Sumatra
Development
www.sumatra.com

DIY: Migrating Microsoft Exchange 2013/16 To Office 365

A WORK IN PROGRESS (be aware of that)

Last Version: August 2017
Product Version: 0.8 Beta

Copyright

Table of Contents

Understanding Migrations

This guide explains how to take legacy data from Microsoft Exchange 2013/ 2010 / 2007 and insert it into Exchange 2013 or Office 365.

While our specialty at Sumatra Development is in server-side calendaring migrations (with contacts and tasks coming along for the ride), we have increasingly found folks coming to us asking about email migrations. It's just a simple nudge down the slippery slope of "gee…. How can we do ALL of this….?" And we've been nudged.

Much of this material we have either blogged about or referenced before, but as we see more and more sites going into Office 365, we decided putting together one guide in one place was a really good way to smooth the transition.

Overview – Train Hard, Fight Easy!

Regardless of the method, number of users, servers, or additional engineering requirements you have, we recommend three things.

1. ALWAYS run your conversion on a test system BEFORE moving it into your production environment.

2. ALWAYS run your conversion on a test system BEFORE moving it into your production environment.

3. ALWAYS run your conversion on a test system BEFORE moving it into your production environment.

We cannot state enough the importance of testing prior to deployment. The most successful migrations we have seen have been the ones with the most preliminary testing. Our motto comes from Marshall Zhukov via the Navy SEALS: *Train hard, fight easy*.

Flat calendar migrations are specifically designed to be done with less testing and quicker deployments.

For any full-state migrations we recommend you start the testing and the user mapping a minimum of six- to eight-weeks prior to your cutover date. We take migration results very seriously. The earlier you start, the better and easier the process is.

Quick Guide for the Impatient

1. We'll add to this list as necessary
2. Provision your users.
3. Make sure your permissions are set properly for a migration.
4. Run imapsync to get a pass on the email (you can sync this)
5. Run eCalReader to get in calendars, tasks, contacts (sync not supported on Flat version)
6. Switch your Outlook Profile over from your source system to your target system

You REALLY Want to Migrate via PSTs?

OK. But TRY it first and see if you replicate the results we did.

You REALLY Want to Migrate via Hybrid Directory?

OK. But TRY it first. Speed is glacial at best. You can accomplish data migration an order of magnitude faster with a separate domain to domain imapsync / Sumatra migration.

Post-Migration Resource Blues

Moving forward, you're probably going to have issues with Double-Booked Conference Rooms. This is normal functioning in Exchange / office 365.

We have a solution.

Contact us, mentioning this guide and we'll probably let you try it out for a while.

Email Migration into Exchange Office 365

imapsync

We have found imapsync to be an excellent product for email migrations. It is inexpensive, efficient, and effective. We at Sumatra do not profit from your use of imapsync. The developer gets all of the funding you supply him. And for a minimum of 50 Euros and a maximum of 100, there is no reason a seasoned Exchange administrator could not migrate all of their email with imapsync.

Please see our blog postings MDaemon Mail to Exchange via imapsync and imapsync vs PST: Tonnage and Speed as well as any other recent email migration postings on our blog.

This application has significant advantages over other products:

1. It is really simple to install and use.
2. The "sync" in the title is serious. You can upload all the data from a user set during working hours and then cut over the incremental changes starting on a Friday after closing time.
3. You can also use imapsync on a Linux environment as well as a Windows environment.
4. It is all in your control as opposed to run through someone else's data center or through a major integrator looking to run up hours.
5. It is very reasonably priced with the most liberal license I have seen.

In a Microsoft environment imapsync runs exclusively in the Command Prompt.

First off, please make sure you have enabled IMAP on your Exchange 2013 server.

Migrating an individual user to Office 365 looks something like this (if you use individual passwords for users).

```
imapsync.exe --host1 147.1.41.1
--user1 zyg@sumatra.local --password1  "XXXX"
--host2 outlook.office365.com
--user2 zyg@sumatra.onmicrosoft.com --password2 "XXXXX"
```

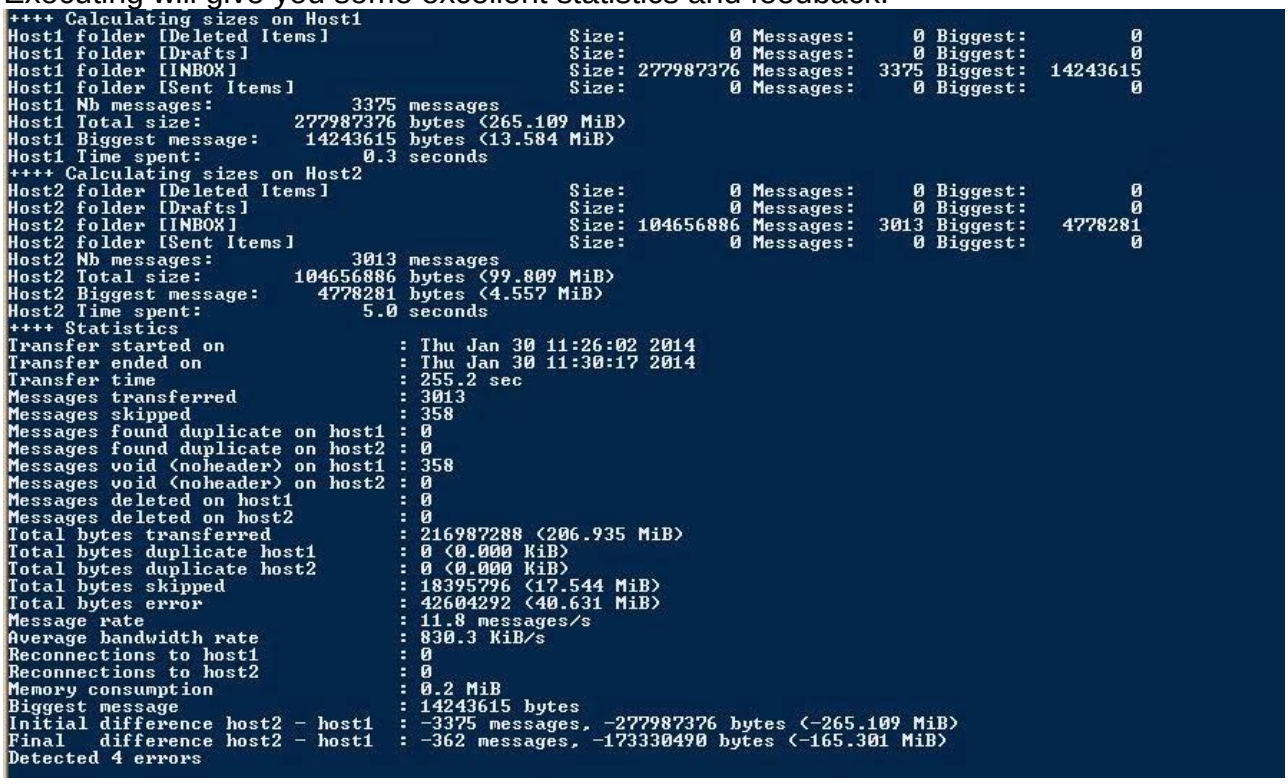
```
--ssl2
```

If you set up a service account with FullAccess, you can accomplish a migration with a command like this:

```
imapsync.exe
--host1 147.1.41.1
--user1 jimi.hendrix@sumatra.local --password1  "XXXX"
--host2 outlook.office365.com --port2 993 --sep2 /
--user2 jimi.hendrix@sumatra.onmicrosoft.com
--authuser2 riuliano@sumatra.onmicrosoft.com
--password2 "XXXXX"  --ssl2
```

Note in the above, for an Office 365 target system we need to use the "--sep2 /" command. This should not be necessary in newer versions of imapsync, but it does not hurt to have it.

