

快速脱! 休日出勤

.NETアプリケーション障害解析の極意

デバッグ技法

第5回

ASP.NETアプリケーションの デバッグ—プログラミング工程—

株式会社NTTデータ
飯山 教史
IIYAMA, Takashi

Webアプリケーションの デバッグ

今回はASP.NETアプリケーションで発生した例外のデバッグについて説明する。ASP.NET (Web) アプリケーションといってもデバッグの方法はWindowsアプリケーションやコンソールアプリケーションなど他のアプリケーションと変わらない。ただし前回説明したダンプファイルの解析となると少々異なるところがある。

レベル >>> Level

1 2 3 4 5

言語 >>> Language

Visual Basic

ツール >>> Tool

- Visual Studio 2005 Team Edition for Software Testers
- WinDbg (Debugging Tools for Windows)
- SOS.DLL

そこでADPlusを設定してuser.dmpを取得する方法や、ダンプの解析の方法などASP.NETアプリケーション独自の解析が求められる部分を中心に説明する。

実行プロセスのダウン

まずASP.NETで例外を起こしてみよう。ここではサンプルとして、0除算をして例外を発生させるアプリケーション(リスト1)を用意した。

アプリケーションを実行する前に、Web.configファイルのdebugオプション

をデフォルトのtrueから以下のようにfalseにしておく。

```
<compilation debug="false"
strict="false" explicit="true"/>
```

この状態でアプリケーションを実行してみよう。上記のようにdebugオプションをオフにしたので、[F5] キー(デバッグ開始)ではなく、[Ctrl] + [F5] キーを押して「デバッグなしで開始」することを忘れないでほしい。

図1に示した「WebDev.WebServer.exe」がASP.NETを実行する(リクエストを処理する)プロセスである。このとき、表示されたInternet Explorer (図

リスト1: 0除算するASP.NETアプリケーション

```
Partial Class _Default
Inherits System.Web.UI.Page

Protected Sub btnException_Click(ByVal sender As Object, _
ByVal e As System.EventArgs) Handles btnException.Click
Dim i As Integer = 0
Dim j As Integer = 1
Dim k As Integer = 0

k = j / i
End Sub
End Class
```

図1: サンプルアプリケーションのプロセス



図2: サンプルアプリケーション

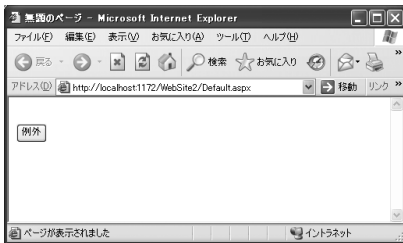
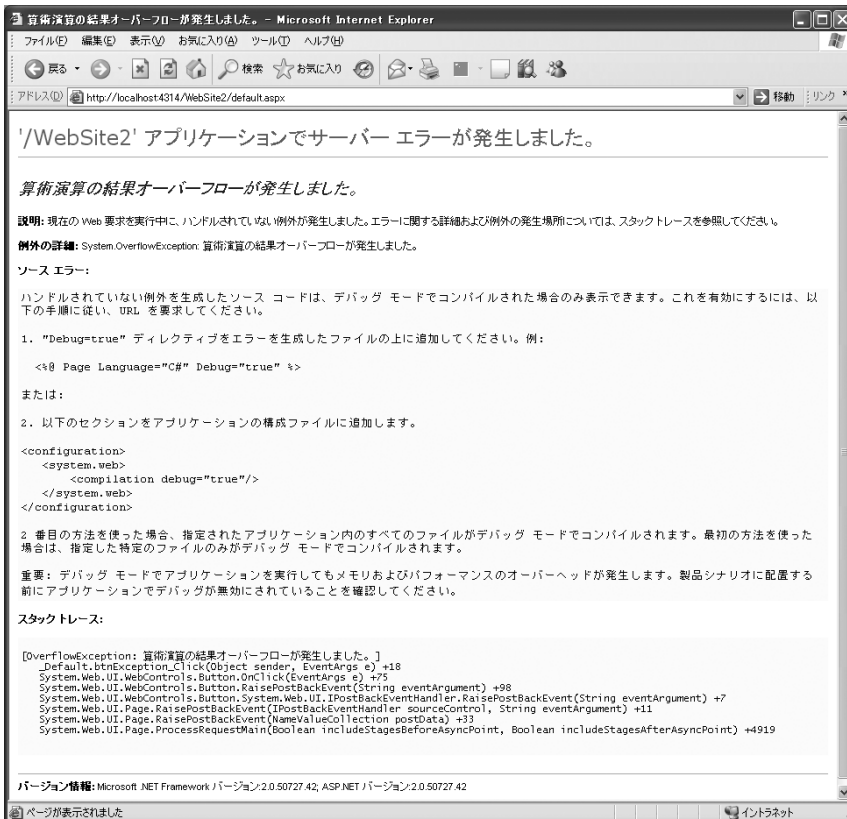


図3: ブラウザに例外が表示される



2) で [例外] ボタンを押して処理を進め例外を発生させる。そうするといつものように例外の説明を表示するページ (図3) へ移動し、処理が終了する (debugオプションがtrueの場合は図4のように表示される)。

● 別スレッドで実行する

ここで少しコードを変更してみよう。リスト2のように、別スレッドを起こしてその中で例外を発生させてみる。

先ほどと同様に [Ctrl] + [F5] キーで実行してみると、図1と同様にWebDev.WebServer.exeは正常に起動している。ここでWebページ上の [例外] ボタンを押して例外を起こすと、図5が表示される。この画面のいずれかのボ

タンを押して処理を進めると、図6のようにWebDev.WebServer.exeの処理が終了してしまう。

この挙動はASP.NET 2.0から採用されたポリシーである。ASP.NET 2.0では、以下の条件で例外が発生するとプロセスが終了してしまうのである。

条件 1 メインスレッド以外のスレッドで発生する

条件 2 アプリケーションに例外を処理する例外ハンドラ (Catchブロック) がなく、ASP.NETが提供するハンドラで処理される

この変更点にはさまざまな批判があるようで、MicrosoftはASP.NET 1.0や1.1のポリシーに戻す設定をサポートオンラインで紹介している。

• Unhandled exceptions cause ASP .NET-based applications to unexpectedly quit in the .NET Framework 2.0
<http://support.microsoft.com/kb/911816/en-us>

上記のサポートオンラインでは、「HT TPモジュールで発生した例外の情報をログに記録する方法」と「ASP.NET 1.0や1.1の設定に戻す方法」という2つの回避策を示している。

ここでは2つ目の過去のポリシーに戻す方法を実際に試してみよう。これはとても簡単だ。「%Windir%\Microsoft.NET\Framework\v2.0.50727」配下のAspnet.configファイルをメモ帳などで開き、legacyUnhandledException Policy enabledオプションをデフォルト