further information contact: info@ust-global.com

