

件名、本文無しのメール対策

※テスト環境

Windows11Pro、Outlook2021

他の環境で（特にOutlookのバージョン）動作するかは未確認です

Outlook のVBAコードの追加

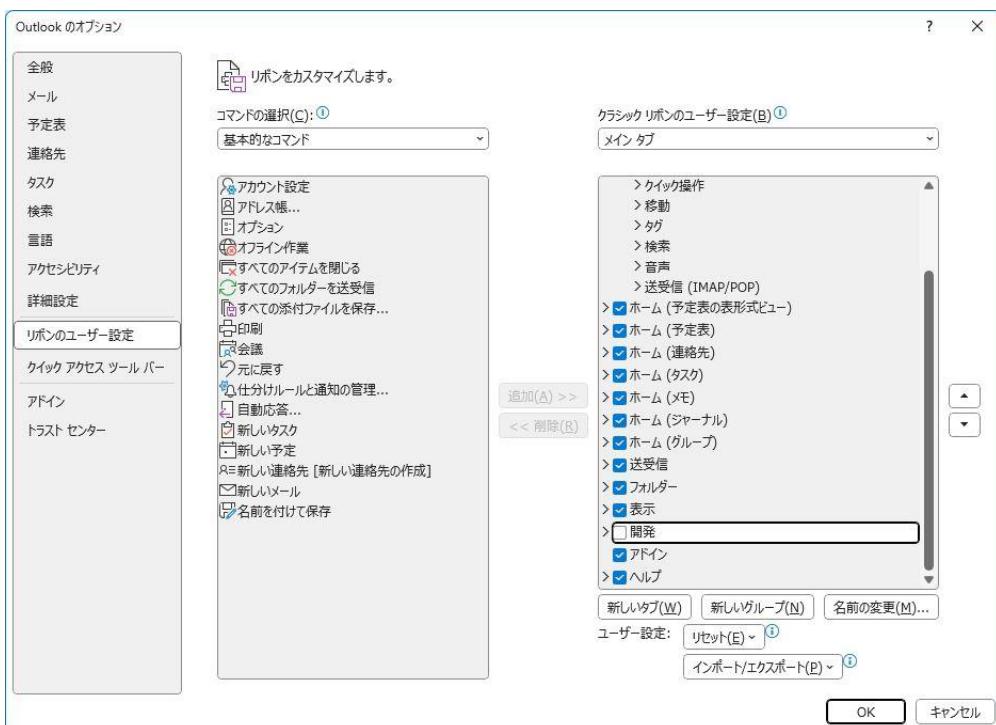
1. Outlookのオプション設定でリボンに「開発」を表示させた方がVBAを開くのに便利ですので、まずリボンの追加をします

