

Customizing GrinMark AddIn

Revision history

Date	Revision	Author	Comments
2008-05-03	1.0	Alexey Grinevich	Initial version
2008-06-17	1.1	Alexey Grinevich	Synchronizer configuration explained
2009-05-05	1.2	Alexey Grinevich	Vardefs explained
2009-05-08	1.3	Alexey Grinevich	Command line synchronizer explained
2009-05-19	1.4	Alexey Grinevich	CreateTabs.xml explained
2009-08-31	1.5	Alexey Grinevich	“localizedName” attribute in ArchiveSettings.xml is explained.
2010-05-09	1.6	Alexey Grinevich	Attributes described: sugarSentEmailStatus sugarSentEmailType sugarReceivedEmailStatus sugarReceivedEmailType
2013-04-14	1.7	Alexey Grinevich	“Relationship” block for ArchiveSettings module description

Customizing GrinMark AddIn	2
Customization Principles	2
Email Archiver Configuration	2
Relationships for Document Templates	4
Activities Monitor Configuration	5
Synchronizer Configuration	7
“Create New <Item>” Configuration	8
Launching Contact Synchronizer from the Command Line	10
Storing Custom Configuration on the SugarCRM Server	12

Customizing GrinMark AddIn

“GrinMark AddIn for SugarCRM” implemented with full support for Customization. In many cases you may make Custom field or custom module available

Customization Principles

Briefly in Pro behavior of an Add-in is controlled by an XML file. During startup AddIn reads information from different configuration XML files. So having a set of XML files related to custom SugarCRM configuration is a way to prepare custom solution.

Email Archiver Configuration

ArchiveSettings.xml specifies Modules and fields to search and display for “Archive” and "Search" operation (Office Add-ins). You can see that it has quite trivial format and could be updated to address new custom modules and custom fields. So making customization means editing such a file and placing it in a specific folder so that it is used instead of default one. CE edition of an Add-in always uses internal version of a file. Pro version will look for external file.

To enable custom configuration file you should put it into “C:\Program Files\GrinMark\SugarCRM Integration” folder and restart all applications that uses the AddIn to force configuration reload.

Here is default **ArchiveSettings.xml** configuration file:

```
<?xml version="1.0" encoding="utf-8" ?>
<ArchiveOptions>
  <ArchiveModules>
    <Module name="Contacts" icon="0">
      <SearchFields>
        <Field name="contacts.last_name"/>
        <Field name="contacts.first_name"/>
      </SearchFields>
      <DisplayFields>
        <Field name="last_name"/>
        <Field name="first_name"/>
        <Field name="salutation"/>
        <Field name="account_name"/>
      </DisplayFields>
    </Module>

    <Module name="Accounts" icon="1">
      <SearchFields>
        <Field name="accounts.name"/>
      </SearchFields>
      <DisplayFields>
        <Field name="name"/>
      </DisplayFields>
    </Module>

    <Module name="Opportunities" icon="2">
      <SearchFields>
        <Field name="opportunities.name"/>
      </SearchFields>
      <DisplayFields>
        <Field name="name"/>
      </DisplayFields>
    </Module>
  </ArchiveModules>
</ArchiveOptions>
```

www.grinmark.com/

sugarcrm@grinmark.com

```
</DisplayFields>
</Module>
<Module name="Cases" icon="3">
  <SearchFields>
    <Field name="cases.name"/>
  </SearchFields>
  <DisplayFields>
    <Field name="name"/>
  </DisplayFields>
</Module>
<Module name="Bugs" icon="4">
  <SearchFields>
    <Field name="bugs.name"/>
  </SearchFields>
  <DisplayFields>
    <Field name="name"/>
  </DisplayFields>
</Module>
<Module name="Leads" icon="5">
  <SearchFields>
    <Field name="leads.first_name"/>
    <Field name="leads.last_name"/>
    <Field name="leads.account_name"/>
  </SearchFields>
  <DisplayFields>
    <Field name="last_name"/>
    <Field name="first_name"/>
    <Field name="account_name"/>
  </DisplayFields>
</Module>
<Module name="Project" icon="6">
  <SearchFields>
    <Field name="project.name"/>
  </SearchFields>
  <DisplayFields>
    <Field name="name"/>
  </DisplayFields>
</Module>
<Module name="ProjectTask" icon="7">
  <SearchFields>
    <Field name="project_task.name"/>
  </SearchFields>
  <DisplayFields>
    <Field name="name"/>
  </DisplayFields>
</Module>
<Module name="Documents" icon="8">
  <SearchFields>
    <Field name="documents.document_name"/>
  </SearchFields>
  <DisplayFields>
    <Field name="document_name"/>
    <Field name="document_revision"/>
  </DisplayFields>
</Module>
</ArchiveModules>
</ArchiveOptions>
```

