Microsoft® Dynamics CRM for Sitecore 6.6-8.0

Save Actions User Guide

A practical guide to using Microsoft® Dynamics CRM Save Actions for Web Forms with Marketers

Table of Contents

Chapter 1

Introduction

The Web Forms for Marketers module allows you to create forms and localize them in different languages The Web Forms for Marketers module also contains Submit Actions that allow you to use the information provided in forms, to trigger complex business logic.

The Microsoft Dynamics CRM Save Actions are an add-on to the Web Forms for Marketers module. They allow marketers to leverage the vital information provided in forms to create or update records directly in CRM. This gives marketers the tools to quickly build integrated and complex business logic between their website and CRM.

The Dynamics CRM Save Actions form the core of Sitecore's Dynamics CRM Integration, together with the Dynamics CRM Security Provider.

This document contains the following chapters:

- **Introduction**
  This introduction to the document including a description of how to set up the CRM integration.
- **Configuring Dynamics CRM Save Actions**
  A description of how to configure the Dynamics CRM Save Actions.
- **Tips for Developers**
  Some tips and tricks for developers.

1.1 Prerequisites

The Dynamics CRM Save Actions require:

- Web Forms for Marketers 2.1 and later.
- CRM Security Provider 2.0.0 rev.130731 and later.
- Sitecore’s Microsoft CRM integration license.

The Web Forms for Marketers Dynamics CRM Save Actions are designed to connect with a single CRM instance.

1.2 Security

When you install Dynamics CRM Save Actions, a new Sitecore role is created in the **Role Manager** called *CRM Client Form Author*. To view and configure the Dynamics CRM Save Actions, you must be a member of this role or another role which inherits this role.

Note
The Dynamics CRM Save Actions require an active connection to the CRM system to function correctly. If you have added Dynamics CRM Save Actions, you will not be able to edit or configure them if there is no connection to the CRM system.

Chapter 2

Configuring Dynamics CRM Save Actions

When you configure the Dynamics CRM Save Actions, you must select the fields in CRM that you want to populate and specify the save actions that you want to use.

This chapter contains the following sections:

- Configuration
- Selecting the CRM fields to Populate
- Populating the Fields
- Individual CRM Save Actions

2.1 Configuration

The following setting in the `crm.config` file allows using CRM Security Provider API to operate with CRM data:

```
<setting name="Crm.ConnectionStringName" value="CRMConnString" />
```

The name of the connection string for CRM is defined in the `ConnectionStrings.config` file.

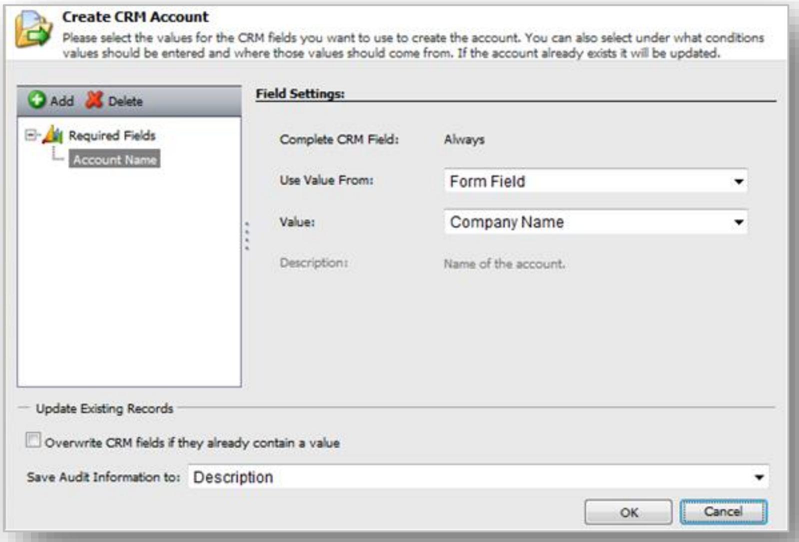
2.2 Selecting the CRM fields to Populate

The Dynamics CRM Save Actions automatically read the available fields for the CRM Entity you are working with, as well as the type of each individual field.

2.2.1 Required fields

The required fields for a CRM record are always listed in the right hand pane of the **CRM Save Action** dialog box.

The following example shows a CRM account that only contains one required field; **Account Name**.



