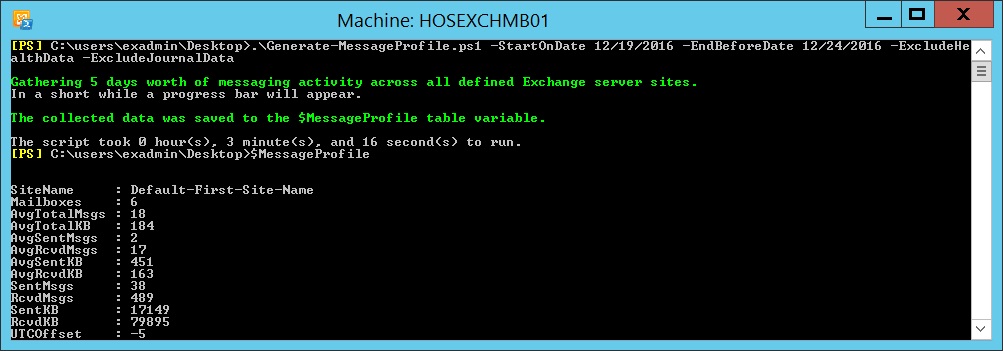
Background

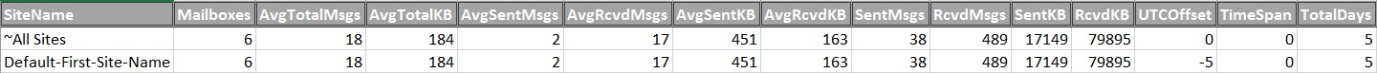
An Exchange “user message profile” represents the amount of messages a user sends and receives in a day, and the average size of those messages. This critical information is used by the Role Requirements Calculator to determine the typical workload a group of users will place on an Exchange system, which in turn is used to properly size a new Exchange environment design. This information is also used by the Client Bandwidth Calculator to estimate potential bandwidth impact email users will have on the network, depending on their client type and versions used. Some Exchange service owners “guesstimate” a couple of different user message profiles based on the anticipated workload, while others use data from their existing environment to try and create a messaging profile based on recent user activity. Gathering the necessary information based on recent user activity and creating a user message profile is not an easy task, and quite often service owners turn to third-party tools for assistance with this process. This PowerShell script was created to assist Exchange service owners who want to generate average user message profiles based upon their current environment, but don’t have or want to use a third-party tool to gather the necessary information and generate a message profile. There are other messaging statistics gathering scripts published on the Internet, such as this [this one](https://gallery.technet.microsoft.com/bb94b422-eb9e-4c53-a454-f7da6ddfb5d6) by Mjolinor on the [TechNet Gallery](https://gallery.technet.microsoft.com/) and [this one](https://techcommunity.microsoft.com/b/neiljohn/archive/2011/08/09/user-profile-analysis-for-exchange-server-2010.aspx) by our own Neil Johnson (who BTW is responsible for the Client Bandwidth Calculator). Typically, those types of “messagestats” scripts create a per-user report of all messaging activity which takes a long time, includes information beyond what is required to create a user message profile, and the output requires further manipulation to come up with an average user message profile. The Generate Message Profile script on the other hand focuses on only gathering the messages sent and received by users, which is faster than gathering all messaging activity, and provides a user message profile per Exchange (AD) site versus individual user results.

Functionality

The script uses native Exchange PowerShell cmdlets to extract the mailbox count from mailbox role servers and mailbox messaging activity from the Hub Transport role server message tracking logs for the specified date range. The information is then processed to obtain a per-site message profiles consisting of averages for sent messages, received messages, and message sizes. The script requires a start and end date, and can be run multiple times to accumulate groups/blocks of days into the final output. For instance, instead of gathering 30 straight days of data from the Exchange servers, which includes weekend days that generally negatively skew the averages due to reduced user load, the script can be run 4 consecutive times using the 4 groupings of weekdays within that 30 day period to help keep the averages reflective of a typical work day. The output to a CSV file can then be performed during the collection the 4th and final week, because the collection of message profiles are additive until the message profiles are exported to a CSV. The script can be run against Exchange servers in specific AD sites, collections of AD sites, or (the default) all AD sites, and the generated message profiles that are returned are organized by AD site name. The ability to specify a specific collection of AD sites is important for multi-site international Exchange deployments because not every location around the world follows a Monday through Friday work week. This selective site functionality can be combined with the script’s ability to accumulate and combine data from multiple data collections into a single report, even if some sites had to be queried using different date ranges. The script can optionally provide a “total” summary message profile for users across all collected sites using the site name of “~All Sites” (which will show up at the top of the output). The collected data can be exported to a CSV file at the end of each script run, otherwise it will be automatically stored as the PowerShell Console variable *$MessageProfile* for further manipulation or data collections. The script provides detailed output to the screen, including tiered progress bars indicating what site is currently being processed, how many server jobs in that site are being processed, and the running server processing results for that site.