必要のない方はAlt+F11でVBAを起動してください

Outlookのメニュー（左上）の「ファイル」をクリック

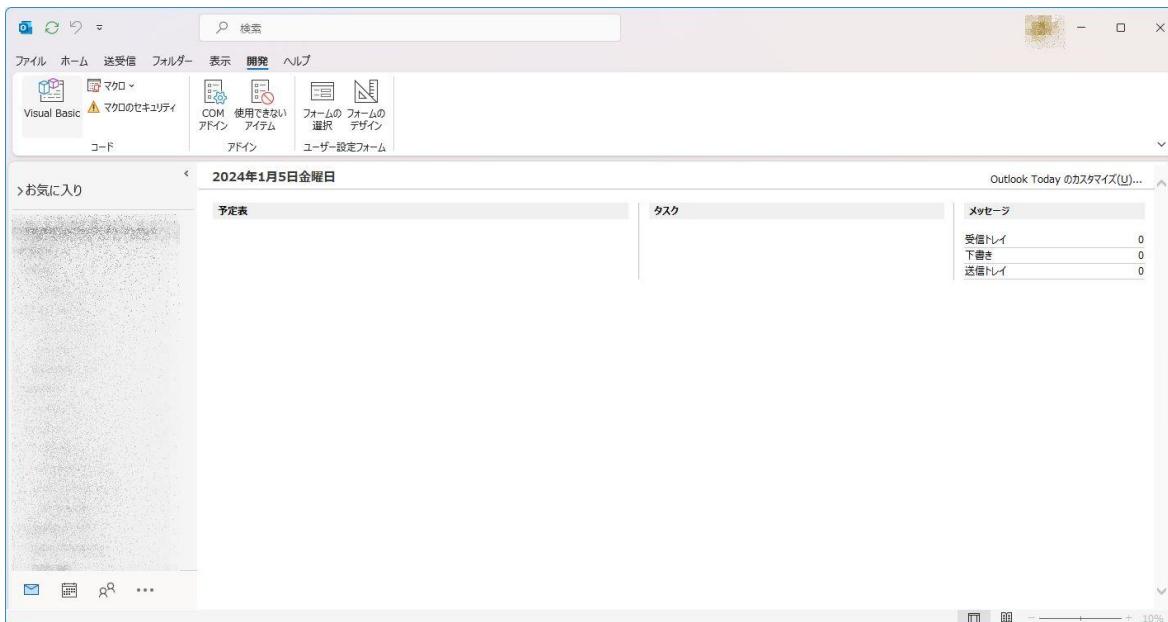


左下の「オプション」をクリック

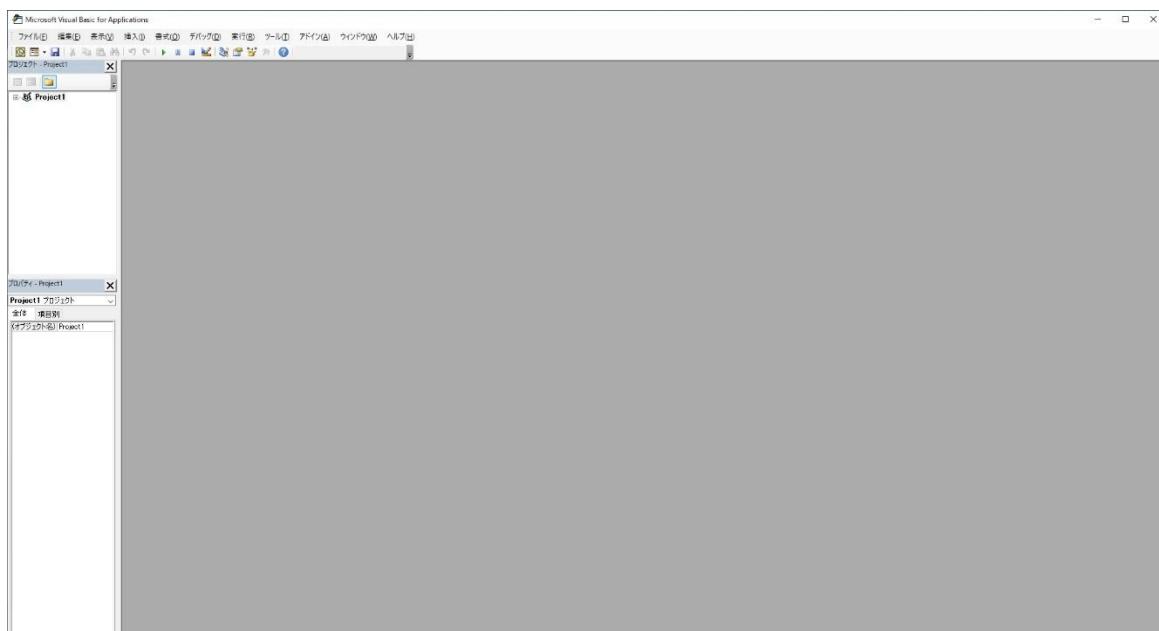


「リボンのユーザー設定」をクリックし、画面右にあるメインタブの「開発」をクリックしチェックを入れる
「OK」をクリックすれば完了です

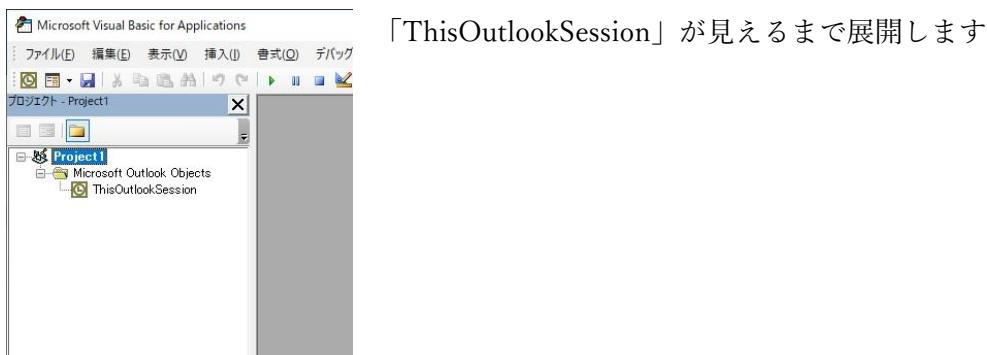
メニューに「開発」が表示されたらクリック



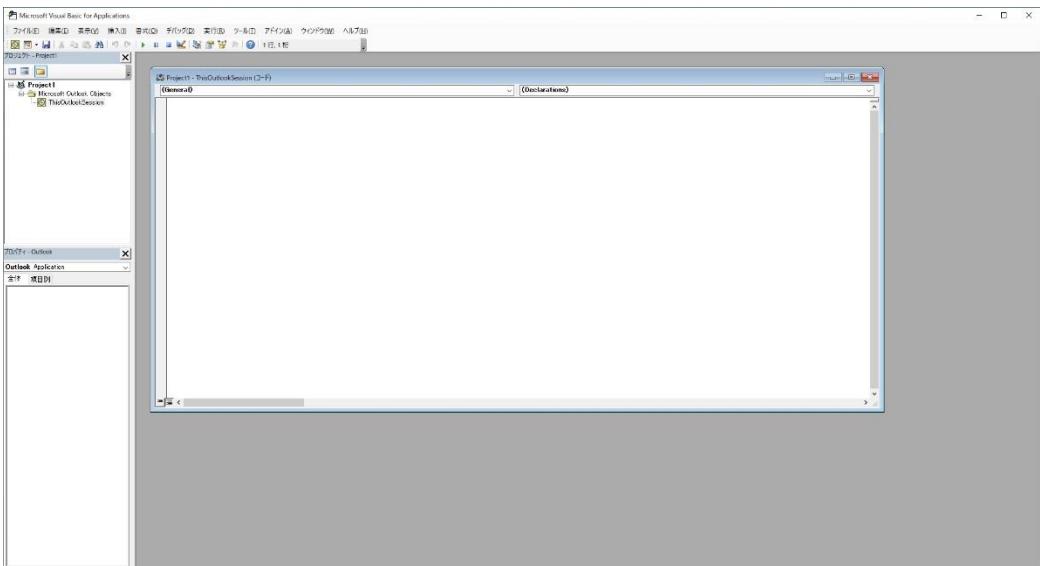
左上の「Visual Basic」をクリック



上記のような画面が出たら、左上の「Project1」の右の+をクリックし



「ThisOutlookSession」を W クリックしてコードの入力ができるようにします



2. VBA コードのコピー・ペースト

以下の VBA コードを全てコピーし、表示されている VBA の白い枠に貼り付けます

```
Private WithEvents inboxItems As Items
Private WithEvents sentItems As Items

Private Sub Application_Startup()
    ' Outlook が起動したときに呼び出される
    Dim outlookApp As Object
    Dim accounts As Object
    Dim account As Object

    Set outlookApp = Outlook.Application
    Set accounts = outlookApp.Session.Accounts

    ' 各アカウントの受信トレイに対して処理を実行
    For Each account In accounts
        ' アカウントごとに受信トレイのフォルダ ID を取得
        Dim inboxFolderId As String
        inboxFolderId = GetInboxFolderId(account)

        ' フォルダ ID が取得できた場合のみ処理を実行
        If inboxFolderId <> "" Then
            ProcessFolders outlookApp.Session.GetFolderFromID(inboxFolderId)
        End If
    Next account

    ' 送信トレイのアイテムに対して処理を実行
