CustomReports

CiviCRM Extension
for
New York New Jersey
Trails Conference

Extension Version 1.0
Document Revision 1.0-2017-08-06

1. The CustomReports Extension

CustomReports is a native extension for CiviCRM allowing for the generation of printable PDF letters using specialized reports as a data source. It was produced by CrusonWeb (info@crusonweb.com), with original code created by Steve Binkowski (steve@binkcms.com).

The extension leverages CiviCRM's native search interfaces to select objects such as contributions or memberships. These objects may be passed to the extension's letter generation interface. Once a letter is selected, the letter generation classes will call custom reports to populate a tokenized Smarty HTML templates, written in standard HTML and CSS. The template is finally converted into a printable PDF letter through the dompdf vendor package.

CustomReports was created with CiviCRM's civix tool to ensure the use of standard CiviCRM architecture and boilerplate. It extends CiviCRM's native classes to provide smooth integration and a familiar user experience.

This extension includes:

- Extension information and installation files, produced by civix,
- New custom CiviCRM reports to act as the token data source for letters,
- Letter generation interfaces, extended from native CiviCRM forms,
- Classes extending native CiviCRM form tasks to generate PDF letters,
- Smarty template files used to produce HTML letters, to be converted to PDFs.

2. Installation

Extension installation follows the normal CiviCRM process. The first requirement is to locate the CiviCRM extension directory. In the admin panel, select [Administer | System Settings | Directories] from the menu bar. The “Extensions Directory” setting defines the desired path. In a standard Drupal-based installation, this should be /sites/default/files/civicrm/ext.