Executing will give you some excellent statistics and feedback.



Iterating over a user list

In any event you are going to need to generate a user list to migrate email. Can you keep your migrating user list separate from your migration script? Answer: YES. This method assumes your legacy ID is the same as your target ID, but allowing for this to change is not a hard extension.

The imapsync ZIP file contains a script for iterating on a user list:
`sync_loop_windows.bat`
Which also contains an excellent primer on running imapsync in parallel.
This batch file assumes a text file in the form "User1;Password1;User2;Password2;..." (TIP: If you use a service account you will **not** need the password)

To get a user list from Exchange get-mailbox is the simplest method :

```
get-mailbox | fl name,emailaddress   > myusers.txt
```

to give you full name, email address.
For just email address run:

```
get-mailbox | emailaddress   > myusers.txt
```

You could also use ldapsearch if you already have scripts for that.

If you want to get a list of all users in an OU to migrate by segments:
`get-mailbox (e.g., -organizationalunit users`

Other things to be aware of

Exchange Configuration Requirements:
Before you can run imapsync, you will have configure Exchange for IMAP. This requires two steps.

Step 1: Start two IMAP4 services (and configure those services to automatically start if you wish.)

By default, in Exchange 2013 the IMAP4 service(s) are stopped:

To start those services: On the computer running the Client Access server role:

1. Set the IMAP4 service to automatically start:

```
Set-service msExchangeIMAP4 -startuptype automatic
```

2. Start the Microsoft Exchange IMAP4 service.

```
Start-service msExchangeIMAP4
```

On the computer running the Mailbox server role:

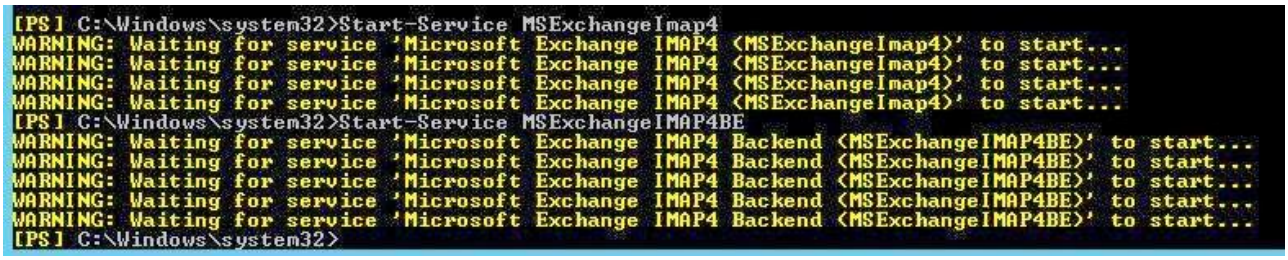
1. Set the Microsoft Exchange IMAP4 Backend service to start automatically.

```
Set-service msExchangeIMAP4BE -startuptype automatic
```

2. Start the Microsoft Exchange IMAP4 Backend service.

```
Start-service msExchangeIMAP4BE
```

In our case, the CAS and Mailbox server roles are on the same box:



Step 2: Configure Exchange IMAP4 External Connection so users can see (and thus use) the IMAP server settings

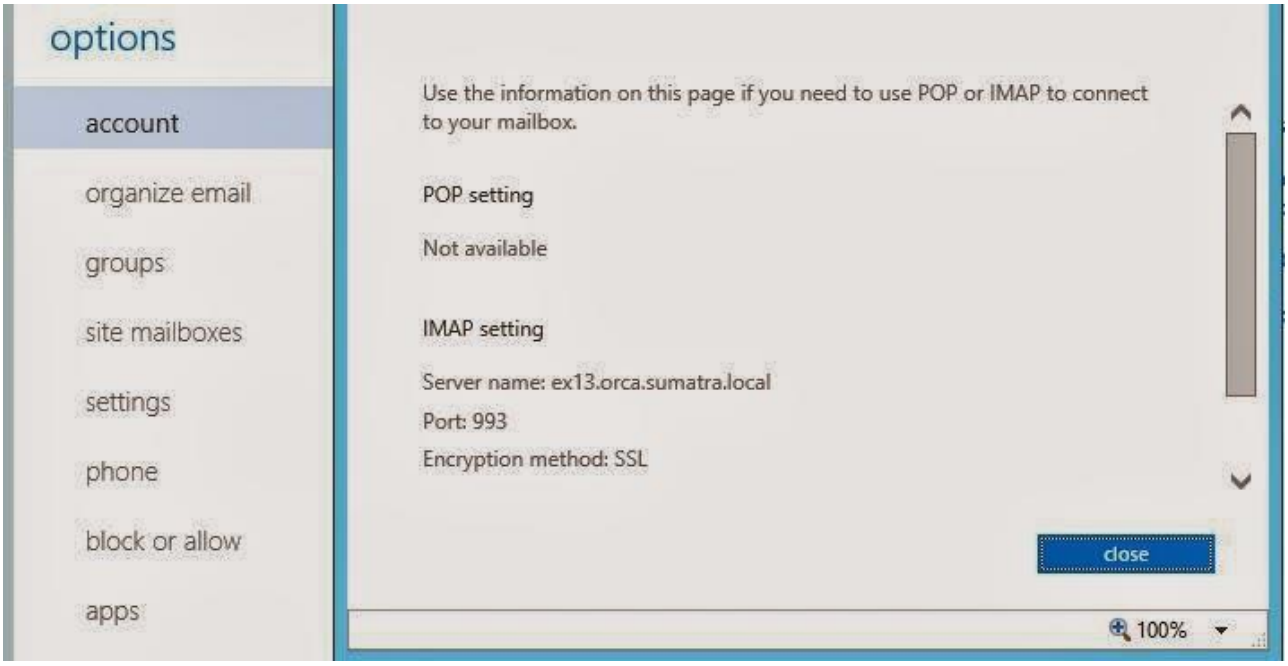
Use the powershell SET-IMAPSettings cmdlet, e.g.:

```
Set-ImapSettings -ExternalConnectionSetting {:993:SSL}.
```

This requires you restart IIS.

This is true even if you are working within the firewall Thus in our case, the External Connection is the same as the InternalConnection.

Finally, verify things are working, using OWA's Options select Account, then pick the "Settings for POP or IMAP access" link.



For more information see: http://technet.microsoft.com/en-us/library/bb124489(v=exchg.150).aspx

These sample scripts for major migrations / multiple users will help you out a lot.
http://imapsync.lamiral.info/examples/sync_loop_windows.bat
http://imapsync.lamiral.info/examples/file.txt

I really like the way this user lays out a basic sequencing for migrations including how and when to change your MX records.
http://hadjigeorgiou.com/blog/mdaemon-to-microsoft-exchange-migration/

And on the more than crucial need for advance planning and testing, please read this thread.

