Sage CRM 2017 R1
Troubleshooting Guide

Revision: TBL-MAN-ENG-2017R1-1.0
Updated: November 2016

Contents

About this guide

This guide is for Sage CRM system administrators who need to troubleshoot issues with Sage CRM. The navigation instructions in the guide assume that you're using the Contemporary Theme.

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.

Email

- Granting SendAs right to an SMTP user
- Email Management
- Formatting an email in Outlook
- Classic Outlook Integration
- Lite Outlook Integration
- Exchange Integration

Granting SendAs right to an SMTP user

Sage CRM is unable to send emails from certain users, it can only send emails to email addresses that are on the same domain as the sender, or all emails sent from Sage CRM appear to be from the same address.

This issue can be caused by the Sage CRM SMTP user having insufficient rights on the Exchange mail server to send emails using a different email address.

To send emails from Sage CRM as any user, you must enable the SendAs right on the Exchange server using PowerShell.

> **Note:** PowerShell is installed on your client. You don't need to install the Exchange remote Powershell tools.

1. To connect to Exchange 2010 and later, run the following command:
   `Set-ExecutionPolicy Unrestricted`
2. Confirm that the following setting should be enabled:
   `$LiveCred = Get-Credential`
3. Enter the credentials for an Exchange administrator. The Exchange connection URL shown below (`https://ps.outlook.com/powershell/`) is for Exchange Online, but you can substitute your own Exchange CAS server, or an Exchange instance hosted by a third party.
   ```
   $Session = New-PSSession -ConfigurationName Microsoft.Exchange -
   ConnectionUri https://ps.outlook.com/powershell/ -Credential
   $LiveCred -Authentication Basic -AllowRedirection
   Import-PSSession $Session
   Enable-OrganizationCustomization
   ```
   This command may take a few moments to complete.
4. When you're connected to the Exchange instance, you can grant the SendAs right in one of the following ways:
   - Add the Exchange users to an active directory group or distribution group and grant the SendAs right to the group. This is a good option if you plan to do something else with these users. See Granting SendAs right to a group.
   - Grant the SendAs right on all mailboxes. This approach is quick but applicable only for smaller organizations where all users use Sage CRM. See Granting SendAs right to all mailboxes.

> **Note:** Use the pipe operator (|) to get an array of mailbox objects and send them to your command. This allows you to quickly grant the SendAs right over a large number of mailboxes, without selecting them all individually.

Granting SendAs right to a group

1. Create a distribution list called *CRM Users* containing two Exchange users who also use Sage CRM. The user email addresses in the example below are `ebdenm@panoply-tech.com` and `mayes@panoply-tech.com`.
   ```
   New-DistributionGroup -Name "CRM Users" -IgnoreNamingPolicy
   "ebdenm@panoply-tech.com","mayes@panoply-tech.com" | %{ Add-
   DistributionGroupMember -Identity "CRM Users" -Member $_ }
   ```
2. Run the following command to list the members of the distribution group:
   ```
   Get-DistributionGroupMember "CRM Users"
   ```
   This returns the members:
   ```
   Name RecipientType
   ---- -------------
   ebdenm UserMailbox
   mayes UserMailbox
   ```
3. Send the members of the distribution group to the `Get-Mailbox` command and forward the array of resolved mailboxes to the `Add-MailboxPermission` cmdlet. The mailbox for the Sage CRM SMTP user is `crmuser@panoply-tech.com`:
   ```
   Get-DistributionGroupMember "CRM Users" | Get-Mailbox | Add-
   MailboxPermission -User crmuser@panoply-tech.com -AccessRights
   FullAccess
   ```
4. Set the SendAs permissions:
   ```
   Get-DistributionGroupMember "CRM Users" | Get-Mailbox | Add-
   RecipientPermission -AccessRights SendAs -Trustee crmuser@panoply-
   tech.com
   ```

Granting SendAs right to all mailboxes

Run the following commands. Do not filter by the distribution group:

```
Get-Mailbox | Add-MailboxPermission -User crmuser@panoply-tech.com -
AccessRights FullAccess
```

```
Get-Mailbox | Add-RecipientPermission -AccessRights SendAs -Trustee
crmuser@panoply-tech.com
```

