

Entourage AppleScript

このドキュメントに記載されている情報 (URL などのインターネット Web サイトに関する情報を含む) は、情報提供の目的で発行されているものであり、将来予告なしに変更することがあります。

別途マイクロソフトのライセンス契約上に明示の規定のない限り、このドキュメントはこれらの特許、商標、著作権、またはその他の無体財産権に関する権利をお客様に許諾するものではありません。

Microsoft、Excel、Entourage、PowerPoint は、米国 Microsoft Corporation およびその関連会社の登録商標または商標です。

Apple、Apple ロゴ、Mac、Mac OS は、米国 Apple Inc. の米国およびその他の国における登録商標または商標です。

© 2008 Microsoft Corporation. All rights reserved.

Entourage AppleScript

Entourage AppleScript の概要	4
メッセージを使用する	4
アドレス帳を使用する	25
カレンダーを使用する	37
メモについて	45
連絡先を Excel にエクスポートする.....	45

Entourage AppleScript の概要

Microsoft Entourage は、多彩な機能を備えた、Macintosh 向けの Microsoft Office アプリケーションですが、これまで Visual Basic for Applications (VBA) はサポートされていませんでした。そのため、このドキュメントでは、VBA マクロの変換ではなく、Entourage に実装されている AppleScript の活用方法について説明します。AppleScript を使いこなせるようになると、Macintosh 用に開発されたすべての AppleScript 対応のアプリケーションで応用することができます。

Entourage は、他の Office アプリケーションと多くの機能を共有しているため、Office AppleScript で習得したことは、Entourage でも応用できます。このドキュメントでは、Microsoft Entourage AppleScript Dictionary には記載されていない役立つテクニックや機能、記述の際に注すべき事項についても説明しています。

Entourage は、以前からスクリプティング可能なアプリケーションであったため、利用可能なスクリプトが多数存在しています。それら既存のスクリプトは、現在必要なタスクの処理にそのまま利用したり、似たようなタスクの処理に応用したり、問題解決に役立ったりと、さまざまな方法で活用することができます。

Entourage AppleScript Dictionary は、以前の Macintosh 用電子メール アプリケーションや予定表アプリケーションから継承した情報を基に構成されています。ほとんどの構文は、その他の Office アプリケーションよりもシンプルで、実際に英語のように読むことができますが、過去の電子メール アプリケーションから派生した (Microsoft Exchange メールに基づいたものではない) 古い情報も含まれています。また、最近 Dictionary に追加された Exchange 機能の説明では、役立つヒントもいくつか紹介されています。

メッセージを使用する

添付ファイル付きのメッセージを作成して送信する

ドキュメントを保存してファイル パスを取得する

Microsoft Word、Excel、または PowerPoint でドキュメントを作成し、そのドキュメントを電子メールの添付ファイルとして送信するのは簡単な作業です。しかし、一日に何度もこの作業を行う必要がある場合は、Microsoft Entourage を使用してスクリプトを記述しておけば、一連の作業を瞬時に実行できるため、大変便利です。

毎回異なる本文を使用する場合は、添付ファイル、受信者名、件名、あいさつ文だけを含めたスクリプトを記述することもできます。本文も毎回同じであれば、メッセージを準備するだけで、作業中のアプリケーションを切り替えることなく、バックグラウンドで送信できます。

どちらの場合も、最初のステップでは添付するファイルが保存されているかどうかを確認し、保存されていない場合は保存して、そのファイルパスを取得します。このステップは、添付ファイルの作成に使用したアプリケーションから行う必要があり、Word、Excel、または PowerPoint のいずれかによって、若干方法が異なります。

Entourage AppleScript

Word の場合は、次のようになります。

```
tell application "Microsoft Word"
    try
        set frontDoc to active document
        save frontDoc
        on error -- if you cancel out of Save dialog for an unsaved doc, avoid error
            return
        end try
        set filePath to full name of frontDoc
    end tell
```

Excel の場合は、次のようになります。

```
tell application "Microsoft Excel"
    try
        set frontDoc to active workbook
        on error -- avoid error if no workbook open
            return
        end try
        set containerPath to path of frontDoc
        if containerPath = "" then -- unsaved
            activate
            set docName to name of frontDoc
            set filePath to (choose file name default name docName) ~
                as Unicode text
            if filePath does not contain "." then set filePath ~
                to filePath & ".xls"
            save frontDoc in filePath
        else -- already saved as file, re-save
            save frontDoc
            set filePath to full name of frontDoc
        end if
    end tell
```

Entourage AppleScript

PowerPoint の場合は、次のようになります。

```
tell application "Microsoft PowerPoint"
try
    set frontDoc to active presentation
on error -- avoid error if no presentation open
    return
end try
set containerPath to path of frontDoc
if containerPath = "" then -- unsaved
    activate
    set docName to name of frontDoc
    set filePath to (choose file name default name docName) ~
        as Unicode text
    if filePath does not contain "." then set filePath ~
        to filePath & ".ppt"
    save frontDoc in filePath
else -- already saved as file, can't re-save (bug)
    beep
    display dialog "Due to a bug in PowerPoint, you need to ~
        make sure you " & "have saved the presentation first." ~
        & return & return & "If you have done so, ~
        click \"Proceed\"." buttons {"Cancel", "Proceed"} ~
        with icon 2
    set filePath to full name of frontDoc
end if
end tell
```

メモ 現在、PowerPoint では、上の例の Excel コードでも使用されている else セクション内に display dialog コードが必要です。

メッセージを作成する

添付するファイルの filePath をテキストとして取得したら、次に電子メールメッセージを作成します。事前に設定した受信者名と電子メール アドレス、CC 行、あいさつ文、件名、および既定の署名が入った新しいメッセージを Entourage で開くには、続けて次のように記述します。

```
tell application "Microsoft Entourage"
    activate
    set mySigType to default signature type of default mail account
    if mySigType is other then set mySig to default signature choice ~
        of default mail account
    set newMsg to make new draft window with properties ~
        {subject:"Today's File", to recipients:~
        "Ellen Adams <ellenadams@northwindtraders.com>, " & ~
        "Jun Cao <juncao@northwindtraders.com>", CC recipients:~
        "Toni Poe <tonipoe@northwindtraders.com>", content:~
        "Hi, Everyone" & return & return, signature type:~
        mySigType, attachment:alias filePath}
    if mySigType is other then set other signature choice ~
        of newMsg to mySig
end tell
```

Entourage AppleScript

このスクリプトでは、Entourage を activate して前面に表示し、新しい **draft window** を作成するときに既定の署名を使用できるように、既定のアカウントの既定の署名を抽出しています。Dictionary には、メール アカウント (**POP account**、**IMAP account**、**Hotmail account**、または **Exchange account**) の **default signature type** は、*random*、*none*、または *other* のいずれかであると説明されています。*other* は標準的な署名で、この場合は、実際の署名である **other signature choice** も抽出する必要があります。アカウントで *random* または *none* が使用されている場合は、**other signature choice** プロパティが存在しないため、下書きを新規に作成すると、後で署名を指定できません。または、スクリプトでエラーが発生します。

このトピックでは行を途中で区切るために行連結文字 (-) を使用しています。そのため、**to recipients** プロパティ (および **cc recipients** と **bcc recipients** プロパティ) は複数の受信者をコンマで区切ったひと続きの文字列であるにもかかわらず、それがわかりにくくなっています。実際にスクリプトを書くときは、行連結文字 (-) を使用する必要はありません。次のように **to recipients** プロパティを指定すると、Macintosh スクリプト エディタ で **make new draft window** コマンド全体に折り返しが指定されます。

```
to recipients:"Ellen Adams <ellenadams@northwindtraders.com>, Jun Cao  
<juncao@northwindtraders.com>"
```

受信者の表示名は省略できます。その場合、電子メールアドレスを山かっこ <> で囲む必要はありません。したがって、上の行を次のように記述することもできます。

```
to recipients:"ellenadams@northwindtraders.com, juncao@northwindtraders.com"
```

Dictionary で **draft window** クラスのエントリを確認すると、**attachment** プロパティが存在しないことがわかりますが、このコードは正しく動作します。これは開発者が単純化のために強制変換を適用しているためです。つまり、暗黙的なプロパティとして、**make new draft window** (または **draft news window**) または **outgoing message** コマンドの **with properties** パラメータ内に記述されています。これは、コードの冒頭でのみ指定することができ、それ以外の場所に記述しても機能しません。要素 **attachment** はプロパティではなく、実際は **draft window** および **message** クラスの要素です。存在しない (0 個) 場合もあれば、複数存在する場合もあります。

受信または送信したメッセージの添付ファイルをそのまま取得しようとしても、エラーが発生して取得できません。**attachments** または *every attachment of message* を取得してから、それぞれの必要な **file** プロパティを取得する必要があります。

下書きウインドウ (新しい **message window**) に戻り、次のように添付ファイルを後から追加することもできます。

```
set newMsg to make new draft window with properties {subject:"Today's File"}  
make new attachment at newMsg with properties {file:alias filePath}
```

しかし、多数のファイルを添付するのは、面倒な作業になります。**make new draft window** コマンドの **with properties** パラメータを使用し、**attachment** プロパティとして、リスト内の中かっこ { } 内にエイリアスのリストを指定することで、大量のファイルを一度に添付できます。

このコードでは、**account**、**bounds**、**encoding**、**priority** などの設定する必要のなかった多くの下書きウインドウのプロパティを省略しています。これらのプロパティには、指定する必要のない既定値が設定されますが、必要に応じて、任意のプロパティを変更できます。

メッセージのプロパティを設定する

メッセージを作成して送信する

ドキュメントを電子メールの添付ファイルとして毎日送信する必要がある場合、毎回メッセージを変更する必要がなければ、これらの電子メールをすべて自動的に送信するスクリプトを記述できます。このスクリプトでは、**outgoing message** を作成してメッセージを送信しますが、Microsoft Entourage を直接操作する必要がないため、他のアプリケーションで作業を続けながらメッセージを送信することができます。

Entourage AppleScript Dictionary で **recipient** を調べると、**message** (およびそのサブクラスである **outgoing message** と **incoming message**) の要素であり、プロパティではないことがわかります。そのため、*to recipient*、*cc recipient*、*bcc recipient* など種類を問わず複数の受信者を指定することができます。

受信または送信したメッセージの受信者を取得するには、*every recipient of theMsg* を使用します。次に受信者のプロパティを調べて、**email address** プロパティと表示名を取得します。

通常、オブジェクトに要素を作成するプロセスでは **at** を使用しますが、新しい送信メッセージを作成する場合は、メッセージを作成した直後に各受信者を設定することはできません。これは、**draft window** と異なり、新しく作成された送信メッセージはインターネット用にエンコードされるため、その内部構造を変更できなくなるためです。ただし、前のトピックで説明した暗黙的なプロパティ **attachment** と同様に、開発者によって実装された強制変換によって "暗黙的な" 受信者プロパティを利用することができます。

この記述について説明したドキュメントはありませんが、これ以外に受信者を指定する方法はありません。前のトピックでも説明したように、これは最初の *make new outgoing message with properties* の呼び出し時にのみ使用できます。

Dictionary の **recipient** クラスの下に列挙されている、**address** と **recipient type** プロパティは両方とも使用できます。**address** プロパティの型は **address** クラスです。このクラスには、異なる **address** プロパティ (テキスト プロパティで、実際の電子メール アドレス) があります。さらに、**display name** テキスト プロパティがあります。

to recipient の 1 種類の宛先だけにメールを送信する場合は、*to recipient* が **recipient type** プロパティの既定の種類であることから、多くの指定項目を省略できます。複雑な受信者レコード構造の代わりに、*Display Name <eaddress@domain>* のように電子メールアドレスを山かっこ <> で囲む略記を使用することもできます。次のスクリプトで、先に説明したプロパティの強制変換が行われます。

```
tell application "Microsoft Entourage"
  launch -- in background
  set theMsg to make new outgoing message with properties ~
    {subject:"Today's File", content:~
      "Please find today's .pdf file attached.", recipient:~
      "Phyllis Harris <phyllisharris@northwindtraders.com>",~
      attachment:alias filePath}
  send theMsg
end tell
```

Entourage AppleScript

最初の行 `launch` に注目してください。Entourage がバックグラウンドで既に開かれている場合、この行では何も実行されません。Entourage が起動していない場合は、前面に表示されずに起動します。この行を省略すると前面に表示されます。これは、AppleScript 言語の一部であり、Entourage に特化したものではないので、すべてのアプリケーションで同様に動作します。

次に、**recipient** のプロパティ値が単純であることに注目してください。このスクリプトではメッセージの作成先が指定されていないため、メッセージは既定の場所であるローカルの [下書き] フォルダに既定のアカウントを使用して作成されます。必要に応じて、これらのプロパティをすべて指定できます。

attachment プロパティで指定されるファイル (`alias filePath`) は、前面の保存ドキュメントのパスを変数 `filePath` として格納するために、Microsoft Word、Excel、PowerPoint のうちのどのアプリケーションでこのコードを指定したかによって変わります。

最後の行でメッセージを送信します。

受信者の完全な構文では、"暗黙的な" **recipient** プロパティに、リストの中かっこ `{ }` を使用してすべての受信者リストを設定することで、`to recipient`、`cc recipient`、`bcc recipient` のどの種類の受信者でも必要な数だけ指定できるようになっています。各受信者には、同じ文字であるレコードの中かっこも使用する必要があります。これは、各受信者が **address** プロパティと **recipient type** プロパティを含む 1 つのレコードであるためです。さらに、各アドレスも、**address** プロパティと **display name** プロパティを含む 1 つのレコードです。そのため、次に示すように、多くの中かっこを使用します。

```
tell application "Microsoft Entourage"
  launch
  set theMsg to make new outgoing message with properties ~
    {subject:"Today's File", content:-
      "Please find today's .pdf file attached.", recipient:-
        {{address:{address:"phyllisharris@northwindtrader.com", display name:-
          "Phyllis Harris"}, recipient type:to recipient}, {address:-
          {address:"ellenadams@northwindtrader.com", display name:-
            "Ellen Adams"}, recipient type:cc recipient}, ~
          {address:{address:"juncao@northwindtrader.com", display name:-
            "Jun Cao"}, recipient type:-
              bcc recipient}}, attachment:alias filePath}
  send theMsg
end tell
```

