

OUR SCENARIO

SOFTWARE USED

Windows Server 2016	(Operating System)
Oracle Database 12c	(Third party application)
'Convert XLS'	(Softinterface application)
MS Office 2013	(Using Excel 2013 by OLE Automation by 'Convert XLS')

A new local OS user was created during Oracle Installation ('owner' of Oracle). The name of the user is: **oraclebase**

Softinterface app '**Convert XLS**' is called through OS_COMMAND package in Oracle DB.

oraclebase calls '**Convert XLS**'. '**Convert XLS**' then OLE automates MS Office (Excel in this case).

NOTE: In this example the third part app is Oracle. It is likely you are not using Oracle. *The solutions presented here are the same for any third party application issuing command line to Softinterface apps.*

USING SOFTINTERFACE APPLICATION

Our process makes command line calls to 'Convert XLS'. In this case some of the calls to 'Convert XLS' invoke MS Excel, and others don't. Those that do invoke MS Excel reveal the issues we found and solved. In our case we do:

1. Command line call to 'Convert XLS' to convert a text file to XLS. (This does not require Excel).
*NOTE: The output file must be saved to a MS Office [Trusted Location](#). If not, you will find opening the output file in MS Office will result in a **protected view** mode.*
2. Command line call to 'Convert XLS' again to merge the Excel file from step 1 with another Excel file containing the headers we need. This process **automates MS Excel!**

BACKGROUND INFORMATION ON MICROSOFT OFFICE APPLICATIONS INTENDED USAGES

Before reading, the solution, please take a look at how MS Applications are meant to be used with an interactively logged on user, and running them as a server-side process is causing issues (this is only one part of the problem, but still a good read to understand one of the issues).

<https://support.microsoft.com/en-us/help/257757/considerations-for-server-side-automation-of-office>

ISSUES ENCOUNTERED AND SOLUTIONS FOUND

Two separate and distinct errors messages were reported:

1. *"MS Excel is Required – Cannot Find Excel"*
This error is related to the fact that Excel SHOULD be run under an interactively logged on user on the Windows server. As we are calling the Softinterface app ('Convert XLS') through the third part app (Oracle) without anyone logged on the host, it leads to this error!

We used to get this error in step 2 above. Once corrected (solution further below) we started getting the second error message described next.

2. *Object variable or With block variable not set | Input File:*

This error message appeared in step 2 after solving the first issue. This error is related to the fact that Excel is used to open the file which was created in step 1 (remember that this file opens in PROTECTED VIEW !!!). If you open the file in Excel manually and you are alerted that it opens it in PROTECTED VIEW, you will notice that you cannot **Save As** this file. So our understanding is that the call to Excel to perform any operation was failing, as this function is *blocked* by Excel.

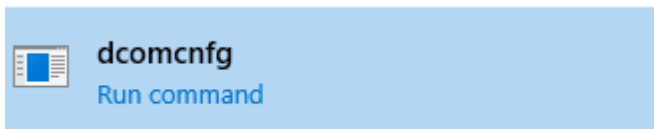
Solution

Issue 1 : *MS Excel Required- Cannot find excel*

To solve issue 1 we need to manually specify a user that Excel will run under! To do this we need to make some changes in the **DCOM** (Distributed Component Object Model), specifying that Excel will run as a specific user. In our case, we will setup Excel to run under the user that was used to install Oracle (oraclebase). Note that you need to make the changes in 2 places:

