UNIVERSITY
OF MIAMI

Interactions for Student Recruitment

TECHNICAL IMPLEMENTATION GUIDE

Information

Interactions for Student Recruitment is a solution for recruiting top prospects to your college or university.

Built on the Salesforce platform, Interactions allows you to track prospective students through all stages of the application process. Interactions also ties recruitment information to Contact and Lead data, simplifying application pipeline management so you can focus recruiting efforts on the most promising prospects.

Developed by the University of Miami with funding from Salesforce.org, we are proud to offer this free, open-source solution to the higher education community.

Contents

Introduction

Purpose

The purpose of this document is to provide a detailed overview of the technical aspects of Interactions for Student Recruitment. It will explain the purpose behind the functionality, features, business logic, expected behavior, and opportunities for customization. It is intended as a specification document for stakeholders, administrators, and developers.

Product Perspective

Interactions serve as a record of a touchpoint with an individual. Interactions can be created manually from an existing Lead, Contact, Opportunity, Campaign, or from the Interaction tab itself. They can also be mass created through integration or a list load.

In addition to tracking points of contact, the Interaction object operates as a staging table. When an Interaction record is created at a status of "New," an Apex process runs and leverages Salesforce functionality such as Lead Conversion and Duplicate Management to upsert other standard and custom records. Interaction Mapping records determine which fields are updated from the Interaction record.

Introduction

Product Perspective (cont.)

HEDA must be installed prior to installing the Interactions for Student Recruitment package. It is possible to install Interactions without HEDA by downloading the Interactions for Student Recruitment repository on GitHub and removing references to HEDA fields and objects in the code and package before installing. This will require a developer, and if this approach is chosen, the process will not use HEDA objects, fields, or the functionality provided through HEDA triggers referenced above. Additional testing will be required to confirm Interactions works as expected. See the Installation and Configuration Guide for more information.

Business Logic

Data is recorded on the Interaction record and used to act on any related records. Entry of a new Interaction may result in all the following depending on the data saved on the Interaction record or found on existing records:

- Lead creation or update
- Campaign Member creation or update
- Lead conversion
- Contact creation or update
- Account creation (for created Contacts)
- Opportunity creation or update
- Affiliation creation or update

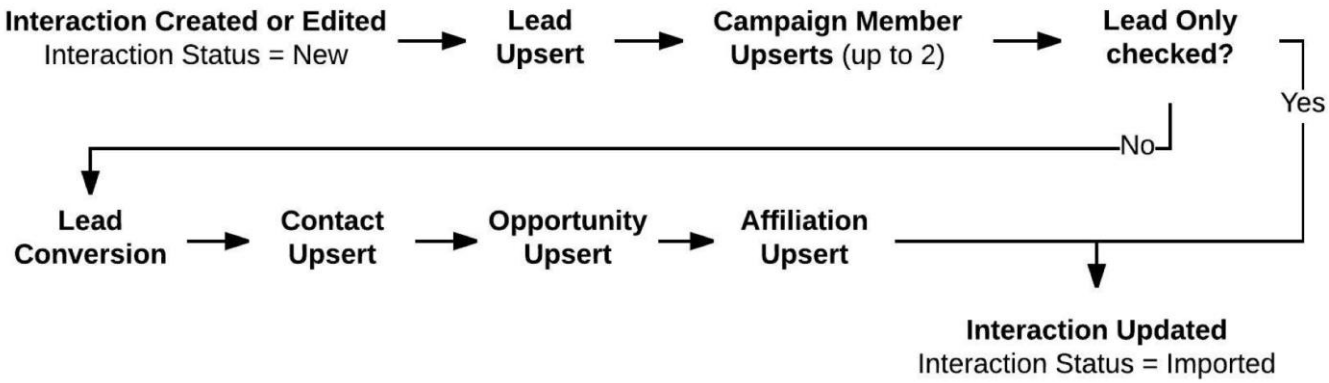
When an Interaction is saved at a "New" status, an Apex process runs. It begins by creating or updating a Lead. Up to two Campaign Members may be created or updated and associated to the Lead. The process ends here if the "Lead Only" checkbox is checked. This may be used to import or create records for individuals who are part of the recruitment process, but have not yet raised their hands according to the university's business processes.

Business Logic

The Lead Only process does not query for Contacts. If a Contact with the same name and email or other matching information is in the system, the Lead will still be created. Users should search Salesforce for a matching Contact before checking the “Lead Only” checkbox when manually creating an Interaction. If you are loading a list of records as Lead Only and you want to merge those into existing Contacts, you will need to complete discovery before or after the list load to find matching Contacts and resolve those.

If the "Lead Only" checkbox is not checked, the Apex process uses native Lead conversion to create or update a Contact and Account. The Account created for the Contact is set to the Administrative Account Record Type used by HEDA. If academic interest information is provided, an Opportunity is created or updated as well. One Affiliation may be created or updated and associated to the Contact by leveraging the HEDA Affiliation creation process.

Here is an overview of this process:



Business Logic

Leveraged Functionality

The Interactions process leverages existing functionality when possible for improved efficiency, maintenance, and scalability. The code relies on standard Salesforce functionality as well as functionality provided through HEDA.

The Interactions code relies on standard Salesforce functionality for the following steps:

1. Avoiding the creation of a duplicate Lead using Duplicate Management (see Lead Creation and Lead Update)

2. Avoiding the creation of a duplicate Contact using Duplicate Management (see Lead Conversion, Contact Creation, and Contact Update)

3. Converting a Lead into a Contact, Account, and Opportunity (See Lead Conversion)

4. Moving Campaign Members and Activities to the Contact during Lead conversion (See Lead Conversion)

5. Merging duplicate Lead and Contact Campaign Member records during Lead conversion (See Lead Conversion)

6. Avoiding the creation of a duplicate Opportunity using a unique external ID field (see Opportunity Creation and Opportunity Update)

Business Logic

Leveraged Functionality (cont.)

The Interactions code relies on HEDA functionality for the following steps:

1. Using the Administrative Account Record Type to set the Account during Lead conversion (see Lead Conversion). The HEDA trigger is not used for this, but the code sets the Account to this HEDA record type.

2. Creating a primary Affiliation based on an update to one of the Primary Affiliation fields on Contact (see Lead Conversion)

3. Updating the appropriate Primary Affiliation field on Contact depending on the type of Affiliation created. (See Affiliation Creation)

Business Logic

Process Overview

The following sections summarize the steps of the process in the order in which they occur.

Lead Creation

The Interaction process always starts by creating or updating a Lead. If the "Lead Only" checkbox is set to true, the process will not attempt to convert the Lead into a new or existing Contact. This may be used to import purchase lists, update information about individuals who have not yet raised their hands, etc.

The Apex process leverages native Salesforce Duplicate Management to update any existing Leads rather than create new ones. For more information about the basic matching and duplicate rule functionality, review Salesforce's Managing Duplicate Records in Salesforce resource.

If the Interaction data does not trigger the Lead Duplicate rule based on the Lead Matching rule, the new Lead will be saved with data from the Interaction record, based on the Interaction Mapping records. The process then sets the Lead lookup field on the Interaction record to the created Lead using the Salesforce ID.

Business Logic

Process Overview (cont.)

Lead Update

If the Lead Duplicate rule is triggered because of a match based on the Lead Matching rule, the code uses the Salesforce ID for the existing Lead provided by the error to update that Lead rather than create a new one. The Lead will be updated with data from the Interaction record based on the Interaction Mapping records. The process then sets the Lead lookup field on the Interaction record to the existing Lead using the Salesforce ID.

Campaign Member Creation

If the Campaign or Additional Campaign lookup fields are populated on the Interaction along with the Campaign Member Status or Additional Campaign Status, the code attempts to create Campaign Members for each of the campaigns associated to the new or updated Lead. The Campaign Member records are created with an Upsert Key that must be unique and an error will be returned if the process attempts to create a Campaign Member record with an Upsert Key that already exists. If no error is returned, the record will be saved. The Campaign Member Status selected must match a Campaign Member Status set up on the Campaign selected. If not, the Campaign Member will be created with the default status assigned to that Campaign.

Business Logic

Process Overview (cont.)

Campaign Member Update

If the system returns any errors because the Upsert Key for a Campaign Member is not unique, the process uses the Salesforce ID found in the error messages to update the existing records rather than create new ones.

Lead Conversion

If an Interaction indicates a Contact should be created or edited (Lead Only checkbox = false), the process attempts to convert the Lead into a new Contact. If the Contact Duplicate Rule is not triggered by the creation of this Contact, the record will save (see Contact creation section below). Otherwise, the Lead will be converted into an existing Contact (see Contact Update section below).

Business Logic

Process Overview (cont.)

Lead Conversion (cont.)

The code leverages the standard Lead conversion process, which creates a Contact and Account, as well as an Opportunity if desired. The Account Name is set using the Company field on Lead, which should be mapped from the Last Name provided on the Interaction. The Account Record Type is set to the HEDA Administrative record type. The conversion process uses the Lead Mappings to update standard fields such as Email, Phone, Lead Source, etc., by default. The Lead Mappings must be set for custom fields included in the package (see the Interaction Mappings section in this document). The standard conversion process also includes moving any standard related records to the Contact record, such as Open Activities, Closed Activities, and Campaign Members. If a Lead is converted into a Contact and both the Lead and Contact have a Campaign Member record for the same Campaign, the system will automatically merge the Lead's Campaign Member record into the existing Campaign Member record. This is standard functionality and not a customization provided by Interactions for Student Recruitment.

One of the Lead fields included in the package is the Primary Institution lookup field. This allows a university to track the Educational Institution associated to the Lead when loading purchase lists and adding individuals as Leads because they have not raised their hands. However, the Interaction process only creates an Affiliation based on the Affiliated Account field on Interaction during the Contact Creation or Contact Update steps.

Business Logic

Process Overview (cont.)

Lead Conversion (cont.)

Although Primary Institution is a useful field for Leads, and is leveraged to prepopulate the Interaction Action on Lead, the Primary Institution on Lead should not be mapped to the Primary Educational Institution on Contact using either the standard Lead Mappings or an Interaction Mapping record. Review the following hypothetical situation to understand how mapping this field could result in conflicting data:

1. A Lead exists in the system with Washington High listed in the Primary Institution field.

2. An Interaction is created for the same individual with Lincoln High listed in the Affiliated Account field and the Primary Affiliation checkbox set to True.

3. The Interaction is saved and the process runs.

4. The process identifies the existing Lead, updates, and converts it to a Contact. Either the standard Lead mappings or an Interaction Mapping sets the Primary Educational Institution field on Contact to Washington High.

