Salesforce to WorkflowMax App Integration

Technical Documentation

Version 1.2
www.wfmcrm.com

Table of Contents

Terminology

- API – Application Program Interface
- UAT – User Acceptance Testing
- WFMCRM – WorkflowMax Customer Relationship Manager
- Code – APEX

Salesforce Edition Requirements

- Package Name: WFMCRM
- Package Prefix: WFMCRM__
- Type: Managed Package
- Editions: Enterprise, Unlimited, Developer
- Languages: English Only
- Custom Objects: 1
- Custom Settings: 1
- Custom Tabs: 1
- Custom Applications: 1

Account Object

Overview:

The Account record is the required parent of the contact record.
1 Account can have many Contact records.
1 Account can have many Opportunity records.

Terminology:

Salesforce: Account
WorkflowMax: Client

Object Custom Fields Used (subject to change):

1. WFMCRM__Included_In_Next_Sync__c (Formula Boolean)
2. WFMCRM__KeyFieldDateLastModified__c (Date Time)
3. WFMCRM__Sync_to_WorkflowMax__c (Boolean)
4. WFMCRM__WFMClientId_Historic__c (Text 255char)
5. WFMCRM__WFM_DateLastModified__c (Date/Time)
6. WFMCRM__WFMId (External Id, Text 255char)
7. WFMCRM__WFMId_Historic__c (Text 255char)

Object Standard Fields Used (subject to change):

Salesforce Field	WorkflowMax Field
Name	Name
WebSite	WebSite
Phone	Phone
Fax	Fax
BillingStreet	Address
BillingCity	City
BillingState	Region
BillingPostalCode	PostCode
BillingCountry	Country
ShippingStreet	PostalAddress
ShippingCity	PostalCity
ShippingState	PostalRegion
ShippingPostalCode	PostalPostCode
ShippingCountry	PostalCountry

Sync Rules:

WFMCRM__Sync_to_WorkflowMax__c and
WFMCRM__Included_In_Next_Sync__c = true

This is formulated as below

```
IF(
(WFMCRM__Sync_to_WorkflowMax__c = TRUE &&
ISNULL(WFMCRM__WFM_DateLastModified__c)) ||
(WFMCRM__Sync_to_WorkflowMax__c = TRUE &&
WFMCRM__KeyFieldDateLastModified__c >
WFMCRM__WFM_DateLastModified__c )
,
TRUE,
FALSE)
```

The field WFMCRM__Sync_to_WorkflowMax__c being checked or unchecked on record creation will be determines by the Settings for that object under the WFMCRM Settings page.

The represented date fields are programmatically updated when key fields are updated. Key fields are as follows

WFMCRM__Sync_to_WorkflowMax__c
Name
WebSite
Phone
Fax
BillingStreet
BillingCity
BillingState
BillingPostalCode
BillingCountry
ShippingStreet
ShippingCity
ShippingState
ShippingPostalCode
ShippingCountry

After the sync has run the field WFMCRM__KeyFieldDateLastModified__c is updated to the value of DateTime.now(), removing it from another sync until the fields are updated again. Once the record is in the queue to be synced, a check is done to see if an Id value already exists in the WFMId__c field.

If an Id value exists (previously has been synced), this is the Workflow Max record that will be updated with all the above fields.

If an Id value does not exist in the field, then a new record is created in Workflow Max using all the above fields.

If the Id field value is removed from the WFMId__c field after a sync has occurred, it will be considered a "new" record in the sync process

and create a new record in Workflow Max. Removing just the WFMId field value will not trigger a sync on its own, another field value would need to be changed to trigger a sync. This can be a cause of duplicates being seen in Workflow Max.

If the Id field value is changed in the WFMId__c field after a sync has occurred, the update will occur to the newly updated record Id in Workflow Max.

Fields are mapped as follows via the API (Create).

```
+ '<Name>'+strName+'</Name>'
+ '<Address>'+strBillingStreet+'
+ '<City>'+strBillingCity+'</City>'
+ '<Region>'+strBillingState+'</Region>'
+ '<PostCode>'+strBillingPostalCode+'</PostCode>'
+ '<Country>'+strBillingCountry+'</Country>'
+ '<PostalAddress>'+strShippingStreet+'</PostalAddress>'
+ '<PostalCity>'+strShippingCity+'</PostalCity>'
+ '<PostalRegion>'+strShippingState+'</PostalRegion>'
+ '<PostalPostCode>'+strShippingPostalCode+'</PostalPostCode>'
+ '<PostalCountry>'+strShippingCountry+'</PostalCountry>'
+ '<Phone>'+strPhone+'</Phone>'
+ '<Fax>'+strFax+'</Fax>'
+ '<WebSite>'+strWebSite+'</WebSite>'
```

Fields are mapped as follows via the API (Update).

```
+ '<Name>'+strName+'</Name>'
+ '<Address>'+strBillingStreet+'
+ '<City>'+strBillingCity+'</City>'
+ '<Region>'+strBillingState+'</Region>'
+ '<PostCode>'+strBillingPostalCode+'</PostCode>'
+ '<Country>'+strBillingCountry+'</Country>'
+ '<PostalAddress>'+strShippingStreet+'</PostalAddress>'
+ '<PostalCity>'+strShippingCity+'</PostalCity>'
+ '<PostalRegion>'+strShippingState+'</PostalRegion>'
+ '<PostalPostCode>'+strShippingPostalCode+'</PostalPostCode>'
+ '<PostalCountry>'+strShippingCountry+'</PostalCountry>'
+ '<Phone>'+strPhone+'</Phone>'
+ '<Fax>'+strFax+'</Fax>'
+ '<WebSite>'+strWebSite+'</WebSite>'
```

Contact Object

Overview:

The Contact record is the non-required child of the Account record.
Contact requires an Account to save.
1 Contact can have 1 Account record.
1 Contact can have many Opportunity records.