ここでは、種類ごとに受信者を 1 人だけ指定していますが、実際は、必要な数の受信者をリストに含めることができます。種類が `to recipient` の場合は、**recipient type** を省略できます。また、表示名も省略できます。スクリプトを記述する際、オンラインソースをコピーしてペーストする場合は、行末エラーが発生しないように、行連結記号による行の区切りを無視してください。スクリプト エディタでは、必要に応じて長い行が折り返されます。

先の説明のように、**attachment** プロパティで指定される `filePath` 変数は、前面の保存ドキュメントのパスを変数 `filePath` として格納するために、Word、Excel、PowerPoint のうちのどのアプリケーションでこのコードを指定したかによって変わります。

Entourage AppleScript

他の Microsoft Office アプリケーションと Entourage のコードを含むスクリプト全体を、システムの [スクリプト] メニューに保存し、その日作業している前面のドキュメントで選択するようにします。メッセージは、作業中のアプリケーションにおける操作や中断を伴うことなく、受信者全員に送信されます。必要であれば、他の Office アプリケーションのブロックにコード行を追加し、アクティブなドキュメントを保存して閉じることもできます。

このスクリプトまたはメッセージを送信するその他のスクリプトを初めて実行するときに、`an external application is trying to send mail` という警告が表示されます。この場合は、クリックして許可します。また、Entourage のスクリプトを何度も使用する予定があれば、以降、警告を表示しないように選択します。UI を使用して Entourage の [セキュリティ設定] を開き、外部アプリケーションがメールを送信しようとした場合と、アドレス帳にアクセスしようとした場合の警告をオフにすることもできます。Entourage では、スクリプトを "外部アプリケーション" と同様のものと見なすため、スクリプトを実行しようとしたときにこのような警告が表示されます。

他のプロパティを使用してメッセージを作成して送信する

新しいメッセージを作成する方法を学習するにあたり、ここまでは受信者に関係することを中心に説明してきました。これは、受信者に関連する部分が難しいためです。スクリプトでは、他にも数多くのプロパティを設定できます。UI で操作できることはすべて設定でき、さらに詳細に設定することもできます。

たとえば、自分の既定のアカウントではない別の電子メール アカウントを指定する必要があるとします。

汎用の `account` クラスがないため、アカウントの種類 (**POP account**、**IMAP account**、**Hotmail account**、**Exchange account** のいずれか) を指定する必要があります。Entourage はデータベースに基づいたアプリケーションであるため、内部のすべての要素には、固有の **ID** が割り当てられています。要素 (アイテム) への参照は、すべて **ID** 参照に解決されます。これは、アカウントであっても同様です (`POP account id 1`、`IMAP account id 2` など)。

しかし、通常、ID はわかりません。Entourage のほとんどすべての要素は、**name**、**index** (知っている場合)、**ID**、または `whose` フィルタで参照できます。アカウントの名前に電子メール アドレスを意図的に設定している場合もありますが、その場合を除いて、アカウント名は電子メール アドレスではなく、[ツール] メニューの [アカウント] をクリックし、[メール] タブで指定した名前です。このため、通常は `POP account My Account` のように名前で参照します。My Account の部分がユーザーが付けた名前を示します。

アカウントで使用する既定の署名がメッセージに自動的に表示されるので、メッセージの内容に署名を追加しないでください。追加すると署名が 2 つ表示されます。このメッセージの署名を変更する必要がある場合は、初めにアカウントの **default signature type** と **default signature choice** を変更するか、`signature type` に `none` を指定してメッセージの内容に必要な文字列を追加します。メッセージを作成した後には設定を元に戻します。

汎用のサブクラスである **outgoing message** は、**message** クラスに含まれます。これは、特定のメッセージを送信した後も、そのメッセージにのみ適用されるいくつかの読み取り専用のプロパティ (**forwarded**、**redirected**、**replied to**、**resent** など) を継承します。実際に必要であれば、開始時に読み取り専用のプロパティのほとんどを設定することができますが、新しいメッセージに対してはこれらのプロパティを設定しないでください。

通常、**connection action** プロパティは受信 POP メッセージにのみ適用されます。このプロパティも、新しいメッセージには設定しないでください。**edited** プロパティは、完全に読み取り専用で、スクリプトでは設定できません。

color プロパティの使用は非推奨なので設定しないでください。このプロパティは **category** に置き換えられ、各カテゴリには設定可能な **color** プロパティが含まれています。ここで使用している **category** プロパティは、カテゴリ {category "Work", category "Personal"} のリストになっています。これは、メッセージなどのオブジェクトに複数のカテゴリが存在する可能性があるためです。リストの最初のカテゴリがプライマリ カテゴリとなり、色を設定します。また、カテゴリのない空のリスト {} になることもあります。None というカテゴリはありません。既定値は {} で、設定する必要はありません。

通常は **source** を設定することはありませんが、メッセージを複製して別のメッセージを作成し、多少の変更を加えるような場合にはこれを設定します。**source** プロパティには、メッセージのテキスト、各部分、添付ファイル、ヘッダー全体が構造化されて含まれます。通常、ヘッダーは自動的に作成されるため、スクリプトで記述する必要はありません。

この他のプロパティはすべて設定可能です。**has html** プロパティの既定値は *false* ですが、開始時に *true* に設定できます。ただし、このプロパティは、既定の HTML フォントとサイズを使用することを指定するだけです。Entourage では、書式付きテキストをスクリプトで処理する方法がなく、**has html** プロパティを *true* に設定すると、メッセージの内容から改行記号が削除されるため、この設定は使用しないでください。Dictionary で "読み取り/書き込み" と明記されているプロパティを使用してください。

メッセージの作成場所を指定しない場合、メッセージは [下書き] フォルダに作成されます。他によく指定される場所としては [送信トレイ] があります。典型的な例を以下に示します。

```
tell application "Microsoft Entourage"
    -- add carriage return before signature used for the account
    set theBody to "Some long text here. As long as you want." ~
        & return & return
    set theMsg to make new outgoing message at outbox folder ~
        with properties {account:POP account "My Account", ~
            subject:"My New Message", content: theBody, ~
            recipient:"Phyllis Harris <phyllisharris@northwindtrader.com", ~
            flagged:true, priority:high, category:{category "Work"}}
end tell
```

受信メッセージを使用する

受信メッセージは他のユーザーから受け取るものであり、新しい **incoming message** を作成するというと疑問に思うかもしれません。そもそも、受信メッセージを新しく作成することはできないと思っている場合もあるでしょう。これは、実際には Microsoft Entourage にメッセージをインポートするための方法です (**import** コマンドは、メッセージ フォルダとなる mbox ファイル形式に対してのみ有効です)。

.eml メッセージ ファイルの内容を読み取るには、「vCard を添付する」で説明している .vcf ファイルの読み取りに似た方法を使用できます。読み取った内容は、**source** プロパティとして使用して、任意のフォルダに新しい受信メッセージを作成することができます。必須のプロパティはこれだけです。または、この内容を使用して、**duplicate** コマンドを利用できない場所でも、既存メッセージのプロパティからメッセージの複製を作成することができます。

Entourage AppleScript

このことが AppleScript では実行できて、UI では実行できないことに注目してください。ほとんどの "読み取り専用" プロパティは作成時に設定できるので、AppleScript を利用すると、必要なことはほとんどすべて実行できます。**outgoing message** を作成する場合と似たような手順で受信メッセージを作成し、確認してみてください。

メッセージ ルールを作成する

Microsoft Entourage では、受信したメッセージに対してスクリプトを実行することができます。Entourage の [ルール] コマンド ([ツール] メニュー) では、ルールを有効にして、受信トレイに入ってくるメッセージに対し自動的にルールを実行することができます。[AppleScript を実行] は、ルールで実行できるアクションの 1 つです。同様に、[ツール] メニューの [スケジュール] コマンドでは、定期的に、または、起動時、終了時のうちいずれかのタイミングでタスクを実行するように設定することができ、ここでも [AppleScript を実行] アクションを指定することができます。AppleScript では、Entourage のオブジェクトに対してほぼどのような操作でも実行でき、自由に機能を選択したり組み合わせることができるため、標準のルールやスケジュールよりも詳細に動作を指定できます。したがって、ルールの場合もスケジュールの場合も高度なカスタマイズと自動化を実現できます。

ルールで指定できるフィルタは、[すべての条件が一致した場合] (ブール型 *and* ですべての条件を結合) または [条件のいずれかが一致した場合] (ブール型の *or* ですべての条件を結合) の 2 つしかなく、不満に感じることがあるかもしれません。しかし、これ以上詳細で複雑なフィルタは実行できません。AppleScript ではどのような複雑な条件でも処理できるので、まずスクリプトとして記述して、ルールからスクリプトを実行してください。

メッセージを選択する

このトピックでは、Microsoft Entourage Dictionary で説明されていない、スクリプトの実行対象となるメッセージを指定したり、ルールを適用したいメッセージを指定する方法について説明します。このような場合は、**current messages** アプリケーション プロパティ (メッセージのリスト) を使用します。

ルール (またはスケジュール) の実行対象となるメッセージを指定するには、メッセージリストの *item 1* を取得する必要があります。メッセージ リストのメッセージを 1 つだけ選択した場合も同様です。ただし、次のように記述することはできません。

```
tell application "Microsoft Entourage"
    set theMsg to item 1 of current messages
end tell
```

これを実行すると、"Microsoft Entourage got an error: Can't get item 1 of current messages." というエラーが表示されます。

Entourage AppleScript

これは、**current messages** プロパティが明示的な **get** コマンドを必要とする数少ないプロパティの1つであるためです。**get** コマンドを使用しない場合は、別の行でプロパティに対して変数を設定する必要があります。これによって、**item 1** を選択するための参照先 (特定のメッセージ リスト) が評価されます。

```
set theMsg to item 1 of (get current messages)
--> incoming message id 120434
```

現在のメッセージの **item 1** が必要な場合は、このように記述します。

application の別のプロパティである **selection** でも、同様に **get** コマンドを明示的に指定するか、変数を設定する必要があります。

ステートメント **get selection** は、UI で選択されている項目をほぼすべて返します。選択項目に応じて、クラスは **list**、**folder**、**text** のいずれかになります。**contacts**、**groups**、**events**、**tasks**、**notes**、**messages** などの選択したオブジェクトは常にリストとして返されます。選択オブジェクトが1つの場合でも、オブジェクトは1つの項目を含むリストとして返されます。唯一の例外は、フォルダの一覧で選択した **folder** または **custom view** の場合で、これらは **folder** オブジェクトとして返されます (一度に選択できるのは1つだけです)。メモのセクションと連絡先、イベント、タスク ウィンドウのすべてのテキスト フィールドなど、メッセージやメモで選択したテキストは、Unicode テキストとして返されます。

メモ 予定表で選択したイベントが1つの場合でも、選択イベントに対応する1つの項目を含むリストが返されます。ただし、一部の細かいオブジェクト (日付選択用のミニ カレンダー、[プリント] ダイアログ ボックスのボタンなど、主にポップアップ オブジェクト) のように Entourage で対応する AppleScript クラスがない場合は、オブジェクトを選択して、**get selection** ステートメントを実行しようとする、スクリプトが機能せずにエラーを返します。まだ開いていない新しい予定表イベントのプレースホルダ、開いているイベント、何も選択されていない状態の **contact window** など、"空白" の選択項目については、ほとんどの場合、空のリスト ({}) または空のテキスト ("") が返されます。

常に選択項目のクラスを確認して、目的のものを取得しているかどうか、操作できるかどうかを確認する必要があります。たとえば、フォルダを取得する場合、選択項目の **item 1** を取得できません。また、テキストを取得する場合、**item 1** は先頭文字になります。フォルダにあるメッセージではなく、フォルダを選択していることを見落とす間違いがよくあります。このため、メッセージの場合は **selection** プロパティの代わりに **current messages** プロパティを使用することをお勧めします。UI でメッセージではなくフォルダが選択されている場合でも、このプロパティは有効です。したがって、通常は選択項目に変数を設定して、そのクラスを確認し、最後に選択項目に対して動作を実行する方法が最適です。

```
tell application "Microsoft Entourage"
    set theSelection to selection
    if class of theSelection is list then
        set theObject to item 1 of theSelection
        -- will then check class of theObject for contact, etc.
    else
        beep
        display dialog "First make sure you have selected one or more " & ~
            "contacts in the Address Book." buttons {"Cancel"} with icon 2
    end if
    --rest of script
end tell
```

[スクリプト] メニューについて

Microsoft Entourage には、Mac OS 付属のスクリプト作成アプリケーション (Script Editor など) が提供される前から使用可能な独自の [スクリプト] メニューがありました。OS 付属のアプリケーションは、もちろん Entourage でも機能しますが、この独自の [スクリプト] メニューは、現在も、古くからの Entourage ユーザーによって支持されています。Entourage の [スクリプト] メニューには、次の 2 つの利点があります。

- このメニューから実行するスクリプトは、OS 付属のアプリケーションから実行するスクリプトより速く動作します。ただし Mac OS X バージョン 10.4 (Tiger) 以降はアプリケーションへの呼び出しが高速になったことで、この差は目立たなくなりました。
- ショートカット キーをスクリプトに追加することができます。ショートカット キーの詳細については、Entourage 2004 ヘルプの「スクリプト メニューについて」を参照してください。