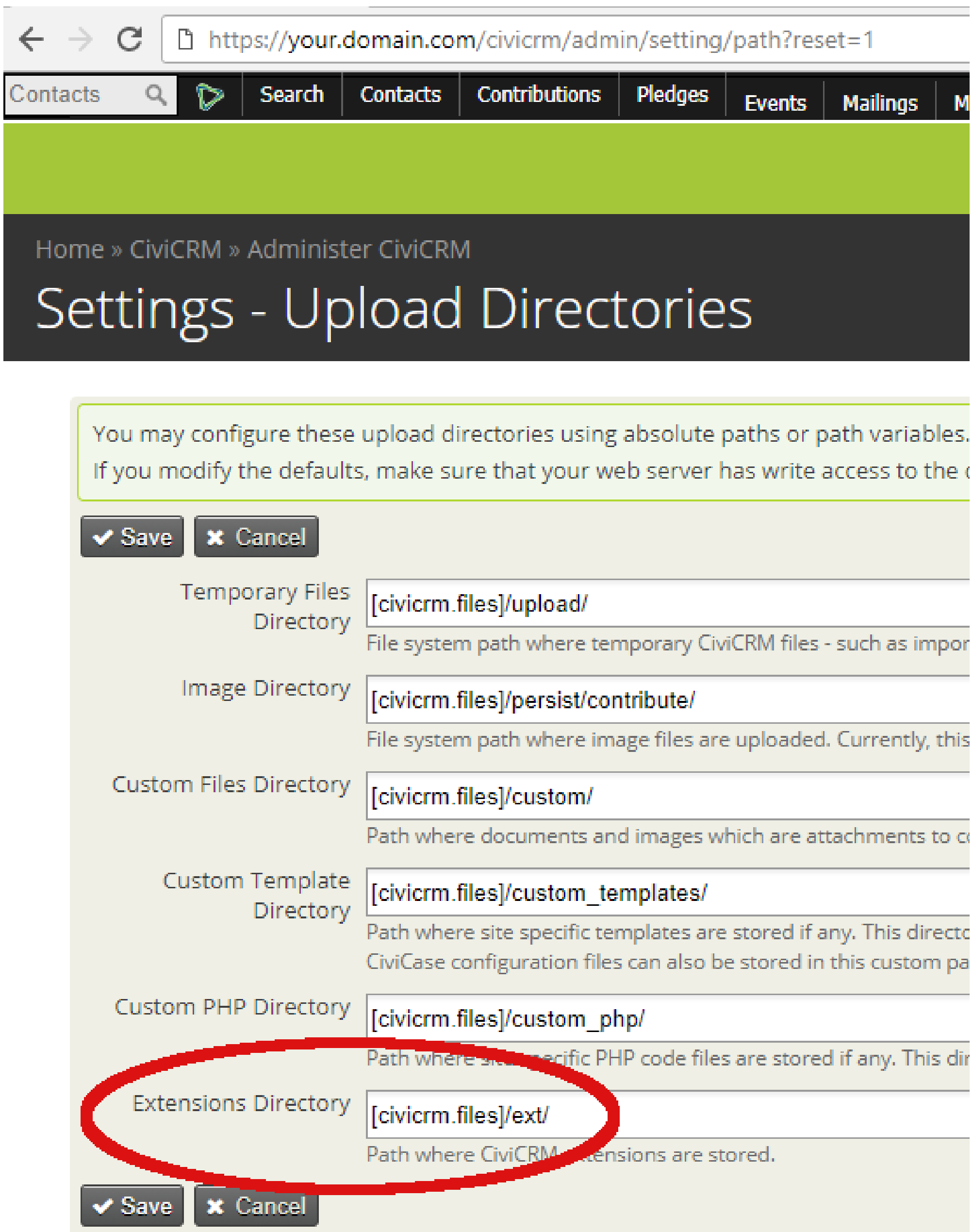

Image 2.1: Finding the extension directory in CiviCRM's admin panel

Once the directory is located, the steps below describe the installation process:

1. Unzip/untar the extension's archive file into the extension directory. This will create a subdirectory named “com.crusonweb.nynjtc.customreports”.
2. From the menu bar, select [Administer | System Settings | Extensions], or browse to /civicrm/admin/extensions?reset=1. Find the extension titled “**NYNJTC Custom Reports**” and enable it. If it is not present in the list, click the “Refresh” button at the top of the page.