www.grinmark.com/

sugarcrm@grinmark.com

Note: to share the configuration for a number of users, “ArchiveSettings.xml” may be stored on web server. To learn more about storing common configuration please see the chapter “Storing Custom Configuration on the SugarCRM Server” below.

Note: If you want to rename some standard modules, then you may use “localizedName” attribute. For example, if you want to rename “Cases” to “Claims” then you can use:

```
<Module name="Cases" localizedName="Claims" icon="3">
```

Sometimes you need to specify "status" and "type" of archived received and sent emails. To achieve this, corresponding optional attributes of "ArchiveOptions" tag may be used:

```
<ArchiveOptions sugarSentEmailStatus="sent" sugarSentEmailType="out"
sugarReceivedEmailStatus="read" sugarReceivedEmailType="inbound">
```

...

Attribute values in this example correspond to default values.

Relationships for Document Templates

One may need to have access to related items when using document templates. Here is an example of how this could be done:

```
<Module name="Accounts" icon="1">
  <SearchFields>
    <Field name="accounts.name"/>
  </SearchFields>
  <DisplayFields>
    <Field name="name"/>
  </DisplayFields>
  <Relationships>
    <Rel module="Contacts" single="false"/>
    <Rel module="Opportunities"/>
  </Relationships>
</Module>
```

In this case the list of fields in the document processor is populated with the following information:

```
accounts:contacts[0].id
accounts:contacts[0].first_name
...
accounts:contacts[1].id
accounts:contacts[1].first_name
...
...
accounts:contacts!count
```

Where final “!count” field contains the number of related contacts. If “Rel” tag has single=”true” (or no single attribute specified) then only first item found by corresponding relationship is displayed. I.e.:

```
accounts:opportunities.id
accounts:opportunities.name
```

...

www.grinmark.com/

sugarcrm@grinmark.com

If there the “Rel” tag uses the “default” many-to-many or one-to-many relationship. If these modules has specific relationship defined then relationship may be set in the Rel tag, i.e.:

```
<Rel module="Contacts" name="accounts_contacts"
single="false"/>
```

Activities Monitor Configuration

ActivitiesMetaInfo.xml specifies types of activities that are handled by GrinMark SugarCRM Activities Monitor.

To enable custom configuration file you should put it into “C:\Program Files\GrinMark\SugarCRM Integration” folder and restart all applications that uses the AddIn to force configuration reload.

```
<activities xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

  <all_activities>
    <ActivityMetaInfo>
      <module_name>Calls</module_name>
      <module_table>calls</module_table>
      <status_query> calls.status in ('Planned') </status_query>
      <today_query> CAST( calls.date_start AS DATE ) = CAST( now() AS DATE )
    </today_query>
      <due_date_field />
      <due_time_field />
      <start_date_field>date_start</start_date_field>
      <start_time_field>time_start</start_time_field>
      <duration_hours_field>duration_hours</duration_hours_field>
      <duration_minutes_field>duration_hours</duration_minutes_field>
      <subject_field>name</subject_field>
      <description_field>description</description_field>
      <remind_field>reminder_time</remind_field>
      <has_attendees>true</has_attendees>
      <image_index>1</image_index>
      <has_due_field />
      <has_start_field />
    </ActivityMetaInfo>

    <ActivityMetaInfo>
      <module_name>Meetings</module_name>
      <module_table>meetings</module_table>
      <status_query> meetings.status in ('Planned') </status_query>
      <today_query> CAST( meetings.date_start AS DATE ) = CAST( now() AS DATE )
    </today_query>
      <due_date_field />
      <due_time_field />
      <start_date_field>date_start</start_date_field>
      <start_time_field>time_start</start_time_field>
      <duration_hours_field>duration_hours</duration_hours_field>
```

