ENTERPRISE APPLICATION SOLUTION FOR YIELD (EASY) TECHNICAL GUIDE

Table of Contents

UPDATED: OCTOBER 2018

OVERVIEW

AUDIENCE

This Technical guide is intended to provide System Administrators with details on various administrator-specific functions such as permission sets and custom settings. It also describes the purpose of each apex class/trigger and visualforce page/component. We recommend you review the User Guide prior to reading this technical guide.

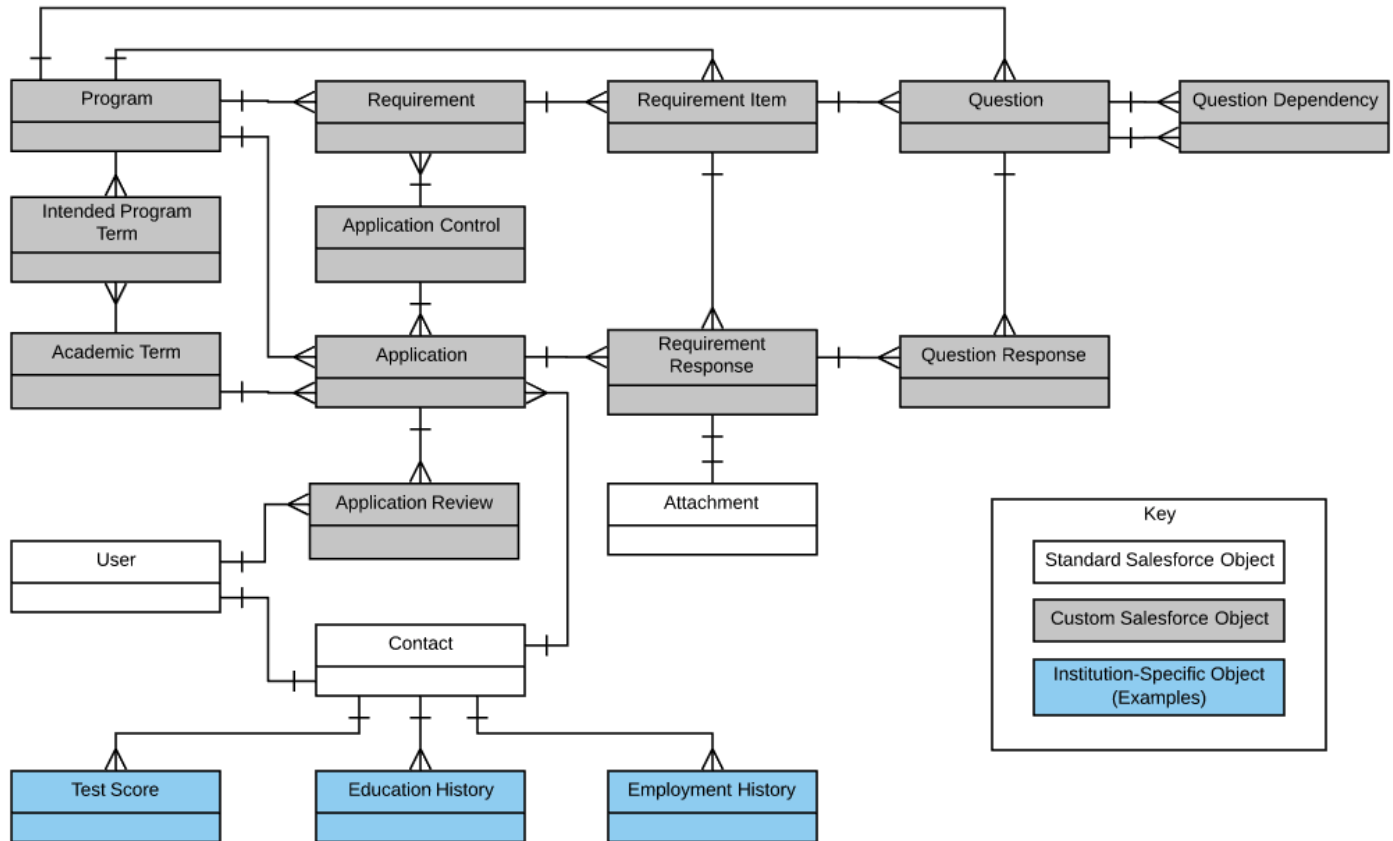## WHAT IS ENTERPRISE APPLICATION SOLUTION FOR YIELD (EASY)?

