



Sitecore CMS 6

プレゼンテーション コンポーネント トラブルシューティング

CMS 管理者と開発者のための問題解決のテクニック

目次

Chapter 1	イントロダクション.....	3
Chapter 2	Sitecore のトラブルシューティング	4
2.1	ブラウザの構成	5
2.2	パブリッシュ、セキュリティ、キャッシュ、言語	6
2.3	Sitecore のログ.....	7
2.3.1	パフォーマンスの悪い要求を見極める	7
2.4	IIS と ASP.NET のリセットと再起動	9
Chapter 3	プレゼンテーション コンポーネントのトラブルシューティング.....	10
3.1	パフォーマンスの悪いプレゼンテーション コンポーネントを見極める	11
3.2	Sitecore のブラウザベースのデバッガー	13
3.2.1	キャッシュをテストするためにレンダリング情報を無効にする.....	14
3.3	無効なプレースホルダー キーまたは無効なプレゼンテーション コンポーネントのネスト順.....	15
3.4	ブラウザのツールバー	16
Chapter 4	XSL レンダリングのトラブルシューティング.....	17
4.1	Xsl ファイルが処理されない.....	18
4.2	例外オブジェクトが合致するアイテム メソッドを含んでいない.....	19
4.3	コンテキスト エlement と特定のアイテム	20
4.4	sc:trace() XSL 拡張メソッドを使ってトレース メッセージを書く.....	21
4.5	属性値テンプレート	22
4.6	生の XML を調べる	23
Chapter 5	既知の問題とさらなるリソース	24
5.1	Visual Studio のデバッガーが応答しなくなる	25
5.2	コントロールにコード ブロックが含まれているためコントロールのコレクションを変更できない.....	26
5.3	Sitecore.Exceptions.CyclicSublayoutException	27
5.4	ページ エディターで編集コントロールが表示されない	28
5.5	トラブルシューティングのためのさらなるリソース.....	29

Chapter 1

イントロダクション

このプレゼンテーション コンポーネント トラブルシューティング ガイドは、CMS 管理者と開発者がプレゼンテーション コンポーネントに関してよくある問題を診断し、問題を解決するためのテクニックを提供します。これにはパフォーマンスの最適化に関する一般的なトラブル解決手順を含みます。

この文書には次の章があります。

- Chapter 1 — イントロダクション
- Chapter 2 — Sitecore のトラブルシューティング
- Chapter 3 — プレゼンテーション コンポーネントのトラブルシューティング
- Chapter 4 — XSL レンダリングのトラブルシューティング
- Chapter 5 — 既知の問題とさらなるリソース

Chapter 2

Sitecore のトラブルシューティング

この章では Sitecore の一般的なトラブルシューティングに関する情報を説明します。

この章には次のセクションがあります。

- ブラウザーの構成
- パブリッシュ、セキュリティ、キャッシュ、言語
- Sitecore のログ
- IIS と ASP.NET のリセットと再起動

2.1 ブラウザーの構成

Sitecore のコンテンツ マネジメント ユーザー インターフェースで問題が発生したら、常にブラウザーの構成を考察します。¹

ブラウザーの構成が問題の原因であるかどうかを見極めるには同じ操作を別のクライアントで行ってみます。

¹ ブラウザーの構成とトラブルシューティングに関する詳細は <http://sdn.sitecore.net/Articles/Administration/Configuring%20IE7.aspx> を参照してください。

2.2 パブリッシュ、セキュリティ、キャッシュ、言語

コードが実行される環境を常に考察します。マスター データベースとパブリッシュのターゲット データベースの相違の可能性も考察します。² プレゼンテーション コンポーネントが意図している Sitecore データベースにアクセスしていることを確認します。必要な情報がそのデータベースにパブリッシュされていることを確認します。コンテンツ マネジメント環境では問題なくて、パブリッシュされた Web サイトで問題が発生する場合には、マスター データベースで Sitecore がパブリッシュしなかった新しいアイテムまたは変更点に依存している機能に問題がある可能性があります。コンテンツ アイテムをパブリッシュするより前かまたは同時に、必ず変更点をコンテンツ アイテムに参照されているデータ テンプレートと新しいメディアにパブリッシュします。ワークフローとパブリッシュの制約のため、パブリッシュはアイテムとバージョン データは移動しないことに留意します。

必要なアイテムをパブリッシュしてもなお CMS とパブリッシュ サイトに相違がある場合は、セキュリティが問題であるかも知れません。デフォルトではパブリッシュ サイトへのアクセスは Extranet ドメインの Anonymous ユーザーとして行われます。CMS へのアクセスは Sitecore ドメインの特定のユーザーとして行われますので、Extranet ドメインの Anonymous ユーザーとはアクセス権が異なる場合があります。アクセス ビューアーを使って Extranet ドメインの Anonymous ユーザーのアクセス権を調べることができます。

