

Unicaページタグ付け
スクリプトのインストールおよび
設定

Copyright

© Copyright, Unica Corporation 2001- 2007. All rights reserved.

Unica Corporation

Reservoir Place North

170 Tracer Lane

Waltham, MA 02451-1379

例およびデータ

すべてのソフトウェアおよび関係する文書は、その使用および開示において、適用される Unica Software License and Services Agreement または End User License Agreement に記載される制限の対象となります。ただし、アメリカ合衆国政府ユーザおよび適用される輸出規制に関する制限的権利を持ちます。本文書内の例で使用される企業、名称およびデータは、特に断りのない限り架空のものです。

商標および特許

Unica、Unica ロゴ、Affinium、SiteTracker、My SiteTracker および Ask SiteTracker は、合衆国特許商標庁による Unica Corporation の登録商標です。 Affinium Campaign、Affinium Campaign Collaborate、Affinium Campaign eMessage、Affinium Campaign Interact、Affinium Campaign Optimize、Affinium Detect、Affinium NetInsight、Affinium Leads、Affinium Leads Referral、Affinium Leads Contact、Affinium Model、Affinium Plan、Affinium Mart、Universal Dynamic Interconnect、Interconnect Modules、Marketic、Powering Marketing Success、Customer Value Maximization、MAP、One-to-One Optimization および Right-time Marketing は、Unica Corporation の商標です。その他のすべての商標は、それぞれの所有者の所有物です。本文書で説明されるソフトウェアの一部は、以下の合衆国特許番号で保護されます。6,317,752、6,269,325、6,542,894 および 6,782,390 SiteTracker および SiteTracker 製品は、以下の特許および特許公報に基づいてライセンス供与されます。

US5,675,510、US6,115,680、US6,108,637、US5,796,952、US6,138,155、US6,653,696、US6,763,386、AU0701813、BR9609217、CA2223919、EP0843946、JP03317705、MX193614、NO09705728、AU735285、CA2246746、CN1174316、CN1547123、CN1547124、DK870234、DE69720186、ES2195170、AU727170、BR9808033、CA2284530、CN1251669、IL131871、JP2000514942、KR341110、NZ337756、WO9641495、EP0870234、EP1130526、EP1168196、US20040078292、WO9810349、US20050114511、US20040221033、WO9843380

注意: 本文書は、Unica Corporation (「Unica」) の機密情報および専売特許情報を含みます。Unica Corporation による明示的な書面による同意なしに本文書を使用、複製、または開示することは禁止されません。

文書情報

Installing and Configuring the Unica Page Tags Script, Version 7.1

はじめに	1
SiteTracker の命名規則	1
このドキュメントの使用法	1
ページタグ付けスクリプトのインストール.....	2
手順.....	2
その他の設定	4
Unica Web サーバプラグイン	4
拡張ロギングを有効にする	5
「キャッシュを制御する」ヘッダーの追加	5
ブラウザの警告を生じさせないセキュアページへのタグ付け	5
FAQ	6
全般	6
ページタグの設定	7
イベントタグ付け	9
SiteTracker へのデータインポート	14
ページタグ付け変数および関数.....	17
ntpametag.js 内の必須変数.....	17
NTPT_IMGSRG	17
NTPT_FLDS.....	18
NTPT_MAXTAGWAIT	18
ntpametag.js 内のオプションの変数	19
NTPT_HTTPSIMGSRG.....	19
NTPT_GLBLEXTA	19
NTPT_GLBLREFTOP	20

ページ固有のオプションの変数	20
NTPT_PGEXTRA	21
NTPT_PGREFTOP	22
NTPT_NOINITIALTAG	22
ユーザからの呼び出しが可能なページタグ付け関数（イベントタグ付け）	22
ntptAddPair	22
引数.....	22
目的.....	22
ntptDropPair.....	23
引数.....	23
目的.....	23
ntptEventTag	23
引数.....	23
目的.....	23
ntptLinkTag	24
引数.....	24
目的.....	24
ntptSubmitTag.....	25
引数.....	25
目的.....	25

はじめに

Unicaページタグ付けスクリプトを使うと、ページタグから収集されたデータを解析することができます。JavaScript を使用してWeb ページに「タグ付け」を行い、訪問者がそのタグ付きのページを訪問すると訪問者のブラウザがその情報をページタグサーバに送信するようにすることで、データを収集します。この情報は、ページタグイメージに対するリクエストとしてページタグサーバのログに表示され、SiteTracker にインポートして解析することができます。

ページタグイメージに対して出されたリクエストは、元のタグ付きページへの要求を含む、Web サーバの実際のログエントリから独立して解析したり、実際のログエントリの解析で補助的に使用することができます。したがって、追跡したいページを選択してタグを付けたり、解析のためにさらに詳しい情報が必要なページにタグを付けることができます。さらに、リクエストするイメージは、必ずしもタグが埋め込まれているページと同じWebサーバ上に存在する必要はありません。ページにタグを付けることによって、Web サーバのログファイルにアクセスしないサイトにあるページでも追跡することができます。

ページタグによって収集されたデータでログファイルを補強することによって、クライアント側の追加情報（スクリーン解像度など）を所得したり、訪問者によるショッピングカートへの商品の追加や削除などの、ページ上の「イベント」を追跡することができます。

SiteTracker の命名規則

このドキュメントは、SiteTracker Standardエディション、SiteTracker Premierエディションの双方に対応できます。特記記載が無い限り、このドキュメント中に記された内容は両製品に適用可能です。

このドキュメントの使用法

このドキュメントは以下の用途に役立てることができます。

- Unicaページタグ付けスクリプト内のコンポーネントのインストールおよび設定。
- ページへのタグ付け、ページ上のイベントタグ付け、およびページタグから収集されたデータのインポートに関する、一般的な疑問や具体的な疑問の解決。
- 必須およびオプションのページタグ付け変数の設定、およびページタグ付け関数の呼び出し。

