



System i

ネットワーキング

OSPF (Open Shortest Path First) サポート

バージョン 6 リリース 1







System i

ネットワーキング

**OSPF (Open Shortest Path First) サポート**

バージョン 6 リリース 1

#### お願い

本書および本書で紹介する製品をご使用になる前に、31 ページの『特記事項』に記載されている情報をお読みください。

本書は、IBM i5/OS (5761-SS1) のバージョン 6、リリース 1、モディフィケーション 0 に適用されます。また、改訂版で断りがない限り、それ以降のすべてのリリースおよびモディフィケーションに適用されます。このバージョンは、すべての RISC モデルで稼働するわけではありません。また CISC モデルでは稼働しません。

IBM 発行のマニュアルに関する情報のページ

<http://www.ibm.com/jp/manuals/>

こちらから、日本語版および英語版のオンライン・ライブラリーをご利用いただけます。また、マニュアルに関するご意見やご感想を、上記ページよりお送りください。今後の参考にさせていただきます。

(URL は、変更になる場合があります)

お客様の環境によっては、資料中の円記号がバックスラッシュと表示されたり、バックスラッシュが円記号と表示されたりする場合があります。

原 典： System i  
Networking  
Open Shortest Path First (OSPF) support  
Version 6 Release 1

発 行： 日本アイ・ビー・エム株式会社

担 当： ナショナル・ランゲージ・サポート

第1刷 2008.2

© Copyright International Business Machines Corporation 2008. All rights reserved.

---

## 目次

### OSPF (Open Shortest Path First) サポート

ポート	1
V6R1 の新機能	1
「OSPF (Open Shortest Path First) サポート」の PDF ファイル	1
OSPF (Open Shortest Path First) サポートの概念	2
OSPF 経路指定ドメインおよびエリア	2
OSPF エリア集約	5
リンク状態通知	7
リンク状態レコードの経時処理	8
OSPF のパケット・タイプ	8
IPv6 用の OSPF	10
OSPF インターフェース	11
OSPF の Point-to-Point リンク	12

i5/OS OSPF 認証	12
i5/OS OSPF ジョブ・トレースの使用可能化	14
OSPF (Open Shortest Path First) サポートの作業	14
OSPF ネットワーキング用の i5/OS の構成	14
i5/OS 上で OSPF 用に TCP/IP を使用可能化する	15
OSPF (Open Shortest Path First) サポートの参照	16
OSPF (Open Shortest Path First) API およびコマ ンド	16
シナリオ: OSPF の構成	18

### 付録. 特記事項. 31

プログラミング・インターフェース情報	33
商標	33
使用条件	33



---

## OSPF (Open Shortest Path First) サポート

i5/OS® サポートには、OSPF (Open Shortest Path First) プロトコルが組み込まれています。OSPF は、ネットワーク経路指定を行う、リンク状態型で階層型の Interior Gateway Protocol (IGP) です。

この一連のトピックでは、OSPF の構成、認証方式、Point-to-Point リンク、パケット・タイプ、および OSPF 自律システム (AS) のエリアへの分割に関する、i5/OS のサポートを説明します。また 3 つのシナリオも含まれています。TCP/IP スタック上の OSPF 経路を説明するシナリオ、マルチパス経路を説明するシナリオ、および i5/OS API を説明するシナリオです。

注: コーディング例をご使用になることで、28 ページの『コードに関するライセンス情報および特記事項』の条件に同意されたものとみなされます。

### 関連情報

OSPF (Open Shortest Path First)

---

## V6R1 の新機能

新しい i5/OS OSPF (Open Shortest Path First) サポートにより、ご使用のシステムを OSPF ネットワーク環境に参加できるように構成する一連の System i コマンドが組み込まれます。