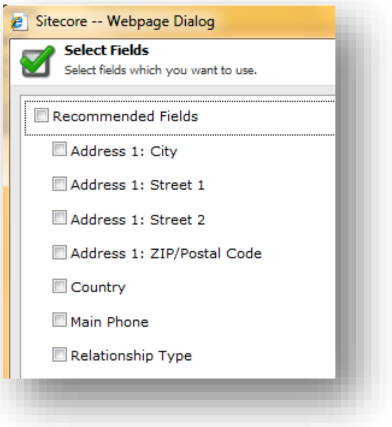
Required fields must be populated. If they are not populated the save action will not work. In this example, the **Account Name** field is populated by the value that the visitor enters in the **Company Name** field in the form.

If the CRM system contains a description of the field, it is displayed in the **Description** field of the CRM save action.

Note
The name of this dialog box always reflects the name of the CRM save action that you are editing.

2.2.2 Recommended fields

To add additional CRM fields to the CRM save action, in the **Create CRM Account** dialog box, in the right hand pane, click **Add**. The CRM save actions will automatically detect which fields are set as *recommended* in the CRM system and display these first in the list of field. This picture shows the recommended fields for a CRM account:

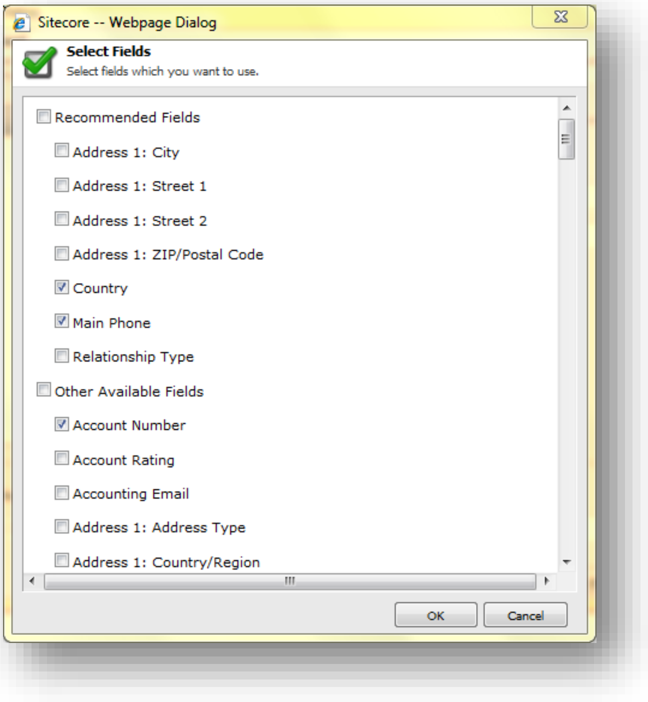


To add all the recommended fields, select the **Recommended Fields** check box.

2.2.3 Adding Other Fields

To add additional CRM fields to the CRM save action, click the **Add** and in the **Selected Fields** dialog box, select the field or fields that you wish to add. The **Selected Fields** dialog box displays all the available fields in the CRM entity you are working with in alphabetical order. The recommended fields are listed first.

This picture shows a list of fields for a CRM account, where the **Country**, **Main Phone**, and **Account Number** fields will be added to the CRM Save Action.



2.3 Populating the Fields

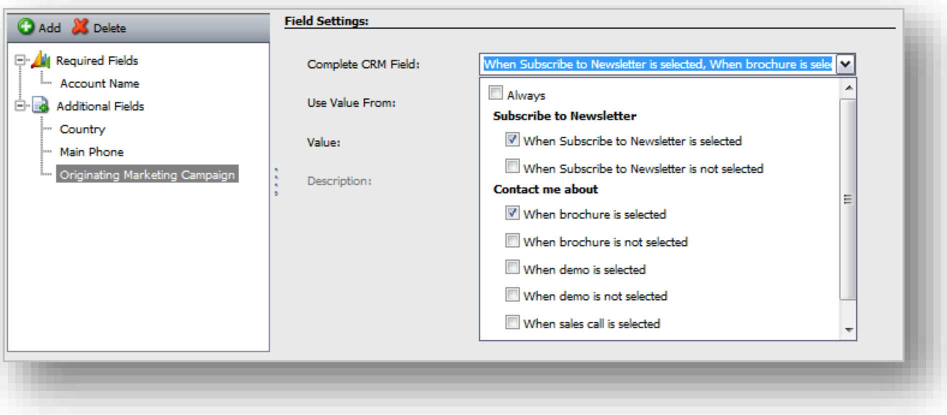
To configure an individual CRM field in the **CRM Save Action Edit** dialog box, click on the field name. The **Field Settings** section displays the values that you can populate the field with and the conditions under which this should occur.

