Customizing CiviCRM

Lisa Rau/Ashma Shrestha

Stuart Gaston

Outline

- Who Are WE?
- Why CiviCRM?
- Why does it need Customization?
- Tools and Techniques
- Pros and Cons

Confluence
where IT all comes together

Who are we?

- Lisa Rau
  - CEO and Co-Founder of Confluence focusing on Non-Profit since 2001
  - 425 Non-Profit clients and counting
  - Specialize in open source – Civi, Joomla and Drupal
- Ashma Shrestha
  - Senior Web Developer
  - Specialized in Joomla/Wordpress/ Drupal
  - Has been working with CiviCRM since 2011
- Stuart Gaston

Why CiviCRM?

- Tools for Non-Profits
  - Constituent management
  - Membership Management
  - Donations
  - Advocacy
  - Events Management
- Active community with 20,340+ Members (and growing)
- Excellent integration with Drupal and Joomla
- Low total Cost of Ownership
  - While free; still requires implementation

Why Does it Need Customization?

- Every Organization works differently:
  – CiviCRM covers the basic needs of non-profits, every organization has it own unique set of requirements
- CiviCRM provides tools for tailoring the needs of organizations using a variety of methods
- We will be focusing on:
  – Custom Data and Profile
  – Templates and PHP files override
  – Custom modules or hooks

Three Methods

- Custom data
- Overriding templates and PHP files
- Creating custom modules
  - Using hooks
  - Creating custom CiviCRM modules

Confluence
where IT all comes together

Problems

- Each technique has pros and cons
- Choice of technique depends on the particular requirements
- Keep in mind
  - When requirements change
  - Appropriateness of technique used changes

Method 1: Custom Data

- Easiest and simplest way of extending CiviCRM
- Standard set of fields for storing, searching and extending CiviCRM elements. Example:
  - Event Registration Information
    - Participant Custom Data type
    - Profiles to add it to the form
  - Credit Union Information
    - Asset Size
    - Credit Union Charter number
- Can the issue be resolved by Custom Data?

Confluence
where IT all comes together

Custom Data Example – Registration Data

Registration Info

SALUTATION: - select -

FIRST NAME *

LAST NAME *

First and last name will be shared with other visitors to the site.

ORGANIZATION *: - Select -

If you do not see the name of your organization in this list, please select "other" and add your organization information. (Affiliation will be required in order to register

JOB TITLE *

Please check all categories that contact would like to receive training invitations for.

TRAINING INTEREST

- ADVOCACY
- COLLECTIONS
- COMPLIANCE
- CREDIT UNION FINANCES
- FINANCIAL LITERACY
- HUMAN RESOURCES
- INFORMATION TECHNOLOGY
- LENDING
- MARKETING/BUSINESS DEVELOPMENT
- OPERATIONS/MEMBER SERVICE

ROLE *: - select -

DEPARTMENT

- ACCOUNTING/FINANCE
- CHECKING
- COLLECTIONS
- COMPLIANCE
- CREDIT/DEBIT CARD SERVICE
- HR/EMPLOYEE BENIFITS
- INFORMATION TECHNOLOGY/INTERNET
- IRA SERVICES
- LENDING
- MARKETING
- MEMBER FINANCIAL SERVICES
- MEMBER SERVICES
- MORTGAGE
- RISK MANANGEMENT
- TRAINING
- OTHER

Custom Data – Participants

▾ GAC Profile Questions

Will you be attending the reception?

GAC RECEPTION - FREE ○ YES ○ NO (clear)