ショートカット キーを使用すると、スクリプトを繰り返し何度も実行できるので、大きなメリットとなります。

ただし、Entourage の [スクリプト] メニューでは、スクリプトの実行中にプロパティが値を保持する必要がある場合、問題が起こることがあります。

Macintosh スクリプト エディタ を使用して作成されたスクリプトは、現在データ フォークのみのファイルとなっています。Entourage 2004 の [スクリプト] メニューで、これらのスクリプトを実行することはできますが、スクリプトの終了時にプロパティ値が変わっても保持されません。以前のリソース フォークのスクリプトは、Mac OS X バージョン 10.3 (Panther) と共にリリースされた スクリプト エディタ 2 から採用されなくなりました。しかし、Microsoft Office 2004 の発売時には、依然として Mac OS X バージョン 10.2 (Jaguar) が広く使用されていました。そのため、Office 2004 では、新しい Mac OS X バージョン 10.3 (Panther) の API をスクリプト メニューに適用できませんでした。

したがって、スクリプトによって値が変わる可能性のあるスクリプト プロパティを使用するスクリプトを作成する場合は、次の 3 つの選択肢があります。

1. Microsoft Word や Excel のスクリプトの場合と同様に、システムの [スクリプト] メニューから実行し、ショートカット キーを使用しないようにします。
2. スクリプト エディタによっては、リソース フォーク形式でスクリプトを保存できるものもあります。
3. UNIX シェル スクリプトを使用してデータ フォーク スクリプトをリソース フォーク スクリプトに変換するスクリプトを探します。これにスクリプトをドロップすると変換されます。

外部で実行するアプリケーションとしてスクリプトを保存する場合は、これらの問題は関係ありません。

Entourage の [スクリプト] メニューは [ヘルプ] メニューの横のメニュー バーの左側にあります。このメニューからスクリプトを実行するには、"/ユーザー/ユーザー名/書類/Microsoft ユーザー データ/Entourage Script Menu Items" フォルダにスクリプトを保存します。これにより、スクリプトは Entourage の [スクリプト] メニューの項目として表示されます。

返信領域を上または下に変更する

ここでは、メッセージごとに返信と署名の位置を上から下、または下から上に変更するスクリプトを紹介します。他のユーザーと電子メールをやり取りするとき、返信を上に入力するユーザーもいれば、下に入力するユーザーもいます。このスクリプトは、返信が上にあたり下にあたりすることでメッセージ間のつながりがわかりにくくなるという事態を解消するのに役立ちます。Microsoft Entourage の [環境設定] で必要な返信方法を選択しますが、ここで設定できる方法とは別の方法で返信する場合にこのスクリプトを実行できます。

最初のセットアップで、スクリプトの先頭にプロパティを設定します。1つのバージョンのスクリプトでは、これらのプロパティを別の PREFS スクリプトを実行して設定します。この場合、その PREFS スクリプトで、スクリプティング機能追加の標準コマンドである **load script** と **store script** を使用して、メインスクリプトにあるプロパティの値を変更します。

この方法は、Macintosh スクリプト エディタ で作成されたスクリプトを Entourage 2004 で使用する場合には機能しません。データ フォーク スクリプトとリソース フォーク スクリプトの詳細については、「[スクリプト]メニューについて」を参照してください。

さらに重要なことは、スクリプトを作成する際に スクリプト エディタ でプロパティを思いどおりに設定することができ、こうして設定したプロパティが正常に機能することです。プロパティを変更する場合は、スクリプト エディタ でスクリプトを開き、変更して保存するだけです。

スクリプト名にショートカット キーを追加し、それを Entourage の [スクリプト] メニューに表示することができます。このスクリプトは特にショートカット キーを利用すると便利です。スクリプトの先頭で指定する 4 つのプロパティは次のとおりです。

```
property replyUsingMessageFormat : true
property htmlText : true
property attributionStyle : "short headers" -- or "internet" or "none"
property insertionPointOnTop : false --(only applies to internet-style)
```

最初の **replyUsingMessageFormat** と **htmlText** の 2 つのプロパティでは、元のメッセージと同じ形式 (テキストまたは HTML) で返信するか、または常にこのうちのどちらの形式で返信するかを指定します。これは Entourage の [環境設定] の [作成] の設定と同じ内容にします。AppleScript はこれらの設定にアクセスできないので、ここで設定する必要があります。

replyUsingMessageFormat を *true* に設定する場合は、**htmlText** も *true* に設定します。

replyUsingMessageFormat を *false* に設定する場合は、**htmlText** を必要な方に設定します (常にテキスト形式にする場合は *false* に設定します)。Entourage の [環境設定] の [返信と転送] で、通常の追加テキストのスタイルとして [メッセージに返信文を挿入し、返信文の前に次のテキストを挿入する] (返信を下に配置) を指定している場合、スクリプトで逆の [メッセージに返信文を挿入し、返信文の前に差出人、日付、宛先、および件名を表示する] (返信も署名も上に配置) を指定するには、**attributionStyle** を *short headers* に設定します。

Entourage AppleScript

また、通常の追加テキストのスタイルとして [メッセージに返信文を挿入し、返信文の前に差出人、日付、宛先、および件名を表示する] を指定している場合、スクリプトで逆の処理を指定するには、

attributionStyle プロパティを *internet* に設定します。また、スクリプトの動作として必要であれば *none* に設定することもできます。attributionStyle を *internet* に設定する場合は、**insertionPointOnTop** を *true* に設定するかどうかも選択できます。これは返信を上配置することを意味しており、これで *internet* 追加テキストの指定と返信を上配置するという指定を両方満たすことができます。しかし、署名は一番下になるので、これを望まない場合は使用できません。または **insertionPointOnTop** プロパティを *false* に設定することもできます (この場合、返信を下に配置することになります)。2つのダッシュ (--) の後のテキストは、通常はコメントでコードではありません。

以下はコンパイルしたときのスクリプトです。プロパティ値は既定のものが選択されています。

```
property replyUsingMessageFormat : true
property htmlText : true
property attributionStyle : "short headers"
-- or "internet" or "none"
property insertionPointOnTop : false
-- (only applies to internet-style attribution)

tell application "Microsoft Entourage"
    try
        if window 1 = main window then
            set theMsg to item 1 of (get current messages)

        else if class of window 1 = message window then
            set theMsg to displayed message of window 1
        else
            error number -128
        end if
        if class of theMsg = incoming message then error number -128
    on error
        beep
        display dialog "You must have a received message in the " & ¬
            "front or selected in the message pane of the main " & ¬
            "window for the Reply script to work." buttons {"OK"} ¬
            default button "OK" with icon 0
    return
end try

if replyUsingMessageFormat then
    if theMsg's has html then
        set htmlText to true
    else
        set htmlText to false
    end if
end if
```

Entourage AppleScript

```
if attributionStyle = "short headers" then
    reply to theMsg attribution style short attribution ↵
        html text htmlText
else if attributionStyle = "internet" then
    reply to theMsg attribution style internet attribution ↵
        place insertion point on top insertionPointOnTop ↵
        html text htmlText
else -- no attribution
    reply to theMsg attribution style no attribution ↵
        html text htmlText
end if
end tell
```

スクリプトの最初の部分では、メイン ウィンドウのプレビュー ウィンドウ枠に表示されているメッセージに返信しているかどうかを確認します。これに該当する場合、メッセージを次のように設定します。

```
item 1 of (get current messages)
```

プレビュー ウィンドウ枠に表示されているメッセージは、メッセージ一覧で選択されているメッセージ、つまり **current messages** の先頭 (かつ唯一) のメッセージであるため、このようになります。返信対象のメッセージがメイン ウィンドウのプレビュー ウィンドウ枠に表示されていない場合、スクリプトでは前面のウィンドウのクラス (種類) が **message window** であるかどうかを確認します。これは、新しいメッセージの下書きウィンドウではなく、保存されたメッセージを表示するウィンドウです。これに該当する場合は、ウィンドウの **displayed message** を取得します。

次に、受信したメッセージを選択したか開いたことを確認するため、**incoming message** クラスかどうかを確認します。前面のウィンドウが **main window** でもメッセージ ウィンドウでもない場合、またはメッセージのクラスが **incoming message** でない場合、スクリプトはエラー番号 -128 を呼び出します。これはキャンセルと同じで、呼び出しても安全なエラーです。このエラーは、`on error` ブロックでキャッチします。その場合、警告のダイアログ ボックスを表示して、[OK] がクリックされたときにスクリプトを終了 (**return**) します。適切な `else` ステートメントと `if` ステートメントの中でダイアログ ボックスを表示して終了するのではなく、このようなエラーを呼び出す理由は、ウィンドウがまったく開いていない場合に発生する本当のエラーをキャッチするためです。またこうすることで、条件にかかわらずダイアログ ボックスを 1 回呼び出すだけで済みます。

エラーにならなければ、スクリプトは次の `if` ブロックに進み、環境設定で `replyUsingMessageFormat` を選択している場合に HTML メッセージに返信しようとしているかどうかを確認します。これに該当する場合、同じように `htmlText` プロパティを `true` に設定します。最後に、`reply to` コマンドを呼び出します。ここでは環境設定に応じて `attributionStyle` と `htmlText` の適切なパラメータを指定 (プロパティを設定) します。

`reply to` コマンドにはオプションで **opening window** というパラメータがあります。これは既定で `true` に設定されるため、指定する必要はありません。したがって、新しい **draft window** がすぐに開き、受信者、引用されたテキスト、追加テキストが挿入された状態で表示されます (スクリプトによって準備済みの返信テキストを設定し、メッセージを自動的にバックグラウンド送信する場合は、**opening window** パラメータを `false` に設定します)。

Entourage AppleScript

スクリプトを [Entourage Script Menu Items] フォルダにスクリプト形式 (.sct) で保存し、名前を "Reply Insertion Reversed \cmR.sct" として、ショートカット キー **control+⌘+R** を適用します。ショートカット キーを追加する際には、**⌘**、**option**、**control**、**shift** のうち 1 つ以上の修飾キーが必要です。既に Entourage や他のスクリプト、または Mac OS で使用されているショートカット キーは避けてください。修飾キーには **control** を使用することをお勧めします。これは、Entourage に組み込みのショートカット キーではまず使用されていないキーであるためです。

同じ内容を実行して、全員に返信することもできます。先に示したスクリプトをコピーして新しいスクリプト ウィンドウにペーストし、最後の方の **reply to** の各行に `with reply to all` を追加します。

```
if attributionStyle = "short headers" then
    reply to theMsg attribution style short attribution ↵
        html text htmlText with reply to all
else if attributionStyle = "internet" then
    reply to theMsg attribution style internet attribution ↵
        place insertion point on top insertionPointOnTop ↵
        html text htmlText with reply to all
else -- no attribution
    reply to theMsg attribution style no attribution ↵
        html text htmlText with reply to all
end if
```

これを "Reply All Reversed \scmR.sct" として保存し、**shift+control+⌘+R** のショートカット キーを適用します。

イベントなしでスクリプトを実行する

Microsoft Office では、アプリケーション内でイベント発生時に自動的にスクリプトが実行されることはありません。

そのため、ボタンのクリックやメニュー項目選択などのイベントに "割り込む" こともできません。これは、セキュリティ上、悪意のある妨害を防ぐため、アプリケーション開発時に意図的にそのように設計されています。したがって、[今すぐ送信] ボタンをクリックした際に自動的にスクリプトを実行することはできません。いわゆる "送信ルール" は、実際には "送信後ルール" です。スクリプトは送信後のメッセージに対してのみ実行されるので、その時点では変更は間に合わないことになります。

しかし、ほとんどの場合に通用するテクニックがあります。それは、繰り返しの **schedule** で 1 分おきにスクリプトを実行する方法です。このスクリプトではさまざまな項目を確認できますが、特に新しい **draft window** (新しいメッセージ ウィンドウ) が前面に表示されているかどうかを確認できます。前面に表示されていない場合はスクリプトでは何も実行されませんが、表示されている場合は、スクリプトで必要な操作を実行して他の条件を確認できます。メッセージの作成には 1 分以上かかるのが普通であり、次のスケジュールの実行までの実際の平均は 30 秒であるため、メッセージの作成が終わる前にキャッチして必要事項を追加できます。次のトピックのスクリプトは、このことを利用しています。詳細については、「vCard を添付する」を参照してください。

Entourage AppleScript

メモ このスクリプト オプションは慎重に使用し、実行するタイミングに注意してください。メッセージが短い場合はすぐに作成できるので、スケジューラの同期が間に合わない可能性があります。この方法を使用する場合は、送信前にメッセージを確認し、スクリプトを実行するタイミングを確保できるようにしてください。

vCard を添付する

Microsoft Entourage には、vCard を送信メッセージに添付するための設定がありません。毎回アドレス帳からメッセージに **me contact** (自分の連絡先) をドラッグすることもできますが、次のスクリプトを利用すると、この作業を簡単に実行することができます。このスクリプトでは AppleScript の読み取り/書き込みコマンドと Finder スクリプトをいくつか使用して vCard を更新した後、その vCard を添付します。すべてのメッセージに vCard を添付して送信する場合は、1 分ごとの **schedule** でスクリプトを繰り返し実行するように設定してください。

```
tell application "Microsoft Entourage"
    set MUDpath to (path to MUD) as Unicode text
    set meContact to me contact
    set vcInfo to vcard data of meContact
    set myname to name of meContact
end tell

set pathName to (MUDpath & myname & ".vcf") as Unicode text

try
    set f to open for access file pathName with write permission
    --will create a new one or update every time
    set eof f to 0
    write vcInfo to f
    close access f

    tell application "Finder"
        set creator type of file pathName to "OPIM"
        set file type of file pathName to "vCrd"
    end tell

    set theVcard to alias pathName

on error errMsg number errNum
    try
        close access f
    end try
    beep 2
    return
end try
```

Entourage AppleScript

```
tell application "Microsoft Entourage"
  try
    if class of window 1 ≠ draft window then error number -128
  on error
    return
  end try
  set newMsg to window 1

  set theAttachments to every attachment of newMsg
  repeat with theAttachment in theAttachments
    -- only one, don't multiply vCards
    if name of theAttachment = (myname & ".vcf") then return
  end repeat

  make new attachment at newMsg with properties {file:theVcard}

end tell
```