The script output also includes an execution time summary at the end so you can plan for timing of future data gathering efforts:[](https://techcommunity.microsoft.com/legacyfs/online/media/2015/04/Example_Run.jpg)

Resultant Data

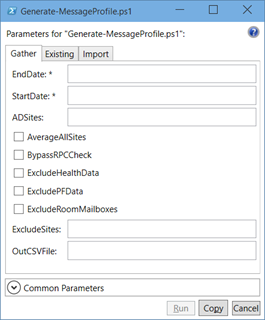
There are several script parameters (covered below) that can be used exclude certain types of mailboxes and messages from the data gathering and subsequent output of the script. For example, if you exclude all Journal data from the data gathering, then journal messages won’t be reflected resulting message profiles. This means the use of the words “all” and “total” below are in reference to the messages and mailboxes the script was told to gather and process, and not necessarily all of the data available on the servers. The data in the output is grouped into the following columns per Exchange site (as well as the optional “~All Sites” entry):[](https://techcommunity.microsoft.com/legacyfs/online/media/2015/04/Example_Output.jpg)

1. **Site Name** – This is the name of the AD site that the Exchange servers live in, as defined in AD Sites and Services.
2. **Mailboxes**– This is the count of all mailboxes discovered in the site. This information is used by both Calculators.
3. **AvgTotalMsgs**– This is the count of sent and received messages for the mailboxes in the site. This information is used by the Role Requirements Calculator.
4. **AvgTotalKB** – This is the average size in KB of all included sent and received messages in the site. This information is used by both Calculators.
5. **AvgSentMsgs** – This is the average count of sent messages for the mailboxes in the site. This information is used by the Client Network Bandwidth Calculator.
6. **AvgRcvdMsgs** – This is the average count of received messages for the mailboxes in the site. This information is used by the Client Network Bandwidth Calculator.
7. **AvgSentKB** – This is the average size in KB of sent messages for the mailboxes in the site.
8. **AvgRcvdKB** – This is the average size in KB of received messages for the mailboxes in the site.
9. **SentMsgs** – This is the total amount of sent messages for the mailboxes in the site.
10. **RcvdMsgs** – This is the total amount of received messages for the mailboxes in the site.
11. **SentKB** – This is the total size in KB of all sent messages for the mailboxes in the site.
12. **RcvdKB** – This is the total size of KB of all received messages for the mailboxes in the site.
13. **UTCOffset** – This is the UTC time zone offset for the AD site. This information is used by the Client Network Bandwidth Calculator.
14. **TimeSpan** – This represents the amount of time difference between the clock on the local computer running the script and the clock of the remote server being processed. This is informational only.
15. **TotalDays** – This represents the number of days collected for the site. This information is needed by the script when you are using it to combine multiple runs into a single output.

Parameters

The script has several parameters to allow administrators control what goes into/is excluded from the user message profile generation process. Most of the parameters are grouped into one of three “parameter sets”, with the exception of one parameter that is in 2 sets and a couple that are not in any set. Parameter sets group related parameters together, so once a parameter is one set is chosen the only other available parameters are those in that same set and those that aren’t assigned to any set. Furthermore a required parameter is only required within its parameter set, meaning if you are using one parameter set, then the required parameters in other sets don’t apply. If the concept of parameter sets is a little confusing and you are using Exchange 2013 (andlater), then you can use the PowerShell 3+ cmdlet Show-Command with the script to create a graphical representation of the parameter sets like this:

Show-Command .\Generate-MessageProfile.ps1

which will pop-up the window:[](https://techcommunity.microsoft.com/legacyfs/online/media/TNBlogsFS/prod.evol.blogs.technet.com/CommunityServer.Blogs.Components.WeblogFiles/00/00/00/31/06/metablogapi/image_73B327E1.png)The script also supports the traditional -Verbose and -Debug switches in addition to what’s listed below:

|  |  |  |  |
| --- | --- | --- | --- |
| **Parameter** | **Set** | **Required** | **Description** |
| **ADSites** | Gather | Optional | Defaults to "\*" which indicates all AD sites with Exchange should be processed. Alternatively, explicit site names, site names with wild cards, or any combination thereof can be used to specify multiple AD sites to filter on. The format for multiple sites is each site name in quotes, separated by a comma with no spaces such as: "Site1","Site2","AltSite\*", etc... |
| **StartOnDate** | Gather | Required | Specifies the date (at 12:00AM) the message tracking log search should start on. The format is MM/DD/YYYY. |
| **EndBeforeDate** | Gather | Required | Specifies the date (at 12:00AM) the message tracking log search should end before. This means that if the desired search window is Monday through Friday, Saturday needs to be specified so the search "ends before" (stops) at 12:00AM Saturday. This will allow for all of Friday to be included in the search. The format is MM/DD/YYYY. |
| **ExcludeHealthData** | Gather | Optional | Excludes messages to or from Managed Availability "HealthMailbox" and the older SCOM "extest\_" mailboxes, which could artificially inflate the message profile for a site. **NOTE**: Because the extest and HealthMailboxes can generate a lot of traffic, it is recommended to use this switch to get a more accurate message profile reflection of your users. |
| **ExcludeJournalData** | Gather | Optional | Excludes journal messages from the data collection. By default messages delivered to journal mailboxes will be included with the message profile, which could artificially inflate the message profile for a site. |
| **ExcludePFData** | Gather | Optional | Attempts to filter out messages sent to or from legacy Exchange 2007/2010 Public Folder databases. This is not needed if there are no legacy Exchange Public Folder databases. **NOTE**: This parameter is not recommended because its filter relies on message subject line filtering which could potentially filter out user messages. Additionally, this does not filter out all Public Folder messaging data because some Public Folder message subject lines were not included due to the high likelihood that users would use them in their own messages. |
| **ExcludeRoomMailboxes** | Gather | Optional | Excludes messages to or from room mailboxes. By default equipment and discovery mailboxes are excluded from the count as they negatively skew the average user message profile. Room mailboxes are included by default because they can send/receive email **NOTE:** This parameter is not recommended if you have active conference room booking in your environment as that means you have active message traffic to and from room mailboxes. |
| **BypassRPCCheck** | Gather | Optional | Instructs the script to bypass the additional RPC connectivity test to remote computers through Get-WMIObject. Basic PING tests are always used to initially test connectivity to remote computers. Bypassing the RPC check should not be necessary as long as the account running the script has the appropriate permissions to connect to WMI on the remote computers. |
| **MaxServerTries** | Gather | Optional | Specifies the maximum number of times to try to gather data from a server when there are issues gathering data. The default value of 3 means the script will try to gather data from each server up to 3 times before giving up on it and marking it as a skipped server. |
| **MinServersPercent** | Gather | Optional | Specifies the minimum percentage of servers in a site, defaulting to 100%, that must be accessible and also return data to adequately generate a message profile. If this percentage is not met, because too many servers are inaccessible or they exceed the MaxServerTries during data gathering, the site is skipped (recorded as a SkippedSite so the script can be quickly re-run against it) and not included in the final message profile collection. The format is a number value without the “%”. **NOTE:** It is highly recommended to leave this value at 100, because missing even one server could result in a potentially skewed message profile. |
| **MaxThreads** | Gather | Optional | Specifies the maximum number of simultaneous server data gathering jobs (threads). Each job increases the memory and CPU load on the server running this script. Therefore, the number of jobs defaults to 1/4 (rounded up) of logical cores if the system running the script is running Exchange services, or 1/2 if it is not. **NOTE:** Monitor CPU and memory impact and adjust as necessary. |
| **Confirm** | Gather | Optional | Bypasses the warning prompts for changes to the MinServersPercent and MaxThreads parameters. |
| **ExcludeSites** | Gather Import | Optional | Specifies which sites should be excluded from data processing. This is useful when you want to use a wild card to gather data from multiple sites, but you want to exclude specific sites that would normally be included in the wild card collection. Likewise, sites that do not house any user mailboxes, such as dedicated Hybrid sites, can be excluded. For data importing, this is useful when a site needs to be excluded from a previous collection. The format for multiple sites is each individual site name in quotes, separated by a comma with no spaces such as: "Site1","Site2", etc.. **NOTE**: Wild cards are not supported. |
| **InCSVFile** | Import | Required | Specifies the path and file name of the CSV to import previously collected data from. |
| **InMemory** | Existing | Required | Instructs the script to only use existing in memory data. This intended only to be used with the AverageAllSites parameter switch. |
| **AverageAllSites** | <None> | Optional | Instructs the script to create an "~All Sites" entry in the collection that represents an average message profile of all sites collected. If an existing "~All Sites" entry already exists, its data is overwritten with the updated data. |
| **OutCSVFile** | <None> | Optional | Specifies the path and file name of the CSV to export the collected data to. If this parameter is omitted, then the collected data is saved in the shell variable $MessageProfile. **NOTE**: Do not use this parameter if you are collecting multiple weeks of data individually, such as successive weeks to avoid weekends, until the last week so only the complete data set exported to a CSV and the $MessaProfile variable is not removed from memory. |