5. A HEDA trigger creates an Affiliation record connecting the Contact to Washington High with the Primary checkbox set to true based on the change to this Contact field (this is a standard HEDA process).

6. The Interaction process updates the other Contact fields, then creates an Affiliation connecting the Contact to Lincoln High with the Primary checkbox set to true.

Business Logic

Process Overview (cont.)

Lead Conversion (cont.)

7. The creation of an Affiliation record with Primary set to true would normally kick off another standard HEDA trigger that would update the Primary Educational Institution field on Contact and uncheck the existing primary Educational Institution Affiliation. However, because the Primary Educational Institution has already been updated once in this transaction (see step 4), the HEDA trigger does not run.

8. You are left with a Contact that has Washington High listed in the Primary Educational Institution field and two Affiliations marked as primary: one for Washington High, and one for Lincoln High.

Business Logic

Process Overview (cont.)

Lead Conversion (cont.)

A university has several options for addressing this situation:

- Consider it an outlier based on the data, map to the Primary Educational Institution field on Contact anyway, and create a report that would identify records with this issue so they can be manually fixed.
- Write a custom Apex Trigger to create an Affiliation from the Lead field upon conversion and do not map to the Primary Educational Institution field on Contact.
- Do not map to the Primary Educational Institution field on Contact. Use the Primary Institution field on Lead only to prepopulate the Lead Action when manually creating an Interaction. Include the Institution on an any mass uploads to Interactions that need it. This is the design used in this package.

Business Logic

Process Overview (cont.)

Contact Creation

If the Lead conversion errors because of the Contact Duplicate rule, the process uses the Salesforce ID provided in the error to merge the Lead into the existing Contact instead of creating a new Contact. Any values on the Lead record that are mapped using standard Lead Mapping on the Lead object will be updated on the Contact. After this happens, the process updates any fields on the Contact based on the values on the Interaction record using the Interaction Mappings. The process then sets the Contact lookup field on the Interaction record to the existing Contact using the Salesforce ID.

Contact Update

If the Lead conversion errors because of the Contact Duplicate rule, the process uses the Salesforce ID provided in the error to merge the Lead into the existing Contact instead of creating a new Contact. Any values on the Lead record that are mapped using standard Lead Mapping on the Lead object will be updated on the Contact. After this happens, the process updates any fields on the Contact based on the values on the Interaction record using the Interaction Mappings. The process then sets the Contact lookup field on the Interaction record to the existing Contact using the Salesforce ID.

Business Logic

Process Overview (cont.)

Opportunity Creation

The Interaction Process attempts to create a new Opportunity when there is a value in the Opportunity Key formula field. By default, the Opportunity Key formula field is blank unless there is a value in the Recruitment Interest or Academic Interest field. See Matching with Opportunity Key for detailed information about the Opportunity Key. If the Opportunity Key is blank, the Interaction skips all Opportunity create or update steps. This is helpful when logging data that does not contain reliable or sufficient recruitment information.

If the Opportunity Key on the Interaction matches an existing Opportunity, Salesforce returns an error and the existing Opportunity is updated.

If no Opportunity with this Opportunity Key exists in Salesforce, the record saves with data from the Interaction based on the Interaction Mapping records. The process then sets the Opportunity lookup field on the Interaction record to the created Opportunity using the Salesforce ID.

Business Logic

Process Overview (cont.)

Opportunity Update

If the process attempts to create an Opportunity with the same Opportunity Key as a record that exists in Salesforce, the system returns an error, which includes the Salesforce ID of the existing Opportunity. The process then updates this existing Opportunity with data from the Interaction record based on the Interaction Mapping records instead of creating a new record.

The process then sets the Opportunity lookup field on the Interaction record to the updated Opportunity using the Salesforce ID.

Affiliation Creation

The Affiliation Key is a formula field on Interaction that represents a unique value leveraged in the Affiliation creation process, similar to Opportunity creation. By default, it is set using the following formula, which ensures only one Affiliation exists per Contact and Account combination:

```
IF(ISBLANK(TEXT(Affiliation_Account__c)), "",
"."+CASESAFEID(Affiliated_Account__r.Id)
```

If the Affiliated Account lookup field on the Interaction is populated, the process will attempt to create a new Affiliation, setting the Upsert Key to the Salesforce ID of the Contact created or updated earlier plus the Salesforce ID of the Account saved in the Affiliated Account lookup field on Interaction. If no Affiliation with this Upsert Key exists in Salesforce, the record will save with data from the Interaction record based on the Interaction Mapping records.

Business Logic

Process Overview (cont.)

Affiliation Creation (cont.)

A workflow rule ensures the Upsert Key field on Affiliation is kept up-to-date in case details on the Affiliation change or an Affiliation is created outside of Interactions.

An admin can change the way this Upsert Key works. For example, Role could be added to the formula as an attribute, which would allow multiple Affiliations to exist per Contact and Account combination, as long as they have different Roles. See the Installation and Configuration Guide for more information.

Affiliation Update

If the process attempts to create an Affiliation with the same Upsert Key as a record that exists in Salesforce, the system will return an error including the Salesforce ID of the existing Affiliation. The process will then update this existing Affiliation with data from the Interaction record based on the Interaction Mapping records instead of creating a new record.

Interaction Mappings

The Interaction Mapping custom object contains records that support Interactions. The code uses these records to determine where to map data from the Interaction to any objects supported in the code. The code supports mapping fields from Interactions to the following objects:

- Leads
- Contacts
- Opportunities
- Campaign Members
- Affiliations

This enables the mapping of additional fields for any of the above objects without updating the Apex code driving the Interactions process. All that is required is an Interaction Mapping record, a field on Interaction to map from, and a field on the target object to map to. The package includes a Sample Data file with a tab for suggested Interaction Mapping records, but an administrator can choose to add additional records, update records to map to different fields, or remove records according to business processes and requirements.

Interaction Mappings

The following fields are included on the Interaction Mapping record:

Field	Field Type	Notes
Target Object API Name	Picklist	The API Name of the object the record maps to
Target Field API Name	Text	The API Name of the field the record maps to
Interaction Source Field API Name	Text	The API Name of the field on Interaction being mapped
Active	Checkbox	The record will be ignored if this field is not checked
Insert Null	Checkbox	Allows the insertion of a null value in the target field
Skip Mapping	Multi-select Picklist	Determines which Interactions will skip the mapping record based on the Interaction Source field

Interaction Mappings

Managing Interaction Mappings

Interaction Mappings allow the Interaction process to update the correct fields with data from the Interaction record without requiring the mappings to be hardcoded. The package contains custom fields on each of the supported objects. A Sample Data file containing suggested Interaction Mappings is also provided for importing.

Adding additional fields is a simple process:

1. Create field on target object(s) and grant any desired Read and Edit access for your users using Profiles or Permission Sets

2. Create field on Interaction record and grant Read and Edit access to your users using Profiles or Permission Sets if they should be able to view and update it

3. Create Interaction Mappings using the target object API Names and Interaction field API Names and set the Active checkbox to true

4. If applicable, update the standard Lead Mappings on the Lead object to map any new Lead fields to the appropriate Contact, Account, or Opportunity fields.

Interaction Mappings

Managing Interaction Mappings (cont.)

An Interaction Mapping record must be created for each target field the Interaction field is being mapped to. Different types of fields can be mapped to each other within certain boundaries. Keep the following in mind, and learn more about field types by reading Salesforce's Custom Field Types Knowledge article.

- The Interaction field cannot have values exceeding the character limit of the target field.
    - Either extend the character limit or change the field type of the target field.
    - Ex: Mapping a Text field to a Text Area field is not a problem, but mapping a Text Area field to a Text field may be. There is no automatic truncation, instead the code will error.
- The target field will not accept values that do not meet the Salesforce formatting requirements for its field type.
    - Number, Percent, and Currency fields cannot contain letters
    - Date fields must be formatted as: YYYY-MM-DD
    - Date/Time fields must be formatted as: YYYY-MM-DD hh:mm:ss
    - Email fields must be valid addresses
    - Phone fields cannot contain letters
    - Postal Code fields cannot contain letters
    - URL fields must be valid web addresses

Interaction Mappings

Managing Interaction Mappings (cont.)

- Formula fields can write to other types of fields, if the Salesforce formatting requirements are followed.
    - Ex: Career is a formula text field returning the Academic Career found on either the Academic Interest or Recruitment Interest, depending on which is filled out. This field can be mapped to a Career picklist field on Opportunity, because picklist fields accept text values.
- The following field types cannot be written to, but they can be used as the Interaction source field to write to other types of fields:
    - Auto Number
    - Formula
    - Roll-up Summary
- Checkbox field considerations
    - Checkboxes will only accept values from checkbox or checkbox formula fields. Attempting to map any other type of field to a checkbox, even if it contains 0, 1, TRUE, or FALSE, will cause the code to error.
    - Be careful when mapping checkboxes, because there is no blank option, only TRUE or FALSE. If an unchecked checkbox is mapped to a checked checkbox, the field will be edited and set to FALSE, even if the intent was to ignore the field. Use the “Skip Mapping” setting to prevent this for certain Interaction Sources, or avoid using checkbox fields.

Interaction Mappings

Managing Interaction Mappings (cont.)

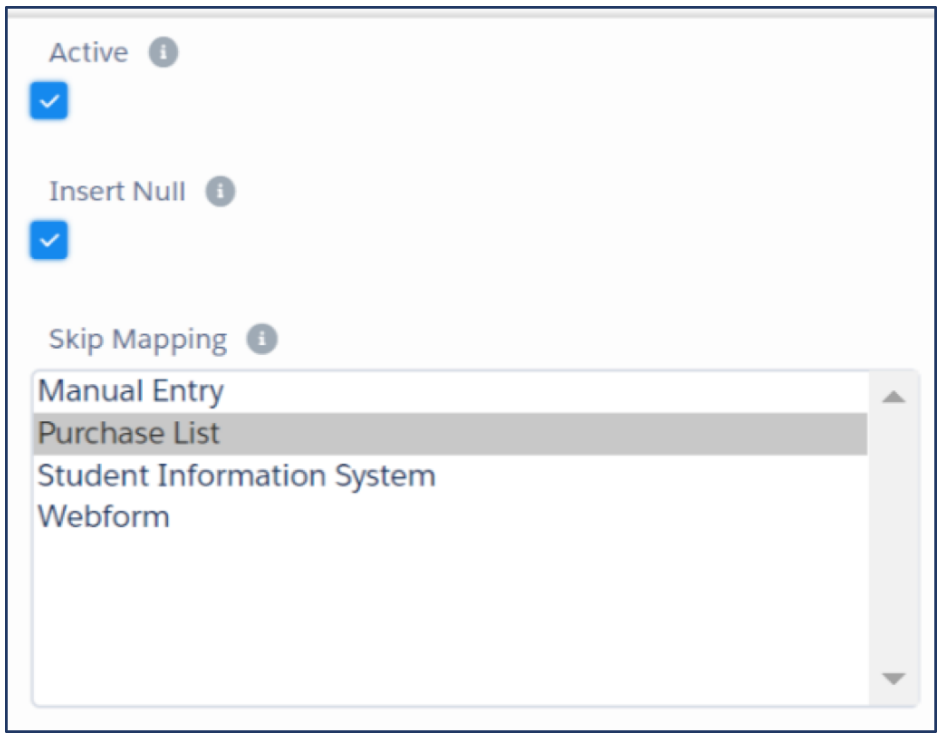
- Lookup fields are set using a valid 15-digit or 18-digit Salesforce ID. This can be done by mapping from a lookup field, formula text field, or text field on the Interaction.
- If a picklist or multi-select picklist field is set to restrict the values and the Interaction source field contains values not defined on the picklist, the code will error at that point, meaning any remaining field mappings will not be processed. If you notice inconsistencies in the way fields are being written from Interactions, this may be the reason. Either ensure the Interaction source field contains only valid values, or uncheck this restriction on the picklist.

