



Technical Note 04-28

外部プロセスコール

By: Frank Chang, 4D Evangelist
Technical Note 04-28

(原題: Launch External Process, the evolution of AP ShellExecute and AP Sublaunch)

概要

外部プロセスコールは、バージョン 2004 の数ある新機能のひとつです。このコマンドは、これまで 4D Pack で提供されていた AP ShellExecute および AP Sublaunch コマンドの発展型に相当します。従来のコマンドは、幾らか柔軟性という点で限られていました。新しい LAUNCH EXTERNAL PROCESS は、クロスプラットフォームの非常に強力な柔軟性のあるコマンドです。

すでに AP ShellExecute および AP Sublaunch コマンドを使用しているならば、この新しいコマンドの特徴を知りたいと思われることでしょう。サンプルデータベースは、このコマンドの 2 つの重要な機能を示すために用意されました。

- ・ 外部スクリプト (perl, php, vbscript, batch script, unix commands, etc.) からの返り値を取得することができる
- ・ 4D から別アプリケーションを起動することができる

内容

AP Sublaunch の概要
AP ShellExecute の概要
Launch External Process の概要
Launch External Process でアプリケーションを起動する
スクリプトから結果を取得する
Launch External process を補うもの
まとめ

AP Sublaunch の概要

AP Sublaunch コマンドを使用すれば、4D からアプリケーションを起動したり、ドキュメントを開いたりすることができます。

Windows の場合、アプリケーションの起動方法について、標準のウィンドウ、フルスクリ

ーン、最小化されたウィンドウ（タスクバー）で開くかどうかを指定することができます。実行可能なファイルに対してこのコマンドを実行することは、そのアプリケーションのアイコンをダブルクリックすることに相当します。AP Sublaunch コマンドには、アプリケーションで開くドキュメントの名前を渡すこともできます。

Macintosh の場合、このコマンドでアプリケーション、あるいはドキュメントを開くことができます。このコマンドを実行することは、それぞれのファイルのアイコンダブルクリックすることに相当します。起動が成功しなければ、エラーが返されます。

Windows の場合

```
C_TEXT($file;$app)
C_LONGINT($err)
$file:="C:\\Documents and Settings\\Database-test\\testPDF\\pub71.pdf"
$app:="C:\\Program Files\\Adobe\\Acrobat 5.0\\reader\\AcroRd32.exe"
$err:=AP Sublaunch ($app+" "+$file;0)
```

Macintosh の場合

```
C_LONGINT($err)
$err:=AP Sublaunch("MyDisk:MyFolder:MyApp";0)
```

AP ShellExecute の概要

AP ShellExecute コマンドを使用すれば、4D からアプリケーションを起動したり、ドキュメントを開いたりすることができます。このコマンドは Windows のために最適化されており、Windows だけで利用できる第 3 引数が用意されています。

Windows では、開くファイルへの絶対パスだけを渡せば、アプリケーションが自動的に起動します。関連付けられたアプリケーションの選択は OS が処理します。AP Sublaunch 同様、アプリケーションの起動方法について、標準のウィンドウ、フルスクリーン、最小化されたウィンドウ（タスクバー）で開くかどうかを指定することができます。第 3 引数で、アプリケーションに渡すパラメータを指定することができます。たとえば、ブラウザに対して URL を渡すことができます。この引数は Macintosh では処理されません。

Macintosh では、ファイル名、あるいは完全なアクセスパスを渡すことによって、アプリケーションあるいは指定したファイルを開くことができます。関連付けられたアプリケーションの選択は OS が処理します。起動が成功しなければ、エラーが返されます。

```
C_TEXT($file)
$file:="pub71.pdf"
$err:=AP ShellExecute ($file)
```

まとめると、どちらのコマンドでも別アプリケーションを起動することが可能ですが、ファイル名で起動できるのは AP ShellExecute だけです。その場合、関連付けられたアプリケーションの選択は OS が処理します。したがって AP Sublaunch の発展型が AP

ShellExecute であるということができるよう。

Launch External Process の概要

The Launch External Process コマンドによって、Windows、Macintosh どちらのプラットフォームにおいても外部プロセスを起動することができます。