Common Problems and Solutions

Problem	Solution
Office 365 re-sends read receipts	Use imapsync ... --disarmreadreceipts
BAD Command Argument Error 11	Exchange IMAP server when it encounters any problem. Most of the time it is one of the following: Messages bigger than the Exchange default 10 MB limit. The Exchange limit can be set. See: Set-TransportConfig. If you can't configure this limit is not an option then use imapsync option: --maxsize 10000000 for 10 MB, change it if needed) for imapsync to skip those messages. On Office 365 the default is 25 MB so:

	imapsync ... --maxsize 10000000 # 10 MB for Exchange imapsync ... --maxsize 25000000 # 25 MB for Office365 Quota reached. The account is inbox is at maximum. Increase the quota: see How to set Exchange Online mailbox sizes and limits in the Office 365 environment Lines too long in messages. Use option –maxlinelength to skip messages whose max line length is over a specified number of bytes. maxlinelength 1000 is specified in RFC2822 but most servers support higher values. Exchange supports 9900: imapsync ... --maxlinelength 9900 **GENERAL SOLUTION TO MANY PROBLEMS**: imapsync ... ^ --maxsize 10000000 ^ --maxlinelength 9900 ^ --regexflag "s/\\Flagged//g" ^ --disarmreadreceipts
BYE Connection closed	Exchange closes the IMAP connection after 10 errors. Solution: wait a few minutes and re-run imapsync

This draws heavily from the imapsync FAQ.

Special case: Zimbra email to Exchange / Office 365

You can easily migrate email from Zimbra to Exchange or Office 365.

See our blog posting #Zimbra Email Migration to #Office365 using #imapsync for more details and use the automap option for folders that do not exactly match.

Use PSTs if you have a few dozen accounts to migrate. If you have more than that explore other options.

For calendar, of course, Sumatra can do full-state migrations as though you'd been running Exchange all along.

Calendars, Tasks, Contacts

Before we begin

The process of moving calendar data is an order of magnitude more complicated than moving email:

- Email is a static object requiring format changes and proper management to move, it is connected fundamentally to a single user. The same holds true for Contacts and (on Exchange) Tasks.
- Calendars and schedules are cross-connected to multiple users. Its value results from exactly those cross-connections. They are true webs of information rather than static threads.
- Full-State calendar migrations cannot be successfully accomplished overnight on a day's notice with no planning or testing. You have been warned.

Sumatra's technology can move these scheduling webs with precision, maintaining the state information on recurrence patterns, guest responses, meeting exceptions, etc. that ensure scheduling remain a crucial enterprise application.

Based upon user demand we have created a mid-way option which is faster than our full-state migration process. This guide goes over those options: Flat and Full.

This DIY guide assumes the FLAT option. But we're writing the guide so we're also going to tell you about our Full State option which we'll be happy to license you if appropriate (but that will blow past the $1000 limit we set for DIY. So just be aware of that).

Contact Sumatra via our web site at www.sumatra.com to get a license.

What is the Difference?

We offer a spectrum of options for an Exchange to Exchange migration.

	FREE	Flat	Full
If you want your Exchange to Exchange calendar migration…	Inserts calendar as-is	Generates Recurring meetings. Add guests to agendas	Full state for all accounts
Free, fast, no frills no live meetings, no recurrence patterns, no support	✓		
Inexpensive, fast recurrence patterns and guests in agendas but no live meetings		✓	
Reasonably priced, fast recurrence patterns and current meetings re-proposed (but not responded to automatically)			✓
Full support, full state-recreation for an enterprise			✓
Each of these typically best serves:	Small sites, the budget-constrained	Small-medium sites education, non-profits	Serious Time-Valuing Enterprises

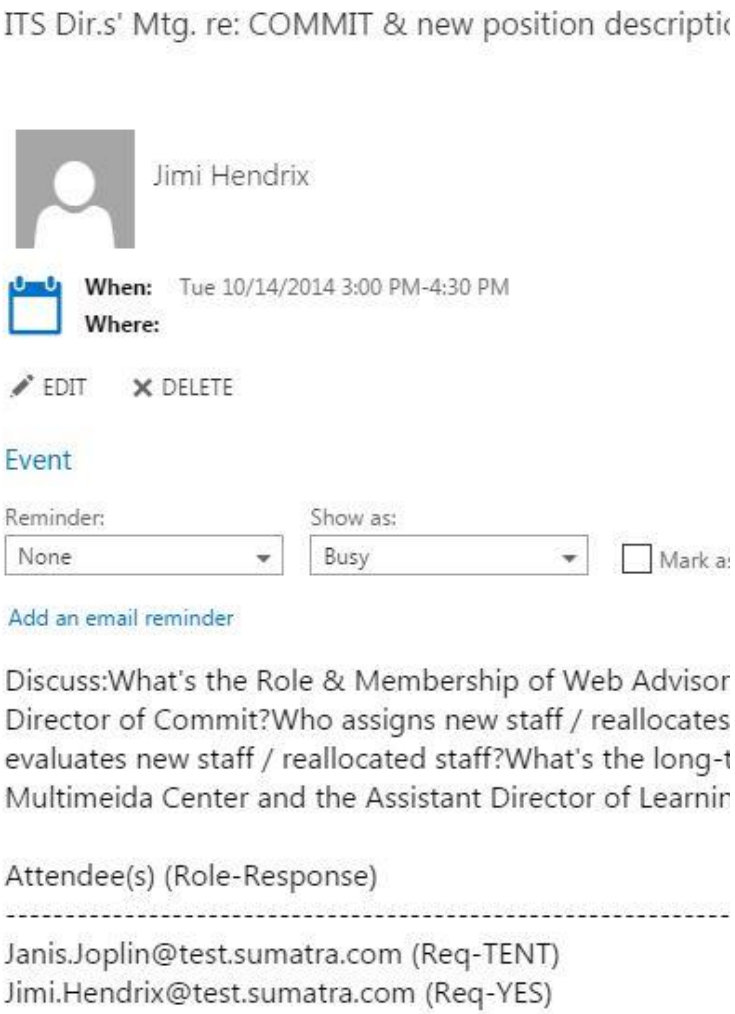
Free

We insert 30 days of data, from “today,” from your Source Exchange, including the legend “Inserted courtesy of Sumatra Development, the Exchange calendaring experts.” This is meant to prove you CAN migrate your data in your control.

Flat

The Flat version extends the Free option by:

- It inserts <u>all</u> current data
- It re-creates recurrence patterns
- It adds meeting attendees and their responses to the agendas.
- It has a per-seat licensing cost. Unless it’s part of the DIY program.
- Meetings proposed by others, including those outside of the organization, appear on the calendar.
- See example to the right.



Full State

If you are an enterprise-scale site placing a high value on your calendars, want your end users to see the calendars without having to recreate or respond to meeting request, Select the Full State option.

Set-up and Configuration

Requirements

Make sure that your environment meets the following software requirements.