2.3.1 Field Conditions

The CRM Save Actions allow you to select the conditions under which each CRM field is populated. This may be important to your business logic. For example, if you are tracking users which have subscribed to a specific newsletter campaign, you may wish to only complete the **Originating Marketing Campaign** field if a visitor has selected the **Subscribe to Newsletter** field on the form.

To define the conditions under which a CRM field is populated, in the **Complete CRM Field** field, click the dropdown list and select the condition or combination of conditions that must be fulfilled before the Save Action will populate the CRM field. The default value is **Always** — the CRM field is always populated, regardless of the values entered in the form.

In this example, the **Originating Marketing Campaign** field in the CRM account record will only be populated if the **Subscribe to Newsletter** checkbox is selected and *brochure* is selected from the **Contact me about** check box list.



2.3.2 Populating the CRM Field

The Dynamics CRM Save Actions automatically detect the types of values that can be entered in a selected CRM field.

To select the type of information that you want to enter in the CRM field, in the **Use Value From** field, select the information source.

The **Use Value From** field can contain four different sources depending on the CRM field type that you selected:

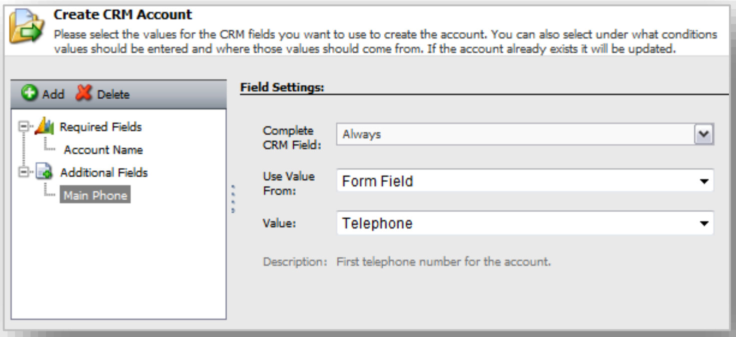
- Form Field
- Previous Save Action
- CRM
- Manual

Form Field

This source uses the value entered by a visitor in a form field.

When you select *Form Field*, all the fields in the form are displayed in the **Value** field in the drop down list. Select the form field whose value will be used to populate the selected CRM field.

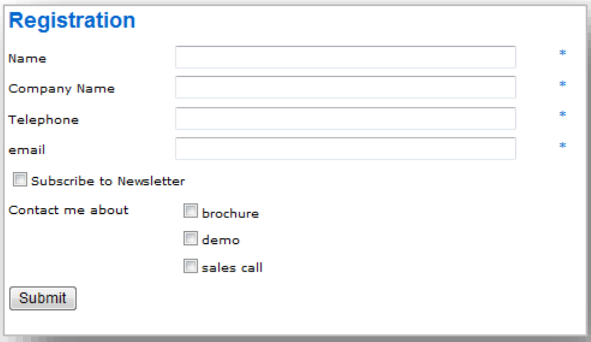
In this example, the **Main Phone** field in the CRM account will be populated with the value entered by the visitor in the **Telephone** field on the form.



Previous Save Action

You can also use values from previous CRM Save Actions on a form in subsequent CRM Save Actions.

For example; if you have a *Registration* form on you site which allows potential partners to register, you might like to add the company name that visitors enter in the form as an account in CRM and the name they enter as a contact that is linked to this account.



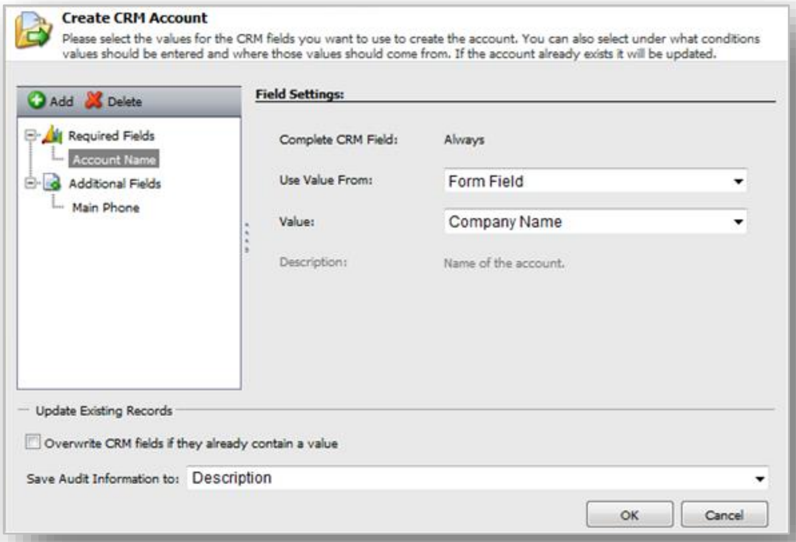
To achieve this, you must:

Create the Create CRM Account save action that uses the uses the values that the user enters in the company name field in the form to create a CRM account.