適切に構成されている場合、適切に行われたすべてのパブリッシュでは Sitecore のキャッシュをクリアします。ASP.NET のアウトプット キャッシュまたはカスタム キャッシュを使用している場合は、自分でパブリッシュ イベントハンドラーを実装し、キャッシュをクリアする Web サービスまたはその他のソリューションを起動する必要があります。

アイテムは言語を含み、言語はバージョンを含みます。アイテムはすべての言語に対して存在し、最初のバージョンはユーザーがアイテムを作成した言語に対して存在しますが、アイテムはバージョンが存在しない言語に対してはフィールド値を持ちません。Sitecore は要求された言語にデータが存在しない場合にデフォルトの言語に戻すことはしません。ページが空白に見えたら、Sitecore の言語構成を確認し、ブラウザーがバージョン データの存在しない言語のアイテムを要求していないかどうか確認します。

² パブリッシュとパブリッシュのトラブルシューティングに関する詳細は

<http://sdn.sitecore.net/Articles/Administration/Sitecore%20Publishing%20Operations.aspx> および

<http://sdn.sitecore.net/End%20User/Site%20Administration/Troubleshooting/Missing%20Content.aspx> を参照してください。

2.3 Sitecore のログ

予期せぬ挙動が発生した時間の前後で Sitecore のログ エントリーにエラーや警告がないかどうかを、常に確認します。Sitecore はログ ファイルを毎日一つ生成し、さらに ASP.NET の再起動がファイル ロックをすぐに開放しないときにロギングを継続するために必要な場合には追加のログ ファイルを作成します。

web.config の /configuration/sitecore/settings/setting エLEMENT の name LogFolder の value 属性が Sitecore ログを含むディレクトリを指定します。この設定は通常は web.config の /configuration/sitecore/sc:variable エLEMENT の name dataFolder の value 属性で定義された \$(dataFolder) 変数を参照します。次の設定では Sitecore はログファイルを C:\inetpub\Sitecore\mywebsite\data\logs ディレクトリーに書き出します：

```
<sc.variable name="dataFolder" value="C:\inetpub\Sitecore\mywebsite\data\" />
...
<setting name="LogFolder" value="$(dataFolder)/logs" />
```

2.3.1 パフォーマンスの悪い要求を見極める

Sitecore ログを使ってパフォーマンスの悪いプレゼンテーション コンポーネントを参照しているレイアウト詳細を含むアイテムを見極めます。許容される処理時間を超えているページに対して Sitecore は下記のような警告をログファイルに書き出します：

WARN (警告)

Timing threshold exceeded for web page. (Web ページのタイミングしきい値超過)

Milliseconds (ミリセカンド)

Threshold (しきい値)

Page URL (ページ URL)

さらに Sitecore は許容されている回数を超えたアクセス要求に警告を出します。一つのコンポーネントによるアクセスの回数はそのパフォーマンスに影響を及ぼすためです。

WARN (警告)

Item threshold exceeded for web page (Web ページのアイテムしきい値超過)

Items accessed (アクセスされたアイテム)

Threshold (しきい値)

Page URL (ページ URL)

さらに Sitecore は許容されている量を超えてメモリを消費している要求に警告を出します。

WARN (警告)

Memory threshold exceeded for web page (Web ページのメモリしきい値超過)**Memory used (使用メモリ)****Threshold (しきい値)****Page URL (ページ URL)**

これらの警告を使って、リソースを異常に消費する要求を見極めます。Sitecore のブラウザーベースのデバッガーを使い、ログ エントリで指定された URL に行き、多くのアイテムにアクセスしているコンポーネント、多くの処理時間を使っているコンポーネント、過度のメモリを消費しているコンポーネントを見極めます。Sitecore デバッガーに関する詳細は「Sitecore のブラウザーベースのデバッガー」のセクションを参照してください。

過度のメモリまたは処理時間を要しているコンポーネントには、コーディング上の問題を調査し、またアウトプット キャッシュの構成を確認します。多くのアイテムにアクセスしているコンポーネントには、情報アーキテクチャの変更を検討します。

web.config の httpRequestEnd パイプラインの StopMeasurements プロセッサのプロパティを調整して警告しきい値を構成することができます。

```
<processor type="Sitecore.Pipelines.HttpRequest.StopMeasurements, Sitecore.Kernel">  
  <TimingThreshold desc="Milliseconds">1000</TimingThreshold>  
  <ItemThreshold desc="Item count">1000</ItemThreshold>  
  <MemoryThreshold desc="KB">10000</MemoryThreshold>  
</processor>
```

メモ

ASP.NET が再起動したあとの最初の要求で許容しきい値を超過するのは正常です。ASP.NET の再起動の直後のしきい値警告は無視してください。Sitecore ログの Sitecore started INFO エントリは ASP.NET の再起動を示しています。