スクリプトの最初の部分では、**path to MUD** を指定しています。これは [Microsoft ユーザー データ] フォルダへのパスで、ここに vCard を保存します。UI で vCard を作成する場合は意識することはありませんが、Entourage ではメッセージに vCard を添付するには、vCard をファイルとしてハード ディスク上に保存する必要があります。通常、Entourage の vCard は Entourage Temp サブフォルダに作成され、ここからメッセージにドラッグされます。このサブフォルダを使用することもできますが、このスクリプトでは代わりに MUD フォルダを使用します。このようにすると、UI で簡単に利用できるようになります。

Entourage では **me contact** は必須であり、これを削除することはできません。したがって、これを指定する行ではエラーは発生しません。**contact** には **vcards data** というプロパティがあります。このプロパティには、vCard を構成するデータがすべてテキスト形式で含まれます。またここでは、vCard の名前となる **me contact** の **name** も使用します。

スクリプトの最初の Entourage ブロックの次のセクションでは、vCard ファイルを作成します。まず、作成するファイルの完全なパスを設定します。これは、MUD へのパス、自分の名前、.vcf 拡張子をつなげたものになります。

その次のコマンドのグループは Standard Additions のコマンドです。Standard Additions の Dictionary は、Macintosh スクリプト エディタ で確認することができます。File Read/Write Suite を探してください。

ファイルに書き込むには、`open for access file [pathname] with write permission` を指定する必要があります。この指定により、ファイルが存在しない場合は新しいファイルが作成され、存在する場合はそのファイルへのアクセスが実行されます。次に、コマンド `set eof f to 0` によってデータを消去します (vCard には前回からの古い情報が残っている可能性があるため、これを更新します)。その後、vCard データをファイルに書き込み (**write**)、ファイルを閉じます (**close**)。write は常に try/error ブロック内に指定し、エラーが起こった場合は、アクセスを終了 (close access) してください。ファイルにアクセスしたままの状態にしておくことはお勧めできません。

Entourage AppleScript

ほとんどの Macintosh コンピュータでは、オペレーティング システムのアドレスブックが .vcf (vCard) ファイルの既定の "所有者" になります。したがって、ファイルはアドレスブック プログラムの vCard アイコンで表示されます。ここで Finder に対し、ファイルの作成元の種類を OPIM に設定するよう指示する必要があります。これは Entourage の 4 文字から成る署名 (作成元の種類) です。また、正しい Entourage アイコンを取得するために、Finder に対しファイルの種類を vCrd に設定するようにも指示します。これで、ファイルの種類がデータベースやメッセージ フォルダなどではなく vCard になります。

ここでまた Entourage に戻り、前面のウィンドウが **draft window** であることを確認します。また、添付ファイルが既にあるかどうかを確認します。スクリプトを 1 分ごとのスケジュールで実行している場合は、毎分新しい vCard のコピーを追加しないようにする必要があります。したがって、自分の名前と .vcf 拡張子で構成される名前の **attachment** が既にあった場合は、何もしないでスクリプトを終了します (return)。該当する添付ファイルがなかった場合は、下書きウィンドウで新しい添付ファイルを作成します。ここではエイリアス オブジェクトを使用し、添付ファイルの **file** プロパティを指定します。これで、メッセージを送信するときには常に vCard が添付されるようになります。

逆の手順で、.vcf 添付ファイルをハード ディスクに保存するスクリプトを実行して、Entourage のアドレス帳に新しい連絡先を作成することができます (メール ルールなどで、名前に .vcf を含む添付ファイルがあるメッセージをフィルタしてから、このスクリプトを実行できます)。添付ファイルの保存先としては、MUD フォルダ内の vCard 用のサブフォルダや別の一時的な場所を使用できます。保存したファイルはスクリプトの最後に削除します。添付ファイルを削除したら、**read** コマンドで as Unicode text を指定してファイルを読み取り、r 変数に格納します。次に Entourage で、make new contact with properties {vcard data:r} を実行します。

ただし、vCard の送信に使用されたアプリケーションが、オペレーティング システムの Mail やアドレスブック アプリケーション、または vCard を別の方法で作成するアプリケーションであった場合、このスクリプトで問題が発生することがあります。Entourage では Unicode テキストを使用しますが、これは vCard データの書き込みを正常に動作させ、as Unicode text を指定して読み戻すためです。一方、Macintosh の他のアプリケーションは、vCard をテキスト形式 (文字列) で書き込みます。

さらに、オペレーティング システムのアドレスブックは別のプロトコルに従うようになり、Entourage 向けに調整できなくなりました。これらの vCard の 1 つを Unicode テキストとして読み取ると、Entourage の連絡先は空白になります。

Unicode の vCard を使用するアプリケーションが Entourage だけである場合は、以下の方法を使用できます。ここでは、テキストの最初の 2 文字に、16 進数の文字 FEFF のバイト オーダー マーク (BOM) があるかどうかを確認します (これは ASCII 文字の 254 と 255 です)。この簡単なテストで十分です。テキストが BOM で始まっている場合、それは Entourage の vCard で作成された Unicode UTF-16 のテキストです。そうでない場合は、Entourage と互換性のないテキスト形式の vCard です。

Entourage AppleScript

```
set r to read theVcard -- an .vcf alias (file) on your computer
if (ASCII number (character 1 of r2)) = 254 and ¬
    (ASCII number (character 2 of r2)) = 255 then -- checks for Unicode
    set r to read theVcard as Unicode text -- gets the data correctly
    tell application "Microsoft Entourage"
        make new contact at address book 1 with properties {vcard data:r1}
    end tell
else
    beep
    display dialog "This vCard is not from Entourage and cannot be " & ¬
        "imported as a contact. Sorry." buttons {"OK"} default button 1 ¬
        with icon 0
end if
```

別のアカウントからメッセージを送信する

ユーザーが電子メールを "パブリック" メール アドレスで受信することはよくあり、そのアドレスが別のメール アドレスに転送するエイリアスになっていることもあります。返信においては、別のアドレスから送信することが必要な場合があります。返信ではなく新しいメッセージを送信する際に手動でアカウントを変更するのを忘れて、当然のことながら Microsoft Entourage の既定のアカウントからメールが送信されます。しかし、特にビジネスのメールでは、自動的に正しいアカウントから送信できるようにしたいものです。

これは、次のスクリプトでメール アカウントと同じ **name** を持つ **category** を作成し、この分類をアドレス帳の連絡先に割り当てて、その連絡先へのメールについては指定したアカウントからのみ送信されるようにすることで実現できます。スクリプトは、繰り返しの **schedule** で 1 分おきに実行します。こうすることで、メッセージを送信する前にメッセージ ウィンドウをキャッチして、メッセージ作成中にアカウントを変更できます。

名前は、アカウントや分類ごとに入力する必要があります。それぞれを引用符で囲んでコンマで区切り、**otherAccountNames** プロパティのリストの中かっこで囲む必要があります。たとえば {"Correspondence", "billing@mybusiness.com"} のようにします。メール アカウント名には、メール アドレスそのものを使用している場合と、わかりやすい別の名前を指定している場合が考えられます。いずれの場合も、[ツール] メニューの [アカウント] をクリックし、[メール] タブをクリックすると表示される名前と同じにする必要があります (POP、IMAP、または Exchange の説明は除きます)。

```
(*enter account/category names in {},
comma-separated "quoted text" items*)
property otherAccountNames : {}

tell application "Microsoft Entourage"

    if otherAccounts = {} then
        beep
        return
    end if
```

Entourage AppleScript

```
try
    set frontWin to front window
    if class of frontWin • draft window then
        return -- quit if no new draft window open
    end if
on error
    return
end try

set recipsList to every recipient of frontWin

set exitRpt to false -- for outer loop
repeat with i from 1 to count recipsList
    repeat 1 times
        set theRecip to item i of my recipsList
        set eAddress to address of the address of theRecip
        if eAddress • "" then
            try
                set foundItem to item 1 -
                    of (find eAddress)
            on error -- not a contact
                exit repeat -- 1 times
            end try
        else -- group
            try
                set dName to display name of address -
                    of theRecip
                set foundItem to group dName
            on error
                -- text typed in without being group
                -- or email address
                exit repeat -- 1 times
            end try
        end if

        set theCats to category of foundItem
        if theCats = {} then exit repeat

        repeat with j from 1 to (count theCats)
            set catname to name of (item j of theCats)
            if {catname} is in otherAccountNames then
                try
                    set otherAccount -
                        to POP account catname
                on error
                    try
                        set otherAccount -
                            to IMAP -
                                account catname
                    on error
```

Entourage AppleScript

```

                                try
                                    set otherAccount ↵
                                        to ↵
                                        Exchange ↵
                                        account ↵
                                        catname
                                on error
                                    beep 2
                                    return
                                end try
                            end try
                        end try

                        tell frontWin to set its account ↵
                            to otherAccount
                        set exitRpt to true
                        exit repeat

                    end if
                end repeat
            end repeat -- 1 times

            if exitRpt then exit repeat -- done
        end repeat

    end tell

```

このスクリプトでは、**every recipient** を確認します。先頭の宛先だけを採用する場合は、外側の repeat ブロックと exitRpt 行を削除し、代わりに次のスクリプトを実行します。

```
set theRecip to first recipient of frontWin
```

AppleScript には Next Repeat 演算子がないため、代わりに外側の repeat ブロックの内側に repeat 1 times ブロックを含めることにより、同じ結果が得られるようにしています。このように、指定された分類を持つ連絡先または **group** 受信者が存在しない場合のように、現在のリスト項目で実行を継続する必要がなくなった場合、スクリプトは repeat 1 times ブロック内の exit repeat で外側の repeat ループ内の次のリスト項目に進みます。

したがって、外側の repeat ブロックを終了する必要がある状況、つまり同じ名前が分類とアカウントの両方に見つかった場合は、まず必要なこととして、変数 exitRpt を *true* に設定し、内側の repeat 1 times ブロックを終了します。repeat 1 times ブロックを終了した後、外側のブロックの exit repeat の条件として、exitRpt の *true* の値を使用します。

また、**message** には **account** プロパティがありますが、account のジェネリック クラスはないため、指定された名前の **POP account**、**IMAP account**、および **Exchange account** を順番に確認する必要があります。

アドレス帳を使用する

連絡先を作成するまたは開く

新しい連絡先を作成する

新しい **contact** は簡単に作成できます。**contact** クラスには 70 個以上のプロパティがあり、そのうち 63 個は **properties** プロパティを使用して取得できます。これらのほとんどは読み取り/書き込みプロパティで、すべて既定値 (ほとんどは "") を持つので設定する必要はありません。最小限必要なプロパティを設定したり、プロパティに関する情報を取得することができます。

Microsoft Entourage の最新バージョンでは、Microsoft Exchange のアカウントがある場合、複数のアドレス帳 (と予定表) を持つこととなります。この場合、どの **address book** に連絡先を作成するか、または作成先を既定のアドレス帳のままにしておくかを指定する必要があります。

Microsoft Office 2004 for Mac Service Pack 2 (11.2.0) 以降の Entourage では、**address book** と **calendar** が完全なクラスになっています。連絡先は **address book** クラスの要素となり、これまでのような **application** の要素ではなくなりました。アドレス帳にない連絡先に対し、`make new contact` を実行して名前で作成先を参照するスクリプトがいくつもできなくなないように、既定のアドレス帳を想定して、従来の用語が使用できるようになっています。

Exchange account がない場合は、アドレス帳はローカルの "このコンピュータ上" の 1 つのみになります。Exchange アカウントがある場合は、既定のアドレス帳は、既定のメール アカウントに関連付けられているアドレス帳になります。それが Exchange アカウントであれば、既定のアドレス帳は、プライマリの **Exchange address book** になります。既定のメール アカウントが **POP account** または **IMAP account** であれば、既定のアドレス帳は "このコンピュータ上" のローカルのアドレス帳になります。

これは、作成する場所を指定しなくても連絡先が作成されることを意味します。Office 2004 for Mac SP2 への更新前の Entourage のバージョンを使用している場合は、連絡先を作成する場所を指定しないでください。指定するとスクリプトでエラーが発生します。これより新しいバージョンであれば、任意の場所を指定することができます。

ローカルのアドレス帳を参照するには、`address book 1` や `address book id 14` のように指定します。Exchange のアドレス帳は次のように参照できます。

```
(get primary address book) of Exchange account 1
```

または

```
address book "Extra" of address book "Contacts" -  
of Exchange account "Name"
```

また、名前、インデックス、ID、または `whose` フィルタで指定する通常の方法も使用できます。

Entourage AppleScript

"サブフォルダ"としてネストされた Exchange アドレス帳がある場合、これらのフォルダは Entourage のメール フォルダの一覧にのみ表示され、中身にはアクセスできません。アドレス帳では、これらは "同列に" 一覧表示されるので、次の操作を行います。

- 追加のアドレス帳に固有の名前を付けます。
- サブフォルダとしてネストしている場合は、Entourage のメール フォルダ内の階層を確認してから、スクリプトを記述します。

既定のアドレス帳では、次のスクリプトを実行します。

```
tell application "Microsoft Entourage"
    set newContact to make new contact with properties ~
        {first name:"Phyllis", last name:"Harris", home address: ~
            {street address:"123 Main Street", city:"Sometown", ~
                state:"CA", zip:"12345", country:"USA"}, ~
            default postal address:home, home phone number:~
                "(312) 555-1234", mobile phone number:~
                "(312) 555-5678", category:{category "Family"}}
    set email1 to make new email address at newContact with properties ~
        {label:home, contents:"someone@example.com"}
    set email2 to make new email address at newContact with properties ~
        {label:work, contents:"someone2@example.com"}
    set default email address of newContact to email1
end tell
```

email address 要素を後で作成する代わりに、**email address** プロパティを someone@example.com へ簡単に設定することもできます。ただし、この方法で設定できるプロパティは 1 つだけであり、またその **label** を *home* (既定は *work*) に設定することはできません。**email address** が 1 つだけで、**label** プロパティの値を *work* にする場合には、便利な方法です。