Windows の場合、AP ShellExecute と同等の機能に加えて、AP ShellExecute にはない機能も提供されています。

Macintosh では、ターミナルで起動可能なあらゆるアプリケーションを起動することが可能です。これは 4D Pack にはなかった機能です。

コマンドを使用するには、第 1 引数にアプリケーションの絶対パスを渡します。

LAUNCH EXTERNAL PROCESS ("C:\\WINDOWS\\notepad.exe")

開くファイルを指定する場合は、第 1 引数にファイル名を追加します。

LAUNCH EXTERNAL PROCESS

("Applications/TextEdit.app/Contents/MacOS/TextEdit file.txt")

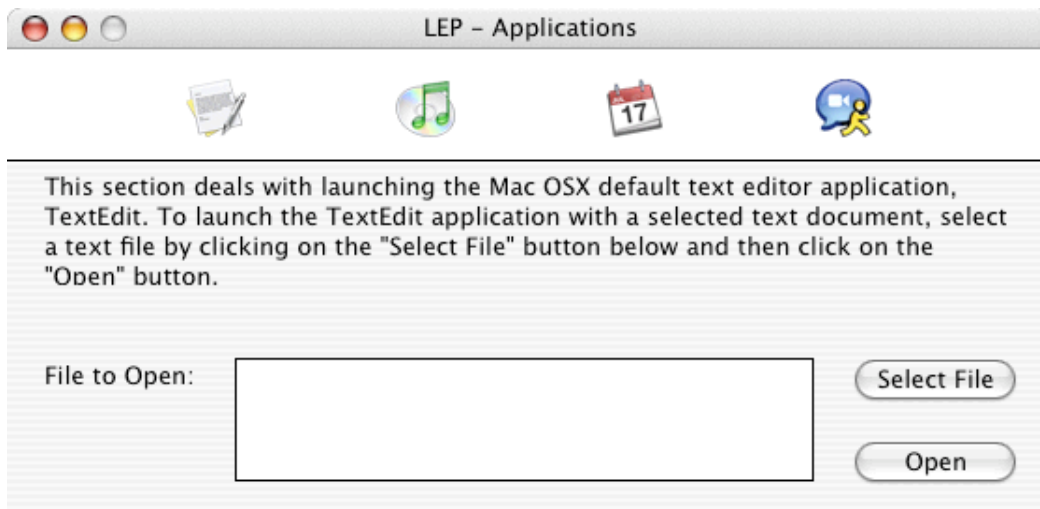
第 2、第 3、第 4 引数の使用は任意です。第 2 は入力ストリームであり、外部プロセスへの標準入力 (stdin) に使用されます。標準入力は、外部プロセスに入力変数を渡すために使用されます。たとえば、PHP スクリプトにパラメータリストを渡すことができます。この場合、第 1 引数で PHP スクリプトをコールし、第 2 引数でパラメータリストを渡すことになります。外部プロセスのスクリプトが実行を終えると、標準出力ストリーム (stdout) および標準エラーストリーム (stderr) が LAUNCH EXTERNAL PROCESS の第 3、第 4 引数に返されます。

コマンドが正常に実行されると、システム変数 OK に 1 が代入されます。ファイルが見つからなかったり、メモリ不足に直面したりした場合には OK に 0 が代入されます。

