

## Inhalt

Die Idee .....	2
Das Problem .....	2
Die Microsoft-Lösung .....	2
Probleme mit der Microsoft-Lösung.....	4
Meine Lösung: „parse-ActiveSyncDebugLog“ .....	4
Anwendung des Scriptes .....	4
Device-Suche und Object-Ausgabe .....	4
Gridview-Ausgabe .....	5
(formatierte) Tabellen-Ausgabe .....	6
HTML-Ausgabe .....	6
weitere Parameter und deren Kombinationen .....	6

## Die Idee

### Das Problem

Exchange Server speichern Mailboxen, auf die wir gerne mit unseren Mobile Devices zugreifen. Mit einer richtigen Konfiguration ist das auch einfach möglich. Aber manchmal verweigern Smartphones die Synchronisierung. Für eine Problemdiagnose liegen dann oftmals keine sinnvollen Fehlermeldungen oder Logfiles im Gerät vor. Hier kann der Exchange Server helfen.

### Die Microsoft-Lösung

Im on-premise Exchange Server kann ab der Version 2010 das so genannte ActiveSyncDebugLog aktiviert werden. Dieses stellt dann sehr detailliert die Kommunikation zwischen Endgerät und dem ClientAccessService dar.

Wichtig sind an dieser Stelle folgende Informationen:

- Das Logging ist nicht standardmäßig aktiv
- Das Logging muss pro Mailbox aktiviert werden.
- Ein aktiviertes Logging wird sehr viele Logdaten pro Mailbox sammeln.
- Daher wird ein aktiviertes Logging vom Exchange Server nach 72 Stunden automatisch deaktiviert.
- Das Logfile liegt als Fließtext vor und kann über die PowerShell abgefragt werden.

So schaut es im Normalfall aus. Das Logging ist deaktiviert:

```
PS C:\>> Get-CASMailbox -Identity "tessa.test" | Format-List -Property *activesync*

ActiveSyncAllowedDeviceIDs      : {}
ActiveSyncBlockedDeviceIDs     : {}
ActiveSyncMailboxPolicy        : Default
ActiveSyncMailboxPolicyIsDefaulted : True
ActiveSyncDebugLogging         : False
ActiveSyncEnabled              : True
HasActiveSyncDevicePartnership  : True
ActiveSyncSuppressReadReceipt  : False
```

Mit der PowerShell kann das Logging aktiviert werden. Der Zustand wird in meinem Exchange Server aber nicht korrekt dargestellt:

```
PS C:\>> set-CASMailbox -Identity "tessa.test" -ActiveSyncDebugLogging $true
PS C:\>> Get-CASMailbox -Identity "tessa.test" | Format-List -Property "*activesync*"

ActiveSyncAllowedDeviceIDs      : {}
ActiveSyncBlockedDeviceIDs     : {}
ActiveSyncMailboxPolicy        : Default
ActiveSyncMailboxPolicyIsDefaulted : True
ActiveSyncDebugLogging         : False
ActiveSyncEnabled              : True
HasActiveSyncDevicePartnership  : True
ActiveSyncSuppressReadReceipt  : False
```

Nach der Aktivierung kann dann das Problemgerät erneut synchronisiert werden. Das Ergebnis kann mit der PowerShell abgefragt werden. Bei der Abfrage wird der Schalter -GetMailboxLog benötigt. Damit wird die zusätzliche EigenschaftMailboxLogReport ausgelesen:

```
PS C:\>> Get-MobileDeviceStatistics -Mailbox "tessa.test" -GetMailboxLog

RunspaceId      : b46732b3-9ad1-47f5-9060-06c017fbb486
FirstSyncTime   : 30.09.2020 07:24:37
LastPolicyUpdateTime : 30.09.2020 07:24:59
LastSyncAttemptTime : 19.10.2021 16:18:10
LastSuccessfulSync : 19.10.2021 16:18:10
DeviceType      : WindowsMail
DeviceID        : 155117c2317744e297c336fc48425f3a
DeviceUserAgent : MSFT-WIN-3/10.0.18363
DeviceWipeSentTime :
DeviceWipeRequestTime :
DeviceWipeAckTime :
AccountOnlyDeviceWipeSentTime :
AccountOnlyDeviceWipeRequestTime :
AccountOnlyDeviceWipeAckTime :
LastPingHeartbeat :
RecoveryPassword  : *****
DeviceModel       : Virtual Machine
DeviceIMEI        :
DeviceFriendlyName : M-CL1
DeviceOS          : windows 10.0.18363
DeviceOSLanguage  : German
DevicePhoneNumber :
MailboxLogReport  :
-----
Log Entry: 0
-----
RequestTime :
10/19/2021 18:01:26
ServerName  :
```



