SAS ユーザー総会 2017

SAS以外の言語を利用した、SASで実現困難なロジックや機能の実装について

○川上　貴弘
（電助システムズ株式会社）

Implementation of difficult-to-implement logic and functions in SAS, using languages other than SAS

Takahiro Kawakami
Densuke Systems Co., LTD.

要旨:

様々な理由により最新バージョンのSASを利用できない状況がある。

最新バージョンで提供されている機能を、旧バージョンのSASで、複雑なコーディングを用いず簡略化して実現する手段を検討した。

キーワード: 旧バージョン,dll, Delphi, SHA-256,

ハッシュ, sascbtbl, module関数

SASのバージョン

- 特定の業務のため、バリデーションを行うと、以降のSASのバージョンアップは(手順上)容易ではない。
- SASのバージョンアップによって得られるはずのメリットが享受できない。(特に新機能について。バグ対応はhotfixが提供されている場合がある)

例えばSAS9.4TS1M1で追加されたSHA-256

- Heartblood, poodle等、ベースとなる技術自体の脆弱性の発覚
- ハッシュアルゴリズムSHA-1の対衝突性を破られた

（ハッシュ…特定の文字列等から計算によって得られた一意の値）

- 暗号化方式は従来一般的であったSSLからTLSへ、ハッシュアルゴリズムはより安全であるSHA-256への移行が急速に進んでいる。
- しかしながら、SAS9.4TS1M0以前ではMD5しか対応していない。

SAS9.3でSHA-256ハッシュを生成する方法を検討

- 以下は除外した。
- インターネット接続環境が必要なもの
  - ブラウザ上で生成できるサービス/サイトは多いが、SASで実行するためには、オフラインで使用できること
- OpenSSLライブラリを利用するもの
  - OpenSSLライブラリには度々脆弱性が見つかっており、その都度ライブラリを更新するのは困難