Enterprise Application Solution for Yield (EASY) is a native form builder with administrative and decision-making tools. It is designed to be a user tool so departments can control their own application. It's an extension of Salesforce's approach to "democratizing" software. That is, the same way that Salesforce democratized software by separating declarative functionality from coding, EASY democratizes declarative and user functionality, separating application and question/requirement setup from the administrative setup of fields and picklist values. These most often are associated with IPEDS and/or SIS fields in global picklist sets and are not likely to change, so this minimizes maintenance time for system administrators.

That also means users control most of the functions of the solution rather than having to heavily rely on the IT department for updates to existing questions or new applications. The user tool approach, then, equates to empowerment for departments.

The solution includes an unmanaged package (the EASY Package) and resource documents (such as this guide).

DECLARATIVE CONFIGURATION

CUSTOM OBJECTS

- Application Control
    - Represents a single application configuration
    - Used to control things like branding, cancellation and withdrawal
- Requirement
    - Used to bundle-together multiple Requirement Items
    - Represented as category on the applicant interface, called the Checklist
- Requirement Item
    - Used to display specific application requirements including questions and uploads (i.e., Question Group, Question, Document Upload, External Requirement and/or Related Object question group)
- Question
    - Represents an individual question asked of the applicant
    - Can be categorized with other questions under a Question Group
- Question Dependency

- ○ Used to show/hide a question based on answers to other questions
- Program
  - ○ Represents a program to which an applicant can apply
- Academic Term
  - ○ Represents a term to which an applicant can apply
- Intended Program Term
  - ○ Used to control application open/close dates for specific types of applicants and/or specific programs
- Application
  - ○ Used to store application-specific data submitted by and associated to an applicant
- Requirement Response
  - ○ Used to store data associated with responses to Requirement Items
  - ○ Keeps track of what has been fulfilled for each Requirement Item on an application
- Question Response
  - ○ Used to store an applicant's response to a question
- Application Administration
  - ○ Summary of application data, responses and files so staff can fulfill application information and documents (e.g., upload official transcripts to the application)
- Application Review
  - ○ Summary of application data, responses and files so readers can "review" an individual application on one screen

INSTITUTION-SPECIFIC OBJECTS

EASY provides functionality to interact with institution-specific custom objects through "Related Object" Requirement Items. Details about how to set up one of these Requirement Items are in the User Guide; however, system administrators will need to create the custom objects, if they don't already exist. To create custom objects for EASY, be sure to follow this guidance:

1. Make sure the object has a lookup or master-detail relationship to Application__c or Contact
2. Add a custom Text field called "Holding_Name__c" with a length of 255 characters
3. Add a custom Checkbox field called "User_Created__c"
4. Add a custom Long Text Area field called "Related_Object_JSON__c" (We recommend setting the length to the maximum allowed.)
5. Add the object API name to the Related_List_Object__c picklist on the Requirement Item object
6. Add the object API name as an entry in the "Related Object Review" custom setting
7. Add read/edit access for the three fields you created to the "Application Community" permission set (or whatever permission set you are using for community users)

PERMISSION SETS

The following permissions sets are included with the EASY package. Administrators will need to manage assignment of these permissions sets for EASY users.

- Application Admin
    - Allows the assignee to create new Application Controls, Requirements, Requirement Items, Questions, Related Object Review, etc.
    - Provides full access to creating/editing applications as if assignee were the applicant
- Application Community
    - Intended for community users who have the persona of an applicant. This allows applicants to access, edit, and create new Applications from the community.
- Application Community Guest
    - This is granted to the community guest user. This permission set allows an unauthenticated user the ability to access the ApplicationRegistration page to create a net new Account, Contact, Application and Community User.
- Application Review
    - Granted to application readers, this will give a reviewer the ability to see Application Reviews as well as a read-only view of the application and its related records.