いずれか 1 つのアドレスを既定の **email address** として選択するには、変数 (email1、email2) を **email address** 要素に設定する必要があります。既定では、最初に作成されたアドレスが既定のメールアドレスになります。

新しい連絡先または既存の連絡先について、設定または変更できる内容は他にもあります。ほとんどのプロパティはテキスト型ですが、**children** はリストであることに注意してください。さらに、**home address** のようにレコードとなるプロパティや、日付関連のプロパティがいくつかあります。後者に該当するのは、**anniversary** プロパティ、**birthday** プロパティ、および 2 つの **custom date field** プロパティです。

連絡先を開く

Entourage では、ユーザーごとにデータベースが保持され、すべての要素は、Microsoft Word 文書のようなコンピュータ上の個別のファイルではなく、このデータベース内に収められています。これにより、要素を開く場合や、変更する場合など、参照が容易になっています。要素は ID、インデックス、名前、ユーザーによって作成可能な whose フィルタのいずれかで、ほぼ常に取得することができます。ID は通常、確認できません。インデックスは通常、作成順に番号が生成され、これも確認できません。名前は確認できます。

Entourage AppleScript

連絡先を参照する場合、場所については、作成したときと同じ条件を適用します。Exchange アカウントがない場合は、場所を指定せずに `contact "Phyllis Harris"` で参照できます。名前で参照する場合は、正確な名前を取得しておく必要があります。

既定のアドレス帳に登録されている人であれば、Exchange に追加のアドレス帳があっても、同様に参照できます。ただし、次の方法では連絡先を参照できません。

```
contact "Phyllis Harris" of address book 1
```

個別に指定したアドレス帳内の連絡先を名前で参照することはできません。次の例は、時間はかかりますが正しく動作します。

```
first contact of address book 1 whose name is "Phyllis Harris"
```

タブ付きの連絡先ウィンドウでこの連絡先を開くには、次のように **contact** オブジェクトの前に有効な **open** コマンドの式を追加します。

```
tell application "Microsoft Entourage"
    open contact "Phyllis Harris" -- if in default address book
    -- or if not:
    open (first contact of address book 1 whose name is "Phyllis Harris")
end tell
```

姓と名を入れ替える

メッセージから [アドレス帳に追加] を使用して連絡先を作成したときに、メッセージの送信元の電子メールクライアントで名前が "姓, 名" の順に設定されていたということに気付く場合があります。他のアプリケーションから、このような連絡先を数多くインポートしたという場合も考えられます。Microsoft Entourage ではこの形式は認識されず、"名" フィールドに姓とコンマ、"姓" フィールドに名が設定され、姓ではなく名で並べ替えが行われることになります。簡単なスクリプトを使用して、正しい配置に設定することができます。

```
tell application "Microsoft Entourage"
    set backToFronts to every contact whose ↵
        first name ends with ", "
    repeat with theContact in backToFronts
        tell theContact
            set {realFirst, realLast} to {last name, first name}
            try
                set realLast to text 1 thru -2 of realLast
            on error
                set realLast to ""
            end try
            set {first name, last name} to {realFirst, realLast}
        end tell
    end repeat
    display dialog "All done!"
end tell
```

Entourage AppleScript

1 行目では、whose フィルタを使用してコンマ付きの連絡先だけを取得します。このようにしないと、配置が正しい連絡先の姓名も入れ替わってしまいます。まず、of theContact ではなく tell theContact ブロックを使用します。これは、次のように 1 行でプロパティ一覧を設定できるようにするためです。

```
set theContact's {first name, last name} to {realFirst, realLast}
```

tell ブロックで囲まずに theContact に対してこのコードを実行するとエラーが発生します。ただし、プロパティ一覧を 1 行で取得する次の処理はどんな場合でも有効です。

```
set {realFirst, realLast} to theContact's {last name, first name}
```

連絡先にアラームを設定する

Microsoft Word、Excel、および PowerPoint では、[ツール] メニューの [アラームの設定] をクリックすることにより、ドキュメントに Entourage **task** が作成され、そのドキュメントに関するアラームを受信できます。一方、Entourage 自体には、電子メール メッセージに関するものを除いて同じユーティリティはありません。

予定表のイベントや仕事には独自のアラーム機能があり、[アラームの設定] と同じ働きをしますが、**contact** については、赤いフラグを設定する以外にフラグを設定する方法はありません。連絡先に対するアラーム、つまり、連絡先に電子メール、電話、または手紙で連絡する必要があることを通知する仕事を作成するための、[アラームの設定] メニュー項目はありません。

次のスクリプトでは、このメニューと同じ動作を実行できます。ここでは、アラームが設定された仕事を作成し、連絡先名とアラームの設定時刻 (変更可能) を指定し、その仕事を連絡先にリンクします。

```
-- Prepare the times of the due date and reminder
set now to current date
copy now to today
set time of today to 0 -- midnight
set nextHour to (((time of now) div hours) + 1) * hours
set time of now to nextHour -- on the hour

tell application "Microsoft Entourage"
  try
    --get the selected contact(s)
    set theContacts to (get selection)
    --check that they are actually contacts
    if class of (item 1 of theContacts) • contact then -
      error number -128
  on error
    beep
    display dialog "First select a contact before running " & -
      "the script." buttons {"Cancel"} default button 1 -
      with icon 0
    return
  end try
```

Entourage AppleScript

```
repeat with theContact in theContacts
    repeat 1 times -- to allow for skipping contacts
        if class of theContact • contact then exit repeat
        set theName to name of theContact
        (* if the contact has no name, check for
        nickname, company, email address, IM address
        in that order until you find one *)
        if theName = "" then set theName to nickname of theContact
        if theName = "" then set theName to company of theContact
        try
            if theName = "" then set theName ↵
                to email address 1 of theContact
        end try
        try
            if theName = "" then set theName to ↵
                instant message address 1 of theContact
        end try
        --if still no name, forget this one
        if theName = "" then exit repeat
        -- go on to next one
        --set a red flag
        set flagged of theContact to true

        -- make the task with the info
        -- get the contact's categories and projects along the way
        set theTask to make new task with properties {name:↵
            ("Flagged Contact: " & theName), due date:↵
            today, remind date and time:now, category:↵
            (get category of theContact), project list:↵
            (get project list of theContact)}

        link theTask to theContact
        open theTask

    end repeat
end repeat
end tell
```

スクリプトの最初のセクションでは、作成する仕事の **due date** とアラームの時刻を設定します。可能な場合、日付操作はアプリケーションの `tell` ブロックの外側で実行するのが最も良い方法です。ブロック内に指定すると、エラーが発生する可能性があります (エラーが発生し、`tell` ブロックから抜け出すことができなくなったときは、`tell AppleScript to ...` を実行すると有効な場合があります)。AppleScript の他のデータ型と異なり、日付、リスト、レコード、およびスクリプト オブジェクトは、変数が設定 (**set**) されている場合、可変となります。

現在の時刻 (**current date**) に設定されている変数 `now` から派生する 2 つの異なる日付 (時刻) が必要となるため、変数 `now` を `today` に設定するのではなく、コピー (**copy**) します。このようにしないと、`today` を変更した際に `now` も変更されます。

Entourage AppleScript

now 変数と today 変数は日付オブジェクトであり、日付と時刻を表します。today の時刻を 0 に設定すると、時刻が午前 0 時に変更されます。時刻は整数の秒で計算されます (この例では、仕事の期限を期限日の午前 0 時に設定する必要があります)。

日付変数 now をもう一度使用して、日付計算を使用する必要があります。アラームの設定時刻は、未来の時間に設定しておきます。また、now から取得する 3:14:47 PM のような時間ではなく、3:00:00 PM などのきりの良い時間にします。AppleScript には、**minutes**、**hours**、**days**、**weeks** などの定数があります。これらは秒の整数値で表される時間単位で、値はそれぞれ 60、3,600、86,400、604,800 となります。これらの定数を使用できるため、時刻設定のたびに数値を思い出したり計算しなくても済むので便利です。

ここでは演算子 div を使用し、now の時刻を時間 (3,600) で割った被除数の整数部分を出しています。この整数部分が現在の時刻の時間部分で、午後 2 時の場合は 14 になります。ここに 1 を足し (ここでは 15 になります)、それに時間 (3,600) をもう一度掛けて、午後 3 時を秒数に換算します。これを now の新しい時刻とします。now の時刻が 23:00 以降の場合、時刻は自動的に 0:00 にリセットされ、曜日と日付が次の日に進められます。

スクリプトの次のセクションでは、まず **selection** を取得します。選択項目として考えられるのは、連絡先、メッセージ、仕事、予定表内の 1 つのイベントなどのアイテムのリストです。1 つのアイテムだけが選択されている場合は、単一アイテムのリストになり、メッセージやメモ ウィンドウでテキストが選択されている場合は、テキストやフォルダのリストになります。また、ごくまれに選択項目がない場合もあります。

このスクリプトを実行する前に、[アドレス帳] またはカスタム ビューで 1 つ以上の連絡先を選択しておく必要があります。このスクリプトでは、選択項目の item 1 の **class** を確認し、それが実際に **contact** オブジェクトであることを確かめます。item 1 がない場合 (空の選択項目の場合)、または選択項目がない場合、on error ブロックでキャッチされます。item 1 のクラスが **contact** オブジェクトではない場合は、エラーが呼び出され、これも on error でキャッチされます。その場合、警告音が鳴ってダイアログボックスが表示され、スクリプトは終了します。スクリプトを終了する **return** コマンドがありますが、これは実際には実行されません。これは、Mac OS X の最新のバージョンでは、ディスプレイ ダイアログボックスの唯一のボタンである [キャンセル] ボタンによって、スクリプトを終了する error number -128 が自動的に発生するためです。

次のセクションでは、まず repeat ブロックがあり、その内部に repeat 1 times ブロックがあります。こうすることで、クラスが **contact** クラスでない場合にアイテムをスキップ (exit repeat) できるようにしています (Entourage の [プロジェクト] 領域に複合的なカスタム ビューがある場合、さまざまな種類のアイテムが選択されている可能性があります)。内側の repeat 1 times ブロックがない場合に exit repeat を指定すると、すべてが終了します。repeat 1 times は、AppleScript にはない Next Repeat の代わりになるものです (これを実現するためのスクリプトの構成方法は他にもありますが、これよりも深い if ブロックのネストが必要になります)。

残りのスクリプトの大部分は、連絡先の名前がない場合に、名前として使用する項目を見つけ出すためのものです。Dictionary には、**contact** クラスの name プロパティはありませんが、連絡先は **application** の要素として "名前" で指定できます。この場合の name は、**contact** クラスの **display name** プロパティをポイントする類義語として実装されています。表示名は、ドキュメントの説明はありませんが、**first name** プロパティと **last name** プロパティを組み合わせた名前であり、姓名両方とも空でない場合に、姓名の間にスペースが入ります (空の場合、スペースは入りません)。

Entourage AppleScript

contact クラスは、実際には **address** クラスのサブクラスであるため、**display name** プロパティを持ちます。したがって、このクラスは 1 つの **address** プロパティを持ちます。ただし、要素として多数の電子メールアドレスを持つこともできます。**address** プロパティは、**default email address** プロパティを示し、プロパティがない場合は "" を示します。

連絡先に姓も名もない場合、ニックネーム、会社名、既定の電子メール アドレス、既定の IM アドレスの順にテキストが検索され、アドレス帳の [名前] 列に表示されます。これもスクリプトの検索対象であり、これらの項目のいずれかが見つかりと検索は停止します。これらの項目のフィールドに何も入力されていない場合、それは誤った連絡先である可能性があり、スクリプトは前述の `exit repeat` を使用して、次の選択された連絡先に移動します。

次は、**contact** オブジェクトの **flagged** プロパティを `true` に設定します。これによってアドレス帳の 1 列に赤いフラグが設定されます。次に、新しい仕事を作成します。仕事の **name** は "Flagged Contact: " プラス連絡先名とし、期限は `today`、**remind date and time** (アラーム) は変更された `now` (次の時間) とします。ちなみに、ここでは既存の行への追加として、仕事の **category** を連絡先のカテゴリ (ある場合) に設定していますが、これには `get` を明示的に指定する必要があります。また、**project list** (プロジェクト) プロパティにも同様のことを行っています。

最後に、仕事と連絡先をリンクさせ、仕事 (のウィンドウ) を開きます。作成時に変数 `theTask` を設定するのは、最後の 2 つのコマンド `link` と `open` で指定できるようにするためです。期限とアラームは、必要に応じて調整できます。

今後このスクリプトを使用することを検討している場合、スクリプトを Entourage の [Script Menu Items] フォルダに保存し、`control+⌘+C` キーなどのショートカット キーを名前の最後に追加してください。

ユーザー設定フィールドを使用してデータをインポートする

Microsoft Entourage で、[ファイル] メニューの [エクスポート] をクリックし、[連絡先] のチェック ボックスをオンにして連絡先をテキスト ファイルにエクスポートしても、連絡先の **category** 情報はエクスポートされません。しかし、Microsoft Excel にテキスト ファイルをインポートする場合など、分類情報がどうしても必要になるときもあります。分類情報を取得する方法の 1 つとして、スクリプトを使用し、Entourage で提供されているユーザー設定フィールドの 1 つに分類名を入力する方法があります。ユーザー設定フィールドは 8 つありますが、通常、必要になるのは一部のフィールドだけです。

たとえば、UI で "メモ 8" という名前のフィールドに分類情報を入力する場合、このフィールドを "分類" という名前に変更できます。フィールド名を変更するには、アドレス帳で任意の連絡先を開き、[メモ 8] ラベルをクリックし、[ラベルの編集] をクリックします。この操作をスクリプトによって行うこともできます。**contact** (およびその他の Entourage オブジェクト) は、複数の分類を持つことができます。

contact クラス (および **event**、**task**、**note**、**group** など) の **category** プロパティが分類名のリストになっているのはこのためです。

