

開発者必読!

“もしも”のときの デバッグ技法

.NETアプリケーション障害解析

株式会社NTTデータ
飯山 教史
IIYAMA, Takashi

第10回

デバッグに役立つツール



ツールを利用する意味

深刻なバグを調査するときには、利用するテクノロジー（OSや言語、ライブラリなど）の理解が不可欠である。しかしこれはソースコードや専門書籍を熟読しすべてを把握しなければならないということではない。関数の呼び

出し順やリソースの状態にどんな意味があるのかなど、「アーキテクチャをざっと理解しておく必要がある」ということなのである。

この理解を助けてくれるのがデバッグツールである。どんなツールであれ有識者が何らかの調査を行なうために開発されたのだから、そのツールの開発目的や処理結果を理解することはアーキテクチャの理解の手助けとなる。

今回は数多く存在するツールの中から、米Sysinternals社が提供するフリーウェア「Process Explorer」を紹介する。Process Explorerは以下のURLからダウンロードできる。

<http://www.sysinternals.com/utilities/processexplorer.html>

Process ExplorerはWindows開発者にとって必須といっても良いユーティリティであり、これを使えばデバッグに役立つさまざまな情報を取得できる。今回はこのツールの機能の中からDLLとハンドルを表示する機能について紹介する。



DLLの調査

Process Explorerは作成したアプリケーションのプロセス空間にどのDLLがロードされるのかを表示する機能を持っている。

.NETアプリケーションではクラスライブラリはDLLとしてビルドされるため、ロードされたDLLを特定のタイミングで確認できるなら、

「どのタイミングでどのクラスライブラリがプロセス空間に存在しているか」

がわかる。これはデバッグをするときにとても重要な情報となる。なぜなら例外発生時の手順を再現化できれば、解析対象を例外発生時に表示されていたDLLに絞り込むことができるからだ。

Process Explorerの機能でとくに優れているのは「前回ロードされていたときとの差分を異なる色によって識別できる」ことだ。まずはこれを簡単なサンプルを使って説明することにする。

Level

1 2 3 4 5

Technology Tools

- Visual Basic
- Visual C#
- Visual C++
- SQL Server
- Oracle
- Access
- ASP.NET
- Other:
 - ↓
 - Visual Studio .NET 2003
 - Process Explorer

リスト1：stringbuffer.dllのソース

```
namespace StringBuffer
{
    public class ArrangeString
    {
        public ArrangeString()
        {
        }
        public void SetString(ref string str)
        {
            str = "In DLL";
        }
    }
}
```

リスト1・2のサンプルコードを見てもらいたい。これは、ボタンのクリックにより「stringbuffer.dll」をロードして文字列を表示する簡単なWindowsアプリケーションである。

▶ 調査の前準備

まずは何も画面に表示していない状態でProcess Explorerを起動しておく。

ここでひとつ注意しなくてはならない機能がある。それは、Process Explorerはデフォルトで表示内容を自動更新してしまうことである。この機能はデバッグ時に致命的な欠点となる場合がある。たとえば一瞬だけロードされるようなDLLが例外を起こしていた場合、自動更新ではアンロードされたDLLが表示されないの、例外を起こしたDLLがProcess Explorerに表示されないことがあるからだ。

そこでProcess Explorerのメニューから [View] - [Update Speed] と移動して [Paused] に設定を変更すると更新条件を手動更新に変更できる (図1)。デバッグ時にはこの状態で作業を進めてもらいたい。

リスト2：stringbuffer.dllをロードして文字列を表示するサンプル

```
private void btnDisplay_Click(object sender, System.EventArgs e)
{
    string str;
    str = "";
    ArrangeString asDisplay = new ArrangeString();
    asDisplay.SetString(ref str);
    txtDisplay.Text = str;
}
```

図1：Process Explorerの画面更新設定を手動更新に変更

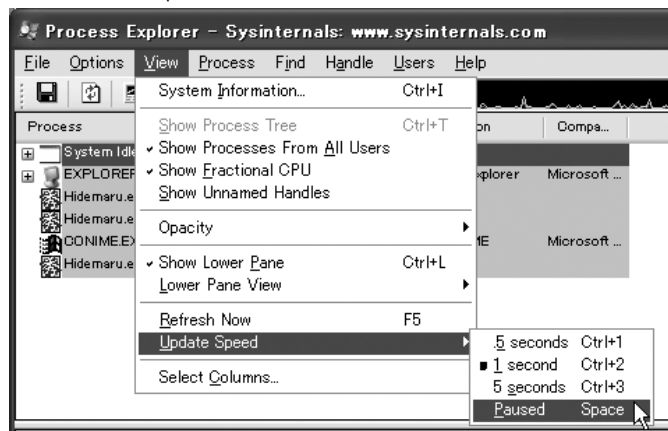


図2：Process Explorerによるプロセス表示



▶ DLLの表示方法

リスト2に示したサンプルのEXEファイル (DisplayString.exe) を実行し、[F5] キーを押して表示内容を更新する

と図2のように表示される。本稿はカラーではないので見にくいかもしれないが、新しく起動された「DisplayString.exe」が上のペインに表示されているの