Sage CRM 7.3 SP2

Troubleshooting Guide

Revision: TBL-MAN-ENG-73SP2-1.0

Updated: April 2016

Contents

Chapter 1: About this guide

This guide is for Sage CRM system administrators who need to troubleshoot issues with Sage CRM.

This guide refers to *Sage CRM* but your system might have a different brand name, such as *Sage 200 Sales and Marketing*. The system works in the same way regardless of its name. The functionality that's available to you depends on the modules that you're licensed to use.

Chapter summary

Chapter	**Description**
Classic Outlook Integration	How to troubleshoot the Classic Outlook Plugin and Classic Outlook Integration.
Lite Outlook Integration	How to troubleshooting the Lite Outlook Plugin and Lite Outlook Integration.

Chapter 2: Classic Outlook Integration

Installing the Classic Outlook Plugin

When installing the Classic Outlook Plugin, the installation process fails half way through with the following error message:

"Could not obtain user's interface language. Try again?"

- In IE, go to **Tools | Internet Options | Security | Trusted Sites** and add the Sage CRM web site to the list of trusted sites.
- If possible, disable antivirus software and firewalls while you install the plugin.

Uninstalling the Classic Outlook Plugin

1. Back up the system registry before making any modifications to the registry keys.
2. Open the registry editor (regedit.exe).
3. Delete the following keys if they exist.
    - HKEY_CURRENT_USER\Software\ACCPAC\CRM
    - HKEY_CURRENT_USER\Software\Microsoft\Office\Outlook\Addins\SageCrmOutlookAddIn2010
    - HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\ACCPAC\CRM
    - HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Microsoft\Code Store Database\Distribution Units\{0AFD9937-10D5-436F-9F2B-08BF61754446}
    - HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Microsoft\Windows\CurrentVersion\ModuleUsage\C:/Windows/Downloaded Program Files/OtlTools.ocx
    - HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\SharedDlls\crmoutlookplugin*.*
    - HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows Messaging Subsystem\MSMapiApps\crmoutlookplugin*.*

Viewing the Classic Outlook Plugin logs

- If there's a problem when installing the plugin, check ...Temp\crmclientinstallLog for details.
- If any errors occur when you run the plugin, check %UserProfile%\Users\<user name>\AppData\Roaming\Sage\CRM\OutlookPlugin\errorsdn.svcLog for details. The best way to view the svcLog file is with the Service Trace Viewer Tool. You can download it here: https://msdn.microsoft.com/en-us/library/ms732023(v=vs.110).aspx

- If there are issues with the Classic Outlook plugin, you should manually uninstall it. Also uninstall the Lite Outlook plugin if it's on the computer, and then install the Classic Outlook plugin.

Using Classic Outlook Integration with Auto Login

The recommended setting when using IIS Auto Login is to disable IIS Anonymous Authentication on the CRM web server. Classic Outlook Integration doesn't work when Anonymous Authentication is disabled and the user is prompted for a username and password when trying to use the Integration. The following workaround lets you use Classic Outlook Integration with Auto Login. The client sends out one or two additional HTTP requests in the background, but no further user interaction is required.

1. Open IIS Manager (inetmgr.exe), expand the CRM virtual directory and double-click **URL Rewrite**.
2. Click **Add Rule(s)**.
3. In Inbound rules, double-click **Blank rule** and name the rule *Outlook plugin - force Windows authentication*.
4. In the Match URL section:
   a. Set **Requested URL** to **Matches the Pattern** .
   b. Set **Using** to **Regular Expressions**.
5. Set **Pattern** to **eware.dll/go* and select **Ignore case**.
6. In Conditions, set **Logical Grouping** to **Match All**.
7. Add a condition with an input of {QUERY_STRING}.
   a. Set **Check if input string** to **Matches the Pattern**.
   b. Set **Pattern** to *OutlookAction=logon*.
   c. Select **Ignore case**.
8. Add a condition with an input of {HTTP_AUTHORIZATION}.
   a. Set **Check if input string** to **Does Not Match the Pattern**.
   b. Set **Pattern** to *Negotiate.**.
   c. Select **Ignore case**.
9. Add a condition with an input of {HTTP_AUTHORIZATION}.
   a. Set **Check if input string** to **Does Not Match the Pattern**.
   b. Set **Pattern** to *NTLM.**.
   c. Select **Ignore case**.
10. In Action, set **Action type** to **Custom Response**.
11. In Action Properties:
    a. Set **Status code** to *401*.
    b. Set **Substatus code** to *0*.
    c. Set **Reason** to *Unauthorized - plugin auth*.
    d. Set **Error description** to *Outlook plugin must use Windows authentication*.
12. Click **Apply**.