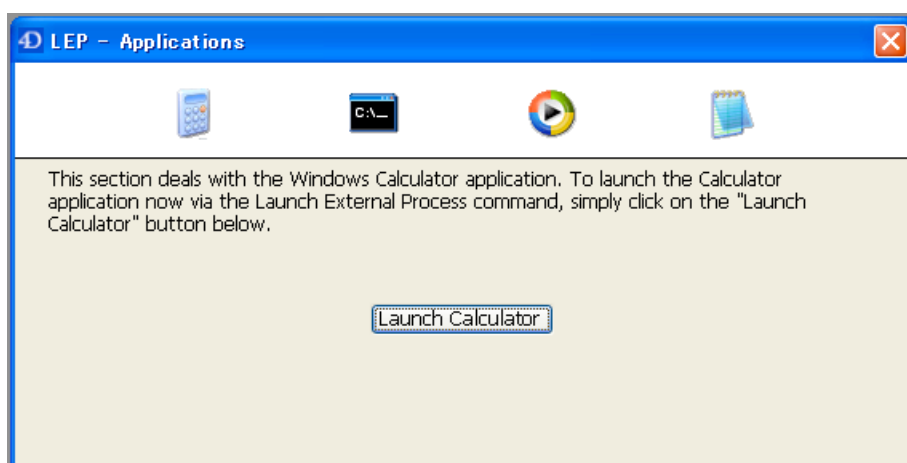
サンプルデータベース

それぞれのプラットフォームで標準のアプリケーションを起動するなど、LAUNCH EXTERNAL PROCESS の基本的な使用方法を確認することができます。

Macintosh 画面



Windows 画面



適当なアプリケーションを指定すると、起動のために新規プロセスが立ち上がります。メインプロセスが別アプリケーションを実行し続けるのは不都合なので、このようにします。立ち上がった新規プロセスは、メソッド LEP_Play_Apps をコールするために使用され、別アプリケーションが終了するまで開いています。この間、4D のプロセスは外部プロセスコールを実行した行で止まっており、別アプリケーションが終了すると、制御が 4D に返されて次の行に移ります。このように別アプリケーションを起動する場合は、外部プロセスコールメソッドを別プロセスで実行する点が大切です。以下はメソッド LEP_Play_Apps をコールしてアプリケーションを起動している例です。

```
C_LONGINT($proc)
```

```
$proc:=New process("LEP_Play_Apps";32*1024;"Application:iTunes";"iTunes";*)
```

```

C_TEXT($app;$1)
C_TEXT($cmd)

$app:=$1
Case of
    ¥ ($app="iCal")
        $cmd:="/Applications/iCal.app/Contents/MacOS/iCal"
    ¥ ($app="iTunes")
        $cmd:="/Applications/iTunes.app/Contents/MacOS/iTunes"
    ¥ ($app="iChat")
        $cmd:="/Applications/iChat.app/Contents/MacOS/iChat"
    ¥ ($app="TextEdit")
        C_TEXT($2;$file)
        $file:=$2
        $cmd:="/Applications/TextEdit.app/Contents/MacOS/TextEdit "
+$file
    ¥ ($app="Calc")
        $cmd:="c:¥¥windows¥¥system32¥¥calc.exe"
    ¥ ($app="WMP")
        $cmd:="C:¥¥Program Files¥¥Windows Media
Player¥¥wmplayer.exe"
    ¥ ($app="CMD")
        $cmd:="c:¥¥windows¥¥system32¥¥cmd.exe"
    ¥ ($app="Notepad")
        C_TEXT($2;$file)
        $file:=$2$cmd:="c:¥¥windows¥¥Notepad.exe "+$file
End case
LAUNCH EXTERNAL PROCESS($cmd)

```

OS X の場合、パッケージではなく、実際のアプリケーションへのパスが必要なことに注目してください。Notepad、Textedit 以外のアプリケーションに関しては、開くファイルを指定しないので、メソッドに対してパラメータをひとつだけ渡しています。

スクリプトから結果を取得する

外部プロセスに特定の処理をさせ、その結果を 4D で取得しようとする場合、これまでは煩雑な処理が必要でした。LAUNCH EXTERNAL PROCESS を使用すれば、そのような処理が容易に実現できます。