Terminology:

Salesforce: Contact
WorkflowMax: Contact

Object Custom Fields Used (subject to change):

1. WFMCRM__Account_WFMId__c (Formula Text)
2. WFMCRM__Included_In_Next_Sync__c (Formula Boolean)
3. WFMCRM__KeyFieldDateLastModified__c (Date/Time)
4. *WFMCRM__Sync_to_WorkflowMax__c (Boolean)
5. WFMCRM__WFMContactsId_Historic__c (Text 255char)
6. WFMCRM__WFM_DateLastModified__c (Date/Time)
7. WFMCRM_WFMId (External Id, Text 255char)
8. WFMCRM__WFMId_Historic__c (Text 255char)

*Account must have WFMCRM__Sync_to_WorkflowMax__c checked true

Object Standard Fields Used (subject to change):

Salesforce Field	WorkflowMax Field
Salutation	Salutation
FirstName + LastName	Name
Email	Email
Title	Position
MobilePhone	Mobile
Phone	Phone

Sync Rules:

WFMCRM__Sync_to_WorkflowMax__c and
WFMCRM__Included_In_Next_Sync__c = true

This is formulated as below

```
IF(
(WFMCRM__Sync_to_WorkflowMax__c = TRUE &&
ISNULL(WFMCRM__WFM_DateLastModified__c) &&
NOT(ISNULL(WFMCRM__Account_WFMId__c))) ||
```

(WFMCRM__Sync_to_WorkflowMax__c = TRUE &&
WFMCRM__KeyFieldDateLastModified__c >
WFMCRM__WFM_DateLastModified__c &&
NOT(ISNULL(WFMCRM__Account_WFMId__c)))
,
TRUE,
FALSE)

A contact must have a related Account already synced to Workflow Max. A validation error will display if this had not occurred.

The field WFMCRM__Sync_to_WorkflowMax__c being checked or unchecked on record creation will be determines by the Settings for that object under the WFMCRM Settings page.

The represented date fields are programmatically updated when key fields are updated. Key fields are as follows

WFMCRM__Sync_to_WorkflowMax__c
Name
Email
Phone
Salutation
MobilePhone
Title

After the sync has run the field WFMCRM__KeyFieldDateLastModified__c is updated to the value of DateTime.now(), removing it from another sync until the fields are updated again. Once the record is in the queue to be synced, a check is done to see if an Id value already exists in the WFMId__c field.

If an Id value exists (previously has been synced), this is the Workflow Max record that will be updated with all the above fields.

If an Id value does not exist in the field, then a new record is created in Workflow Max using all the above fields.

If the Id field value is removed from the WFMId__c field after a sync has occurred, it will be considered a "new" record in the sync process and create a new record in Workflow Max. Removing just the WFMId field value will not trigger a sync on its own, another field value would need to be changed to trigger a sync. This can be a cause of duplicates being seen in Workflow Max.

If the Id field value is changed in the WFMId__c field after a sync has occurred, the update will occur to the newly updated record Id in Workflow Max.

If the WFMCRM__Account_WFMId__c (cross object formula field) is changed after the update (change on the related Account), the contact will not be updated to the new Account. At time of writing, current WorkflowMax API does not support updating a contact to a different Account Id after its created. See https://www.workflowmax.com/api/client-methods#PUT contact/[id]

Fields are mapped as follows via the API (Create).

```
+ '<Client>'
+ '<ID>'+CONREC.get('WFMCRM__Account_WFMId__c')+'</ID>'
+ '</Client>'
+ '<Name>'+strName+'</Name>'
+ '<IsPrimary></IsPrimary>'
+ '<Salutation>'+strSalutation+'</Salutation>'
+ '<Addressee></Addressee>'
+ '<Mobile>'+strMobilePhone+'</Mobile>'
+ '<Email>'+strEmail+'</Email>'
+ '<Phone>'+strPhone+'</Phone>'
+ '<Position>'+strTitle+'</Position>'
```

Fields are mapped as follows via the API (Update).

```
+ '<Name>'+strName+'</Name>'
+ '<IsPrimary></IsPrimary>'
+ '<Salutation>'+strSalutation+'</Salutation>'
+ '<Addressee></Addressee>'
+ '<Mobile>'+strMobilePhone+'</Mobile>'
+ '<Email>'+strEmail+'</Email>'
 + '<Phone>'+strPhone+'</Phone>'
+ '<Position>'+strTitle+'</Position>'
```

Opportunity Object

Overview:

Opportunity requires an Account to save.
A Contact is not required to save an Opportunity.
1 Opportunity can have 1 Account record.
1 Opportunity can have 1 Contact record.

Terminology:

Salesforce: Opportunity
WorkflowMax: Job

Object Custom Fields Used (subject to change):

1. *WFMCRM__Sync_to_WorkflowMax__c (Boolean)
2. WFMCRM__Account_WFMId__c (Formula Text)
3. WFMCRM__Contact_WFMId__c (Formula Text)
4. WFMCRM__Due_Date__c (Date)
5. WFMCRM__KeyFieldDateLastModified__c (Date/Time)
6. WFMCRM__Start_Date__c (Date)
7. WFMCRM__WFMSettingStatusAllowed__c (Checkbox)
8. WFMCRM__WorkflowMax_Job_No__c
9. WFMCRM_WFMId (External Id, Text 255char)
10. **WFMCRM_Contact__c (Lookup)
11. WFMCRM__WFM_DateLastModified__c (Date/Time)
12. WFMCRM__Included_In_Next_Sync__c (Formula Boolean)

*Account (& Contact if selected) must have WFMCRM__Sync_to_WorkflowMax__c checked true or cannot save record.

