HTML Validation HOWTO

Keith M. Corbett <kmc@specialform.com> v0.2, 29 Octobre 1995 Traduction française: Thierry Bézecourt, Avril 1999. (14 Octobre 1997 : La mini-HOWTO Linux HTML Validation n'est plus maintenue par son auteur. Si vous souhaitez vous en occuper, en français ou en anglais, merci de contacter <gregh@sunsite.unc.edu>)

Cette documentation a pour objet d'expliquer l'utilisation du programme nsgmls afin de vérifier qu'un document HTML est conforme à la définition de type de document (ou "DTD") HTML 2.0. Parmi les définitions de HTML à partir de SGML, ce DTD est la plus fréquemment adoptée. Il définit ainsi un sous-ensemble du balisage HTML correspondant à la pratique courante, ayant les meilleures chances de portabilité sur un nombre important d'agents utilisateurs de HTML (ou "navigateurs").

Contents

1 Introduction

Ceci est un guide d'utilisation de l'analyseur nsgmls pour valider et traiter des documents HTML.

1.1 Coûts et bénéfices

Vous enrichirez vos documents HTML en utilisant toutes les capacités de SGML. Mais la validation de vos documents par référence au DTD HTML a un certain nombre d'avantages et d'inconvénients, avant tout parce que vous avez affaire à un dialecte de HTML plus restreint que celui qui est utilisé actuellement. Les règles "officielles" spécifiant la structure d'un document en HTML et les règles concernant le balisage des données en SGML sont plus strictes que ne l'est la pratique courante sur le Web.

La premier point, et le plus important, est d'avoir conscience que le codage HTML correct se réduit à un ensemble normalisé de balises.

Il n'existe pas de DTD couramment accepté qui reflète exactement le "HTML des navigateurs", tel qu'il est compris par de nombreux programmes clients de navigation. Pour l'essentiel, le DTD HTML 2.0 contient les balises et les attributs les plus communément utilisés sur le Web autour de juin 1994. Divers efforts visant à définir un DTD HTML+ ou HTML 3.0 se trouvent plus ou moins dans une impasse. Et aucun des DTD en circulation ne reconnaîtra l'ensemble des balises rendues populaires par des éditeurs de navigateurs comme Netscape et Microsoft.

1.2 Pour débuter

Contrairement à ce que beaucoup de gens croient, il est possible de travailler en SGML sans y consacrer trop de temps ou d'argent. On peut mettre en place un environnement de développement solide en utilisant exclusivement des logiciels disponibles gratuitement sur un grand nombre de plateformes, y compris Linux, DOS et la plupart des stations de travail Unix. Grâce à quelques personnes qui s'y sont consacrées, tous les outils dont vous avez besoin pour travailler en SGML sont disponibles publiquement sur Internet.

La mise en place de votre environnement (l'analyseur et les bibliothèques) prend un peu de temps, mais pas autant qu'on pourrait le croire.

Au cas où vous éprouveriez l'envie de lire un texte d'introduction à SGML (en anglais) : "SGML: An Author's Guide to the Standard Generalized Markup Language" par Martin Bryan, ou "Practical SGML" par Eric van Herwijnen.

2 Les Outils

2.1 "HTML Check Toolkit"

Si vous voulez un logiciel s'installant automatiquement et qui soit bien intégré, récupérez `HTML Check Toolkit` à l'addresse `http://www.halsoft.com/html-tk/index.html` .

Le seul inconvénient du logiciel HalSoft est qu'il utilise le vieil analyseur sgmls, qui produit des messages d'erreur parfois (encore) plus difficiles à comprendre que ceux de nsgmls.

J'ai utilisé nsgmls sur Linux et Windows (3.x et NT) ; il marche en principe sur de nombreuses autres plateformes.

2.2 L'analyseur nsgmls

James Clark a mis au point sun ensemble logiciel nommé `sp`, qui contient un analyseur et un validateur SGML, nsgmls (c'est le successeur de l'analyseur sgmls, longtemps considéré comme le programme de référence).

Pour plus d'informations sur sp, voyez l'URL `http://www.jclark.com/sp.html` .

Et vous pouvez le télécharger directement depuis `ftp://ftp.jclark.com/pub/sp/` .

Vouz pouvez peut-être aussi récupérer des fichiers exécutables pour votre plateforme. Ou bien télécharger le code source, et dans ce cas suivez les instructions du fichier README pour lancer make.