メモ

Sitecore ユーザー インターフェイス コンポーネントへの要求がパブリッシュされたサイトのページへの要求より多くのリソースを消費するのは正常です。一般に、Sitecore ユーザー インターフェイス ページへの警告は無視できます。またはコンテンツ マネジメント環境にはコンテンツ デリバリー環境と異なるしきい値を構成します。

2.4 IIS と ASP.NET のリセットと再起動

次のアクションを取る前に真の原因を見極めるために問題を調査することはいつでも望ましいことですが、IIS と ASP.NET のリセットまたは再起動は Sitecore のインスタンスをただちに稼働状態に復帰させる最も早い方法です。

IIS をリセットするには Windows の管理者としてコマンド プロンプトを開き、次のコマンドを実行します：

```
iisreset
```

IIS の再起動は IIS のリセットより多くの問題を解決できる場合があります。IIS を再起動するには Windows のローカルの管理者としてコマンドプロンプトを開き、次のコマンドを実行します：

```
net stop w3svc  
net start w3svc
```

Chapter 3

プレゼンテーション コンポーネントのトラブルシューティング

この章では Sitecore プレゼンテーション コンポーネントに特有なトラブルシューティングの方法を説明します。

この章には次のセクションがあります。

- パフォーマンスの悪いプレゼンテーション コンポーネントを見極める
- Sitecore のブラウザーベースのデバッガー
- 無効なブレースホルダー キーまたは無効なプレゼンテーション コンポーネントのネスト順
- ブラウザーのツールバー

3.1 パフォーマンスの悪いプレゼンテーション コンポーネントを見極める

「Sitecore のログ」のセクションで説明した Sitecore ログを使う方法に加え、プレゼンテーション コンポーネント統計情報ページを使用してパフォーマンスの悪いプレゼンテーション コンポーネントを見極めることができます。

Sitecore のインストール プログラムはプレゼンテーション コンポーネント統計情報ページを含む `/sitecore/admin` ディレクトリへの匿名アクセスを不許可とするように IIS を構成します。プレゼンテーション コンポーネント統計情報ページにアクセスするには、ブラウザーに Windows の認証情報を入力する必要があります。または IIS でこのディレクトリへの匿名アクセスを有効にすることもできます。IP または IIS の別の制限によってこのディレクトリのページへの許可されないアクセスを防ぐことができます。

IIS 6 で `/sitecore/admin` ディレクトリへの匿名アクセスを有効にする方法:

1. IIS 管理コンソールで `/sitecore/admin` ディレクトリを右クリックし、[プロパティ] をクリックします。
2. [ディレクトリ セキュリティ] タブをクリックします。
3. [認証とアクセス制御] のセクションで編集をクリックします。
4. [匿名アクセス] にチェックを入れて有効にし、OK をクリックします。

IIS 7 で `/sitecore/admin` ディレクトリへの匿名アクセスを有効にする方法:

1. IIS 管理コンソールで `/sitecore/admin` ディレクトリをクリックし、[認証] をダブルクリックします。
2. [匿名認証] をクリックし、[有効にする] をクリックします。

プレゼンテーション コンポーネント統計情報ページにアクセスするには、ブラウザーで Sitecore サーバーの `/sitecore/admin/stats.aspx` にアクセスします。レポートの各行は特定の論理サイトのレンダリング、プレースホルダー、サブレイアウトを表しています（各サイトはそれぞれのキャッシュを持ちます）。列ではコンポーネントのすべての起動（ASP.NET の最後のリセットまたは再起動からの相対時間）に基づく統計情報を示しています。次の表ではプレゼンテーション コンポーネント統計情報レポートの値を説明しています。

列	値
Rendering	個々のプレゼンテーション コンポーネントを識別する
Site	論理サイト
Count	レイアウト エンジンがコンポーネントを使用した回数（それを起動したか、またはキャッシュから出力を読み出した）
From Cache	レイアウト エンジンがコンポーネントの出力をキャッシュから読み出した（起動するのではなく）回数

列	値
Average Time (ms)	コンポーネントの起動あたりの平均処理時間
Average Items	コンポーネントの起動あたり平均アイテム読み込み数
Max Time	コンポーネントの一度の起動による最大処理時間
Max Items	コンポーネントの一度の起動による最大のアイテム読み込み数
Total Time	コンポーネントのすべての起動の合計処理時間
Total Items	コンポーネントのすべての起動のアイテム読み込み合計数
Last Run	最後のコンポーネントの起動日時

パフォーマンスの悪いコンポーネントを見極めるためには合計処理時間、最大アイテム読み込み数、平均処理時間、最大処理時間、の高い値のものを見つけます。パフォーマンスの悪いコンポーネントを見つけたら、`sc:trace()` XSL 拡張メソッドを使って XSL レンダリングのエラーの場所を見つけます。または Visual Studio を使用して .NET コードをデバッグします。³