    Set sentItems = outlookApp.Session.GetDefaultFolder(5).Items ' 5 は送信トレイのフォルダ ID
```

```

ProcessExistingItems sentItems
End Sub

Private Function GetInboxFolderId(ByVal account As Object) As String
    ' アカウントに対応する受信トレイのフォルダ ID を取得する関数
    On Error Resume Next
    Dim inboxFolder As Object
    Set inboxFolder = account.DeliveryStore.GetDefaultFolder(6) ' 6 は受信トレイのフォルダ ID
    On Error GoTo 0

    If Not inboxFolder Is Nothing Then
        GetInboxFolderId = inboxFolder.EntryID
    Else
        GetInboxFolderId = ""
    End If
End Function

Private Sub inboxItems_ItemAdd(ByVal Item As Object)
    ' 新しいアイテムが受信トレイに追加されたときに呼び出される
    ProcessItem Item
End Sub

Private Sub sentItems_ItemAdd(ByVal Item As Object)
    ' 新しいアイテムが送信トレイに追加されたときに呼び出される
    ProcessSentItem Item
End Sub

Private Sub ProcessFolders(ByVal parentFolder As Object)
    ' フォルダ内のメールに対して処理を実行
    Dim items As Items
    Set items = parentFolder.Items
    items.Sort "[ReceivedTime]", True ' 日付を降順にソート

    ProcessExistingItems items

    ' サブフォルダが存在する場合は、そのサブフォルダに対して再帰的に処理を実行
    Dim subfolder As Object
    For Each subfolder In parentFolder.Folders
        ' サブフォルダ内に 1 週間以内のメールが存在しない場合はスキップ
        If HasRecentMail(subfolder) Then
            ProcessFolders subfolder
        End If
    Next
End Sub