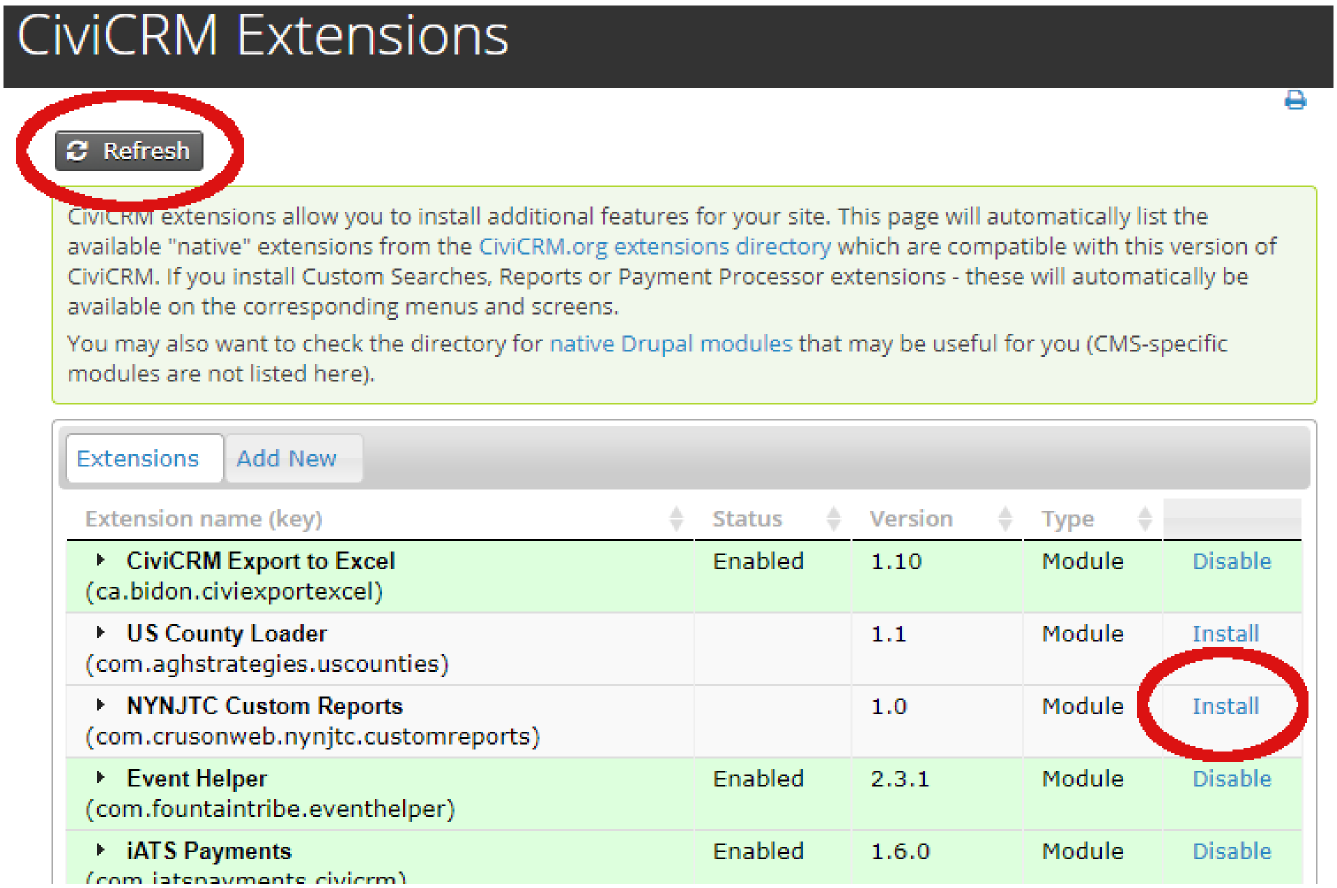

Image 2.2: Installing the extension into CiviCRM

CiviCRM will ask for confirmation. Once confirmed, the list of extensions should show it as “Installed”. Proper installation can be verified by checking custom reports (see Section 3, Reports) or by checking the tasks available from an object search (such as contributions, see Section 4, Letters).

From this same interface, the extension may be disabled to remove its functionality. Disabling the extension will prevent it from being available, but will leave the extension's files on the server. If, after disabling, the extension is uninstalled, it will be removed from the server entirely.

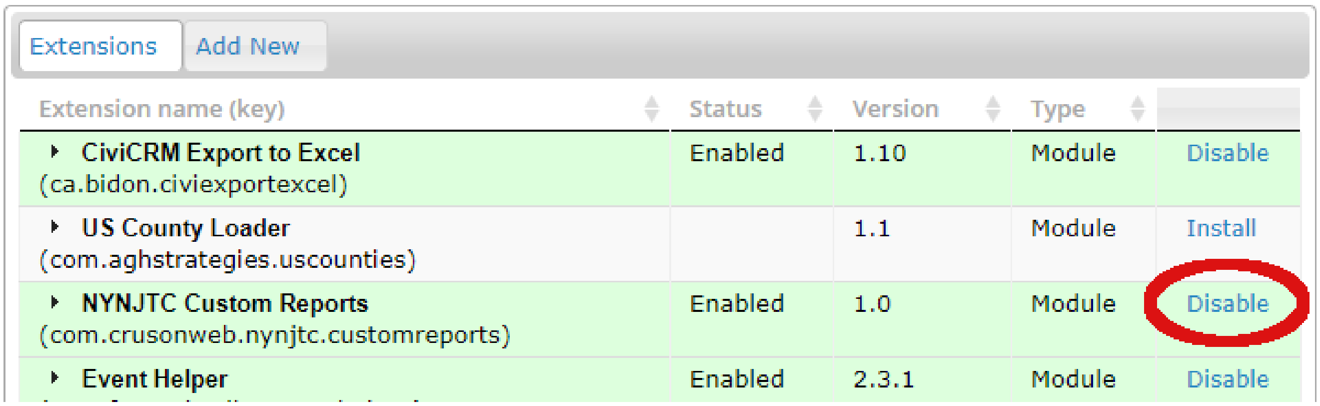

Image 2.3: Disabling the extension. After disabling, "Uninstall" will appear here.

IMPORTANT! If new custom reports are added to the extension after installation, it will need to be disabled and re-enabled in order for CiviCRM to discover the new reports.

3. Reports

CustomReports installs with a number of custom CiviCRM reports. These reports are created by extending CiviCRM's `CRM_Report_Form` class. They are accompanied by `MGD` files to support auto-discovery on installation. Each report is tailored to provide specific information relevant to the purpose of the report.

The reports were designed specifically to act as data sources for the PDF letters which use them. They contain a large number of fields, making them inappropriate for normal, stand-alone usage. As such, it is not recommended to use them as one would use the standard native CiviCRM reports. No report instances using these custom reports are installed or configured by default, though they can still be created through CiviCRM's report interface. If the reports are used as stand-alone data sources, it is recommended to export the resulting data to an application better suited to manipulating the data, such as a spreadsheet application (e.g., Excel), or other custom databases.

