アプリケーションの IPv6 対応ガイドライン
Web アプリケーション編
第 0.1 版

2014 年 5 月 30 日

IPv6 普及・高度化推進協議会
IPv4/IPv6 共存 WG
アプリケーションの IPv6 対応検討 SWG

目次

変更履歴

版	改版日	摘要
0.1	2014 年 5 月 30 日	パブリックコメント版

1. はじめに

2011 年の IPv4 アドレス在庫枯渇を機に、通信事業者や ISP 各社から IPv6 対応のサービス展開が進み、インターネットのユーザ側、サーバ側いずれもこれまでの IPv4 のみでの運用から IPv4 と IPv6 の共存期に入りつつあります。一方で、スマートフォンに代表されるスマートデバイスが普及し、ネットワークを使うアプリケーションは一般的になっています。こうした状況の中、アプリケーションの開発者が目にする情報は、これまでの IPv4 で運営されているネットワークを前提としたものが多く、共存状況で注意すべきことやするべきことが整理されていませんでした。

IPv6 普及・高度化推進協議会 IPv4/IPv6 共存 WG では、2011 年 9 月に「アプリケーションの IPv6 対応検討 SWG」を発足し、IPv4/IPv6 共存期を前提としたアプリケーション開発についての情報整理を行い、アプリケーション開発者に向けた情報発信と情報共有の活動を始めました。この「アプリケーションの IPv6 対応検討 SWG」では、2012 年 12 月に BSD Socket を用いたプログラムの IPv6 対応方法についてまとめた「アプリケーションの IPv6 対応ガイドライン 基礎編」[1]を公開しました。

本文書は、「アプリケーションの IPv6 対応ガイドライン 基礎編」の続編として、Web アプリケーションの IPv6 対応についてまとめた資料です。本文書を通じて Web アプリケーションとして IPv6 を意識しなければいけない箇所や IPv4 との違いをご確認いただき、IPv6 に対応した Web アプリケーション開発にお役立ていただければ幸いです。

検討メンバー

会務担当者以外のメンバーは、氏名の 50 音順に従っている。

氏名	所属
波田野 裕一（部会長）	運用設計ラボ合同会社/日本 UNIX ユーザ会 (jus)
廣海 緑里（部会長）	株式会社インテック
藤崎 智宏（部会長）	日本電信電話株式会社
新 善文	アラクサラネットワークス株式会社
大平 浩貴	株式会社リコー
佐藤 良	株式会社コナミデジタルエンタテインメント
高田 美紀	NTT コミュニケーションズ株式会社
高宮 紀明	NTT ソフトウェア株式会社
渡辺 露文	富士ソフト株式会社

2. 本文書について

本文書は、IPv6 に対応する Web アプリケーションの開発に必要な基礎的事項について解説するものである。

3 章では、アプリケーション開発における IPv6 対応の全体像について解説している。
4 章では、プログラミング言語と実行環境の IPv6 対応状況についてまとめている。
5 章では、通信処理の IPv6 対応についてまとめている。
6 章では、データとして IPv6 アドレスを扱う場合の対応方法について解説している。
7 章では、上記以外の考慮事項について解説している。

3. アプリケーション開発における IPv6 対応

アプリケーションにおける IPv6 対応とは、そのアプリケーションが IPv6 と IPv4 の両方の通信環境で動作することである。今後、数年あるいは十数年という長い期間、IPv4 と IPv6 が共存することが予想される。そのため、当分の間、IPv4 と IPv6 の双方で動くシステムが求められる。システムのライフサイクルを見据え、アプリケーションの保守性を考慮すると、単一のソースプログラムで IPv4 と IPv6 の両方で動作するように開発することが望ましい。

まず、ネットワークプログラミングにおける以下の基礎的事項を押さえておくことが重要である。

- IP アドレスの直書きは行わず、FQDN(Fully Qualified Domain Name：完全修飾ドメイン名)を使用する
- IP アドレスでユーザを識別しない

次に、アプリケーションにおける IPv6 対応の主なポイントとして下記の 3 つの対応を行なう。

- IPv6 対応のプログラミング言語と実行環境を使用する
- 通信部分を IPv6 に対応させる
- データとして IP アドレスを扱う箇所を IPv6 に対応させる

次章以降でそれぞれについて説明する。

4. IPv6 対応のプログラミング言語と実行環境を使用する

アプリケーションを IPv6 に対応させるには、その前提として、使用するプログラミング言語および、フレームワークやライブラリ、ミドルウェア、データベース等の実行環境が IPv6 に対応している必要がある。