次のプロジェクトメソッド LEP_Play は、スクリプトの特定の処理をさせ、その結果を 4D で取得するコードの例となっています。

```

C_TEXT($1;$cmd)
C_TEXT($error)
C_TEXT($0;$out)
$cmd:=$1

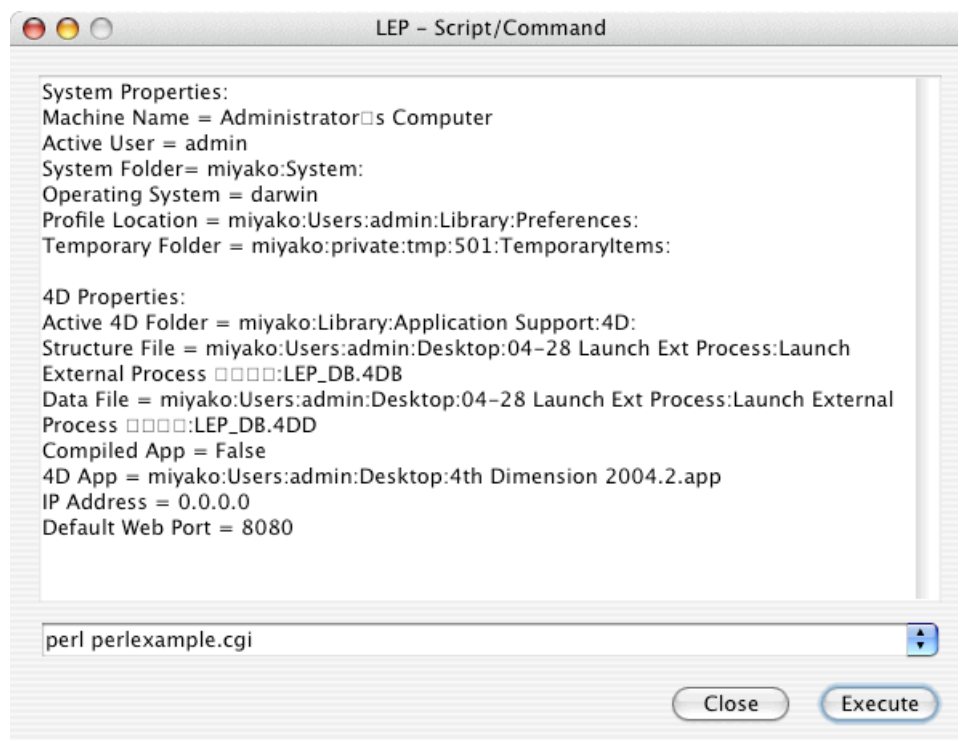
```

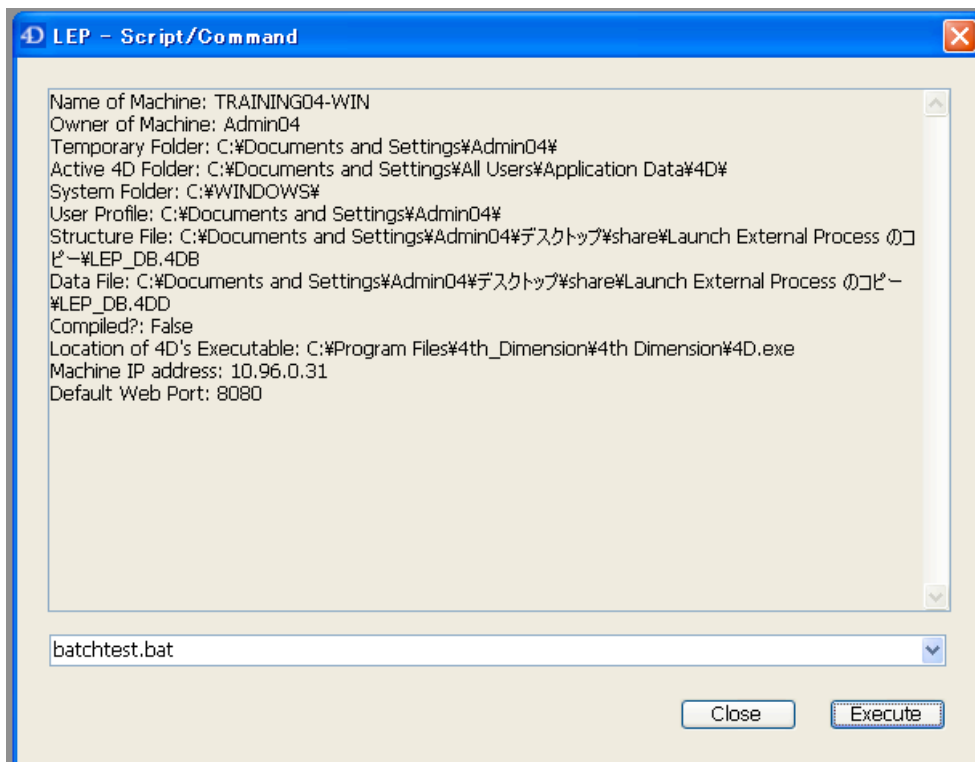
```

If (Count parameters=2)
    C_TEXT($2;$in)
    $in:=$2
Else
    C_TEXT($in)
End if
LAUNCH EXTERNAL PROCESS($cmd;$in;$out;$error)
If (OK=1)
    If ($out="")
        $0:=$error+Char(13)+Char(10)+$out Else
        $0:=$out
    End if
Else
    $0:=$error+Char(13)+Char(10)+$out
End if