It is also important to note that the reports do not natively contain all information relevant to their respective letters. The complexity of joining multiple objects with different contexts is a significant obstacle not easily accomplished using CiviCRM's report structure. For example, contribution detail reports normally contain contact information for the contact donating the contribution. Soft credit contribution reports will replace that contact information with that of the contact being soft credited. The CustomReports version of the soft credit report follows this strategy, but will additionally load the donating contact's information programmatically. The final letter has access to both sets of information, even though the report only has access to the soft credit set.

Because no instances of the reports have been configured, they will normally not be available in CiviCRM's report interfaces. They can be accessed as stand-alone reports by selecting “New Report” from CiviCRM's “Reports” admin page.

A. List of Reports

Reports included in this extension are shown below, with a brief description of their purpose and the information they include. All reports will load any custom fields associated with any objects known to the report (i.e., contacts, contributions, memberships, etc.)

1. **Full Contribution Detail**: for standard contributions, with no consideration of soft credit status. Available fields include contact information (incl. address, email, phone), contribution details (incl. Notes, and details concerning the financial type, tax status, and the associated transaction), and product/premium details.
2. **Soft Credit Contribution Detail**: for soft credit contributions, in which a third party

aggregates contributions from multiple primary contacts. Available fields include contact information of the original donating contact, contact information of the aggregating contact, and contribution and product/premium details. As with the full contribution report, secondary contact (phone, email, etc.) and contribution (tax, transaction, etc.) fields are included.

3. **Pledge Contribution Detail**: for contributions spread into payments through a pledge. This report contains all the same fields as the full contribution detail, and additionally includes details of the pledge (total pledge amount, number of scheduled payments, etc.) and the payment history (total count and value of previous payments).

4. **Tribute Contribution Detail**: for soft credit contributions specifically associated with "In Memory Of" or "In Honor Of" types of contributions. This report contains the same information as the Soft Credit Contribution Detail, except:

   a) it natively contains only the contact information of the primary donor, and

   b) it includes only soft contributions with a type of "In Memory Of" or "In Honor Of".

5. **Full Membership Detail**: for memberships. This report contains all information of the member's contact record, and the contact record of any dependent members. For memberships with associated contributions, each record will contain all contribution data for <u>the most recent contribution only</u>. The contribution must be of financial type "Member Dues" to be selected. Dependent members are discovered by way of the "Provides Membership To" relationship.

4. Letters

CustomReports also provides a number of printable PDF letters. Like reports, letters are extensions of CiviCRM's native classes (the `CRM_*_Form_Task` family of classes). Each letter is preformatted with a specific template.

There are two categories of letters – contributions and memberships – and each is available as a custom task on their respective search result pages. CiviCRM's native search interface can filter on every attribute of an object, including custom fields, and filtered results can be further focused through manual selection.

A. Generating Letters

The first step is to execute a search for the target objects. Searches can be started by selecting the “Find” option under a menu branch, as shown below.

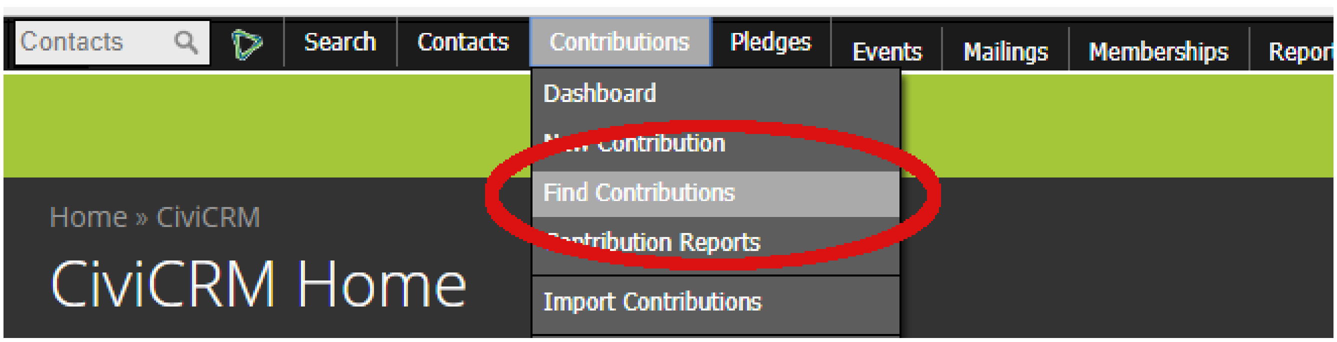

Image 4.1: "Find Contributions" on the menu bar. For a membership search, look under "Memberships"