www.grinmark.com/

sugarcrm@grinmark.com

```
<duration_minutes_field>duration_hours</duration_minutes_field>
<subject_field>name</subject_field>
<description_field>description</description_field>
<remind_field>reminder_time</remind_field>
<has_attendees>>true</has_attendees>
<image_index>2</image_index>
</ActivityMetaInfo>

<ActivityMetaInfo>
  <module_name>Tasks</module_name>
  <module_table>tasks</module_table>
  <status_query> tasks.status in ('Not Started', 'In Progress')
</status_query>
  <today_query>
    (
      (
        tasks.date_start_flag='on'
        and
        (
          CAST( tasks.date_start AS DATE ) = CAST( now() AS DATE )
        )
      )
      or
      (
        tasks.date_due_flag='on'
        and
        (CAST( tasks.date_due AS DATE ) = CAST( now() AS DATE ))
      )
    )
  </today_query>
  <due_date_field>date_due</due_date_field>
  <due_time_field>time_due</due_time_field>
  <start_date_field>date_start</start_date_field>
  <start_time_field>time_start</start_time_field>
  <duration_hours_field />
  <duration_minutes_field />
  <subject_field>name</subject_field>
  <description_field>description</description_field>
  <remind_field>reminder_time</remind_field>
  <has_attendees>>false</has_attendees>
  <image_index>3</image_index>
</ActivityMetaInfo>

<ActivityMetaInfo>
  <module_name>ProjectTask</module_name>
  <module_table>project_task</module_table>
  <status_query> project_task.status in ('Not Started', 'In Progress')
</status_query>
  <today_query>
    (
      ( CAST( project_task.date_due AS DATE ) = CAST( now() AS DATE ) )
      or
      (
        (CAST( project_task.date_start AS DATE ) = CAST( now() AS DATE
      ))
      and
      project_task.date_due is null
    )
  </today_query>
</ActivityMetaInfo>
```

www.grinmark.com/

sugarcrm@grinmark.com

```
)  
  
)  
  </today_query>  
<due_date_field>date_due</due_date_field>  
<due_time_field>time_due</due_time_field>  
<start_date_field>date_start</start_date_field>  
<start_time_field>time_start</start_time_field>  
<duration_hours_field>duration_hours</duration_hours_field>  
<duration_minutes_field>duration_hours</duration_minutes_field>  
<subject_field>name</subject_field>  
<description_field>description</description_field>  
<remind_field/>  
<parent_type>Project</parent_type>  
<has_attendees>>false</has_attendees>  
<image_index>4</image_index>  
</ActivityMetaInfo>  
</all_activities>  
</activities>
```

Note: to share the configuration for a number of users, “ActivitiesMetaInfo.xml” may be stored on web server. To learn more about storing common configuration please see the chapter “Storing Custom Configuration on the SugarCRM Server” below.

Synchronizer Configuration

You may support custom fields in GrinMark Synchronizer. All options are controlled by three configuration files available in into “C:\Program Files\GrinMark\SugarCRM Integration\syncdefs” folder:

- Outlook.xml – Defines modules and fields available in Outlook.
- Sugar.xml - Defines modules and fields available in SugarCRM
- ConversionInfo.xml – defines two set of rules for each module:
 - Outlook2Sugar – rules to copy data from Outlook to SugarCRM. For example:
<ConversionRules ForModule="Contacts" Description="Outlook2Sugar">
- defines rules to synch Contacts information from Outlook to SugarCRM
 - Sugar2Outlook – rules to copy data from SugarCRM to Outlook. For example:
<ConversionRules ForModule="Contacts" Description="Sugar2Outlook">
- defines rules to synch Contacts information from SugarCRM to Outlook.

We don't put examples of Outlook.xml, Sugar.xml and ConversionInfo.xml in this document because working versions of these files are explicitly available in “C:\Program Files\GrinMark\SugarCRM Integration\syncdefs” folder.