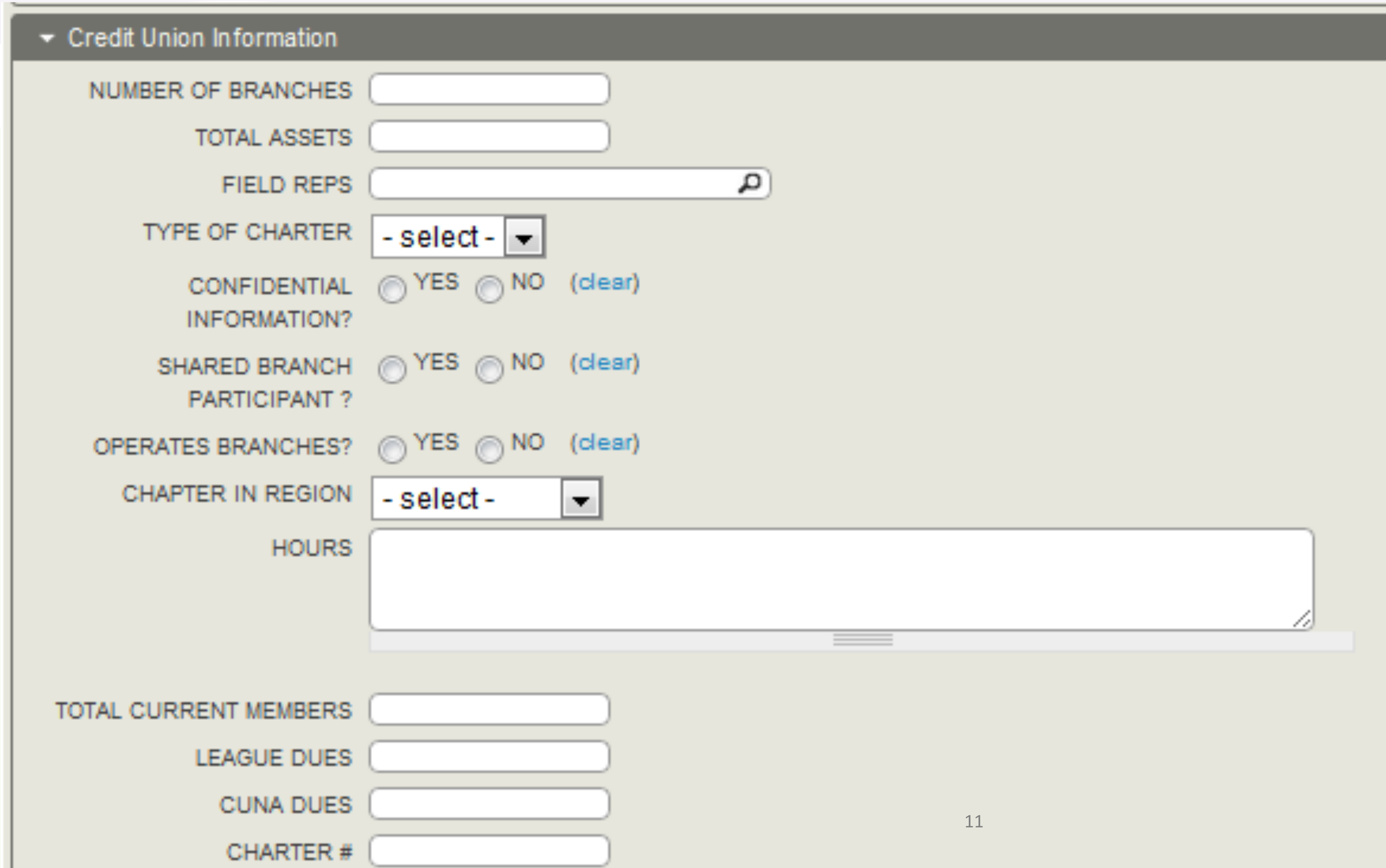
Will you join us for dinner?

PARTICIPANT GAC DINNER - FREE ○ YES ○ NO (clear)

Would you like us to set up visits with your legislator(s)?

LEGISLATOR VISIT ○ YES ○ NO (clear)

Custom Data – Credit Union



Credit Union Information

NUMBER OF BRANCHES

TOTAL ASSETS

FIELD REPS

TYPE OF CHARTER - select -

CONFIDENTIAL INFORMATION? YES NO (clear)

SHARED BRANCH PARTICIPANT ? YES NO (clear)

OPERATES BRANCHES? YES NO (clear)

CHAPTER IN REGION - select -

HOURS

TOTAL CURRENT MEMBERS

LEAGUE DUES

CUNA DUES

CHARTER #

Custom Data - Pros

- Build in Core
- Provides flexibility extend:
  - Contact, Events, activities, Contribution, Membership, Relationship, Participants
- Advanced Search
- Can be accessed via Profiles, Webform, Views

Confluence
where IT all comes together

Custom Data - Cons

- Custom Data variable needs to be manually entered into Drupal “settings.php” file
- Not all the custom data are accessible via Views and Webforms
- Adds tables in the database for each custom data group and columns for the field
  - Overhead - Deleting Custom data will not delete the tables