---

## 「OSPF (Open Shortest Path First) サポート」の PDF ファイル

本書の PDF ファイルを表示および印刷することができます。


本書の PDF 版を表示またはダウンロードするには、「OSPF (Open Shortest Path First) サポート」を選択します。

### PDF ファイルの保管

表示用または印刷用の PDF ファイルをワークステーションに保存するには、次のようにします。

1. ご使用のブラウザの PDF リンクを右クリックする。
2. PDF をローカルに保管するオプションをクリックする。
3. PDF を保存したいディレクトリーに進む。
4. 「保存」をクリックする。

### Adobe® Reader のダウンロード

これらの PDF を表示または印刷するには、システムに Adobe Reader がインストールされていることが必要です。Adobe Web サイト ([www.adobe.com/products/acrobat/readstep.html](http://www.adobe.com/products/acrobat/readstep.html))  から、無償コピーをダウンロードできます。

---

## OSPF (Open Shortest Path First) サポートの概念

OSPF ネットワーク環境で i5/OS サポートを使用する前に、OSPF 参加の要件を理解することが重要です。このトピックでは、エリアおよびインターフェースという OSPF の概念について説明し、ご使用のシステムでこれらを構成する i5/OS コマンドを示します。

OSPF は、リンク状態アルゴリズムを使用して、OSPF ネットワーク環境内の中央ノードと他のすべてのノードの間の最短距離を計算し、使用します。各ノードは、自身のリンクの状況、および自身とすべての隣接ノードによって作成された、完全なトポロジー、または経路指定構造を記述して送信します。

i5/OS OSPF サポートは、OMPROUTED サーバーを使用して、システムを OSPF ネットワークのノードまたは参加者として登録します。各 OSPF ノードまたは経路宛先は、小数点付き 10 進数フォーマットの IPv4 アドレスまたは IPv6 アドレスです。OMPROUTED サーバー・デーモンは UNIX® 標準に準拠し、POSIX に対応しています。QSYSWRK サブシステム内で QTOOROUTE ジョブを実行するために、i5/OS プラットフォームで提供される TCP/IP プロトコルを使用します。QTOOROUTE ジョブは IPv4 操作と IPv6 操作の両方をサポートします。

OSPF の概念には、エリア、およびエリア内に定義されるルーターへのインターフェースが含まれます。OSPF プロトコルは、最初の隣接者探索、および IP アドレスのネットワーク全体でのリンク状態通知 (LSA) の送受信に、パケット・タイプを使用します。OSPF プロトコルは、さまざまなネットワークのタイプをサポートするために、さまざまな種類のルーター・タイプおよびバス・タイプを使用します。i5/OS OSPF サポートを使用する前に、OSPF 経路指定ドメインをエリアに分割する方法、エリア集合および LSA などの他の OSPF 機能を使用する方法を決めます。

i5/OS サポートは、OSPF 経路指定を使用する時、IPv4 と IPv6 では異なります。このトピックでは、これらの違いについて、および TCP/IP OSPF インターフェースを提供するための i5/OS 回線記述の使用など、他の重要な情報についても説明します。

## OSPF 経路指定ドメインおよびエリア

OSPF 経路指定は、経路指定ドメイン内のエリア間に定義されている関係によって決まります。エリア・タイプおよび i5/OS 経路指定隣接者を定義するには、CL コマンドを使用します。

OSPF プロトコルにおいて、自律システム (AS) とは共通の管理下にある IP ネットワークの集合であり、共通の経路指定方針を共用し、ただ 1 つの経路指定プロトコルのみを実行します。AS を形成するルーターおよびリンクは、エリアと呼ばれる論理グループになります。エリアは、固有の割り当て番号で識別され、AS は少なくとも 1 つのエリアを定義する必要があります。

AS が複数のエリアに分割される場合、それらのエリアはエリア境界ルーター (ABR) として指定されたルーターで相互接続されます。本質的に ABR は異なる OSPF エリアに接続されている異なる OSPF インターフェースを持つため、複数のエリアで作動することが可能です。ABR は、関連付けられている各エリアごとにリンク状態データベースのコピーを保持します。