SAS9.3でSHA-256ハッシュを生成する方法を検討

- アプリケーションと連携し易いdll, exeを作成できるDelphiを採用

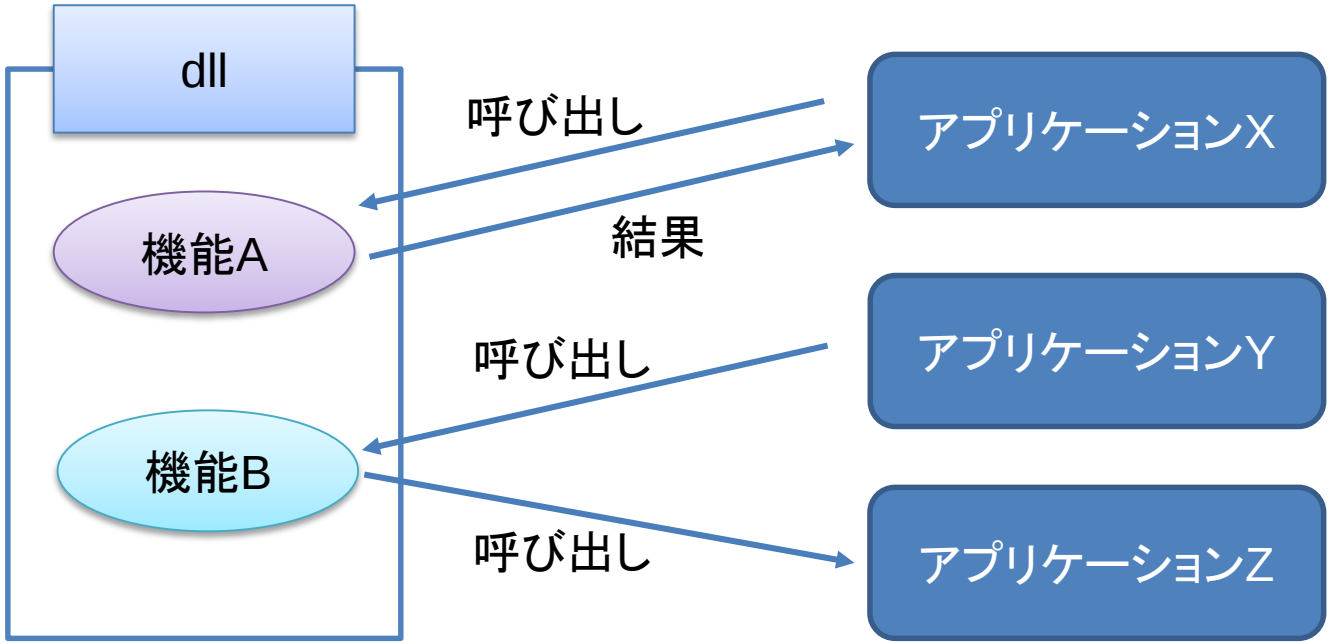
Delphiのコード

以下のコードを「myHash.dpr」として保存する。

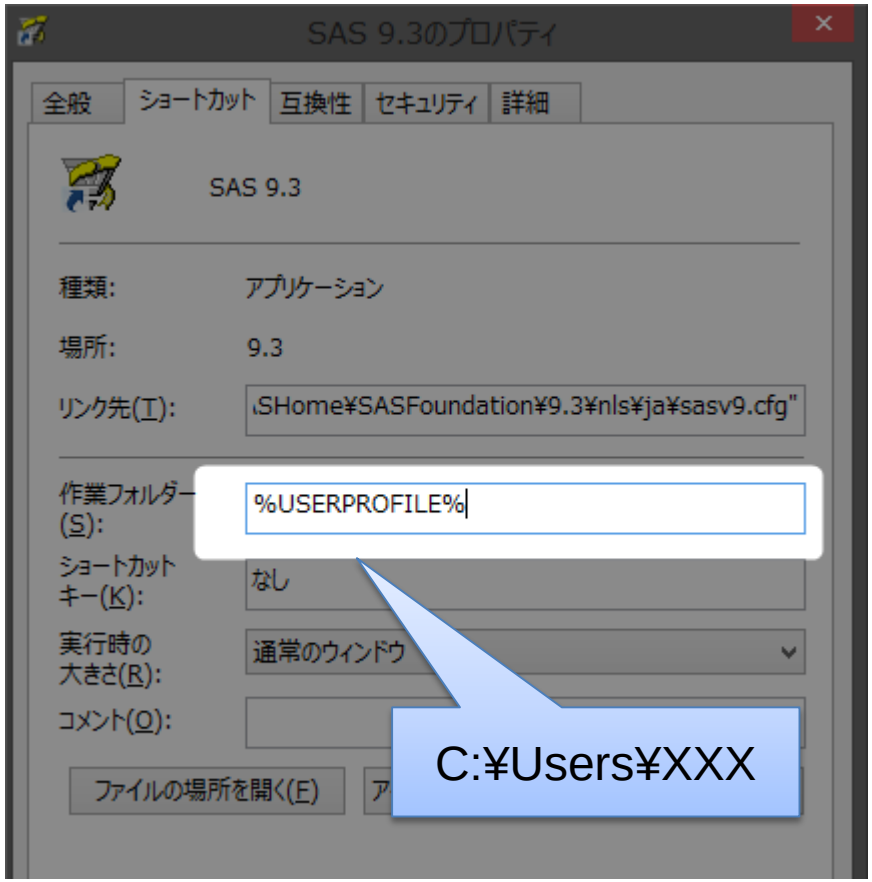
```
1.  library myHash;
2.  uses System.SysUtils, System.Classes, System.Hash;
3.  {$R *.res}
4.  function GetHash(s:PAnsiChar):PAnsiChar stdcall;
5.  var
6.    Hash: THashSHA2;
7.  begin
8.    Hash := THashSHA2.Create(SHA256);
9.    Hash.Update(trim(String(s)));
10.   Result:=PAnsiChar(AnsiString(Hash.HashAsString));
11. end;
12. exports GetHash;
13. begin
14. end.
```

dllとは

- それ自体では動作しないが、汎用的な機能がモジュール化され、アプリケーションプログラムからその機能を呼び出せるようにしたもの



dllの作成と配置



- Delphiを起動して、先程作成した「myHash.dpr」を開き、コンパイルを行う（Ctrl+F9）。
- コンパイルが完了すると、「myHash.dll」が作成される。
- SASでdllを使用するにはパスの通ったフォルダにコピーしておく必要がある。（例：%USERPROFILE%）

SAS9.3で実行

```
1.  filename SASCBTBL 'tmp.txt';
2.  data _null_;
3.    file sascbtbl;
4.    put 'routine GetHash';
5.    put '  minarg=1';
6.    put '  maxarg=1';
7.    put '  MODULE=myHash';
8.    put '  stackpop=called';
9.    put '  returns =char64;';
10.   put 'arg 1 char input byaddr format=$200.;';
11. run;
12. data _null_;
13.   length hash $64;
14.   source='ABC';
15.   hash = modulec('GetHash', source);
16.   put hash=;
17. run;
```

ログに
hash=b5d4045c3f466fa91fe2cc6abe79232a1a57cdf104f7a26e716e0a1e2789df78
と出力される

SASでdllを使用するときの作法

- SASCBTBLにfilenameで属性テーブルを割り当てる。
- 属性テーブルにはdll名、dllに埋め込まれた機能名、引数の数、呼び出し方、返り値、引数の詳細を指定する。
- WinAPIを使用した事例は多いが、独自dllを利用する事例はほとんどなく、うまく動作しない原因については、SASのMODULEC関数にコントロール文字列 '*E' を加え、実行時エラーをログに出力して確認した。
  「rc=modulec('*E','dll function',param);」

例）dll(myHash)が見つからない場合や、bitが異なる場合
ERROR: Module myHash could not be loaded.

検証コード（SAS9.4）

```
1.  data _null_;
2.  length hash $64;
3.    source='ABC';
4.    hash = put(SHA256(source),hex64.);
5.    put hash=;
6.  run;
```

ログに
hash=B5D4045C3F466FA91FE2CC6ABE79232A1A57CDF104F7A26E716E0A1E2789DF78
と出力される

まとめ

- Delphiを用いてdllを自作することで、旧バージョンのSASでは使用できない機能を比較的容易に実現することができた。
- 既存のソフトウェア資産等を利用し、SASをより有効に活用する方法が広まることへの一助となれば幸いである。

参考文献

日向俊二: Delphi XE2プログラミング入門,カットシステム (2012)
Embarcadero Technologies: Delphi online help docwiki(Delphi 10.2 Tokyo)
http://docwiki.embarcadero.com/Libraries/Tokyo/ja/System.Hash.THashSHA2
SAS Institute Inc.: Windows版SAS® 9.4第3版 (2014)
SAS Institute Inc.: SAS® 9.4関数とCALLルーチン リファレンス第4版 (2015)
SAS Institute Inc.: SAS® 9.4 Companion for UNIX Environments, Sixth Edition (2016)
David H. Johnson.: SAS® with the Windows API (SUGI 30)
Edward Foster.: Using the WIN32 API from SAS (PhUSE 2006)

謝辞

SAS Technical Support (SAS Institute Japan Ltd.)