Method 2 - Template override

- Overriding smarty templates to change layout and modify presentation
- TPL files
- Overridden templates are stored in separate folder
- Changing layouts of Pages and Forms
- Leveraging API to add more features and data
- Leveraging jQuery for enhancement
- Example
  - Restrict registrants to register associated individuals to the

Template Override - Registration

```
<div class="crm-section select-coworker">
    <div class="label">Register someone at your company</div>
    <div class="content">
        <select id="findcoworkers" class="form-select" name="findcoworkers" onchange="fillInfo()">
            <option value="" selected>-select-</option>
            {foreach from=$employees item="employee"}
                <option value="{$employee.cid}">{$employee.display_name}</option>
            {/foreach}
        </select><br>
        <noscript>
            <span class="description" style="color: red;">
                Your browser has Javascript disabled: this feature will not work until you enable it.
            </span><br></noscript>
        <span class="description">Select a co-worker's name, and his or her information will fill in below.</span>
    </div>
```


Confluence
where IT all comes together

Template Override- Continued

```
{if $allowRegistration}
        <div id="event_registration">
            <a href="{$registerURL}" title="{$registerText}" class="button crm-register-button">
                <span>{$registerText}</span></a><br />
        </div>
{else}
    {php}

    global $user;

    if($user->uid < 1) {
        echo '  <div id="event_registration">';
               echo ' <a href="/user" title="Log In to Register" class="button crm-register-button">
                <span>Log In To Register</span></a><br />';
            echo '</div>';

    }
    {/php}
{/if}
```


Confluence where IT all comes together

Template Override - Pros

- Profile ID specific form edit
  - Example: Edits to the form with profile id 52 can be done by creating a subfolder with ID / custom_template/CRM/Profile/Form/52/Edit.tpl
- Quick way of adding functionality and design changes to the forms and pages
- Leverage API for extending functionality

Template Override - Cons

- The core templates might get modified during the upgrades
  - Might break the overridden template
- Causes problems during upgrade if proper procedures are not followed
- Use hooks for complex modifications

Method 3 - Custom modules (Hooks)

- Hooks
  - Technique used in programming for altering/enhancing behavior
  - Extends CiviCRM functionality
  - 39 hooks available
  - Example:
    - Build Price hook: Modify the price set for Events depending on the Organization Asset size
    - Token hook - Add extra Custom mail merge token

Hook: Asset Size Discount

- Custom Data
  - Event Custom Data – Asset Size Discount (custom_170)
  - Organization Custom Data - Credit Union Asset Size (custom_62)
- API Used
  - UFMatch – Get Employer Information
  - Event – Check if Asset Size Discount is applicable
  - Contact – Get Organization Asset Information
- civicrm_asset_discount_civicrm_buildAmount($pageType, &$form, &$amount)

Hook- Asset Size Discount

```
//Update the amount
 $fee_default = $form->_values['event']['default_fee_id'];

 $fee_discount_default = $form->_values['event']['default_discount_fee_id'];

 foreach ( $amount as $amountId => $amountInfo ) {
    if( $amount[$amountId]['amount_id'] == $fee_default
            || $amount[$amountId]['amount_id'] == $fee_discount_default )
    {
        $amount[$amountId]['value'] = $amount[$amountId]['value'] -
        ceil($amount[$amountId]['value'] * $discountPercent / 100);

        $amount[$amountId]['label'] = $amount[$amountId]['label'] .
            "\t - with {$discountPercent}% discount";
    }
 }
```

Creating Custom modules (API)

- API
  - Standard way of interacting with CiviCRM entities (Contacts, activities)
  - Easy to extend custom API Class
- Ways to Call API
  - PHP – Custom Code Override or Custom Modules
  - REST Interface – Accessing CiviCRM from External site
  - Ajax Interface - JavaScript
  - Smarty Interface - Custom Template Override

API - Examples

- PHP – Custom Code Override or Custom Modules
  - require_once 'api/api.php';
  - $event = civicrm_api('Event', 'Get', array('id'=> $eventID, 'version' => 3));
  - REST Interface – Accessing CiviCRM from External site
  - https://www.test.org/{path_civicrm}/civicrm/extern/rest.php?q=civicrm/login&name=user&pass=password&key=key&site_key=sitekey
- Ajax Interface – JavaScript
  - cj().crmAPI ('entity','action',params);
- Smarty Interface - Custom Template Override
  - {crmAPI var="myContactList" entity="Contact" action="get" version="3" first_name="Mr" last_name="T" }