Email Management

Email Management not filing outbound emails

If you can send emails from Sage CRM but Email Management doesn't file them as communications (or cases) in the Sage CRM database, do the following:

1. Ensure that you've configured the Mail Manager Filing Address and set a prefix in **<My Profile> | Administration | Email and Documents | Email Configuration**.
2. Check that you've created an entry for the Email Management Filing Address in **<My Profile> | Administration | Email and Documents | Email Management Server Options**.
3. To check that outbound email is sent to the Mail Manager Filing Address mailbox, disable Email Management on this mailbox. To do this, open the Email Management Filing Address entry in **Email Management Server Options** and deselect **Enabled**.
4. Send an email from Sage CRM and check the Mail Manager Filing Address mailbox for the email.
5. To re-enable Email Management on the mailbox, open the Email Management Filing Address entry in **Email Management Server Options** and select **Enabled**.
   If Email Management is running on the mailbox and reading emails successfully, the email you sent disappears from the mailbox. If it doesn't disappear, recheck the Email Management Filing Address, specifically the Email Account logon and password.
6. To enable debugging, click **<My Profile> | Administration | Email and Documents | Email Configuration**and choose **Yes** from **Debug**.
7. Check the log file (**yyyymmdd<installname>MailManager.log**) in **...\Program Files\Sage\CRM\Services\Logs**.
   If a *cannot log* message is displayed in the file, the Email Account logon and password you specified for the Email Management Filing Address are incorrect.
8. Check the RogueMails folder to ensure there's no erroneous formatting in the email (which can be generated by a bug in an email server or client). The folder is located in **...\Program Files\Sage\CRM\Services\CustomPages\Scripts\RogueMail**.

Email Management creating two communications

Email Management creates two communications in Sage CRM for each email it files if you're using the Communication template and you've set **Default Ruleset Action** to **Create a Communication**.

To stop Email Management creating two communications for each email it files:

1. Click **<My Profile> | Administration | Email and Documents | Email Management Server Options.**
2. Set **Default Ruleset Action** to **None**.

CDOSYS not sending emails out

If you choose **CDONTS/CDOSYS** from **<My Profile> | Administration | Email and Documents | Email Configuration | Send Mail Using** but can't send outbound emails, do the following:

1. To ensure CDOSYS is installed, open IIS and look for a **Virtual Server** folder. This folder is present only if CDOSYS is installed.
2. To check that CDOSYS is running, open IIS, right-click the **Default SMTP Virtual Server Folder** and click **Start**.
3. Open the **CDOSYS BadMail** folder. If there are emails in this folder, emails are not getting from the CDOSYS SMTP server to the mail server, or are being bounced back. If this is the case, open the emails to find the error.
   - The **From** address may not be configured for emails that are sent as notifications from Sage CRM. To correct this, click **<My Profile> | Administration | Advanced Customization | Workflow** and configure **Notify Email Name** and **Notify Email Address** .
   - The **To** address might not be configured. This is also possible for emails that are sent as notifications or from workflows. Check the bad email to see where it's coming from, then configure the email address.
4. Open the **CDOSYS Queue** and **Pickup** folders. If there are emails in these folders, CDOSYS is not running, or the mail server is not relaying emails. Ensure the mail server is configured to relay emails from the Sage CRM server.
5. To ensure that CDOSYS can find the mail server, open IIS Manager, right-click **Default SMTP Server** and select **Properties | Delivery/Advanced**. Enter the IP address of the mail server in **Smart Host** and restart CDOSYS.
6. Check that the mail server allows relaying. If it doesn’t, emails can be delivered to internal company mailboxes but not to external mailboxes. Ask the mail server administrator to enable relaying.

Formatting an email in Outlook

When a user files an email in Microsoft Outlook and then opens the email in Sage CRM, the email body may contain incorrect formatting such as extra paragraphs, lines, and spaces.

This issue is caused by the default settings in the CKEditor configuration file. As a result, CKEditor overrides the original Outlook formatting and extra paragraphs, lines, and spaces may appear in filed emails.

CKEditor is a third-party component used by Sage CRM to process emails. For more information about the CKEditor configuration file, go to http://docs.ckeditor.com/#!/guide/dev_configuration.