エリア内のすべてのルーターは、自律システム・トポロジー・データベースの同一コピーを持ちます。各ルーターは、Dijkstra アルゴリズムと呼ばれる、Shortest Path First を計算するスパンニング・ツリー・アルゴリズムを使用して、自身の経路指定テーブルを計算します。

OSPF はマルチパスをサポートします。これは、OSPF が、同一の宛先またはシステムに対して複数の経路をサポートすることを意味します。新しいリンク状態更新パケットが受信されると、ツリー全体が再計算されます。その他のマルチパス計算についての考慮事項は、次の通りです。

- 経路指定ツリーは、複数の等価コスト経路をすべて保存します。



- 経路指定ツリーは、同一宛先に対して複数の経路が存在するとき、保存用には最短パスのみを選択します。
- コストが同一である 2 つの経路が同一の宛先に対して存在するときは、両方のコストがスタックに追加されます。
- OSPF は必ず、最低コストの経路を TCP/IP スタックに追加します。
- 同一宛先に対する複数のマルチパス経路のコストが同一である場合、これらのマルチパス経路は TCP/IP 経路指定テーブルに追加されます。

OSPF エリア内の各ルーターはリンク状態通知 (LSA) を発信します。これは、あるルーターのトポロジーを、同一の OSPF エリア内の他のすべてのルーターに転送するための、OSPF 通信の基本的な方法です。発信された各リンク状態通知は、各ルーターのリンク状態データベースに保管されます。このデータベースはルーター間で同期が取られ、OSPF エリア内の各ルーターはリンク状態データベースの同一コピーを持つこととなります。

ABR は、ABR にとっては認識されるがエリア外にある他のすべての宛先に対する通知を行うために、別の要約 LSA を使用します。

バックボーン・エリアとして指定されるエリアは、OSPF AS における必要条件です。バックボーン・エリアは OSPF AS 内のすべてのエリア間の経路指定情報を提供する役目を持つため、すべてのエリアがバックボーン・エリアに接続される必要があります。このバックボーンあるいはツリー状の設計は、バックボーン・ネットワーク・セグメントだけを全探索することにより、ある ABR から他の ABR への経路指定をサポートします。

エリアは、4 オクテットの長さの ID で識別されます。バックボーン・エリアには、必ず予約済み ID である 0.0.0.0 が割り当てられます。AS 内の他のすべてのエリアは、固有の ID を持つ必要があります。本質的に OSPF エリアはネットワークの集合であって、ルーターの集合ではありません。バックボーン・ネットワーク・セグメントは、0.0.0.0 で識別されるエリアに属する IP サブネットです。バックボーンに物理的に接続されていないエリアは、OSPF 仮想リンクを使用してバックボーン ABR に論理的に接続されません。OSPF エリア追加 (ADDOSPFAREA) コマンドは、i5/OS バックボーン・エリアおよび非バックボーン・エリアを OSPF 構成に追加します。

経路指定ツリーのサイズによりますが、OSPF は数値計算主体であるため、アプリケーション・サーバーには非バックボーン・エリアを入れるのが一般的です。あるエリアのトポロジーは、経路指定のオーバーヘッドを削減するため、AS の他のエリアからは隠蔽されます。これは、送信される経路指定の更新が少なければ少ないほど、計算して維持する経路指定ツリーが小さくて済むためです。これはまた、ストレージと CPU の使用量を削減することにもなります。