The search interface will allow for finding objects based on specific criteria. Some possibilities include searching by name, email address, date of contribution, financial type, transaction batch, or any custom fields. After entering criteria and clicking “Search”, a list of results will be displayed. On the search results, selecting at least one object will allow for selecting a task. In the task drop-down, select “Custom Contribution Reports” to visit the CustomReports landing page.

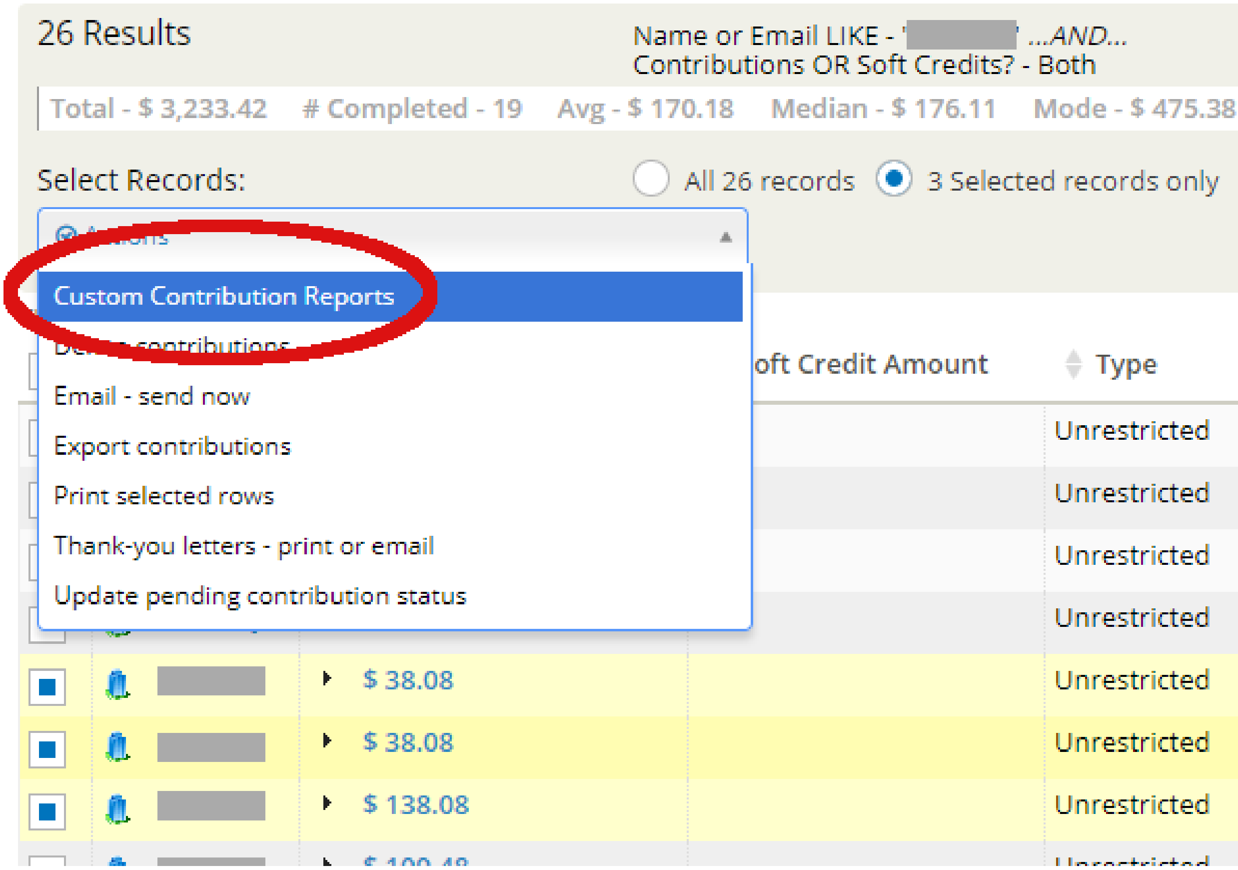

Image 4.2: Search results, with the CustomReports task selected in the drop-down.