To fix this issue, edit the CKEditor configuration file on your Sage CRM server:

1. Locate the **Config.js** file.
   Below are the default locations of the file:
   - On a 32-bit system: **%ProgramFiles%\Sage\CRM\CRM\WWWRoot\ckeditor**
   - On a 64-bit system: **%ProgramFiles(x86)%\Sage\CRM\CRM\WWWRoot\ckeditor**
2. Open **Config.js** in a text editor such as Notepad.
3. To prevent the insertion of extra paragraphs, add the following code immediately above the line `config.toolbar = "Full";`:

   ```
   config.autoParagraph = false;
   ```

4. To prevent the insertion of extra spaces and disable default CKEditor styling, append the following code to the file:

   ```
   config.allowedContent =
   {
       $1:
       {
           elements: CKEDITOR.dtd,
           attributes: true,
           styles: true,
           classes: true
       }
   };
   ```

   ```
   config.contentsCss=[];
   ```

5. Save your changes and close the file. As a result, the **Config.js** file should look as shown in Sample Config.js file.
6. Open IIS Manager and restart the Sage CRM web site.

Sample Config.js file

After you've edited the **Config.js** file as described in Formatting an email in Outlook, the file should look as follows:

```
CKEDITOR.editorConfig = function( config )
{
    config.autoParagraph = false;
    config.toolbar = "Full";
    config.disableNativeSpellChecker = false;
    config.resize_enabled = false;
    config.fillEmptyBlocks = false;
    config.pasteFromWordRemoveFontStyles = false;
    config.pasteFromWordRemoveFontStyles = false;
    config.extraAllowedContent = 'img(*){*}[*]';
    config.toolbar = [
    {
        name: 'document',
        groups: [ 'mode', 'document', 'doctools' ],
        items: [ 'Source', '-', 'Preview', 'Print','Templates']
    },
    {
        name: 'clipboard',
        groups: [ 'clipboard', 'undo' ],
        items: [ 'Cut', 'Copy', 'Paste', 'PasteText', 'PasteFromWord',
        '-', 'Undo', 'Redo' ]
    },
    {
        name: 'editing',
        groups: [ 'find', 'selection', 'spellchecker' ],
        items: [ 'Find', 'Replace', '-', 'SelectAll']
    },
    {
        name: 'basicstyles',
        groups: [ 'basicstyles', 'cleanup' ],
        items: [ 'Bold', 'Italic', 'Underline', 'Strike', 'Subscript',
        'Superscript', '-', 'RemoveFormat' ]
    },
    {
        name: 'links',
        items: [ 'Link', 'Unlink', 'Anchor' ]
    },
    {
        name: 'insert',
        items: [ 'Table', 'HorizontalRule', 'SpecialChar',
        'PageBreak','Maximize']
    },
    '/',
```

```
    {
        name: 'paragraph',
        groups: [ 'list', 'indent', 'blocks', 'align', 'bidi' ],
        items: [ 'NumberedList', 'BulletedList', '-', 'Outdent',
        'Indent', '-', 'Blockquote', 'CreateDiv', '-',
        'JustifyLeft', 'JustifyCenter', 'JustifyRight',
        'JustifyBlock', '-', 'BidiLtr', 'BidiRtl' ]
    },
    {
        name: 'styles',
        items: [ 'Styles', 'Format', 'Font', 'FontSize' ]
    },
    {
        name: 'colors',
        items: [ 'TextColor', 'BGColor' ]
    }
    ];
    config.fontSize_sizes =
    '8/8pt;9/9pt;10/10pt;11/11pt;12/12pt;14/14pt;16/16pt;
    18/18pt;20/20pt;22/22pt;24/24pt;26/26pt;28/28pt;36/36pt;
    48/48pt;72/72pt';

    config.allowedContent =
    {
         $1:
         {
              elements: CKEDITOR.dtd,
              attributes: true,
              styles: true,
              classes: true
         }
};

config.contentsCss=[];
```

Classic Outlook Integration

- Installing the Classic Outlook Plugin
- Uninstalling the Classic Outlook Plugin
- Viewing the Classic Outlook Plugin logs
- Using Classic Outlook Integration with Auto Login

Installing the Classic Outlook Plugin

When installing the Classic Outlook Plugin, the installation process fails half way through and an error message is displayed.