### Probleme mit der Microsoft-Lösung

Die Responses können sehr groß werden. Auf meinem Testclient wurde von einem vollständig im Sync stehenden Device innerhalb von 15 Minuten 2300 Zeilen Log generiert. Ebenso ist das Auffinden von Fehlern in dieser Darstellung nicht wirklich einfach.

Dazu kann ein Mailboxbenutzer durchaus mehrere ActiveSync-Devices mit seiner Mailbox verbunden haben. Die Logfiles werden pro Device gespeichert. Ohne Angabe des Devices werden also die Logdaten mehrerer Devices miteinander vermischt. Vorab muss also das ggf. passende Device gesucht werden.

### Meine Lösung: „parse-ActiveSyncDebugLog“

Die Suche nach dem richtigen ActiveSync-Device ist kein wirkliches Problem. Das Lesen und Analysieren des Logtextes aber schon. Und da setzt meine Lösung auf.

Der Text im Logfile folgt einer gewissen Struktur. Meine PowerShell-Funktion parst diese Struktur und erstellt daraus ein PowerShell-Object. Dieses lässt sich wie gewohnt filtern, sortieren bzw. weiterverarbeiten. Dazu habe ich verschiedene Ausgabeoptionen, wie eine optimierte Tabellenausgabe und eine GridView-Ausgabe integriert. Und eine Suche für das ActiveSync-Device ist natürlich auch dabei 😊.

Wie gewohnt sind im oberen Teil des Scriptes die Arbeitsfunktionen enthalten. Meine Funktion „verbinde-MX“ stellt die Verbindung zum Exchange Server her. Im unteren Teil habe ich einige Beispielaufrufe abgelegt. Und natürlich ist auch die Aktivierung des Loggings mit dabei:

```

ActiveSyncDebugLogging.ps1 X
1 ##### Scriptinfo #####
2 # Scriptreihe:      parse-ActiveSyncDebugLog
3 # Datum:           2020-10-22
4 # Version:         v1.01
5 # Programmierer:   Stephan Walther (WS IT-Solutions)
6 #####
7
8 # Funktionen
9 function verbinde-MX {...}
173
174 function parse-ActiveSyncDebugLog {...}
440
441     break
442
443 # Vorbereitung
444     verbinde-MX
445
446     Set-CASMailbox -Identity tessa.test -ActiveSyncDebugLogging $true
447
448 # Aufrufe
449     Get-Help -Name parse-ActiveSyncDebugLog -showwindow
450
451     parse-ActiveSyncDebugLog -Mailbox "tessa.test"
452     parse-ActiveSyncDebugLog -Mailbox "tessa.test" -verbose
453     parse-ActiveSyncDebugLog -Mailbox "tessa.test" -output gridview
454     parse-ActiveSyncDebugLog -Mailbox "tessa.test" -output FormatTable
455     parse-ActiveSyncDebugLog -Mailbox "tessa.test" -output html
456     parse-ActiveSyncDebugLog -Mailbox "tessa.test" -AllDevices -output FormatTable
457
458 # FailChecks
459     parse-ActiveSyncDebugLog -Mailbox "tessa.test" -DeviceID 12345
460     parse-ActiveSyncDebugLog -Mailbox "keine.Mailbox"
461     parse-ActiveSyncDebugLog -Mailbox "Paul.Paulsen" # kein AS-Device
  