このドキュメントを最後までお読みいただいてから、SiteTracker ページタグ付けスクリプトのインストールおよび設定を行うことをお勧めします。

ページタグ付けスクリプトのインストール

SiteTracker ページタグ付けスクリプトは、`ntpagetag.zip` という名前の ZIP ファイルにまとまっています。このファイルは SiteTracker インストール CD に含まれています。また、下記からダウンロードすることもできます。

<http://www.sitetracker.jp>

`ntpagetag.zip` ファイルの内容には以下のとおりです。

- `ntpagetag.gif` --- 1 ピクセルの透明なページタグイメージ
- `ntpagetag.js` --- Web サイト上で SiteTracker ページタグ付けを実行するためのJavaScript
- `readme.pdf` --- このドキュメント
- `sample.html` --- イベントタグを含むサンプルページタグのファイル

手順

1. `ntpagetag.zip` を任意のディレクトリに解凍します。
2. `ntpagetag.gif` イメージを誰からでもアクセスできる公開されたWeb サーバにコピーします。通常、ページタグイメージは HTML ドキュメントのルートまたはページタグWeb サーバのイメージフォルダに配置します。

備考： `ntpagetag.gif` イメージは、タグ付けしたいWeb サイトと同じサーバ上に配置することもできますが、これは必ず必要というわけではではありません。別のWeb サーバに配置したほうが好ましい場合もあります。

3. イメージには一般からのアクセスが可能であることを確認します。そのためには、ブラウザにイメージのURL を入力してロードします。`ntpagetag.gif` は 1 ピクセルの透明なイメージのため、ロードに成功すると空白のページが表示されます。エラーページが表示される場合は、アクセスしようとしている HTML ドキュメントパスにイメージがあり、公開アクセスの許可が設定されているか確認し

てください。

リクエストに対するWeb サーバのログファイルをチェックすることでイメージが利用可能かどうかを確認できます。

備考： ページタグサーバ上で収集されたデータをSiteTrackerで解析するために、Web サーバのログファイルにアクセスできることを確認してください。

4. テキストエディタで `ntpagetag.js` を開きます。以下の手順を実行します。

- a. “`var NTPT_IMGSRC = '/images/ntpagetag.gif';`” という行を検索します。もし、画像ファイルやJSファイルがWebサーバと同じサーバにある場合、この行を`ntpagetag.gif`のパスを相対パス、または絶対パスに変換してください。もし、画像ファイルが別のページタグサーバーから配信されているようであれば、この行をドメイン名を含む絶対パスのイメージに変換してください。

例：`http://mypagetagserver.com/ntpagetag.gif`

もし、`ntpagetag.js`ファイルを別のページタグサーバーへ配信させないようにするには、JSファイルのページタグが同じサーバーのイメージページタグであっても、絶対パスを`gif`のページタグに指定させなければなりません。（詳細は、18ページの「NTPT_IMGSRC」を参照。）

- b. “`var NTPT_FLDS = new Object();`” に後続する行を検索します。ページタグ付けの要件に従い、`NTPT_FLDS` 変数のプロパティを“`true`” から“`false`” に変更します。（詳細は、19ページの「NTPT_FLDS」を参照。）

- c. 変更を `ntpagetag.js` に保存します。

5. `ntpagetag.js` をページタグ付けを実装するサーバーかWebサーバーにコピーします。

備考： `ntpagetag.js` がWeb サーバ上で一般からのアクセスが可能であることを確認してください。

6. テキストエディタで `sample.html` を開きます。以下の手順を実行します。

- a. Unicaページタグを検索します。このタグは以下の行で構成されています。

```
<!-- BEGIN: Unica Page Tag -->
<!-- Copyright 2001-2006, Unica Corporation All rights
reserved. -->
```

```
<script language="JavaScript" src=
"/scripts/ntpametag.js"></script>
<noscript>

</noscript>
Unica
```

- b. 必要に応じて、`script` 要素の“`src`” 属性を、Web サーバまたはページタグサーバ上の `ntpametag.js` ファイルの格納場所（相対パスまたは絶対パス） に一致するように変更します。もし、ページタグサーバにスクリプトが存在する場合には、絶対パスを使用してください。
 - c. 同様に、`img` 要素の“`src`” 属性を、Web サーバまたはページタグサーバ上の `ntpametag.js` ファイルの格納場所（相対パスまたは絶対パス） に一致するように変更します。
 - d. **SiteTracker** ページタグ（ステップ6a で示した行） を選択してコピーします。
 - e. 変更を `sample.html` に保存します。
7. Web サイト上のページにタグを付けるには、タグ付けしたい各 HTML ページを開きます。Unica ページタグを各ページにペーストします。なるべく<head> タグ内にペーストするようにします。
8. タグ付けした HTML ページをWeb サーバにコピーします。これで、ページが表示されるとログに記録されるようになります。

その他の設定

ページタグから収集したデータの利用価値をより向上させるには、Web サーバのクッキープラグインを設定してページタグWeb サーバの拡張ロギングを有効にし、ページタグWeb サーバに「キャッシュを制御する」ヘッダーが含まれるように設定します。

Unica Web サーバプラグイン

Unica Web サーバプラグインを使うと、Web サーバで訪問者を認識するクッキーを生成してログに残すことができます。セッション化（SiteTracker がページビューの閲覧を訪問としてグループ化する

ること) は、クッキーを使用した場合、はるかに正確に実行できます。

Unica Web サーバプラグイン は、<http://www.sitetracker.jp>からダウンロードできます。

拡張ロギングを有効にする

Web サーバで拡張ロギングを有効にする方法についての詳細は、SiteTracker8インストールガイドの「解析するログファイルのあるWebサーバを設定する」を参照してください。

「キャッシュを制御する」ヘッダーの追加

タグ付きページへの訪問者は、ページをロードするたびにページタグイメージをリクエストします。訪問者のブラウザで JavaScript が有効な場合は、個々のリクエストは一意となります。