- In 'My Computer' properties
- The DCOM config for Microsoft Excel

Run **dcomcnfg**



'My Computer' properties

Expand Component Services → Computers

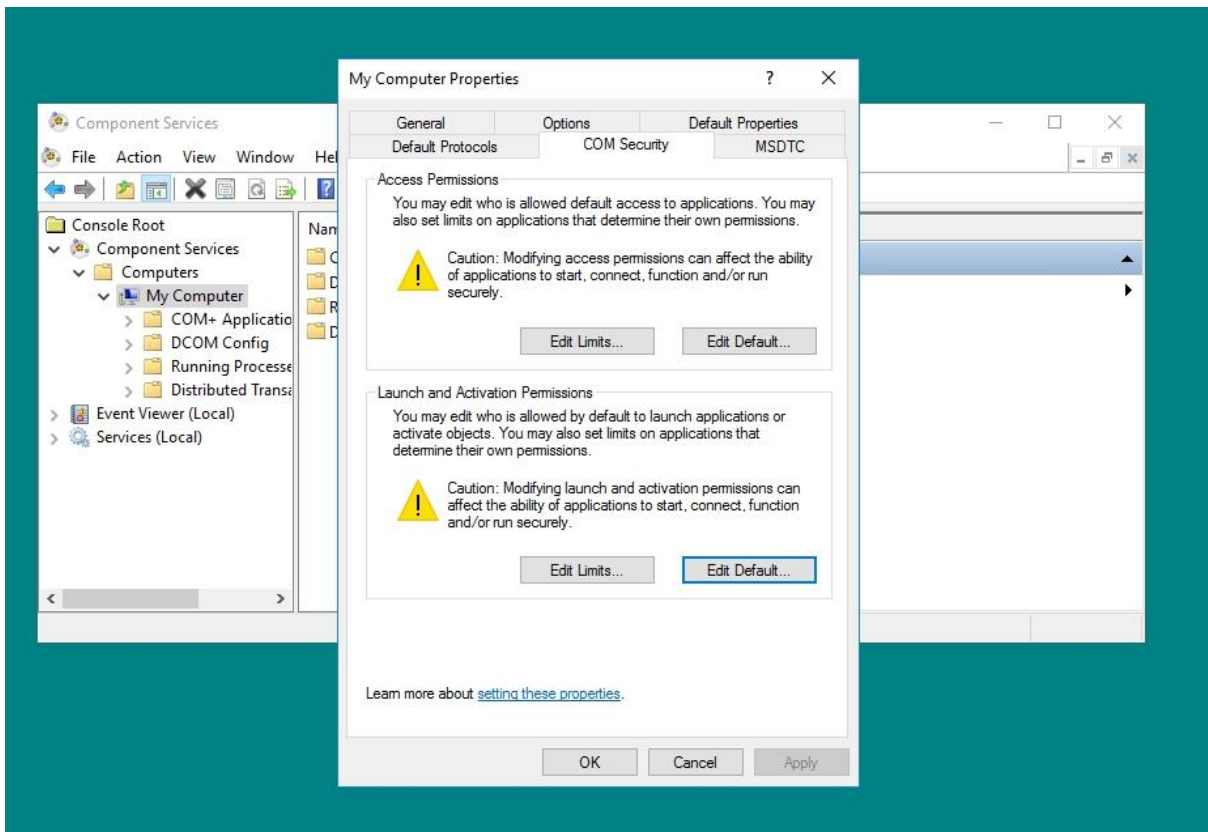
Right Click on **My Computer** → **Properties** → **COM security tab**

Open each dialog in **Access Permissions (Edit Limits, Edit Default)**

And **Launch and Activation Permissions (Edit Limits, Edit Default)**

And add oraclebase (or whichever user in your case) with full access in all the lists.

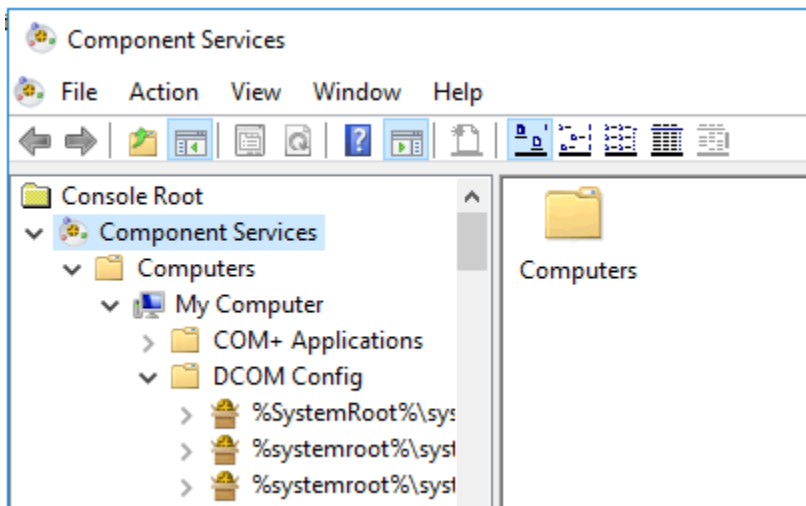
You might not need the user in all the lists, but add it anyway! better be on the safe side ☺