アプリケーションを IPv6 対応させる上で、プログラミング言語と実行環境には、「名前解決機構が IPv6 アドレスを適切に扱えること」と「IPv6 で通信できること」の 2 点が求められる。

名前解決機構で IPv6 アドレスを扱うケースとして、通信相手のホスト名から IPv6 アドレスを名前解決すること（正引き）、通信相手の IPv6 アドレスからホスト名を名前解決すること（逆引き）などがある。

「IPv6 で通信できること」とは、何らかの通信方法により通信相手と IPv6 で疎通できることを指す。通信方法の例として、SMTP や HTTP などの既存プロトコルや、ソケット通信により独自で実装したプロトコルがある。

これらはいずれもプログラミング言語や、関連するライブラリ、ミドルウェア、データベースなどのプロダクトにおいてそれぞれサポートされるべきものであるが、実際にはサポート状況に差異があるのが実情である。

アプリケーション開発者は、開発するアプリケーションが提供する機能を考慮し、IPv6 への対応に必要な機能の過不足について、プログラミング言語やプロダクトそれぞれ個別に判断することが必要となる。

5. Web アプリケーションにおける通信処理の IPv6 対応

一般に通信処理はサーバとクライアントの双方で行なわれる。サーバは、接続を受け付け、送信されたリクエストに応じて処理を行い、レスポンスを返す。クライアントは、意図するサーバへ接続し、リクエストを送信し、レスポンスを受信する。通信処理の IPv6 対応では、IPv4 と IPv6 の両方で通信できることを目指す。そのため、サーバの IPv6 対応としては、IPv4 および IPv6 で接続を受付けることが必要不可欠であり、クライアントの IPv6 対応としては、IPv4 および IPv6 で意図するサーバへ接続することが必要不可欠である。また、IPv4/IPv6 のいずれかで通信できない状況も想定する必要がある。

IPv6 環境および IPv4/IPv6 共存環境ではクライアントおよびサーバが複数の IP アドレスを持つことがある。このような場合に、クライアントからサーバへ

接続するに際し、複数あるクライアントのアドレスの中のいずれを始点アドレスとして選択し、複数あるサーバのアドレスの中のいずれを終点アドレスとして選択するかを Web アプリケーション側で予測できないことを考慮する必要がある。Web アプリケーション側では特定のアドレスに依存したシステムを構成するべきではない。

Web アプリケーションの場合、サーバ／クライアント共に既存のプロダクトを使用することが多いと想定される。広く使用される Apache HTTP Server や Microsoft Internet Information Service(IIS)といった Web サーバプロダクトは IPv6 に対応しており、クライアントである Web ブラウザも主要なものは IPv6 に対応している。

ソケットを用いてサーバもしくはクライアントのソフトウェアを開発する場合には、本ガイドラインの姉妹書「アプリケーションの IPv6 対応ガイドライン 基礎編」[1]にて BSD Socket を用いたプログラムの IPv6 対応方法を説明しているのでご参照いただきたい。

5.1. IPv6 アドレスの名前解決

アプリケーションの通信においては、IP アドレスではなく FQDN で接続先を指定することが望まれる。FQDN で接続先を指定して IPv6 で通信を行うには、DNS にて FQDN から IPv6 アドレスが名前解決できることが不可欠であり、以下の 2 点が必要である。

- 接続先サーバが属するドメインの権威 DNS サーバ上で、接続先サーバの AAAA レコードに IPv6 アドレスが登録されている
- クライアントから接続先サーバの AAAA レコードが引ける

Web アプリケーションの開発においては FQDN の IPv6 アドレスが正しく名前解決できることを確認すべきである。

5.2. 通信の試行順序

クライアントあるいはサーバが複数の IP アドレスを持つ場合、クライアントが通信に使用する始点アドレスと終点アドレスを選択する必要がある。これらのアドレス選択に関するルールが RFC6724 [2] で定義されている。

この RFC6724 におけるデフォルトのルールに従うと IPv6 アドレスが IPv4 アドレスより優先される。すなわち、IPv6 通信が IPv4 通信より優先される。ただし、デフォルトを変更している実装や RFC6724 に準拠していない実装ではこの優先順位が変わる場合がある。