下のスクリプトでは、分類名のリストをテキストに変換する際に区切り文字が必要になります。このスクリプトではコンマを使用しますが、セミコロンやスラッシュなどの他の区切り文字を使用することもできます。これを行うには、リストの中かっこ内にある選択した区切り文字に対して `set AppleScript's text item delimiters` を実行し、リストをテキストに強制型変換します。

Entourage AppleScript

次に、区切り文字を元の状態 (ほとんどの場合、既定の "") に戻します。後で元の状態に戻す理由は、スクリプトを実行しているアプリケーションが Macintosh スクリプト エディタ か、メニューから実行された Entourage か、[システム] メニューから実行されたシステム イベントかに関係なく、そのセッション全体 (Quit まで) にわたってこの区切り文字が保持されるためです。区切り文字の設定を厳密に行っていない他のスクリプトが、リストをテキストに強制型変換する際に影響を受ける可能性があります。

スクリプトを次に示します。

```
tell application "Microsoft Entourage"
    set custom field eight name of address book 1 to "Categories"
    set categoryContacts to every contact ↵
        where its category • {}
    repeat with i from 1 to (count categoryContacts)
        set theContact to item i of my categoryContacts
        set catNames to {}
        set theCats to category of theContact
        repeat with j from 1 to count theCats
            set theCategory to item j of theCats
            set end of catNames to name of theCategory
        end repeat
        set AppleScript's text item delimiters to {"", "}
        set catNames to catNames as Unicode text
        set AppleScript's text item delimiters to {""}
        set custom field eight of theContact to catNames
    end repeat
    beep
    display dialog All done!" buttons {"OK"} ↵
        default button "OK" with icon 1
end tell
```

2行目は、whose ではなく where its となっていることに注目してください。この場合、これは必須です。このようにしないと、空のリスト {} を取得します。これは、Entourage では、**contact** クラスの **category** プロパティと **category** クラスの両方 (およびその他のオブジェクトの **category** プロパティ) でキーワード **category** が使用されるためです。UI またはスクリプトで作成するすべての分類は、**category** クラスのメンバになります。

whose フィルタで使用する際に、同じ名前のクラスとプロパティの間に名前空間の競合があると、AppleScript ではプロパティよりもクラスが優先されます。連絡先はすべて **contact** クラスに属し、**category** クラスになることはないため、結果は常に {} (空) になります。しかし、同等の式で単語 *its* を使用することで、**application** ではなく連絡先をコンテナに指定し、そのプロパティとして **category** を指定すると、すべてが明確になり、求めている結果を得られます。every message of some folder where its account is... を使用してフィルタ処理するときも、同じようにする必要があります。

この競合を避けるために、Entourage 開発チームは Entourage 2004 の **project** クラスを作成したときに、対応するプロパティに、"**project list**" という名前を付けました。これは **contact**、**event**、およびその他のクラスのプロパティです。その結果、どのような場合でも *whose* を使用できます。

また、repeat ブロックの 1 行目には、別の代名詞 my があります。これにより、リストでの反復が最適化され、リストが膨大になったときに役に立ちます。処理速度が飛躍的に向上するので、この代名詞を使用してください。ただし、この代名詞はハンドラ内ではなくスクリプトの上位でのみ有効です。ハンドラ内で有効にするには、反復処理のハンドラを含むスクリプトの先頭でリスト (categoryContacts) をグローバル変数として宣言する必要があります。

メッセージからグループを作成する

多くの受信者に向けて送信されたニュースレターなどの電子メール メッセージを受信したり、複数の受信者を指定して電子メールを送信することがあります。それらのメッセージに含まれるすべての受信者を 1 つのグループにまとめてアドレス帳に保持しておけば、毎回同じグループのメンバー全員にメッセージを送信できるようになります。

Microsoft Entourage では、仕分けルールの [差出人をアドレス帳に追加] を使用するか、単一の名前とアドレスをクリックして [アドレス帳に追加] をクリックすることによって、1 人ずつアドレス帳に追加することはできますが、一度に複数の受信者を追加する方法はありません。そこで、次のスクリプトを使用すると、複数の受信者を同時に追加してグループ化することができます。最初に対象のメッセージを選択してから、スクリプトを実行します。

```
tell application "Microsoft Entourage"
    display dialog "Enter a name for the group to be made from " & ¬
        "the To recipients of selected messages:" default answer "" ¬
        with icon 1
    set groupName to text returned of result
    display dialog "Do you want the group set to show " & ¬
        "individual addresses or NOT show addresses in messages " & ¬
        "to the group?" buttons {"Cancel", "Show addresses", ¬
        "Don't show"} default button 3 with icon 1
    if button returned of result = "Don't show" then
        set showOnlyName to true
    else
        set showOnlyName to false
    end if

    set theGroup to make new group with properties ¬
        {name:groupName, show only name in messages:showOnlyName}
    set selectedMessages to (current messages)

    set entryAddresses to {} -- initialize an empty list
    repeat with i from 1 to (count selectedMessages)
        set theMsg to item i of my selectedMessages

        repeat with toRecip in (every recipient of theMsg ¬
            whose recipient type is to recipient)
            set {dName, eAddress} to {display name, address} ¬
                of (address of toRecip)
            if dName "" then
                set nameAddress to "\" & dName & "\" <" ¬
                    & eAddress & ">"
            end if
        end repeat
    end repeat
end tell
```

Entourage AppleScript

```
else
    set nameAddress to eAddress
end if
if {eAddress} is not in entryAddresses then
    make new group entry at theGroup -
        with properties {content:nameAddress}
        -- note content syntax!
    set end of entryAddresses to eAddress
end if
end repeat
end repeat
beep 2
open theGroup
end tell
```

このスクリプトでは、最初に新しい **group** の名前の入力を求めるダイアログ ボックスが表示されます。**display dialog** に空のテキスト ボックスを含めるために `default answer:""` を使用することに注目してください。その他のオプションについては、Standard Additions Dictionary の **display dialog** エントリを参照してください。次に、スクリプトでは、変数 `groupName` が `text returned of result` (テキスト ボックスへの入力) に設定されます。

次に、グループの名前と電子メール アドレスを表示するかどうかの確認を求めるダイアログ ボックスが表示されます。これは、UI を使用してアドレス帳のグループのウィンドウを開いたときに表示される、同様のチェック ボックスに対応しています。続いて、ユーザーの回答に応じて、スクリプトで変数 `showOnlyName` が `true` または `false` に設定されます。

ここで、新しいグループを作成し、変数 `theGroup` を割り当てて、**name** プロパティと **show only name in messages** プロパティに、これらの値のために割り当てた変数を指定します。

次に、選択したメッセージを取得し、`repeat` ループ内で `selectedMessages` リストの各アイテム (各 **message**) を処理します。必要に応じて、このスクリプトを複数のメッセージに対して一度に実行することも、単一のメッセージに実行することもできます。**current messages** は、アイテムが 1 つだけの場合でも、常にリストです。

外側の `repeat` ループを開始する前に、空のリスト `entryAddresses` を初期化します。このリストにエントリを作成して追加します。これを外側の `repeat` ループの外に 1 回だけ設定することで、処理対象のメッセージすべてについて連絡先の重複を避けることができます。

各メッセージを処理する外側の `repeat` ループ内で、特定のメッセージの各 **to recipient** を処理する内側の `repeat` ループに進みます。**cc recipients** も含める必要がある場合は、`whose recipient type is to recipient` 句を削除します。

recipient クラスは、特殊なクラスです。このクラスの **address** プロパティは、別の **address** プロパティ (文字列、つまり実際の電子メール アドレス) と **display name** プロパティ (メッセージに含まれていない場合は "" (空) のこともあり) の両方を含むレコードになっています。これらのプロパティには、変数 `eAddress` と `dName` を割り当てます。次に、表示名の有無を確認する必要があります。これは、表示名が空ではない場合 (`dName ≠ ""`)、表示名の後にスペースを 1 つ入れ、続けて電子メール アドレスを `< >` 山かっこで囲み、それを `nameAddress` 変数に割り当てる必要があるためです。表示名が空の場合 (`else`)、電子メール アドレスのみを含めます。

Entourage AppleScript

dName が存在する場合に、追加の引用符で囲まれることに注目してください。Entourage では、表示名に句読点 (イニシャルの後のピリオドなど) が含まれている場合に、引用符で囲む必要があります。そうしないと、スクリプトの後半で **group entry** を作成するときにエラーが発生します。句読点が含まれていないものを引用符で囲んでも問題は起こりません。UI では、表示名に引用符が不要であれば自動的に削除されます。

また、AppleScript では、文字としての二重引用符を \ (円記号またはバックスラッシュ) でエスケープする必要があることに注意してください。文字としての引用符を、テキストを表す変数 (ここでは dName) に連結する場合は、"\\" & dName & "\"" となります。このスクリプト内では、最後の (外側の) 閉じる引用符は、文字テキストの残り (<) が追加された後、次の連結文字列が追加される前に付けられます ("\" & dName & "\" <")。

次に、電子メールアドレス eAddress が既に entryAddresses のリストに追加されているかどうかを確認します。これは重複を避けるためです。nameAddress ではなく eAddress をチェックする理由は、前のインスタンスに表示名が含まれているかどうかや、表示名が異なるかどうかは問題ではないためです。現在のメッセージが重複していても、今後送信するメッセージが毎回重複しないようにします。

entryAddresses リストに、この電子メールアドレスが含まれていない場合は、**at** でグループを指定してこのグループ内に新しいグループ エントリを作成し、リストに電子メールアドレスを追加します。

AppleScript でリストに新しいアイテムを追加するには、次のコードが最も優れていて効率的です。常にこの方法を使用することをお勧めします。

```
set end of entryAddresses to eAddress
```

新しい単一アイテムのリストを既存のリストに連結する方法では処理が遅くなります。この方法では、AppleScript によって強制的にメモリ内にリストが複製され、そのリストも反復処理されます。copy eAddress to end of entryAddresses を使用する方法も同じ理由で効率が下がります。

重要

以下の情報は、このスクリプトで最も特殊な部分です。後でグループを操作するときに使用できるように覚えておくか、記録して保存しておくことをお勧めします。

Entourage Contact Suite の **group entry** についての Dictionary エントリには以下のように記載されています。

```
group entry n, pl group entries : every group entry group entry
```

```
ELEMENTS
```

```
contained by groups.
```

```
PROPERTIES
```

```
content (address, r/o) : address of entry
```

Entourage AppleScript

唯一のプロパティである **content** の "型" を確認すると、**address** クラスであることがわかります。Macintosh スクリプト エディタ で **address** へのリンクをクリックすると、次のように表示されます。

```
address n, pl addresses : every address

ELEMENTS
contained by application.

PROPERTIES
display name (Unicode text) : the name used for display
address (string) : the e-mail address
```

この **address** クラスには、**address** と **display name** の 2 つのプロパティがあります。たとえば、次を実行したとします。

```
get content of every group entry of group "Group Name"
```

既存のグループの名前を使用すると、各エントリの内容が次のレコード構造を持っていることがわかります。

```
{address:"someone@example.com", display name:"Phyllis Harris"}
```

したがって、新しいグループに新しいグループ エントリを作成する際、次のような構文を考える可能性があります。

```
make new group entry at theGroup ~
    with properties {content:{address:eAddress, display name:dName}}
```

しかし、実際は次の構文を使用します。

```
make new group entry at theGroup ~
    with properties {content:nameAddress}
```

ここでは、nameAddress テキストを次の形式にしています。

```
set nameAddress to "\"" & dName & "\" <" & eAddress & ">"
```

実際、重複の確認以外の目的では、元のメッセージの各受信者の **address** プロパティを取得し、それを個別の 2 つのプロパティ (**address** と **display name**) に解析しなくても、新しいグループ エントリの内容にできるように思われます。

しかし、Entourage AppleScript では、新しいグループ エントリを作成するときは、Phyllis Harris <someone@example.com> というテキスト形式を使用するか、表示名がない場合は someone@example.com という形式を使用する必要があります。

{address:"someone@example.com", display name:"Phyllis Harris"} というコンテンツのレコード構造は使用できません。

Entourage では、従来からこの構造が採用されているため、変更すると多くのスクリプトが機能しなくなります。

カレンダーを使用する

イベントを作成するまたは開く

新しい **event** ("calendar event" と呼ばれる AppleScript クラス) は、使用できるプロパティは少なくなりますが、**contact** と同じ方法で作成することができます。**Exchange account** を持っていれば、複数の新しいイベントを作成することもできます。

```
tell application "Microsoft Entourage"
    set newEvent to make new event with properties {subject:-
        "New Event", start time:-
        date "1/8/07 10 am", end time:-
        date "1/8/07 11 am", remind time:15, location:-
        "My event", content:"Some notes here." category:-
        {category "Recreation"}}
end tell
```

start time プロパティと **end time** プロパティには、コンピュータの短い日付形式を入力できます。この日付形式は Macintosh スクリプト エディタ によって日付オブジェクトにコンパイルされます。

```
tell application "Microsoft Entourage"
    set newEvent to make new event with properties {subject:-
        "New Event", start time:-
        date "Monday, January 8, 2007 10:00:00 AM", end time:-
        date "Monday, January 8, 2007 11:00:00 AM", remind time:15, location:-
        "My event", content:"Some notes here.", category:-
        {category "Recreation"}}
end tell
```