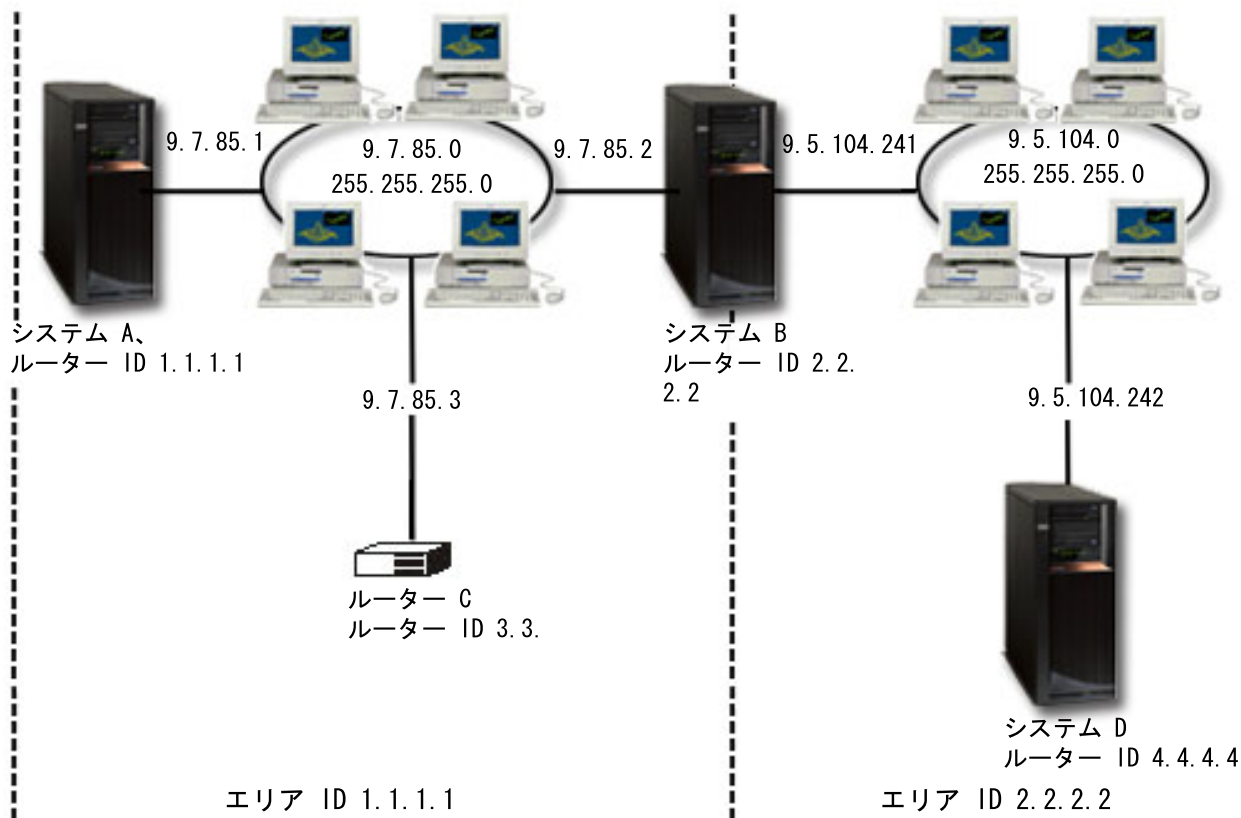


図1. OSPF 経路指定ドメインをエリアに分割する。

例えば、バックボーン・エリアが IP アドレス 9.7.85.0 にあるとします。3 つの非バックボーン・エリア、すなわちエリア 1.1.1.1、エリア 2.2.2.2、およびエリア 3.3.3.3 が接続されており、エリア 3.3.3.3 が仮想リンクを介してバックボーン・エリアに接続されているとします。

## OSPF バックボーン・エリア、非バックボーン・エリア、スタブ・エリア

バックボーン・エリアの要件は、サブネット・エリアからバックボーン・エリアへ経路指定情報の配布が行われること、およびその後バックボーンがこの情報を他の非バックボーン・エリアに送信することです。さまざまなエリアを介して経路指定情報を配布するために、ABR は、バックボーン・エリアから受信された要約 LSA という経路指定テーブルの項目から、計算を行います。その後 ABR は、この経路指定情報をやはり LSA 要約として、接続されている非バックボーン・エリアに送信します。