NOTE: This list of parameters will be updated on the TechNet Gallery posting as the script is updated.

Examples

The following are just some examples of the script being used:

1. Process Exchange servers in all sites starting on Monday 12/1/2014 through the end of Friday 12/5/2014. Export the data, excluding the message data for Exchange 2013+ HealthMailboxes and any extest\_ mailboxes, to the AllSites.CSV file:

Generate-MessageProfile.ps1 -StartOnDate 12/1/2014 -EndBeforeDate 12/6/2014 -ExcludeHealthData -OutCSVFile AllSites.CSV

2. Process Exchange servers in AD sites whose name starts with "East", starting on Monday 12/1/2014 through the end of Monday 12/1/2014 (I.E. It's data gathering for just one day). Output the additional Verbose and Debug information to the screen while the script is running. The collected data is available in the $MessageProfile variable after the script completes:

Generate-MessageProfile.ps1 -ADSites East\* -StartOnDate 12/1/2014 -EndBeforeDate 12/2/2014 -Verbose -Debug

3. Process Exchange servers in the EastDC1 AD site, and any sites that start with the name "West", starting on Monday 12/1/2014 through the end of Tuesday 12/30/2014. Export the data, which should exclude most Public Folder traffic and all Journal messages, to the MultiSites.CSV file:

Generate-MessageProfile.ps1 -ADSites "EastDC1","West\*" -StartOnDate 12/1/2014 -EndBeforeDate 12/31/2014 -OutCSVFile MultiSites.CSV -ExcludePFData -ExcludeJournalData

4. Import the data from the PreviousCollection.CSV file in the current working directory, and store it into the in-memory data collection $MessageProfile for future us::

Generate-MessageProfile.ps1 -InCSVFile .\PreviousCollection.CSV

5. Process the previously collected data stored in the in-memory $MessageProfile variable, and add an average for all sites to the collection as the site name "~All Sites":

Generate-MessageProfile.ps1 -InMemory -AverageAllSites

FAQ

**1.     Why don’t I see any per-user information? Why is this site based?**

* This script was designed to maximize speed by gathering messaging profile information on a per-site basis to facilitate the use of both the Role Requirements and Client Network Bandwidth Calculators. The Client Bandwidth Calculator wants the message profile information on a per-site basis, and the per-site basis works for the Requirements Calculator as well.
* Per-user information is not needed for either Calculator as the users should be intermixed between all databases anyway. Separate user profiles can be optionally put into each Calculator using the same message profile but reflecting other differences such as larger mailboxes or expected increases to IOPS or megacycles (such as when a group of users also using mobile devices).
* If you require per-user reporting, please use one of the scripts I referenced in the Background section.