Chapter 3: Lite Outlook Integration

Installing the Lite Outlook Plugin

If a security certificate is missing on a user's machine, an error message is displayed when the user tries to install the Lite Outlook Plugin.

1. Uninstall the plugin.
2. Open a browser and go to Thawte Root Certificates (http://www.thawte.com/roots/).
3. Click **Download root package**.
4. When the certificates are installed, run the Lite Outlook Plugin Installer as normal.

Uninstalling the Lite Outlook Plugin

1. Back up the system registry before making any modifications to the registry keys.
2. Open the registry editor (regedit.exe).
3. Delete the following keys if they exist.
   - HKEY_CURRENT_ USER\Software\Microsoft\Office\Outlook\Addins\SageCrmOutlookAddIn2010ForExchange
   - HKEY_CURRENT_USER\Software\Sage
   - HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Microsoft\Code Store Database\Distribution Units\{0AFD9937-10D5-436F-9F2B-08BF61754446}
   - HKEY_LOCAL_ MACHINE\SOFTWARE\Wow6432Node\Microsoft\Windows\CurrentVersion\ModuleUsage\C:/Windows/Downloaded Program Files/OtlTools.ocx
   - HKEY_LOCAL_ MACHINE\SOFTWARE\Wow6432Node\Microsoft\Windows\CurrentVersion\SharedDLLs (C:\Windows\Downloaded Program Files\OtlTools.ocx)
   - HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Sage\Sage CRM\Outlook Plugin

Working with Lite Outlook Integration

When using Lite Outlook Integration, if an error occurs with the CRM Server, do the following:.

1. Log on to Sage CRM.
2. Check the Sage CRM URL and note if the protocol is HTTPS or HTTP.
3. Back up the system registry before making any modifications to the registry keys.
4. Open the registry editor (regedit.exe).
5. Open HKEY_CURRENT_USER\Software\ACCPAC\CRM\OutlookPlugin.
6. Update the OutlookPort value to 80 for HTTP and 443 for HTTPS.
7. Restart Outlook.

Filing emails with Lite Outlook Integration

When a user tries to file an email from a sender that does not have a corresponding record in Sage CRM, the following Javascript error occurs.

Unable to get value of the property 'value': object is null or undefined

This issue is caused by missing values for cmli_comm_companyid and _HIDDENcmli_comm_companyidTEXT in the getSelection function.

1. Click **<My Profile> | Administration | Customization | Translations**.
2. Delete FileIT in **filecrm** caption and add the following instead.

```
E-mail Target
<script>
window.attachEvent("onload", function () {
      if (document.location.href.indexOf("Act=6011")>-1) {
            if (!getElm("cmli_comm_companyid")&&!getElm("_HIDDENcmli_comm_
companyidTEXT")) {
                  var dummy1 = document.createElement("input");
                  with (dummy1) { type = "hidden"; id = "cmli_comm_
companyid"; value = ""; }

                  var dummy2 = document.createElement("input");
                  with (dummy2) { type = "hidden"; id = "_HIDDENcmli_comm_
companyidTEXT"; value = "";}

                  getElm("_Datacmli_comm_personid").appendChild(dummy1);
                  getElm("_Datacmli_comm_personid").appendChild(dummy2);
            }
      }
});
function getElm (a) { return document.getElementById(a);}
</script>
```

Relinking orphaned email records

If a user files an Outlook email in Sage CRM but does not link the email to a Person or Company, the resulting Communication record becomes an orphan record in Sage CRM and is difficult to find.

To solve this issue, you can create an Interactive dashboard gadget to display orphan Communication records, a flag to identify orphan records, and a table level script to link an orphan Communication record to an existing Person or Company record.