- 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 user name 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**.

Lite Outlook Integration

- Installing the Lite Outlook Plugin
- Uninstalling the Lite Outlook Plugin
- Working with Lite Outlook Integration
- Filing emails with Lite Outlook Integration
- Relinking orphaned email records
- Load balancing 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**.
   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.
   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**.

Load balancing Lite Outlook Integration

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.

Exchange Integration

Connecting to Exchange Server

Issue	Solution
No Impersonation rights	The Exchange Server User Name specified in Exchange Server Connection Setting requires impersonation rights in Exchange. Check the user rights in Exchange.
Cannot access EWS URL	If the EWS URL cannot be accessed, the connection cannot be set up. Paste the EWS URL into a browser and check it can be accessed using the impersonation user’s username and password.
Wrong authentication settings on EWS folder	If the EWS folder in IIS on Exchange Server has the wrong authentication settings, the connection cannot be set up. For more information, see *Configuring Exchange Server* in the *System Adminstrator Help*.
Tomcat is not running	The Tomcat service is required to establish the Exchange connection and to run the Sync Engine. If the Tomcat service is not running, errors usually occur in other areas of Sage CRM that rely on this service. For example, the Interactive Dashboard. Restart the Tomcat service if it has stopped. Alternatively, you can schedule a Tomcat service restart.
jdbc.properties or syncengine.properties contain incorrect server or port information	The jdbc.properties file contains the wrong server name or port number, or the syncengine.properties file contains the wrong server name. These files are typically located in **..\Program Files\Sage\CRM\[installname]\tomcat\webapps\[installname]ExchangeSyncEngine\WEB-INF**.

Using a non-default IIS port

If you change the default port for Sage CRM on IIS to any port other than port 80 and then set up Exchange Integration, the following error is displayed.

CRM is attemping to establish a connection with the Sync Engine. Please wait..

The Exchange Server connection could not be saved as CRM was unable to connect to the Sync Engine.

To resolve this issue:

1. Click **<My Profile>| Administration | Email and Documents | Exchange Server Integration | Connection Management**.
2. Set **Use Default Sync Engine Location** to **No**.
3. Enter the server name and port number separated by a colon(:) in **Sync Engine Location**. For example, *http://SERVER:8080/sdata/crmExchangeSyncEngine/crmExchange/-/*.

Synchronizing contacts

When you enable contact synchronization in Exchange Integration, the contacts are added to the **EcngSyncResource** table. They are then moved to the **EWSSyncResource** table and the number of contacts decreases as they're processed.

However, running the following SQL statement shows that x number of records have been added to the **EcngSyncResource** table but they have not been successfully moved to the **EWSSyncResource** table.

```
select * from EcngSyncResource with (NOLOCK)

where exsr_UUID not in (select EWRS_UUID from EWSSyncResource WITH
(NOLOCK)) and exsr_Deleted is null and exsr_processed = 0 and exsr_
EndpointID in (6009,6014)
```

To troubleshoot this issue, update one of the contacts before performing a synchronization to check if it processed. Also check the log files for any errors or conflicts when processing the records.

> **Note:** The **exsr_NoUpdateToRecord** column is set to **1** during Exchange synchronization if the record has not been updated since the last synchronization session.

Enabling logs to identify errors

Enable the following logs to help identify the cause of Exchange Integration failures.

- Use the **IIS log** to check if a request reached IIS Web Server. Enable this log in IIS Manager.
- Use the **CRMRewriter log** to analyze what happened to a request that was sent to the Tomcat web application. To enable this log:
  a. Open **<Program Files>\Sage\Services\IISUtils\CRMRewriter\CRM.Rewriter.rules**.
  b. Change
     ```
     #RewriteLog C:\Program
     Files\Sage\CRM\Services\IISUtils\CRMRewriter\rewriter.log
     #RewriteLogLevel 2
     ```
     to
     ```
     RewriteLog C:\Program
     ```

```
Files\Sage\CRM\Services\IISUtils\CRMRewriter\rewriter.log
RewriteLogLevel 5
```