Interaction Mappings

Interaction Mappings Advanced Functionality

The Interactions process uses Interaction Mappings to push values into target fields, not blanks. For example, if a Contact has a Home Phone and an Interaction matching that Contact is created that does not include Home Phone, the existing Contact's Home Phone will not be overwritten with a blank, even though that field is mapped. Interaction Mapping records include optional "Insert Null" business logic to bypass this feature.

The Interaction process reviews all Interaction Mappings when creating or updating records, but the Active checkbox and Skip Mapping multi-select picklist business logic on Interaction Mappings can bypass this as well.



Interaction Mappings

Interaction Mappings Advanced Functionality (cont.)

Active Checkbox

If the Active checkbox is not checked, the Interaction Mapping will not be used by any Interaction record. This is helpful when creating additional Interaction Mappings that are not yet ready to be tested, or inactivating existing Interaction Mappings that are causing errors.

Insert Null Checkbox

The Insert Null checkbox indicates a null value should write over a non-null field for that Interaction Mapping.

Skip Mapping Multi-Select Picklist

This multi-select picklist contains values matching the Interaction Source picklist on Interactions. If an Interaction is created or set to “New” with an Interaction Source matching one of the values in the Skip Mapping field, the Interaction Mapping will be ignored. On the following pages, we have two examples of how this functionality can be used:

Interaction Mappings

Interaction Mappings Advanced Functionality (cont.)

Skip Mapping Multi-Select Picklist (cont.)

Example 1

Purchase lists are loaded into Salesforce as Interaction records. They sometimes include a Middle Name. This is considered important enough information to store on the Interaction record for someone to review, but the data is often messy and should not be mapped to the Contact record. However, when a user manually creates an Interaction record with a Middle Name, it should map to the Contact. To meet these requirements, the Interaction Mapping for Middle Name is created and the Skip Mapping field includes the value “Purchase List” to ensure Middle Name is not mapped for Interactions with a “Purchase List” Interaction Source.

Interaction Mappings

Interaction Mappings Advanced Functionality (cont.)

Skip Mapping Multi-Select Picklist (cont.)

Example 2

Another common example is in handling checkboxes. By default, null fields do not overwrite fields with values in the Interaction process, but a null checkbox is not the same thing as a null text field; it could represent a legitimate value of “False.” Because of this, any Interaction containing a mapped unchecked checkbox will write over the checked checkbox it is mapped to. If a certain Interaction Source will never contain data related to this checkbox, this source should be listed in the Skip Mapping field on the Interaction Mapping record so the checkbox is not incorrectly set to “False.”

The Skip Mapping field on Interaction Mappings and the Interaction Source field on Interactions contain four default values. These can be renamed or deleted and additional values can be added to match business processes, but the values must be the same in both fields. These are the values that come out-of-the-box:

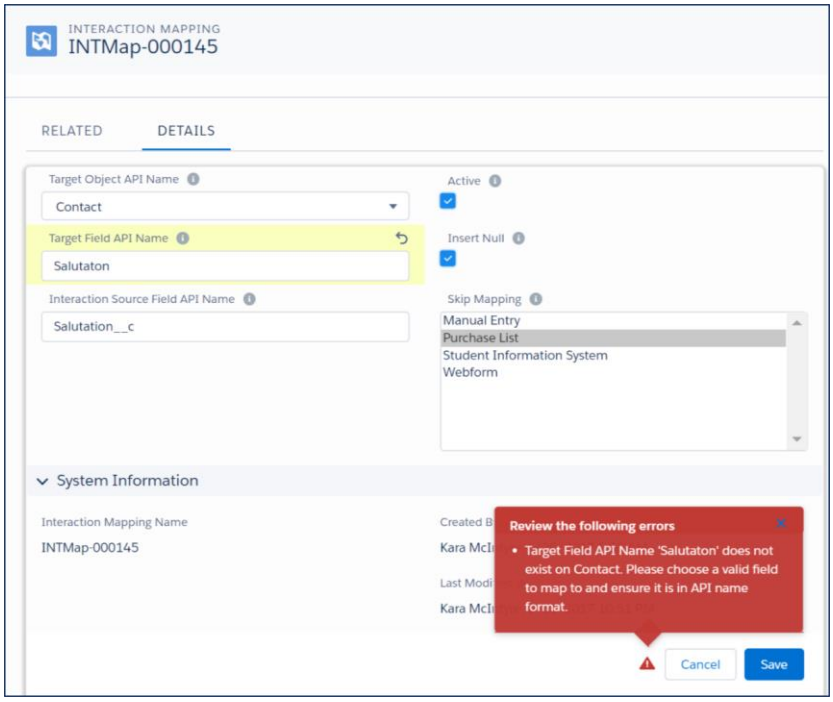
- Manual Entry
- Purchase List
- Student Information System
- Webform

Interaction Mappings

Errors

Each time an Interaction Mapping is created or edited, an Apex process included in Interactions for Student Recruitment queries Salesforce to ensure the value in the Interaction Source Field API Name is valid, and the value in the Target Field API Name is valid based on the Target Object API Name selected.

In the following example, the Target Field API Name is misspelled:



If the API Name of a field included in an Interaction Mapping is changed, the related Interaction Mapping will not work and will cause the Interaction process to error. Interactions for Student Recruitment does not include functionality preventing API Names from being changed, but standard functionality will warn a user that the change could affect other processes. If the change is necessary, the Interaction Mapping must be updated afterward to prevent any issues.

Interaction Mappings

Connect an Additional Object

It is possible to add functionality to the Interactions process by connecting an additional object. While it involves click configuration, it also requires a developer who understands the Apex code.

Please keep the following in mind when considering adding processes to the Apex code:

- Additional processes add API Calls and could slow down processing times when importing or manually creating Interactions. Always follow Salesforce Best Practices to reduce the impact and avoid hitting limits. See the Trigger and Bulk Request Best Practices and Execution Governors and Limits Apex Developer Guide articles for additional information.
- Salesforce requires unit tests that cover a minimum of 75% of the code before it can be deployed in Production, but a higher percentage is preferred. Consider the best practices outlined in the How to Write Good Unit Tests Apex Developer Guide article.

Interaction Mappings

Connect an Additional Object (cont.)

Here are the required steps:

1. Log into sandbox environment (Apex cannot be edited directly in Production)
2. Create new object or identify existing object
3. Add fields to object, if applicable
4. Add matching fields to Interaction object
5. Update Apex code with additional processes
6. Update Apex test code with additional tests
7. Update Target Object API Name picklist on Interaction Mappings to include API Name for new object
8. Create Interaction Mapping records for fields added (see Interaction Mappings)

Interaction Mappings

Connect an Additional Object (cont.)

For more information on developing in Apex, review the following Salesforce trails and documentation:

Trailhead Modules

- Apex Basics & Database
- Apex Triggers
- Apex Testing
- Asynchronous Apex

Developer Guides

- Apex Quick Start
- Apex Developer Guide
- Trigger and Bulk Request Best Practices
- Execution Governors and Limits
- How to Write Good Unit Tests Apex

Duplicate Matching

The Interaction process utilizes native Salesforce Data.com Administration Duplicate Management Matching Rules and Duplicate Rules. Review the Customize Duplicate Management Rules and Reports Salesforce Knowledge article for more information about creating duplicate and matching rules in Salesforce.

A Lead Duplicate Rule, Lead Matching Rule, Contact Duplicate Rule and Contact Matching Rule are required for the Interactions process to work, and steps for creating basic versions of these rules are included in the Installation and Configuration Guide. These rules can be altered by a system administrator to meet a university's business processes, but there are parameters to follow to ensure Interactions work properly.

Leveraging Duplicate Management

The Interaction process finds matching records in Salesforce by relying on system errors generated by Duplicate Rules. These errors include the Salesforce ID of the record found, and the Interaction process uses that ID to merge or update records.

Duplicate Matching

Leveraging Duplicate Management (cont.)

Lead Duplicate Rule

The Lead Duplicate rule is important not only for Lead Only Interactions, but as the first step in a standard Interaction, when a Lead is created so the Lead Conversion process can be leveraged.

Duplicate Rule Detail — Edit | Delete | Clone | Deactivate

Rule Name	Interaction Lead Duplicate Rule	Order	2 of 3 [Reorder]
Description	This rule is leveraged by the Interaction Process. Action on Create and Edit must be set to Block.		
Object	Lead		
Record-Level Security	Bypass sharing rules		
Action On Create	Block	Operations On Create	☐ Alert ☐ Report
Action On Edit	Block	Operations On Edit	☐ Alert ☐ Report
Alert Text	You're creating a duplicate Lead record. We recommend you use an existing record instead.		
Active	✓		
Matching Rule	Interaction Lead Matching — Mapped	Matching Criteria	((Lead: FirstName FUZZY: FIRST NAME MatchBlank = FALSE) AND (Lead: LastName EXACT MatchBlank = FALSE) AND ((Lead: Email EXACT MatchBlank = FALSE) OR (Lead: Constituent_ID EXACT MatchBlank = FALSE))) OR ((Lead: Company EXACT MatchBlank = TRUE) AND (((Lead: FirstName FUZZY: FIRST NAME MatchBlank = FALSE) AND (Lead: LastName EXACT MatchBlank = FALSE) AND ((Lead: Email EXACT MatchBlank = FALSE) OR (Lead: Constituent_ID EXACT MatchBlank = FALSE)))))
Conditions			

Duplicate Matching

Leveraging Duplicate Management (cont.)