5.3. フォールバックとその解決

プロトコル非依存のネットワークプログラミング手法においては伝統的に、FQDN を名前解決して得られた IP アドレスのリストを元に相手先への通信を試みる。このリストには IPv4 と IPv6 のアドレスが混在し、更に IPv6 アドレス、IPv4 アドレスそれぞれについて複数のアドレスが得られる可能性がある。クライアントアプリケーションは得られたアドレスリストの先頭から順に接続を試すが、リストの先頭の IP アドレスが接続可能とは限らない。ネットワークの接続やサーバが提供するアプリケーションサービスに不具合等があり、相手先の IP アドレスへの到達性がない場合、クライアントアプリケーションは接続に失敗したことを検出し、得られたリストに記載されている次の IP アドレスでの接続を試す。本文書ではこの一連の接続動作をフォールバックと呼ぶ。

フォールバックが発生する要因は様々であるが[1]、発生すると接続成功（アプリケーションサービスの開始）までの時間が長くなり、サービス利用における快適性が損なわれることがある。

これをできるだけ発生させないためには、IP の接続性を健全に保つ、アプリケーションサービスを行っていないサーバアドレスを DNS に登録しないなどの対策を実施する必要がある。

[1] フォールバックは、ICMP などによる経路不達通知の認識、TCP RST によるセッション切断、TCP SYN に対する応答のタイムアウトなどによって発生する

6. データとして IPv6 アドレスを扱う箇所の対応

アプリケーションによっては、接続履歴の記録やアクセス制御のために IP アドレスをデータとして扱うことがある。IPv6 アドレスは IPv4 アドレスとビット長や表記法などが大きく異なるためその処理には注意が必要である。

表 6-1 に IPv4 アドレスと IPv6 アドレスの違いを示す。

表 6-1 IPv4 アドレスと IPv6 アドレスの違い

		IPv4 アドレス	IPv6 アドレス
アドレス長		32 ビット	128 ビット
文字列表記	表記法	8 ビットずつ区切り、10 進数で表記	16 ビットずつ区切り、16 進数で表記
	区切り文字	.（ドット）	:（コロン）
	文字列長	15 文字以内	39 文字以内

IPv6 アドレスの文字列表記では、16 進数を用いること、区切り文字に「:」（コロン）を用いること、文字列長 39 文字以内となることが IPv4 アドレスと異なる点である。この違いが、IP アドレスをデータとして扱う DB への格納やログ出力に影響する。

IPv6 アドレスの表記法

特段の事情がない限り RFC5952 [3]に文書化されている IPv6 アドレス推奨テキスト表記ルールに従い省略して表記する（以下、「省略表記」）か、省略を行わず表記する（以下、完全表記）が望ましい。ライブラリを用いて IPv6 アドレスを扱う場合、利用するライブラリが RFC5952 に対応していることを確認する必要がある。

IPv6 アドレスの区切り文字「:」が IP アドレスとポート番号との区切り文字として使われている場合があるので注意を要する。

IPv6 アドレスの文字列長

IPv6 アドレスの文字列長について、下記の例外があるので留意が必要である。

- アプリケーションでリンクローカルアドレスを取り扱う場合、複数のインターフェースを有するとき、同一のアドレス範囲（fe80::/10）を使用する。そのため、fe80::1%eth1 のように IPv6 アドレスにゾーン ID（スコープ ID）を付加して対象のインターフェースを識別する場合がある。この ID の文字数分 IP アドレスの最大文字数が増えることの考慮も必要となる。
- 一部の特殊アドレス（IPv4 射影アドレスなど）では 39 文字を超える場合がある。

6.1. データベースへの格納

データベースに IPv4 アドレスを格納する際に、多くの場合は以下の方式が用いられてきた。

- 1 つの文字列として扱い、15 文字までの文字列（VARCHAR(15)）に格納
- 4 つの整数型を（INT）に格納
- 1 つの 32 ビット以上の整数型（INT）に格納

これらいずれの方法も、そのままでは IPv6 アドレスは格納できずエラーとなり、IPv6 に対応する場合には変更が必要となる。

文字列型で格納する場合

IPv6 アドレスは、その最大文字数である 39 文字の文字列が格納できるよう VARCHAR(39)に格納するべきである。プレフィックス長も一緒に格納する場合には、区切り文字である「/」（スラッシュ）を含めて 4 文字分追加して VARCHAR(43)に格納すべきである。

データベースで定義されているデータ型で格納する場合

IPv6 アドレスに対応するデータ型がデータベースで提供されている場合は、そのデータ型を使う事で入力時のエラー検査や専用の演算子の利用が可能になる事が多いため、それを使う事が望ましい。