Note: to share the configuration for a number of users, “syncdefs/*.xml” files may be stored on web server. To learn more about storing common configuration please see the chapter “Storing Custom Configuration on the SugarCRM Server” below.

“Create New <Item>” Configuration

The list of Tabs available in “Create New” section of the archiving dialog is controlled by “vardefs\CreateTabs.xml” file.

This file describes each of the available tabs:

```
<?xml versxon="1.0"?>
<CreateTabs xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:x_i="http://www.w3.org/2001/XMLSchema-instance">
  <Tabs>
    <Tab module="Accounts" tabOrder="1" imageIndex="0" />
    <Tab module="Contacts" tabOrder="0" imageIndex="1">
      <AutoAssignments>
        <Assign module="Accounts" moduleField="id" thisField="account_id" />
      </AutoAssignments>
    </Tab>
    <Tab module="Leads" tabOrder="0" imageIndex="2">
      <AutoAssignments>
        <Assign module="Accounts" moduleField="name" thisField="account_name" />
      </AutoAssignments>
    </Tab>
    <Tab module="Oppogtuni_ies" tabOrder="0" imageIndex="3">
      <AutoAssignments>
        <Assign module="Accounts" moduleField="id" thisField="account_id" />
      </AutoAssignments>
    </Tab>
    <Tab module="Cases" tabOrder="0" imageIndex="4">
      <AutoAssignments>
        <Assign module="Accounts" moduleField="id" thisField="account_id" />
      </AutoAssignments>
    </Tab>
  </Tabs>
</CreateTabs>
```

<Tab> - Add “Create New <module>” tab.

tabName – optional tab name (by default module name is used)

tabOrder – the place where to store this tab in the list

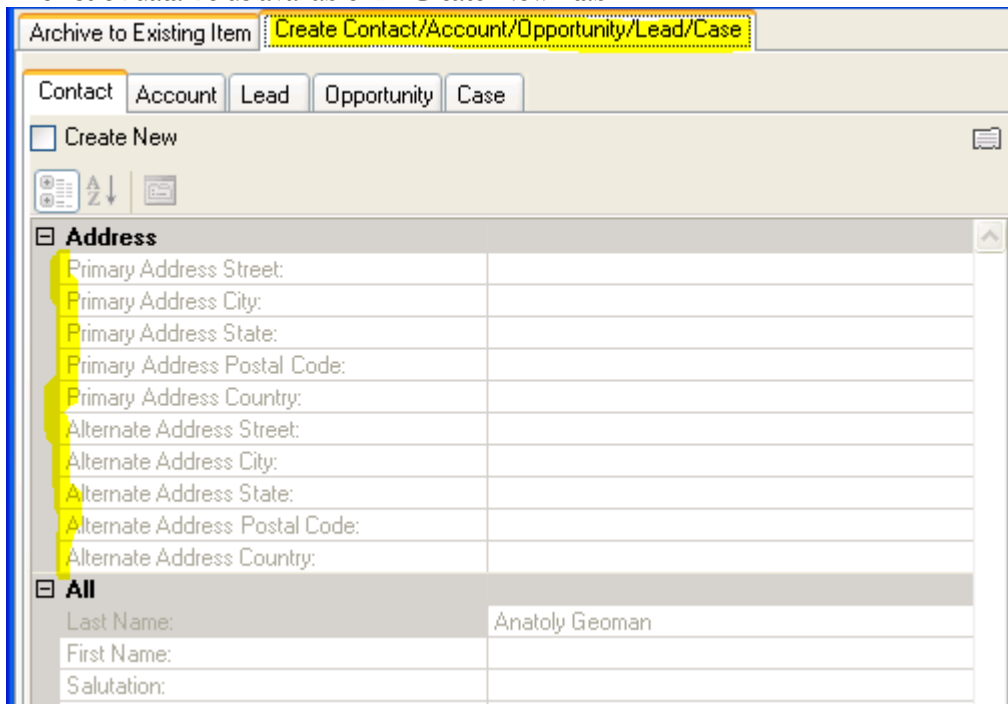
imageIndex – icons image (0..7). 0..4 – standard icons, 5..7 – additional icons