```

サンプルデータベースのコンボボックスには、Windows ではバッチファイルと vbscript、Macintosh では UNIX コマンドと Perl CGI と PHP スクリプトがデモンストレーション用としてデフォルトで用意されています。スクリプトのパスは LAUNCH EXTERNAL PROCESS の第 1 引数、第 3、第 4 引数はそれぞれ stdout、stderr に使用されています。標準入力ストリームは今回、使用していませんが、それでもプレースホルダーとして変数を使用しています。





この場合、LAUNCH EXTERNAL PROCESS は新規プロセスではなく、メインプロセスで起動していることに注目してください。このように、スクリプトを実行する場合は、結果が返されるまで 4D を待機させるようにします。

Launch External process を補うもの

バージョン 2004 で追加されたもうひとつの新しいコマンドに SET ENVIRONMENT VARIABLE があります。このコマンドを使用すれば、環境変数とその値を設定することができます。このコマンドな強力な点は、起動しようとする外部スクリプトの変数を定義できる点にあります。たとえば、サンプルの batchtest.bat は次のようなスクリプトです。

```
Echo Name of Machine: %Machine_Name%
Echo Owner of Machine: %Machine_Owner%
Echo Temporary Folder: %Temporary_Folder%
Echo Active 4D Folder: %Active_4D_Folder%
Echo System Folder: %System%
Echo User Profile: %Profile%
Echo Structure File: %Structure_File%
Echo Data File: %Data_File%
Echo Compiled?: %Compiled_APP%
Echo Location of 4D's Executable: %Location_Of_4D%
Echo Machine IP address: %IP_Address%
Echo Default Web Port: %Default_Web_Port%
```

前後を%で囲われているのがシステム変数です。それぞれの値は、4D のプロジェクトメソッド LEP_Set_ENV_Var で設定されています。

```
SET ENVIRONMENT VARIABLE("Machine_Name";Current machine)
SET ENVIRONMENT VARIABLE("Machine_Owner";Current machine owner)
SET ENVIRONMENT VARIABLE("Temporary_Folder";Temporary folder)
SET ENVIRONMENT VARIABLE("Active_4D_Folder";Get 4D folder)
SET ENVIRONMENT VARIABLE("System";System folder(System ))
SET ENVIRONMENT VARIABLE("Profile";System folder
(Preferences or Profiles_User ))
SET ENVIRONMENT VARIABLE("Structure_File";Structure file)
SET ENVIRONMENT VARIABLE("Data_File";Data file)
SET ENVIRONMENT VARIABLE("Compiled_APP";String(Num
(Compiled application);"True;;False"))
SET ENVIRONMENT VARIABLE("Location_Of_4D";Application file)
SET ENVIRONMENT VARIABLE("IP_Address";LEP_Get_IP )
SET ENVIRONMENT VARIABLE("Default_Web_Port";String
(Get database parameter(Port ID )))
```

SET ENVIRONMENT VARIABLE コマンドがコールされるたびに、新しい環境変数が作られます。次に LAUNCH EXTERNAL PROCESS がコールされると、アプリケーション環境にこれらの変数がバインドされるので、結果として起動されるアプリケーションの変数を定義されたことになります。

まとめ

新しいコマンド LAUNCH EXTERNAL PROCESS は、4D Pack の AP ShellExecute や AP Sublaunch のような機能を提供するだけでなく、それ以上の力を有しています。ただ別アプリケーションを起動するだけでなく、結果を取得することも可能です。サンプルデータベースには両プラットフォーム用のデモンストレーションが含まれているので、可能であれば、是非、Windows および Macintosh それぞれで試してみてください。