MS Exchange 2013/2016 (READING from Exchange 2010 and 2007 will probably work, Sync will not)

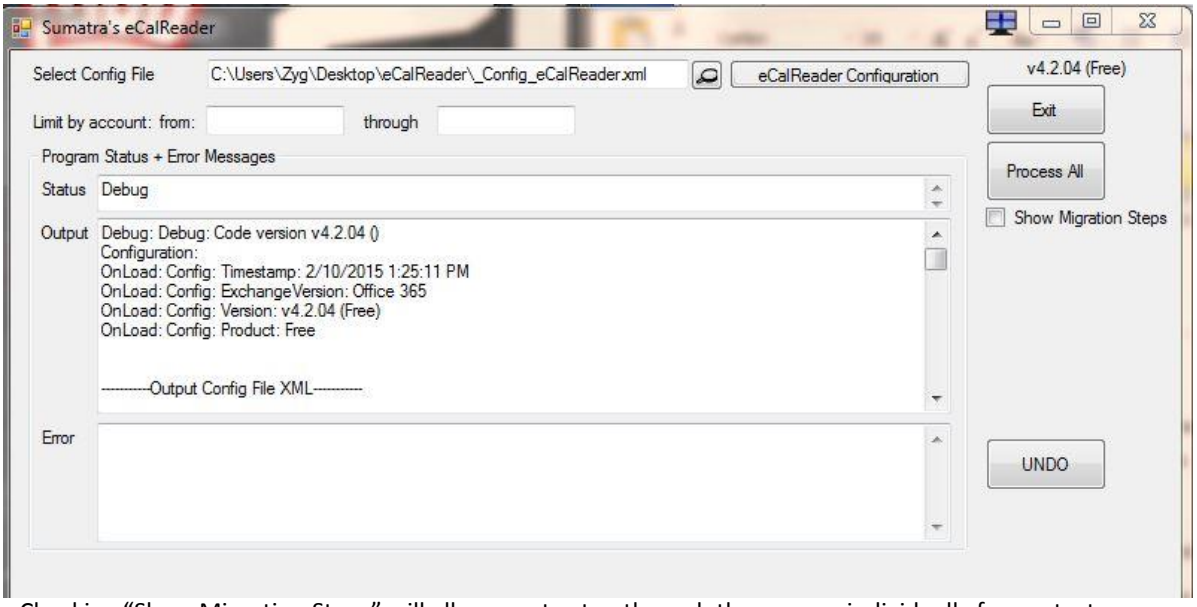
.NET Framework 4.5

The Microsoft Exchange Web Services Managed API V2.2

To insert data into Exchange or Office 365 you will need a service account with Impersonate role.

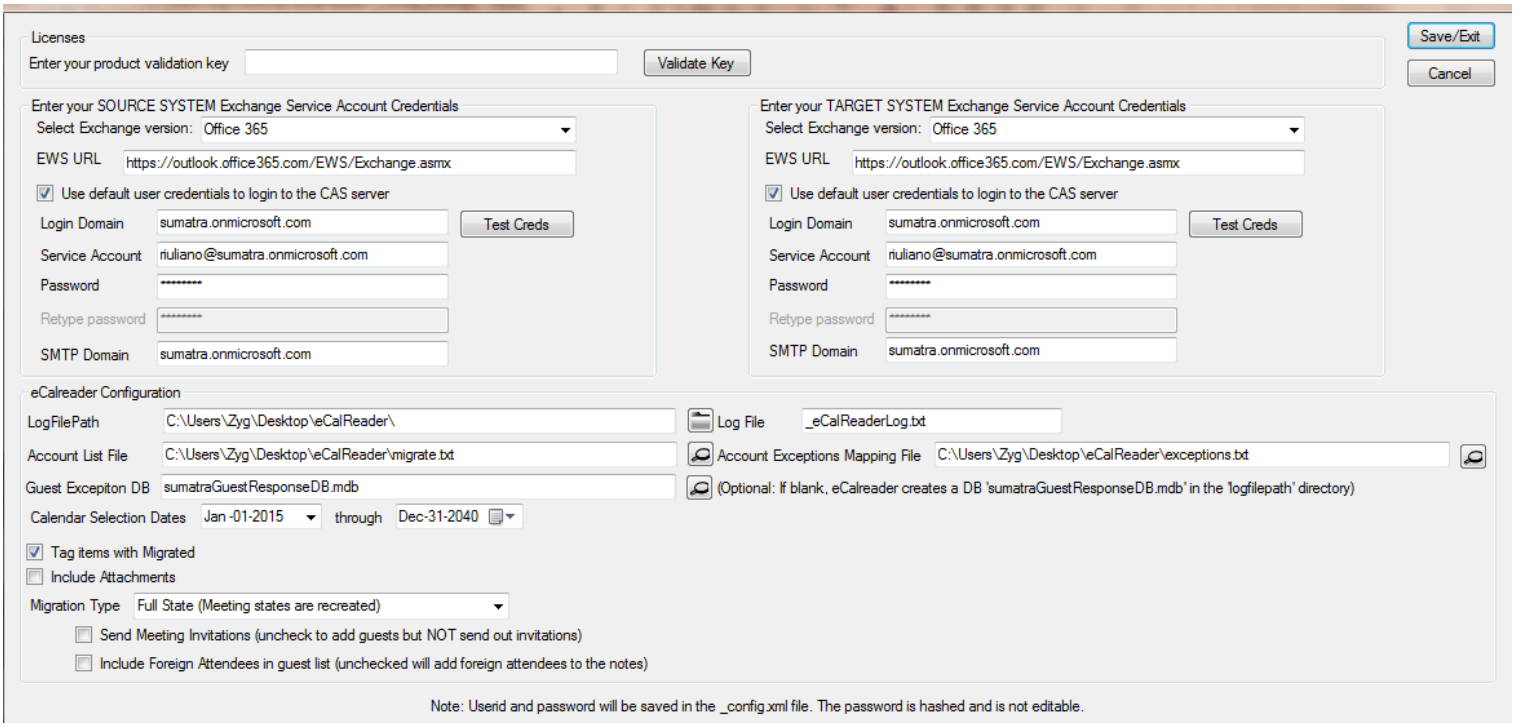
eCalReader

Meet eCalReader, your best friend for moving calendars from Microsoft Exchange 2013 into Exchange. This will read through all calendar, task, and contact data in your source Exchange and insert them into the corresponding users in your target Exchange domain.



Checking “Show Migration Steps” will allow you to step through the process individually for contacts, calendars, and Tasks.

Configuration



Most of these options are fairly straight-forward and are dealt with below in this document.

1. Your biggest choice is to do wither a Full migration (recreating meetings and optionally guest responses) or a Flat migration (making meetings into appointments in everyone's calendar). Flat is faster, Full recreates your calendars completely.
2. All email addresses will be reassigned to a new domain, i.e., changed from USER@source.Exchange.com to USER@Target.Exchange.com.
3. If your domain doesn't change, then keep the Exchange domain names the same.
    a. What happens when a meeting organizer or attendee's email address is not from your old Exchange domain? That's what we mean by a "Foreign user"
        i. If it's a meeting organizer, that item is added to the calendar as an appointment.
        ii. If it's an attendee, for the Full State, and the "Skip All Foreign Users" is checked, that address not be added to the attendee list, although it will be included in the agenda.) If unchecked, the account will be added to the attendee list. CAUTION: this will send out email to that user. If you run this multiple times, particularly in a test environment, you will flood that user with multiple requests from your test domain!
4. You will need CAS Login credentials (and this user must have Impersonate role to all users set via ManagementRoleAssignment)

5. We recommend you use “Test Creds” button to insert a “test” appointment into one calendar to confirm you have access to a user’s mailbox on Exchange.