API – Asset Size Discount

```
$eventID = $form->getVar( '_eventId' );
$default_fee = $form->getVar('default_fee_id');
if ( $pageType != 'event' || $user->uid == 0) { // use event ID here
        return;
    }
require_once 'api/api.php';
//Get the Event custom field for Asset size discount
$event = civicrm_api('Event', 'Get', array('id'=> $eventID, 'version' => 3));

$discount = $event['values'][$event['id']]['custom_170'];
if( !$discount ) {
    return;
}
//Get Logged in user
$result= civicrm_api('UFMatch','Get', array('version' =>3, 'uf_id' => $user->uid));
$contactid = $result['values'][$result['id']]['contact_id'];
$contact = civicrm_api('Contact ','Get',array('contact_id' => $contactid, 'version' =>3));

// Get employer
$employername = $contact['values'][$contact['id']]['current_employer'];
$employer = civicrm_api('Contact ',
                'Get',array('organization_name' => $employername,
                'contact_type' => 'Organization',
                'return' => 'custom_62,display_name', 'version' =>3));

Set Discount Percentage
  if( $employer['is_error'] ) {
      return;
  }
```


Confluence
where IT all comes together

Hooks - Cons

- Not feasible in all the cases
  - To invoke an activity from Civi in a program, use the API
  - Adding auto complete field
  - Adding CSS
- Without proper documentation:
  - Can loose track of the flow – applies to all Customization techniques

API -Cons

- Not feasible in all cases
  - Need of complex query: Get the list of all the Organizations which are associates and affiliates
- Might need additional help from Smart groups
  - Smart Groups – List of all the contacts who fulfill a certain criteria
    - All the people attending CiviCON 2012
- Expensive when requiring multiple API calls
  - Get list of all individuals associated with member Organizations who Contributed > $1000 during this year

Using jQuery and CSS to Gain Easy Wins in CiviCRM

The CMS agnostic, cross browser way to get (mostly) what you want

By Stuart from Korlon LLC (find me as "Stoob")

Why is this method OK to use?

- CiviCRM already uses jQuery and CSS
- jQuery and CSS can be used with any CMS supported by CiviCRM: Drupal, Joomla, Wordpress - any version
- jQuery takes the guesswork out of cross-browser script compatibility issues
- jQuery cures disease and feeds children ;-)

What are some common uses?

- Showing and hiding certain parts of a form
- Changing label width and label contents
- Hiding tabs and hiding fields unnecessary in your use case
- Adding headers, changing styles, adding descriptive text

When do I use what?

- Showing and hiding certain parts of a form
- Changing label width and label contents
- Hiding tabs and hiding fields unnecessary in your use case
- Adding headers, changing styles, adding descriptive text
- jQuery is red.   CSS is blue.   Magenta is either.
- ...and CiviCRM loves you.

Why do it this way?

- No merging/diffing .tpl or .php files each time you upgrade Civi
- No modules and PHP that are different for each CMS. Reuse code & save time
- Requires no knowledge of Smarty or PHP
- Flexibility to add jQuery/CSS wherever you want: blocks, templates, modules, themes...

Why not do it this way?

- When you require actual security. These methods only hide, show and alter things
- When your user base turns off Javascript
- When you want to manipulate data and processing, not just presentation and UI
- Because you feel more comfortable doing it another way

How do I jQuery?

- <script>
- jQuery goes here.
- </script>

- in a Drupal block, Joomla module, template, or theme but... Drupal 7 now uses a wrapper like:
- <script>
- (function ($) {
- jQuery goes here.
- }(jQuery));
- </script>

How do I CSS?

- Add CSS to your theme, template, or block that is called ***after*** civicrm.css. This should override whatever CSS defaults there are
- Suggest not configuring alternate civicrm.css file or editing this file directly
- Use Firebug or Chrome to inspect element, use exact or more specific selector to override

Do's and Don'ts

1. **Don't** use drupal_add_js() function
2. **Do** make sure that wherever you are putting the jQuery reads it as <script>. Don't use filters or input modes that wreck it
3. If put in Smarty tpl, **do** <literal> tags like:

```
<literal>
  <script> your code </script>
</literal>
```

Cool Trick! whatever.extra.tpl

Save yourself the headache of diff'ing or merging .tpl on upgrade.