Lead Duplicate Rule (cont.)

The following settings must be included on the Lead Duplicate Rule:

- Object = Lead
- Record-Level Security = Bypass sharing rules
- Action On Create = Block
- Action On Edit = Block
- Active = True
- Matching Rule = Interaction Lead Matching

Setting the Action on Create and Edit to “Block” ensures a match generates an error the Interactions process is able to use to identify the record. Setting the Record-Level Security to “Bypass Sharing Rules” means all records will be included regardless of sharing settings the Salesforce environment may contain.

Duplicate Matching

Leveraging Duplicate Management (cont.)

Lead Matching Rule

The Lead Matching Rule is used by the Lead Duplicate Rule to determine the criteria that represents a record match.

Matching Rule Detail Delete | Clone | Deactivate

Object	Lead
Rule Name	Interaction Lead Matching
Unique Name	Interaction_Lead_Matching
Description	Leveraged by the Interaction process. The rule must be formatted exactly like this: (Your Matching Criteria) OR (Company, Exact, Match Blank TRUE AND (Your Matching Criteria)
Matching Criteria	((Lead: FirstName FUZZY: FIRST NAME MatchBlank = FALSE) AND (Lead: LastName EXACT MatchBlank = FALSE) AND ((Lead: Email EXACT MatchBlank = FALSE) OR (Lead: Constituent_ID EXACT MatchBlank = FALSE))) OR ((Lead: Company EXACT MatchBlank = TRUE) AND (((Lead: FirstName FUZZY: FIRST NAME MatchBlank = FALSE) AND (Lead: LastName EXACT MatchBlank = FALSE) AND ((Lead: Email EXACT MatchBlank = FALSE) OR (Lead: Constituent_ID EXACT MatchBlank = FALSE)))))
Status	Active

The Lead Matching Rule used in the documentation contains the criteria:

> Lead FirstName is a FUZZY match AND Lead LastName is an EXACT match AND (Lead Email is an EXACT match OR Constituent ID is an EXACT match)

For the Interactions process to receive all the information needed in the error message, the Company field must also be included in the rule by following this format:

> (Matching Criteria) AND (Company EXACT OR (Matching Criteria))

Here is how it looks in the Contact Matching Rule used in the documentation:

> (Lead FirstName is a FUZZY match AND Lead LastName is an EXACT match AND (Lead Email is an EXACT match OR Constituent ID is an EXACT match)) AND (Company is an EXACT match OR (Lead FirstName is a FUZZY match AND Lead LastName is an EXACT match AND (Lead Email is an EXACT match OR Constituent ID is an EXACT match))

Duplicate Matching

Leveraging Duplicate Management (cont.)

Lead Matching Rule (cont.)

Keep the following in mind when altering this Matching Rule:

- Follow the formatting described above, including the parentheses
- Any field included in the matching criteria must be mapped from Interaction to Lead using an Interaction Mapping record

Duplicate Matching

Leveraging Duplicate Management (cont.)

Contact Duplicate Rule

The Contact Duplicate rule is important during the Lead Conversion step in the Interaction process.

Duplicate Rule Detail Edit | Delete | Clone | Deactivate

Rule Name	Interaction Contact Duplicate Rule	Order	2 of 3 [Reorder]
Description	This rule is leveraged by the Interaction Process. Action on Create and Edit must be set to Block.		
Object	Contact		
Record-Level Security	Bypass sharing rules		
Action On Create	Block	Operations On Create	Alert Report
Action On Edit	Block	Operations On Edit	Alert Report
Alert Text	You're creating a duplicate Contact record. We recommend you use an existing record instead.		
Active	✓		
Matching Rule	Interaction Contact Matching Mapped	Matching Criteria	((Contact: FirstName FUZZY: FIRST NAME MatchBlank = FALSE) AND (Contact: LastName EXACT MatchBlank = FALSE) AND ((Contact: Email EXACT MatchBlank = FALSE) OR (Contact: Constituent_ID EXACT MatchBlank = FALSE))) OR ((Contact: Account EXACT MatchBlank = TRUE) AND ((Contact: FirstName FUZZY: FIRST NAME MatchBlank = FALSE) AND (Contact: LastName EXACT MatchBlank = FALSE) AND ((Contact: Email EXACT MatchBlank = FALSE) OR (Contact: Constituent_ID EXACT MatchBlank = FALSE))))
Conditions			

Duplicate Matching

Leveraging Duplicate Management (cont.)

Contact Duplicate Rule (cont.)

The following settings must be included on the Contact Duplicate Rule:

- Object = Contact
- Record-Level Security = Bypass sharing rules
- Action On Create = Block
- Action On Edit = Block
- Active = True
- Matching Rule = Interaction Contact Matching

Setting the Action on Create and Edit to “Block” ensures a match generates an error the Interactions process can use to identify the record. Setting the Record-Level Security to “Bypass Sharing Rules” means all records will be included regardless of sharing settings the Salesforce environment may contain.

Duplicate Matching

Leveraging Duplicate Management (cont.)

Contact Matching Rule

The Contact Matching Rule is used by the Contact Duplicate Rule to determine the criteria that represents a record match.

Matching Rule Detail Delete Clone Deactivate

Object	Contact
Rule Name	Interaction Contact Matching
Unique Name	Interaction_Contact_Matching
Description	Leveraged by the Interaction process. The rule must be formatted exactly like this: (Your Matching Criteria) OR (Account Name, Exact, Match Blank TRUE AND (Your Matching Criteria)
Matching Criteria	((Contact: FirstName FUZZY: FIRST NAME MatchBlank = FALSE) AND (Contact: LastName EXACT MatchBlank = FALSE) AND ((Contact: Email EXACT MatchBlank = FALSE) OR (Contact: Constituent_ID EXACT MatchBlank = FALSE))) OR ((Contact: Account EXACT MatchBlank = TRUE) AND ((Contact: FirstName FUZZY: FIRST NAME MatchBlank = FALSE) AND (Contact: LastName EXACT MatchBlank = FALSE) AND ((Contact: Email EXACT MatchBlank = FALSE) OR (Contact: Constituent_ID EXACT MatchBlank = FALSE))))
Status	Active

The Contact Matching Rule used in the documentation contains the criteria:

> Contact FirstName is a FUZZY match AND Contact LastName is an EXACT match AND (Contact Email is an EXACT match OR Constituent ID is an EXACT match)

For the Interactions process to receive all the information needed in the error message, the Account field must also be included in the rule by following this format:

> (Matching Criteria) AND (Account EXACT OR (Matching Criteria))

Here is how it looks in the Contact Matching Rule used in the documentation:

> (Contact FirstName is a FUZZY match AND Contact LastName is an EXACT match AND (Contact Email is an EXACT match OR Constituent ID is an EXACT match)) AND (Account is an EXACT match OR (Contact FirstName is a FUZZY match AND Contact LastName is an EXACT match AND (Contact Email is an EXACT match OR Constituent ID is an EXACT match))

Duplicate Matching

Leveraging Duplicate Management (cont.)

Contact Matching Rule (cont.)

Keep the following in mind when altering this Matching Rule:

- Follow the formatting described above, including the parentheses
- The matching criteria on the Contact Matching Rule must match the matching criteria from the Lead Matching rule as closely as possible
- Any field included in the matching criteria must be mapped to the Contact in the standard Lead Mappings in setup on the Lead Object
- Any field included in the matching criteria must be mapped from Interaction to Contact using an Interaction Mapping record

Duplicate Matching

Matching with Opportunity Key

Duplicate Management does not support Opportunities, so the Interactions process uses the Opportunity Key field on Interactions and Opportunities to match records. The Opportunity represents the prospect or applicant's interest in enrolling in a plan and term. Some information received should update the Opportunity, while other information may signify a new recruitment interest and require a new Opportunity to be created. This key can be customized to meet your university's recruitment process. See the Installation and Configuration Guide for more information. Read on for an explanation of the key provided in the package.

Duplicate Matching

Matching with Opportunity Key (cont.)

The Opportunity key is made up of five main parts:

Contact ID	The 18-digit Salesforce ID for the Contact matched or created from the Interaction	It is not in the Opportunity Key formula field on Interactions, because it is added by the Interactions code after the Contact is updated or created, but before the Opportunity is updated or created.
Career	A text value from a picklist set on Plan records. Possible values are Graduate and Undergraduate.	The Career from the Recruitment Interest associated to the Academic Interest will be used, unless Academic Interest is blank, then the Career from Recruitment Interest will be used.
Term ID	The 18-digit Salesforce ID for the Term listed on the Interaction	
Academic Interest	The official degree an applicant applies to and is enrolled in	It is only used in the key for a decentralized recruitment model where a plan change represents a new Opportunity. By default, when the Career = Graduate.
Recruitment Interest	The marketable degree a prospect first shows interest in, which may be a combination of Academic Interests or a plan that is not yet offered	It is only used in the key when Academic Interest is blank for a decentralized recruitment model, where a plan change represents a new Opportunity. By default, when the Career = Graduate.

These pieces of data are used in the Opportunity Key formula field on Interactions, the Apex code in the Interactions process, and the Opportunity workflow rules.

Duplicate Matching

Matching with Opportunity Key (cont.)

Interaction Opportunity Key Formula

Here is what the Opportunity Key formula field looks like on Interaction, including comments. Keep in mind the Interaction code uses the 18-digit Salesforce ID of the Contact matched or created and adds the value returned to the beginning of the below formula to create the full Opportunity Key:

```
/*The below formula allows for a centralized and decentralized recruitment
model. If there is no Plan, this formula is blank and no Opportunity will be
created or updated.*/


 IF(Academic_Interest__c + Recruitment_Interest__c = "","",


/*If the Academic Interest Career = Undergraduate, or Academic Interest is
blank and the Recruitment Interest Career = Undergraduate, this formula uses
the centralized model (one Opportunity per Term)*/


 IF(OR(ISPICKVAL(Academic_Interest__r.Career__c, "Undergraduate"),
AND(ISBLANK(Academic_Interest__c),
ISPICKVAL(Recruitment_Interest__r.Career__c, "Undergraduate"))),
".Undergraduate."+Term__r.Id,
```

Duplicate Matching

Matching with Opportunity Key (cont.)

Interaction Opportunity Key Formula (cont.)