1. Click **<My Profile> | Administration | Customization | Primary Entities | Communication | Views** and click **New** to create a view that filters on orphaned Communication records.
    a. Name the view *vCommOrphanedEmail*.
    b. To make the view available when creating a new gadget, select **Reports View**.
    c. Enter a short description and a translation.
    d. Enter the following in **View Script**.
    ```
    create view vCommOrphanedEmail as

    select * from vcommunication

    where comm_type = N'Email' and Comm_Action = N'EmailIn'

    and CmLi_Comm_PersonID is null and CmLi_Comm_CompanyID is null

    and Comm_Deleted is null

    and CmLi_Comm_UserID is not null
    ```
    e. Click **Save**.
2. Click **Reports | General** and click **New** to create a report that displays orphaned Communication records.
    a. Name the report *Orphaned Emails*.
    b. Choose **vCommOrphanedEmail** as the source view.
    c. Select **Filter by Current User** and **Filter by User's Primary Team**.
    d. Add **Subject**, **From**, and **To** columns to the report. Add any other columns you want to display on the dashboard.
    e. Click **Continue** and then click **Save**.
3. Click **My CRM | Dashboards** and open the interactive dashboard to which you want to add the new gadget.
    a. Click **New Gadget | Create Gadget** and select **List**.
    b. Choose the **Communication** entity. Click **Next** to proceed through the wizard.
    c. Choose the **Orphaned Emails report** as the data source.
    d. Select **Go To Summary Screen** from **Drill Down**.
    e. Name the gadget and click **Finish**.
4. Click **<My Profile> | Administration | Customization | Primary Entities | Communication | Fields** and click **New** to create a field that acts as a flag on a filed email.
    a. Select **Adv Search Select** from **Entry Type**.
    b. Enter *comm_linkorphan* in **Column Name**.
    c. Enter *Link Orphan* in **Caption**.
    d. Select **Person** from **Search Entity**. **Note:** To link an orphan Communication record to a Company

record, select **Company** instead.

e. Click **Save**.

5. Click **<My Profile> | Administration | Customization | Primary Entities | Communication | Screens** to add the new field to the Email Filing screen.
   a. Click the pencil icon beside EmailFilingBox.
   b. Select **comm_linkorphan** from **Field**.
   c. Enter the following in **Create Script** to hide the field for emails that are not orphans:

```
if ((CRM.GetContextInfo("Communication","comm_type")=='Email')
&& (CRM.GetContextInfo("Communication","Comm_Action")=='EmailIn')
&& (CRM.GetContextInfo("Communication","CmLi_Comm_PersonID")=="")
&& (CRM.GetContextInfo("Communication","CmLi_Comm_CompanyID")=="")) {
Hidden = false;
}
else Hidden = true;
```

   d. Click **Add** and then click **Save**.

6. Click **<My Profile> | Administration | Customization | Primary Entities | Communication | TableScripts** to create a table level script on Communication that checks whether there's a value in comm_linkorphan.
   a. Click **New**.
   b. Add the following to **Table level script**. If there's a value in the comm_linkorphan field, the script links the Communication record to the Person entity and clears the existing comm_linkorphan value. **Note:** To link an orphan Communication record to a Company record, replace CmLi_Comm_PersonId with CmLi_Comm_CompanyId in the script below.

```
function UpdateRecord () {
if (IsValid(Values("comm_linkorphan"))) {
var PrimaryCommLink = CRM.FindRecord("Comm_Link",
"CmLi_Comm_Communicationid="
+ WhereClause.replace(/comm_communicationid\=(\d*).*/i,"$1")
+ " and CmLi_Comm_UserId is not null");
PrimaryCommLink.item("CmLi_Comm_PersonId") = Values("comm_linkorphan");
PrimaryCommLink.SaveChanges();
Values("comm_linkorphan") = "";
}
}
function IsValid(str) {
str = str + "";
if ((str!="undefined") && (str != "null") && (str != "")) {
return true;
```

```
    }
    else return false;
    }
```

c. Click **Save**.

Lite Outlook Integration load balancing

In a multi-server environment, the Lite Outlook Plugin should connect to Load Balancing and identify which server is available. Instead, it reads the name of the server from the time of installation and adds it to the Windows registry. The plugin reads this key each time it tries to connect to the Sage CRM database.

To work around this issue:

1. Update the Windows Registry key HKEY_CURRENT_USER\Software\Sage\Lite Outlook Plugin\ to the correct server name.
2. During installation, change the server name to the main server name rather than the backup server name to ensure service integrity.