CUSTOM SETTINGS

The following custom settings need to be configured for the EASY application to work properly:

- Application Setting
    - Community Profile IDs
        - Semi-colon delimited list of profile IDs
        - This custom setting is used in conjunction with Permission Set ID to assign new applicant community users the Application Community permission set. This should include the profile ID that is assigned to new applicant users.
    - Community URL
        - The base URL for the community should be put in this setting
    - Permission Set ID
        - This is the permission set Id for "Application Community"
- Related Object Review
    - The list of API names for objects that used with Related Object Requirement Items
    - One API name is allowed per record
    - The objects in this list that are visible on the Related Object Review Visualforce page

COMMUNITIES

Details on creating/setting up a community for use with EASY are included in the Installation Guide. Once a community is setup and an application is built-out, end users will need to be provided with the application URL. Here's the format:

[COMMUNITY_URL]/ApplicationPortal?at=[APPLICATION_CONTROL_URL_PARAMETER]

- [COMMUNITY_URL] is the URL of the community you have set-up
- [APPLICATION_CONTROL_URL_PARAMETER] is the value of the URL Parameter field on the Application Control record for each application type. If this isn't included in the URL, the Application Control with the "Default" field checked will be used.

When a user navigates to this link, one of two things will happen:

1. If already authenticated/logged-in, the user will be taken to his/her Application Community, where he/she can view/edit/create new applications
2. If not already logged-in, the user will be taken to the community login page. Salesforce provides a default page; however, a custom page can be created for an organization-specific experience. Communities also can be configured to use Single Sign-On in place of this page.
   a. Note: The standard login page has a link that takes users to the ApplicationRegistration page. If a custom login page is used, it will need to have the same type of link so new applicants can start an application (i.e., create an account).

FIELD SETS

- Application Review Header
  - These are the API names of fields on Application_Review__c that readers complete while reviewing an application on the Application Review page.

PROCESS BUILDERS

- External Response Requested
  - Used to send an email to an external user for External Requirements (e.g., recommender for letter of recommendation)
  - When a new Requirement Response is created and the External_Email__c field is populated, this process builder stamps the record with a hashcode and sends out an email using the External Response Requested email template
  - If the Send Reminder flag is toggled on the Requirement Response record, this process builder will send out the a reminder to the requestee using the External Response Reminder email template

EMAIL TEMPLATES

- External Response Requested
  - Sent to the external user when an applicant populates and saves an external requirement item
- External Response Reminder
  - Sent to the external user when an applicant clicks the "Send Reminder" button on an external requirement item

CUSTOM DEVELOPMENT

APEX TRIGGERS

- UserApplicationTrigger on User
  - Calls the logic in UserApplicationTriggerHandler (detailed below)

APEX CLASSES

- ApplicationAdminView
  - Controller for ApplicationAdminView
- ApplicationAdminViewTest
  - Test class for ApplicationAdminView
- ApplicationChecklist
  - Controller for ApplicationChecklist
- ApplicationChecklistTest
  - Test class for ApplicationChecklist
- ApplicationCreate
  - Controller for ApplicationCreate
- ApplicationCreateTest
  - Test class for ApplicationCreate
- ApplicationFilter
  - Utility class used to filter Requirements, Requirement Items, and Questions based on filtering fields (i.e. show requirement based on student type or program)
- ApplicationFilterTest
  - Test class for ApplicationFilter
- ApplicationPortal
  - Controller for ApplicationPortal
- ApplicationPortalTest
  - Test class for ApplicationPortal
- ApplicationQuestion
  - Controls how a question is displayed
  - This class has several methods that control how a Question__c is translated and converted into an object that is usable by the Application components
  - Contains the ability to create a question without a Question__c such as on the ApplicationRegistration page
  - Handles association of Question Responses to Quesetions
- ApplicationQuestionGroup
  - Controls how Requirement Items are displayed