```

## Anwendung des Scriptes

### Device-Suche und Object-Ausgabe

Nachdem das Logging aktiviert und ein Sync durchgeführt wurde, kann die Funktion für die Mailboxbenutzerin in meinem Beispiel aufgerufen werden. Wie eingangs erwähnt kann ein Benutzer mehrere Devices verbunden haben. Und hier gibt es dann verschiedene Szenarien:

- Findet meine Funktion kein Device, wird einfach eine Fehlermeldung zurückgegeben.
- Wird nur ein Device gefunden, dann wird für dieses Device direkt die Ausgabe des Logfiles angezeigt.
- Sind mehrere Devices registriert, dann wird über ein GridView die Device-Abfrage gestartet. Es sind hier ein oder mehrere Devices möglich. Die Informationen zum Auffinden des richtigen Devices werden natürlich mit angezeigt:



### (formatierte) Tabellen-Ausgabe

Ebenso ist aber auch eine tabellarische Anzeige sinnvoll. Natürlich könnte der Befehl auch an Format-Table gepiped werden. Aber das Ausgabeobject hat sehr viele Spalten. Viele werden nicht direkt weiterhelfen. Um eine aufwändige Aussortierung zu ersparen habe ich hinter dem Parameter **-output FormatTable** eine vorgefilterte Tabelle verbaut. So gelingt der Überblick sehr einfach:

```
PS C:\>> parse-ActiveSyncDebugLog -Mailbox "tessa.test" -output FormatTable
```