```

```
Next subfolder
```

```
End Sub
```

```
Private Function HasRecentMail(ByVal folder As Object) As Boolean
```

```
    ' フォルダ内に1週間以内のメールが存在するかどうかを確認する関数
```

```
    Dim items As Items
```

```
    Set items = folder.Items
```

```
    items.Sort "[ReceivedTime]", True ' 日付を降順にソート
```

```
    Dim oldestDate As Date
```

```
    oldestDate = DateValue(Now - 7) ' 一週間以前の日付
```

```
On Error Resume Next
```

```
HasRecentMail = (items.Count > 0) And (items(1).ReceivedTime >= oldestDate)
```

```
On Error GoTo 0
```

```
End Function
```

```
Private Sub ProcessExistingItems(ByVal items As Items)
```

```
    ' 既に受信済みのメールに対して処理を実行
```

```
    Dim item As Object
```

```
    Dim oldestDate As Date
```

```
    oldestDate = Now - 7 ' 一週間以前の日付
```

```
On Error Resume Next
```

```
For Each item In items
```

```
    ' メールの日付が一週間以上前であれば処理を中断
```

```
    If item.ReceivedTime <= oldestDate Then
```

```
        Exit For
```

```
    End If
```

```
    ' メールのタイトルと本文が空であるかどうかを確認して処理を実行
```

```
If Len(item.Subject) = 0 And Len(item.Body) = 0 Then
```

```
    Debug.Print "Processing item with no subject: " & item.Subject
```

```
    ' メールを既読にする
```

```
    item.UnRead = False
```

```
    ' メールを削除済みアイテムフォルダに移動する
```

```
    On Error Resume Next
```

```
    item.Move Outlook.Application.GetNamespace("MAPI").GetDefaultFolder(3) ' 3 は削除済みアイテムフォルダのフォルダ ID
```

```
    If Err.Number <> 0 Then
```

```

        Debug.Print "Error moving item: " & Err.Description
    End If
    On Error GoTo 0
End If
Next item
On Error GoTo 0
End Sub

```

```

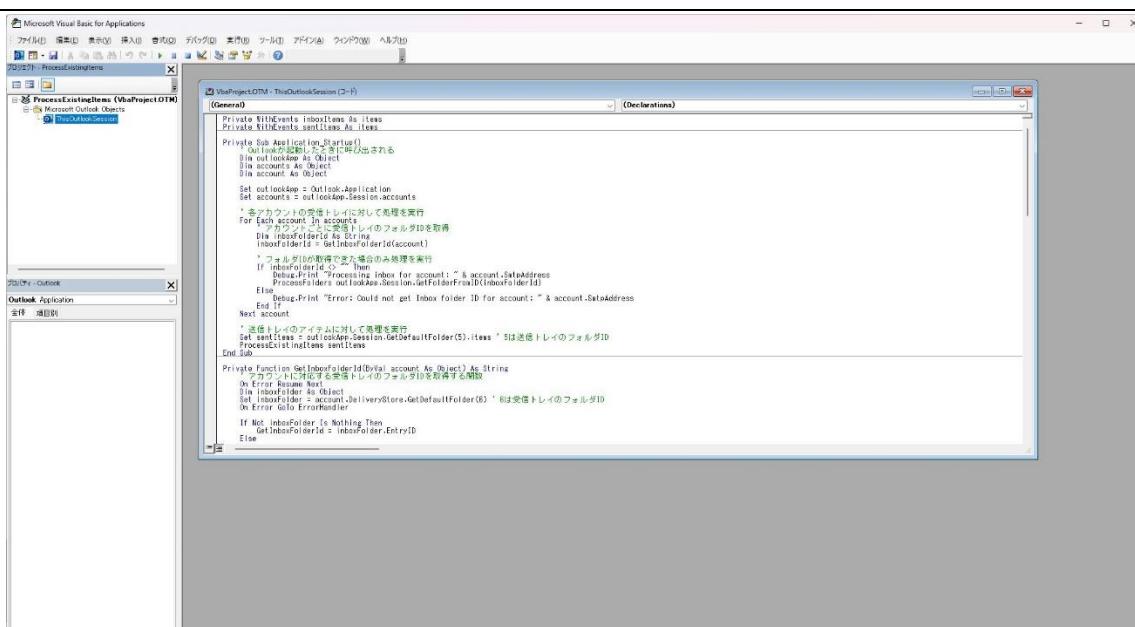
Private Sub ProcessItem(ByVal Item As Object)
    ' 受信したメールの日付が一週間以上で、かつ件名が空である場合
    If Len(Item.Subject) = 0 Then
        ' メールを既読にする
        Item.UnRead = False
        ' メールを次のフォルダに移動する（ここでは“処理済み”フォルダに移動）
        Item.Move Outlook.Application.GetNamespace("MAPI").GetDefaultFolder(4) ' 4は処理済みフォルダのフォルダID
    End If
End Sub