仮想リンクは ABR 間に構成され、また非バックボーン・エリアと OSPF バックボーン・エリア 0.0.0.0 との間に物理接続がない場合には、仮想リンクがそれらの間の Point-to-Point インターフェースと見なされます。仮想リンクが非バックボーン・エリアを介して隣接者との間に完全な隣接性に到達した場合、この非バックボーン・エリアは通過エリアとして通知されます。通過エリアは、自身のエリア内から発信されていないトラフィック、または自身のエリアを宛先にしていないトラフィックを通過させます。

スタブ・エリアとして構成されているエリアは、外部 LSA による外部経路指定情報のフラッドイング (全ポートにパケットを送出) はサポートしません。スタブ・エリアは、CPU サイクルおよびリンク状態データベースのサイズを削減するために、外部の経路指定を省略して構成されます。i5/OS スタブ・エリアが外

部の経路指定情報を使用できる唯一の方法は、デフォルト経路です。スタブ・エリアを構成するには、ADDOSPFARA コマンドまたは OSPF エリア変更 (CHGOSPFARA) コマンドで次のようにスタブ・エリア・オプションを \*YES に設定します。

```
ADDOSPFARA AREA('1.1.1.1') STUB(*YES)
```

別のタイプのエリアは、トータル・スタブ・エリアと呼ばれ、受信する経路指定情報はスタブ・エリアよりも少なく、デフォルト経路のみを受信します。トータル・スタブ・エリアは、経路指定ツリーを作成するために必要となる数値計算操作を最小化するとともに、トポロジー・データベースが必要とするストレージ要件も最小化します。エリアをトータル・スタブ・エリアとして構成するには、ADDOSPFARA コマンドまたは CHGOSPFARA コマンドで次のように IMPORT パラメーターを \*YES に設定します。

```
ADDOSPFARA AREA('1.1.1.1') STUB(*YES) IMPORT(*YES)
```

OSPF *Not-So-Stubby* エリア (NSSA) タイプは、i5/OS ではサポートされません。

#### 関連概念

8 ページの『OSPF のパケット・タイプ』

OSPF は、隣接性を確立および維持し、要求を送受信し、隣接者間でリンク状態通知 (LSA) を高い信頼度で確実に配信し、リンク状態データベースを記述するために、隣接者にパケットを送信します。リンク状態データベースは、エリア・ルーターが送受信するすべての LSA を元に生成されます。その後で、リンク状態データベースは、Shortest Path First (SPF) アルゴリズムを使用して最短パス・スパンニング・ツリーを計算するために使用されます。

## OSPF エリア集約

OSPF 集約は、共通アドレスを持つ経路のグループを、単一の経路指定テーブル項目に結合します。i5/OS は、OSPF 集約を実現するためにサブネット・マスクをサポートします。

OSPF 集約は、経路指定テーブルのサイズを削減するために使用されます。説明のために、1 つのバックボーン・エリアと 1 つの非バックボーン・エリアを持つ構成を考えてみましょう。

- バックボーン・エリア 0.0.0.0
  - ルーター 1
  - ルーター 3
  - ルーター 4
- 非バックボーン・エリア 1.1.1.1
  - ネットワーク・セグメント 9.92.2.0、サブネット・マスク 255.255.255.0
  - ネットワーク・セグメント 9.92.1.0、サブネット・マスク 255.255.255.0
  - ABR2、バックボーンとこの非バックボーン・エリアとの間に定義される

この構成において、非バックボーン・エリア 1.1.1.1 には 2 つのネットワーク・セグメントがありますが、これらは OSPF 集約により、バックボーン・エリアに対してはただ 1 つのセグメント 9.92.0.0/16 として通知されます。ABR2 は、バックボーン・エリアに転送される要約 LSA の数を削減するために、範囲を通知するようカスタマイズされます。次のように OSPF 範囲追加 (ADDOSPFRRNG) コマンドを使用することにより、新しい 9.92.0.0/16 範囲が 9.92.2.0 および 9.92.1.0 の両方のセグメントをカバーします。

```
ADDOSPFRRNG AREA('1.1.1.1') IPADDRNG('9.92.0.0') SUBNETMASK('255.255.0.0') ADVERTISE(*YES)
```