** Filtered. Can only be existing contact on related Account. Not required. If null, then no value passed to WorkflowMax record.

Object Standard Fields Used (subject to change):

Salesforce Field	WorkflowMax Field
Name	Name
Description	Description
WFMCRM__Job No__c	ID <Job><ID></ID></Job>
WFMCRM__Contact__c	ContactID
CreatedDate	StartDate
WFMCRM__Due_Date__c	DueDate
Amount	Budget
WFMCRM__Start_Date__c	StartDate

Sync Rules:

WFMCRM__Sync_to_WorkflowMax__c and
WFMCRM__Included_In_Next_Sync__c = true

This is formulated as below

```
IF(
(WFMCRM__WFMSettingStatusAllowed__c = TRUE &&
WFMCRM__Sync_to_WorkflowMax__c = TRUE &&
ISNULL(WFMCRM__WFM_DateLastModified__c) &&
NOT(ISNULL(WFMCRM__Account_WFMId__c))) ||
(WFMCRM__WFMSettingStatusAllowed__c = TRUE &&
WFMCRM__Sync_to_WorkflowMax__c = TRUE &&
WFMCRM__KeyFieldDateLastModified__c >
WFMCRM__WFM_DateLastModified__c &&
NOT(ISNULL(WFMCRM__Account_WFMId__c)))
,
TRUE,
FALSE)
```

An opportunity must have a related Account already synced to Workflow Max. A validation error will display if this had not occurred.

The field WFMCRM__Sync_to_WorkflowMax__c being checked or unchecked on record creation will be determines by the Settings for that object under the WFMCRM Settings page.

The represented date fields are programmatically updated when key fields are updated. Key fields are as follows

WFMCRM__Sync_to_WorkflowMax__c
Name
Description
Amount
WFMCRM__Start_Date__c
WFMCRM__Due_Date__c
WFMCRM__Contact_WFMId__c

After the sync has run the field WFMCRM__KeyFieldDateLastModified__c is updated to the value of DateTime.now(), removing it from another sync until the fields are updated again. Once the record is in the queue to be synced, a check is done to see if an Id value already exists in the WFMId__c field.

If an Id value exists (previously has been synced), this is the Workflow Max record that will be updated with all the above fields.

If an Id value does not exist in the field, then a new record is created in Workflow Max using all the above fields.

If the Id field value is removed from the WFMId__c field after a sync has occurred, it will be considered a "new" record in the sync process and create a new record in Workflow Max. Removing just the WFMId field value will not trigger a sync on its own, another field value would need to be changed to trigger a sync. This can be a cause of duplicates being seen in Workflow Max.

If the Id field value is changed in the WFMId__c field after a sync has occurred, the update will occur to the newly updated record Id in Workflow Max.

If the WFMCRM__Account_WFMId__c (cross object formula field) is changed after the update (change on the related Account), the opportunity will not be updated to the new Account. At time of writing, current WorkflowMax API does support updating an opportunity to a different Account Id however this has not been implemented into WFMCRM. See https://www.workflowmax.com/api/job-methods#PUT update

Fields are mapped as follows via the API (Create).

```
+ '<Name>'+strName+'</Name>'
+ '<Description>'+strDescription+'</Description>'
+
'<ClientID>'+OPPREC.get('WFMCRM__Account_WFMId__c')+'</Clie
ntID>'
+ '<ContactID>'+strWFMCRMContactWFMId+'</ContactID>'
+ '<StartDate>'+strStartDate+'</StartDate>'
+ '<DueDate>'+strCloseDate+'</DueDate>'
+ '<State>Planned</State>'
+ '<Budget>'+OPPREC.get('Amount')+'</Budget>'
```

Fields are mapped as follows via the API (Update).

```
+
'<ID>'+OPPREC.get('WFMCRM__WorkflowMax_Job_No__c')+'</ID>'
+ '<Name>'+strName+'</Name>'
+ '<Description>'+strDescription+'</Description>'
+ '<StartDate>'+strStartDate+'</StartDate>'
+ '<DueDate>'+strCloseDate+'</DueDate>'
```

WFMCRM__Log_File__c Object

Overview:

Write log file record on each successful APEX batch job completion of key data counts. Email admin address with failure and reason.

Terminology:

Salesforce: Log File
WorkflowMax: N/A

Object Custom Fields Used (subject to change):

Salesforce Field	Field Value / Purpose
WFMCRM__API_Response__c	Name
WFMCRM__Batch__c	Async Id
WFMCRM__Created_Date_Time__c	Stringof CreatedDate
WFMCRM__Job_Items_Processed__c	JobItemsProcessed from AsyncApexJob
WFMCRM__Number_of_Errors__c	NumberofErrors from AsyncApexJob
WFMCRM__Object__c	Object Name
WFMCRM__Status__c	Failed,Completed,Aborted,Queued
WFMCRM__AccountObjSyncDefault__c	Default setting for Before Insert
WFMCRM__ContactObjSyncDefault__c	Default setting for Before Insert
WFMCRM__OpportunityObjSyncDefault__c	Default setting for Before Insert

Custom Setting (WorkflowMaxAPI) Object

Overview:

Custom setting object to hold key API credentials and setting data.