- Use the **Tomcat access log** to establish if the request reached Tomcat server. To enable this:
  a. Open **<installed instance dir>\tomcat\conf\server.xml**.
  b. Change
  ```
  <!--
  <Valve className="org.apache.catalina.valves.AccessLogValve"
  directory="logs"
  prefix="localhost_access_log." suffix=".txt" pattern="common"
  resolveHosts="false"/>
  -->
  ```
  To
  ```
  <Valve className="org.apache.catalina.valves.AccessLogValve"
  directory="logs"
   prefix="localhost_access_log." suffix=".txt" pattern="common"
  resolveHosts="false"/>
  ```
- Use the detailed **Sync Engine log** to view detailed information about what happened in the Sync Engine, including detailed information about requests made from Sync Engine. To enable this:
  a. Open **<installed instance dir>\tomcat\webapps\<Sync Engine folder>\WEB-INF\log4j.properties**.
  b. Change
  ```
  # defaultLog - default catch-all
  log4j.rootLogger=ERROR, defaultLog
  ```
  To
  ```
  log4j.logger.httpclient.wire.header=DEBUG
  log4j.logger.org.apache.commons.httpclient=DEBUG
  # defaultLog - default catch-all
  log4j.rootLogger=ALL, defaultLog
  ```

System customization

Updating the name of a workflow rule

When you update the name of a workflow rule, it is not updated in the user interface.

When you create a workflow rule, Sage CRM inserts the rule name in both the **Translation** and the **Caption Code** fields. If you update a workflow rule name in **<My Profile> | Administration | Advanced Customization | Workflow**, only the **Caption Code** field is updated. To update the **Translation** field, do the following.

1. Click **<My Profile> | Administration | Customization | Translations**.
2. Enter *WorkflowRule* in **Caption Family**.
3. Enter *<your new workflow rule>* in **Caption Code**.
4. Click **Find**.
5. Click the caption code and click **Change**.
6. Enter the new workflow rule name in the **Translations** fields and click **Save**.

Creating a lead using Web to Lead

When you create a lead using Web to Lead, the Lead workflow action buttons are not displayed. Web to Lead automatically tries to save the lead in the Lead workflow and is unable to find a primary rule.

> **Note:** When you create a new lead directly in Sage CRM, the Lead workflow action buttons are displayed.

1. Open the Lead workflow.
2. Open the primary rule (probably called **New Lead**) and ensure **Table or view** is set to **Lead**.

System configuration

- Internet Explorer Clear button
- Viewing Quick Find logs
- Restarting the Quick Find service
- Reinstalling the Quick Find service
- Changing the Quick Find service port
- Rebuilding the Quick Find index
- Using Quick Find with a 64-bit JRE
- Restarting the CRM Indexer Service

Internet Explorer Clear button

When you view Sage CRM in Internet Explorer, the **Clear** ([X]) button on small input fields lets you clear the field contents. The button is controlled by an IE pseudo-element and is displayed based on a combination of the element width, font size, and padding. For more information about the -ms-clear pseudo-element, see https://msdn.microsoft.com/en-us/library/windows/apps/hh465740.aspx.

Depending on the version of Sage CRM and Internet Explorer that you're using, the button might not appear. You can use a Sage CRM CSS workaround to ensure the button always appears. This example displays the **Clear** button on date picker time fields in Sage CRM when using the Color theme. Add this to **\CRM\WWWRoot\Themes\color1.css**.

```
.EDIT.hasDatepicker {

      font-family: Tahoma, Arial; font-size: 11px;

}

.EDIT[name$="_TIME"] {

      font-family: Arial; font-size: 11px;

}
```

Viewing Quick Find logs

Quick Find logs for a particular Sage CRM installation are located in **\CRM\Logs** and called **yyyymmddkeywordSearch.log**.

Quick Find service logs are located in **\CRM\Services\Logs\QuickFind**.

To set logging levels, modify the level for the keyWordSearch logger in **log4j.xml** in **\CRM\tomcat\webapps\crmj\WEB-INF**. The most useful settings are DEBUG and ERROR.