6. The Product Validation key is sent to you by Sumatra. The Free version of the application does not require a key. After entering the key and before inserting data, press “Validate Key” to ensure the key is valid. Otherwise, eCalReader overwrites the configuration with “free” version settings.
7. Whether to take attachments? Attachments slow the process down. In general, we do not recommend taking them, but you do have that option.

Test Creds

The Test Creds button is your check that you have the proper credentials to insert data into your target Exchange system. If you have your permissions configured, press “Test Creds” and insert a user (other than your Service Account). If your permissions are configured correctly it will insert an appointment in that account’s calendar at the next hour boundary and inform you this has succeeded. If your permissions are not configured correctly it will inform you of failure and you need to evaluate what is wrong in your environment.

Accounts and Exceptions

If you want to have invitations properly sent and you are modifying any user IDs between your Source and your Target system, PAY ATTENTION TO MAPPING!!!! The account file and the Exceptions file are two things you really need to understand.

The accounts file is the list of accounts you want to take data from. If you want to take a limited number of users for testing purposes, do so by editing this file to contain only that list of users.

The exceptions file is the listing of account mappings that are not simple transpositions of domain names (i.e., Jhendrix@source.company.com maps to jimi.hendrix@target.company.com in the Exceptions file).

Jjoplin,Janis.jopln
Liberace, walter.liberace

Or

Jjoplin@source.yourdomain.com,Janis.jopln@ target.onmicrosoft.com
liberace@source.yourdomain.com,walter.liberace@target.onmicrosoft.com

If you include the domain name(s), eCalReader will remove them and replace them with the domain names defined in its configuration section.

IF they are not changing (i.e., jjoplin@ source.yourdomain.com is going to be jjoplin@target.onmicrosoft.com) you do not need them in the Exceptions file.

When the eCalReader application runs it is checking that accounts you want to take from your Source Exchange actually exist in Target Exchange. You will come to appreciate this foresight (especially if you want to re-propose meetings).

UNDO

One of the great benefits of our technology is the selective UNDO capability.

If something goes wrong with your migration (like inserting data into the wrong user, not that that has ever happened in the real world or anything….) you can remove the data Sumatra's application inserted, leaving all other data in place. We urge you to test this feature. It's a very powerful feature that will keep you from having to revert to backups in the event of catastrophe.

Determining Your EWS URL

In Office 365 your EWS URL is

https://outlook.office365.com/EWS/Exchange.asmx

You MAY be able to go to the Exchange Control Panel (ECP) and use these more specific URLs for speed, but Microsoft is rapidly removing this capability:

- Click on "Options" (upper right of the screen). This switches to ECP and the domain in the URL changes (in our case it's chNprdNNNN.outlook.com) - or -
- Sign in to Office 365. Click on Outlook. Look at the domain in the URL, in our case it is snNprdNNNNoutlook.com where N = a number.

For on-premises Exchange, the EWS URL formula is something like: HTTPS://CAS_server/EWS/Exchange.asmx

In ON-PREMISES you will usually have your IIS set for Windows Authentication (see http://technet.microsoft.com/en-us/library/gg247612.aspx for more details). This is also the default in hosted Exchange. Should you need to change this you may do so in the eCalReader's configuration file (_Config_XML) file by changing the **HTTPAuthType** parameter (options are Basic, Negotiate, ntlm, and Kerberos)

NB: You hear us talking about Exchange being a moving target in a migration. That's true here. The default is **Negotiate** in Exchange 2013, and **Basic** in Exchange 2007 and 2010. And any rollup, service pack, or bug fix could change the way Exchange permissions are managed or default. Use the "Test Creds" button in setup to ensure your permissions are set correctly.

Setting 0365 Permissions (Quick Guide)

GLOBAL ADMINISTRATOR rights give you administration rights over Exchange / Active Directory, but they do not give you the rights to access mailboxes – which is what you will need to move in data and re-create state.

We're going to take setting permissions in stages. We'll do this assuming your domain is hosted in Office 365. The process is similar for Exchange 2013 On-premises.

1.) Your ADMINISTRATOR account needs to be able to:

   a. Use REMOTE POWERSHELL to Log into Office 365

   b. Create a separate service account (this keeps your ADMIN function separate from your MIGRATION function)

i. We call the Service Account EXSU. When you create it, make sure it is mailbox-enabled (you will be sending email on behalf of this account)

ii. In Office365 you want to make sure that your administrative account is assigned to the built-in Role Group "Organization Management." On Role Groups see: http://technet.microsoft.com/en-us/library/dd638105.aspx#Builtin

iii. Grant EXSU three rights:

1. Impersonation

2. No throttling. This is relevant (i.e., in your control) only for on-premises Exchange. For Office 365 you will need to contact your Microsoft rep and explain what you are doing and ask throttling turned off for the duration of your migration.

3. If you grant the service account FULL ACCESS to mailboxes, it will be easier for you to use OWA to check the results for individual users in testing and migration.

2.) To do this – use the various Exchange PowerShell cmdlets which execute the appropriate actions.

a. Start POWERSHELL.

b. REMOTE to your OFFICE365 account

c. IMPERSONATION: You're creating a ROLE called "_suImp8" that allows Impersonation and then assigning it to EXSU

```
new-ManagementRoleAssignment    -Name:_suImp8    -Role:ApplicationImpersonation

 -User:exsu
```

d. THROTTLING: You're creating a policy called SuThrottling Policy and then assigning it to EXSU. (Otherwise Office 365 might shut you off mid-migration)

```
New    -ThrottlingPolicy SuThrottlingPolicy

       -EWSMaxConcurrency $null

       -EWSMaxSubscriptions $null

Set-ThrottlingPolicyAssociation
```