You must create a CRM Contact save action that uses the account you have just created as its parent customer.

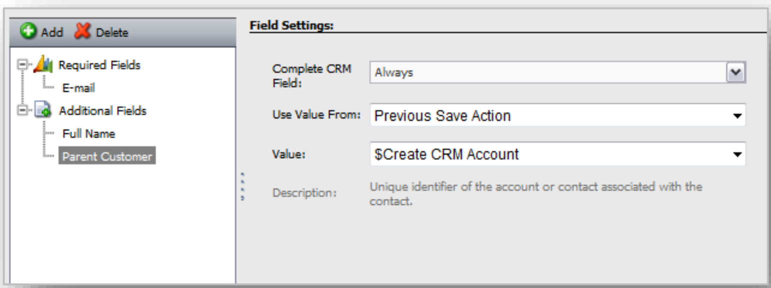
To configure this Dynamics CRM Save Action:

1. In the **Form Designer**, add a Create CRM Account save action.



2. In the left hand pane of the Create CRM Account dialog box, select the **Account Name** CRM field.
3. In the **Use Value From** field, select *Form Field* and in the **Value** field, select *Company Name*.
4. Click **Save**.
5. In the **Form Designer**, add a Create CRM Contact save action.
6. Add, select, and configure the CRM fields that you want to populate.
7. Add and select the **Parent Customer** CRM field
8. In the **Use Value From** field, select *Previous Save Action* and in the **Value** field, select *$Create CRM Account*.

   The Dynamics CRM Save Action should look like this:



If no previous action exists or the previous save action has failed, the CRM field is not updated.

CRM

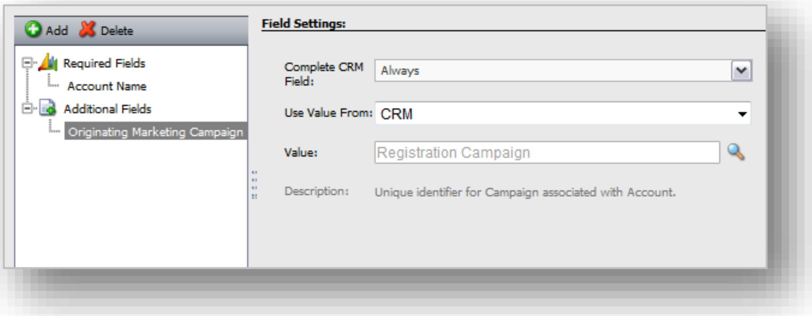
You can also populate a CRM field with the value from another CRM field. This is used for CRM lookup fields where the field values can be other fields in CRM.

If the Registration form used in the previous example was a part of an online registration campaign, you might want all the accounts created using this form to be marked with an appropriate originating marketing campaign.

To configure this:

1. Select the **Originating Marketing Campaign** CRM field.
2. In the **Use Value From** field, select *CRM*.
3. In the **Value** field, click the **Browse** button and a list of the campaigns in CRM is displayed.
4. Select the campaign you want to use and click **OK**.

   The save action configuration should look like this:



If a CRM field has a limited number of possible values, these are shown in a dropdown list in the **Value** field.

Manual

You can enter text into several types of CRM field. You should use this option if you want to populate a field with a fixed text value.

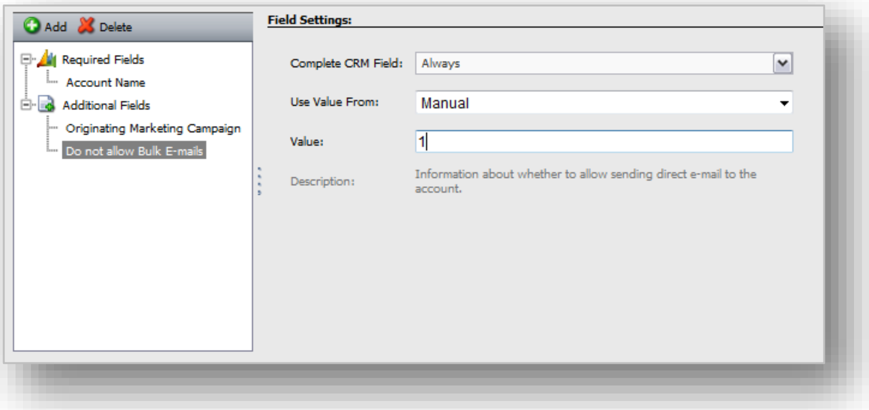
To populate a manual field:

1. In the **Use Value From** field, select *Manual.*

   The **Value** field displays the default value if there is one.
2. In the **Value** field, enter the text that you want to populate the CRM field with.