Restarting the Quick Find service

If an error occurs when you restart the Quick Find service from **<My Profile> | Administration | System | Quick Find**, you can restart the service on the server using the Services Control Panel, or by running the following commands from a command prompt:

```
net stop SageCMRQuickFindService
```

```
net start SageCRMQuickFindService
```

If you can restart the service from the Services Control Panel or the command prompt but not from Sage CRM, the Sage CRM user account probably does not have the Start/Stop Windows Services permission, or cannot access **net.exe**.

> **Note:** Restarting the service does not rebuild the index.

Reinstalling the Quick Find service

To reinstall the Quick Find service, go to **CRM\Services\QuickFind\bin** and run **SageCRMQuickFind-install-svc.cmd** as a user with local administrator rights. This removes the Windows service and recreates it using settings in the .cmd file. The service starts when it's recreated.

Changing the Quick Find service port

You can change the port used by the Quick Find service if it's used by another application on the Sage CRM web server.

By default, the Quick Find service listens on port 8983 on localhost. The Quick Find service is bound to localhost so it does not respond to requests originating from a different Sage CRM web server. This is necessary because the filtering of Quick Find results is carried out in another part of Sage CRM.

To change the port:

1. In Management Studio, configure the port and hostname in the `SolrEngineUrl` entry on the **Custom_Sysparams** table.
   ```
   UPDATE Custom_SysParams SET Parm_Value =
   N'http://<hostname>:<port>/solr/'
   WHERE Parm_Name = N'SolrEngineUrl'
   ```
   This entry contains a URL endpoint for the Quick Find service, such as
   `http://localhost:8983/solr/`
   Do not bind the Quick Find service to any IP address that's accessible remotely. On a multi-server installation of Sage CRM, each CRM web server hosts its own instance of the Quick Find service.
2. In Sage CRM, click **<My Profile> | Administration | System | Metadata**.
   - Select **Refresh System Parameters**.
   - Click **Execute Refresh**.
3. Use Notepad to open the Quick Find service installer file: **\CRM\Services\QuickFind\bin\SageCRMQuickFind-install-svc.cmd**. Configure the port and other options and save your changes.

- The Quick Find port parameter is in the `QUICKFIND_PORT` parameter setting.
- The host / IP address is in the `QUICKFIND_HOST` parameter.
- It is strongly advised that you do not change the `QUICKFIND_HOST` variable from localhost to a port accessible remotely from the Sage CRM web server.

4. Double-click **SageCRMQuickFind-install-svc** and run it as a local administrator. Alternatively, run it as a local administrator from a command prompt. The Quick Find service is stopped, uninstalled, and reinstalled with the new settings.
5. If you encounter an error starting the service, check the logs in **\CRM\Services\Logs\QuickFind**.

Rebuilding the Quick Find index

If the Quick Find index is corrupted, you can rebuild it by adding and removing an entity in **<My Profile> | Administration | System | Quick Find**.

> **Warning:** Rebuilding the full index can consume significant system resources if a large amount of data is added to the index.

To manually rebuild the index:

1. Stop the Quick Find service using the following command:
   `net stop SageCRMQuickFindService`
2. Delete the **\CRM\QuickFind** directory.
3. Clear the value for the `SolrLastIndexScan` entry in the **Custom_Sysparams** table.
4. Restart the Sage CRM Tomcat service, or refresh the system parameter and tables and columns metadata in **<My Profile> | Administration | System | Metadata**.
5. Start the Quick Find service using the following command:
   `net start SageCRMQuickFindService`
6. If the index is not rebuilt and **\CRM\QuickFind** is not recreated, do the following:
   - Start the Quick Find service from the Services Control Panel. For more information, see Restarting the Quick Find service.
   - Check the keywordSearch.log in **\CRM\Logs**.
   - Ensure the `ServerNames` value in the **Custom_Sysparams** table is correct. The Quick Find service starts on the first server in the list.

Using Quick Find with a 64-bit JRE

Quick Find uses the 32-bit Java Runtime Environment (JRE) supplied with Sage CRM. This limits the RAM available to the Quick Find service to approximately 1GB, which is sufficient for large Sage CRM databases (multiple tens of gigabytes).

You can use 64-bit JRE with the Quick Find service so the service can index more data.