- ○ Takes in a requirement items and either splits out questions, sets up the document upload page, or pulls in all of the relevant data to set up a related object requirement item
  - ○ Handles association of Requirement Responses to Requirement Items
- ApplicationQuestionSection
  - ○ Instantiated with the lists of requirement items, question responses, and requirement responses and creates an instance of ApplicationQuestionGroup for each Requirement Item
- ApplicationRegistration
  - ○ Controller for ApplicationRegistration
- ApplicationRegistrationTest
  - ○ Test class for ApplicationRegistration
- ApplicationRequirement
  - ○ Controller for ApplicationRequirement
  - ○ Queries an application requirement from the "requirementId" URL parameter to determine if the requirement is applicable to the applicant and creates an ApplicationQuestionSection
- ApplicationRequirementPreview
  - ○ Controller for ApplicationRequirementPreview
- ApplicationRequirementTest
  - ○ Test class for ApplicationRequirementPreview
- ApplicationReview
  - ○ Controller for ApplicationReview
- ApplicationReviewTest
  - ○ Test class for ApplicationReview
- ApplicationUtilities
  - ○ Utility methods for use in other classes
- BatchApplicationArchiver
  - ○ Batch class that can be scheduled to flatten question responses into their given requirement responses
  - ○ Stores flattened data into a JSON string on the requirement response
  - ○ This is useful for reclaiming used storage from old applications
- BatchApplicationArchiverTest
  - ○ Test class for BatchApplicationArchiver
- RelatedDocumentUpload
  - ○ Controller for RelatedDocumentUpload
- RelatedDocumentUploadTest
  - ○ Test class for RelatedDocumentUpload

- RelatedObjectReview
  - Controller for RelatedObjectReview
- RelatedObectReviewTest
  - Test class for RelatedObjectReview
- UserApplicationTriggerHandler
  - When a new user is created, this class looks at the list of profile IDs included in the Application Setting custom setting. If the new user's profile is included in this list, this user is assigned the permission set defined in the custom setting.
- UserApplicationTriggerHandlerTest
  - Test class for UserApplicationTriggerHandler

VISUALFORCE PAGES

- ApplicationAdminView
  - Renders an application and related records in a unified view
  - Allows the appropriate users to fulfill Requirement Items, upload supporting documentation, and answer internal-only Questions
- ApplicationChecklist
  - Displays the checklist of Requirements and Requirement Items for an application
  - On load, this page checks if any Requirement Items were fulfilled outside of an applicant saving his/her own application
- ApplicationCreate
  - Allows an existing applicant with a community profile to create a net new application without creating a new Account, Contact, and Community User
- ApplicationPortal
  - Contains a button that takes the applicant to ApplicationCreate
  - Provides access to the Applicant Community and existing application(s)
- ApplicationRegistration
  - Initial page for a new applicant. Asks for basic bio-demo data necessary to create a Salesforce account as well as any Requirement that is marked "Show on Application Creation"
  - On save, this creates a new Community User, creates or matches to an exiting Contact, and creates a new Application record
- ApplicationRequirement
  - Displays a Requirement to the applicant to complete
- ApplicationRequirementPreview
  - Preview page that allows Application Administrators to preview how a Requirement will look to different types of applicants
  - This view is similar to ApplicationRequirement without being tied to a specific

application

- ApplicationReview
  - This is a read-only view of the application for a reader to review a given application, creating an Application Review record tied to the given Application and current user
- RelatedDocumentUpload
  - This allows an Application Administrator to upload supporting documentation on a related object (e.g., official transcripts on education history)
  - Accessed through the related object requirement item on ApplicationAdminView
- RelatedObjectReview
  - Page to "approve" or link related objects to their reference (e.g., approving a net new company that will be linked to an employment history record)
  - On approval, the new record will be created and the related object record will be updated to link to this new record
  - This page also lets users match to existing records

VISUALFORCE COMPONENTS

- ApplicationAddressField
  - Displays address fields
    - Street
    - City
    - State
    - PostalCode
    - Country
  - Detects if State/Country picklist is enabled
- ApplicationBody
  - Contains scripts that can be called from elsewhere such as hiding and showing loading spinners
  - Contains the logo and title for the page
- ApplicationCheckboxField
  - Displays a checkbox
- ApplicationCurrencyField
  - Displays an input box with the currency symbol
- ApplicationDateField
  - Displays an input box with a date picker
- ApplicationDateTimeField
  - Displays an input box with a datetime picker
- ApplicationDocumentUpload
  - Allows for documents to be uploaded to Requirement Responses