/*Otherwise, this formula uses the decentralized model (one Opportunity per Academic Interest or Recruitment Interest and Term)*/

".Graduate." +

/*If there is no Academic Interest, the decentralized model adds the Recruitment Interest Id. Otherwise, the model uses the Recruitment Interest ID from the Academic Interest, or if there is none, the Academic Interest ID is used as a last resort. Most Academic Interests should have a Recruitment Interest ID to ensure a smooth process from inquiry to application.*/

IF(ISBLANK(Academic_Interest__r.Id), Recruitment_Interest__r.Id,
IF(ISBLANK(Academic_Interest__r.Recruitment_Interest__c),
Academic_Interest__r.Id,
Academic_Interest__r.Recruitment_Interest__r.Id))+"."+Term__r.Id))

Duplicate Matching

Matching with Opportunity Key (cont.)

Interaction Opportunity Key Formula (cont.)

The formula on the previous page returns one of the following options:

- .Undergraduate.{Interaction.Term__c}
- .Graduate.{Interaction.Academic_Interest__r.Recruitment_Interest__c}.
- {Interaction.Term__c}
- .Graduate.{Interaction.Academic_Interest__r. Interaction.Term__c}
- .Graduate.{Interaction.Recruitment_Interest__r. Interaction.Term__c}
- [NULL]

Please note the following:

- If both Academic Interest and Recruitment Interest are blank or have no Career listed, the formula will return NULL, and will skip the Opportunity steps in the process.
- If the Term is blank, the Opportunity Key will not contain all the necessary information, and this could cause duplicate issues.
- Because inquiries normally provide only a Recruitment Interest and applications an Academic Interest, the Recruitment Interest associated to the Academic Interest is used in the decentralized model when possible so applications received can be easily matched to previous inquiries. Associate Academic Interests to Recruitment Interests where possible to take advantage of this design and prevent duplicate Opportunities from being created.

Duplicate Matching

Matching with Opportunity Key (cont.)

Apex Code Opportunity Key Creation

The Interactions process attempts to create an Opportunity with a CloseDate of Today, a StageName matching the Opportunity_Stage__c field on Interaction, the AccountId of the Administrative Account found or created in an earlier step in the process, and the Opportunity_Key__c equaling the Contact ID of the Contact found or created in an earlier step plus the Opportunity_Key__c field on Interaction, shown below.


```
404
405             // Create new Opportunity for upsert if Opportunity Key is populated.
406             if (!String.isEmpty(interaction.Opportunity_Key__c)) {
407                 Opportunity newOppty = new Opportunity(
408                     CloseDate = System.today(),
409                     StageName = interaction.Opportunity_Stage__c,
410                     Opportunity_Key__c = lcr.getContactId() + interaction.Opportunity_Key__c,
411                     AccountId = lcr.getAccountId()
412                 );
```

Because the Opportunity Key field on Opportunity is a unique text field, Salesforce will send back an error if the key matches an existing Opportunity. This error will include the ID of the existing Opportunity, which the code will use to update the record instead of creating a new one.

Duplicate Matching

Matching with Opportunity Key (cont.)

Apex Code Opportunity Key Creation (cont.)

Opportunity Key Workflow

Opportunities can be created or updated outside of Interactions, so it is necessary to ensure the Opportunity Key is set and correct for all Opportunity records. This is done through a workflow rule and field update.

Workflow Rule

Opportunity: Update Opportunity Key

Help for this Page

Workflow Rule Detail Edit | Clone | Deactivate

Rule Name	Opportunity: Update Opportunity Key	**Object**	Opportunity
Active	✓	**Evaluation Criteria**	Evaluate the rule when a record is created, and every time it's edited
Description	Sets the Opportunity Key based on the Career, Recruitment Interest or Academic Interest, and Term, according to the formula used in the Opportunity Key field on the Interactions object.		

Rule Criteria	/*The below formula allows for a centralized and decentralized recruitment model.*/
	
	/*If the Academic Career from the Academic Interest = Undergraduate or the Academic Interest is blank and the Recruitment Interest Career = Undergraduate, this formula uses the centralized model (one Opportunity per Term). If the same is true for the Graduate Career, this formula uses the decentralized model (one Opportunity per Opportunity Plan and Term). Otherwise, the formula is blank, which means the Opportunity is missing important information.*/ Opportunity_Key__c <> CASESAFEID(Contact__r.Id)+ IF(Academic_Interest__c + Recruitment_Interest__c = "","",
	
	/*If the Academic Interest Career = Undergraduate, or Academic Interest is blank and the Recruitment Interest Career = Undergraduate, this formula uses the centralized model (one Opportunity per Term)*/ IF(OR(ISPICKVAL(Academic_Interest__r.Career__c, "Undergraduate"), AND(ISBLANK(Academic_Interest__c), ISPICKVAL(Recruitment_Interest__r.Career__c, "Undergraduate"))), ".Undergraduate."+Term__r.Id,
	
	/*Otherwise, this formula uses the decentralized model (one Opportunity per Academic Interest or Recruitment Interest and Term)*/
	".Graduate." +
	
	/*If there is no Academic Interest, the decentralized model adds the Recruitment Interest Id. Otherwise, the model uses the Recruitment Interest ID from the Academic Interest, or if there is none, the Academic Interest ID is used as a last resort. Most Academic Interests should have a Recruitment Interest ID to ensure a smooth transition from inquiry to application.*/
	IF(ISBLANK(Academic_Interest__r.Id), Recruitment_Interest__r.Id,
	IF(ISBLANK(Academic_Interest__r.Recruitment_Interest__c), Academic_Interest__r.Id,
	Academic_Interest__r.Recruitment_Interest__r.Id))+"."+Term__r.Id))		

Duplicate Matching

Matching with Opportunity Key (cont.)

Apex Code Opportunity Key Creation (cont.)

Opportunity Key Workflow (cont.)

The workflow rule evaluates to true when an Opportunity is created or edited and the Opportunity_Key__c does not match the expected value based on the Contact ID followed by the Opportunity Key formula.

When the workflow rule evaluates to true, the field update runs, which sets the Opportunity_Key__c field to the Contact ID + the Opportunity Key formula.

Field Update

Opportunity Key: Set Based on Career

Help for this Page

Rules Using This Field Update [1] | Approval Processes Using This Field Update [0]

Field Update Detail Edit Delete

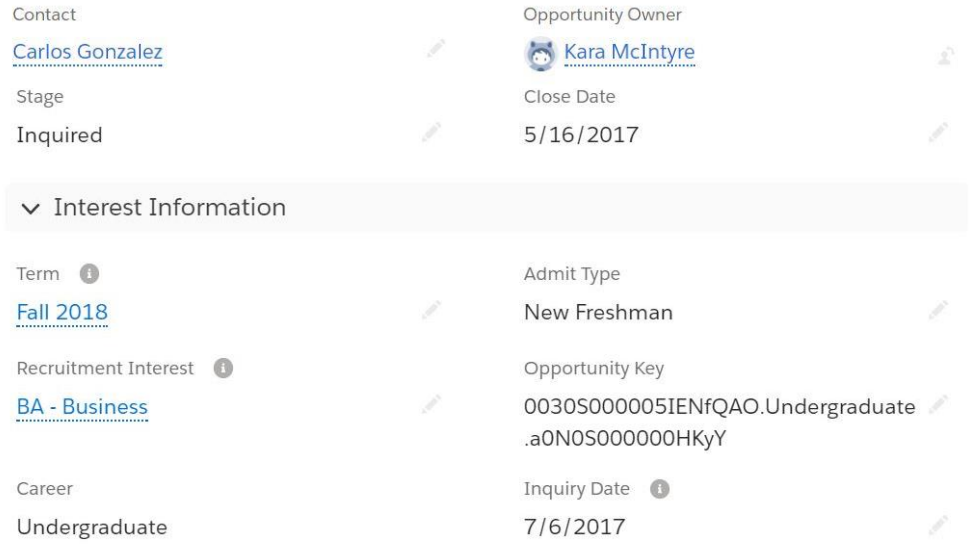
Name	Opportunity Key: Set Based on Career
Unique Name	Opportunity_Key_Set_Based_on_Career
Description	For UG, sets key to Contact ID + Opportunity Career + Term ID, for GR, Contact ID + Opportunity Career + ((Academic Interest ID.Recruitment Interest ID or Academic Interest ID) or Recruitment Interest ID) + Term ID.
Object	Opportunity
Field to Update	Opportunity: Opportunity Key
Field Data Type	Text
Re-evaluate Workflow Rules after Field Change	☐

Formula Value	CASESAFEID(Contact__r.Id)+
	
	/*The below formula allows for a centralized and decentralized recruitment model.*/
	
	/*If both Academic Interest and Recruitment Interest are blank, the formula is blank, which means the Opportunity is missing important information and no Opportunity will be created.*/
	IF(Academic_Interest__c + Recruitment_Interest__c = "","",
	
	/*If the Academic Interest Career = Undergraduate, or Academic Interest is blank and the Recruitment Interest Career = Undergraduate, this formula uses the centralized model (one Opportunity per Term)*/
	IF(OR(ISPICKVAL(Academic_Interest__r.Career__c, "Undergraduate"),
	AND(ISBLANK(Academic_Interest__c), ISPICKVAL(Recruitment_Interest__r.Career__c,
	"Undergraduate"))), ".Undergraduate."+Term__r.Id,
	
	/*Otherwise, this formula uses the decentralized model (one Opportunity per Academic Interest or Recruitment Interest and Term)*/
	".Graduate." +
	
	/*If there is no Academic Interest, the decentralized model adds the Recruitment Interest Id. Otherwise, the model uses the Recruitment Interest ID from the Academic Interest, or if there is none, the Academic Interest ID is used as a last resort. Most Academic Interests should have a Recruitment Interest ID to ensure a smooth transition from inquiry to application.*/
	IF(ISBLANK(Academic_Interest__r.Id), Recruitment_Interest__r.Id,
	IF(ISBLANK(Academic_Interest__r.Recruitment_Interest__c), Academic_Interest__r.Id,
	Academic_Interest__r.Recruitment_Interest__r.Id))+"."+Term__r.Id))

Duplicate Matching

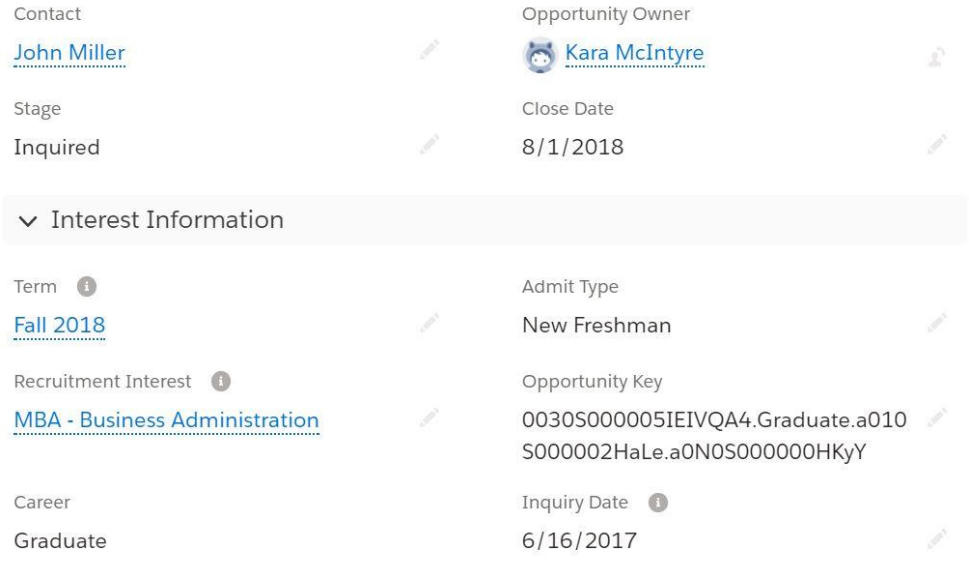
Matching with Opportunity Key (cont.)