ブラウザの JavaScript が無効に設定されている訪問者は、タグ付きページをロードするたびに同じイメージをリクエストします。ブラウザ（基本環境）でイメージのキャッシュが有効に設定されていると、ページタグWeb ブラウザにタグのリクエストを送信せずにページをロードできます。この結果、JavaScript を無効にしている訪問者の表示数は減少します。

ブラウザで JavaScript が無効な訪問者の数をより正確に把握するためには、`ntpagetag.gif` イメージの「キャッシュを制御する」ヘッダーが含まれるようにWeb サーバを設定します。ヘッダーには以下の行が含まれるようにします。

```
Cache-control: no-store
```

```
Pragma: no-cache
```

ブラウザの警告を生じさせないセキュアページへのタグ付け

セキュア保護されたコンテンツとそれ以外のコンテンツがページ上に混在すると、ブラウザが警告を出す場合があります。このため、セキュア (HTTPS) ページを利用する場合はセキュア非対応 (HTTP) タグを発行しないようにすることが重要です。

セキュアページを利用する場合、つまり、このページが `https://` で始まる場所からアクセス可能な場合、`ntpagetag.js` のスクリプトタグの“src”属性は、この属性が`https://` で始まる場合を除き絶対パスで指定しないようにしてください。

さらに、`ntpagetag.js` ファイルの `NTPT_IMGSRC` 変数がセキュア非対応の格納場所に設定されている場合は、`NTPT_HTTPSIMGSRC` 変数（同様に `ntpagetag.js` ファイルのヘッダーに含まれる）をセキュア対応の格納場所に設定する必要があります。

FAQ

このセクションでは、Unica ページタグ付けスクリプトに関する全般的な質問や具体的な疑問をリストします。疑問は次の4つのカテゴリに分類されます。

- 全般
- ページタグの設定
- イベントタグ付け
- データの SiteTracker へのインポート

全般

1. SiteTracker は ページタグ付けソリューションですか。

はい。ただし、SiteTracker は単なるページタグ付けソリューションではありません。

SiteTracker は従来のページタグ付け用に設定されたサイトからのログだけでなく、Web サーバの設定から生成されたログファイルも解析することができます。SiteTracker は、いずれの方法でもデータのインポートが可能です。

また、SiteTracker では従来のWeb サーバログ情報を、ページタグ付けによって収集される訪問者情報で補強することもできます。たとえば、ログファイルにページタグで収集されたデータを追加することによって、画面の解像度などのクライアント側の詳細情報を入手できます。

2. ページタグ付けによる解析は、従来のWeb サーバログの解析とどのように異なりますか。

収集できる情報のタイプと、その情報を収集するために使われる方法の両方が異なります。ページタグ付けの解析では、訪問者のブラウザからの訪問者に関する情報の送信に重点が置かれています。従来のWeb サーバログの解析は、このような方法でブラウザに依存するわけではありません。逆に言うと、ページタグ付けソリューションは、従来のWeb サーバログよりも訪問者情報を豊富に提供できます。

3. ページタグ付けと、従来のWeb サーバログ解析と、どちらを使うべきでしょうか。

Web サイトのサーバログへのアクセス、Web サイトの種類、訪問者の訪問について得たい情報の種類などの複数の要因によって採用すべき方法は異なります。現実には、両方とも使いたいと思われるかもしれません。SiteTracker では、ページタグ付けと従来のサーバログのいずれかのみを使うことも、2 つの方法を組み合わせることも可能です。

ページタグの設定

1. ページタグに含ませることのできる情報。

NTPT_FLDS変数によってページタグに含めることのできるフィールドは以下のとおりです。

(詳細は19ページの「NTPT_FLDS」を参照)

フィールド	説明	例
cd	ユーザ側スクリーンの表示色深度	cd=24
ju	ユーザブラウザが Java をサポートするかどうか	ju=0
lc	ドキュメントの格納場所位置 (タグ付け対象のページ)	lc=http%3A/testserver/sample.html
ln	ユーザのブラウザが使用するよう設定されている言語のエンコード	ln=en
rf	リクエストの参照元	rf=http%3A/testserver/
rs	ユーザの画面の解像度 (幅x 高さ)	rs=1280x1024
tz	ユーザ側サーバのタイムゾーン	tz=GMT%20-04%3A00

NTPT_PGEXTRA変数によるページタグには、以下のページ固有のフィールドをオプション設定することもできます。

フィールド	説明	例
pv	SiteTracker でページタグリクエストが通常破棄されていた場合は、リクエストが通常破棄されていた場合は、リクエストが出されるたびに個別のページの閲覧としてカウントします。また、ページの閲覧としてカウントされていたリクエストは破棄されます。	pv=0
sc	リクエストのステータス (HTTP ステータスコード)	sc=404

2. 自分のWeb サイトのページタグに情報を含めたり除外するにはどのような方法がありますか。

利用できるフィールドや必要なフィールドは、`ntpagetag.js` ファイルのヘッダー内に関連付けられている `NTPT_FLDS` プロパティを設定することによって含めたり除外することができます。たとえば、ページタグにユーザの画面の解像度を含める場合は、`NTPT_FLDS.rs` プロパティを `true` にします。このフィールドを除外するためには、このプロパティを `false` にします。`NTPT_FLDS` の詳細については、19 ページの「`NTPT_FLDS`」を参照してください。

3. フレーム付きのWeb サイトに送信する参照元情報はどのようにして制御できますか。

デフォルトの動作では、各フレームの参照元を、フレームを開いたページとしてレポートされます。各フレームが、ページタグの参照元フィールドとしての最高位レベルに位置する「親」をレポートできます。変更は以下のようにできます。

- すべてのページに関する動作を変更するには、`ntpagetag.js` ファイルのヘッダー内の `NTPT_GLBLREFTOP` 変数を `true` に設定します。`NTPT_GLBLREFTOP` の詳細は、21ページの「`NTPT_GLBLREFTOP`」を参照してください。
- 個々のページに関する動作を変更するには、ページに `NTPT_PGREFTOP` 変数を設定します。`NTPT_PGREFTOP` の詳細は、24ページの「`NTPT_PGREFTOP`」を参照してください。