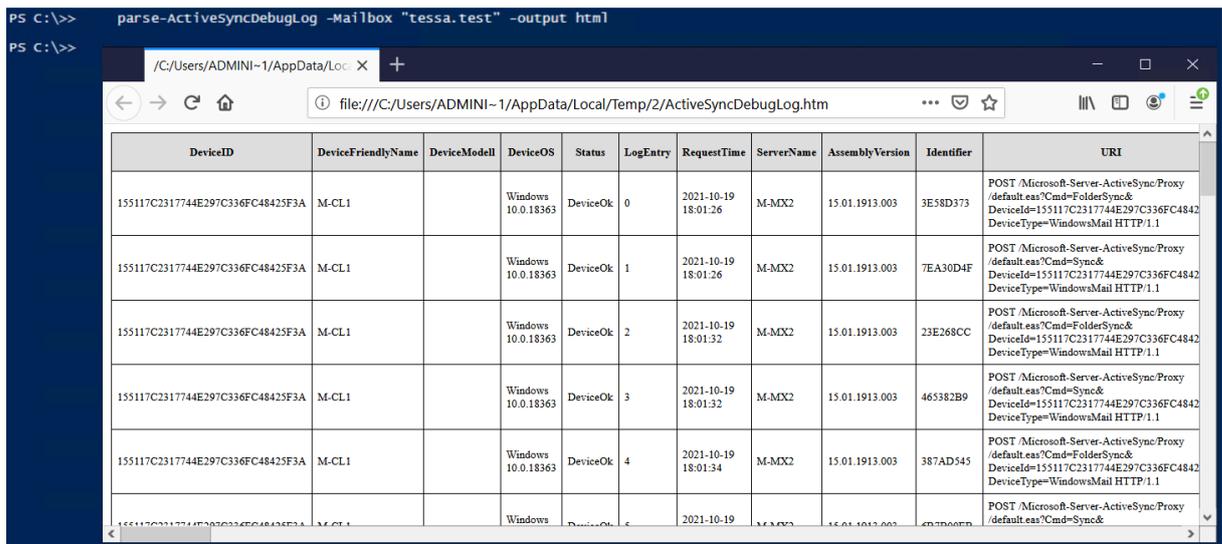
DeviceFriendlyName	LogEntry	RequestTime	Latency	ServerName	ResponseHeader	AccessState	AccessStateReason	X-Fe-ClientIP	X-OriginalRequestHost
M-CL1	0	2021-10-19 18:01:26	0 M-MX2	HTTP/1.1 200 OK	Allowed	Global	192.168.101.254	mail.crashwork.de	
M-CL1	1	2021-10-19 18:01:26	0 M-MX2	HTTP/1.1 200 OK	Allowed	Global	192.168.101.254	mail.crashwork.de	
M-CL1	2	2021-10-19 18:01:32	0 M-MX2	HTTP/1.1 200 OK	Allowed	Global	192.168.101.254	mail.crashwork.de	
M-CL1	3	2021-10-19 18:01:32	0 M-MX2	HTTP/1.1 200 OK	Allowed	Global	192.168.101.254	mail.crashwork.de	
M-CL1	4	2021-10-19 18:01:34	0 M-MX2	HTTP/1.1 200 OK	Allowed	Global	192.168.101.254	mail.crashwork.de	
M-CL1	5	2021-10-19 18:01:34	0 M-MX2	HTTP/1.1 200 OK	Allowed	Global	192.168.101.254	mail.crashwork.de	
M-CL1	6	2021-10-19 18:05:42	1 M-MX2	HTTP/1.1 200 OK	Allowed	Global	192.168.101.254	mail.crashwork.de	
M-CL1	7	2021-10-19 18:05:43	0 M-MX2	HTTP/1.1 200 OK	Allowed	Global	192.168.101.254	mail.crashwork.de	
M-CL1	8	2021-10-19 18:06:10	0 M-MX2	HTTP/1.1 200 OK	Allowed	Global	192.168.101.254	mail.crashwork.de	
M-CL1	9	2021-10-19 18:06:10	1 M-MX2	HTTP/1.1 200 OK	Allowed	Global	192.168.101.254	mail.crashwork.de	
M-CL1	10	2021-10-19 18:06:11	0 M-MX2	HTTP/1.1 200 OK	Allowed	Global	192.168.101.254	mail.crashwork.de	
M-CL1	11	2021-10-19 18:12:54	0 M-MX2	HTTP/1.1 200 OK	Allowed	Global	192.168.101.254	mail.crashwork.de	
M-CL1	12	2021-10-19 18:12:54	1 M-MX2	HTTP/1.1 200 OK	Allowed	Global	192.168.101.254	mail.crashwork.de	
M-CL1	13	2021-10-19 18:17:07	0 M-MX2	HTTP/1.1 200 OK	Allowed	Global	192.168.101.254	mail.crashwork.de	
M-CL1	14	2021-10-19 18:17:47	0 M-MX2	HTTP/1.1 200 OK	Allowed	Global	192.168.101.254	mail.crashwork.de	
M-CL1	15	2021-10-19 18:17:51	0 M-MX2	HTTP/1.1 200 OK	Allowed	Global	192.168.101.254	mail.crashwork.de	
M-CL1	16	2021-10-19 18:17:51	0 M-MX2	HTTP/1.1 200 OK	Allowed	Global	192.168.101.254	mail.crashwork.de	
M-CL1	17	2021-10-19 18:18:10	0 M-MX2	HTTP/1.1 200 OK	Allowed	Global	192.168.101.254	mail.crashwork.de	
M-CL1	18	2021-10-19 18:18:10	0 M-MX2	HTTP/1.1 200 OK	Allowed	Global	192.168.101.254	mail.crashwork.de	

Achtung: Diese Ausgabeform kann nicht mehr mit der Pipeline weiterverarbeitet werden!