Centralized Opportunity Key Example



The Opportunity Key is Contact ID.Undergraduate.Term ID

Decentralized Opportunity Key Example



There is no Academic Interest on this record, so the Opportunity Key is Contact ID.Graduate.Recruitment Interest ID.Term ID

Managing Interactions

Interaction Status

The Interaction Status field on the Interaction object manages the process. To run the code, set the Interaction Status to “New.” This will run the Interaction process regardless if it is a new Interaction record or an existing one. When the code runs successfully, the Interaction Status will change to “Imported.” If something goes wrong during the process, the Interaction Status is changed to “Audit Required” and an error message is saved in the Audit Reason textbox.

New	Used to run the Interaction process for that record. If the Interaction Status is still “New” after the record has been saved, something went wrong and the code never ran. See Troubleshooting Tips.
Imported	Everything processed successfully
Audit Required	The Interaction was created successfully, but the code hit an error and did not fully run. An error should be displayed in the Audit Reason field.

Audit Reasons

There are four main types of Audit Reasons. Each one is based on a set of steps, and some include a system error message that indicates there was an issue (using the “error.getMessage” method.

Managing Interactions

Audit Reasons (cont.)

Interaction Duplicate Error

This error only occurs when loading Interactions in a batch. This section of code prevents the process from failing when multiple Interactions in a batch match to the same Contact. It does this by concatenating First Name + Last Name + Email to create a filter key. If a second Interaction in a batch returns a duplicate filter key, the process will end for that Interaction record. It will be saved at an Interaction Status of Audit Needed, and an Audit reason matching the following:

> Reason: this Interaction was not processed because it is a possible duplicate of -
> ' + filteredMap.get(filterKey).Id + ': ' + filteredMap.get(filterKey).First_Name__c +
> ' ' + filteredMap.get(filterKey).Last_Name__c + ' ' + filteredMap.get(filterKey).Email__c;

To resolve this error, an admin should find the Interaction record that matched based on First Name, Last Name, and Email, and decide whether to:

1. Run it anyway by changing the Interaction Status from Audit Needed to New, either because it is not really a duplicate or it contains the best data and should overwrite the data from the Interaction it matched with
2. Delete the duplicate Interaction and run the correct one if needed by changing the Interaction Status to New
3. Edit fields on one of the Interactions and then run it by changing the Interaction Status to New

Managing Interactions

Audit Reasons (cont.)

Opportunity Update Error

This error occurs if there is no Opportunity associated to the Interaction, but the Opportunity Key is not blank. This could be because no Opportunity was created, or an Opportunity was created and another error prevented it from being written back to the Interaction. See Troubleshooting Tips.

> Reason: Interaction has an Opportunity Key, but could not find associated Opportunity in the system. Please verify that the Opportunity Key is valid.

Lead Insert Error

This error occurs when an issue prevents the process from continuing during the Lead insert step.

> Reason: Error during Lead insert - ' + error.getMessage() + '.';

Lead Conversion Error

This error occurs when an issue prevents the process from continuing during the Lead Conversion step.

> Reason: Error during Lead conversion - ' + error.getMessage()

Managing Interactions

Interaction Source

The Interaction Source field on Interaction is used to track the type of Interaction it is, whether it came from a purchased list, another database, manual entry, or other custom sources. Add options here and in the Skip Mapping multi-select picklist on the Interaction Mapping object to indicate a field should not be updated for that Interaction Source. The Interaction Source field can also be used for reporting, or to clarify for users where the data came from on a technical level.

The out-of-the-box values are:

- Manual Entry
- Purchase List
- Student Information System
- Webform

Managing Interactions

Creating Interactions

Interactions can be created in multiple ways:

- Integration/List load
- Manually by a user from the Interaction object
- Through Quick Actions on the Lead, Contact, Opportunity, or Campaign by a user

Integration/List Load

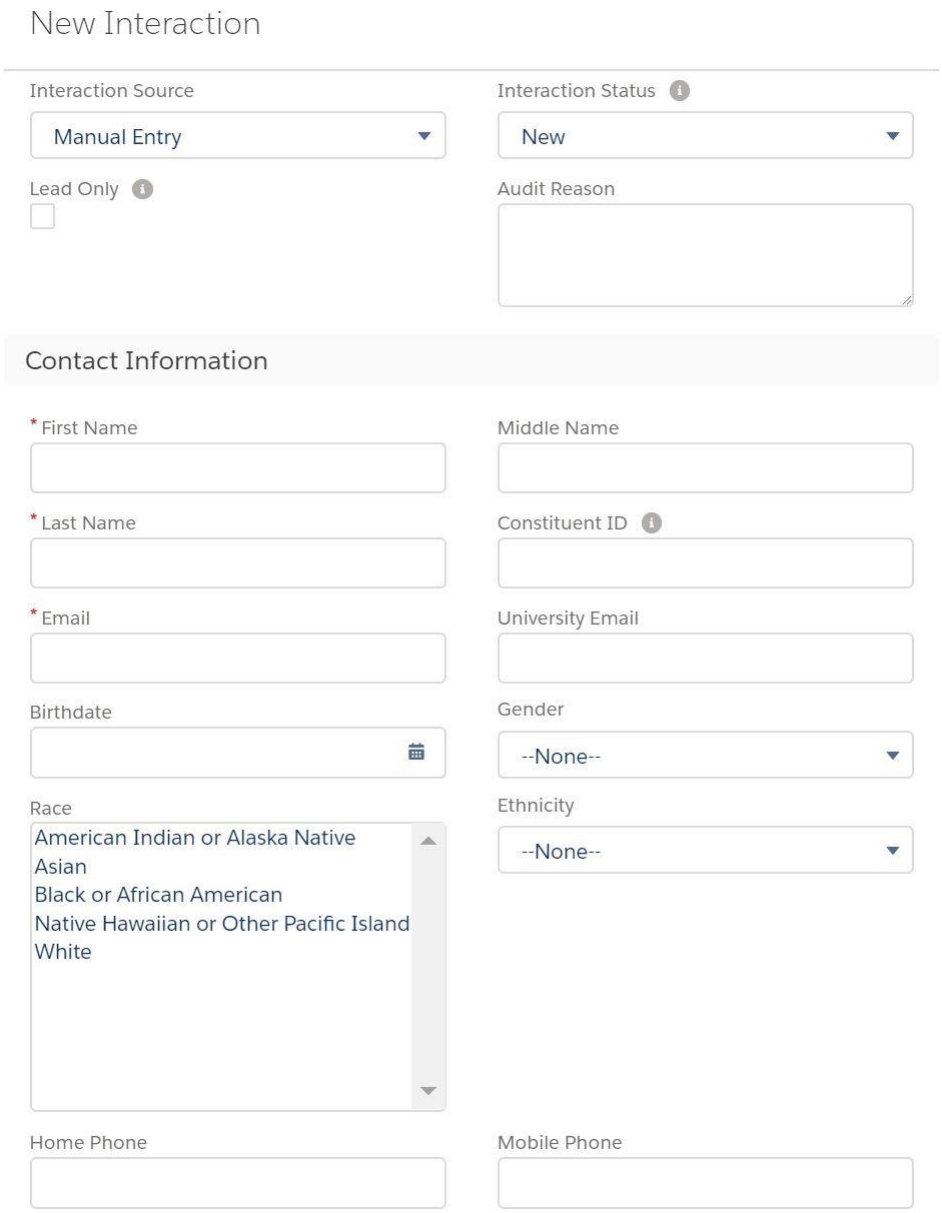
It is best to test the process before loading multiple Interactions through integration or a list load to ensure there are no CPU timeouts. Customizing the Interaction process by adding more steps, or including Flow, Process Builder, or outside Apex processes can slow down load times. Test a small load, then a larger one to determine if the batch size needs to be reduced. Because the Interactions code runs in batches of 200, a batch size of 200 is recommended for integration and list loads to Interactions.

Managing Interactions

Creating Interactions (cont.)

Manually By a User

Any user with access to the Interactions tab can create an Interaction record there by navigating to the tab and clicking the “New” button. Only two fields are defaulted: the Interaction Status is defaulted to “New,” and the Interaction Source is set to “Manual Entry.”



Managing Interactions

Creating Interactions (cont.)

Manually By a User (cont.)

The most important fields to fill out are the ones included in the Duplicate Matching Rules (See Duplicate Management). By default, these fields are First Name, Last Name, and Email or Constituent ID.

Checking the Lead Only checkbox causes the process to ignore Contacts and only match on or create Leads.

Leaving both the Recruitment Interest and Academic Interest blank will result in a blank Opportunity Key and no Opportunity will be created. It is important to also fill out the Term field when setting a Recruitment or Academic Interest to ensure the Opportunity Key contains all the necessary data to match on existing records. The Opportunity Stage defaults to Inquired and can be changed if needed.

Managing Interactions

Creating Interactions (cont.)

Quick Action Overview

A new Interaction can be quickly created for an existing Lead, Contact, Opportunity, or Campaign Member using an Interaction creation quick action on the Lead, Contact, Opportunity, or Campaign respectively. The quick actions are useful when the user has already found the individual in the system and needs to update data without having to determine which Opportunities to update or worry if there are other duplicate records in the system. When the Interaction is created, the full process will run and any duplicate records found will be merged automatically. Each quick action has a different layout with default values for verifying or updating. Other values can be defaulted without including them on the layout.