4. 個々のページのタグに情報を追加することはできますか。

できます。ページタグに含まれるデフォルトのフィールド以外にも、`NTPT_PGEXTRA` 変数を使うと特定のページのタグ他のフィールドを追加することができます。この変数は、送信されるページタグに適用されるタグ修飾子です。たとえば、あるページのタグに、“red” と設定されたフィールド “color” を含めたい場合は、タグ付けしたいページの HTML で、ページタグ自体の前に `TPT_PGEXTRA` を “color= red” に設定します。

この方法で既存のフィールドを変更することもできます。たとえば、複数のページの格納場所（“lc” フィールド）に “welcome.html” などの同じ値を設定したい場合、各ページの `NTPT_PGEXTRA` 変数を “lc=welcome.html” に設定します。`NTPT_PGEXTRA` の詳細は、23ページの「`NTPT_PGEXTRA`」を参照してください。

5. サイト上のすべてのページのタグに情報を追加することはできますか。

NTPT_GLBLEXTA 変数を使うと、サイト上のすべてのページのタグに情報を追加することができます。NTPT_GLBLEXTA 変数はntpagetag.js ファイルのヘッダーに設定する点を除いて NTPT_PGEXTRA 変数と同様に機能し、この変数はタグ付けされたすべてのページに適用されます。NTPT_GLBLEXTA の詳細は、21ページの「NTPT_GLBLEXTA」を参照してください。

送出タグを変更すると、NTPT_GLBLEXTA 修飾子は NTPT_PGEXTRA 変数より先に適用されます。したがって、NTPT_PGEXTRA 変数を使って NTPT_GLBLEXTA 変数で設定されたフィールドを上書きすることができます。

イベントタグ付け

1. “イベントタグ付け” とは何ですか。

ページのロード時のタグ付けに加えて、ページ上で特定のイベントが発生したときにページタグを生成すると便利な場合があります。このタイプのページタグ付けは、“イベントタグ付け” と呼ばれます。

2. イベントタグ付けはページタグ付けとは異なりますか。

イベントタグ付けは、ページタグ付けのなかでも特殊なタグ付けを指します。ページ上にイベントタグを生成するには、標準的なページタグとは異なる方法でページを変更する必要があります。

3. イベントタグ付けは何に使われますか。

イベントタグ付けを使うと、製品のショッピングカートへの追加、フォーム上のフィールド変更、ドロップダウンリストボックスのオプション選択、フォーム送信など、ページ上のイベントを解析することができます。イベントタグ付けは、追跡したいイベントが発生したときにJavaScript のイベントタグ付け関数を使って実装します。

最も基本的な方法としては、イベントタグ付けを行うと、ページ上で発生する任意の JavaScript イベントにタグ付けすることができます。

ntpagetag.js ファイルは、さまざまなタイプのイベントタグ付けに関する JavaScript 関数を提供します。

4. ntpagetag.js には、イベントタグ付けを利用しやすくするどのような JavaScript がありますか。

以下の関数を使用すると、イベントタグ付けをより簡単に実行できます。

タグ付けの対象	使用する関数
リンクイベント	ntptLinkTag (詳細は、26 ページの「ntptLinkTag」を参照。)
フォームの送信	ntptSubmitTag (詳細は、27 ページの「ntptSubmitTag」を参照。)
一般的なイベント	ntptEventTag (詳細は、25 ページの「ntptEventTag」を参照。)

ntptAddPair および ntptDropPair 関数を使って、送信するタグを送出前に変更することができます。ntptAddPairの詳細は、24ページの“ntptAddPair”を参照してください。ntptDropPairの詳細は、25 ページの「ntptDropPair」を参照してください。

5. サイト上の PDF ファイルまたはその他の HTML 以外のドキュメント、およびダウンロードにはどのような方法でリンクにタグを付け付けることができますか。

ファイル内にタグを生成するのに必要な JavaScript を含める方法がないため、Web サーバ上の HTML 以外のファイルに標準的なページタグ付けを使うことはできません。ただし、ファイルへのリンクにタグを付けて、それをタグ付きページに類似している、ページタグのログに出力させることはできます。

HTML 以外のファイルにリンクするページには、通常通り SiteTracker ページタグが含まれている必要があります。HTML 以外のコンテンツへのリンクにタグを付けるには、リンクアンカー要素の“onclick”属性を使って ntptLinkTag 関数を呼び出します。

例

```
<a href="some.pdf" onclick="return ntptLinkTag( this );">SomePDF</a>
```

ntptLinkTag の詳細、26 ページの「ntptLinkTag」を参照してください。

6. フォームの送信にはどのような方法でタグ付けができますか。

フォームの送信にタグを付けるには、フォーム 要素の `onsubmit` 属性を使って `ntptSubmitTag` 関数を呼び出します。

例

```
<form name="myform" method="post" onsubmit="return ntptSubmitTag( this );">
```

`ntptSubmitTag` の詳細は、27ページの「`ntptSubmitTag`」を参照してください。

7. 入力途中のフォームにタグを付けることはできますか。

フォーム入力要素の “`onchange`” メソッドを使うと、フォーム入力要素が変更されるたびにイベントタグが生成されることにより、入力途中のフォームからの情報を取得できます。

例

```
<input name="mytext" type="text" onchange="ntptEventTag(
' mytext_changedto='+ escape( this.value ) );"></input>
```

`ntptEventTag` の詳細は、25 ページの「`ntptEventTag`」を参照してください。

8. 自分のページ上で発生する他のユーザーイベントはどのような方法でタグ付けできますか。

`ntptEventTag` 関数を使うと、ページ上の任意の JavaScript イベントにタグ付けできます。たとえば、ユーザーがリンクでない画像をクリックするたびにタグが生成されるようにする場合は、イメージ要素の “`onclick`” 属性を使うことができます。