> **Warning:** Change to a 64-bit JRE with caution and only if memory-related issues occur with the Quick Find service.

1. Install a 64-bit Java 8 JRE on the Sage CRM web server or another machine.
2. Copy the contents of the JRE folder to a new folder called **\CRM\Services\JRE\jre8x64**. Ensure the path to the x64 **Java.exe** is **\CRM\Services\JRE\jre8x64\bin\java.exe**.
3. Stop the Quick Find service if it's running.
4. Navigate to **\CRM\Services\QuickFind\bin** and make a copy of **SageCRMQuickFind-install-svc.cmd**.
5. Open **SageCRMQuickFind-install-svc.cmd** in a text editor.
6. Change the following settings:
   - `PR_JAVA_HOME`: Point this to the new JRE. For example, **%CRM_SERVICES_DIR%\JRE\jre8x64**.
   - `QUICKFIND_JVM`: There is no client mode for the x86 JRE. Point this to **%PR_JAVA_HOME%\bin\server\jvm.dll**.
   - `ALLOCATED_MEM`: The amount of memory to make available for the Quick Find service.
   - `SERVICE_EXE`: Select a service executable from **\CRM\Services\QuickFind\bin**.
7. Save and close **SageCRMQuickFind-install-svc.cmd**.
8. Double-click **SageCRMQuickFind-install-svc.cmd**. A command prompt window opens and displays the status of the Quick Find service. The service should start when you've completed these steps.

Restarting the CRM Indexer Service

The CRM Indexer Service that's used for Keyword Search stops. A message is displayed stating that the keyword search index has not been updated recently and the results may be incorrect.

1. Reset IIS.
2. Restart Sage CRM.
3. If the message persists, initialize the indexing service in one of the following ways:
   - Put field-level security on any primary entity text field. For example, deny rights. You can take the field-level security off the field immediately.
   - Select a Keyword Search view. For example, **vSearchListCompany**. Unselect the **Keyword Search** checkbox for the view and save. Then reselect the **Keyword Search** checkbox and save again.
   - Delete the **FullText.ix** folder from the Library. After a default interval of five minutes, the indexer service starts to build a full index. You can specify the gap between incremental indexes in **<My Profile> | Administration | System | Keyword Search | Interval**.

Mobile

- Configuring SSA fields on a mobile device
- Using Sage CRM Mobile

Configuring SSA fields on a mobile device

When you specify a restrictor field for an SSA field in **<My Profile> | Administration | Customization | Primary Entities / Secondary Entities | <Entity> | Restrictor Fields**, the restrictor field is not applied on a mobile theme.

You must also specify the restrictor field in the relevant mobile theme file.

1. Open the relevant mobile theme file using a text editor.
   - **WWWRoot\mobile\lib\ssa\ssarestrictors.js**
   - **WWWRoot\tablet\ssa\ssarestrictors.js**
   - **WWWRoot\mobile\SmartPhone\ssa\ssarestrictors.js**
2. Add the following line to the file:
   ```
   <ssa field containing the restriction>:['<field containing the filter value>','<field that's filtered>']
   ```
   For example:
   ```
   oppo_erpaccountid:['oppo_primarycompanyid','werp_companyid']
   ```
3. Save the file.

Using Sage CRM Mobile

If you experience problems while using Sage CRM Mobile from a smartphone or tablet, check the following troubleshooting tips.

Issue	Resolution
Sage CRM Mobile is not displayed	Ensure that the user agent for the mobile device is correctly mapped to the device in Sage CRM. The user agent might either be unassigned to a theme (device), or assigned to a different theme (device). For more information, see *Mapping a user agent* in the System Administrator Help.
Devices link is not available in Advanced Customization area	The mobile license for Sage CRM is not installed. Contact your Business Partner about obtaining a license key with the mobile option.

Issue	Resolution
Sage CRM Mobile is not displayed when Sage CRM is accessed through a link on the home screen	The mobile device sends a different user agent when a link is accessed from the home screen. Check unassigned user agents for a mobile device user agent, and map it to the correct theme (device).
Mobile device cannot locate the Sage CRM web server	Ensure that the Sage CRM installation is either publicly accessible or that the smartphone is connecting to a VPN which can access your Sage CRM account.
Default logon screen displayed instead of the Sage CRM Mobile screen	Sage CRM metadata has not yet been loaded. For example, due to an IIS reset. When you log on, Sage CRM Mobile is displayed as expected.