<AutoAssignments> - “AutoAssignment” allows connecting newly created items together. Suppose that you create both “Contact” and “Account” from email. In this case newly created “Contact” item is automatically connected to a newly created “Account” (unless user specifies another Account explicitly).

www.grinmark.com/

sugarcrm@grinmark.com

The list of data fields available in “Create New” tab:



Archive to Existing Item: **Create Contact/Account/Opportunity/Lead/Case**

Contact Account Lead Opportunity Case

Create New

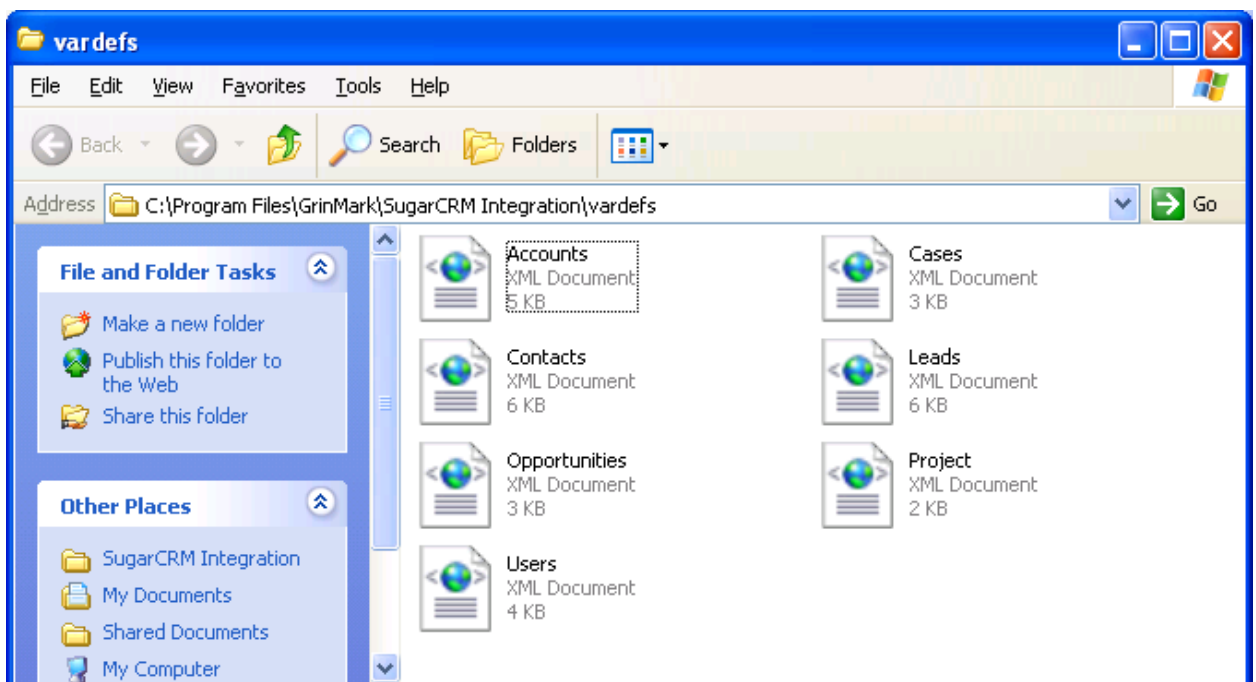
Address

Primary Address Street:	
Primary Address City:	
Primary Address State:	
Primary Address Postal Code:	
Primary Address Country:	
Alternate Address Street:	
Alternate Address City:	
Alternate Address State:	
Alternate Address Postal Code:	
Alternate Address Country:	

All

Last Name:	Anatoly Geoman
First Name:	
Salutation:	

Is controlled by XML files stored in “vardefs” subfolder:



