Microsoft Outlook

- Get calendar entries via PowerShell
- Get task entries via PowerShell
- Update mailbox data via PowerShell

Get calendar entries via PowerShell

> Requirements: Basic PowerShell knowhow and Outlook Application.

This documentation shows how to read calendar data from an Outlook attached mailbox in PowerShell.

Use case

This Outlook API can be used to migrate or evaluate the calendar data inside of an Outlook mapped mailbox. This data can be valuable, for example, when time evaluations should be made using the calendar.

Custom calendar entries

This PowerShell code gets all entries from a new custom calendar of the users mailbox.

```
$UPN = "<upnofmailboxuser>"
$Calendarname = "<calendarname>"
$ol = New-Object -comobject Outlook.Application
$ns = $ol.GetNamespace('MAPI')
$folder = $ns.Folders.Item("$UPN").Folders.Item('Calendar').Folders.Item($Calendarname)
$AllItems = $folder.items |  Select-Object -Property Subject, Start, Duration
```

Default calendar entries

This PowerShell code gets all entries from the default calendar of the users mailbox.

```
$UPN = "<upnofmailboxuser>"
$Calendarname = "<calendarname>"
$ol = New-Object -comobject Outlook.Application
```

```
$ns = $ol.GetNamespace('MAPI')
$folder = $ns.Folders.Item("$UPN").Folders.Item('Calendar')
$AllItems = $folder.items |  Select-Object -Property Subject, Start, Duration
```

Get task entries via PowerShell

> Requirements: Basic PowerShell knowhow and Outlook Application.

This documentation shows how to read task data from an Outlook attached mailbox in PowerShell.

Use case

This Outlook API can be used to migrate or evaluate the task data inside of an Outlook mapped mailbox. This data can be valuable, for example, when project evaluation should be made based on data from tasks.

Custom task list entries

This PowerShell code gets all entries from a new custom task list of the users mailbox.

```
$UPN = "<upnofmailboxuser>"
$Calendarname = "<calendarname>"
$ol = New-Object -comobject Outlook.Application
$ns = $ol.GetNamespace('MAPI')
$folder = $ns.Folders.Item("$UPN").Folders.Item('tasks').Folders.Item($Calendarname)
$AllItems = $folder.items |  Select-Object -Property Subject, Start, Duration
```

Default task list entries

This PowerShell code gets all entries from the default task list of the users mailbox.

```
$UPN = "<upnofmailboxuser>"
$Calendarname = "<calendarname>"
$ol = New-Object -comobject Outlook.Application
$ns = $ol.GetNamespace('MAPI')
```

```
$folder = $ns.Folders.Item("$UPN").Folders.Item('tasks')
$AllItems = $folder.items |  Select-Object -Property Subject, Start, Duration
```

Update mailbox data via PowerShell

This documentation contains code blocks in PowerShell to get better acquainted with the Outlook API and describes how data can be modified or written.

Use case

This interface can be handy when calendar items or task folders need to be moved to other folders or when items with certain conditions need to be renamed accordingly.

Update task property

With this PowerShell script you can append a string to each entry in an Outlook task list. This PowerShell code can be modified to append or rename certain entries based on conditions. This code block is only there to help you understand how to write task data using the Outlook API.

```
$UPN = "<upnofmailboxuser>"
$TaskFolderName = "<taskfoldername>"
$ol = New-Object -comobject Outlook.Application
$ns = $ol.GetNamespace('MAPI')
$tasks = $ns.Folders.Item("$UPN").Folders.Item("tasks").Folders.Item("$TaskFolderName").Items
foreach($task in $tasks)
{
   $task.subject = $task.subject + "<addedstring>"
   $task.save()
}
```

On the variable $task you can modify much more than just the property "subject" as described in this code snippet. The object $task has the following properties, some of them are read only.

```
Application               : Microsoft.Office.Interop.Outlook.ApplicationClass
Class                     : 48
Session                   : Microsoft.Office.Interop.Outlook.NameSpaceClass
Parent                    : System.__ComObject
```