This could also be used for true/false fields. For example, if you wanted to ensure that you did not send bulk e-mails to customers who registered using the form, you might want make this value true. In CRM *0* is false (unselected) and *1* is true (selected).

The save action configuration should look something like this:



You could also user this for more complex operations, by entering GUIDs or other unique identifiers.

2.3.3 Updating Existing Fields and Audit Information

The Dynamics CRM Save Actions create and update CRM records automatically without any manual intervention. The save actions can therefore write audit information to fields in CRM when they update records. The Dynamics CRM Save Actions register the current and new values entered by the save action in the CRM record.

The Dynamics CRM Save Actions which update existing CRM records — *Create CRM Account* and *Create CRM Contact* — also give users the option to specify whether or not the action should overwrite the values in CRM or only enter information in a selected audit field.

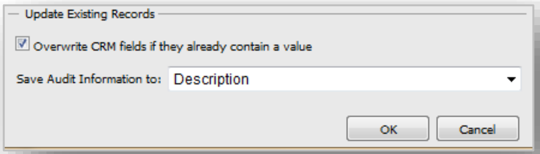
If the **Overwrite CRM values if they already exist** checkbox is selected, all the selected values in the save action will overwrite any existing values in the CRM record. If the checkbox is not selected, the selected values in the audit field will be registered in the selected audit field.

Audit information is entered in the selected field when a record is created, updated, or if the field is not overwritten. If the field is not overwritten, the value that the visitor entered is stored in the designated audit field and the original value remains in the CRM record.

The field can be selected in the “Save Audit Information to” drop down list.

The *Don’t Save* option is selected by default, and therefore no information about field changes is saved.

In this example, CRM fields that contain values are overwritten by values from the form field and the old and new values of the fields are entered as audit information in the **Description** field in the corresponding CRM record.



Audit information can only be saved to CRM fields of the *ntext* field type.

2.4 Individual CRM Save Actions

Here is a short description of the individual CRM save actions that the dynamics CRM Save Action module contains.

2.4.1 Create CRM Contact Save Action

CRM Contacts are the main way to keep a track of a business' individual customers.

The Create CRM Contact Save Action can be used to create and update existing contacts.

- The action checks whether the contact already exists, using the contact's unique identifier.
- If the record does not exist, it is created.
- If the already exists, the information in the selected fields in the CRM is updated.

Whether the fields are updated or the information is written to an audit field depends on the setting in the **Overwrite user field if it already contains a value** field. For more information about audit fields, see Updating Existing Fields and Audit Information.

The unique identifier of the CRM contact is set to the CRM e-mail field by default.

To set the unique identifier of the CRM contact, in the *Create CRM Contact Save Action* template, in the **CRM** group, change the value in the **Primary Field** field.

This is stored in `/sitecore/system/modules/Web Forms for Marketers/Settings/Actions/Save Actions/Create CRM Contact`

2.4.2 Create CRM Account Save Action

CRM Accounts are the main way to keep a track of a company's partners and business to business clients. The Create CRM Account Save Action can be used to create and update existing accounts.

- The save action checks whether the account already exists, using the unique identifier.
- If the record does not exist, it is created.
- If the already exists the information in the selected fields in the CRM is updated.

Whether the fields are updated or the information is written into an audit field depends on the setting in the **Overwrite user field if it already contains a value** field. For more information about audit fields, see Updating Existing Fields and Audit Information.

The unique identifier of the CRM account is set to the account name by default.

To set the unique identifier of the CRM contact, in the *Create CRM Contact Save Action* template, in the **CRM** group, change the value in the **Primary Field** field.

This is stored in `/sitecore/system/modules/Web Forms for Marketers/Settings/Actions/Save Actions/Create CRM Contact`

2.4.3 Create CRM Entity Save Action

A CRM system can have a large and varied number of entities. The Create CRM Entity Save Action can be used to create any CRM entity that is present in the CRM instance to which it is connected.