Il serait utile de créer un répertoire public au sommet de la hiérarchie, pour y mettre les fichiers relatifs à SGML. Par exemple, sur mon PC Linux, j'ai plusieurs répertoires pour SGML, dont :

- /usr/sgml/bin

- /usr/sgml/html
- /usr/sgml/sgmls
- /usr/sgml/sp

2.3 Télécharger les spécifications de HTML

Le projet de norme HTML 2.0 contient le fichier de définitions SGML dont vous aurez besoin pour utiliser l'analyseur, c'est-à-dire le DTD (Document Type Definition), la déclaration SGML, et le catalogue d'entité. Vous pouvez obtenir le texte public de HTML 2.0 à l'adresse

`http://www.w3.org/hypertext/WWW/MarkUp/html-spec/`

Téléchargez et installez les fichiers suivants :

- DTD : html*.dtd
- Déclaration SGML : html.decl
- Catalogue d'entités : catalog

Vous pouvez ajouter deux entrées au catalogue des entités HTML pour faciliter l'utilisation de nsgmls :

```
----------------------------------------------------------------------
        -- catalog: SGML Open style entity catalog for HTML --
        -- $Id: HTML-Validation.sgml,v 1.1.1.1 2003/01/03 02:40:54 traduc Exp $ --
 :
 :
        -- Ajouts pour faciliter l'utilisation de nsgmls --
SGMLDECL        "html.decl"
DOCTYPE HTML    "html.dtd"
----------------------------------------------------------------------
```

Ou bien vous pouvez créer un second catalogue contenant ces entrées. Vous passerez ensuite ce catalogue à nsgmls comme argument, avec l'option -m.

3 Analyser un document HTML

Voici une "recette" pour valider un document unique. Lancez simplement le programme nsgmls en lui passant le chemin du (ou des) fichier(s) du catalogue HTML, ainsi que le document :

```
% nsgmls -s -m /usr/sgml/html/catalog <test.html
```

L'option -s supprime la sortie de l'analyseur (voyez ci-dessous).

3.1 Entrée de l'analyseur

Votre document doit être conforme à la norme SGML, ce qui signifie, entre autres, que le type de document doit être déclaré au début de l'entrée (vous pouvez contourner cette règle en spécifiant cette information sur la ligne de commande de nsgmls).

Voici un document HTML simple, qui peut être analysé sans erreur en suivant les règles que j'ai décrites :

```
-----------------------------------------------------------------------
<!doctype html public "-//IETF//DTD HTML 2.0//EN">
<html>
<head>
<title>Document HTML simple.</title>
</head>
<body>
<h1>Document de test</h1>
<p>Ceci est un document de test.</p>
</body>
</html>
-----------------------------------------------------------------------
```

3.2 Sortie de l'analyseur

La sortie standard du programme nsgmls est une forme concentrée de l'entrée SGML que les systèmes de traitement peuvent utiliser comme un analyseur lexical afin de parcourir la structure du document. Pour les besoins de la validation, vous pouvez oublier la sortie standard et ne vous occuper que de l'erreur standard.

Si vous souhaitez avoir la sortie complète, ne spécifiez pas l'option -s et redirigez la sortie vers un fichier

```
% nsgmls -m /usr/sgml/html/catalog <test.html >test.out
```

3.3 Messages de l'analyseur

Les messages d'erreur et d'avertissement de nsgmls peuvent être très difficiles à déchiffrer. Et vous trouverez un grand nombre d'erreurs résultant d'un balisage non-conforme.

Pour rediriger les messages vers un fichier, utilisez l'option -f :

```
% nsgmls -s -m /usr/sgml/html/catalog -f test.err <test.html
```

3.4 Code de retour

L'analyseur indique si le document est conforme au DTD HTML de deux manières :

Code de retour - l'analyseur renvoit un code de sortie de 0 en cas de succès, différent de 0 sinon.

Sortie - si le document est conforme au DTD, la dernière ligne de la sortie standard contient un caractère C unique.

4 Liens

"HTML Check Toolkit" de HalSoft se trouve à l'adresse `http://www.halsoft.com/html-tk/index.html` .

La page de James Clark sur sp se trouve à l'adresse `http://www.jclark.com/sp.html` .

La page du W3C sur la spécification de HTML se trouve à l'adresse `http://www.w3.org/hypertext/WWW/MarkUp/html-spec/` .