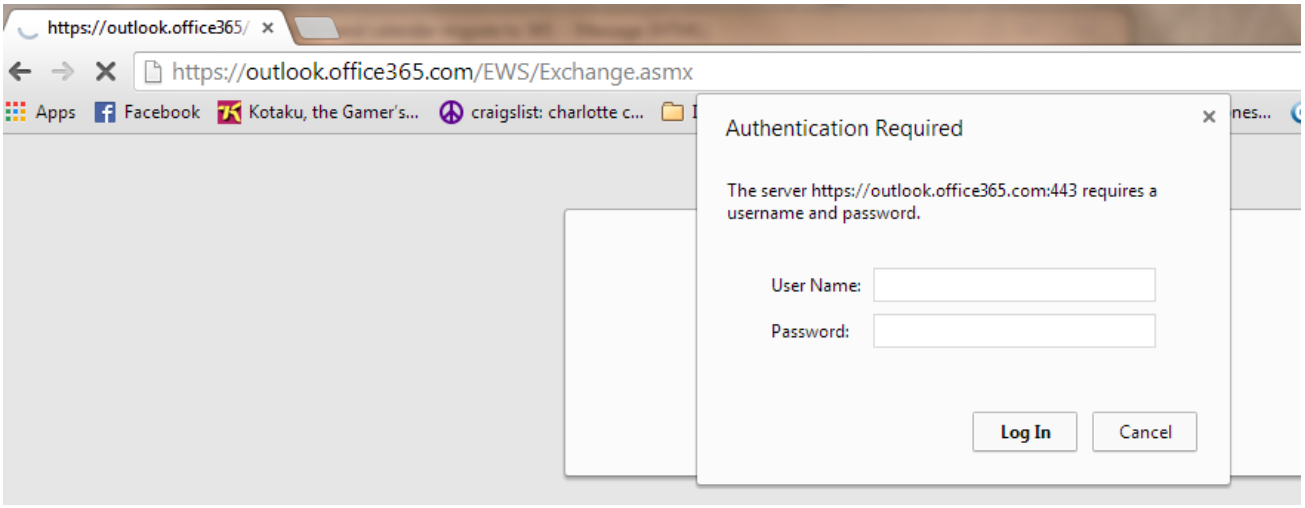
```
-Identity exsu

-ThrottlingPolicy SuThrottlingPolicy
```

e. FULL ACCESS: this grants access to ALL MAILBOXES in your domain to EXSU.

```
Get-Mailbox -resultsize unlimited | add-mailboxpermission
            -user exsu -accessrights: fullaccess
            -InheritanceType: All
```

3. TEST

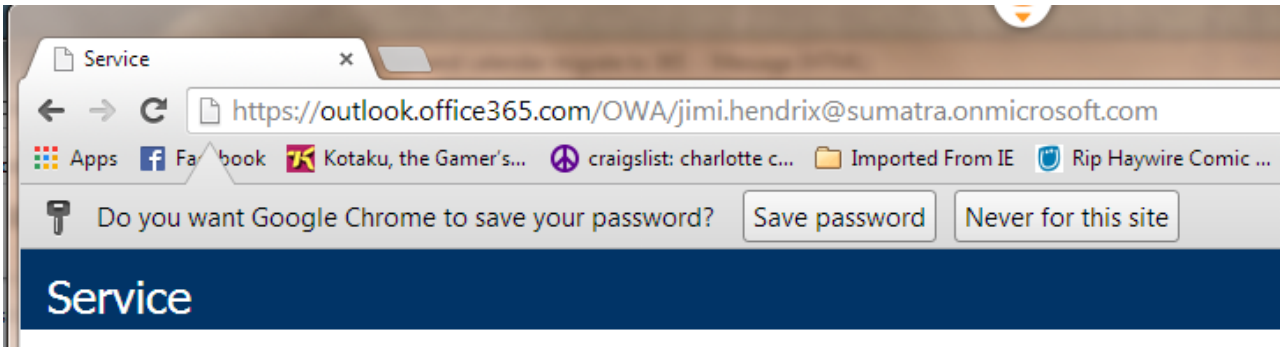
Can you put the EWS URL in a BROWSER and when prompted for credentials get this?



LOG IN with your EXSU credentials, and see the Exchange Web Service page:

This example shows access to Office 365. Obviously if you are going into your on-premises or your own hosted domain, your URL and service name will be different.

Now to test FULLACCESS go to the URL box and modify it as I have with a user on your domain:



Hit ENTER

Now you will be prompted for your end user mailbox credentials. Use the service account (EXSU) and the password to access to the mailbox. This is where FullAccess comes in – you don't have to crack all of your end users' passwords!

NOW it should display your test users' mailboxes in OWA

If all of these are successful, now you can do a test insertion.

Exchange Categories: Do NOT Clear Them

The Sumatra insertion process uses categories to cleanly remove our inserted data for testing and in the event of disaster.

During a migration make sure your Exchange server is NOT clearing categories from email. Post migration you can change it if you wish.

This cmdlet takes care of the issue.

```
Set-TransportConfig -ClearCategories: $false
```

Managing Delegate Forwarding Rules

This is relevant ONLY for FULL-STATE Migrations. If you're running a Flat migration you may ignore this.

Existing Exchange rules that act on meeting invitations are the enemy of clean calendar migrations. This means in a production environment you want to cleanly manage Delegate Forwarding Rules.

Leaving them on in Exchange 2013 is rarely fatal – but it is confusing and annoying for end users.
For versions of Exchange prior to 2010 SP1 the Administrator must either remove these by a script or run a clean installation.

How to Disable Meeting Forward Notifications

Disable External Meeting Forward Notifications for a Domain

```
Get-RemoteDomain | Set-RemoteDomain -MeetingForwardNotificationEnabled $false
```

Disable Internal Meeting Forward Notifications for a Single User

```
Set-CalendarProcessing -Identity user email@yourdomain.com
-RemoveForwardMeetingNotifications $true
```

Note on Permissions: Impersonate vs. Delegate

When do you use which permissions?

- Impersonate is typically used for ENABLED user accounts. Note, Impersonate fails when it tries to access a disabled account
- Delegate is used when dealing with DISABLED accounts, such as ROOMS disabled end user mailbox accounts, or in environments with a Resource Forest Trust. Note: the actual mailbox permission is "FullAccess" (Full access is set via add-mailboxpermission command shell)
- We refer to Room and Equipment accounts as "Resource" accounts (because it is more general). Room / resource accounts are provisioned as DISABLED accounts (by default).

An excellent Microsoft summary of the differences in permission is here:

Exchange Impersonation vs. Delegate Access:
http://blogs.msdn.com/exchangedev/archive/2009/06/15/exchange-impersonation-vs-delegate-access.aspx

Exchange Web Services Throttling

It seems that every SP and Roll-up of Exchange makes throttling more and more.

Now EWS is included in Exchange throttling. You can read about it at *More throttling changes for Exchange Online*.

Often after you apply a patch or roll-up you will find your throttling defaults re-set or that the behavior has changed (yet another reason we are maniacs about constant testing). You might need to delete and recreate your applicable policies for this process.

Into an on-premises installation turn this off during migration.

Our recommendations going forward for Hosted Migrations:
During validation, if you can, point to different CAS servers to reduce CAS-server throttling.

During an insertion, use MULTIPLE service accounts which means using parallel insertion processes and point these to different CAS servers. We're set up for this already, but we now recommend it in smaller migrations than we used to.

During migration, set the batch input to at least 50 calendar objects.

Exchange Accounts

There are four kinds of accounts in Exchange:

- Users
- Resources
- Contacts
- Shared[1]

Your migration will definitely make use of the first two (and on occasion the third)

Within Resources there are two types:

- Rooms
- Equipment

Within rooms, there are two basic types

- AutoAccept (think of this as “First-Come-First-Served”)
- Managed (think of this as “Janet approves booking this room”)

User accounts are fairly obvious and straight-forward. Every user you migrate needs to have an account, and this account needs to be enabled on Exchange.

[1] See our blog posting at http://calendarservermigration.blogspot.com/2008/08/shared-calendars-in-exchange-2007-sp1.html for more information on what you can or should do with making legacy group calendars into Shared calendars post-migration.