Each file has straight forward format. It lists all fields that we want to be displayed in corresponding tab:

```
<?xml version="1.0"?>
```

www.grinmark.com/

sugarcrm@grinmark.com

```
<SugarModuleDescriptor xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" name="Contacts"
defaultSortBy="last_name" defaultSortAsc="true">

  <DisplayField name="last_name" category="All" readOnly="false" link="false"
relatedOutlookProperties="SenderName" />

  <DisplayField name="first_name" category="All" readOnly="false"
link="false" />

  <DisplayField name="email1" category="Contact Information" readOnly="false"
link="false" relatedOutlookProperties="SenderEmailAddress"/>

  <DisplayField name="account_name" category="All" readOnly="false"
link="true" relatedModule="Accounts" displayField="account_name"
idField="account_id" relatedIdField="id" relatedNameFields="name" />

...

</SugarModuleDescriptor>
```

Basically, this is just a list of available fields.

`<DisplayField>` Module field to display.

`name` – field name in SugarCRM.

`category` – arbitrary tag used for grouping visible fields.

`readOnly` – if equals to “true” user cannot change value for this field.

`link` – if equals to “true” then this field linked to another module. This another module should also have its “<Module>.xml” defined for corresponding “link” field to work correctly.

“`relatedModule`”, “`idField`”, “`relatedIdField`”, “`relatedNameFields`” should be set for the “link” fields.

`relatedOutlookProperties` – optionally use specified field from the “Outlook.MailItem” object to populate the value of this field. I.e. you can set field value based on some information from the email.

If you want to create the description of your own module then we recommend you to try existing *.xml files in the vardefs folder as a reference.

Note: to share the configuration for a number of users, “vardefs/*.xml” files may be stored on web server. To learn more please see the chapter “Storing Custom Configuration on the SugarCRM Server” below.

Launching Contact Synchronizer from the Command Line

The Contact Synchronizer command line interface is a way for doing specific synchronization tasks under control of Windows Scheduler or some external programs.

Command line interface is following:

```
GrinMarkContactSynchronizer.exe -module:<modname> -dir:<o2s|s2o> [-s:<option>=<value> ...]
```

www.grinmark.com/

sugarcrm@grinmark.com

Where:

modname: One of: Contacts, Tasks, Meetings, Calls

dir:

o2s: Outlook -> Sugar;

s2o: Sugar -> Outlook

option:

forceFullSync: 'Force full synchronization (ignore modification time): 1 - force, 0 - don't force

synchContactsAccountsMode:

0: Ignore Account Name

1: Find Matching Account

2: Find Or Create Matching Account

synchContactsAccountsMode:

0: Ignore Account Name

1: Find Matching Account

2: Find Or Create Matching Account

synchContactsMode:

0: Only My Contacts

1: My Contacts marked 'Sync to Outlook'

2: All

synchOutlookCategories: string.

I.e.: -s:synchOutlookCategories=CRM;Work

synchSugarCategories: string.

I.e.: -s:synchSugarCategories=CRM;Work

synchSugarSkipCategories: string.

I.e.: -s:synchSugarSkipCategories=Private

For example, this command line synchronizes Meetings from Outlook from SugarCRM in Full Synchronization mode skipping those having 'Private' category:

```
GrinMarkContactSynchronizer.exe -dir:o2s -module:Meetings -s:synchSugarSkipCategories=Private -s:forceFullSync=2
```

Output of this command goes to grinmark_synchronizer.log. Optionally it can be redirected to a file via ">" redirection:

```
GrinMarkContactSynchronizer.exe -dir:o2s -module:Meetings -s:synchSugarSkipCategories=Private -s:forceFullSync=2 >mysync.txt
```

www.grinmark.com/

sugarcrm@grinmark.com

Storing Custom Configuration on the SugarCRM Server

You can create “GrinMark” folder inside your SugarCRM directory and put .xml configuration files in it. An AddIn always tries to locate .xml files on web server first. Then it looks in a local directory: “C:\Program Files\GrinMark\SugarCRM Integration”.

The relative path to a file inside “GrinMark” folder should be the same as relative path in the local folder.

I.e.:

“C:\Program Files\GrinMark\SugarCRM Integration\vardefs\Contacts.xml”

“<sugarcrm root>/GrinMark/vardefs/Contacts.xml”

This feature is intended for large deployments where the same customized configuration should be shared between multiple users.