CRM Magic with Data Migration & Integration

Daniel Cai

http://www.kingswaysoft.com
http://danielcai.blogspot.com
About me

• Daniel Cai
  – Principal Developer @KingswaySoft
    • An independent software company offering integration software and solutions

• Main interests
  – Microsoft Dynamics CRM
  – Business Intelligence
  – .NET
  – Enterprise Architecture

Agenda

• Challenges

• Data Migration vs. Data Integration

• Data Migration / Integration Approaches
  – CRM Import Wizard
  – Custom Integration Development
  – ETL
  – Service Bus / BizTalk

• Get data into CRM faster!

• Closing notes
Challenges of Data Migration / Integration

• Diversity of data and systems

• Complexity and intricacy associated with working with CRM web service interfaces

• An often overlooked piece of work in CRM projects

• Limited space for performance tuning
# Data Integration vs. Data Migration

| Data Migration | A “One-Off” activity | • Often large volume of data in initial load  
|                |                     | • Cost to fix any data issues thereafter is high  
|                |                     | • Data cleansing is usually part of the effort  
|                |                     | • Time sensitive  
| Data Integration | On-going data synchronization or replication | • Managing incremental changes  
|                |                     | • Different requirements call for different design  
|                |                     | • Real-time  
|                |                     | • Batch  
|                |                     | • Messaging  
|                |                     | • Need to be done fast in order to minimize performance impact on production system  

Available options

• Leverage existing technologies and tools
  – CRM Import Data Wizard
  – ETL tools
    • SSIS
    • Informatica
    • Scribe
    • Connectors for Microsoft Dynamics
    • etc.
  – BizTalk / Service Bus

• Custom integration solution development
  – Program against CRM Web Service Interfaces (SDK Programming)
CRM Import Data Wizard
CRM Import Data Wizard

• What’s it?
  – Free utilities offered by the platform

• Pros
  – Works for simple and small data import scenarios
  – Works within application, available for CRM users for self-served data imports
  – Undoable
    • works well for new insert, but not update though
  – Free

• Cons
  – No delete
    • don’t confuse with the above undo capability
  – No transformation
  – No scheduling, no automation
  – Difficult to manage incremental changes
  – Maximum number of records is constrained by file size
  – Limited capability of handling relationship
  – Exceptions to be expected when used for some special entities, fields
CRM Import Data Wizard
Custom Integration Development
Custom Integration Development

• How does it work?
  – Write custom code against CRM web service interfaces using SDK or service references

• Pros
  – Leverage your .NET programming (C# or VB.NET) skills
  – More granular control
  – Flexible integration points
    • Plugins
    • Workflows / Processes
    • Standalone applications (Console, Windows Form, and probably third-party apps)

• Cons
  – Could be an exhaustive effort due to large amount of coding effort
    • Scheduling
    • Threading
    • Intricacies associated with working with CRM web service interfaces
  – Most likely much higher maintenance cost down the road
  – Often the case, the implementation ends up with a tightly-coupled architecture style, which leads to poor maintainability
Decisions – Custom Integration Development

• **SDK assemblies**
  – v6.0 can be used to talk to either CRM 2013 or 2011
  – v5.0 can also be used to talk to either CRM 2013 or 2011, however with no access to CRM 2013 specific features
    • entityimage field

• **Programming Styles**
  – Early-bound
  – Late-bound

• **Performance Considerations**
  – Multi-threading
  – Bulk Data Load API
### Programming Styles – Early-bound vs. Late-bound

<table>
<thead>
<tr>
<th>Pros</th>
<th>Early-bound</th>
<th>Late-bound</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Compile-time validation through strongly-typed entity classes and fields</td>
<td>• Slightly better performance comparing to early-bound</td>
<td></td>
</tr>
<tr>
<td>• Intellisense</td>
<td>• More flexibility</td>
<td></td>
</tr>
<tr>
<td>• CRM LINQ query APIs</td>
<td>• Smaller binary delivery</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Cons</th>
<th>Early-bound</th>
<th>Late-bound</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Small performance overhead</td>
<td>• No compile-time validation or intellisense</td>
<td></td>
</tr>
<tr>
<td>• Dependency on command-line tool whenever CRM metadata has been changed</td>
<td>• Less productive CRM LINQ query APIs</td>
<td></td>
</tr>
<tr>
<td>• Larger binary delivery</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

```csharp
var contact = new Contact
{
    FirstName = customer.FirstName,
    LastName = customer.LastName,
    Telephone1 = customer.Phone,
    EMailAddress1 = customer.Email,
};
xrmServiceContext.AddObject(contact);
xrmServiceContext.SaveChanges();
```

```csharp
var contact = new Entity("contact");
contact["firstname"] = customer.FirstName;
contact["lastname"] = customer.LastName;
contact["telephone1"] = customer.Phone;
contact["emailaddress1"] = customer.Email;
organizationServiceProxy.Create(contact);
```
Custom Integration Development

- Demo
  - Data integration through CRM Web Service Interfaces utilizing CRM SDK
ETL
ETL Tools