例えば PostgreSQL ではネットワークアドレス型[4]とネットワークアドレス関数が提供されている。

6.2. ログ出力への影響

IP アドレスをログに出力する場合、アドレス部分の文字列長が長くなる等の影響がある。

図 6-1 に Apache HTTP Server のログの例を示す。1 行目が IPv4 の、2 行目が IPv6 のアクセス情報である。先頭の IP アドレス表記が変わっている。Web サーバログ解析プログラムを利用している場合は IPv6 アドレス対応について確認する必要がある。AWStats[5]、Webalizer[6]といった OSS ログ解析プログラムは IPv6 アドレスについて問題なく処理できるが、確認が必要である。また、ログ解析プログラムを自作している場合や、ログ出力を自作している場合には影響の有無を確認するべきである。

1	192.0.2.10 - - [06/Jan/2014:15:42:15 +0900] "GET / HTTP/1.1" 403 5039 "-" "Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/31.0.1650.63 Safari/537.36"
2	2001:db8:0:1::10 - - [06/Jan/2014:15:42:23 +0900] "GET / HTTP/1.1" 403 5039 "-" "Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/31.0.1650.63 Safari/537.36"

図 6-1 Apache HTTP Server ログの例

6.3. Web フォームへの入力

Web サイトの入力フォームにおいて、入力値に含まれる文字があらかじめ明らかになっている場合は、脆弱性対策および Web アプリケーションへの異常値混入防止のため、入力値を検証するべきである。例えば、IP アドレスを扱う

[4] PostgreSQL ネットワークアドレス型については、PostgreSQL のマニュアル（http://www.postgresql.jp/document/9.2/html/datatype-net-types.html）を参照

[5] http://awstats.sourceforge.net/

[6] http://www.webalizer.org/news.html

場合には、表 6-1 の「文字列表記」欄に従って IP アドレスとして取りうる値であることを検証することが望ましい。

入力値の検証は、Web サービスのブラウザ側（HTML5 のクライアントサイドフォームバリデーション）とサーバ側の処理プログラムの 2 箇所での実施が可能である。

プログラミング言語によっては、IPv6 用のアドレス処理ライブラリを利用することにより上記の検証が可能な場合がある。例えば PHP では、PEAR[7]で提供されている Net_IPv6 パッケージに含まれる *Net_IPv6::checkIPv6()*がこれにあたる。ライブラリの関数でアドレスの検証が行える場合にはライブラリを活用するべきである。

6.4. IPv6 アドレスの検索や整列

IP アドレスの検索や整列を行う場合には、省略表記の存在に注意が必要である。完全表記で格納された IP アドレスに対して省略表記で検索を行なってもマッチしない。また省略表記されたアドレスの整列も注意が必要である。例えば 2001:db8:0:1::1:1、2001:db8:0:2::1、2001:db8:0:1::50、2001:db8:0:10::1 の 4 つのアドレスを 128bit の数値としての昇順（以下「アドレス昇順」）で整列する際に、省略表記のまま文字列として扱うと期待に反した結果になる。この例を図 6-2 に示す。