例

```

```

`ntptEventTag` の詳細は、25 ページの「`ntptEventTag`」を参照してください。

9. ページ上でタグ付きのリンクやフォーム送信をクリックするとしばしば遅延が発生するのはなぜですか。

`ntptLinkTag` および `ntptSubmitTag` 関数がタグ付けするイベント（リンクをたどってフォームを送信）は、最終的には既存ページがアンロードされることとなります。`ntpagetag.js` JavaScript ファイルは、ページがアンロードされる前にタグが確実に正しく生成されるようにするために、これらのイベントにタグ付けするときに短時間の遅延を生じさせます。

この遅延時間は、`ntpagetag.js` ファイル のヘッダー内の `NTPT_MAXTAGWAIT` 変数を使うと設定できます。詳細は、20 ページの「`NTPT_MAXTAGWAIT`」を参照してください。

10. ページ上のイベントだけにタグ付けを行い、ページ自体にはタグ付けを行いたくない場合はどうすればいいですか。

ページにはタグを付けたいが、最初にページがロードされるときにはタグを付けたくない場合があります。この場合は、`NTPT_NOINITIALTAG` 変数を `true` に設定します。最初のページタグが抑制される一方で、後続するページ上のイベントタグは起動します。

`NTPT_NOINITIALTAG` の詳細は、24 ページの「`NTPT_NOINITIALTAG`」を参照してください。

11. Flash Presentationでは、どうやってイベントタグ付けができますか。

Flash Presentation中のあらゆるユーザーアクションに対しページタグを生成できます。例えば、以下の場合に全てのユーザーにページタグを送信することができます。

- クリックボタンを離す
- キーフレームの入力
- シンボルのロールオーバー
- シンボルのドラッグ
- クリックボタンを押し続けている状態
- Flashフォームの送信
- 特定文字をキーボードで入力
- 拡張URLのロード
- マウスを特定方向の特定距離移動

Flash内のイベントを追跡する前に、追跡させたいイベントをハンドルする、ActionScriptイベントハンドラを作成する必要があります。作成したイベントハンドラは、JavaScript のイベントタグ付け機能呼び出す必要があります。(詳細は24ページの「User-Callable Page Tagging Functions (Event Tagging)」を参照。)

例えば、プレゼンテーション中、ページタグを送信したい時にntptEventTag 機能が呼び出されます。(詳細は25ページの「ntptEventTag」を参照。) 例として、ユーザーが特定のボタンを離し、ページタグを送る度に、Actions Inspector (Windowsメニュー上) に以下のコマンドを入力してください。

```
function MyButtonReleasedEventHandler ()
{
    fscommand ("ntptEventTag","button=released");
}
```

ページタグ付けJavaScript APIへイベントを送るためには、Adobeで提供されるJSファイルをHTML格納ファイルへ追加するHTMLテンプレート「Flash with FSCommand」を用いて作成すると良いでしょう。AdobeがHTMLに追加する機能の一つが、DoFSCommandです。Unicaによって提供されている、JavaScriptイベントタグ呼び出すためにDoFSCommandを変更しなければなりません。(詳細はAdobe社より提供されているFlashドキュメントの「FlashとJavaScriptの接続」参照。) DoFSCommand内のサンプルコードを以下のJavaScriptコードで置き換えてください。

```
if (null == args)
    return;

var tmpargs = args.split(",");
if (command == "ntptEventTag")
{
    (0 == tmpargs[0].length) ? ntptEventTag() : ntptEventTag(
        tmpargs[0] );
}
else if (command == "ntptAddPair")
{
```

```
if ( 2 != tmpargs.length )

    return;
else
    ntptAddPair( tmpargs[0], tmpargs[1] );
}
else if ( command == "ntptDropPair" )
{
    if ( 1 != tmpargs.length )
        return;
    else
        ntptDropPair( tmpargs[0] );
}
```

12. SiteTracker では、イベントタグはどのように処理されますか。

SiteTracker では、イベントタグは通常のページタグと同様に処理されます。イベントタグに含まれるフィールドに関するレポートが必要な場合は、カスタム Page Tag パラメータを定義する必要があります。カスタムPage Tag パラメータの定義の詳細は、SiteTracker ユーザガイドの「ページタグ内のパラメータの追跡」を参照してください。

備考： デフォルトでは、“pv” フィールドに値“0” を指定 (“pv=0” など) しない限り、イベントタグはページビューとしてカウントされます。

SiteTracker へのデータインポート

1. ページタグ付けで収集した情報の保存場所はどこですか。

サイト上のタグ付けしたいページの HTML ソースに Unica ページタグ付けブロックを正しく挿入し、ntpagetag.gif ファイルおよび ntpagetag.js ファイルを適切な場所にコピーすると、サイト上のタグ付きページをユーザがリクエストするたびにタグが生成されます。このタグは、ntpagetag.gif イメージを提供する Web サーバのログファイルに保存されます。

2. ページタグ付けログに対して、どのようなタイプの Web サーバとログファイルモードがサポートされていますか。

ページタグサーバで収集されたデータをインポートするときには、**SiteTracker** でサポートされるすべての**Web** サーバログフォーマットが利用できます。これらのフォーマットには以下のものが含まれます。

- Microsoft IIS Standard および Extended
- NCSA Common および Combined
- Netscape Flexible
- W3C Extended

ページタグ付けログフォーマットの自動検出とカスタムログフォーマットの定義も利用できます。ログフォーマットに関するその他の情報は、**SiteTracker** ユーザガイドの「ログファイルのインポート」セクションを参照してください。

3. ページタグで情報が収集されているかどうかは、どのように認識できますか。

Web サーバがログファイルを保存する場所はさまざまです。**ntpagetag.gif** イメージに関連付けられているログファイルが検出できると、**ntpagetag.gif** イメージのリクエストに対応するログファイルを検索できます。以下に、該当する行の例を示します。

```
192.168.0.64 - - [25/Jul/2004:07:30:49 -0400] "GET
/images/ntpagetag.gifjs=1&ts=1089199849489&lc=
http%3A//testserver/sample.html&rf=http%3A//testserver/&rs=
1280x1024&cd=24&ln=en&tz=GMT%20-04%3A00&jv=0 HTTP/1" 200 85
"http://testserver/sample.html" "Mozilla/5 (X11; U; Linux i686;
en-US; rv:1) Gecko/20030425"
```