Terminology:

Salesforce: WorkflowMaxAPI
WorkflowMax: N/A

Object Custom Fields Used (subject to change):

Salesforce Field	Field Value / Purpose
WFMCRM__AccountKey__c	Store WorkflowMax API Key
WFMCRM__AccountObjSyncDefault__c	Store Default Check
WFMCRM__APIKey__c	Store WorkflowMax Account Key
WFMCRM__ContactObjSyncDefault__c	Store Default Check
WFMCRM__Create_Log_Files__c	Store IF User Created Log Files
WFMCRM__Dataflow__c	Store Dataflow Rule
WFMCRM__OpportunityObjSyncDefault__c	Store Default Check
WFMCRM__OpportunityPostAddState__c	Store Default Job Post State
WFMCRM__OpportunityStatusSyncAll__c	Store Default Check
WFMCRM__OpportunityStatusSyncAllowed__c	Store Allowed StageNames

- Option to have data push from Salesforce (Salesforce master data) objects
- Option to have data pull to Salesforce (WorkFlowMax master data) objects
- Option to have data check to run rules on which record is Master Data (LastModifiedDate)

Visualforce Pages

The following pages have been identified as required Visualforce pages. In most cases, Visualforce pages require their own extension or controller.

1. /Settings

The main controlling page with tabs where users configure their API settings and access other functionality. The settings page is for GUI access to the Custom Setting object record that hold secure configuration API data and settings. Only can be one Custom Setting record in database. Validation in place. Record created on first trigger of page open to check if exists.

Tab	Version Release
API Configuration	1.0

Overview

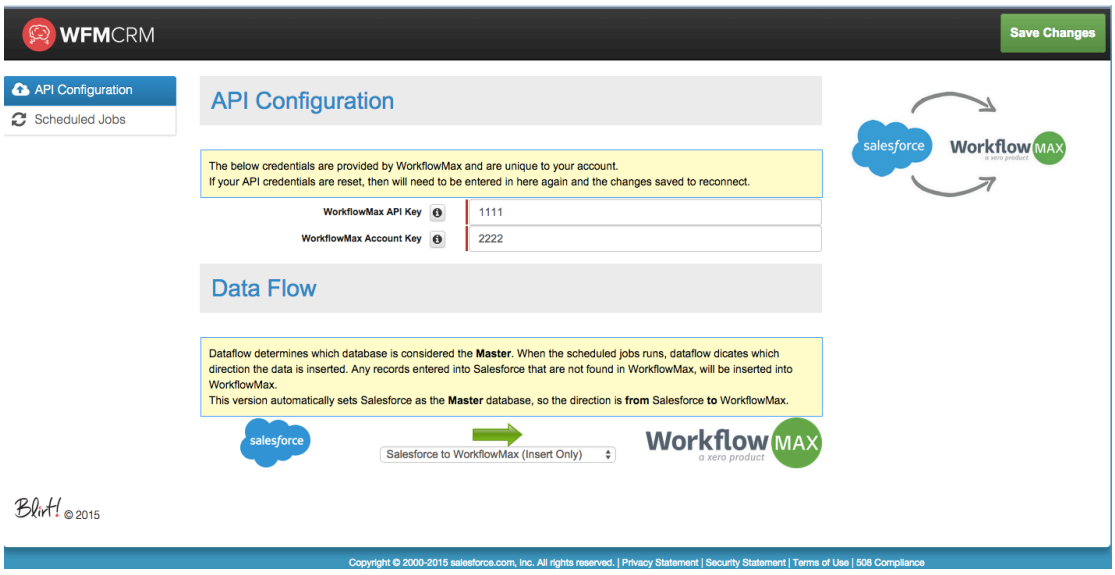
Display static text and images

Dashboard

Display key metrics of Log Files

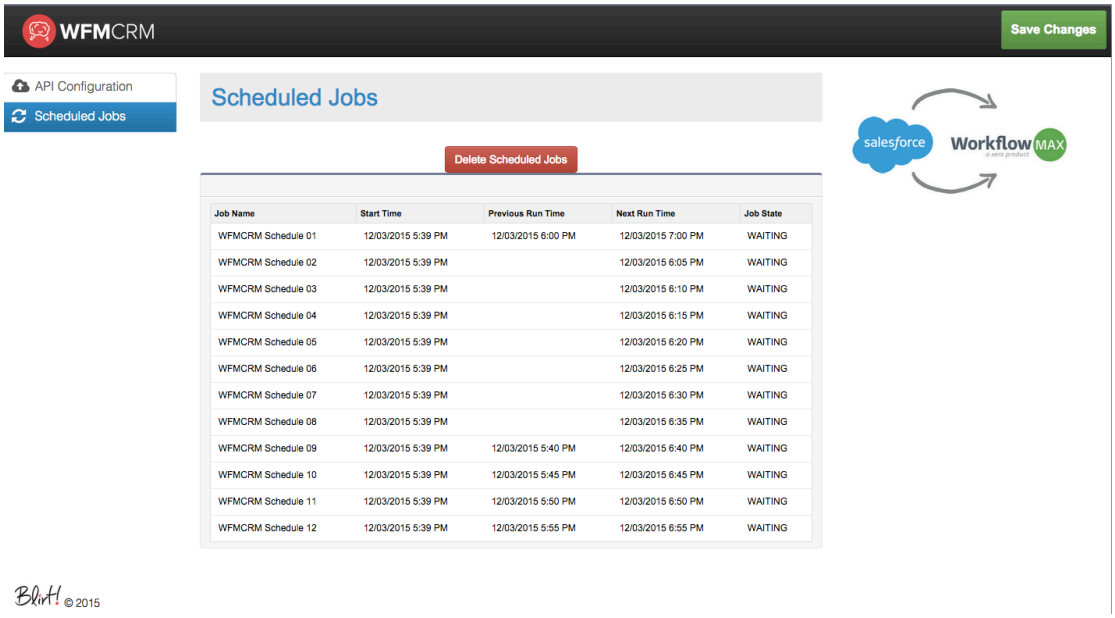
API Configuration

User can enter WorkflowMax API credentials and save. User must have these credentials or a save will not be possible.