Actions : System.__ComObject
Attachments : System.__ComObject
BillingInformation :
Body :
Categories :
Companies :
ConversationIndex : 01D7B5E1EE74C215AED8CE57C14A94EE669B2CE739AC
ConversationTopic :
CreationTime : 30.09.2021 12:00:00
EntryID :
00000000BDDCD184419E744399016488F52BF3350700709714C02317F5438DBD0748BFEB3A0C0000DE20368
70000709714C02317F5438DBD0748BFEB3A0C0002974E719A0000
FormDescription : System.__ComObject
GetInspector : System.__ComObject
Importance : 1
LastModificationTime : 30.09.2021 15:40:50
MAPIOBJECT : System.__ComObject
MessageClass : IPM.Task
Mileage :
NoAging : False
OutlookInternalVersion : 0
OutlookVersion :
Saved : True
Sensitivity : 0
Size : 7514
Subject : <tasksubject>
UnRead : False
UserProperties : System.__ComObject
ActualWork : 0
CardData :
Complete : True
Contacts : {}
ContactNames :
DateCompleted : 30.09.2021 00:00:00
DelegationState : 0
Delegator :
DueDate : 01.01.4501 00:00:00
IsRecurring : False
Ordinal : 2147483647
Owner : <taskowner>

```
Ownership                  : 0
PercentComplete            : 100
Recipients                 : System.__ComObject
ReminderTime               : 01.01.4501 00:00:00
ReminderOverrideDefault    : False
ReminderPlaySound          : False
ReminderSet                : False
ReminderSoundFile          :
ResponseState              : 0
Role                       :
SchedulePlusPriority       :
StartDate                  : 01.01.4501 00:00:00
Status                     : 2
StatusOnCompletionRecipients :
StatusUpdateRecipients     :
TeamTask                   : False
TotalWork                  : 0
Links                      :
DownloadState              : 1
ItemProperties             : System.__ComObject
InternetCodepage           : 20127
MarkForDownload            : 0
IsConflict                 : False
AutoResolvedWinner         : False
Conflicts                  : System.__ComObject
PropertyAccessor           : System.__ComObject
SendUsingAccount           : System.__ComObject
ToDoTaskOrdinal            : 01.01.4501 00:00:00
ConversationID             : C215AED8CE57C14A94EE669B2CE739AC
RTFBody                    : {123, 92, 114, 116...}
```

Update calendar entry property

With this PowerShell script you can append a string to each entry in an Outlook calendar. This PowerShell code can be modified to append or rename certain entries based on conditions. This code block is only there to help you understand how to write calendar data using the Outlook API.

```
$UPN = "<upnofmailboxuser>"
$Calendarname = "<calendarname>"
```

```
$ol = New-Object -comobject Outlook.Application
$ns = $ol.GetNamespace('MAPI')
$calendarentries = $ns.Folders.Item("$UPN").Folders.Item('Calendar').Folders.Item($Calendarname).Items
foreach($calendarentry in $calendarentries){
$calendarentry.Subject = $calendarentry.Subject+ "<addedstring>"
$calendarentry.Save()
}
```

On the variable $calendarentry you can modify much more than just the property "subject" as described in this code snippet. The object $calendarentry has the following properties, some of them are read only.

```
Application                : Microsoft.Office.Interop.Outlook.ApplicationClass
Class                      : 26
Session                    : Microsoft.Office.Interop.Outlook.NameSpaceClass
Parent                     : System.__ComObject
Actions                    : System.__ComObject
Attachments                : System.__ComObject
BillingInformation         :
Body                       :
Categories                 :
Companies                  :
ConversationIndex          : 01D7B5EF7B35E7MEID40851A74372A7528A4DBD0B01D4
ConversationTopic          : <calendarentrysubject>
CreationTime               : 30.09.2021 13:37:02
EntryID                    :
00000000BDDCD184419E744399016488F52BF3350700709714C02317F5438DBD0748BFEB3A0C00000000010
D0000709714C02317F5438DBD0748BFEB3A0C00025A3F31BB0000
FormDescription            : System.__ComObject
GetInspector               : System.__ComObject
Importance                 : 1
LastModificationTime       : 20.10.2022 07:53:55
MAPIOBJECT                 : System.__ComObject
MessageClass               : IPM.Appointment
Mileage                    :
NoAging                    : False
OutlookInternalVersion     : 1615629
OutlookVersion             : 16.0
Saved                      : True
Sensitivity                : 0
Size                       : 6644
```

```
Subject                    : <calendarentrysubject>
UnRead                     : False
UserProperties             : System.__ComObject
AllDayEvent                : False
BusyStatus                 : 2
Duration                   : 15
End                        : 30.09.2021 13:45:00
IsOnlineMeeting            : False
IsRecurring                : False
Location                   :
MeetingStatus              : 0
NetMeetingAutoStart        : False
NetMeetingOrganizerAlias   :
NetMeetingServer           :
NetMeetingType             : 0
OptionalAttendees          :
Organizer                  : <calendarentryorganizer>
Recipients                 : System.__ComObject
RecurrenceState            : 0
ReminderMinutesBeforeStart : 5
ReminderOverrideDefault    : False
ReminderPlaySound          : False
ReminderSet                : False
ReminderSoundFile          :
ReplyTime                  : 01.01.4501 00:00:00
RequiredAttendees          : <calendarentryrequiredattendees>
Resources                  :
ResponseRequested          : True
ResponseStatus             : 0
Start                      : 30.09.2021 13:30:00
NetMeetingDocPathName      :
NetShowURL                 :
Links                      :
ConferenceServerAllowExternal : False
ConferenceServerPassword   :
ItemProperties             : System.__ComObject
EndTimeZone                : System.__ComObject
ConversationID             : E7MEID40851A74372A7528A4DBD0B01D4
RTFBody                    : {123, 92, 114, 116...}
BodyFormat                 : 2
DoNotForwardMeeting        : False
```

```
FOthersAppt                 : False
FDelegateAppt               : False
OnlineMeetingProvider       : 5
DefaultOnlineMeetingEnabled : False
```