Installation

Running Sage CRM on Windows Server Essentials

After installation, complete the following steps to run Sage CRM on Windows Server Essentials.

1. Click **Control Panel | User Accounts** and disable User Account Control.
2. Launch Sage CRM.
3. If a service unavailable error occurs, set the bitness64 precondition in %windir%\system32\inetsrv\config\host.config.
   Change:
   ```
   <add name="PasswordExpiryModule"
   image="C:\Windows\system32\RpcProxy\RpcProxy.dll" />
   ```
   To:
   ```
   <add name="PasswordExpiryModule"
   image="C:\Windows\system32\RpcProxy\RpcProxy.dll"
   preCondition="bitness64"
   />
   ```
4. Reset IIS.
5. Launch Sage CRM. If an internal server error occurs, run the following command to turn off HTTP compression.
   ```
   %windir%\system32\inetsrv\appcmd.exe set config -
   section:system.webServer/httpCompression /-[name='xpress']
   ```
6. Reset IIS.
7. Launch Sage CRM.

Resetting the database logon password

If you need to change the database logon password, first change it in the SQL Server and then in Sage CRM.

1. In SQL Server, click the appropriate server.
2. Click the **Security** folder and then click **Logins**.
3. Right-click the relevant login, click **Properties** and change the password.
4. Go to the **Management** folder and right-click **SQL Server Agent**.
5. Click **Properties | Connection** and enter the new password.
6. Stop and start IIS. At the command prompt, type the following:
   net stop iisadmin /y

net start w3svc
net start msftpsvc

7. Log on to Sage CRM with system administrator rights.
8. Click **<My Profile> | Administration | System | Database** to view the log on to access the SQL database.
9. Click **Change**, enter a new password, and click **Save**.

Starting the CRM Indexer Service

The CRM Indexer Service feeds data to the CRM Keyword Search function. The service starts automatically but if it has been stopped and you want to start it again, you can do so manually.

1. Click **Control Panel | Administrative Tools | Services**.
2. Right-click **CRM Indexer Service** and click **Start**.

Uninstalling services

If the 2017 R1 Email Management Service and the Escalation Service are not correctly uninstalled when you restore a live environment, complete the following steps:

1. Click **Start | Run** and type *cmd*.
2. Browse to the folder where **EwareEmailManager.exe** is saved, type *eWareEmailManager /u*, and press **Enter**.
3. Browse to the folder where **CRMEscalationService.exe** is saved, type *CRMEscalationService /u*, and press **Enter**.

Re-installing services

To re-install the 7.3 Email Management Service and the Escalation Service when you restore a live environment, complete the following steps:

1. Click **Start | Run** and type *cmd*.
2. Browse to the folder where **EwareEmailManager.exe** is saved, type *eWareEmailManager /i* and press **Enter**.
3. Browse to the folder where **CRMEscalationService.exe** is saved, type *CRMEscalationService /i* and press **Enter**.

Redirector errors

It's possible to run Sage CRM 2017 R1 on a machine that's also running Sage CRM 7.2 although you should not do this unless it's required for testing an upgrade to 2017 R1.

Sage CRM 7.2 uses the Managed Fusion URL Rewriter and Sage CRM 2017 R1 uses the ARR and URL Rewriter modules of IIS. The Sage CRM 7.2 Rewriter is located in ..:\Program Files (x86) \Sage\CRM\Services\IISUtils\CRMRewriter. If you install Sage CRM 7.2 after Sage CRM 2017 R1, it overwrites the SData settings in IIS and removes the rewriter rules for Sage CRM 2017 R1.

To recreate the rules:

1. Run Sage CRM 2017 R1 **Setup.exe**.
2. Click **Change existing install**.
3. Select the Sage CRM 2017 R1 installation. The required IIS rewriter rules are recreated for SData in IIS. Requests for the Sage CRM 7.2 installation won't find a match in the IIS 7.x rewriter rules and will use the Managed Fusion URL Rewriter.