

Re: RDO objects and C#

Source:

<http://www.tech-archive.net/Archive/Outlook/microsoft.public.outlook.interop/2007-10/msg00035.html>

- *From:* "Dmitry Streblechenko" <dmitry@xxxxxxxxxxxx>
 - *Date:* Mon, 29 Oct 2007 14:27:25 -0700
-

Marshal.ReleaseCOMObject releases a *particular* COM object. If you use multiple dot notation, the compiler creates an implicit variable to hold the intermediate result(s). You cannot explicitly access and release these implicit variables.

Send an e-mail to dmitry@xxxxxxxxxxxx, including the ShareIt reference number of at least the name of the company would be helpful.

Dmitry Streblechenko (MVP)

<http://www.dimastr.com/>

OutlookSpy – Outlook, CDO
and MAPI Developer Tool

"AK" <AK@xxxxxxxxxxxxxxxxxxxxxxxxxxxx> wrote in message
<news:C3455E06-AFCE-48B6-B165-D1FA7921692F@xxxxxxxxxxxxxxxxxxxx>

Hi Dmitry,

Is it a limitation from .NET side that it's not able to release the COM objects? It seems bit weird that it's not able to release the COM object even after calling Marshal.ReleaseCOMObject as well as using GC.Collect.

Anyhow, it surely will help if you can send me the beta version to try it out. Do let me know what exactly I need to do for the same? Which account I should send an email to?

We have recently purchased the 4.4 version of Redemption library. Do you want me to give any reference number for the same?

Many thanks for your help.

Regards,
Amit Kathane

"Dmitry Streblechenko" wrote:

Re: RDO objects and C#

The line
rdoFolder.Items[i];
evaluates to
rdoFolder.Items.Item(i)
So you are still using multiple dot notation.

Do you want a beta version to see if it fixes this problem for you?

Dmitry Streblechenko (MVP)
<http://www.dimastr.com/>
OutlookSpy – Outlook, CDO
and MAPI Developer Tool

"AK" <AK@xxxxxxxxxxxxxxxxxxxxxxxxxxxx> wrote in message
<news:CE9E3D24-C503-4791-A8B9-FFD508624A1B@xxxxxxxxxxxxxxxxxxxx>

Hello Dmitry,

I'm using GC.Collect and modified my code to remove
multiple dot
notations
(e.g. rdoFolder.Items.count) but still I'm getting an error for
"MAPI_E_CALL_FAILED".

This is happening especially for RDOMail object.

My code for it reads as

```
=====
Queue DeleteEMailQueue = new Queue();
RDOItems rdoItems = rdoFolder.Items;
for (int i = 0; i < rdoItems.Count; i++)
{
RDOMail rdoMail = rdoFolder.Items[i];
try
{
// check if email can be deleted
if (DeleteEmail(rdoMail.UnRead, rdoMail.ReceivedTime,
rdoMail.Mileage))
{
rdoMail.Delete(0);
}
}
catch (Exception ex)
{
Util.LogException(ex);
}
finally
{
System.Runtime.InteropServices.Marshal.ReleaseComObject(rdoMail);
}
```

Re: RDO objects and C#

```
rdoMail = null;  
GC.Collect();  
}
```

=====

The error occurs after reading say around 700/800 emails in one folder.

Can you please suggest any workaround for this?

Thanks and Regards,
Amit Kathane

"Dmitry Streblechenko" wrote:

You are most likely running out of the RPC channel limit (<http://support.microsoft.com/kb/830829>) – since .Net does not release COM objects immediately, you can easily go over the limit even if you explicitly release COM objects using Marshal.ReleaseCOMObject. Avoid using multiple dot notation (e.g. rdoFolder.Items.Count) to make sure the compiler does not create implicit variables that you cannot reference and release. GC.Collect() might help. Next version of Redemption (send me an e-mail if you want a beta version) will keep track of the open MAPI objects and transparently release oldest unmodified MAPI objects that can be easily reopened on demand (Redemption objects will stay alive of course) when the number of open objects of a particular kind approaches the limit.

Dmitry Streblechenko (MVP)
<http://www.dimastr.com/>
OutlookSpy – Outlook, CDO

Re: RDO objects and C#

and MAPI Developer Tool

"AK"

<AK@xxxxxxxxxxxxxxxxxxxxxxxxxxxx>

wrote in message

news:0D7CB539-49DA-4449-A26B-09CB2DB2CD8F@xxxxxxxxxxxxxxxxxxxx

Hello Dmitry,

Thank you very much for
your assistance.

I'm getting following error
randomly.

"Error in
IMAPIFolder.GetContentsTable:
MAPI_E_CALL_FAILED"

Please find below my code

```
=====
private static void
ProcessUserMailbox(string
userName)
{
RDOSession rdoSession =
new RDOSessionClass();
try
{
// get the admin user name
from the configuration
file
string systemUserName =
Properties.Settings.Default.userName;
string exchangeSrvrName =
Properties.Settings.Default.UserMailboxServer;

if
(string.IsNullOrEmpty(systemUserName))
systemUserName =
WindowsIdentity.GetCurrent().Name.ToString();

rdoSession.LogonExchangeMailbox(systemUserName,
exchangeSrvrName);
IRDOFolder rootFolder =
rdoSession.GetSharedMailbox(userName).RootFolder;

// iterate through all folders
and subfolders
RDOFolders rdoFolders =
```