1. Create custom TPL directory (read docs)
2. Rather than add the whole file name it whatever.***extra***.tpl
3. It will be appended to the end of the file each time
4. As of 4.2 will even work with /n/ versions

Thanks Dave Greenberg!

Do's and Don'ts

4. Do use visibility settings to limit the <script> to CiviCRM where you want it such as: civicrm/contribute/transact

5. Do use javascript to limit the code to the specific page you want using id=N in URL

```
<script>
if (document.location.href.indexOf('id=N') > 0 ) { ... }
</script>
```

6. Do use the cj("id#element") namespace

Confluence
where IT all comes together

CSS Commonly Used

- element#id { display: none; }
- element.class { width: Npx; }
- element#id {
-   text-decoration: none;
-   font-weight: normal;
- }
- element#id.double.class { float: clear; }
- You may have to get specific like:
- #content #crm-container div.class { }

jQuery Commonly Used

- .hide() and .show()
- .click()
- .val()
- .before() and .after()
- .html()
- .replaceWith()
- .change()
- .ready()
- .addClass() and removeClass()

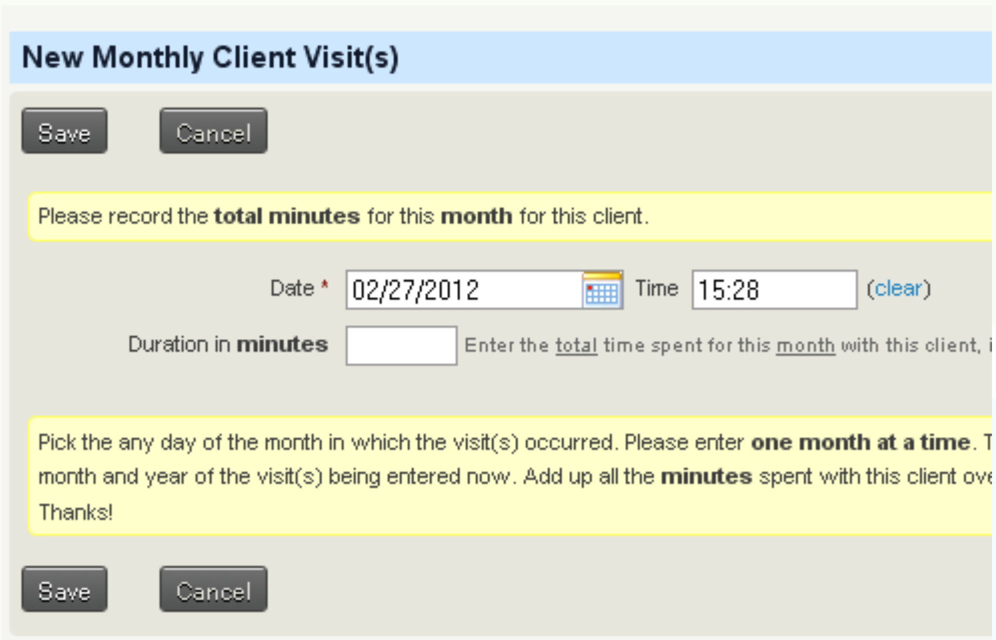
Real Example: Price Set Subheaders

```
<style type="text/css">
  div.date-subheader { width:
   22%; font-weight: bold; text-
   align: right; }
</style>
<script>
cj("div.Panel_Presentations_in_Co
   llabor-section").before("<div
   class='crm-section date-
   subheader'>Thursday, May 24</
   div>");
</script>
```

Inserted into page.php in Wordpress theme.

Real Example: Simple Activities

- This is an actual 'create activity' screen in Civi simplified for non-techie users to show only the fields that matter.



New Monthly Client Visit(s)

Save Cancel

Please record the **total minutes** for this **month** for this client.

Date * 02/27/2012 Time 15:28 (clear)

Duration in **minutes** Enter the total time spent for this month with this client, i