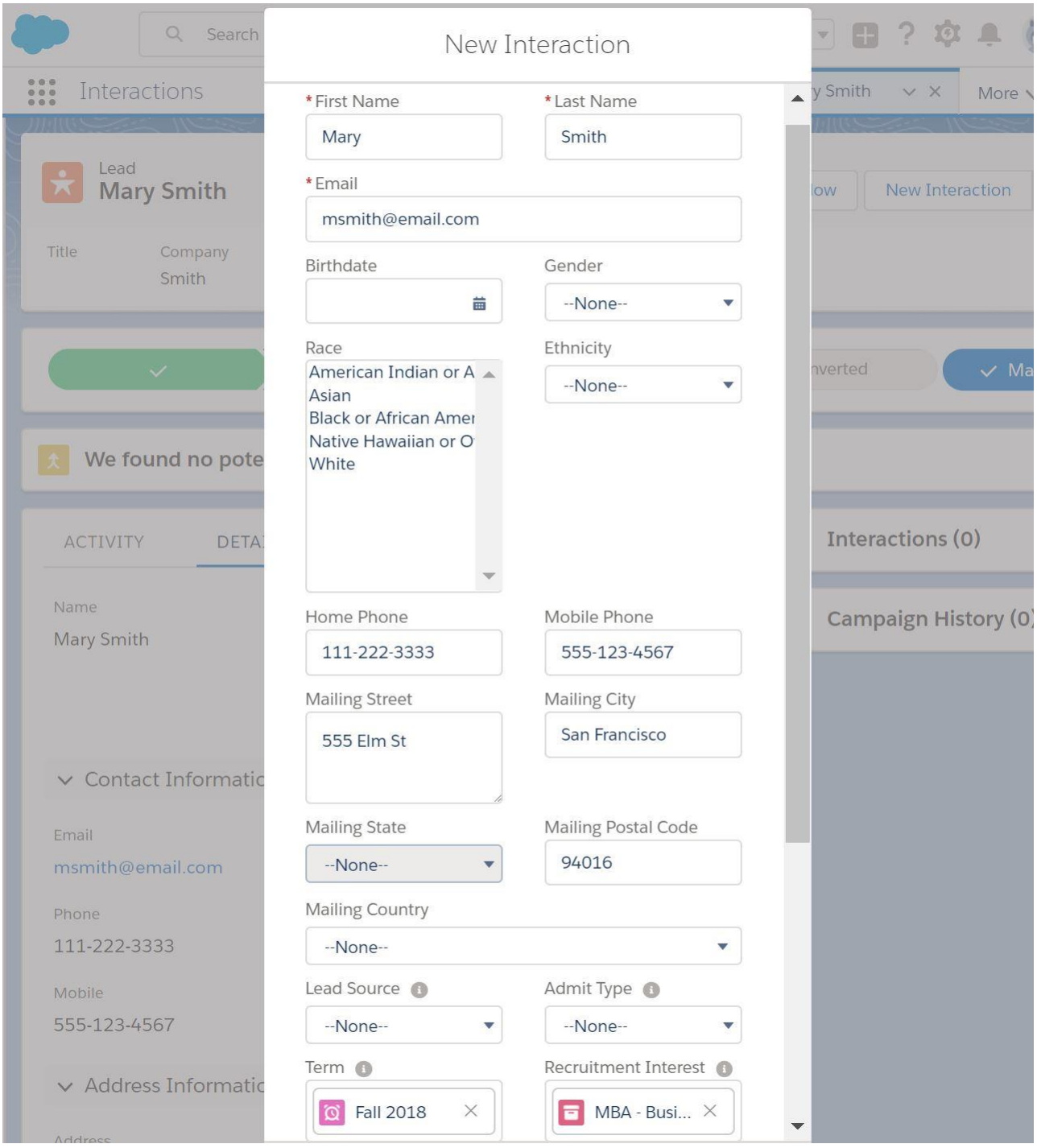
Lead Quick Action

The Lead quick action defaults fields from the Lead that have values. Picklist fields are not fully supported for quick actions and will not default based on their values on the Lead. The Lead Only checkbox is not included on the default layout. The fields defaulted and displayed in the quick action for the initial package are shown in the images in the following images:

Managing Interactions

Creating Interactions (cont.)

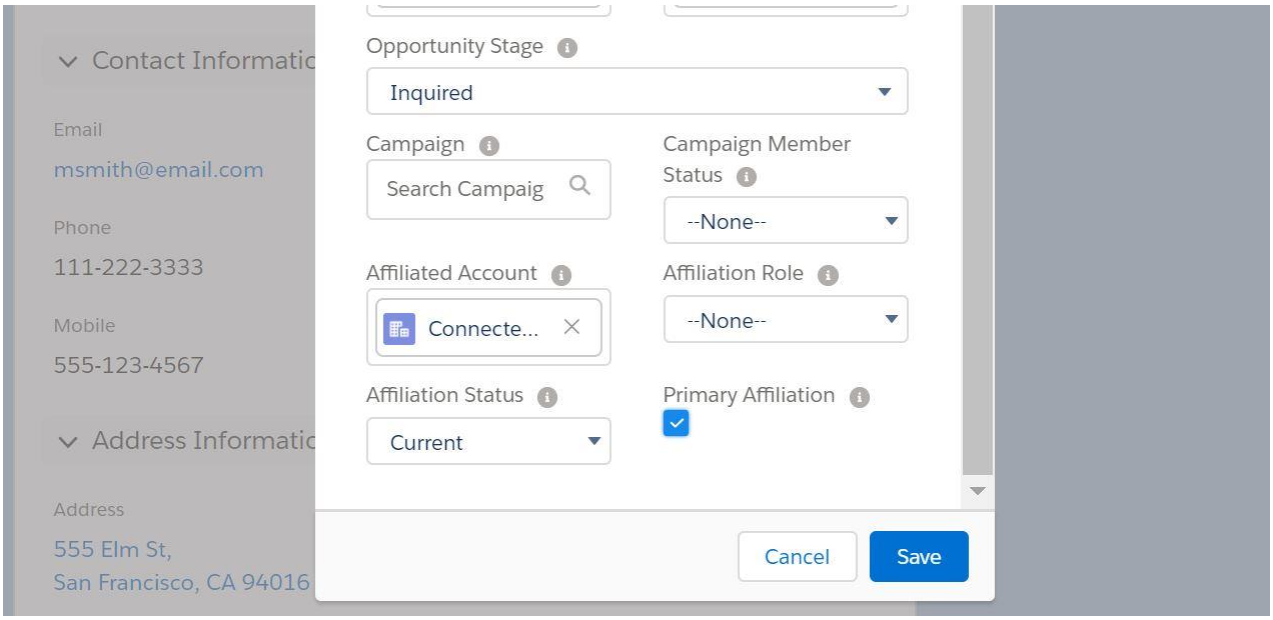
Lead Quick Action (cont.)



Managing Interactions

Creating Interactions (cont.)

Lead Quick Action (cont.)

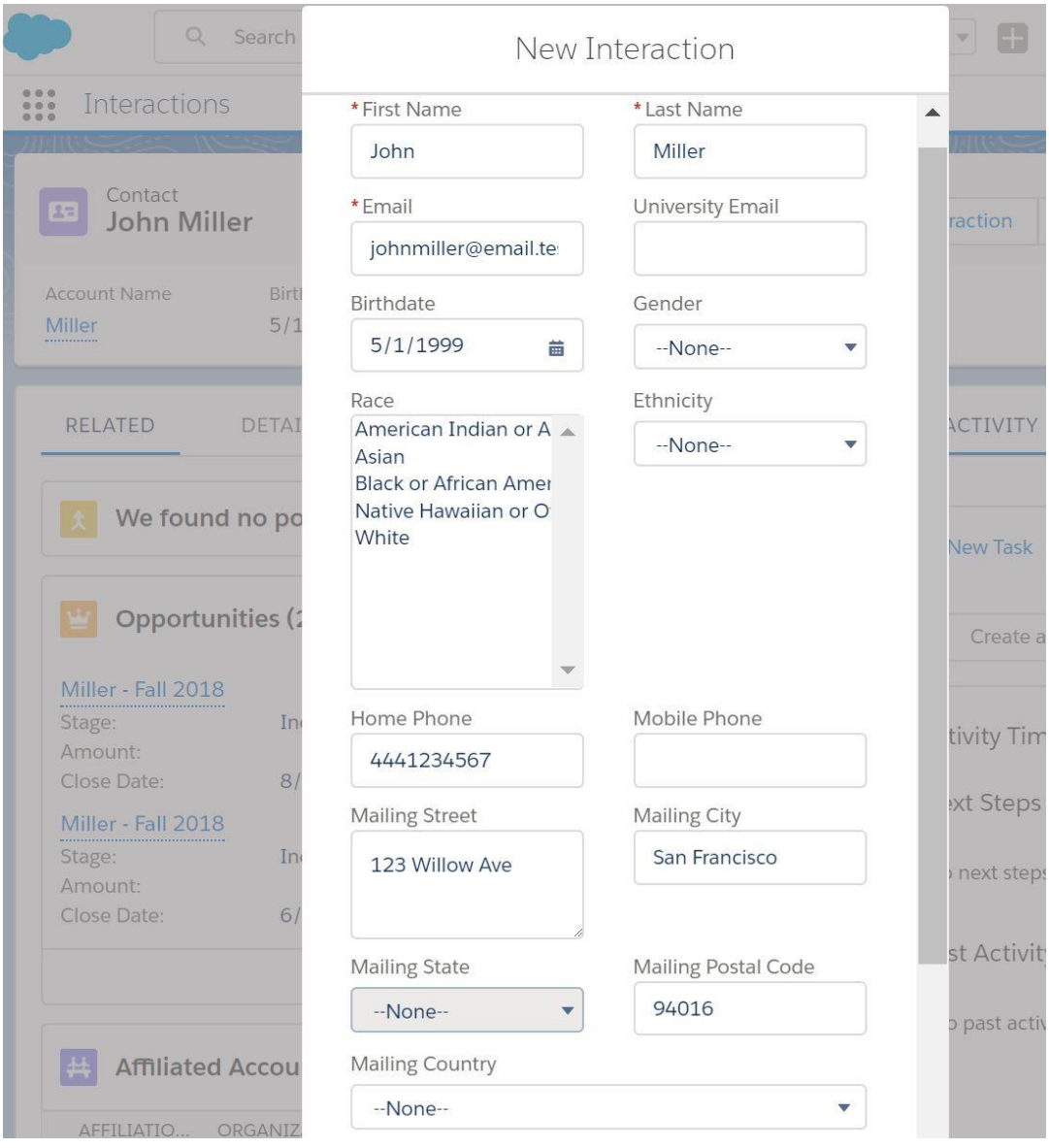


Managing Interactions

Creating Interactions (cont.)