ADDOSPFRRNG コマンドまたは OSPF 範囲変更 (CHGOSPFRRNG) コマンドを使用して、通知オプションを \*YES に設定します。

ADVERTISE パラメーターを \*YES に設定することにより、LSA で経路要約が行われます。範囲または経路集約が指定されない場合、ABR2 が送信する非バックボーン・エリアからの LSA 要約は、次のようになります。

- 9.92.1.0
- 9.92.2.0

経路集約を指定し、ABR2 に対して範囲を追加することにより、非バックボーンに関して送信される LSA 要約は次の通りになります。

- 9.92.0.0

範囲 9.92.0.0/16 を指定する場合、使用されるサブネット・マスクは、各セグメントごとに以下の例で示すサブネット・マスクとは異なります。サブネット・マスク 255.255.0.0 が、9.92.0.0/16 の中のすべてのサブネットに共通であるために使用されます。

新しいセグメントがサブネット・マスクによって正しく表されるように、範囲の追加は注意深く行う必要があります。このことを説明するために、構成例に新しい非バックボーン・エリア 2.2.2.2 を追加して考えてみましょう。

- バックボーン・エリア 0.0.0.0
  - ルーター 1
  - ルーター 3
  - ルーター 4
- 非バックボーン・エリア 1.1.1.1
  - ネットワーク・セグメント 9.92.2.0、サブネット・マスク 255.255.255.0
  - ネットワーク・セグメント 9.92.1.0、サブネット・マスク 255.255.255.0
  - ABR2
- 非バックボーン・エリア 2.2.2.2
  - ネットワーク・セグメント 9.92.5.0、サブネット・マスク 255.255.255.0
  - ネットワーク・セグメント 9.92.6.0、サブネット・マスク 255.255.255.0
  - ABR4

集約されたサブネット・マスクは、その背後にあるエリアまたは ABR のセットの背後にあるエリアでサブネットの隣接クラスタリングをサポートします。ただし、代表となるサブネット・マスクが分離されたセグメントにならないことが重要です。一般に、分離されたセグメントとは、エリアに対してローカルなセグメントで、集約の中の該当エリアの外部からは見えません。

この例において、セグメント 9.92.5.0 および 9.92.6.0 が追加されますが、集約されたマスクである 255.255.0.0 が、9.92.1.0 および 9.92.2.0 という番号が付けられたクラスターまたはサブネットに対して継続する場合は、バックボーン・エリアでは、9.92.0.0/16 が ABR2 と ABR4 の両方の背後にあるかのように表示されます。これが分離されたセグメントであり、サブネットが継続セクションではなく、ネットワークの対向側で分割されているということを意味します。集約されたマスクを 255.255.252.0 に変更し、ABR2 の背後のサブネットを 9.92.0.0/22 とし、また ABR4 の背後のサブネットを 9.92.4.0/22 とし、集約します。双方の分離されたエリアにおける実際のサブネット・マスクは、255.255.255.0 のままである可能性があります。ADDOSFPRNG コマンドを使用して次のように異なる SUBNETMASK を指定します。

```
ADDOSFPRNG AREA('1.1.1.1') IPADDRNG('9.92.0.0') SUBNETMASK('255.255.252.0') ADVERTISE(*YES)
```

注: スラッシュの後に数値が続くのは、IP アドレスにおいては標準的なサブネット・マスクの表記であり、クラスレス・ドメイン間ルーティング (CIDR) と呼ばれます。CIDR を使用することにより、例えば経路 9.92.0.0/16 は、ネットワーク 9.92.0.0 用のサブネット・マスクが値 1 に設定された 16 ビットを持つ、という意味になります。したがってこの例ではサブネット・マスクは 255.255.0.0 です。サブネット・マスクには 4 つのオクテットがあり、そのそれぞれが 8 ビットです。しかし最初の 2 つのオクテットだけは、すべてのビットが 1 になります。ADDOSPRNG コマンドの SUBNETMASK パラメーターが、バックボーンに対して通知されるネットワークの範囲を決定します。