Scheduled Jobs

Displays SOQL list of APEX Batch Jobs. User can disable (delete) and enable (mass insert) job records.



Data Import (via XML)

At time of documentation (Feb 2017), WorkflowMax reporting is unable to expose all necessary fields via the native reporting solution to export into csv format.

To export all your data successfully, you will require your WorkflowMax API credentials (API Key and Account Key) and some minimal expertise Excel formatting and formulas.

Install the WFMCRM from AppExchange here

A help reference can be found here
https://www.workflowmax.com/api/overview
https://www.workflowmax.com/api/client-methods
https://www.workflowmax.com/api/job-methods

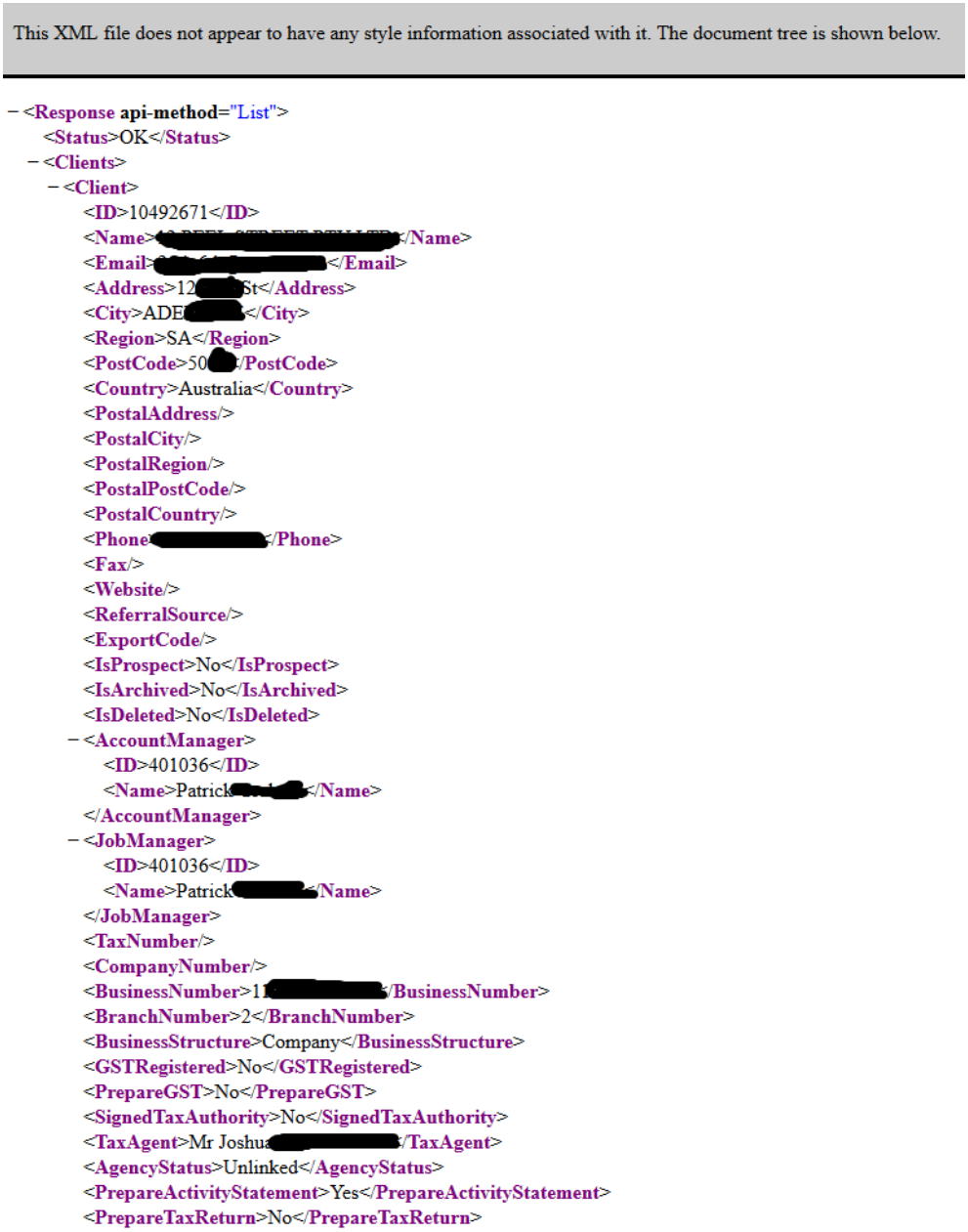
To import all your data successfully, you will require your Salesforce API credentials (username and security token) and some minimal expertise Apex dataloader.

1. Download your Client List and related objects data via a URL string to XML.
2. Recommended Firefox as a browser.

3. Get all your Client data and related Contacts via this string. Replace the variables with your own API Key and Account Key.