### HTML-Ausgabe

Vielleicht ist euch die Visualisierung im Browser lieber? Kein Problem: mit **-output html** wird eine html-Datei erzeugt und automatisch im Standard-Browser geöffnet:

```
PS C:\>> parse-ActiveSyncDebugLog -Mailbox "tessa.test" -output html
```



DeviceID	DeviceFriendlyName	DeviceModell	DeviceOS	Status	LogEntry	RequestTime	ServerName	AssemblyVersion	Identifier	URI
155117C2317744E297C336FC48425F3A	M-CL1		Windows 10.0.18363	DeviceOk	0	2021-10-19 18:01:26	M-MX2	15.01.1913.003	3E58D373	POST /Microsoft-Server-ActiveSyncProxy/default.eas?Cmd=FolderSync&DeviceId=155117C2317744E297C336FC48425F3A&DeviceType=WindowsMail HTTP/1.1
155117C2317744E297C336FC48425F3A	M-CL1		Windows 10.0.18363	DeviceOk	1	2021-10-19 18:01:26	M-MX2	15.01.1913.003	7EA30D4F	POST /Microsoft-Server-ActiveSyncProxy/default.eas?Cmd=Sync&DeviceId=155117C2317744E297C336FC48425F3A&DeviceType=WindowsMail HTTP/1.1
155117C2317744E297C336FC48425F3A	M-CL1		Windows 10.0.18363	DeviceOk	2	2021-10-19 18:01:32	M-MX2	15.01.1913.003	23E268CC	POST /Microsoft-Server-ActiveSyncProxy/default.eas?Cmd=FolderSync&DeviceId=155117C2317744E297C336FC48425F3A&DeviceType=WindowsMail HTTP/1.1
155117C2317744E297C336FC48425F3A	M-CL1		Windows 10.0.18363	DeviceOk	3	2021-10-19 18:01:32	M-MX2	15.01.1913.003	465382B9	POST /Microsoft-Server-ActiveSyncProxy/default.eas?Cmd=Sync&DeviceId=155117C2317744E297C336FC48425F3A&DeviceType=WindowsMail HTTP/1.1
155117C2317744E297C336FC48425F3A	M-CL1		Windows 10.0.18363	DeviceOk	4	2021-10-19 18:01:34	M-MX2	15.01.1913.003	387AD545	POST /Microsoft-Server-ActiveSyncProxy/default.eas?Cmd=FolderSync&DeviceId=155117C2317744E297C336FC48425F3A&DeviceType=WindowsMail HTTP/1.1
155117C2317744E297C336FC48425F3A	M-CL1		Windows 10.0.18363	DeviceOk	5	2021-10-19 18:01:34	M-MX2	15.01.1913.003	6D7D606D	POST /Microsoft-Server-ActiveSyncProxy/default.eas?Cmd=Sync&DeviceId=155117C2317744E297C336FC48425F3A&DeviceType=WindowsMail HTTP/1.1

### weitere Parameter und deren Kombinationen

Natürlich ist eine Hilfe in der Funktion enthalten. Hier gibt es ausführliche Erklärungen und viele Beispiele zum Verständnis:

```
PS C:\> get-help -Name parse-ActiveSyncDebugLog -Showwindow
```

PS C:\>

**Synopsis**  
Die Funktion kann die ActiveSyncDebuglogs einer Mailbox auslesen und parsen.

**Description**  
Mit dieser Funktion kann von einer Mailbox für ein oder mehrere auswählbare ActiveSync-Device das ActiveSyncDebugLog ausgelesen werden. Normalerweise liegen diese Daten in einem Fließtext vor. Die Funktion parst die Informationen aus dem Text heraus und gibt sie als PowerShell-Objekt strukturiert zurück.

**Wichtig:** Für das Auslesen des Logs sind 2 Voraussetzungen zu erfüllen:  
 1) Das ActiveSyncDebugLog muss aktiviert sein.  
 2) Nach der Aktivierung muss ein Synchronisierungsversuch vom Device erfolgen, der dann protokolliert wird

Das DebugLog muss pro Mailbox mit diesem Aufruf aktiviert werden:  
`Set-CASMailbox -Identity "Tessa.Test" -ActiveSyncDebugLogging $true`

Nach 72 Stunden deaktiviert der Exchange Server das Logging für die Mailbox automatisch. Nach der Analyse sollte es aber manuell mit Set-CASMailbox wieder deaktiviert werden.

**Parameters**  
`-Mailbox <String>`

100%