`sc:trace()` XSL 拡張メソッドの詳細については「`sc:trace()` XSL 拡張メソッドを使ってトレースメッセージを書く」のセクションを参照してください。

最適にキャッシュ構成されていないコンポーネントを見つけるには、ユーザー数カウントがキャッシュからのキャッシュヒット数に対して高い値となっているコンポーネントを見つけます。

ヒント

いろいろな列の値によってソートするには、Internet Explorer で表で右クリックし、[Microsoft Excel にエクスポート] を選択します。

³ Visual Studio のデバッガーに関する詳細については <http://sdn5.sitecore.net/Reference/Sitecore%20.aspx> から『プレゼンテーション コンポーネント クックブック』を参照してください。

3.2 Sitecore のブラウザーベースのデバッガー

Sitecore のブラウザーベースのデバッガーを使用してプレゼンテーション コンポーネントの問題を調査します。デバッガーはあなたがサイトをブラウズするときにページにデバッグ ユーザー インターフェイス エlementを重ね合わせます。これにはパフォーマンスの悪いコンポーネントを見つけるのに役立つページ プロファイルとトレース機能を含みます。

ページ プロファイルはコンポーネントのパフォーマンスの全体的なサマリーを示します。⁴ ページ トレースはページ アセンブリー処理の各ステップの詳細を提供します。ページ トレースはパブリッシュされたサイトには表示されないエラー メッセージを含みます。たとえばプレゼンテーション コンポーネントを存在しないプレースホルダーにバインドしようとしたときなどです。

Sitecore のブラウザーベースのデバッガーにデベロッパー センターからアクセスするには、[デバッグ] メニューをクリックし、[デバックを開始] をクリックします。Sitecore デバッガーに Sitecore デスクトップからアクセスするには、[Sitecore] ボタンをクリックし、[デバッグ] をクリックします。デバッガーは新しいブラウザーのウィンドウまたはタブで開きます。

デバッガーでは：

- ページの左上の隅の [リボン] をクリックし、リボンを表示または非表示にします。
- [ブラウザ表示] グループでページ エディター、プレビュー、デバッグ モードから選択します。
- [プロファイル] グループで、[アクティベート] コマンドを押してデバッガーがページ プロファイル（ページの全体的なパフォーマンスの概要）を含むかどうかを制御します。
- [トレース] グループで、[アクティベート] コマンドを押してデバッガーがページ トレース（ページ アセンブリー処理の詳細のサマリー）を含むかどうかを制御します。
- [レンダリング] グループで、[レンダリング情報] チェックボックスを選択し、デバッガーがレンダリング情報を含むかどうかを制御します。レンダリング情報は各プレゼンテーション コンポーネントに関する情報をデバッガー上で緑の三角を使って表示します。
- 各プレゼンテーション コンポーネントに見える [レンダリング情報] アイコンの上にカーソルをホバーします。表示されるパネルで、[タブ] をクリックしてプレゼンテーション コンポーネントに関する情報を参照します。[詳細] タブはプレゼンテーション コンポーネントに関する全体的な情報を提供します。（Web コントロールの名前や XSL ファイルのパスなど） [プロファイル] タブはプレゼンテーション コンポーネントのパフォーマンス情報を提供します。[キャッシュ設定] タブはプレゼンテーション コンポーネントのキャッシュ構成を示しています。[アウトプット] タブは個々のプレゼンテーション コンポーネントによって生成されたアウトプットを含んでいます。
- 分析を要するページ、つまり望ましいパフォーマンスを出せていないページなどに行きます。レンダリング情報、ページ プロファイル、ページ トレースなどを調査し、問題の原因を見極めます。

ページ プロファイル メッセージは次の情報を含んでいます：

- Time: 個々のタスクで全ページのレンダリングにかかった時間のパーセンテージ

⁴ Sitecore パフォーマンスの最適化については <http://sdn.sitecore.net/Articles/Administration/Sitecore%20Performance.aspx> を参照してください。

- Action: 行われた処理ステップの説明
- Total: タスクの総実行時間（ミリ秒）
- Own: 子孫のプレゼンテーション コンポーネントを除いた、タスクの実行時間（ミリ秒）
- Items Read: タスク中に読み込まれた Sitecore アイテムの数。パフォーマンスに影響を与えます。

重要

Sitecore のブラウザーベースのデバッガーはデフォルトのパブリッシュされたサイトにアクセスします。それはデフォルトでは Web データベースを使います。他のデータベースにアクセスするには、`sc_databaseURL` クエリ文字列パラメータを使用してデータベース名を指定します。（たとえば、`sc_database=master`）