Sample: https://api.workflowmax.com/client.api/list?apiKey=[your API key]&accountKey=[WorkflowMax account key]

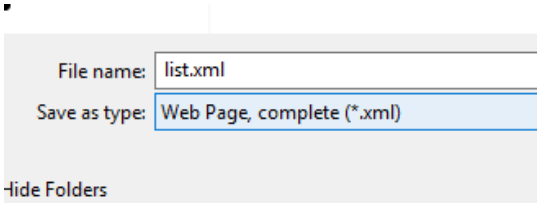
4. If successful, you will see a page like this, holding all your client data as a returned XML file.



This XML file does not appear to have any style information associated with it. The document tree is shown below.

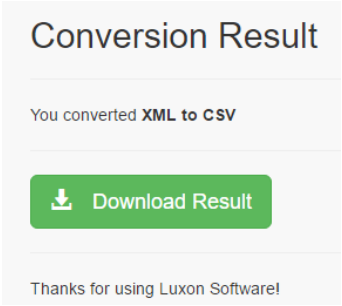
```
− <Response api-method="List">
    <Status>OK</Status>
  − <Clients>
    − <Client>
        <ID>10492671</ID>
        <Name>[redacted]</Name>
        <Email>[redacted]</Email>
        <Address>12[redacted]St</Address>
        <City>ADE[redacted]</City>
        <Region>SA</Region>
        <PostCode>50[redacted]</PostCode>
        <Country>Australia</Country>
        <PostalAddress/>
        <PostalCity/>
        <PostalRegion/>
        <PostalPostCode/>
        <PostalCountry/>
        <Phone[redacted]</Phone>
        <Fax/>
        <Website/>
        <ReferralSource/>
        <ExportCode/>
        <IsProspect>No</IsProspect>
        <IsArchived>No</IsArchived>
        <IsDeleted>No</IsDeleted>
      − <AccountManager>
          <ID>401036</ID>
          <Name>Patrick[redacted]</Name>
        </AccountManager>
      − <JobManager>
          <ID>401036</ID>
          <Name>Patrick[redacted]</Name>
        </JobManager>
        <TaxNumber/>
        <CompanyNumber/>
        <BusinessNumber>11[redacted]</BusinessNumber>
        <BranchNumber>2</BranchNumber>
        <BusinessStructure>Company</BusinessStructure>
        <GSTRegistered>No</GSTRegistered>
        <PrepareGST>No</PrepareGST>
        <SignedTaxAuthority>No</SignedTaxAuthority>
        <TaxAgent>Mr Joshua[redacted]</TaxAgent>
        <AgencyStatus>Unlinked</AgencyStatus>
        <PrepareActivityStatement>Yes</PrepareActivityStatement>
        <PrepareTaxReturn>No</PrepareTaxReturn>
```

5. Right click and “Save Page As” an XML file to a selected folder

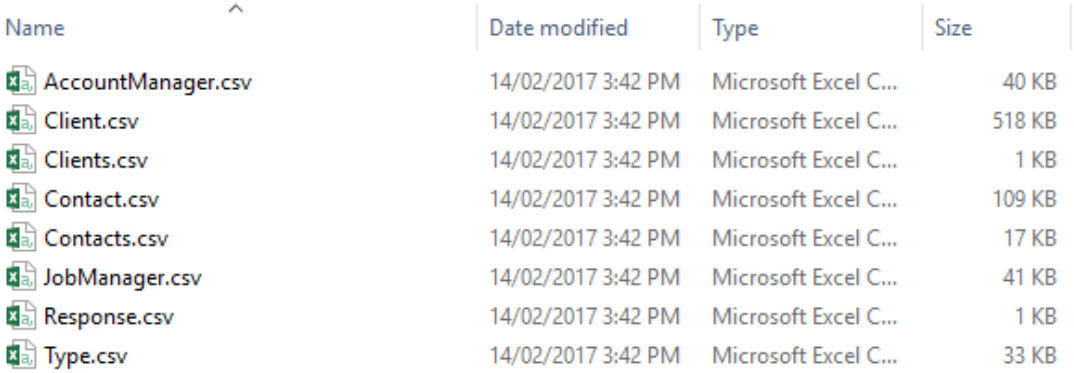


6. Open a XML to CSV convertor. Recommended Page: http://www.luxonsoftware.com/converter/xmltocsv

7. Upload your XML file via this website and download your .csv file package.

8. 

9. You should receive some data files like the below.



Name	Date modified	Type	Size
AccountManager.csv	14/02/2017 3:42 PM	Microsoft Excel C...	40 KB
Client.csv	14/02/2017 3:42 PM	Microsoft Excel C...	518 KB
Clients.csv	14/02/2017 3:42 PM	Microsoft Excel C...	1 KB
Contact.csv	14/02/2017 3:42 PM	Microsoft Excel C...	109 KB
Contacts.csv	14/02/2017 3:42 PM	Microsoft Excel C...	17 KB
JobManager.csv	14/02/2017 3:42 PM	Microsoft Excel C...	41 KB
Response.csv	14/02/2017 3:42 PM	Microsoft Excel C...	1 KB
Type.csv	14/02/2017 3:42 PM	Microsoft Excel C...	33 KB

10. Client.csv is considered Accounts in Salesforce. This object has unique ID field, and a Client_Id field which is required for formulate the future relationship correctly.

11. Contact.csv is considered your Contacts in Salesforce. This object has unique ID field, and a Contacts_Id field which is required for formulate the future relationship correctly.