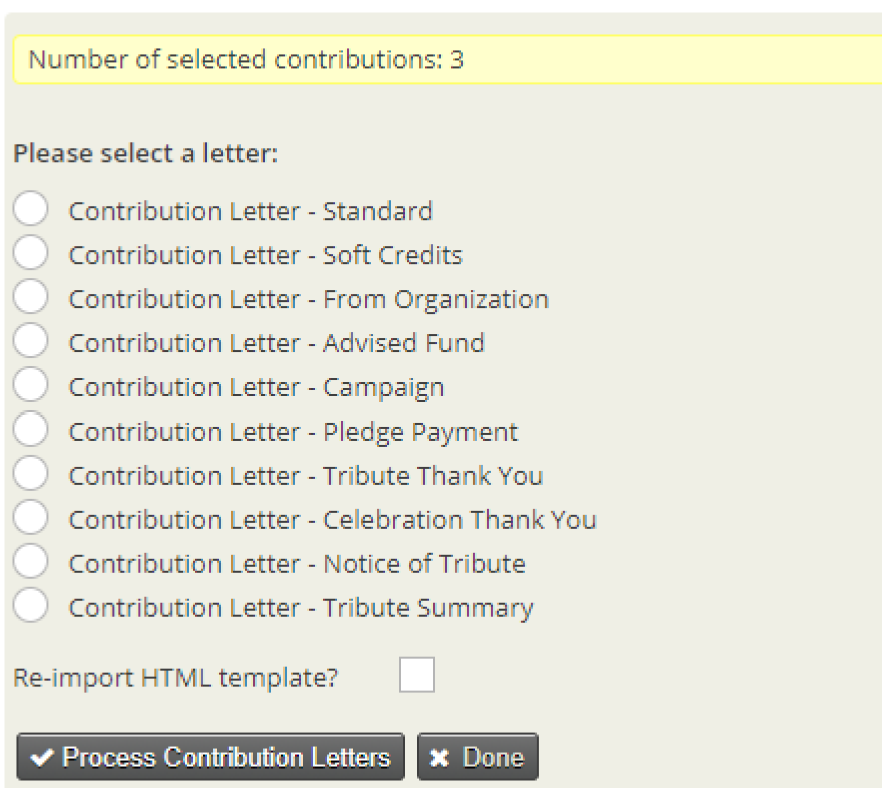

Image 4.3: The CustomReports landing page. Note the number of contributions passed to the letter is displayed at the top

IMPORTANT! The search page filter item “Contributions OR Soft Credits?” must be set appropriately for the desired type of letter. For example, searching for soft credit donations for a “Standard” letter, or for only hard credit donations for a “Soft Credit” letter, could cause those letters to not provide the desired results. If the letters do not contain the correct contact information, double-check the search filter.

On the landing page, the desired letter type must be selected. The same contribution can be printed in a variety of contexts, but only one at a time. If multiple letters must be printed for the same contribution (e.g., a thank-you to an organization, as well as the individual donors), each type of letter must be printed separately.

All templates (see section 5) are stored as files on the server, but they are cached in the database after the first use. If any changes have been made to the template files, selecting the “Re-import HTML template” checkbox at the bottom of the form will force a fresh read of the file, and refresh the database cache.

After selecting a letter, clicking the “Process” button will generate the PDF letters and make them available as a file. All generated letters are printed into one file, regardless of the number of pages. The user's browser should automatically save the file, or ask the user where to save it.

For contributions, the “Thank You Date” field is set to the current timestamp as it is printed in the PDF. The timestamp is only set if it is blank, so the field will track only the first “Thank You” letter sent. Note that the date field is an attribute of the contribution itself, not the associated soft credit donors.

B. List of Letters

There are ten contribution letters and two membership letters. Each type of letter will use one of the reports describe in section 3 as its data source, and will have a template file to define the content. The template file uses Smarty template system, and the letter will generate sets of tokens to be made available during template processing. Normally, a letter will generate token variables for the component (either `$contribution` or `$membership`) and the primary contact (`$contact`). It will be noted in the description if a letter defines additional token variables, or assigns its data differently than the norm.

