

データ伝送ツール curl の国際化 – IRI への対応 –

小林 健雄 伊藤 一成 Martin J. DÜRST

青山学院大学理工学部

1 はじめに

現在, IRI (Internationalized Resource Identifiers) の仕様が RFC 3987 [1] として公開されている. RFC 3987 は, Web アドレスである URI の国際化のための標準化提案 (Proposed Standard) である. この提案は, IETF (Internet Engineering Task Force) と W3C (World Wide Web Consortium) の双方の協調により発行された. URI は Web 上に存在するリソースを繋ぐ重要な要素であるが, URI には US-ASCII の文字しか扱えないという制約がある [2]. 一方, IRI は世界全体の文字を扱える Unicode / ISO 10646 への拡張により, 様々な言語を用いたリソースの記述と識別を可能にする.

Web 上には, URI を元にリソースの送受信を行うツールがいくつか存在する. 本研究では, そのツールの中でも多数のプロトコルに対応し, 様々なプログラム言語で利用可能である curl [3] を選択し, URI の拡張である IRI の実装を試みた.

2 URI における問題を解決する IRI の長所

IRI の最大の長所は, URI では不可能であった世界中の人々が普段利用している言語をそのまま IRI に利用できる点にある. 例えば, 我々日本人は, 漢字, ひらがな, カタカナを普段利用しているが, IRI では, 以下に示すようにこれらの文字が使用可能である.

`http://伊藤一成.jp/日本語.html`

また, IRI はすべての URI を包含するため, IRI に対応するツールは当然 URI にも対応する. 逆にソフトウェアが IRI に未対応の場合でも, 全ての IRI は UTF-8 への変換とパーセントエンコーディングにより URI に変換可能 [1] であるため, 手作業, もしくは独立した変換ツールを使用し URI に変換すれば IRI が利用可能となる.

3 curl と libcurl の概要と特徴

3.1 概要

curl と libcurl [3] は “cURL” と呼ばれるプロジェクトから作られたオープンソースのツールである. この “cURL” の名称は “Client for URLs” にかけてあり, URL (=URI) を扱うツールであることを示してい

る. libcurl はライブラリであり, curl は libcurl を利用したコマンドラインツールである.

3.2 curl

curl は, URL を元にファイルを送受信する C 言語で記述されたツールである. また, curl のすべての転送関連の機能には libcurl ライブラリが利用されている.

3.3 libcurl

libcurl は, FTP, FTPS, HTTP, TELNET, LDAP 等, 数多くのプロトコルに対応している. また, HTTPS 認証, HTTP POST, HTTP PUT, FTP アップロード, kerberos, HTTP form によるアップロード, プロキシ, クッキー, ユーザ名とパスワードに基づく認証, ファイル転送時のレジュームなど多くの機能を持つ. さらに, libcurl は移植が容易で数多くのプラットフォームで利用可能な上, libcurl を C 言語以外の言語で動作させるための言語バインディングを利用することで, Perl, PHP, Java, Ruby を始め 35 個に及ぶ言語の環境下でも利用可能である.

4 実装内容

4.1 IRI から URI への変換手順

curl で IRI を指定しリソースを正しく取得するためには, IRI の仕様 [1] に沿って IRI から URI に変換する必要がある. 変換手順は以下に示す. また 図 1 では http プロトコルを例に説明する. ただし, 他のプロトコルも同様に適用できる.

手順 1 入力文字コードの指定・取得 (4.2 節)

手順 2 入力の文字コードを UTF-8 に変換 (4.3 節)

手順 3 UTF-8 に変換された IRI をパーセントエンコーディング (4.4 節)

手順 4 IRI のドメイン名部分をパーセントデコード後, ACE (ASCII Compatible Encoding) 変換により英数字に変換 (4.5 節)

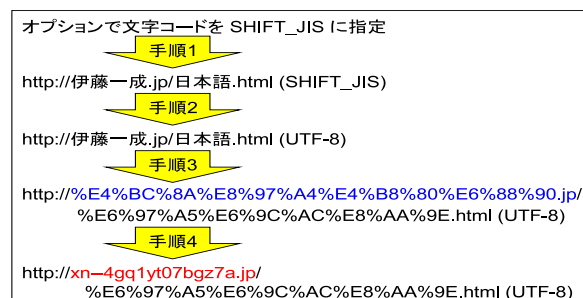


図 1: IRI から URI への変換例

Internationalization of the Data Transfer Tool curl – Processing IRIs

Takeo KOBAYASHI, Kazunari ITO
and Martin J. DÜRST

College of Science and Engineering, Aoyama Gakuin University
5-10-1 Fuchinobe, Sagami-hara, Kanagawa 229-8558, Japan
takeo@sw.it.aoyama.ac.jp, {kaz, duerst}@it.aoyama.ac.jp