[7] PHP の拡張ライブラリ集

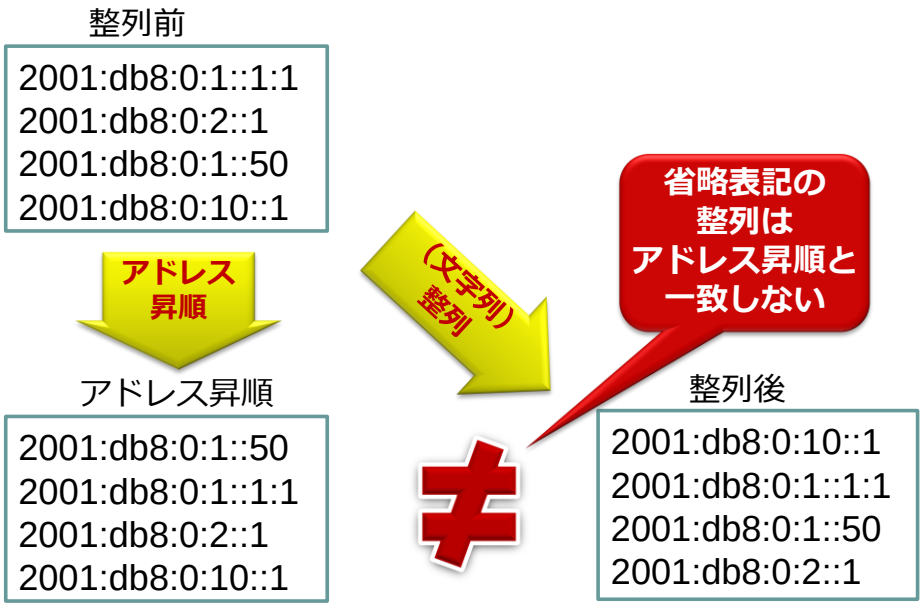

図 6-2　IPv6 アドレスの整列に関する注意点

完全表記で整列することにより、アドレス昇順にすることができる（図 6-3）。

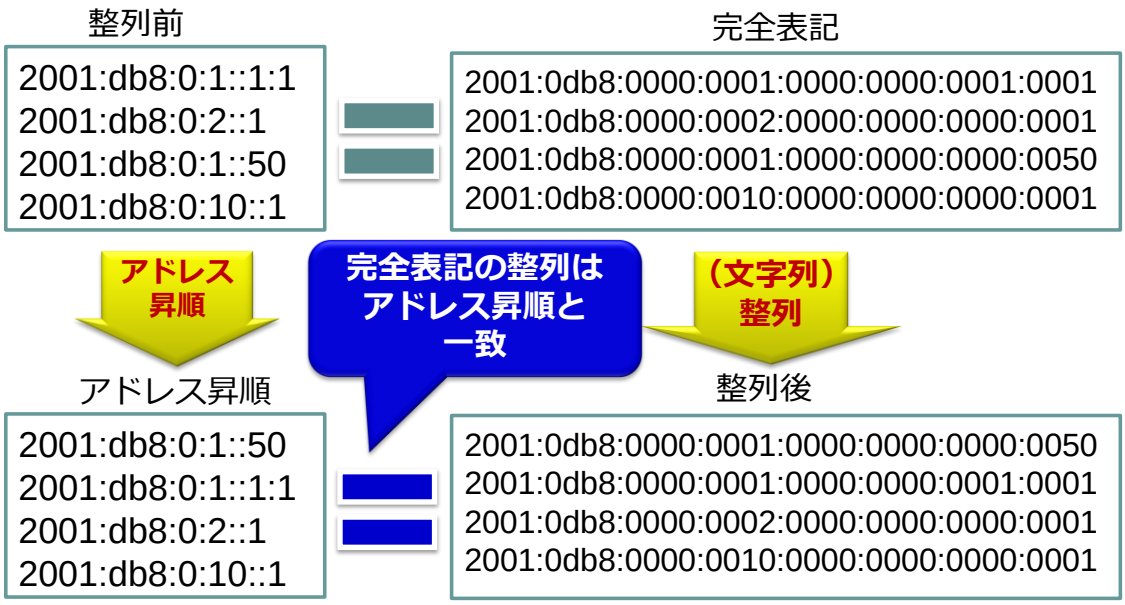

図 6-3　IPv6 アドレスのソートは完全表記で実行

IP アドレスの検索・整列を正しく行うためには、プログラミング言語やデータベースなどの実行環境に応じて、IP アドレスを以下のように取り扱う必要がある。

- IP アドレス型が定義されている場合: IP アドレス型を使用する
- IP アドレス型が定義されていない場合: 完全表記を使用する

7. その他の考慮事項

7.1. DMZ の IPv6 対応方式と Web アプリケーションがアクセス元 IP アドレスを取得する方法

Web アプリケーションにてクライアントの IP アドレスを取得する必要がある場合、環境変数を参照することが多いが、ネットワークの形態によっては、必ずしもその IP アドレスがクライアントのものとは限らない。リバースプロキシやプロトコル変換装置を用いた場合に、この事象が発生する。

IPv6 普及・高度化推進協議会 セキュリティ WG が作成した「IPv6 対応セキュリティガイドライン（第 1.0 版）」[4] において、セキュリティ確保の観点から、企業ネットワークの DMZ を IPv6 に対応させる際の構成方法、考慮点、懸念点、対策等が解説されている。

上記ガイドラインでは、DMZ の IPv6 対応モデルとして、パラレルスタック、デュアルスタック、トランスレーションの 3 モデルが定義されている（図 7-1）。パラレルスタックモデルおよびデュアルスタックモデルにおいては、クライアントから直接 Web サーバへ IPv6 でアクセスされることになる。しかし、トランスレーションモデルにおいては、トランスレータがプロトコル変換を行うため、Web サーバへのアクセスはトランスレータ経由となる。
そのため、トランスレーションモデルでは “X-Forwarded-For: ” ヘッダを付与できる装置を用い、アプリケーション側でこのヘッダの値を環境変数から取得するといった手段が必要となる。

