CRM is the backbone of any organization. It manages all your business data in one place, like no other tool. It is one that keeps the organization in perfect shape; unless you bend it the wrong way. But there are those minor, yet important tasks that you could get off your back by stretching your CRM a bit. What CRM needs, is a shot in the arm; custom apps that bridge gaps. These custom applications extend the capabilities of CRM, while managing to keep it in one piece.

Assume a Sales and Marketing team that conducts campaigns, closes sales, tracks expenses and revenue, generates commission reports of agents and a lot more. All the data required to perform these tasks will be available in the CRM software they use. But a CRM software is not meant for calculating commission reports based on sales or handling travel and expense requests, or even keeping track of campaign spendings. That is when you need an application built for that specific purpose. What does that leave you with? High spending on hiring developers, maintaining and upgrading of software and hardware resources. Plus, you'll end up having two copies of the same data, because the CRM and your custom application are possibly from two different vendors and would not get on with each other all that well. Mess?

There is a solution.
The best side of custom app-building, is the other side, where you get to build your custom applications all by yourself, in minutes. That is where Zoho Creator comes in. Zoho Creator offers an easy-to-use application-builder that lets you put together, just the application you need. To make it work the way you want, you just need to drag and drop the required elements, specify what tasks the modules should perform, and when. The intuitive drag-&-drop interface is a boon to those without any prior experience in app-building. In addition to all this, workflow and business rules too can be specified the same drag-&-drop way. These customized workflow adds power and flexibility to your custom application, making it ideal for all your business needs.

1. NO TECHNICAL EXPERTISE NEEDED
With Zoho Creator, just about anyone can build custom applications for their unique needs. It demands neither technical expertise, nor previous experience with programming languages.

2. DRAG-DROP INTERFACE
Without having to depend on a developer, you can have the pride of building custom applications all by yourself. Regardless of your expertise and experience.

3. FASTER DEPLOYMENT
Unlike in conventional software applications, there are no maintenance downtime involved. You don’t have to install patches or updates manually. We do updates so that all users are on the latest version, without experiencing any outages. Plus, any modification you make to your application is resulted instantly, real-time.

4. SCALABLE
You’d like your business to grow. And when it does, your software application too has to meet new demands. With Zoho CRM, you can add new modules to expand your application as your business grows, without disrupting access to your application.
Who can activate custom app-building for Zoho CRM?
1. You need to be an Administrator of your Zoho CRM account.
2. By default, it will be disabled on your Zoho CRM account. To enable it, You’ll need an Enterprise Edition of Zoho CRM (Available to Enterprise trial users too)

Activating Custom App-Builder
1. Access the Setup page. (You will find the link in the upper right corner)
2. In “Zoho Apps” category, select “Zoho Creator”
3. Click on “Activate Now” button.
Once you activate the custom application builder, you can get started right away. Create your first application from scratch by clicking on “Create” button. This will take you to the drag-drop app-builder.

Or, if you already have created applications on the associated Zoho Creator account, you can import a form from those applications into a custom tab.

**NAMING**

When you click on “Create”, a dialog box appears, prompting for a name for your first form and the application in which it should be created. You also want to display this form on a custom CRM tab, so specify a preferred name for the tab too.

**BUILDING FORMS**

Zoho Creator supports 22 field (data) types. Depending upon what your application should do, you can include fields simply by dragging and dropping them on the builder. Specify the field properties if needed, and your good to enter data into your application.

There are some basic fields which are commonly used, like text, number and email address fields. Then there are some advanced fields like the file upload and CRM fields. Here is more about the advanced fields:
FIELDS (ADVANCED)

1. FORMULA
Formula field automatically calculates a value based on the input of other fields. Comes in handy on applications such as commission calculation and other monetary applications.

2. LOOKUP
“Lookup” fields are for creating a relationship between two forms. The look-up field on one form can fetch the data entered into a field on another form.

3. SUBFORM
Users will be able to add data into the child form just by accessing the parent form. The entire child form will act like a single field on the parent form.

4. FILE UPLOAD
This field allows users to attach documents and files supporting the data they enter into forms. These files will be listed alongside the records on your custom tab, and can be downloaded whenever needed.

4. ZOHO CRM MODULES
In Zoho CRM, your data will be categorized into several modules; Contacts, Leads, Potentials, Vendors, Users, Invoices and more. Using the Zoho CRM field, you can fetch data of any module from Zoho CRM, into your custom application. The value of a Zoho CRM module can be one of the fields on your application.
Custom Workflow and Business Rules are essential in adding robustness and intelligence to your applications. To let you do that yourself, we have Workflow which ranges from simple if-else scenarios to user-role based sharing and approval modules. These too can be inserted, just by dragging and dropping.