ユーザによるタグ付きページのロードをするか、イベントタグの実行などによって、**ntpagetag.gif** イメージへのリクエストが出されます。このリクエストごとに上記のような行が存在するはずです。

4. **SiteTracker** プロファイルをページタグ付けログと組み合わせて使用するには、どのように設定できますか。

ntpagetag.gif イメージに関連付けられたログファイルが検出できると、**SiteTracker** を設定す

ることにより、ページタグ付け用に設定されたWeb サイトからのログを解析することができます。従来のWeb サーバログ情報に、ページタグ付けで収集された訪問者の詳細情報を追加するように SiteTracker を設定することもできます。たとえば、ページタグによって収集されたデータをログファイルに追加すると画面の解像度などのクライアント側の詳細情報が入手できます。

SiteTracker プロファイルをページタグ付けログと組み合わせて使用する場合は、SiteTracker ユーザガイドの「ページタグから収集されたデータの解析」を参照してください。

5. どのページタグ付けオプションを使うのが適切ですか。

SiteTracker がページタグ要求を処理する方法は、選択するページタグ付けオプションによって異なります。詳細は、SiteTracker ユーザガイドの「ページタグから収集されたデータの解析」を参照してください。

6. どのイメージがページタグ付けイメージとして使われたかはどのように指定できますか。

ページタグからデータを収集するときに、どのイメージから収集するかを記述するフィルタを指定することができます。デフォルトでは、フィルタは“/ntpagetag.gif” を含むイメージからデータを収集するように設定されています。フィルタの指定の詳細は、SiteTracker ユーザガイドの「ページタグから収集されたデータの解析」を参照してください。

7. SiteTracker は、ページタグから収集されたデータをどのようにレポートしますか。

SiteTracker は、ページタグから解析した格納場所 (“lc”) および参照元 (“rf”) フィールドを、すべてのリクエスト要求のページおよび参照元として使用します。画面解像度サマリは、ページタグの “rs” フィールドから解析した値に基づいてユーザの画面の解像度をレポートします。

SiteTracker は、“sc” フィールドの値をリクエスト要求のステータスコードとして使用します (たとえば値 “200” はリクエストの成功を示す)。ページタグに “sc” フィールドがあり、その値が400 以上の場合 (“sc=404” など)、このリクエスト要求はステータスコードとともに SiteTracker エラーサマリに表示されます。

SiteTracker に [追加データのあるログファイルの補助としてのみ使用] オプションが設定されている場合、通常ではページビューを追跡しているときにページタグリクエストは破棄されま

すが、“pv” フィールドが“pv=1” に設定されていると、この場合でもページタグはSiteTracker でページの閲覧としてカウントされます。ほかのページタグ付けオプションでは、ページタグは SiteTracker でページの閲覧としてデフォルトでカウントされます。つまり、“pv” フィールドが“0” に設定されていると、ページタグのリクエストは SiteTracker でページの閲覧としてカウントされないことになります。

Page Tag パラメータを定義すると、SiteTracker で追加情報をレポートできます。詳細は、SiteTracker ユーザガイドの「ページタグのパラメータの追跡」を参照してください。

ページタグ付け変数および関数

タグ付けするページに利用可能なJavaScript 変数および関数は以下のとおりです。

- ntpagetag.js 内の必須変数
- ntpagetag.js 内のオプションの変数
- ページ固有のオプションの変数
- ユーザが呼び出すことのできるページタグ付け関数（イベントタグ付け）

次に、これらの変数および関数について説明します。

ntpagetag.js 内の必須変数

以下の3 つの変数は必ず指定する必要があります。ntpagetag.js 内で設定されます。

NTPT_IMGSRC

この文字列変数は、ページタグイメージの URL です。プロトコルとサーバを含めることによって、URL を完全修飾できます。以下のように指定します。

```
var NTPT_IMGSRC = 'http://mysite.com/images/ntpagetag.gif'
```

ページタグイメージが、タグ付けページと同一Web サーバ上に存在する場合は、プロトコルとサーバを省略できます。以下のように指定します。

```
var NTPT_IMGSRC = '/images/ntpagetag.gif'
```

できないようであれば、完全修飾されたURLを入力してください。もし、Webページを配信するサーバーと、別のサーバーにntpagetag.jsファイルを配信させないようにするためには、gifページタグへの完全修飾されたURLを入力しなければなりません。たとえ、ntpagetag.jsファイルとgifページタグが同じサーバー上にあってもです。

NTPT_FLDS

このオブジェクト変数には、ntpagetag.js がページタグ内に含めることができる各フィールドに対する属性が含まれています。これらのフィールドは、以下のように定義されています。

```
var NTPT_FLDS = new Object();
NTPT_FLDS.lc = true; // ドキュメントの格納場所
NTPT_FLDS.rf = true; // ドキュメントの参照元
NTPT_FLDS.rs = true; // ユーザの画面の解像度
NTPT_FLDS.cd = true; // ユーザの表示色深度
NTPT_FLDS.ln = true; // ブラウザの使用言語
NTPT_FLDS.tz = true; // ユーザのタイムゾーン
NTPT_FLDS.jv = true; // ブラウザの Java のサポートの有無
```

フィールド属性は、true または false に設定します。

- 値が true の場合は、そのフィールドはページタグに含まれます。
- 値が false の場合は、そのフィールドはページタグから省略されます。

NTPT_MAXTAGWAIT

この変数の数値は、ntptLinkTag() または ntptSubmitTag() の呼び出しが、ページタグリクエストの送信を待機する、秒数の最大値を定義します。ページタグリクエストはこのタイムアウト以前に戻される場合があるため、この数値は、リンクタグおよび送信タグの遅延の「想定される最長時間」にする必要があります。