The Create CRM Entity creates new records, but does not check for existing records.

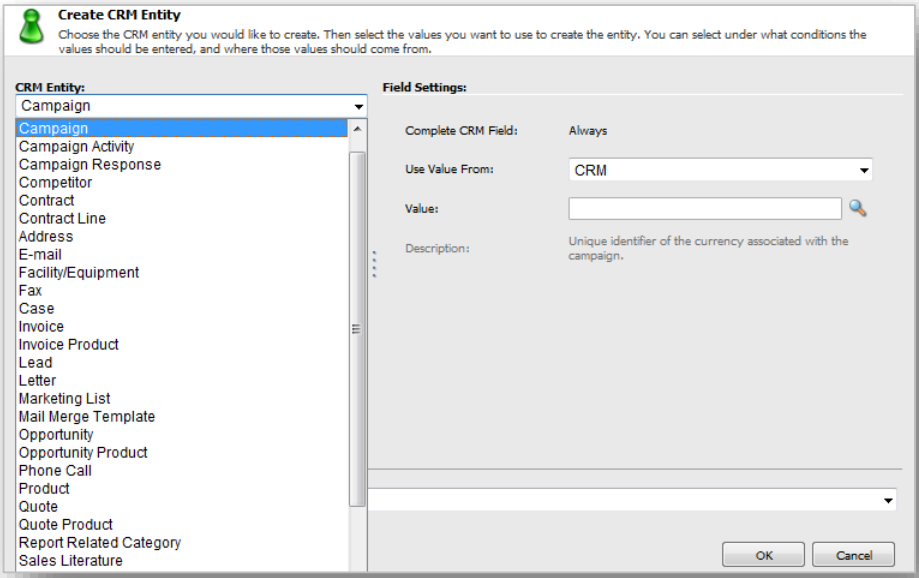
To select the type of entity that you wish to create:

1. Click the **CRM Entity** dropdown box. All the available entities in the CRM system that you are connected to are displayed.

2. Select the type of entity for which you want to create a record.

   The required fields are displayed automatically. You can add all the other available fields.
3. Select and configure the desired fields.
4. Click **OK**.

Here is an example where you decide to create a new campaign record in CRM:



Audit information can also be registered when you create records to indicate that the record was created by the CRM Save Action. This also tells you when the record was created and which fields were created automatically by the save action.

The Create CRM Entity Save Action cannot update fields in existing records.

For more information about audit fields, see Updating Existing Fields and Audit Information.

Chapter 3 Tips for Developers

This chapter describes some things that developers might find useful when they are working with dynamics CRM Save Actions.

This chapter contains the following sections:

3.1 The AuditRender Pipeline

Audit information can be parsed using a custom processor. For example, if you want to keep only 5000 characters of the audit message in the audit storage due to a system character limit, you could use a custom processer that looks like this:

```
namespace Sitecore.Form.Core.Pipelines.AuditRender
{
  public class ShrinkAuditMesssages
  {
    /// <summary>
    /// Processes the specified args.
    /// </summary>
    /// <param name="args">The args.</param>
    public void Process(AuditPipelineArgs args)
    {
      Assert.ArgumentNotNull(args, "args");

      if (args.Current.Length > 5000)
      {
        args.Current.Remove(0, args.Current.Length - 5000);
      }
    }
  }
}
```

In order to register the processor, add it to `configuration/sitecore/pipelines/auditRender` pipeline.

```
<auditRender>
  <processor type="Sitecore.Form.Core.Pipelines.AuditRender.AuditTimeStamp, Sitecore.Forms.Core"/>
  <processor type="Sitecore.Form.Core.Pipelines.AuditRender.AuditUpdatedTitle, Sitecore.Forms.Core"/>
  <processor type="Sitecore.Form.Core.Pipelines.AuditRender.AuditUpdatedEntities, Sitecore.Forms.Core"/>
  <processor type="Sitecore.Form.Core.Pipelines.AuditRender.AuditSkippedTitle, Sitecore.Forms.Core"/>
  <processor type="Sitecore.Form.Core.Pipelines.AuditRender.AuditSkippedEntities, Sitecore.Forms.Core"/>
  <processor type="Sitecore.Form.Core.Pipelines.AuditRender.AuditMesssages, Sitecore.Forms.Core"/>
  <processor type="Sitecore.Form.Core.Pipelines.AuditRender.ShrinkAuditMesssages, Sitecore.Forms.Core"/>
</auditRender>
```