**2.     Is the output generated by this script an accurate representation of my users’ messaging profile, which I can use in other tools such as the Role Requirements Calculator?**

* This script generates a point in time reflection of your user’s messaging activity. The data is only as good as the date range(s) you selected to run it in, the data you opted to include or exclude, and the information stored on the accessible servers. For example if you ran this script during date range that included a holiday and a lot of users took vacation, or your servers were missing message tracking logs, then the information is going to reflect a lower average message profile than a more “normal” work period would reflect.
* Taking into consideration that this script will only reflect the messaging activity of your users during your selected date range, you should use the output as a guideline for formulating the message profile to represent your users in other tools.

**3.     Should I inflate/enhance the message profile produced by this script to give myself some “elbow room” in my Exchange system design?**

* If you are designing an email system that is going to need to last for multiple years, it’s probably a good idea to increase the numbers slightly to account for future growth of your system and the likelihood that yours will increase their message profile over time. How much you inflate the information is up to you.

**4.     The messaging profile for my users seems lower than I expected. What are some factors that could attribute to this/how can I increase the values generated by the script?**

* Review the data range(s) you chose when running the script to see if they were periods of time where user activity was expected to be low.
* If your date range(s) include weekends/non-work days, re-run the script excluding those days. This may require multiple cumulative runs if you want to include multiple work weeks in the average.
* If you have a lot of resource rooms that are rarely used but you did not exclude them, then try re-running the script with the ExcludeRoomMailboxes parameter to see if the averages increase. Conversely if you used some of the script’s parameters to exclude data, re-running the query without the exclusions may increase the average as well. You will need to test various parameter combinations in your environment until you are happy with the results.
* If you recently decommissioned any Hub Transport role servers in a site, then the message tracking logs stored on those servers that provide user activity details were removed as well. Therefore, it his highly recommended that this script only be run on sites that have not had any Hub Transport role servers decommissioned during the specified time ranges. The script even has a built-in warning when it detects a Hub Transport role server was added to a site during the specified date range, to remind you that if another Hub Transport role server was recently removed from that site as well then the user message profile could be negatively affected.

**5.     Why did I get an alert that one or more sites were skipped or excluded?**

* A site will be skipped if there were connectivity issues to a percentage of its servers that exceeds the MinServersPercent parameter value (default of 100%). Since a message profile for a site should contain data from all its servers, missing data from even one server could result in incomplete information. Therefore the script will skip the site if it encounters connectivity issues to a percentage of servers exceeding the MinServersPercent value versus reporting skewed message profile data.
* A site will be excluded if there are no mailboxes or messaging activity found in it. Passive Exchange DR or dedicated Hybrid server sites with no active mailbox databases are an example of a site that will be safely excluded. Even though there may be active Hub Transport servers in those sites, their message tracking data is not needed as they will hand messages off to Hub Transport role servers in the site(s) with the target mailboxes. The logs from those final Hub Transport role servers will in turn be used for the message profile generation.
* If any sites were skipped for data collection issues, they will be recorded in a $SkippedSites variable which will be available after the script finishes. This allows you to re-run the script and specify the $SkippedSites as the value for the ADSites parameter, which causes the script to focus gathering data only from those skipped sites. This is helpful in cases where server connectivity issues were due to temporary WAN connectivity issues, and another run of the script will process those skipped sites successfully.

**6.     Why can’t I specify the hours of a day I want to be searched in addition to the days?**

* The script is designed to work with whole/entire days, not fractions of a day, to create the averages. Specifying a time of day would result in a faction of a day which is not supported in creating a “per day” user message profile average.

**7.     Why does the EndBeforeDate need to be the day following the day I want to stop reporting on?**

* When specifying only a date for a “DateTime” variable, PowerShell assigns the time for that day as 12:00AM. For the StartOnDate, that time is exactly what needs to be used as that represents the entire day starting 12:00AM. However for the EndBeforeDate this causes the data collection to stop at 12:00AM on the specified day, therefore the EndBeforeDate needs to be the day following the last day you want included in the output.
* The script has logic built in to ensure that the StartOnDate does not occur in the future, that the EndBeforeDate does not occur before the StartOnDate, the StartOnDate is at least on day prior to the current date, and that the EndBeforeDate is no later than the current date.