The drag-&-drop script builder is powered by our very own scripting language called DELUGE. Best thing about it is that you need not be a programmer to build the right custom application for you.

**CRM TASKS:**

The 6 tasks explained here solely operate on your Zoho CRM data. CRM tasks are code-snippets that perform specific actions on your CRM, whenever they are executed in your custom application. Using these, you can create, fetch, search for, and update records on any CRM module, right from within your custom application.

**Create a record in Zoho CRM:**

When your users add data into the custom application you built, the data can be viewed only within the custom tabs. But if you want a copy of the data to be available in the CRM modules too, you need to execute this task whenever a user enters data into the custom application.

Code-format:

```
<CRM Response> = zoho.crm.create(<CRM Module Name>,
<Field values as map value>,
<optional duplicateCheck as Integer>);
```
**Fetch records from CRM modules**

When you want to use data from multiple CRM modules on your custom application, you can fetch them using this task. For instance, you can fetch records from Leads module, and the list of items from Products module, and update them both in your custom application.

Code Format:

```javascript
<list-variable> = zoho.crm.getRecords("" 
<ModuleName>"",<fromIndex>,<toIndex>); 
```

---

**Search Records in CRM**

This method enables you to search for records in CRM by specified criteria. You can drill deep down exactly to the data you need, by specifying multiple criteria. Filter data by searching for part of, or the exact field value, or by specifying the expression to exclude while searching. If you know the record ID to begin searching from, then you are even closer to finding your data.

Code Format:

```javascript
<variable>= zoho.crm.eachRecords("<moduleName>"", 
"<criteria>", fromIndex, toIndex); 
```

---

**Search records in CRM by PDC**

PDC stands for “Predefined Column Names”. This method is useful if you know exactly what you want to find. For instance, if you want to find the account information of a particular vendor, this method lets you extract that record.

Code Format:

```javascript
zoho.crm.searchRecordsByPDC(<moduleName>, 
 criteriacolumnName>,<criteriaColumnValue>); 
```
**Update record in CRM for a given record ID**

Every record in Zoho CRM has a unique ID. By specifying the ID of an existing record, you can update its value on Zoho CRM. To do that, you enter the new values in your custom application, and specify the ID of the CRM record that has to be updated with these values. As you submit, these new values will be updated in the record of the specified ID.

Code Format:

```
<map-variable> = zoho.crm.updateRecord(<CRM Module Name>, <ID>, <New values as map object>);
```

---

**Get related records**

If you know the record ID of a record in one module, and you want all the information from another module which are associated with this record ID, this is the task that you’ve got to use. For example, if you know the ID of a Potential, you can fetch all the notes associated with this potential.

Code Format:

```
zoho.crm.getRelatedRecords(<ModuleName>, <PrentModuleName>, <RecordID>, fromIndex, toIndex);
```
CUSTOM ACTIONS

You can configure more than 40 tasks to be performed at different stages of your application. For instance, When new records are submitted into a form, when a form is being loaded, when a user enters data into a field and so on. Two broad classification of these custom actions are,

1. **FORM ACTIONS**  
2. **FIELD ACTIONS**

**FORM ACTIONS**

Any task that needs to be performed at the form level, has to be specified within these cases. For example, the email address field in a form can be automatically filled with the email address of the person accessing it. All these three scenarios are form-specific, and are carried out whenever an action is done on a form (Adding, editing or deleting a record in a form.)

**On Add:**
Tasks defined within this case are executed when a user adds data into a form. As and when they click on the submit button, these tasks are carried out.

**On Edit:**
When editing a record, tasks within case are executed. For instance, in a travel request form, when a user edits his request, the approval modules can be hidden.

**On Delete:**
Tasks in this case are performed upon deletion of a record. The owner of the application might want to be notified whenever a record is deleted by users. Then he can configure an automatic email notification within the On Delete case.

| **ON ADD** | Set Date Field's default value to the Current Date |
| **ON EDIT** | Enable Approval module to the Admin only |
| **ON DELETE** | Notify Admin when a record is deleted by the Users |
FIELD ACTIONS

The other scenario is performing custom actions whenever an action is done on a field. For example, based on which of the two options he selects on a radio button, the next field can be displayed or remain hidden.

On User Input:
When a user enters data into a field, it can be analyzed and based on predefined conditions, the specified custom tasks can be carried out.

On Update:
The value of a field can be set to any default value, initially. And, it can be modified later, or updated to a new value. When it is updated, set of tasks defined under this case can be performed. For example, when a travel request is raised by an employee, the approval status would initially be “Pending”. But once the manager makes a decision, the value is changed to either “Approved” or “Rejected”. In either case, the value has been updated, and the requester needs to be notified via an email notification.
CRITERIA AND VARIABLES

Whenever you are checking a condition, you need to define some criteria based on which the condition can be checked. Criteria is a condition, based on which tasks are executed. To define a criteria, you just select the field whose value you want to check, specify what value should be checked against it, and the operator to compare the two.