Contact Quick Action

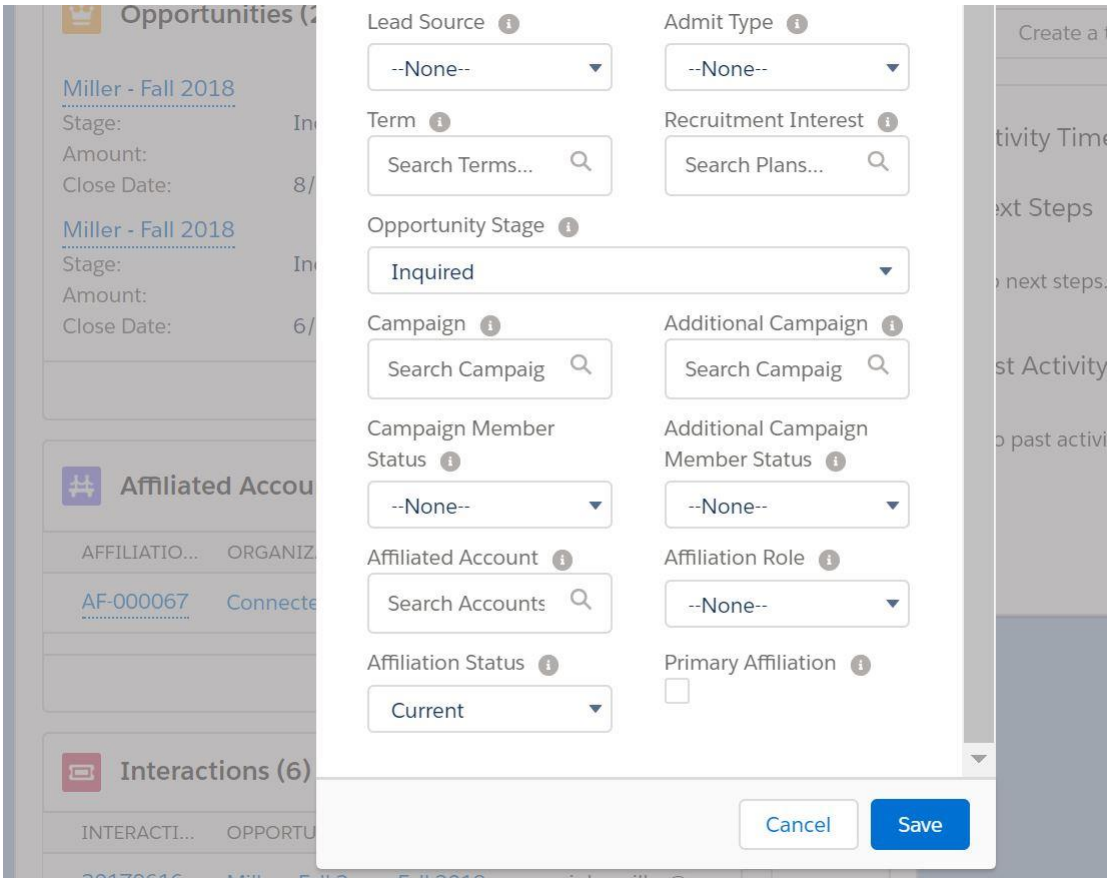
The Contact quick action defaults fields from the Contact that have values. Picklist fields are not fully supported for quick actions and will not default based on their values on the Contact. The Lead Only checkbox is not included on the default layout. The Contact fields defaulted and displayed in the quick action for the initial package are shown in the following images:



Managing Interactions

Creating Interactions (cont.)

Contact Quick Action (cont.)

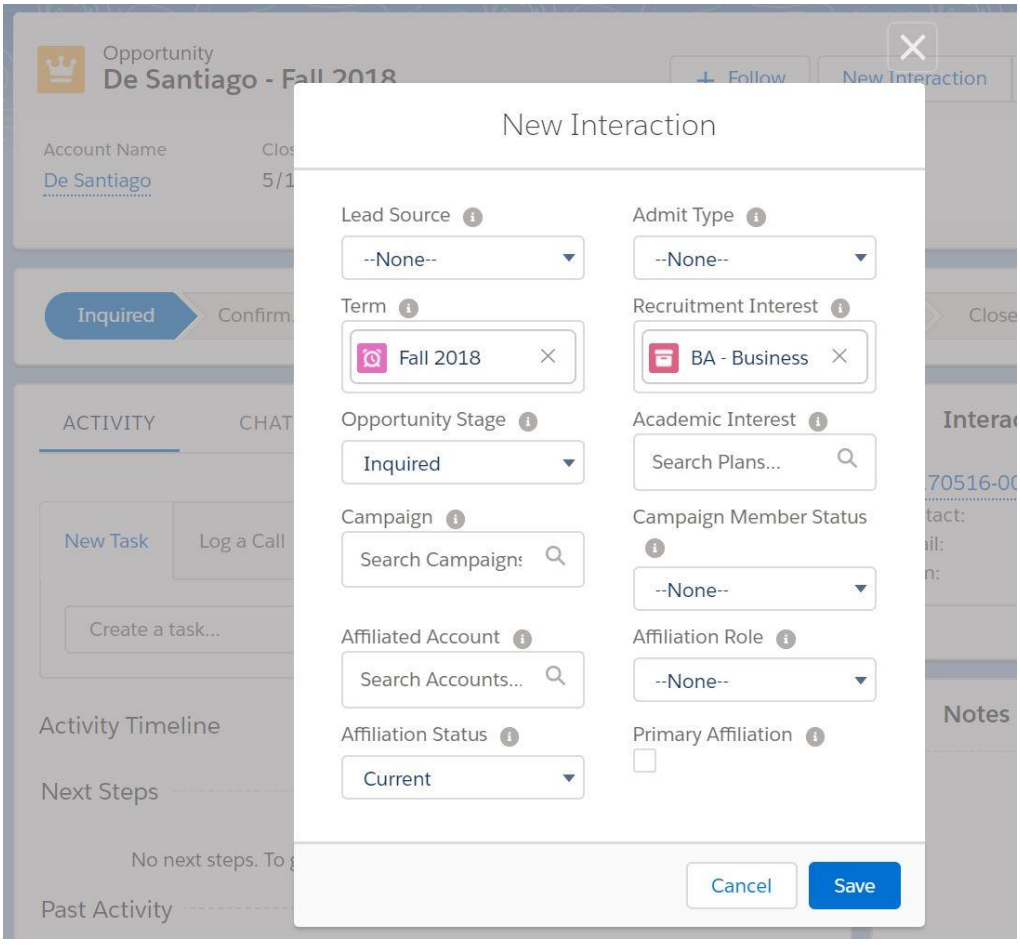


Managing Interactions

Creating Interactions (cont.)

Opportunity Quick Action

The Opportunity quick action defaults fields from the Opportunity and associated Contact that have values. Picklist fields are not fully supported for quick actions and will not default based on their values on the Opportunity or Contact. The Lead Only checkbox is not included on the default layout. Many of the Contact fields are defaulted and not displayed to provide a more streamlined experience. If users need to verify or update Contact fields, the Contact Quick Action should be used instead. The Opportunity and Contact fields included and defaulted in the quick action for the initial package are shown in the following image:



Managing Interactions

Troubleshooting Tips

There are some common errors you may run into when the Interaction Process does not run as expected. The Audit Reasons field is designed to help, but here are some tips for narrowing down the issue quickly:

- If the Interaction Status is at “Imported” but you notice inconsistencies in the way the fields were updated or if duplicate records were created by the process, check the following:
    - Review the data on the Interaction record involved in the process to see if it could be causing individual fields that are mapped to error (See Managing Interaction Mappings).
    - Review Duplicate Rules and Matching Rules and compare the Interaction record involved to see if it could be matching to an unexpected record (See Duplicate Matching).
- If the Interaction Status is at “Audit Needed,” check if a Lead or Contact is associated at the bottom of the Interaction record. If not, the process hung up on matching to or creating a Lead or Contact.
    - Search for a matching Lead and/or Contact to see if there is a matching record and compare their fields to see if you can determine what information caused it. It’s possible fields are blank that are preventing duplicate matching from working, but standard Salesforce functionality is identifying the duplicate another way, like through a field that is set as unique but not in the matching criteria.
    - Look over the data to see if any required field to create a Lead or Contact is missing.

Managing Interactions

Troubleshooting Tips (cont.)

- For “Audit Needed,” if there is a Contact and there is an Opportunity Key, check if an Opportunity is associated at the bottom of the Interaction record. If not, the process hung up on matching to or creating an Opportunity.
    - Look at the Opportunity Key Formula field on Interaction. The value will contain the part of the key following the Contact ID. If this key is missing information (like Term), the process may have trouble matching to an existing Opportunity or determining whether a new Opportunity should be created.
    - Look at other fields required on Opportunity that are not part of the Opportunity Key. If any of these are blank, the process will not be able to create an Opportunity.
    - Search for the Opportunity in Salesforce using the Last Name and Term. If an Opportunity was created but not written back to the Interaction, another system error prevented this step from occurring.

If you have read the Audit Reason and tried these tips and still cannot identify the problem, turn on debug logs and edit and save the Interaction to find more information about the issue. See the Debug Log section in the Salesforce Developer Guide for more information.

Metadata

Apex Classes

INT_InteractionMappingService

Copies data from Interactions to specified Objects based on Interaction Mappings

INT_InteractionMappingService_TEST

Test class for the INT_InteractionMapping Service

INT_InteractionProcessor

Main class for handling Interaction logic, processes Interaction data into related Objects:

- Insert/Convert Leads
- Upsert Opportunities
- Update Contacts
- Upsert Affiliations
- Upsert Campaign Members

INT_InteractionProcessor_TEST

Test class for the INT_InteractionProcessor

INT_OpportunityContactRoleHandler

Creates a Contact Role when an Opportunity is created with a value in the custom Contact lookup field. This allows standard Opportunity report types and Campaign Influence reports to work properly when an Opportunity is associated to an individual (Contact) rather than an organization (Account).

Metadata

Apex Classes (cont.)

INT_OpportunityContactRoleHandler_TEST

Test class for the INT_OpportunityContactRoleHandler

INT_TestDataFactory

Creates test data for Apex Unit Tests

Apex Triggers

INT_Interaction.trigger

Calls the INT_InteractionProcessor class on “after insert” and “after update” actions for Interaction records with a status of “New”

INT_InteractionMapping.trigger

Ensures Interaction Mapping records created are valid, e.g. ensuring First_Name__c is a valid field on the target Object to copy Interaction data to

Custom Objects

- Interaction__c
- Interaction_Mapping__c
- Plan__c