Contacts (or mail-enabled contacts) are important if you are planning on migrating in stages, or domain by domain. We'll deal with this case later since it is not common, but it is useful in very large migrations.

There is NO capability in bCalReader to change account types during a migration. If you have a Resource account in Beehive, it will migrate into a Resource in Exchange, not a User.

Shared accounts: Migrate shared accounts as user mailboxes. Change them to shared post-migration using this cmdlet for the shared calendar IT_Vacation:

```
Set-Mailbox -Identity IT_Vacation -Type Shared
```

Resources: Before and After

For a Free or Flat migration this section is mostly provided for your information.

For a Partial migration, set your resources as you wish. Since invitations will be sent to them from the meeting organizer you may have them set to automatically book or be managed by specific individuals, the choice is yours but so is the necessity and responsibility to communicate this to your organization.

For a Full State migration continue to read the following.

Resources in a migration require special handling. To re-create state from a previous calendar system we need to be able to take direct control during the migration – but post-migration you obviously want to start using the capability Exchange is built for.

Resource Accounts in Exchange 2010/2013 are **DISABLED** upon account creation.

For the migration process the Sumatra process for Exchange **requires that Resource accounts be temporarily ENABLED with AutomateProcessing set to NONE and that resource accounts have a password** (or you cannot ENABLE the accounts which is necessary for Sumatra insertion). This is because without the Resource accounts ENABLED we cannot re-create the state that existed in Meeting Maker / OCS and we must do this based on source system data, not on the AutoAccept rules Exchange employs.

Automatic Booking:

To use the Exchange Management Shell to **Disable** automatic booking on a resource mailbox:

```
# In Exchange 2007
Set-MailboxCalendarSettings <Identity> -AutomateProcessing: None
```

```
# In Exchange 2010, Exchange 2013, Office 365
Set-CalendarProcessing <Identity> –AutomateProcessing: None
```

To **Enable** (post migration):

```
# Exchange 2007
Set-MailboxCalendarSettings <Identity> -AutomateProcessing: AutoAccept
# Office365 et al
Set-CalendarProcessing <Identity> –AutomateProcessing: AutoAccept
```

The Microsoft documentation can be helpful :

http://technet.microsoft.com/en-us/library/dd335046(v=exchg.150).aspx

An excellent summary of creating resource mailboxes can be found here:

http://help.outlook.com/en-us/140/dd569933.aspx

Your actions:

Put all resources in one (or more) Organization Units (OUs) for ease of administration

Just prior to the migration:

1. ENABLE all of the resource accounts via Active Directory Users and Computers
2. Hide the accounts from the GAL
3. Configure resources not to AutomateProcessing: AutoAccept meetings

After the migration: disable the accounts, add them to the GAL and configure to:

- AutomateProcessing: AutoAccept (this will result in a "first-come-first-served" room) or
- Use group-policy settings for managed rooms

In Exchange Management Shell the commands for setting will look like this

```
Get-Mailbox -resultsize unlimited  -filter {isResource
        -eq $true} | Set-MailboxCalendarSettings[2]
        -AutomateProcessing: None -deletesubject:$False
        -AllowConflicts: $true -EnforceSchedulingHorizon: $False
```

Note that after executing this re-start the Exchange Information Store Service (otherwise there is a default of 2 hours on the refresh for these properties).

We have found this table of settings to work well:

Setting	"Default" Value	Pre-Insertion Value	Post-Insertion Value
AutomateProcessing	AutoUpdate	NONE	AutoAccept
AllowConflicts	False	TRUE	FALSE
BookingWindowInDays	180		YOU DECIDE
EnforceSchedulingHorizon	True	FALSE	TRUE
ForwardRequestsToDelegates	True	FALSE	FALSE
DeleteSubject	True		YOU DECIDE
AddNewRequestsTentatively	True		YOU DECIDE
RemoveForwardedMeetingNotifications	False	TRUE	TRUE
MailboxOwnerId	xxx		YOU DECIDE

If you decide to set the booking windows in days (to, say, 180 days), remember that "ongoing" meetings will extend beyond the 180 days. Caveat: Many migrated meetings are ongoing or have an end date outside of your booking window. Once your end users change those meetings, the booked resource will decline those previously booked meetings because they fall outside of the booking window.

[2] In Office 365 this is now Set-CalendarProcessing –AutomateProcessing

Post-Migration Conference Room Blues

Moving forward, you're probably going to have issues with Double-Booked Conference Rooms.

We have a solution.

Contact us, mentioning this guide and we'll probably let you try it out for a while.

What does NOT migrate into Exchange

The following table outlines explicitly what is not migrated from Microsoft Exchange 2013 to Exchange

Data we do not migrate at all:

- Colors on meetings and activities. Categories DO come over. (Colors however are a user default in Outlook)
- Categories do not come over on Attendee Responses to Meetings if the Attendee has set their own. Brief reason: in re-creating states of meetings we use the information the Organizer has as definitive. User categories on meetings do not feed back to the Organizer so we do not take them forward.
- If an Attendee accepted or declined a meeting and did not send a response to the organizer, we set the status as the Organizer has it.
- Notes (Exchange Web Services does not give us access to read them or to create them). If you want to take them over tell your users to copy them into Tasks which DO migrate.
- Task assignments – EWS does not allow us to propose tasks as we can with meetings.
- Reminders greater than 24 hours. Also Office 365 supports email reminders, which Outlook does not. In going from an Office 365 environment to an on-premises environment email reminders are lost.
- User defaults (including work hours) – though this you can handle yourself. See: Switch your Outlook Mail Profile below.

Other aspects of the complete migration process

User Provisioning

This can be accomplished via the Office 365 portal, via PowerShell cmdlets, and via a CSV.

For Office 365 check out this posting Plan for creating user accounts in Office 365.

If you need more detailed directions, let us know.

Delegate Rights

We think these come along with the Outlook Profile. But if they don't it's scriptable in a variety of ways.

Switch your Outlook Mail Profile

As you've been using Outlook (and if you're in Exchange this is a pretty safe bet) you can keep all your user preferences by switching your Outlook mail profile from your legacy system to Office 365.

Deploy Outlook mail profile settings via GPO or script does a stellar job of going through the options using (d'uh!) Group Policy Logon scripts. We made some modifications to the scripts in that link.