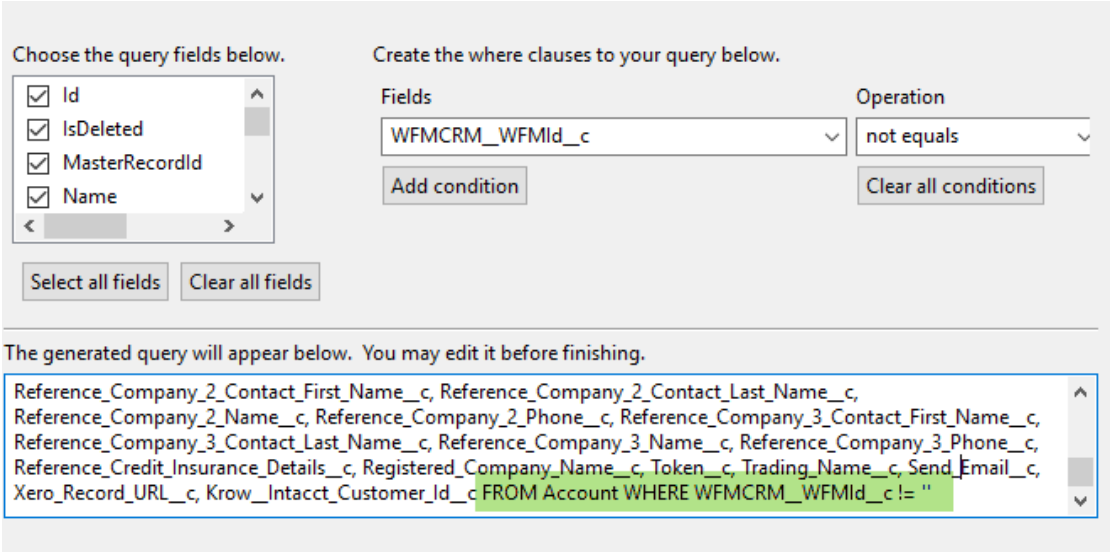
12. Contacts.csv is considered the relationship to map between Accounts and Contacts using the Client_Id and Contacts_Id fields.

13. Map and import your Client.csv selected data to the Account object in Salesforce using the APEX dataloader. Be sure to map

the Client_Id field to the Salesforce field WFMCRM__WFMClientId_Historic__c and the ID field to the Salesforce field WFMCRM__WFMId__c, as both will be required later for a relational VLOOKUP in Excel. Also the field WFMCRM__WFMId_Historic__c is available if you want to duplicate the ID field and store it statically for later reference should the ID change once the sync in functioning. Map respectfully all the other fields 1:1 you require.

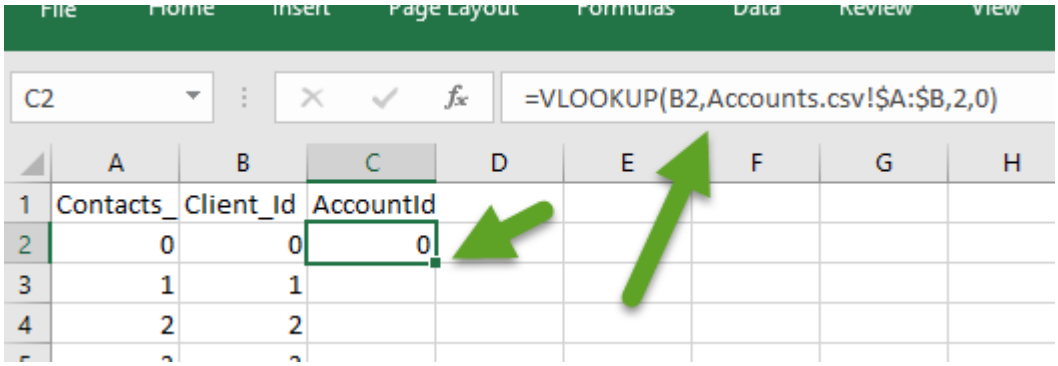
14. Delate, repeat and rinse until you receive no errors from the Salesforce import.

15. After your successful import, export all the Account records (name if Accounts.csv) via Apex Dataloader and all fields and add a filer of : ***FROM Account WHERE WFMCRM__WFMId__c != ''***. This will only get the records which have a WorkflowMax Id from the import.



16. Open the Contacts.csv and create a column called AccountId next to Client_Id. Open Accounts.csv files and VLOOKUP the Salesforce Account Id via the WFMCRM__WFMClientId_Historic__c so you get the Salesforce Account Id onto your Contacts.csv file. Save the values to the file so they are no longer formulated.

17. Open the Contact.csv and VLOOKUP the Salesforce Account Id via the Client_Id from the Contacts.csv. You will need to use a VLOOKUP formula similar to ***=VLOOKUP(B2,Accounts.csv!$A:$B,2,0)*** and click the bottom right box square to iterate over all the rows and mass collect the data.