Microsoft Excel DCOM config

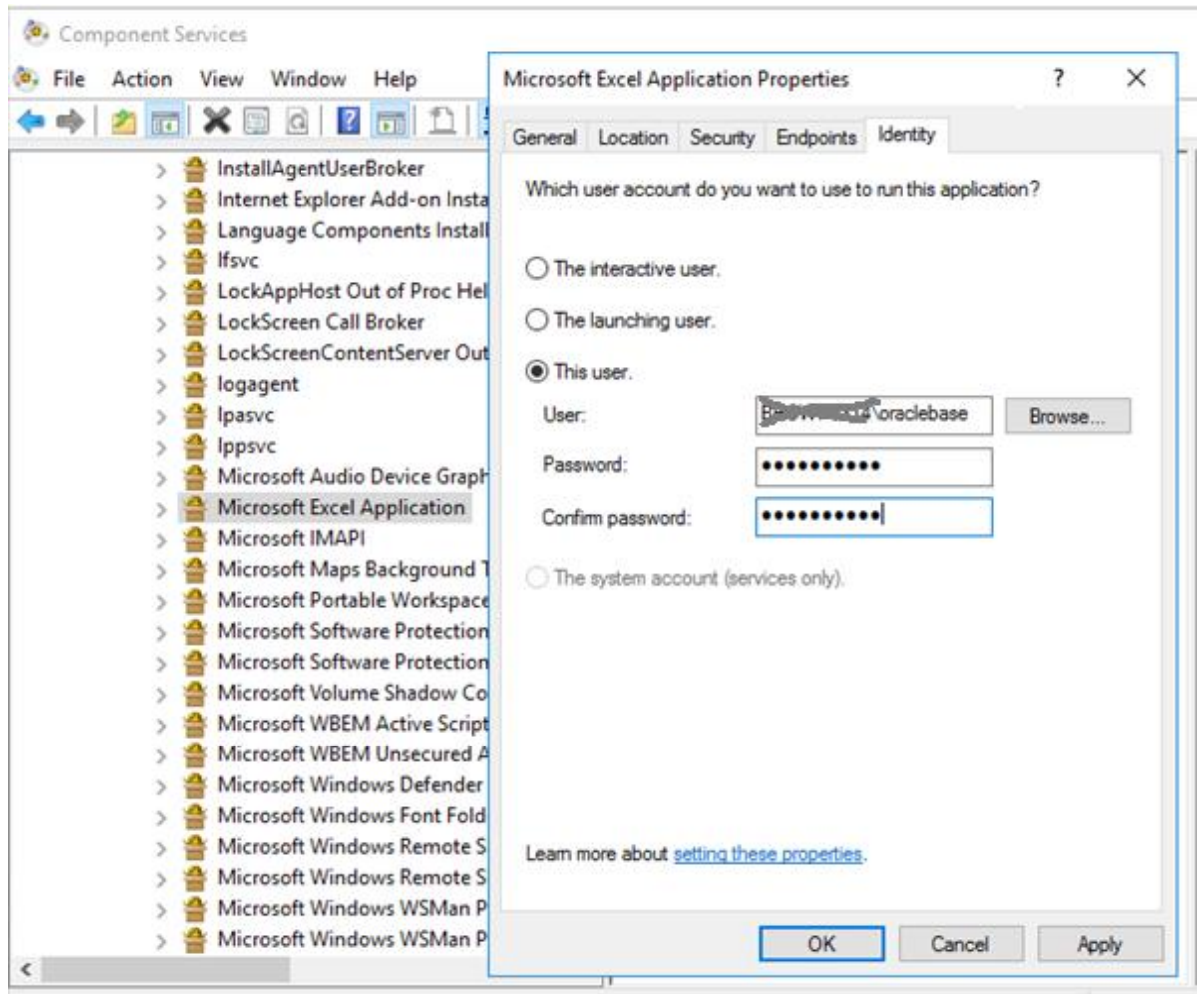
Expand:

Component Services → Computers → My Computer → DCOM Config



Select **Microsoft Excel Application** → Right Click → Properties

- Select **Identity Tab** and tick the **This user** bullet choice.
Provide **oraclebase** as the user (or the local user used to install Oracle) and set the password



This should solve the first problem, that MS Excel cannot be found!

Issue 2 : *Object variable or With block variable not set | Input File:*

This error, as mentioned earlier occurs at step 2 and 3. Our reasoning for this not working, is that Excel could not **Save As** the Excel file as it was opening in PROTECTED VIEW!

You can easily click on 'enable' and edit the document, BUT only when you are interactively logged on the host, and you can observe and interact with the dialogs Excel is showing you. BUT unfortunately this is not the case when calls to Excel are made by the third party app (Oracle Database), without the presence of a logged on user (please read the MS link on the top of this write up).

The easy solution to this is to add the folders you are working in as **Trusted Locations** in Excel. Setting the folders as trusted locations, the Excel file can be opened without restrictions, and excel is able to perform the necessary operations on it!

You can either setup the Trusted locations through Group policy or, in our case manually for the individual user.