1. **Standard**: Based on report “**Full Contribution Detail**”, uses template “**Standard**”. For standard direct contributions, usually from an individual. The donor is `$contact`.
2. **Soft Credits**: Based on report “**Soft Credit Contribution Detail**”, uses template “**ThroughOrg**”. For donations made from a soft credit donor through another contact, such as an organization. The soft credit donor is `$contact`.

3. **From Organization**: Based on report "**Full Contribution Detail**", uses template "**FromOrg**". For direct contributions from an organization contact. This letter loads "point of contact" information separately from the report data. The organization is `$contact`. The point of contact for the organization, if available, is `$related`.
4. **Advised Fund**: Based on report "**Soft Credit Contribution Detail**", uses template "**AdvisedFund**". For direct contributions which are aggregated by an unrelated contact. The donor is `$contact`.
5. **Campaign**: Based on report "**Full Contribution Detail**", uses template "**Campaign**". For donation coming through campaign pages. The donor is `$contact`.
6. **Pledge Payment**: Based on report "**Pledge Contribution Detail**", uses template "**Pledge**". For donations made as part of a scheduled pledge. The donor is `$contact`.
7. **Tribute Thank You**: Based on report "**Tribute Contribution Detail**", uses template "**Tribute**". For donations made in memory or honor of another (usually deceased) contact. The donor is `$giver`, the honored contact is `$beneficiary`.
8. **Celebration Thank You**: Based on report "**Tribute Contribution Detail**", uses template "**Celebration**". For donations made in memory or honor of another (usually alive) contact. The donor is `$giver`, the honored contact is `$beneficiary`.
9. **Notice of Tribute**: Based on report "**Soft Credit Contribution Detail**", uses template "**TributeNotice**". For tribute donations, this letter is sent to the deceased's point of contact. This letter loads "point of contact" information separately from the report data. The donor is `$contact`, the point of contact is `$related`.
10. **Tribute Summary**: Based on report "**Soft Credit Contribution Detail**", uses template "**TributeSummary**". For tribute donations, this letter sends a recap of tribute donations made over a period of time. This letter loads "point of contact" information separately from the report data. The point of contact is `$contact`, the honored contact is `$beneficiary`.
11. **New/Renew/Return**: Based on report "**Full Membership Detail**", uses template "**General**". For all memberships. The primary member is `$contact`. If a secondary member exists, it is `$second_contact`.
12. **Replacement Card**: Based on report "**Full Membership Detail**", uses template "**Replacement**". Replacement card for any membership. The primary member is `$contact`. If a secondary member exists, it is `$second_contact`.

5. Templates

Every letter in CustomReports uses a Smarty template file as the content of the PDF it generates. These template files are found in the `templates` directory of the extension. Templates may be edited with any normal text editor, such as vim, emacs, or notepad.

A. Editing Templates

While templates consist mostly of HTML, they are not full HTML pages. Templates are more like an excerpt from inside an HTML document. They contain no head/body/html elements; all content is enclosed in a single parent div.

> **IMPORTANT!** Even though CiviCRM has facilities to edit templates in the admin panel, doing so will very likely break the template. This is due to the editor used by CiviCRM translating certain characters within Smarty control statements into HTML entity strings. The result causes errors in the template parsing system, effectively breaking the template. If a template is to be edited, edit the file directly on the server, or edit a copy and replace the version on the server. Use the “Re-import HTML template” option (see Image 4.3) when printing letters to refresh the template cached within CiviCRM's database.

When editing templates, any standard HTML4 and CSS2 may be used. Scripting should ideally be removed, but should always be kept to a minimum. See section 6 for further details on how the different PDF generation libraries handle javascript and CSS.

B. Template Assets

The HTML templates require some assets to render properly. These are stored in the `assets` subdirectory under the extension. Assets currently available are:

- default-style.css, an external style sheet. This file is injected into the HTML as it is sent to the PDF generator. Any necessary changes to the overall style of all letter should be put here.
- goodell-signature-200x43.png, an image file depicting the signature to insert into the letters. The HTML required for this image is generated by the letter, and is injected into the templates via the `electronic_signature` property of the component variable.
- Interstate-Regular.ttf, a font file used for every template. It is referenced in default-style.css.

C. Tokens

Every template file has several token variables available to it. For all letters, a component variable is created to hold the primary object of the letter - `$contribution` for contributions and `$membership` for memberships. These variables are arrays of all the available fields for that component, including custom field names. For example:

> `$contribution.receive_date` = the date a contribution was received
> `$membership.receive_date` = the date a membership started

Generally, each element name is the same as its database field. Custom fields elements are referenced by the CiviCRM field number. Refer to custom fields in the admin panel to review how they are named.