有効な値は以下のとおりです。

- 1 以上の任意の値 -- 正の数値は、リンクをたどるか送信する前に待機する秒数を示します。
NTPT_MAXTAGWAIT 変数を1 (1 秒、デフォルト) に設定すれば、ユーザから意識されることがほとんどないため、すべてのタグ付きリンクとフォーム送信を効率よくキャプチャできる

十分な長さです。

ブラウザによっては、ページタグの実行中に同時にページがアンロードされると、ページタグの JavaScript コードが実行されないことがあります。この場合は、ページのアンロード前にページタグが確実に実行されるようにするために、1以上の値を使用できます。

備考：任意の `ntptLinkTag()` および `ntptSubmitTag()` の呼び出しで、“`maxtagwait`” 引数に値を渡すと、デフォルト値の1.0 秒を上書きすることができます。

- `-1` 値が `-1` は、タイムアウトを使わないことを意味します。ブラウザによっては、リンクのクリックやフォームの送信などのページをアンロードするイベントに確実にタグ付けするために、短い遅延が必要な場合があるため、この値を使用する場合は注意が必要です。

例

```
var NTPT_MAXTAGWAIT = 1; // 1 秒
var NTPT_MAXTAGWAIT = 2.5; // 2.5 秒
var NTPT_MAXTAGWAIT = 0.1; // 0.1 秒
var NTPT_MAXTAGWAIT = -1; // 遅延なし
```

ntpametag.js 内のオプションの変数

以下の3 つの変数はオプションです。ntpametag.js 内で設定されます。

NTPT_HTTPSIMGSRC

この文字列変数は、NTPT_IMAGESRC と同様に、ページタグイメージの URL です。

NTPT_HTTPSIMGSRC が設定され、空でない場合、タグ付けされているページが “https:” プロトコルを使ってアクセスされるときに使われます。

NTPT_GLBLEXTA

この文字列変数は、ページタグに適用される最初の「クエリ修飾子」です。クエリ修飾子には、空、または初期ページタグクエリ文字列を修飾するキーと値のペアを含む文字列を使用することができます。初期ページタグクエリ文字列にキーがない場合、クエリ修飾子のキーと値のペアが結果のクエリ文字列に付け加えられます。

クエリ修飾子のキーがすでにページタグの初期クエリ文字列にある場合は、そのキーの値がクエリ修飾子の新しい値に置換されます。その修飾子の値が空の場合は、キーと値のペアが結果のクエリ文字列から削除されます。

NTPT_GLBLEXTTRA クエリ修飾子はすべてのページタグに適用されます。`ntpagetag.js` を含むすべてのページ上のすべてのページタグの修飾に使用します。

例

```
// 'sitetheme=blue' ペアをすべてのページタグに付加
var NTPT_GLBLEXTTRA = 'sitetheme=blue';
```

NTPT_GLBLREFTOP

この変数は `true` または `false` に設定します。`true` の場合は、現在のページのトップ（最大数が含まれる）フレームから参照元（`rf` フィールド）が検索されます。それ以外の場合は、参照元は現在のページから検索されます。この値を設定しない場合、デフォルトは `false` になります。

備考： `NTPT_PGREFTOP` 変数を設定すると、この変数の値は上書きされます。

ページ固有のオプションの変数

次に説明するページ固有の変数はオプションです。例で示すように、ページ固有の変数は `SiteTracker` ページタグがロードされる前に定義しなければならないことに注意してください。

<head>

<!-- BEGIN: このセクションは、`SiteTracker` ページタグのページ固有の変数を設定するために使います。 -->

```
<script language="JavaScript">
```

```
var NTPT_PGEXTRA = 'pagetheme=blue&rf=';
```

```
var NTPT_PGREFTOP = false;
```

```
var NTPT_NOINITIALTAG = false;
```

```
</script>
```

<!-- END: このセクションは、`SiteTracker` ページタグのページ固有の変数を設定するために使いま

```
す。-->

<!-- BEGIN: SiteTracker Pagetag->
<!-- Copyright 2004 Sane Solutions, LLC. All rights reserved. -->
<script language="JavaScript" src="/scripts/ntpametag.js"></script>
<noscript>

</noscript>
<!-- END: SiteTracker Pagrtag-->

</head>
```

NTPT_PGEXTRA

この変数は、ページタグに適用される2番目の「クエリ修飾子」です。NTPT_PGEXTRA 修飾子は NTPT_GLBLEXTA（最初のクエリ修飾子）の後で適用されるため、初期クエリ文字列と NTPT_GLBLEXTA のどちらのキーと値のペアも上書きすることができます。クエリ修飾子の詳細は、21 ページの「NTPT_GLBLEXTA」を参照してください。

備考：“sc”（状態コード）フィールドは、リクエストのステータスコードの代わりに、SiteTracker にステータスコードを提供します。NTPT_PGEXTRA で“sc”フィールドを定義すると、主にエラーリクエストのログを記録のに便利です。たとえば、エラーページのページタグの“sc”フィールドに400以上の値を割り当て、SiteTrackerがそのリクエストをエラーとして扱うようにすることができます。

例

```
// 'pagetheme=red' ペアを付加。'rf' フィールドを削除。
var NTPT_PGEXTRA = 'pagetheme=blue&rf=';

// このページを 404（見つかりません）エラーとして扱う。
var NTPT_PGEXTRA = 'sc=404';
```

備考：リンクタグの目的は、ほかのページ上のページタグをシミュレーションすることなので、ntptLinkTag() の呼び出しで生成されたページタグにはこのクエリ修飾子は適用されません。リンクタグのクエリ文字列を修飾するには、その関数の“querymod”引数を使います。

SiteTracker 内でページタグが独立したページビューとして確実に表示されるようにするには (SiteTracker に[追加データのあるログファイルの補助としてのみ使用] オプションを設定すると、通常、ページ閲覧の追跡で ページタグリクエストは破棄されますが、この場合でも)、“pv” フィールドを“1” に設定します。