Deployprf.vbs should look like this:

```
Option Explicit

'This script determines if a specified mail profile already exists.
'If it doesn't, it will set the path to the prf-file containing
'the mail profile configuration settings.

'Script created by: Robert Sparnaaij
'Updated by R Iuliano, Suamtra Development 2/20/2014
'For more information about this file see;
'http://www.howto-outlook.com/howto/deployprf.htm

Dim ProfileName, ProfilePath, ProfileVersion, strProfileVersionName, strKeyProfileVersionPath
Dim strComputer, strKeyProfilePath, strLastChangeVer
Dim strValueLastChangeVer, strValueProfileVersion
Dim strOutlookPath, strOutlookPathValue, strKeyOutlookAppPath, strKeyOutlookSetupPath
Dim strOutlookVersion, strOutlookVersionNumber
Dim strImportPRFValueName, strImportPRFValue
```

```
Dim strFirstRunValueName, strFirstRun2ValueName
Dim ReapplyPrf
Dim objRegistry, objFSO
dim DEBUGME

'=====BEGIN EDITING=====

'Name of mail profile as in the prf-file
ProfileName = "%UserName%"

'Path to the prf-file
ProfilePath = "\\ntq5\base\root\apps\setup\exchange\YOURCHOICE_Windows7.prf"

'Increase the ProfileVersion whenever you want to reapply a NEW prf-file
ProfileVersion = 1

'Force the re-application of the Profile
ReapplyPRF = False

'Added 2/20/2014 RVI Sumatra Development
'Set DEBUGME to TRUE if you want to see message windows
DEBUGME = FALSE

'======STOP EDITING UNLESS YOU KNOW WHAT YOU ARE DOING=====

const HKEY_CURRENT_USER = &H80000001
const HKEY_LOCAL_MACHINE = &H80000002

strComputer = "."
Set objRegistry = GetObject("winmgmts:\\" & _
   strComputer & "\root\default:StdRegProv")

strKeyProfilePath = _
          "Software\Microsoft\Windows NT\CurrentVersion\Windows Messaging Subsystem\Profiles\" _
          & ProfileName & "\9375CFF0413111d3B88A00104B2A6676"
strLastChangeVer = "LastChangeVer"
objRegistry.GetBinaryValue _
```

```
        HKEY_CURRENT_USER,strKeyProfilePath,strLastChangeVer,strValueLastChangeVer

If DEBUGME = TRUE then
        if isnull(strValueLastChangeVer) then
                wscript.echo "RegRead: Profile (LastChangeVer) does not exist."
        else
                wscript.echo "RegRead: Profile exists. LastChangeVer: " & cstr(strValueLastChangeVer(0))
        end if
end if

'Retrieve the Profile Version Number  (Modified 2/20/2014 RVI/Sumatra Development)
strKeyProfileVersionPath = "SOFTWARE\HowTo-Outlook\DeployPRF"
strProfileVersionName = ProfileName
objRegistry.GetDWORDValue _
        HKEY_CURRENT_USER,strKeyProfileVersionPath,strProfileVersionName,strValueProfileVersion

'
' THREE logical tests (Added 2/20/2014 RVI/Sumatra Development)
'1. Is the "ValueLastChangeVer" Null:  If true the profile does not exist, so create one
'2. Is the strValueProfileVersion NULL or the Profile Version is greater than the stored profile version, so
recreate it
'3. Is ReapplyPRF True (Force the reapplication of the profile, regardless)
'
If (IsNull(strValueLastChangeVer) ) OR _
  (IsNull(strValueProfileVersion) OR ProfileVersion > strValueProfileVersion) OR _
  (ReapplyPRF = TRUE) Then
        ReapplyPrf = True
ELSE
        ReapplyPrf = False
End If

If DEBUGME = TRUE then
  wscript.echo "ReapplyPRF: " & cstr(reapplyprf) & _
        "; isNull(LastChgver): " & cstr(isnull(strValueLastChangeVer)) & _
        "; ValueProfileVersion):  " & strValueProfileVersion
end if
```

```
'New logical test to see if we have to reapply the profile (Added 2/20/2014 RVI/Sumatra Development)
If ReapplyPrf Then
   'The mail profile doesn't exist yet -or we want to reapply the PRF -- set the the ImportPRF key and remove the FirstRun keys

   'Determine path to outlook.exe
   strKeyOutlookAppPath = "SOFTWARE\Microsoft\Windows\CurrentVersion\App Paths\OUTLOOK.EXE"
   strOutlookPath = "Path"
   objRegistry.GetStringValue _
         HKEY_LOCAL_MACHINE,strKeyOutlookAppPath,strOutlookPath,strOutlookPathValue

         If DEBUGME = TRUE then
               Wscript.Echo "GetOutlookPath: " & strOutlookPathValue
         end if

   'Verify that the outlook.exe and the configured prf-file exist
   Set objFSO = CreateObject("Scripting.FileSystemObject")
   If objFSO.FileExists(strOutlookPathValue & "outlook.exe") AND objFSO.FileExists(ProfilePath) Then

         'Determine version of Outlook
         strOutlookVersionNumber = objFSO.GetFileVersion(strOutlookPathValue & "outlook.exe")
         strOutlookVersion = Left(strOutlookVersionNumber, inStr(strOutlookVersionNumber, ".0") + 1)

         'Create the Setup key, set the ImportPRF value and delete the First-Run values.
         strKeyOutlookSetupPath = "SOFTWARE\Microsoft\Office\" & strOutlookVersion & "\Outlook\Setup"

         strImportPRFValueName = "ImportPRF"
         strImportPRFValue = ProfilePath
         objRegistry.CreateKey HKEY_CURRENT_USER,strKeyOutlookSetupPath
         objRegistry.SetStringValue HKEY_CURRENT_USER,_
            strKeyOutlookSetupPath,strImportPRFValueName,strImportPRFValue

         If DEBUGME = TRUE then
               If isnull(strImportPRFValue) then
                     Wscript.Echo "Error: SetProfilePath Key not defined: strImportPRFValue"
               else
                     Wscript.Echo "SetProfilePath Key: " & strImportPRFValue
               end if
```

```
            end if

            strFirstRunValueName = "FirstRun"
            objRegistry.DeleteValue HKEY_CURRENT_USER,_
               strKeyOutlookSetupPath,strFirstRunValueName

            strFirstRun2ValueName = "First-Run"
            objRegistry.DeleteValue HKEY_CURRENT_USER,_
               strKeyOutlookSetupPath,strFirstRun2ValueName

            'Save the applied ProfileVersion if larger than 1. (Added 2/20/2014 RVI/Sumatra Development)
            If (IsNull(strValueProfileVersion) OR ProfileVersion > strValueProfileVersion) Then
               objRegistry.CreateKey HKEY_CURRENT_USER,strKeyProfileVersionPath
               objRegistry.SetDWORDValue HKEY_CURRENT_USER,_
                     strKeyProfileVersionPath,strProfileVersionName,ProfileVersion
            End If

            If DEBUGME = TRUE then
                     Wscript.Echo ("Profile Created.")
            end if

   Else
      Wscript.Echo "Crucial file in script could not be found." & vbNewLine & _
               "Please contact your system administrator."
   End If

Else
            If DEBUGME = TRUE then
                     wscript.echo "We SKIPPED the addition of the profile."
            end if
   'The mail profile already exists so there is no need to launch Outlook with the profile switch.
   'Of course you are free to do something else here with the knowledge that the mail profile exists.
End If

'Cleaup
Set objRegistry = Nothing
Set objFSO = Nothing
```

Contact Sumatra Development

We never learned anything listening to ourselves talk.
We only learn it when you folks tell us what you want.
If it involves calendaring technology, feel free to contact us!

The Managing Partners of Sumatra can be reached at:
zyg@sumatra.com
riuliano@sumatra.com

Check us out at www.sumatra.com as well as

Our blog:

http://calendarservermigration.blogspot.com/

And Twitter:

http://twitter.com/sumatra_dev