以下、手順 1～4 までを 4.2 節～4.5 節まで順に説明する。

4.2 手順 1: 入力文字コードの指定・取得

curl に入力された IRI の文字コードを UTF-8 に変換するためには、IRI の文字コードを知る必要がある。しかし、IRI は世界中の様々な文字コードを使用可能な上、IRI 自体の文字列も短いため文字コードを自動判別することは不可能である。そこで、実装では入力された IRI を判別するのではなく、文字コードを指定する場合と、指定しない場合と 2 通りに分ける。

文字コードを指定する場合は、curl のオプションで文字コードを指定できるようにする。現在 curl には、3 章で述べたように数多くの機能が存在するため、それに伴いオプションも多数存在する。そこで、実装では curl のオプションとして使用されていない大文字の“Z”を、文字コード指定のオプションとした。実際に curl で IRI を入力する際に、文字コードを指定する場合には、以下のようにコマンドを入力する。

```
curl -Z 文字コード IRI
```

一方、特に文字コードを指定しない場合は、システムのデフォルトの文字コード情報を取得し、それを UTF-8 に変換する前の文字コードとして利用する。これは、通常、IRI がデフォルトの文字コードで curl に入力されるためである。MS-Windows 上では、システムのコードページを取得し、Linux/Unix では、ロケール設定から文字コードを取得する。

4.3 手順 2: UTF-8 への変換

UTF-8 への変換には、GNU が提供する文字コード変換ライブラリである iconv [4] を利用した。iconv は、入力文字列を指定した別の文字コードの文字列に変換するためのライブラリである。iconv を用いて文字コードを変換する場合、変換する前の文字コードを指定する必要がある。変換前の文字コードには、手順 1 で決定した文字コードを利用する。

4.4 手順 3: パーセントエンコーディング

パーセントエンコーディングは、URI で使用できないバイトを % 記号と 16 進数 2 桁に変換する処理であるが、すでにこの機能は curl に実装されているため、これをそのまま利用する。

4.5 手順 4: IRI のドメイン名部分の ACE 変換

IRI の仕様 [1] では、IRI のドメイン名部分は国際化ドメイン名である IDN (Internationalized Domain Name)[5]を採用している。国際化ドメイン名は、名前の通り様々な言語で表現可能である。しかし、DNS のシステム内におけるドメイン名の扱いは英数字のみに限られるため、次に示すように国際化ドメイン名を英数字に変換する必要がある。

```
http://伊藤一成.jp (変換前)
```

```
http://xn--4gq1yt07bgz7a.jp (変換後)
```

この変換は ACE 変換と呼ばれ、“Punycode”というアルゴリズムが用いられる。変換された国際化ドメイン名は“xn--”の 4 文字で始まる英数字で構成される。

IRI のドメイン名の ACE 変換には JPNIC が提供する idnkit ライブラリ [6] を利用した。

5 確認テストについて

実装後、IRI 指定したリソースを curl が正しく取得可能であるかテストを実施した。テストの実施には、昨年度我々が作成した IRI テスト環境 [7] を利用した。テスト環境は、URI や IRI が使用される様々な技術使用に応じた項目と、様々な言語や文字コードの組み合わせからなる。curl の機能は、URI もしくは IRI で指定したリソースを取得するのみであるため、テストでは a 要素の href 属性を用いた純粋なリンク先のリソースを取得可能かテストした。テスト環境は、現在以下のアドレスで Web 上に公開されており、誰でもテスト可能となっている。IRI テストの実施には、ぜひ利用していただきたい。

```
http://www.sw.it.aoyama.ac.jp/2005/iritest/
```

6 まとめと今後の課題

本稿ではデータ伝送ツール curl に対し、URI の国際化識別子である IRI を実装した。また、テストでは実際に IRI で指定したリソースを正しく取得できることを確認した。現在、正常にリソースを取得できるのは C 言語の環境下のみである。今後の課題は、Perl, PHP, Java, Ruby といった他の言語環境下でもリソースを取得できることである。

参考文献

- [1] Martin Dürst and Michel Suignard. Internationalized Resource Identifiers (IRIs), RFC 3987, January 2005.
- [2] Tim Berners-Lee, Roy Fielding, and Larry Masinter. Uniform Resource Identifier (URI): Generic Syntax, RFC 3986, STD 66, January 2005.
- [3] Daniel Stenberg. curl and libcurl. <http://curl.haxx.se/>.
- [4] Inc JFree Software Foundation. libiconv. <http://www.gnu.org/software/libiconv/>.
- [5] Patrik Fältström, Paul Hoffman, and Adam M. Costello. Internationalizing Domain Names in Applications (IDNA) RFC 3490, March 2003.
- [6] Japan Network Information Center. idnkit ライブラリの api. <http://www.nic.ad.jp/ja/idn/>.
- [7] 米川和宏, 伊藤一成, Martin Dürst. IRI テスト環境構築のための枠組み提供及びその応用. 第 68 回情報処理学会全国大会, 2006.