カウントされるはずの要求を破棄するには、“pv” フィールドを“0” に設定します。これはイベントタグを使うときに便利です。イベントタグを起動するときに“pv=0” を使うと、そのリクエストをページビューとしてカウントせずに、パラメータを解析して対応する訪問に関連付けることができます。

NTPT_PGREFTOP

この変数は、NTPT_GLBLREFTOP と同じ目的で使われます。この変数は、トップフレームと現在のページのいずれから参照元を検索するかを指定します。NTPT_PGREFTOP を設定すると、NTPT_GLBLREFTOP が上書きされます。NTPT_GLBLREFTOP の詳細は、21 ページの「NTPT_PGREFTOP」を参照してください。

NTPT_NOINITIALTAG

NTPT_NOINITIALTAG を true に設定すると、現在のページの最初のページタグが抑制され、ページ上のイベントタグだけが起動します。

ユーザからの呼び出しが可能なページタグ付け関数 (イベントタグ付け)

以下のページでは、ユーザからの呼び出し可能なページタグ付け関数を説明します。sample.html の例をを参照してください。

ntptAddPair

引数

(key, value)

目的

“key” と “value” 引数は、次に起動するイベントタグの実際のクエリ文字列に追加されるキーと

値のペアです。クエリ文字列にすでにそのキーがある場合は置換されます。“value” 引数が空 (') の場合は、そのペアは削除されます。この関数は、イベントタグが起動する前に実際のクエリ文字列を修飾する場合に呼び出します。

備考： 実際のクエリ文字列は、イベントタグが起動した後必ず初期クエリ文字列にリセットされます。

ntptDropPair

引数

(key)

目的

この関数は“key” 引数にマッチするペアを実際のクエリ文字列から削除します。この関数は、イベントタグが起動する前に実際のクエリ文字列を修飾する場合に呼び出します。

備考： ntptDropPair('mykey') の呼び出しは、ntptAddPair('mykey','') の呼び出しと等価です。

ntptEventTag

引数

([querymod])

目的

この関数は実際のクエリ文字列を使ってイベントタグを起動します。この関数は document 要素のイベントハンドラから呼び出します。

オプションの“querymod” 引数で実際のクエリ文字列を修飾することができますが、これは NTPT_GLBLEXTA および NTPT_PGEXTRA と同様のクエリ修飾子です。

“querymod” クエリ修飾子は、ページタグが要求される前に実際のクエリ文字列に適用される最後の修飾子のため、実際のクエリ文字列内にある任意のキーと値のペアを置換できます。

実際のクエリ文字列は、イベントタグが要求された後で初期クエリ文字列にリセットされます。

例

```
<input type="text" name="mytext" onchange="ntptEventTag(
  'mytext_changed=1&mytext='+ escape( this.value ) + '");">
```

ntptLinkTag

引数

(linkobj [, querymod [, maxtagwait]])

目的

この関数は、ページタグ付けではアクセスできないリンクにタグ付けするために使います。これらのリンクには、ダウンロードおよび HTML 以外のページが含まれます。この関数は常にリンクの “onclick” 属性から呼び出します。

ntptLinkTag() を呼び出すときは、この関数の戻り値を必ず “onclick” ハンドラに返さなくてはなりません。すなわち、アンカーの “onclick” 属性には “return ntptLinkTag(this);” が含まれている必要があります。ntptLinkTag()呼び出しの前の “return” を省略しないでください。

引数は以下のとおりです。

- | | |
|------------|---|
| linkobj | この関数がリンクにタグ付けを行った後でリンクをたどれるように、この引数は必ずキーワード “this” にします。 |
| querymod | このオプションの引数はリンクタグのクエリ修飾子です。ntptEventTag() のオプションの “querymod” 引数と同様に、ページタグ要求の直前に実際のクエリ文字列を修飾するために使います。 |
| maxtagwait | このオプションの引数は、グローバルな NTPT_MAXTAGWAIT 変数をこのリンクタグについてだけ上書きします。リンクタグ要求が確実に起動するようにするために待ち時間の最大秒数を指定します。“maxtagwait” 秒が経過するかタグ要求が返ってくるか、どちらか早いほうが起きるまで、ページはアンロードされず、リンクはたどられません。“maxtagwait” の値として -1 を渡すと、遅延なしでリンクをたどります。 |

備考：リンクタグはNTPT_PGEXTRA 変数で、変更できません。

例

```
<a href="mymovie.mpeg" onclick="return ntptLinkTag( this );">
```

ntptSubmitTag

引数

(formobj [, querymod [, maxtagwait]])

目的

この関数はフォーム送信にタグ付けします。以下のように、必ずフォームの“onsubmit” 属性から呼び出してください。

```
<form name="myform" onsubmit="return ntptSubmitTag( this );">
```

ntptSubmitTag() を呼び出すときは、この関数の戻り値を必ず“onsubmit” ハンドラに返す必要があります。言い換えると、フォームの“onsubmit” 属性には“return ntptSubmitTag(this);” が含まれている必要があります。ntptSubmitTag() 呼び出しの前の“return” を省略しないでください。

引数は以下のとおりです。

- | | |
|-------------------|--|
| Formobj | この関数がフォーム送信にタグ付けを行った後でフォームを送信できるように、この引数は必ずキーワード“this” にします。 |
| querymod | このオプションの引数は送信タグのクエリ修飾子です。ntptEventTag() および ntptLinkTag() のオプションの“querymod” 引数と同様に、ページタグ要求の直前に実際のクエリ文字列を修飾するために使います。 |
| maxtagwait | このオプションの引数は、グローバルな NTPT_MAXTAGWAIT 変数をこの送信タグについてだけ上書きします。ntptLinkTag() の“maxtagwait” 引数と同様に、送信タグ要求が確実に起動するようにするために待ち時間の最大秒数を指定します。“maxtagwait” 秒が経過するかタグ要求が返ってくるか、どちらか早いほう起きるまで、ページはアンロードされず、フォームは送信されません。“maxtagwait” の値として -1 を渡すと、遅延なしでフォームが送信されます。 |