3.2.1 キャッシュをテストするためにレンダリング情報を無効にする

Sitecore デバッガーがレンダリング情報を表示するときは、レイアウト エンジンがコンポーネントのキャッシュ基準を無視します。この場合、レイアウト エンジンが各プレゼンテーション コンポーネントを各ページ要求に対して起動します。この場合、キャッシュから出力を読み込みません。

キャッシュ構成を調査するには、レンダリング情報を無効にし、ページ トレースを調査します。レンダリング情報を無効にするには、[リボン] を表示し、[レンダリング] グループで、[レンダリング情報] のチェックボックスのチェックを外します。

using cache メッセージはレイアウト エンジンが XSL レンダリングのアウトプットをキャッシュから読み出したことを示しています。

メモ

ページトレースの中の `Xslt file loaded from cache` などのメッセージはシステムが XSL 変換オブジェクトをキャッシュから読み出した（XSL レンダリングのアウトプットではなく）ことを示しています。

3.3 無効なプレースホルダー キーまたは無効なプレゼンテーション コンポーネントの ネスト順

レイアウト エンジンがコントロールをプレースホルダーにバインドすべきであるとレイアウト詳細が指示するが、レイアウト エンジンがすでに処理されたレイアウトとサブレイアウトの中にそのキーではプレースホルダーを見つけられない場合、レイアウト エンジンがブラウザで見えるエラーを作成しません。そのかわりに、レイアウト エンジンは次のような警告メッセージを Sitecore のブラウザベースのデバッガーで見ることのできるページ トレースに書き出します。

Warning (警告)

A rendering was not used on the page (ページでレンダリングは使用されませんでした)

The placeholder was not found (プレースホルダーはみつかりませんでした)

ページ トレースの中にこれらの警告メッセージがあった場合、アイテムのレイアウト詳細のサブレイアウトとレンダリングが、指定されたレイアウトとサブレイアウトに存在するプレースホルダー キーを指定していることを確認します。プレースホルダー設定がコンポーネントのネスト順にコントロールを記載していることを確認します。たとえば、プレースホルダーを含むサブレイアウトはそのプレースホルダーにバインドするコントロールよりも前に記載される必要があります。完全修飾のプレースホルダー キーが正しいプレースホルダーのネスト順を反映していることを確認します。

3.4 ブラウザーのツールバー

Internet Explorer の Developer Toolbar や HttpWatch や Firebug (Firefox 用) などのブラウザーのツールバーを使用して、さまざまなプレゼンテーションの問題を診断することができます。⁵

メモ

ブラウザーのツールバーの中には Windows のポップアップをブロックするなど、Sitecore ソリューションと干渉するものもあります。ブラウザー ツールバーを無効にするか Sitecore と適切に共存するように構成する必要がある場合があります。ブラウザーのツールバーに関連して問題が発生した可能性のある場合、問題を調査する前に、ツールバーをインストールしていないクライアントで問題を再現してみてください。

⁵ Internet Explorer Developer Toolbar については <http://www.microsoft.com/downloads/details.aspx?familyid=e59c3964-672d-4511-bb3e-2d5e1db91038&displaylang=en> を参照してください。HttpWatch については <http://httpwatch.com> を参照してください。Firebug については <https://addons.mozilla.org/en-US/firefox/addon/1843> を参照してください。

Chapter 4

XSL レンダリングのトラブルシューティング

この章では XSL レンダリングのトラブルシューティングの手順を説明します。

この章には次のセクションがあります。

- Xsl ファイルが処理されない
- 例外オブジェクトが合致するアイテム メソッドを含んでいない
- コンテキスト エlementと特定のアイテム
- `sc:trace()` XSL 拡張メソッドを使ってトレースメッセージを書く
- 属性値テンプレート
- 生の XML を調べる

4.1 Xsl ファイルが処理されない

XSL レンダリングが処理される間に、レイアウト エンジンに XSL コードで構文エラーが発生した場合、または .NET 拡張で例外が発生した場合、レイアウト エンジンは警告アイコンと次のメッセージを出力します。

Xsl file could not be processed (Xsl ファイルが処理されません)

[警告] アイコンをクリックしてエラーの詳細を調査します。最初の数行はエラー メッセージとソースコード中でエラーの可能性のある場所を含んでいますので、これを問題の診断に役立てます。残りの行はエラーの箇所の .NET スタック トレースを含んでいます。

メモ

Sitecore ログにも例外の詳細が含まれています。

4.2 例外オブジェクトが合致するアイテム メソッドを含んでいない

`sc:item()` XSL 拡張メソッドに第 2 の必須パラメータを渡さなかった場合、ブラウザーで下記のようなエラーが発生します：

Extension object http://www.sitecore.net/sc does not contain a matching 'item' method that has 1 parameter(s) (拡張オブジェクト `http://www.sitecore.net/sc` は 1 つのパラメータを持つ合致するアイテム メソッドを含んでいません。)

