|  |
| --- |
|  |
| Web Service Integration - CRM |
|  |
|  |
|  |
|  |

|  |
| --- |
| The Document represents and explains the process for Integration between any Client System with Dynamics CRM |

Contents

[Introduction 2](#_Toc487016989)

[Components utilized 2](#_Toc487016990)

[Process followed:- 2](#_Toc487016991)

[Functionalities in the Integration:- 3](#_Toc487016992)

[1. GetRecord 3](#_Toc487016993)

[2. CreateUpdateRecord 3](#_Toc487016994)

[3. CreateandUpdateRecord 3](#_Toc487016995)

[Why this approach has been undertaken!! 4](#_Toc487016996)

## Introduction

The document is all about the integration which has been followed to integrate a “Legacy System” to “CRM” system , to automate the process of data flow between both the systems.

The client has a legacy system which has been used for the years, and the data for “Business Opportunities” along with “Opportunity Products” is been stored in the system. The data in the system needs to be synced with CRM on real time scenario.

# Components utilized

* Schedulers ( C# executables)
* Web Services
* Databases

# Process followed:-

CRM

Web Service communicating with CRM



Database at Client Side

The process which has been followed is as follows :-

Database at CRM Side

Schedulers at Client Side

1. The Data from the Client’s Legacy system is pushed into a separate Database at Client’s end, on regular basis. The Data is specifically the one which needs to synced with Dynamics CRM, the data has information about the “Business Opportunities” along with the associated “Opportunity Products”.
2. The Schedulers at the Client’s side, are executed using the “Windows Scheduler” at every 3 Minutes, which pulls the data from the this database based on the value of the “Sync Flag”, and generates an XML , which has a standard pattern and is sent to the web service.
3. Once the request is received by the web service, the web service has been designed to undertake all the required validation checks like in , whether the Input XML is a valid one or not, whether the data which is needed for processing is in place or not.  
   All the required necessary checks has been undertaken in the web service, to ensure that the data which is been pushed to CRM meets up our requirement.
4. Once the request has been validated in the web service, and meets up our data validation the same is processed further to CRM for creation/updating/retrieval of the information from CRM.

# Functionalities in the Integration:-

1. GetRecord: -  
    The functionality of the “GETRECORD” InputXML is to extract the records from CRM and push the records to the common database at the client’s end. The records which needs to be extracted from CRM are decided based upon the flag, which needs to be marked by the user over the record in CRM. Once the record is marked to get synced, whenever the scheduler at client’s end makes a request to web service, the records are extracted from CRM and are sent in the OutputXML, and the flag is marked back to “Sent” so as to ensure that the same record is not sent twice, though in future if the user wants to sync the changes again they can mark the flag again for “syncing” again
2. CreateUpdateRecord :-  
   The records which has been extracted from CRM , during the request of the “GETOpportunty” now, needs to be created in the client system, and once the record has been created, a Record Number is generated at the client end, this number needs to be updated in the CRM, with the respective record.  
   The new request is generated by another scheduler, which has the Record Number ( CRM) along with the number which has been generated in the Client System, and the request is sent. Based upon the unique number which has been sent by the Scheduler, corresponding record is extracted from CRM and a subsequent request to update the “Client Records Number” is undertaken.
3. CreateandUpdateRecord :-  
   We have covered the scenario, where in the record has been created in CRM and has been synced and updated in the Client System. There is another scenario where in the record would be created in the client system, and now the record needs to be pushed in the CRM, by the scheduler.  
   Once the record has been created by the user in their own system, the relevant information is pushed by a job into the database connected with the scheduler. The scheduler executes to fetch this data and generate an INputxml , which is sent to the webservice for further processing.  
   The webservice has been coded in a way, so as to receive the request and check whether the information which is being received is in the specific manner or not, if not and exception is sent to the client, else the subsequent action of update is undertaken and then a response is shared with the client, which updates a flag in the client database stating that the opportunity has been synced and need not to be sent again !!

# Why this approach has been undertaken!!

The approach of sync records in bulk, through database has been undertaken instead of sync each and every record in CRM, on creation in the Client System so as to save the load and performance issues in CRM. If we would have taken an approach of undertaking the update for single record.. then it would have caused performance degradation. Along with the same, the client team need to understand and get themselves over CRM, the approach of exchange of data over XML has been undertaken to ensure.. that none of the team need to be exposed to the system, even the basic coding skills along with the basic understanding of XML have sufficed the integration process.  
  
Along with the same, the InputXML has a “OFFICE ID” tag which assists us, to decide using which credentials we need to invoke the organization service in CRM, as we have multiple countries and we have service accounts specific to each and every country, which have specific roles and permissions to ensure that the data to their specific country is only exposed, so is been pulled from CRM, to ensure the privacy of business is maintained.  
lets say we have a requirement to separate the data, for a specific branch we simply has to create a new user, that user has to be given specific role and permission to ensure that the access is just for the specific branch and have to assign a “Office ID” which would be used in the inputxml.