```

```

Private Sub ProcessSentItem(ByVal Item As Object)
    ' 送信トレイのメールに対して処理を実行
    If Len(Item.Subject) = 0 Then
        ' メールを削除する
        Item.Delete
    End If
End Sub

```

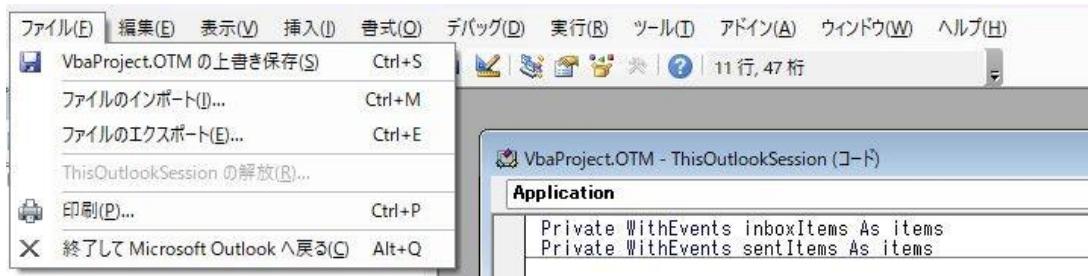


このようになれば OK です

一旦、保存して VBA は終了します

メニューの「ファイル」から一番上の「上書き保存」をしてください

続けて VBA 画面の右上の×で VBA の画面を閉じます



メニューの「ファイル」から一番下の「終了して Microsoft Outlook へ戻る」でも OK です

デジタル証明書の設定

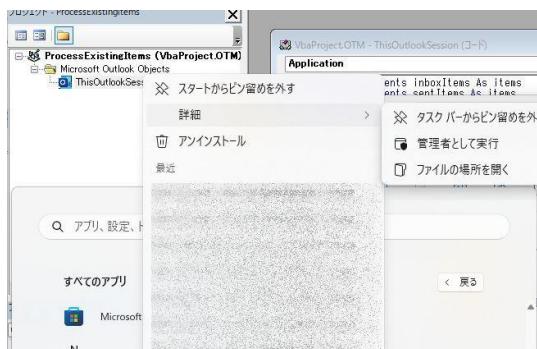
1. デジタル証明書の作成

Office 製品はセキュリティ対策で標準設定ではマクロは無効になります

そのため、2. で作ったコードは実行されません

デジタル署名があれば、実行するかどうかの確認がでて、承認したらそれ以後マクロが自動で動作するようになりますので、そのためにデジタル署名を作成し作った VBA にデジタル署名を登録します
ここではデジタル証明書の作成方法を説明します