**8.     Why would I want to store data in a CSV file and then later import it with the script?**

* Sometimes some sites just can’t be reached over the WAN. This allows for the data collection to be performed locally on server in the remote site, and then the data transferred back to the main site via a CSV file where it can be imported into the main data collection.
* This functionality also allows you to take data collections from different points in time, such as over the course of several weeks or months, and import it into a single longer term user message profile generation.
* This functionality also allows you to take the data in-memory and remove sites from the collection by exporting it to a CSV, and then re-importing the data to a new collection and using the ExcludeSites parameter to block the import of the unwanted sites.

**9.     What is the purpose of the InMemory parameter?**

* The only reason to use this switch is if you already have your data loaded into memory, either through one or more gathering or importing processes, and want to use the AverageAllSites parameter to provide a single global user message profile under the site name of “~All Sites”. Essentially this parameter allows you to bypass gathering or importing data and just use what is already “in memory”.

**10.  Why do I get an error about “inconsistent number of days” when I try to use the AverageAllSites?**

* The process that generates a single global user message profile requires that the value for TotalDays be the same for all collected sites. Otherwise the aggregated data would be represented incorrectly because the TotalDays value is used to calculate the “per day” average. You need to review your site data, most likely by exporting it to a CSV file and reviewing it manually, to determine which sites have different TotalDays recorded and deal with them accordingly.

**11.  Why is the information saved to the $MessageProfile variable in the console if I don’t use the –OutToCSV parameter? Also how do I “wipe” the collected data from memory so I can start over?**

* Storing the data inside of PowerShell variable is necessary if you want to run the script multiple times to accumulate data, because the script uses this variable to store the cumulative data in between runs.
* This also allows you to take the in-memory $MessageProfile variable data and pass it to other PowerShell scripts or commands that you wish.
* You have the option of using the command “$MessageProfile | Export-CSV ….” to create your own CSV if you decide to later store the collected data in a CSV file.
* The $MessageProfile is cleared when you output to a CSV file, but to manually clear the $MessageProfile data from memory use the following command:

$MessageProfile = $Null **12.  Why does the output of the script include a value called “TimeSpan” and also the time zone of the remote site?**

* The time span represents the delta in hours, positive or negative, between the server running the script and the remote server it is connecting to. By default, when the Get-MessageTrackingLog cmdlet is executed against a remote server, the DateTime values used for the start and end dates passed to it are always from the perspective of the server running the cmdlet. This means that if the computer running the cmdlet is 5 hours behind the remote server, then the dates (which include a time of day) passed to that remote server by the cmdlet would actually be 5 hours behind your intended date.
* The script uses this time span to properly offset the DateTime values as they are passed to the Get-MessageTrackingLog cmdlet, so they are always processed by the remote server with the original intended dates (and the 12:00AM time of day). Following the example above, the script will add 5 hours to the date when the cmdlet is run against the remote server. Since this value is crucial to accurate script execution, it is recorded in the output for tracking purposes.
* The Client Network Bandwidth Calculator wants to know the time zone of the user message profile being specified. To facilitate use of this calculator, the site’s time zone information is recorded in the output of the script.

**13.  Why did you build in an ExcludePFData parameter switch if it doesn’t exclude all legacy Public Folder traffic?**

* Initial testing of the script showed that dedicated Public Folder servers reflected a large amount of Public Folder replication based Hub Transport messaging activity.
* Because the most accurate depiction of the user messaging profile was desired, a switch was added to try and filter out some Public Folder replication data. Since the only way to consistently identify the Public Folder traffic was by message subject line keyword matching, a filter was created that strips out messages with Public Folder replication subject phrases not likely to be used by users to try and limit accidentally stripping actual user messages.

**14.  I see Equipment and Discovery mailboxes are excluded, why aren’t Arbitration Mailboxes excluded?**

* Equipment and Discovery mailboxes do not send and receive email through the Hub Transport service, so including them would only serve to negatively impact the user message profile.
* Arbitration mailboxes on the other hand are normally limited in number and therefore including them in the mailbox count is not expected to dramatically impact the message profile in a negative way. At the same time messages can be sent to and received from Arbitration mailboxes, depending on the organization’s use of features like moderated Distribution Groups, so including them could positively impact the message profile