Re: RDO objects and C#

```
rootFolder.Folders;
if (rdoFolders != null)
{
    RDOFolder rdoFolder =
    rdoFolders.GetFirst();
    if (rdoFolder != null)
    {
        // iterate folder list
        for (int i = 0; i <
        rdoFolders.Count; i++)
        {
            // process the individual
            folder to
            search
            for
            emails that can be deleted
            ProcessFolder(userName,
            rootFolder.Name +
            "
            >
            ",
            rdoFolder);
            rdoFolder =
            rdoFolders.GetNext();
        }
    }
}
// release the COM object
for the root folder.

System.Runtime.InteropServices.Marshal.ReleaseComObject(rootFolder);
rootFolder = null;
}
catch (Exception ex)
{
    StringBuilder strInfo = new
    StringBuilder(string.Format("{0},
    Failed to process mailbox
    for user
    :
    {0}",
    userName));
    Util.LogMessage(strInfo.ToString());
    Util.LogException(ex);
    strInfo.AppendFormat(ex.Message
    + Environment.NewLine
    +
    Util.GetInnerException(ex));
    throw new
    ApplicationException(strInfo.ToString());
}
```

Re: RDO objects and C#

```
finally
{
rdoSession.Logoff();

System.Runtime.InteropServices.Marshal.ReleaseComObject(rdoSession);
rdoSession = null;
GC.Collect();
}
}

private static void
ProcessFolder(string
userName, string
folderPath, RDOFolder
rdoFolder)
{
RDOFolders subFolders;
RDOFolder subFolder;

try
{
subFolders =
rdoFolder.Folders;
if (subFolders != null)
{
subFolder =
subFolders.GetFirst();
if (subFolder != null)
{
// iterate sub folder list
for (int i = 0; i <
subFolders.Count;
i++)
{
// process the subfolder
ProcessFolder(userName,
rdoFolder.Name +
"
>
", subFolder);
subFolder =
subFolders.GetNext();
}
}
// if found an individual
folder (rdoFolder)
without
the
subfolder then release the
subfolder and subfolders
```

Re: RDO objects and C#

COM objects

```
//System.Runtime.InteropServices.Marshal.ReleaseComObject(subFolder);  
subFolder = null;
```

```
System.Runtime.InteropServices.Marshal.ReleaseComObject(subFolders);  
subFolders = null;
```

```
// if there are no sub folders  
then process the  
rdoFolder and iterate  
through all items  
// process the individual  
items/emails in the  
folder  
for (int i = 0; i <  
rdoFolder.Items.Count; i++)  
{  
RDOMail rdoMail =  
rdoFolder.Items[i];  
try  
{  
// check if email can be  
deleted  
if  
(DeleteEmail(rdoMail.UnRead,  
rdoMail.ReceivedTime,  
rdoMail.Mileage))  
{  
Util.LogMessage(string.Format("{0},  
Email  
with subject '{1}' Read :  
'{2}' Received Time : '{3}'  
Mileage :  
'{4}'  
is  
deleted.", userName,  
rdoMail.Subject,  
!(rdoMail.UnRead),  
rdoMail.ReceivedTime,  
rdoMail.Mileage));  
if (DeleteMail)  
rdoMail.Delete(0);  
//dfSoftDelete  
(0) –  
default. Deletes the item.  
Can still be recoverable if  
retention  
policy  
is  
set on Exchange Server.
```

Re: RDO objects and C#

```
//dfMoveToDeletedItems
(1) – the item is moved to
the Deleted Items folder
//dfHardDelete
(2) –
Exchange only. Permanently
deletes the item; will not be
recoverable
}
}
catch (Exception ex)
{
// in case of exception for
processing
individual mail log the
exception and process next
email
Util.LogException(ex);
}
finally
{

//System.Runtime.InteropServices.Marshal.ReleaseComObject(rdoMail);
rdoMail = null;
}
}
}
catch (Exception ex)
{
StringBuilder strInfo = new
StringBuilder(string.Format("{1},
Failed to process folder :
{0} for
user
:
{1}", rdoFolder.Name,
userName));
Util.LogMessage(strInfo.ToString());
Util.LogException(ex);
}
finally
{

System.Runtime.InteropServices.Marshal.ReleaseComObject(rdoFolder);
rdoFolder = null;
}
}
=====
```

The above error is coming

Re: RDO objects and C#

for some of the emails for
user. Can you
please
let me know what could be
the reason of this error.

Thanks and Regards,
Amit Kathane

"Dmitry Streblechenko"
wrote:

1. Exchange
transparently
manages the
locatiton of
the
mailboxes.
It
will
list all
mailboxes
in the
organization.
As an extra
check, you
might
want
to
read the
PR_EMS_AB_HOME_MDB
property
using
RDOAddressEntry.Fields[]
(the
property
returns the
name of the
server
where teh
user has a
mailbox) –
there can be
entries that
have no
mailboxes.
2. Yes, use
either (but
not both)
Logon or

Re: RDO objects and C#

LogonExchangeMailbox.
Or
set
the
MAPIOBJECT
property if
you already
have a
running
instance of
Outlook
(use
Namespace.MAPIOBJECT)
or CDO
1.21 session
(Session.MAPIOBJECT).

Dmitry
Streblechenko
(MVP)
<http://www.dimastr.com/>
OutlookSpy
– Outlook,
CDO
and MAPI
Developer
Tool

"AK"
<AK@xxxxxxxxxxxxxxxxxxxxxxxxxxxx>
wrote in
message
<news:319E9C32-4D06-4142-B3A7-15A4FB7F6581@xxxxxxxxxx>

Hello
Dmitry,