この問題を解決するには、`$sc_currentitem` か、またはアイテムを `sc:item()` XSL 拡張メソッドへの第 2 パラメータとして表している他のエレメントを渡します。

たとえば、次のコードは `sc:item()` XSL 拡張メソッドへの第 2 パラメータを含んでいません。

```
<xsl:variable name="content" select="sc:item('/sitecore/content') " />
```

このコードの一つの修正方法の例は下記です：

```
<xsl:variable name="content" select="sc:item('/sitecore/content',$sc_currentitem) " />
```

メモ

渡される特定のアイテムは任意です。

4.3 コンテキスト エLEMENTと特定のアイテム

XSL コンストラクトが処理するコンテキストを常に考察します。または `select` 属性を明示的に使っている特定のELEMENTを明示的にリファレンスします。そうしない場合、XSL コンストラクトは正しくない出力を生成するか、出力を生成しない場合があります。

たとえば、開発者は次のようなコードを使ってマルチリストのフィールドからリファレンスされるアイテムを処理する可能性があります：

```
<xsl:for-each select="sc:Split('FieldName',$sc_currentitem)">
  <sc:link>
    <sc:text field="FieldName" />
  </sc:link>
</xsl:for-each>
```

この例の`<sc:link>`と`<sc:text>` XSL 拡張コントロールでは、開発者はコンテキスト アイテムを設定しなかったか、またはリンクして `select` 属性を使ってフィールド値を取り出すアイテムを指定していません。コントロールはコンテキスト ELEMENTの処理をデフォルトにします。for-each により、`<xsl:for-each>`の中で、コンテキスト ELEMENTは `sc:Split()` XSL 拡張メソッドによって返されるXML 構造体のメンバーであり、Sitecore データベースを表すXML ドキュメントの一部ではありません。

問題を修正するには、`select` 属性を使ってアイテムを指定するか、そのアイテムへのコンテキスト ELEMENTを設定します。アイテムを `select` 属性を使用して明示的に指定する方法：

```
<xsl:for-each select="sc:Split('FieldName',$sc_currentitem)">
  <xsl:variable name="VariableName" select="sc:item(text(),$sc_currentitem)" />
  <sc:link select="$VariableName">
    <sc:text select="$VariableName" field="FieldName" />
  </sc:link>
</xsl:for-each>
```

コンテキスト アイテムを設定する方法：

```
<xsl:for-each select="sc:Split('FieldName',$sc_currentitem)//value">
  <xsl:for-each select="sc:item(text(),$sc_currentitem)">
    <sc:link><sc:text field="FieldName" /></sc:link>
  </xsl:for-each>
</xsl:for-each>
```

4.4 sc:trace() XSL 拡張メソッドを使ってトレースメッセージを書く

sc:trace() XSL 拡張メソッドを使って Sitecore のブラウザベースのデバッガーで見ることのできるページ トレースへのメッセージを書きます。たとえば：

```
<xsl:value-of select="sc:trace('Message')" />
```

<xsl:value-of>をこのように使って sc:trace() XSL 拡張メソッドを呼び出しますが、出力は生成しません。

メモ

カスタムの .NET XSL 拡張を除き、Microsoft Visual Studio のデバッガーは XSL コードのステップ スルーをサポートしません。

4.5 属性値テンプレート

XSL 変換エンジンが `xsl` ネームスペースの要素 (たとえば `<xsl:value-of>`) を評価するとき、それは自動的に属性値の XPath 表現を評価します。たとえば、次のコードでは、XSL 変換はリテラル値 `sc:path(.)` を出力しませんが、その表現を XPath ステートメントとして評価します。この場合は XSL 拡張メソッドへの呼び出しです。

```
<xsl:value-of select="sc:path(.)"/>
```

XSL 変換エンジンは他の要素の属性値の XPath 表現を自動的に評価しません。ある場合にはこれは XSL 変換に XPath 表現を出力させる (対応する値の出力や、出力を生成しないのでなく) こととなります。たとえば、次のコードでは、`<a>` 要素が認識されたネームスペース (たとえば `xsl` や `sc`) でないため、XSL 変換エンジンは、XSL 拡張関数 `sc:path()` を起動して `href` 属性のための適当な値を決めるのでなく、リテラル値 `` を出力します。

```
<a href="sc:path(.)"/>
```

XSL の仕様では、XSL 変換エンジンに認識されないコンストラクトでの属性値を XPath ステートメントとして評価させる属性値テンプレートを定義しています。⁶ 中かっこ ("`{ }`") は属性値テンプレートを示し、XPath ステートメントが非 XSL コンストラクトの属性値の中で評価される場合に使用される必要があります。

XSL 変換エンジンに `sc:path(.)` XSL 拡張メソッドを起動させて `<a>` 要素の `href` 属性へ出力する値を決定するには、XSL 拡張メソッドを中かっこで囲みます :

```
<a href="{sc:path(.)}"/>
```