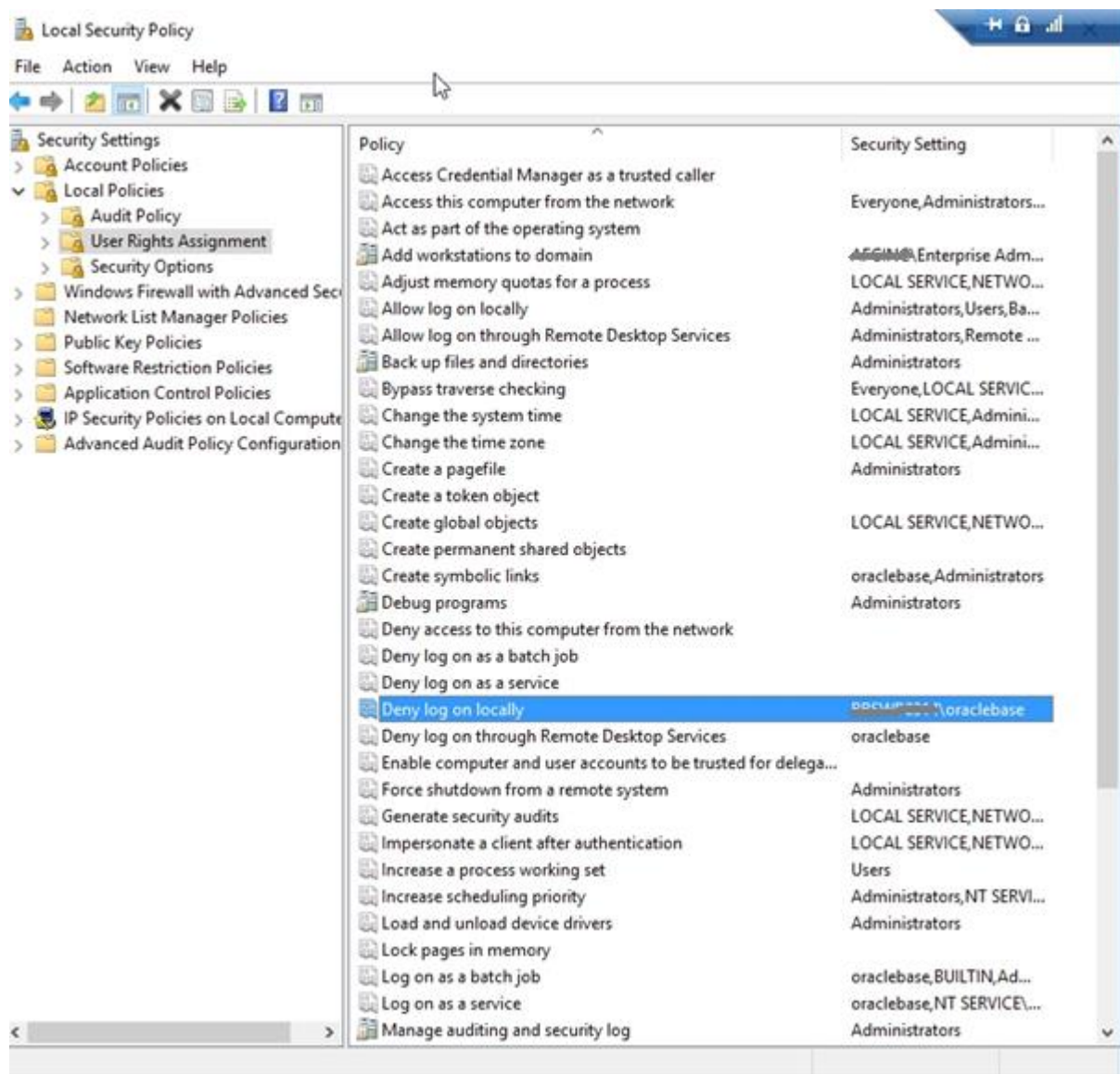
So In our case, we chose to add the Excel trusted locations for **oraclebase** user.

As **oraclebase** is a user created to install oracle, the oracle installation creates the user and denies log on rights (either locally or remotely). So in order to log on the host as oraclebase, start excel and add the trusted locations we needed, we had to:

Open the **Local Security Policy** of the host

Expand **Local policies** and click on **User Rights Assignment**

Remove the user from the **Deny log on locally** policy



Log on locally on the host as **oraclebase**, make the necessary changes to trusted locations, log off and change add the user back to the **deny log on locally** (to keep in line with the Oracle installs, your user log on policies at work etc!)

And that solves the second issue, and everything works as it should!