

# ASP.NET アプリケーションの ログ収集法

## TraceContextクラスやDiagnosticsクラスを使って

初音 玲

HATSUNE, Akira

### Technology Tools

- Visual Basic .NET
- Visual C# .NET
- SQL Server 2000
- Oracle 9i
- Access 2003
- ASP.NET
- Internet Information Services
- Other:

### Level



### Samples

・この記事で取り上げたソースコードおよびサンプルプログラムは、付録CD-ROMの¥DOTNET¥FEATURE01\_04ディレクトリに収録しています。

¥0403APP

アプリケーショントレース

¥0403EVENT

イベントログ

¥SETUP

セットアップ

¥WINEVENTLOG

イベントログ (Win)

・APPTTRACE.HTML

アプリケーショントレースの例

### はじめに

Visual Studio .NETのデバッグ機能を使ってASP.NET Webアプリケーションをデバッグする方法は、本特集記事「運用環境＝開発環境」で必ず動かすWebアプリケーション」と「リモートデバッグコンポーネントのノウハウ」で紹介した。しかし、ここにはいくつかの大きな問題がある。それは、

- ・ソリューション構成を「Debug」としてビルドしたASP.NET Webアプリケーションを前提としている
- ・ブレークポイントを作成してステップ実行でデバッグすることを前提としている

という点で、実は、ソリューション構成を「Debug」にしていると、デバッグ用の付加的な動作が含まれたクラスライブラリがビルド (図1) され、「Release」にしたときよりもパフォーマンスが低下するうえ、ステップ実行時には他のWebブラウザのリクエストも待たされ

てしまうのだ。

これでは、運用しつつ、バグの原因を特定するのは難しく、デバッグ中は該当するWebアプリケーションの使用を中断してもらえない。また、障害が発生した時点での状況についても、後から必ずしも十分な情報を得られるとは限らない。

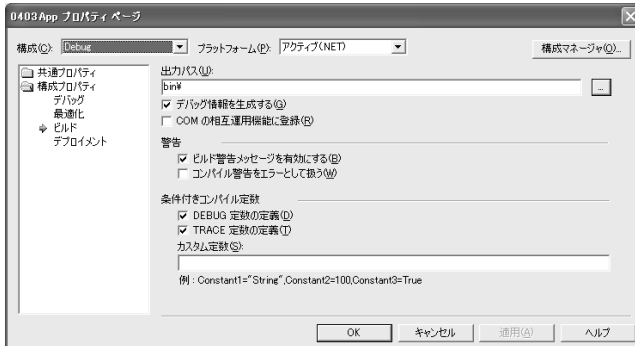
そこで、Webアプリケーションの動作記録 (ログ) をとることで、障害の再現に必要な情報収集やデバッグするための情報収集を行なえるように“あらかじめ”プログラミングしておく必要が生じる。

### アプリケーション トレースの設定

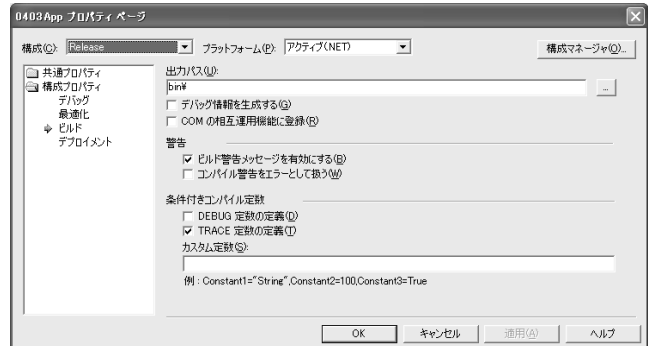
ASP.NET以前であれば、プログラムの動きを記録する場合、ログファイルとしてテキストを出力する機能をコーディングしていただろう。

ASP.NETでも、その方法は依然として有効ではあるが、ログの出力場所や検索性を考慮すれば、必ずしもベストな選択肢とは言えない。たとえば、ログ

図1：構成の違い



Debug構成の既定値



Release構成の既定値

ファイルのアクセス許可の設定を間違えたら、“ログに書かれた個人情報が流出する”などの最悪のケースも考えられる。それを回避するために、ログに書き出す内容を吟味するなど、本末転倒なことになってしまうので、ログファイルよりも、ASP.NETで追加されたトレース機能を使ってみることをお勧めする。

トレース機能を有効にする方法は簡単で、ASP.NET Webアプリケーションのプロジェクトに含まれているWeb.configファイルの<trace>要素のenabled属性を「True」に変更するだけだ。これだけで再ビルドを行なわなくてもトレース機能の有無を設定できる (図2)。

なお、このトレース機能の設定は、該

当するWebアプリケーションのフォルダ以降にのみ影響を与えるので、Webアプリケーションごとに決定できる。

## アプリケーション トレースの取得と表示

トレース機能が有効な状態で、Webアプリケーションを使用すると、ソースコードに対して何も変更を加えていなくても、その動作状況がトレースとして自動的に取得可能だ。

取得したトレースの表示には、トレースビューアを使う。トレースビューアの起動は、IISが稼動しているマシン上のWebブラウザで図3のように、WebアプリケーションのURLの最後にある0403app.aspxを「Trace.axd」に書き換えてアクセスすることで行なう。

なお、Web.configのlocalonly属性がTrueである場合は、IISが稼動しているマシン以外のWebブラウザでトレース

ビューア用のURLを入力しても、トレースビューアを使うことはできない。

## アプリケーショントレースの 内容

図3のトレースの一覧から詳細が知りたいWebブラウザからの要求を選択すると、要求の詳細が表示される。詳細部の構成は、

- ① 要求の詳細
- ② トレース情報
- ③ コントロールのツリー
- ④ クッキーコレクション
- ⑤ ヘッダーコレクション
- ⑥ フォームコレクション
- ⑦ サーバー変数

からなっている。

このなかでも、フォームコレクションは、画面上の値を参照できるので、どのような入力値が原因で障害が発生したのかなどを特定しやすい。

## TraceContext クラスの利用

ソースコードに何も手を加えなくて

図2：構成の違い