- What’s ETL
  - Extract, Transform, Load

- Pros
  - Development productivity
  - Integration automation with scheduling engine
  - Performance, scalability and extensibility
  - Can be part of your overall BI and data warehousing strategies

- Cons
  - Learning curve of the ETL tool
  - License cost of the ETL tool and/or the adapters
  - Probably not the ideal solution for real-time requirements

- Options
  - SSIS (SQL Server Integration Services)
  - Informatica
  - Scribe
  - Connectors for Microsoft Dynamics
  - ...
SSIS as CRM Data Integration Platform

• Why SSIS?
  – SSIS is Microsoft’s answer to enterprise data integration
  – Scalability
  – Performance
  – Extensibility
  – Easy to work with
  – Works for almost any data source or target
  – Accessible technical resources

• A third-party adapter is required
  – KingswaySoft SSIS Integration Toolkit for Microsoft Dynamics CRM
    • Free developer edition available at www.kingswaysoft.com
    • 4 SSIS components included
• Demo
  – Data integration using KingswaySoft SSIS Integration Toolkit for Microsoft Dynamics CRM
Connectors for Microsoft Dynamics

• A small footprint ETL engine

• Support the integration between Microsoft Dynamics CRM and most of Microsoft Dynamics ERP applications (AX, NAV, GP, SL)

• SDK is available to develop your own adapters

• Offered under free license since its v3.0 release
Connectors for Microsoft Dynamics (cont.)

Service Bus / BizTalk
Service Bus / BizTalk – Messaging-based integration

• What’s service bus?
  – A software architecture model used for designing and implementing the interaction and communication between mutually interacting software applications in service-oriented architecture (SOA)
  – A messaging-based integration solution

• Pros
  – Optimized to move single transactions between different systems or processes in near real time or real time
  – Decoupled integration architectural model
    • Best suited for decoupled heterogeneous systems by using Service Bus or BizTalk as the middleware

• Cons
  – Probably not best fit with large volume data load
  – Performance overhead due to serialization and deserialization
  – Complexity associated with the messaging platform
Service Bus Implementation Patterns

- Various Implementation Patterns (Azure Service Bus)
  - Queue
    - No active listener is required
    --destructive read vs. non-destructive read
    - two types of queues
      - message buffer queue
      - persistent queue
  - One Way
    - requires an active listener
    - retries through asynchronous system job
  - Two Way
    - requires an active listener
    - a string value can be returned
  - REST
    - essentially a two-way listener in REST style
  - Topic
    - Similar to a queue, except that listener(s) can subscribe to receive messages from the topic
Service Bus Integration for Microsoft Dynamics CRM

CRM Service Bus Integration

• Integration Points
  – Plugin
  – Workflow

• Some Limitations
  – There is no way to use custom messages
    • You publish the entire execution context, which could contain unnecessary information for other parties
    • Remove sensitive information from the context object if necessary
  – Although you can host Windows Service Bus on-premise, there isn’t a way to talk to your service bus on-premise from CRM plug-in or Workflow
  – It is possible to write listener to listen to service bus, and push the data (messages) to CRM server, however this is not currently an out-of-the-box feature provided by the platform
  – There will be some technical challenges if you want to utilize a third-party service bus solution
CRM + Azure Service Bus

• Demo
  – Service Bus integration in CRM Plugin
Get data into CRM faster!
Why?

• Web service interface is generally slow

• It is never faster enough to load data
Multi-threading

Make sure to update your .config file in order to overcome the connection limit of concurrent service calls.

```xml
<configuration>
  ...
  <system.net>
    <connectionManagement>
      <add address="*" maxconnection="100"/>
    </connectionManagement>
  </system.net>
</configuration>
```
Leverage multi-node cluster
Bulk API

- Bulk API is more beneficial when network latency is high
- There is an API throttling with **CRM Online**
  - At most 2 concurrent `ExecuteMultiple` requests per organization
Other considerations

• Minimize the number of fields you work with when reading from or writing to CRM

• Mind the performance impact that your plugins or workflows may have on the system
  – disable them if you don’t need them in initial load

• Watch out your query (if any) performance during data migration/integration
  – make sure database indexes are available for the queries
  – add custom indexes if needed

• Watch out the resource usage on your integration server, CRM server and database server
  – Poor performance could be caused by poor infrastructure
  – More often, poor performance can be caused by poorly-written custom code in plugins or workflows
Closing Notes
How to survive your CRM data migration/integration

• Don’t ever try to write to CRM database directly at any time!

• Choose the right tools, and know your tools
  – there is often more than one way to get a job done

• Know your data
  – both source and target
  – document and establish the mapping between your source data and target data

• Error happens
  – define a proper error handling strategy
  – and possibly implement a retry mechanism if necessary, particularly if you are working with CRM online

• Define your data quality standards
  – make sure your data migration/integration adhere to the standards

• Plan ahead, expect changes
Q & A

Daniel Cai

Personal Blog: http://danielcai.blogspot.com
Email: daniel.cai@kingswaysoft.com