単体の中かっこが XSL 変換エンジンにリテラルの属性値を属性値テンプレートとして解釈させてしまう場合には、二重の中かっこ ("`{{ }}`") を使って単体の中かっこを表すことができます。たとえば、次のコードでは、XSL 変換エンジンは `` を出力します :

```
<a href="{{strangelink}" />
```

`<xsl:element>` と `<xsl:attribute>` XSL 要素を使って属性値テンプレートを避けることができます。

```
<xsl:element name="a">
  <xsl:attribute name="href"><xsl:value-of select="sc:path( .)" /></xsl:attribute>
</xsl:element>
```

または :

```
<a>
  <xsl:attribute name="href"><xsl:value-of select="sc:path( .)" /></xsl:attribute>
</a>
```

XSL 拡張コントロール属性値の属性値テンプレートを使用することもできます。たとえば、`ImageField` という名前のフィールドで参照されたメディア アイテムの名前をコンテキスト アイテムでそのメディア アイテムのための `` 要素の `title` 属性として使うためには :

```
<sc:image title="{sc:item(sc:fld('ImageField',..,'mediaid'),..)/@name}"
  field="ImageField" />
```

⁶ 属性値テンプレートの詳細については <http://www.w3.org/TR/xslt#attribute-value-templates> の「セクション 7.6.2」を参照してください。

4.6 生の XML を調べる

XSL レンダリングに使える生の XML を調査したい場合には下記を参照してください。

XSL レンダリングの `<xsl:copy-of>` XSL エレメントを使って、XML 構造体のコンテンツを出力することができます。たとえば、`<xsl:copy-of>` を使ってコンテキストアイテムを調べることができます。

```
<xsl:copy-of select="$sc_currentitem" />
```

メモ

`<xsl:copy-of>` エレメントは選択されたアイテムのすべての子孫を含みます。多くの子孫をもつアイテムでは、これはかなりの処理リソースを消費し、大量のアウトプットを生成します。

`<xsl:copy-of>` XSL エレメントを使って、任意の XML 構造体のコンテンツを出力できます。たとえば、`sc:Split()` XSL 拡張メソッドに返されるデータは：

```
<xsl:variable name="split" select="sc:Split('FieldName',$sc_currentitem)" />
<xsl:copy-of select="$split" />
```

メモ

次のコードは変数を作成することなく同じ目的を達成します：

```
<xsl:copy-of select=" sc:Split('FieldName',$sc_currentitem)" />
```

この出力を参照するには、レンダリングをデベロッパー センターで開き、[プレビュー] ウィンドウで右クリックし、[ソース参照] を選択します。または、レンダリングを使っているページをブラウザで開き、そのソースを参照します。

または、レイアウトを使って XSL レンダリングに利用できるフォーマットで XML を返すこともできます。たとえば、クエリ文字列パラメータ `x=1` をトリガーとし、アイコン `Control/32x32/treeview.png` を使う、XML という名前のデバイスを作成します。

次のコードを含むレイアウトを作成します：

```
<%@ Page Language="c#" Inherits="System.Web.UI.Page" CodePage="65001"
  ContentType="text/xml" %>

<script runat="server">
protected override void Render(System.Web.UI.HtmlTextWriter writer)
{
  Sitecore.Xml.XPath.ItemNavigator navigator =
    Sitecore.Configuration.Factory.CreateItemNavigator(Sitecore.Context.Item);
  writer.Write(navigator.OuterXml);
}
</script>
```

調査するアイテムのレイアウト詳細で、このレイアウトを XML デバイスに関連付け、アイテムにブラウザでアクセスし、`x=1` クエリ文字列パラメータをブラウザのアドレスバーで追加します。

Chapter 5

既知の問題とさらなるリソース

この章ではプレゼンテーションコンポーネントのトラブルシューティングに関連した既知の問題を説明します。

この章には次のセクションがあります。

- Visual Studio のデバッガーが応答しなくなる
- コントロールにコード ブロックが含まれているためコントロールのコレクションを変更できない
- Sitecore.Exceptions.CyclicSublayoutException
- ページ エディターで編集コントロールが表示されない
- トラブルシューティングのためのさらなるリソース

5.1 Visual Studio のデバッガーが応答しなくなる

デバッグするため既存のプロセスにアタッチしようとしたときに Visual Studio が応答しなくなることがあります。この問題を解消するには :

1. Visual Studio を再起動します。
2. Web アプリケーション プロジェクトを開きます。
3. メニューから [表示] をクリックし、ソリューション エクスプローラーをクリックします。
4. Visual Studio のソリューションエクスプローラーで、プロジェクトに含まれないファイルをすべて非表示にします。⁷
5. Visual Studio のデバッガーを ASP.NET のワーカー プロセスにアタッチします。