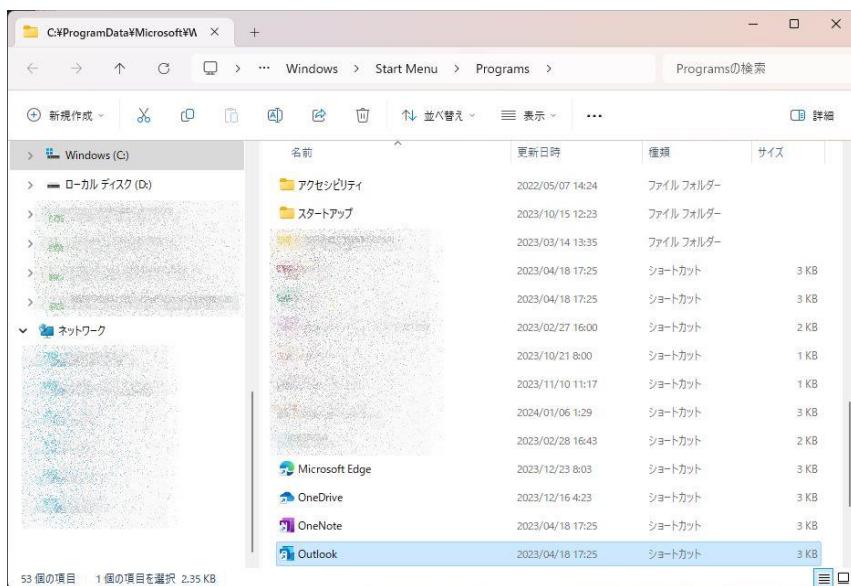
SELFCERT の実行



スタートメニューの「ピン留め済み」または「すべてのアプリ」を表示させ

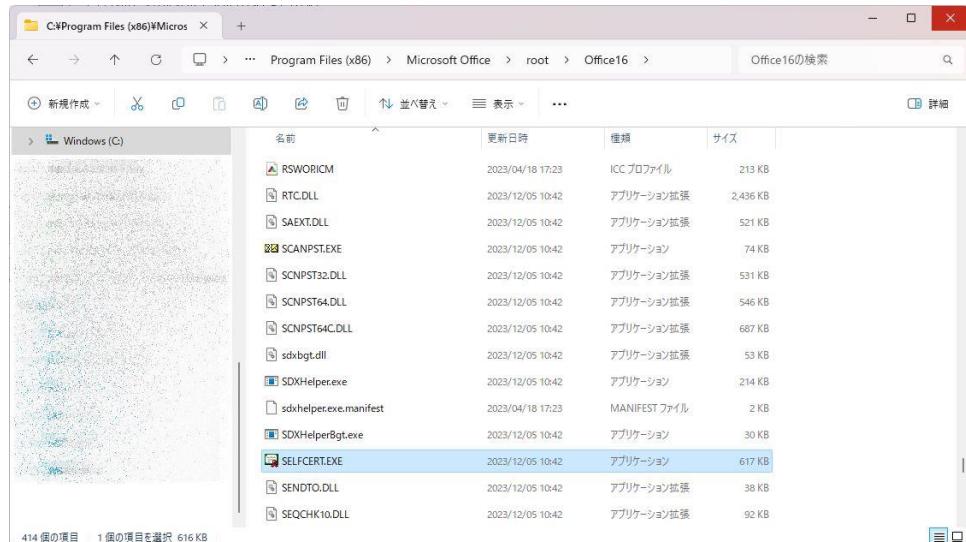
「Outlook」を右クリック

表示されたメニューの「詳細」(ピン留め済みの場合は出ません)から「ファイルの場所を開く」をクリック

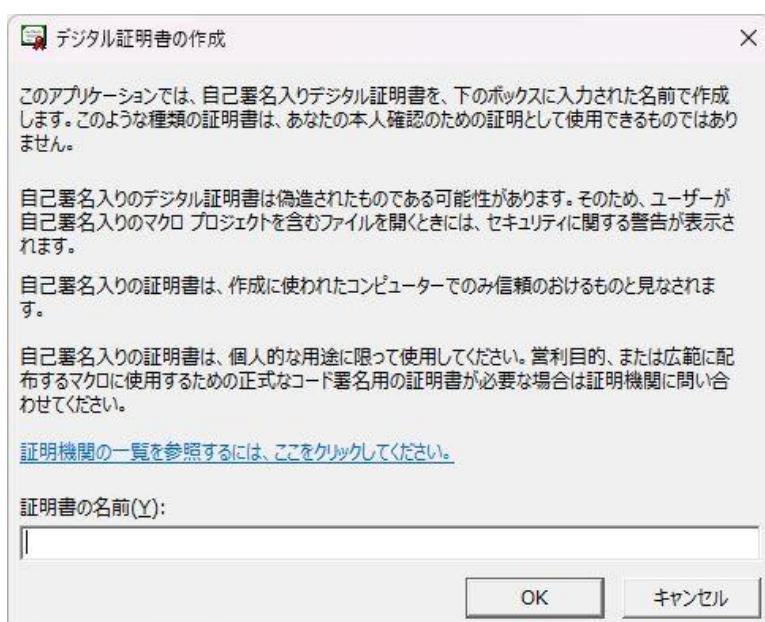


Outlook のショートカットアイコンが表示されたら、アイコンをキーボードの「Shift」キーを押しながら右クリック

表示されたメニューの「ファイルの場所を開く」をクリック



表示されたフォルダ内にある「SELFCERT.EXE」または「SELFCERT」をWクリックで起動します



デジタル証明書の作成画面で「証明書の名前」に好きな名前を入力して「OK」をクリック

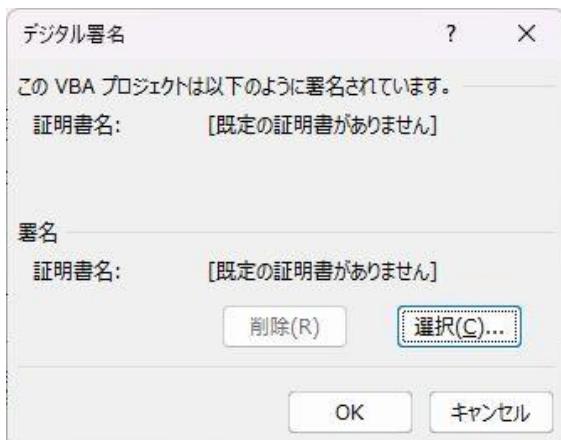
※自分しか使用しないので、何でもOKですが個人名にしておけばわかりやすいかもしれません