For example, if you want to let users see only those data which they have entered into the database, then the criteria you set is:

```
input.Added_User == zoho.loginuser
```

CONDITIONAL STATEMENTS

Conditional statements mostly are the if-else statements. You define a condition, and two sets of tasks; one to be executed if the condition holds good, and the other if the condition fails. The second set of actions is optional.

For example, you check if it is Christmas Eve today. If it is, then you trigger a Christmas greeting to all your contacts. If it isn’t Christmas Eve, then you do nothing. This condition is checked everyday, but holds good only on one day of the year, and the email is sent on Christmas Eve only.

```
if (today is Christmas Eve)
{
 Send greeting email to all users;
}
else if (today isn’t Christmas)
{
 do nothing. Check again tomorrow;
}
```

HIDE/SHOW & ENABLE/DISABLE FIELDS

At times, some fields might have to be hidden to users. As in a questionnaire, based on the option users pick to the gender question, you can display two different sets of questions. Another instance is, approval modules, which should only be visible to managers.

To prevent users from modifying default values of a field, it should be disabled. For example, in an event registration form, the user should be able to see the date, but not alter it.
EMAIL NOTIFICATION

Custom tabs can be shared to any number of users within an organization. To view the data, you need not access the custom tab every time. Stay updated on what data is entered into the forms, by configuring email notifications. Whenever a user submits data, you will instantly receive it as a notification email, in your preferred inbox. It can be sent to multiple recipients too.

CUSTOM ACTION BUTTON

When clicked on, the custom action button performs multiple actions that you specify. Just like the add and delete buttons, this too is displayed on the View. Select the records that you want to operate on, and click on this button.

If you want to thank customers who have made this month’s payment, define a custom button to trigger an email. You then select the records, and click on this button. The email will be sent to all those contacts you selected.

FETCH RECORDS FROM ONE FORM AND DISPLAY IN ANOTHER

If you have multiple forms to manage your customer contacts, their purchases and payments, not every form needs to have all the fields. The contact form alone can have the name and email address. The other forms can fetch data from the contact form, and display it on their View. By this, there will be no duplication of data in a database.

DYNAMIC PICKLIST

Of two picklists, one lists categories, and another lists subcategories. When a category is picked, only those subcategories of that main category should be displayed in the second.

The first picklist has list of countries, and you select U.S. Then only those states in the U.S. should be listed in the second picklist.

If the user cannot find his country or state, he should be able to add it to the list first, and then select it. New values can be dynamically added by clicking on the + button beside the picklist.
Schedules are the automatic execution of those redundant tasks, whenever they should be, with the set of data that it needs to be executed upon. To configure schedules, you just need to tell it what to do, and when to do. It will be periodically executed, without your manual intervention.

**FORM SCHEDULES:**
Form Schedules depend on the date field on your form. They are executed based on what its value is. The condition has a date, which is compared with the values of the date field on every record of a form. Whenever the condition is met, the defined tasks are executed. The same can be executed if the form has a date-time field instead of the date field.

**REPORT SCHEDULES:**
Report Schedules are based on a View of your application. When a configured report schedule for a particular View is executed, an email will be triggered to all the specified email addresses. This email will have the reports of all the data stored in that View. With Report schedule, no custom actions can be defined. It is exclusive for emailing reports.
**CUSTOM SCHEDULES:**

Custom Schedules is immensely flexible and powerful, because you can perform any validation using scripts, on any form or view. In short, you can do anything you would normally be able to do with the script-builder. All user-defined actions fall under custom schedules.

---

**Sharing your App**

This is the most important part of data collection. You need to share your forms with your users so that they can enter information into them. Here, as you are creating your form on Zoho CRM and using a custom tab to display it, you have to share the CRM tab with your users.

When you complete building the application, the next step would be setting “Custom Tab accessibility”. All the roles of users on your Zoho CRM account will be listed on this screen, of which the administrator and developer (if any) will be selected. Administrators and developers of an application will have access to that custom CRM tab by default. Check the box of other roles you wish to share this tab with.

---

**Developer Access**

Zoho CRM provides the flexibility of assigning applications to a developer. This developer will be able to create new apps, as well as edit the ones you have created. That way, he can help you in building the application. The application can be reassigned to another developer at anytime.
Congratulations!
You’ve learnt enough.

Now go build Custom Apps on your own.

Go!

For questions on Custom App Building, please contact:
Cecil Adrian, Customer Relations.
cecil.adrian@zohocorp.com