Additionally, most templates receive a `$contact` variable. Also an array, it holds all the fields available for the contact being presented, including base information, address fields, email, phone numbers, and any custom fields. Some letters rename this variable, and other will have additional contact variables available. Refer to section 4B for details on what each letter provides to its template.

All contact data points are the standard CiviCRM tokens generated with internal letters. Refer to CiviCRM documentation for a list of available tokens.

i. Component Tokens for Full Contribution Detail

contribution_id	cancel_reason	creditnote_id
financial_type_id	receipt_date	tax_amount
contribution_page_id	thankyou_date	revenue_recognition_date
payment_instrument_id	source	product_name
receive_date	amount_level	product_description
non_deductible_amount	contribution_recur_id	product_price
total_amount	is_test	financial_type_name
fee_amount	is_pay_later	financial_type_description
net_amount	contribution_status_id	is_deductible
transaction_id	address_id	is_reserved
invoice_id	check_number	is_active
currency	campaign_id	contribution_note
cancel_date		

ii. Component Tokens for Soft Credit Contribution Detail

soft_amount
soft_credit_type_name
contribution_id
financial_type_id
contribution_page_id
payment_instrument_id
receive_date
non_deductible_amount
total_amount
fee_amount
net_amount
transaction_id
invoice_id
currency
cancel_date
cancel_reason
receipt_date
thankyou_date
source
amount_level
contribution_recur_id
is_test
is_pay_later
contribution_status_id
address_id
check_number
campaign_id
creditnote_id
tax_amount
revenue_recognition_date
product_name
product_description
financial_type_name
financial_type_description
is_deductible
is_reserved
is_active
contribution_note

iii. Component Tokens for Pledge Contribution Detail

contribution_id
financial_type_id
contribution_page_id
payment_instrument_id
receive_date
non_deductible_amount
total_amount
fee_amount
net_amount
transaction_id
invoice_id
currency
cancel_date
cancel_reason
receipt_date
thankyou_date
source
amount_level
contribution_recur_id
is_test
is_pay_later
contribution_status_id
address_id
check_number
campaign_id
creditnote_id
tax_amount
revenue_recognition_date
product_name
product_description
product_price
financial_type_name
financial_type_description
is_deductible
is_reserved
is_active
contribution_note
payment_id
pledge_id
scheduled_amount
actual_amount
scheduled_date
payment_id
pledge_financial_type_id
pledge_page_id
pledge_full_amount
pledge_install_amount
pledge_frequency
pledge_interval
pledge_installments
past_payment_total
past_payment_count

iv. Component Tokens for Tribute Contribution Detail

`credit_id`	`invoice_id`	`campaign_id`
`contribution_id`	`currency`	`creditnote_id`
`beneficiary_id`	`cancel_date`	`tax_amount`
`soft_amount`	`cancel_reason`	`revenue_recognition_date`
`soft_credit_type_id`	`receipt_date`	`contribution_id`
`giver_id`	`thankyou_date`	`product_name`
`financial_type_id`	`source`	`product_description`
`contribution_page_id`	`amount_level`	`product_price`
`payment_instrument_id`	`contribution_recur_id`	`financial_type_name`
`receive_date`	`is_test`	`financial_type_description`
`non_deductible_amount`	`is_pay_later`	`is_deductible`
`total_amount`	`contribution_status_id`	`is_reserved`
`fee_amount`	`address_id`	`is_active`
`net_amount`	`check_number`	`contribution_note`
`transaction_id`		

v. Component Tokens for Full Membership Detail

`membership_id`	`join_date`	`minimum_fee`
`membership_type_id`	`source`	`sum_total_amount`
`membership_start_date`	`status_name`	`receive_date`
`membership_end_date`	`membership_type_name`	

Additionally, the following letters generate additional component tokens:

> Celebration: $contribution.soft_credit_type
>
> Tribute Summary: `$contribution.min_date`, `$contribution.max_date`, and `$contribution.summary`. The latter is an array also, with each row holding summary information of a single contribution. The summary information includes all contact data points for the individual donor, as well as `address_summary`, `amount`, and `contribution_date`.