上に示したスクリプトでは使用されていませんが、Dictionary には Microsoft Exchange イベントにのみ使用できるプロパティが数多く存在し、出席依頼の作成に使用されるプロパティ (**to recipients**) も 1 つあります。**to recipients** を指定した場合、イベントが作成されるとすぐに電子メール メッセージとして出席依頼が送信されることに注意してください。

all day event を *true* に設定する場合、**start time** プロパティは午前 0 時に、**end time** プロパティは翌日の午前 0 時 (複数日のイベントを作成する場合は最終日翌日の午前 0 時) に設定する必要があります。Microsoft Entourage によって終了時間が修正されますが、予定したその日になっていない場合があるため、正しい日程を設定するには、この方法をお勧めします。

recurring イベント (`recurring:true`) を作成するには、UI で作成した同じ定期的なパターンのイベントから、定期的なパターンを示す適切な文字列を取得します。このイベントは複雑すぎて、ここですべてのルールを説明することは困難です。興味のある方は、RFC の「iCal recurrence」を参照してください。

イベントを開くことは、Entourage データベース内のその他のアイテムを開くことと同じです。イベントを特定し、それを開きます (**open**)。アイテムの開き方に関する詳細については、「連絡先を作成するまたは開く」を参照してください。

Entourage AppleScript

イベントと他のアイテムとの主な相違点は、同じ名前を持つイベントが複数存在することです。そのため、event "Meeting" などの名前だけでイベントを取得することは推奨されません。この場合、その名前で作成された最初のイベントのみが取得されます。

名前と時間帯の両方でフィルタ処理する場合は、whose 句が必要です。正しく時間が設定されていない可能性もあるので、その時間だけを指定することも推奨されません。1日や1週間単位で指定することをお勧めします。

```
tell application "Microsoft Entourage"
    set theEvent to first event whose subject is "My Event" and ¬
        start time ? date "Monday, January 8, 2007 12:00:00 AM" and ¬
        start time < date "Tuesday, January 9, 2007 12:00:00 AM"
end tell
```

重要

Microsoft Exchange アカウントを持っている場合は、複数のアドレス帳と予定表について知っておく必要があります。詳細については、「連絡先を作成するまたは開く」を参照してください。アドレス帳と連絡先に関する情報はすべて、予定表とイベントにも適用されます。

イベントから仕事を作成する

Microsoft Entourage 2004 の仕事とメモは Microsoft Exchange サーバーではなくローカルに保存されています。**task** の作成方法は、**message**、**contact**、および **event** の作成方法と同じです。新しいアイテムの作成方法の詳細については、「連絡先を作成するまたは開く」を参照してください。

次に、新しい仕事を作成するスクリプトを示し、仕事とイベントの主な相違点であるアラームの性質の違いについて説明します。

Entourage メニューには、[メッセージからイベントを作成]、[メッセージからメモを作成]、[メッセージから仕事を作成] の3つのスクリプトがあります。イベントから仕事を作成できると便利なのがよくあります。特に、別のアプリケーションで作成された .ics ファイル (Entourage にインポートできるが、決まった時間のイベントとして表示される) を受け取り、それを実際には仕事として保存したい場合などです。

これには、次にスクリプトを使用します。

```
tell application "Microsoft Entourage"
    try
        (*get the currently selected event(s)
           (in calendar or custom view) *)
        set theEvents to (get selection)
        if theEvents = {} or class of (item 1 of theEvents) ¬
            is not event then error number -128
    on error
        (*if there are no events selected, or the first item selected
           is not a calendar event, warn the user and then quit*)
        beep 2
        display dialog "First select an event in the calendar or " & ¬
            "a custom view. Then try again" buttons {"Cancel"} ¬
            default button 1 with icon 0
```

Entourage AppleScript

```
end try

repeat with theEvent in theEvents
    repeat 1 times
        (* if any subsequent item (in a custom view) selected
        is not a calendar event, warn the user and then quit *)
        if class of theEvent is not event then exit repeat

        (* get the information from the event,
        and store it in variables*)
        set {theTitle, theLocation, theStartTime, hasReminder, -
            theReminder, ifRecurring, theRecurrence, -
            theCategories, theProjects, theNotes, theLinks} -
            to theEvent's {subject, location, start time, -
            has reminder, remind time, recurring, -
            recurrence, category, project list, content, -
            links}

        (*if it's a recurring event, find when the next
        occurrence is due and make that the start time *)
        if recurring of theEvent then
            set now to (current date)
            copy now to checkDate -- copy, not set
            set year of checkDate to ((year of checkDate) + 1)
            -- a year from now

            set allOccurrences to calculate recurring dates for -
                theEvent from theStartTime to checkDate
            set nextOccurrence to missing value -- initialize
            repeat with i from 1 to (count allOccurrences)
                set thisOccurrence to item i -
                    of my allOccurrences
                if thisOccurrence > now then
                    -- first occurrence after now
                    set nextOccurrence to thisOccurrence
                    exit repeat
                end if
            end repeat
            if nextOccurrence = missing value then exit repeat
            -- skip, since no longer occurring

            set theStartTime to nextOccurrence
            -- forget occurrences in the past
        end if

        (*need to convert from event remind time (minutes before
        event as integer) to precise remind date and time,
        as date object, for task) *)
        if hasReminder then set theReminder to -
            (theStartTime - (theReminder * 60))
```

Entourage AppleScript

```
(*set the task's due date to just the date
(i.e., midnight) of the event's start time*)
set time of theStartTime to 0

(* remove or comment out (--) the next line if you
don't want the location added to the title *)
if theLocation ≠ "" then set theTitle to ~
    (theTitle & "(" & theLocation & ")")

(*if it's not a recurring task, must omit 'recurring'
from properties*)
if ifRecurring then
    set newTask to make new task with properties {name:-
        theTitle, due date:theStartTime, ~
        completed:false, recurring: true, ~
        recurrence:theRecurrence, ~
        category: theCategories, ~
        project list:theProjects, content:theNotes}
else
    set newTask to make new task with properties {name:-
        theTitle, due date:theStartTime, ~
        completed:false, category: theCategories, ~
        project list:theProjects, content:theNotes}
end if

(*sets the task's reminder to the same as the original
event's reminder: or comment out the line with --
to not duplicate the reminder *)
if hasReminder then set remind date and time ~
    of newTask to theReminder

link newTask to theEvent
repeat with i from 1 to (count theLinks)
    set theLink to item i of theLinks
    try
        link theLink to newTask
    end try
end repeat

(* optionally delete the event by uncommenting
(remove -- from next line)
*)
--delete theEvent

    end repeat -- 1 times
end repeat

if (count theEvents) = 1 then open newTask

end tell
```

コード中段は、**recurring** のイベントと仕事に関するセクションです。たとえば、今週または今月の予定表で選択したイベントが定期的なイベントで、その初回が数週間前または数年前であるような場合、それを定期的な仕事に変換する際は、2 年前や 2 週間前の **due date** を過去の期限切れ分として残すよりも、次回の分を初回の仕事とする方が適切です。定期的な仕事の形式が、そのときどきで完了していない仕事は 1 つだけあるものだからです。Entourage では、現在発生している仕事を完了するまで次の仕事を作成されないため、元のイベントの **start time** の古い期限のみが表示されることとなります。

このスクリプトでは、Entourage 2004 で導入された **calculate recurring dates for** コマンドを使用しています。このコマンドでは、**from** と **to** を使用して期間を設定する必要があります。期間の開始は元の開始時間とし、期間の終了は現在より 1 年後とします。これで、誕生日のような 1 年に 1 回のイベントであっても、次回の分も含まれることとなります。期間内のすべての日付を順に確認し、現在より後 (> now) に最初に見つかったものが次回の分です。次回の日付を新しい `theStartDate` とし、それが仕事の期限として設定されます。

event オブジェクトの **remind time** プロパティは、UI にある設定のように、イベントの開始時間までの時間 (分) を表す整数です。一方、**task** オブジェクトの **remind date and time** プロパティは、正確な日付と時間 (UI では、ミニ カレンダーと内蔵時計を使用して指定) を表す **date** オブジェクトです。この理由は、期限は日付のみで時刻のコンポーネントを持たないためです。

したがって、どのような場合でも、イベントのアラーム (開始までの時間) を 60 倍して分を秒に変換し、それをイベントの開始時間から減算して、仕事のアラームに使用する日付 - 時刻形式を得る必要があります。また、別の変数を使用して、イベントの開始時間の時刻を 0 (午前 0 時) に設定し、仕事の期限に適した形式にします。

次の行では、イベントの "場所" フィールドの情報がある場合に、それをかっこで囲んで仕事のタイトルに追加しています。これを行わない場合は、この行を削除してください。

最後に、イベントが定期的な仕事かどうか再確認する必要があります。定期的な仕事でない場合、**recurring** プロパティが `false` であっても、新しい仕事の作成時に **recurring** プロパティと **recurrence** プロパティを完全に削除する必要があります。そうしないと、新しい仕事は、毎日既定のパターンで繰り返す定期的な仕事になります。したがって、2 つの可能性に対応するために、`if` ブロックが必要です。

オプション行として、仕事を作成した後、元のイベントを削除 (**delete**) する行があります。元のイベントを削除する場合は、**delete** の前のダブルダッシュ `--` を削除してこの行のコメントを解除してください。定期的でないイベントを選択した場合 (`count theEvents = 1`)、そのアイテムが新しい仕事として開かれ、必要に応じて内容を修正することができます。

期限の過ぎた仕事のスケジュールを再設定する

ここで紹介するスクリプトでは、今日完了できなかった仕事をすべて更新し、明日を新しい **due date** として同じ時刻にアラームを設定します。このようにして、期限の過ぎた仕事が放置されて忘れられるのを防ぎます。このスクリプトは、午前 8 時などの就業前 (ただし午前 0 時以降) に毎日実行するように設定できます。スクリプトが完了すると、多くのアラームが作動する可能性があります。

```
set today to (current date)
set time of today to 0
set yesterday to (today - (1 * days))
tell application "Microsoft Entourage"
    set procrastinators to every task whose due date = yesterday -
        and completed is false
    repeat with procrastinator in procrastinators
        tell procrastinator
            set due date to today
            if has reminder then
                set reminderTime to remind date and time
                set remind date and time to (reminderTime + -
                    (1 * days))
            end if
        end tell
    end repeat
end tell
```

Microsoft Entourage では、アラームが設定されていない仕事でも **remind date and time** として "1904 年 1 月 1 日金曜日午前 0 時" が設定されていると見なされます。Entourage 2004 以前は、ブール値の **has reminder** プロパティは存在しませんでした (上のスクリプトでは、**has reminder** プロパティが *true* となっている仕事を確認しています)。したがって、スクリプトではこの代わりに "1/1/1904" という日付を確認する必要がありました。このようなスクリプトはいまだに数多く見られます。

一部の Macintosh アプリケーションでは、ある目的で 1904 年 1 月 1 日を "開始時間" として扱っています。この時間から 2³¹ 秒、つまり 2035 年 2 月 5 日を "終了時間" とし、この時間を超えて処理できないように機能に制限を設定しています。Microsoft Excel にも同じような制限があり、開始時間は 2000 年 1 月 1 日です。これは Microsoft Windows バージョンとの互換性のためです。

Exchange の予定表とローカルの予定表を結合する

Microsoft Exchange Server を実行するサーバーで作業していて、Microsoft Office 2004 for Mac Service Pack 2 (11.2.0) 以降のバージョンの Microsoft Entourage を使用している場合は、複数の予定表とアドレス帳を使用できるなど、大幅に向上した機能を利用できます。たとえば、非公開のアイテムは "このコンピュータ上" のローカルの **calendar** に、作業アイテムは **Exchange calendar** に保存するなど、予定表の使い分けが可能です。ただし、これらの予定表を重ね合わせて、スケジュール全体を一度に確認するような機能はありません。

Entourage AppleScript

1つの予定表にあるイベントを別の予定表にすばやく挿入したり、別々のアドレス帳にある連絡先をすばやく取得することができれば便利です。これらを実際に同期するには複雑なスクリプトが必要になりますが、1つの予定表から別の予定表にすべてのイベントを移動する方法で結合するのであれば、スクリプトを定期的に行うことで同期することができます。

次のスクリプトでは、"このコンピュータ上"のローカルの予定表から、**Exchange account**にある個人用の **primary calendar** にすべてのイベントを移動します。

```
tell application "Microsoft Entourage"
    move (every event of calendar id 13) to primary calendar ↵
        of Exchange account 1
end tell
```

これ以外に、別の方法を使用してもかまいません。ローカルの予定表には固定の **ID** が組み込まれており、常に `id 13` になります (ローカルの **address book** は `address book id 14` です)。これらは、実際にはフォルダになっています。この理由は、Office 2004 for Mac SP2 への更新以降、**calendar** クラスと **address book** クラスは **folder** のサブクラスになったためです。calendar 1 や address book 1 のように参照することもできます。

calendar クラスは、Entourage が Office 2004 for Mac SP2 に更新される前には存在しませんでした。以前の **event** クラスと **contact** クラスは **application** の要素であり、現在のように **calendar** クラスや **address book** クラスの要素ではありませんでした。知っておくと便利なことは、イベントと連絡先は現在もアプリケーションの要素として扱うことができるということです。これで以前のスクリプトも無駄になりません。しかしこの場合、検出されて認識されるのは、既定の電子メールアカウントに関連付けられた予定表とアドレス帳に属するイベントと連絡先だけになります。

Microsoft Exchange アカウントを持っていない場合は、ローカルの予定表、イベント、アドレス帳、および連絡先だけを使用していることになり、すべて検出されるので問題はありません。Exchange アカウントを持っていて、それが既定の電子メールアカウントである場合は、以前のスクリプトを使用するとしても、どのイベントまたは連絡先であるかを指定しない限り、標準として設定されている Exchange の予定表と **Exchange address book** の Exchange イベントおよび連絡先だけが検出されます。

POP account または **IMAP account** を既定の電子メールアカウントとして設定している場合は、"このコンピュータ上"にあるローカルのイベントと連絡先だけが検出されます。[ツール]メニューの[アカウント]をクリックし、[メール]タブをクリックして既定のアカウントを切り替えると、別のアカウントに切り替えることができます。スクリプトを作成して編集する際には、**name** または **ID** によって、どの予定表またはアドレス帳であるかを常に正確に指定することができます。従属する予定表と予定表のサブフォルダもすべて指定できます。名前で指定する場合は、「calendar "Extra" of calendar "Calendar" of Exchange account 1」のように常に階層全体を示す必要があります。