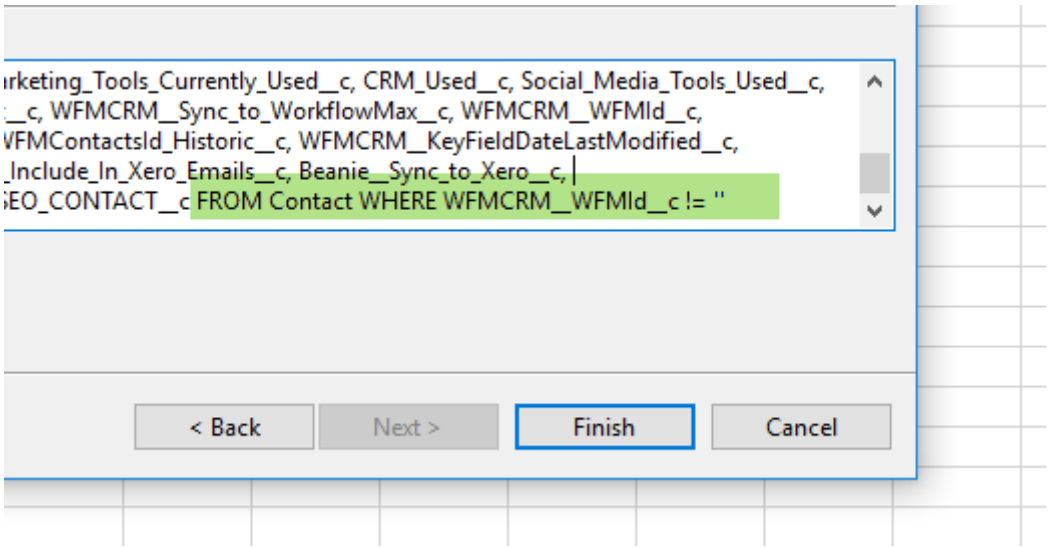


18. Save your file with the non-formulated values.

19. Open Contact.csv and VLOOKUP via Contacts_Id the AccountId from the Contacts.csv file. This will give you the Salesforce AccountId to relate your inserts of contact records.

20. Once you have your Contact.csv records related without #N/A values, import the contacts via Apex Dataloader.

21. After your successful import, export all the Contact records (name if Conacts.csv) via Apex Dataloader and all fields and add a filer of : *FROM Contact WHERE WFMCRM__WFMId__c !=* ''. This will only get the records which have a WorkflowMax Id from the import.



22. Download your Job (Opportunity) data via XML.

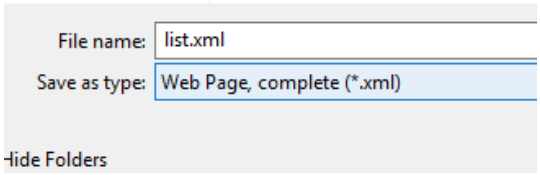
23. Sample: https://api.workflowmax.com/job.api/list?apiKey=[your API key]&accountKey=[WorkflowMax account key]&from=20010801&to=20200901

24. A helpful link is here https://www.workflowmax.com/api/job-methods

25. If successful, you will see a page like this, holding all your job data as a returned XML file.

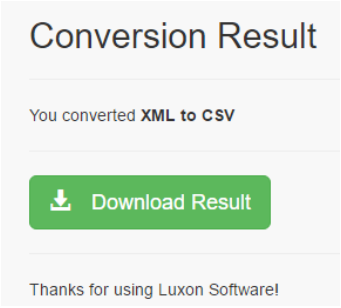
<Response api-method="List">
<Status>OK</Status>
<Jobs>
<Job>
<ID>J000128</ID>
<Name>Jumper</Name>
<Description>Jumper</Description>
<Client>
<ID>7310557</ID>
<Name>Yes 1111</Name>
</Client>
<ClientOrderNumber/>
<State>Planned</State>
<StartDate>2015-10-12T00:00:00</StartDate>
<DueDate>2015-11-12T00:00:00</DueDate>
<Contact>
<ID>4940349</ID>
<Name>Yes 222</Name>
</Contact>
<InternalID>9478096</InternalID>
<Assigned/>
</Job>
<Job>

26. Right click and “Save Page As” an XML file to a selected folder

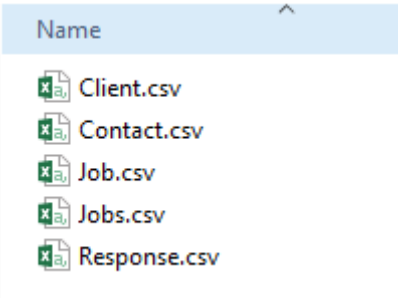


27. Open a XML to CSV convertor. Recommended Page: http://www.luxonsoftware.com/converter/xmltocsv

28. Upload your XML file via this website and download your .csv file package.

29. 

30. You should receive some data files like the below.



31. Client.csv is considered Accounts in Salesforce. This object is used to relate Job.csv to the correct Account in Salesforce via Job_Id.

32. Contact.csv is considered your Contacts in Salesforce. This object is used to relate Job.csv to the correct Contact in Salesforce via Job_Id.

33. Job.csv is considered Opportunities in Salesforce. Accounts and Contacts are mapped via the Job_Id field and additional VLOOKUP.

34. Open Account.csv (from your previous export).

35. Open and create a column called AccountId next to Job_Id field. VLOOKUP the ID field to the WFMCRM__WFMId_Historic__c field to obtain your Salesforce Account Id into the Client.csv file. Save the file.

36. Open Job.csv and create a column called AccountId next to Job_Id field. VLOOKUP the AccountId field to the AccountId field on Client.csv to obtain your Salesforce Account Id into the Jobs.csv file. Save the file.

37. Open Contacts.csv (from your previous export) and create a column called ContactId next to Job_Id field. VLOOKUP the ID field to the WFMCRM__WFMId_Historic__c field to obtain your Salesforce Contact Id into the Client.csv file. Save the file.

38. Open Job.csv and create a column called WFMCRM__Contact__c next to Job_Id field. VLOOKUP the WFMCRM__Contact__c field to the ContactId field on Client.csv to obtain your Salesforce Account Id into the Jobs.csv file. Save the file.

39. Map and import your Job.csv selected data to the Opportunity object in Salesforce using the APEX dataloader. Be sure to map the ID field to the Salesforce field WFMCRM__WFMId__c. Map respectfully all the other fields 1:1 you require.

40. Delate, repeat and rinse until you receive no errors from the Salesforce import.

41. Once completed, activate the sync process on new records.