⁷ Visual Studio のソリューションエクスプローラーでプロジェクトに含まれないファイルをすべて非表示にするには、<http://sdn5.sitecore.net/Reference/Sitecore%206.aspx> から『プレゼンテーション コンポーネント クックブック』を参照してください。

5.2 コントロールにコードブロックが含まれているためコントロールのコレクションを変更できない

レイアウトまたはサブレイアウトが ASP.NET コード レンダリング ブロックを含む場合に、ブラウザで次のメッセージが表示されることがあります。

The Controls collection cannot be modified because the control contains code blocks (コントロールにコードブロックが含まれているため、コントロールのコレクションを変更できません)

一般にコード レンダリング ブロックは避けるようにします。

コード レンダリング ブロックが必要な場合にはサーバー コントロールの中でのみ使用するようにします。たとえば、次のコードはコード レンダリング ブロックを含みます。(<%= と %> の間のコンテンツ) :

```
<script language="javascript">
    alert("<%= myObject.ClientID %>");
</script>
```

このコードを修正する唯一の方法は、runat 属性の値を server とした <div> エlementなどのサーバー コントロールの中に配置することです。

```
<div runat="server">
    <script language="javascript">
        alert("<%=myObject.ClientID%>");
    </script>
</div>
```

5.3 Sitecore.Exceptions.CyclicSublayoutException

レイアウト詳細または静的バインドによってサブレイアウトのそれ自身へのバインドが生じた場合、無限再帰となり Sitecore は `Sitecore.Exceptions.CyclicSublayoutException` を出します。エラーメッセージには関連するコンポーネントが示されます。

Exception (例外)

Sitecore.Exceptions.CyclicSublayoutException

Server Error in / Application.

A sublayout has been recursively embedded within itself (サブレイアウトがそれ自身に再帰的に組み込まれました)

静的にも、レイアウト詳細の使用でも、サブレイアウトをそれ自身にバインドしないでください。

5.4 ページ エディターで編集コントロールが表示されない

レイアウトが ASP.NET ルートの `<form>` エlementを含んでいない場合、ページ エディターが編集コントロールを出力しないことがあります。この問題を解消するには、レイアウトに `<form>` エlementを追加します。

Sitecore が新規のレイアウトに含めるデフォルトの `<form>` エlementは次のとおりです：

```
<form method="post" runat="server" id="mainform">Insert your controls here.</form>
```

最初の `<form>` エlementはレイアウトの最初の `<body>` エlementの直後に置かれ、最後の `</form>` エlementは通常は最後の `</body>` エlementの直前に置かれます。この `<form>` エlementは他のマークアップとプレゼンテーションコントロールのすべてを囲みます。

5.5 トラブルシューティングのためのさらなるリソース

トラブルシューティングのためのさらなるリソースとしては下記を参照してください：

- Sitecore デベロッパー ネットワーク (<http://sdn.sitecore.net>)
- Sitecore デベロッパー ネットワークのフォーラム (<http://sdn.sitecore.net/Forum.aspx>)
- Sitecore デベロッパー ネットワークの Internet Explorer の構成とトラブルシューティングのリソース (<http://sdn.sitecore.net/Articles/Administration/Configuring%20IE7.aspx>)
- Sitecore デベロッパー ネットワークの製品インストールに関するドキュメント (<http://sdn.sitecore.net/Products/Sitecore%20V5/Sitecore%20CMS%206/Installation.aspx>) およびオプションのモジュールに関するドキュメント (<http://sdn.sitecore.net/Products.aspx>)
- Sitecore デベロッパー ネットワークの Sitecore の製品インストールのトラブルシューティングガイド (<http://sdn.sitecore.net/Products/Sitecore%20V5/Sitecore%20CMS%206/Installation%20Troubleshooting.aspx>)
- Sitecore デベロッパー ネットワークの一般のトラブルシューティングに関するリソース (<http://sdn.sitecore.net/Articles/Troubleshooting/Troubleshooting%20Sitecore%205,-d-,3.aspx>)
- Sitecore デベロッパー ネットワークのトラブルシューティング FAQ (<http://sdn.sitecore.net/FAQ/Troubleshooting.aspx>)
- Sitecore デベロッパー ネットワークのサイト管理者のトラブルシューティングに関するリソース (<http://sdn.sitecore.net/End%20User/Site%20Administration/Troubleshooting.aspx>).
- Sitecore デベロッパー ネットワークのパフォーマンス トラブルシューティングに関するリソース (<http://sdn.sitecore.net/Articles/Administration/Sitecore%20Performance/Miscellaneous/Troubleshooting.aspx>)
- 各国各地域での Sitecore の問い合わせ窓口 (<http://sitecore.net/contact.aspx>)
- Sitecore サポート ポータル (<http://support.sitecore.net>)