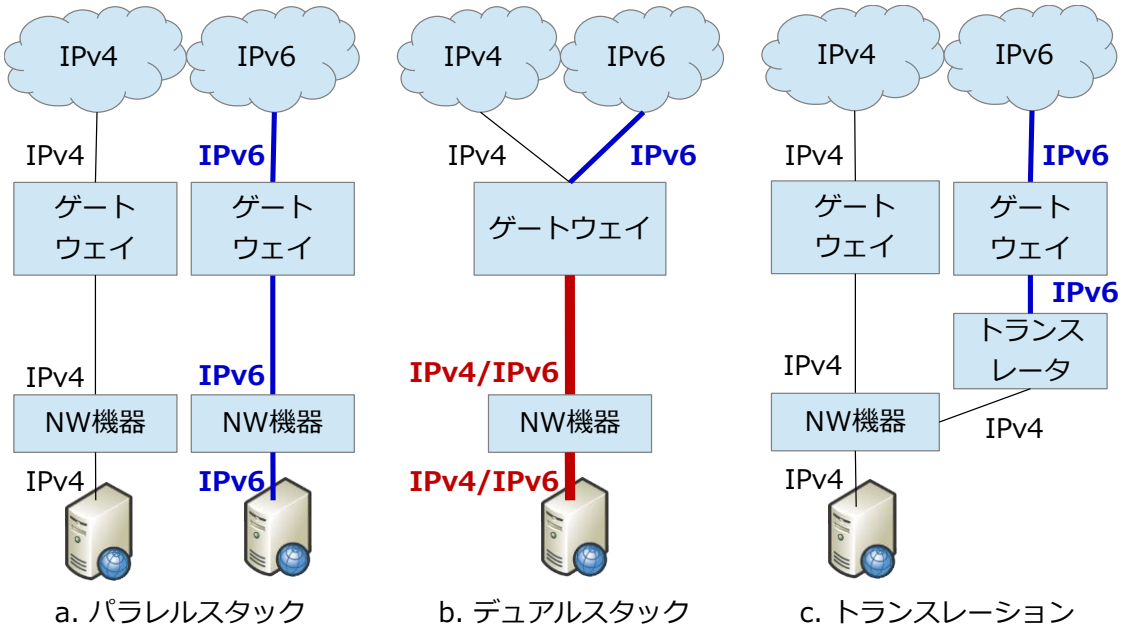

図 7-1 DMZ の IPv6 対応モデル

なお、上記ガイドラインではアプリケーションを IPv6 対応させる場合に考慮すべきセキュリティ関連項目についても言及されており、一読することをおすすめする。

7.2. Web ページへの IP アドレスの埋め込み

Web サイトによっては、リンク先などとして IPv4 アドレスを含んだ URL がページ内に記載されているケースが見受けられる。この場合、IPv6 のみの環境からはリンク先にアクセスできないため、IPv4 に依存するようなリンク先 URL の記述は避けるべきである。

8. おわりに

本文書は、Web アプリケーションの IPv6 対応に必要な基礎的事項をまとめた資料である。IPv6 普及・高度化推進協議会は日本国内の Web コンテンツの IPv6 対応の促進を目的とし、関係諸氏のご協力を得て本文書を作成した。本文書を今後のアプリケーション開発にお役立ていただければ幸いである。

参考文献

[1]
アプリケーションの IPv6 対応ガイドライン 基礎編
IPv6 普及・高度化推進協議会 IPv4/IPv6 共存 WG アプリケーションの IPv6 対応検討 SWG
December 2012
http://www.v6pc.jp/jp/entry/wg/2012/12/ipv610.phtml

[2]
RFC6724
Default Address Selection for Internet Protocol Version 6 (IPv6).
D. Thaler, Ed., R. Draves, A. Matsumoto, T. Chown.
September 2012.

[3]
RFC5952
A Recommendation for IPv6 Address Text Representation
S. Kawamura, M. Kawashima
August 2010

[4]
IPv6 対応セキュリティガイドライン（第 1.0 版）
IPv6 普及・高度化推進協議会 セキュリティ WG
September 2012
http://www.v6pc.jp/jp/entry/wg/2012/09/ipv610_1.phtml