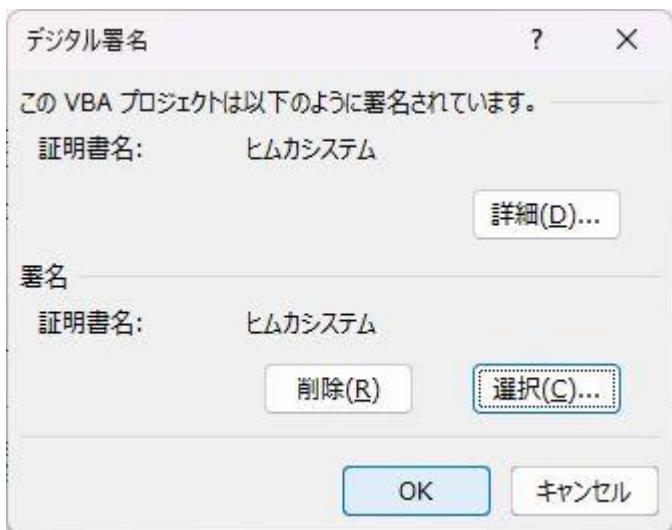
2. VBAへデジタル署名の登録

「VBA コードの追加」の要領で VBA の画面を表示させます

メニューの「ツール」から「デジタル署名」をクリックします

「選択」ボタンをクリックし、デジタル証明書の作成で作った証明書を選択します





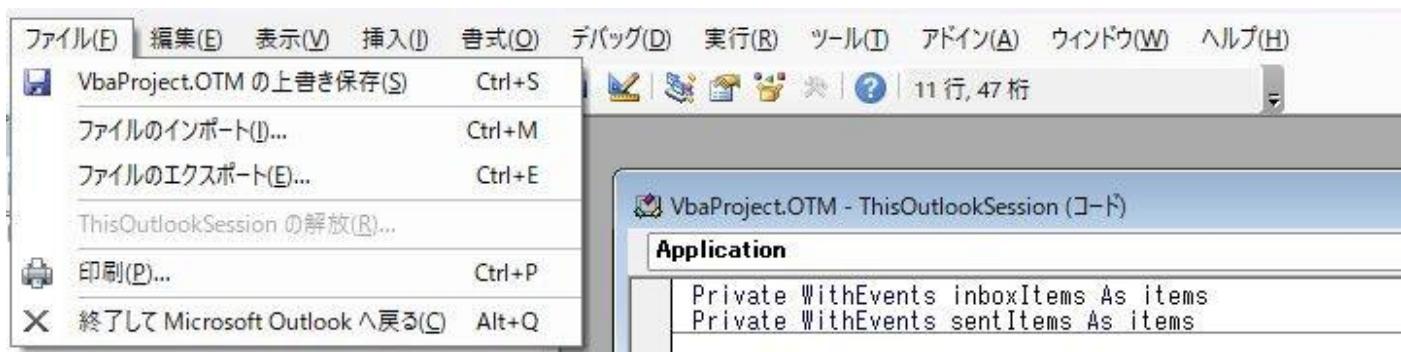
このように表示されれば「OK」で閉じます

証明書が表示されない場合は、証明書の作成がうまくできていないと思いますので、再度作成をしてください

3. VBA の手動保存

証明書の登録が完了したら、必ず行ってください

メニューの「ファイル」から「VbaProject.OTM の上書き保存」をクリックし、手動で保存します



保存できたら VBA の画面は閉じて OK です

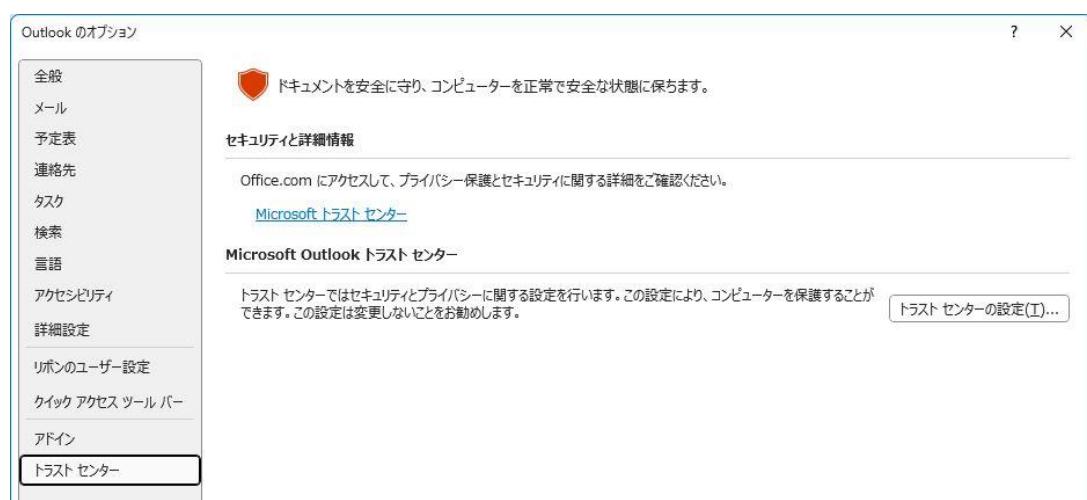
閉じるときに保存を聞いてきた場合、保存をしてください

Outlook のセキュリティ設定

デジタル証明書を登録したので、特に問題はないと思いますが念のためセキュリティ設定を確認・設定します

1. Outlook のメニューの「ファイル」をクリックし、「オプション」をクリックします

表示されたメニューの一番下の「トラストセンター」をクリックし、「トラストセンターの設定」をクリックします

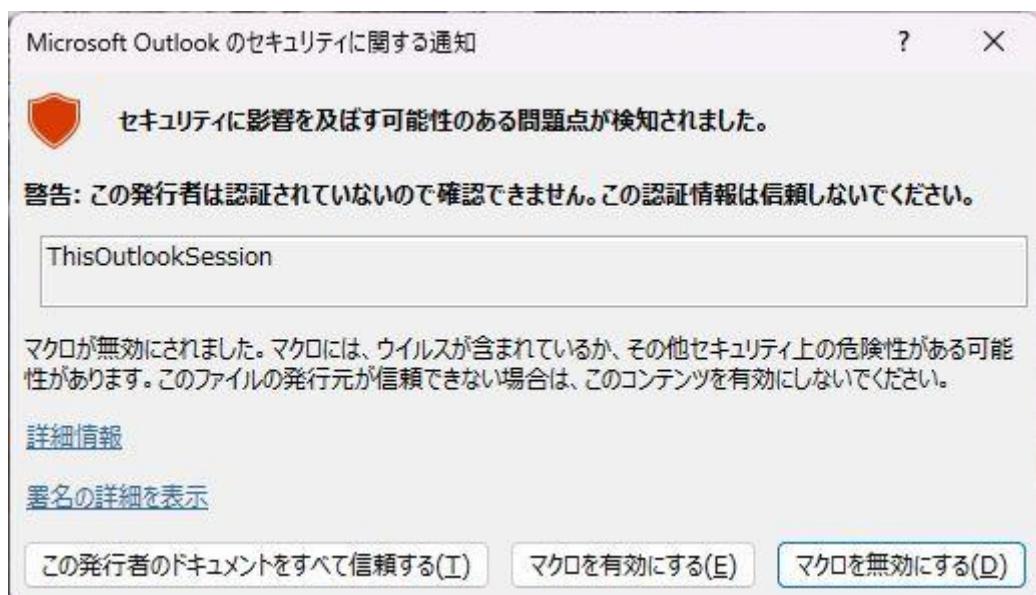


「マクロの設定」を選択し、上から2番目の「デジタル署名されたマクロに対して・・・云々」がマークされているのを確認し、「OK」で閉じます（違う項目が選択されていたら変更してください）



ここまでできたら一旦、Outlook を終了させます

2. 再度 Outlook を起動すると警告が表示されます