- ○ If enabled, renders a rich text box that writes to Requirement Response
- **ApplicationEmailField**
  - ○ Displays an input box that enforces email formatting
- **ApplicationField**
  - ○ Detects which type of field to display based on the ApplicationQuestion class that is passed to it
- **ApplicationForm**
  - ○ Takes an ApplicationQuestionSection and builds a carousel from the Requirement Items
- **ApplicationFormSection**
  - ○ Takes an ApplicationQuestionGroup and renders all of the ApplicationFields, Document Uploads, and Related List Objects
- **ApplicationHead**
  - ○ Component to import all of the nececary scripts and stylesheets for the application pages/components
- **ApplicationLongTextAreaField**
  - ○ Displays an input text box
- **ApplicationMultiPicklistFleld**
  - ○ Displays checkboxes, allowing the user to select multiple values
- **ApplicationNumberField**
  - ○ Displays an input box with numeric enforcement
- **ApplicationPercentField**
  - ○ Displays an input box with a % symbol and numeric enformement
- **ApplicationPhoneField**
  - ○ Displays an input box with a phone symbol
- **ApplicationPicklistField**
  - ○ Displays a select list dropdown
- **ApplicationQuestionHelpText**
  - ○ Included in each of the field components, this will render text underneath the field
- **ApplicationRadioField**
  - ○ Similar to a picklist, this will display all options as radio buttons, allowing the user to select one
- **ApplicationReferenceField**
  - ○ Allows type ahead lookups for reference fields on related object requirement items (i.e. school on enrollment history)
- **ApplicationStaticField**
  - ○ Displays static text on the page
  - ○ Accepts basic html markup

- ApplicationTextAreaField
    - Displays an input text box
- ApplicationTextEncryptedField
    - Displays an input box that is masked
- ApplicationTextField
    - Displays a basic input box
- ApplicationUrlField
    - Displays an input box

STATIC RESOURCES

- Bootstrap
    - Bootstrap css and javascript for application visuals and responsive design
- Ckeditor
    - Rich Text markup editor
    - This is used to get around errors that prevents rich text areas from being rerendered
- Jqueryui
    - Visual library for applications

OTHER CONSIDERATIONS

REPORTING

Because EASY allows application administrators to create questions without having to engage a system administrator to create a field, applicants' responses to questions are stored as records in the Question Response object. This makes reporting on the values of those responses difficult. Because of this, we recommend using a field on the Application object in conjunction with the Application Field attribute of the Question object for any questions requiring response reporting.

Additionally, the Question Response object includes several fields to track whether or not applicants have completed certain questions. By utilizing roll-up summary fields in conjunction with the IsComplete, IsPopulated, IsRequired, and IsShown fields on Question Response, you can report on application completion percentages.

DATA STORAGE

Because there is a Question Response record for each Application-Question pair, this object can use a lot of data. EASY provides two mechanisms for managing this data storage:

1. Application Field attribute of Question. This allows you to store answers to questions directly in a field on the Application object. Please know: Using this feature means a System Administrator will have to create a field for each Question. For that reason, we recommend only using this for questions common to all/most applications and data needed for reporting.
2. BatchApplicationArchiver Apex Class. This is a Batch Apex class that can be used to archive Question Response, Requirement Response, and Application Review records into an attached .txt file in JSON format called ApplicationBackup.txt on the Application record. By default, this class will archive any applications that were created in the last calendar year; however, a developer could modify the criteria and even schedule the batch class to run on a repetitive basis.

HIGHER-EDUCATION DATA ARCHITECHTURE (HEDA) MAPPING

EASY is compatible with both HEDA and non-HEDA Salesforce implementations. This means that organizations that use HEDA may have to map EASY objects to HEDA objects. More specifically, EASY uses custom objects for Program and Academic Term, which is separate from HEDA's use of Academic Program Accounts and the HEDA Term object. We recommend using the HEDA objects for official SIS records while leveraging the EASY objects for application-specific records. You can then map the application-specific records to their SIS counterparts in HEDA.

INTERACTIONS FOR STUDENT RECRUITMENT

EASY includes a modified version of University of Miami's open-source solution called Interactions for Student Recruitment. Specifically, any references to HEDA objects in the codebase have been abstracted to first test for the existence of those objects before executing the code. Note: The Plan__c object has been renamed to Program__c in EASY. If you're installing EASY after first installing/implementing Interactions, this may mean you need to migrate all Plan__c records over to the Program__c object.

Much more information, including the Documentation Guides that explain the use and benefit of Interactions, can be found on the Salesforce Foundation GIT Respository.