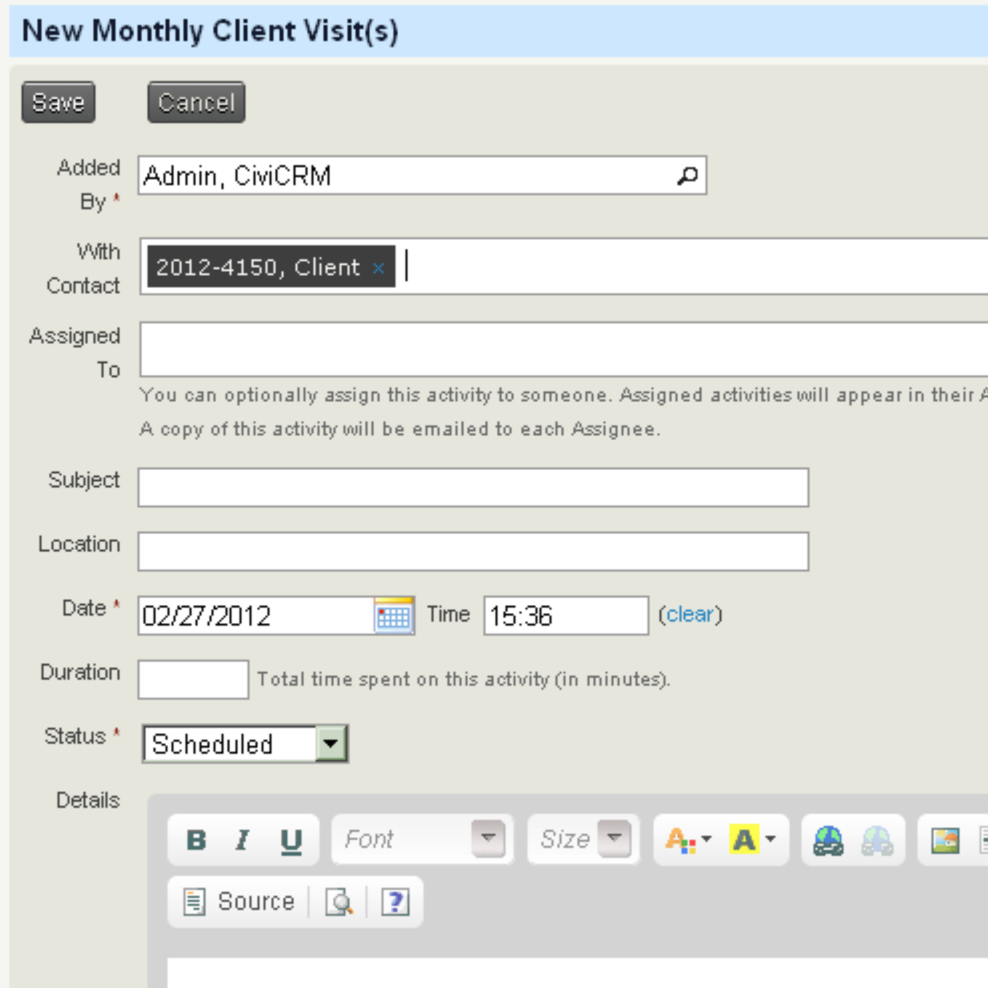
Pick the any day of the month in which the visit(s) occurred. Please enter **one month at a time**. T
month and year of the visit(s) being entered now. Add up all the **minutes** spent with this client ove
Thanks!

Save Cancel

Real Example: Simple Activities

This is what

it used to look like.



Real Example: Simple Activities

- First get the activity type id#, then go nuts with
- hide(), replaceWith() and before()

- if (getParameterByName('atype')) {
-   var choice = getParameterByName('atype');
-   if (choice == '12') {
-   cj("tr.crm-activity-form-block-source_contact_id").hide();
-   cj("tr.crm-activity-form-block-location").hide();
-   cj("tr.crm-activity-form-block-subject").hide();
-   cj("tr.crm-activity-form-block-assignee_contact_id").hide();
-   cj("tr.crm-activity-form-block-duration label").replaceWith("<label for='duration'>...
-   cj("tr.crm-activity-form-block-target_contact_id").before("<tr>...

Confluence where IT all comes together

Real Example: Peekaboo Forms

Show a Profile on a contribution form only when a user selects an option.

1. Find the correct Contribution page with:

```
document.location.href.indexOf('id=N') > 0
```

2. Find the id# of the element you want to trigger and use it:

```
$('input#CIVICRM_QFID_4_4').click(function() {
  cj(".custom_post_profile-group").show();
});
```

3. While you're at it, preselect the state:

```
cj("select#state_province-6").val('1045');
```

4. And trigger a 'same as' address copy:

```
cj('#custom_6\\[Yes\\]').click(function() {
  cj("#city-7").val($("#city-6").val());
});
```

TIME FOR QUESTIONS!!

Confluence where IT all comes together