「ThisOutlookSession」と表示されているのを確認し、左の「この発行者のドキュメントをすべて信頼する」をクリックします

数秒の迷惑メールの処理後、Outlook が起動します

警告は「この発行者のドキュメントをすべて信頼する」を選択すれば最初の1回だけ聞いてきます

「マクロを有効にする」を間違ってクリックした場合は、一旦 Outlook を閉じて再度起動時に「この発行者のドキュメントをすべて信頼する」を選択してください

※本ツールを使用して問題が発生しても保証もサポートも出来かねますので、自己責任でご使用ください
うまく動作しない、設定の仕方がわからないなどの
問い合わせにも対応できませんのであしからずご了承ください

本ツールの内容

◆チェック・判断の条件

1. タイトルと本文が空であること（タイトルだけ無しにする方もいますので、両方が空であること）
2. Outlook 起動時と受信済み（1週間以内）のメールも対象にする
3. 複数のメールアカウントに対応
4. 受信フォルダ以下サブフォルダ（複数階層）もチェックする

◆処理内容

1. 該当したメールは「既読」にし、「削除済みアイテム」フォルダに移動する
2. 念のために受信フォルダの該当メールの削除処理をする
3. テスト段階で該当メールが送信フォルダに移動していたことがあったため、送信フォルダの該当メールも削除する

※即削除でもいいのですが、必要なメールだったということもあるかもしれないと思い、削除済みアイテムに移動させています

◆問題点

複数アカウントの場合、該当メールは1番目のアカウントの「削除済みアイテム」フォルダに移動しています
アカウント毎のフォルダに移動させようとしたが、うまくいかなかったのと別に問題はないと判断したので
そのままにしています

2024年1月5日作成

ヒムカシステム

<https://himuka-sys.com>