Exchange アカウントを複数持っている場合は、それぞれを名前で参照 (またはその ID を取得) する必要があります。インデックス番号で参照する方法は、アカウントの追加と削除を行うと番号が変更される可能性があるため信頼できません。ID を取得した場合でも、データベースを再構築すると変わる可能性があります。これは Entourage のすべてのオブジェクトに当てはまります。名前もいつでも変更可能なため、再構築を行わない限りは通常 ID が最も信頼できます。

Entourage AppleScript

次は、イベントを移動するのではなくコピーすることについて検討してみます。ローカル イベントを同じ場所に複製してから別の予定表に移動しようとする場合、Standard Suite の **duplicate** コマンドは、**to** パラメータを指定しない場合でも正しく動作しません。このコマンドを試すとエラー (Can't set event id 1295 to event id 1295) が返されます。これは ID も複製しようとしているが不可能であるという意味です。ID は読み取り専用で一意であり、新しいオブジェクトの作成時に自動的に生成されるものです。

これまで、メッセージ、連絡先、またはイベントをサーバー上で、またはサーバーに移動することはできませんでした。その理由は、Standard Suite の **move** コマンドと **duplicate** コマンドが両方とも、結果 (移動または複製されたアイテム) を返す必要があるためです。しかし、IMAP サーバーと Exchange サーバーは新しいアイテムの結果を返す前にタイムアウトになる可能性があるため、この機能は実現できませんでした。

Office 2004 for Mac SP2 以降のバージョンに更新された Entourage では、この問題が回避されています。このバージョンでは、“移動した” アイテムに対して同じ ID が保証されます。移動したアイテムは新しい場所で新しいアイテムとなり、元のアイテムは削除されます。コンピュータ上のあらゆるコンテキストで行われる “移動” は、これと同じ方法で行われています。したがって、Entourage はサーバーから ID が返されるまで待機することなく、サーバーを信用して ID 結果をすぐに返すことができます。つまり、内部処理において、ID はコピーまたは置換されるようになりました。これは、複製されたアイテムに対しては不可能です。両方のアイテムが同じ ID を持つことはありえません。したがって、移動は可能ですが複製は不可能です。

“コピー” を行うには、古いイベントのプロパティに基づいて新しいイベントを作成します。この方法では、元のイベントのプロパティをすべて取得することができ、その後、コピー先の予定表に新しいイベントを作成し、プロパティなどを設定できます。これは従来からある方法です。

しかし、たとえば 3,000 件のイベントを別の予定表にこの方法で移動するとなると、個々のプロパティを取得して設定するための個別の AppleEvent を送信することになり、with properties ブロックで実行したとしても時間がかかります。連絡先の場合は、確認するプロパティと要素が約 65 個あり、さらに時間がかかります (空の既定のプロパティを記述する作業はおそらく回避できます)。

iCal data プロパティ (連絡先の場合は **vcard data** プロパティ) を利用すると、より速く移動することができます。このプロパティでは、ほとんどのプロパティを一度に取得して作成できます。

ただし **category**、**project list**、および **links** の 3 つのプロパティだけは、これらが重要なものである場合は、個別に取得して作成する必要があります。これらは Entourage だけにあるプロパティで、**iCal data** プロパティには含まれていません (移動時間を追加する機能も、**iCal data** や AppleScript のプロパティで取得できない機能です)。連絡先については、**vcard data** プロパティで画像も取得できるというメリットがあります。これに対応する AppleScript のプロパティはありません。

ただし、Exchange の予定表では、**iCal data** プロパティから新しいイベントを作成することはできません。これはローカルの予定表でのみ可能であるため、代わりにローカルの予定表で作成し、移動する必要があります。

Entourage AppleScript

1つのイベントをコピーするスクリプトを次に示します (選択したアイテムのクラスを確認するには各自でコードを追加してください)。リンクとプロジェクト一覧を使用しない場合は、これらを省略できます。

```
tell application "Microsoft Entourage"
    set theEvent to item 1 of (get selection)
    set {iCalData, theCategories, theProjects, theLinks} ←
        to theEvent's {iCal data, category, project list, links}
    set dup to make new event at calendar id 13 with properties ←
        {iCal data:iCalData, category:theCategories, project list:→
            theProjects}
    repeat with i from 1 to (count theLinks)
        set theLink to item i of theLinks
        link theLink to dup
    end repeat
    move dup to primary calendar of Exchange account 1
    -- or to any calendar
end tell
```

メモについて

メモは他のアイテムと同じように作成したり開くことができます。アイテムの作成方法および開き方の詳細については、「連絡先を作成するまたは開く」を参照してください。メモに対して設定できるプロパティは、**name**、**content**、**category**、および **project list** の4つだけです。

連絡先を Excel にエクスポートする

ここでは、Microsoft Entourage を Excel と連携させる便利なスクリプトを紹介します。これは、Visual Basic for Applications (VBA) では実行できないことを Microsoft Office の AppleScript で実行できる例です。実際に、このスクリプトの Excel 部分を VBA で実行しても、たとえば、**run VB macro** コマンドを使用したとしても AppleScript に結果を返すことができないため、意味がありません。

連絡先などの情報を Entourage から Excel にエクスポートする操作が必要になることはよくあります。AppleScript を使用しない場合は、**address book** 全体 (ローカルの "このコンピュータ上" のアドレス帳のみ) をタブ区切りテキスト ファイルにエクスポートしてから、Excel にインポートする方法しかありません。

この方法では、不要な連絡先が数千件と、**contact** ごとに 13 個以上の電子メール アドレスを含む 65 個のフィールドがインポートされます。これらのフィールドには、重要な **category** 情報が欠けています。これでは、不要な行 (連絡先) と列 (フィールド) の削除に何時間も費やすこととなります。Microsoft Word にメールの差し込みを行うと少し選択しやすくなりますが、それでも情報は Excel にエクスポートできる形式 (テーブル) にはなりません。

Entourage AppleScript

AppleScript を使用すると、この操作を簡単に実行できます。この例では、連絡先をカテゴリ (category "Work" のすべての連絡先) で選択しています。任意のカテゴリ、または複数のカテゴリで選択することができます。別の方法として、contacts whose last name starts with "B" や contacts whose company is "Northwind Traders" のみを指定したり、任意のフィールドに対する whose フィルタを指定することもできます。

アドレス帳や **custom view** の連絡先を選択して取得することもできます。このとき、アイテムのクラスを確認することを忘れないでください。必要なフィールドだけを指定して、他の指定は省略することもできます。Excel で列を削除する必要はありません。

次の例では 16 個のフィールドがありますが、通常、必要なのは 3 ~ 4 個のフィールド (**last name**、**first name**、電子メール、(勤務先) **phone** など) だけです。

```
tell application "Microsoft Entourage"
    set theContacts to every contact where its category -
        contains {category "Work"} --OR:
    --set theContacts to the selection

    set allContactProps to {"Last Name", "First Name", "Title", -
        "Company", "Department", "Email", "Work Phone", "Cell Phone", -
        "Home Phone", "Assistant", "Street", "City", "State", "Zip", -
        "Spouse", "Notes"} -- header row as first sublist
    repeat with theContact in theContacts
        try
            set email to (first email address of theContact whose -
                label is work)
        on error
            try
                set email to default email address
            on error
                set email to ""
            end try
        end try
        tell theContact
            set contactProps to {last name, first name, -
                job title, company, department, email, -
                business phone number, -
                mobile phone number, -
                home phone number, -
                assistant phone number, -
                business address's street address, -
                business address's city, -
                business address's state, -
                business address's zip, spouse, -
                description}
        end tell
        set end of allContactProps to contactProps
    end repeat
    set numRows to count allContactProps
    set numColumns to count (item 1 of allContactProps) -- 16 here
end tell
```

Entourage AppleScript

```
tell application "Microsoft Excel"
    set newWkbk to make new workbook with properties ~
        {name:"Work Contacts"}
    set lastCell to get address (get offset (range ~
        "A1" of active sheet of newWkbk) row offset (numRows - 1) ~
        column offset (numColumns - 1))
    set theRange to range ("A$1:" & lastCell) ~
        of active sheet of newWkbk

    set value of theRange to allContactProps
    autofit theRange --? could make Notes column very wide
end tell
```

Entourage セクションの最初の行で連絡先を取得します。この例では、category "Work" に該当する連絡先を選択します。whose ではなく where its のインスタンスを使用していることに注目してください。これは、**category** クラスが必要です。また、contains 演算子の構文では、含まれているアイテム ({category "Work"}) が含む側のアイテムと同じ **list** クラスである必要があります。連絡先は複数のカテゴリを含むことがあるので、カテゴリはリストです。

これは必須です。category "Work" をリスト中かっこ { } で囲まないと、can't make category id 4 into a vector というエラーメッセージが表示されます。

ほとんどの場合に扱うのは文字列のリストであり、文字列であれば AppleScript により単一アイテムのリストに強制的に型変換されるため、かっこで囲むことを忘れがちです。しかし、この強制型変換は Entourage オブジェクトには存在しません。そのため、contains 演算子と、逆の演算子である is in で、含む側のオブジェクトがリストである場合は、リスト中かっこ { } を忘れずに使用してください。

連絡先をカテゴリではなく **selection** で取得することもできます。上に示した例では、この方法のコードサンプルはコメントアウトしています。こちらを使用する場合は、選択アイテムのクラス (**contact**) をチェックし、内部 repeat 1 times ループを使用して、連絡先ではないアイテムを exit repeat でスキップできるようにします。

allContactProps 変数は、リストのリストであり、行と列を含む範囲 (2次元) の値に使用します。取得するフィールドの名前を含むヘッダー行で、この変数を初期化します。ここでは 16 個になります。フィールド数は必要に応じて増減できます。また、フィールドにわかりやすい名前を自由に付けることができます。

ここからは、関連する **email address** を見つける方法です。連絡先には、必要な数の電子メールアドレスを含めることができます。そのため、プロパティの取得よりも複雑な手法が必要です。通常は **default email address** を探しますが、これは連絡先のアドレスを問い合わせるだけで取得できます。しかし、ここでは、勤務先関連情報のみが必要なので、スクリプトで first email address of theContact whose label is work を取得してみます。

label などの別のプロパティを求めない限り、強制変換型では、常に **email address** がその **contents** プロパティ (実際の文字列であるアドレス) に解決されます。ラベルが導入される前の Entourage 2004 より古い Entourage のバージョンには、この contents プロパティだけが存在します。この強制型変換を保持することで、古いスクリプトがそのまま機能するようにしています。

Entourage AppleScript

default email address の取得は、**first email address** の取得に失敗 (on error) したときだけにしてください。これは、自宅の電子メールアドレスでも機能しますが、失敗する (連絡先に **email address** がまったく存在しない) 場合は、**email** 変数を "" に設定すると、エラーが発生しません。

これで、必要な他のプロパティも取得できるようになりました。他のプロパティとして 1 行で取得し、さらに今指定した **email** 変数も組み込むことができるのは、**tell theContact** 構文だけです。そこで、次のコードを試したとします。

```
set contactProps to {last name, first name, ... email, ... } of theContact
```

これではエラーが発生します。コンパイラは **email** を変数として認識せず、**contact** オブジェクトの **email** プロパティは存在しません。この **tell** ステートメントは、最初に単語をアプリケーションキーワード (できれば **tell** の対象オブジェクト **theContact** のプロパティ) に解決し、該当するアプリケーションキーワードが見つからない場合にのみコンパイラが変数を受け入れるので、うまく機能します。

Excel では、行数 (つまり連絡先レコード数) と列数 (各レコードのフィールド数、この例では 16) が必要です。任意のサブリストをカウントできますが、これは最初に置くことをお勧めします。このようにして、**Entourage** での処理が行われます。

Excel では、新しい **workbook** を作成し、値を設定 (挿入) する前に正しいサイズの **range** を取得します。これを行わないと、スクリプトでエラーが発生します。Excel 2004 の AppleScript では、A1 形式名でのみ範囲を参照できます。R1C1 形式では参照できません。

つまり、列数が判明しているだけでは不十分です。範囲アドレスを指定するには、最後の列を示す文字を確認する必要があります。最も簡単に取得するには、列数から 1 を引いた数と行数から 1 を引いた数を使用して **get offset of range (cell) "A1"** を実行します。1 を引くのは、列 A と行 1 が既にカウントされているためです。これにより、範囲内の最後のセルを取得できます (範囲全体ではありません)。

この最後のセル (範囲) に **get address** を使用すると、範囲指定に使用できる A1 アドレス形式を取得できます (この場合、必要な A1 形式のアドレス全体が取得されます)。この結果は常に "絶対参照" 形式の \$A\$1 であるため、範囲アドレスの最初の部分 (左上隅) と、**get address** コマンドで 2 番目の部分 (範囲の右下隅) 用に取得した **lastCell** 結果に対しても同様の方法を使用できます。

これで、この範囲の値をリストのリストに設定できるようになり、**worksheet** にすぐにデータを挿入できるようになりました。その後、この範囲に対して **autofit** を実行し、列幅を広げて各セルの内容を完全に表示することもできます。

"メモ" 列は最後の列ですが、長いメモの付いた連絡先がある場合、自動サイズ調整を実行したときに "メモ" 列の幅が広くなりすぎることがあります。また、連絡先メモの内容に行数 (改行文字) が多いと、行も縦長になることがあります。この場合の対処方法は、該当シート内の行の高さを特定の値に設定することくらいです。幅が広くなりすぎないようにするには、全範囲に自動サイズ調整を適用するのではなく、最後の列以外のすべてに自動サイズ調整を適用します。列も範囲であるため、自動サイズ調整は任意の列に個別に適用できます。