## リンク状態通知

i5/OS ルーターは、OSPF ネットワークの参加者として、1 つ以上のリンク状態通知 (LSA) を発信し、自分自身に関する経路指定情報を送信したり、または隣接者からの経路指定情報を受信したりします。この情報は、リンク状態データベースの作成に使用します。

OSPF ネットワーク・タイプは、Point-to-Point、ブロードキャスト・マルチアクセス、非ブロードキャスト・マルチアクセス、Point-to-Multipoint、または仮想リンクのいずれかです。次の LSA が i5/OS でサポートされます。

- ルーター LSA

ルーター LSA は、エリア内のすべてのルーター宛先、インターフェース、および状態について記述します。また、発信元ルーターに仮想リンクがあるか、自律システム境界ルーター (ASBR) であるか、あるいは ABR であるかについても示します。この LSA タイプは、各リンクが Point-to-Point 接続、通過ネットワークへの接続、スタブ・エリアへの接続、または仮想リンクのいずれであるかについても示します。

- ネットワーク LSA

ネットワーク LSA は、ブロードキャスト・ネットワークへの代表接続を記述します。これは代表ルーターによって発信され、代表ルーターに隣接するすべてのルーターのリストを含んでいます。

- 要約 LSA

この LSA は ABR によって作成され、エリア間の経路を記述します。ABR に接続されている他のエリア内部にある各 IP 宛先ごとに、1 つの通知が記述されます。

- ASBR 要約 LSA

この LSA は ABR によって作成され、エリアの外部の AS 境界ルーター (ASBR) へのすべての経路を記述します。AS 境界ルーターの IP アドレスが含まれます。

- 外部 LSA

外部 LSA は AS 境界ルーターによって作成されます。AS 外部の宛先への経路を記述します。

次のタイプの LSA は、i5/OS の IPv6 でのみサポートされます。

- エリア間プレフィックス LSA

この LSA は要約 LSA と類似しています。各エリア間プレフィックス LSA は、エリアに対しては外部であるが自律システムにとってはまだ内部であるプレフィックスについて、記述します。この LSA のフラッド範囲はエリア内です。

- エリア間ルーター LSA

この LSA は ASBR 要約 LSA と類似しています。エリアに対しては外部であるが AS にとっては内部である ASBR 宛先ルーターへのパスを記述します。この LSA のフラッド範囲はエリア内です。

- リンク LSA



i5/OS は、2 つ以上のルーターをサポートする接続リンクごとに、別のリンク LSA を発信します。この LSA のフラッディング範囲はリンク・ローカルです。

- エリア内プレフィックス LSA

この LSA は、IPV6 アドレス・プレフィックスと通過ネットワークとを、ネットワーク LSA を参照して関連付けます。あるいは、IPV6 アドレス・プレフィックスのリストを、ルーター LSA を参照することによりルーターと関連付けます。この LSA のフラッディング範囲はエリア内です。

次のコマンドを、生成される IPv6 LSA を検査する例として使用してください。

- 特定のエリアについて、リンク状態データベースを表示する。

```
DSPOSPF IPVERSION(*IPV6) OPTION(*STATE) STATE(*LSA) LSA('66.66.66.66')
```

- すべての外部 LSA を表示する。

```
DSPOSPF IPVERSION(*IPV6) OPTION(*STATE) STATE(*LSA) LSA(*EXTERNAL)
```

次のコマンドを例として使用して、i5/OS で受信または発信された IPv4 LSA 情報を表示します

- 特定のエリアについて、リンク状態データベースを表示する。

```
DSPOSPF IPVERSION(*IPV4) OPTION(*STATE) STATE(*LSA) LSA('1.1.1.1')
```

- すべての外部 LSA を表示する。

```
DSPOSPF IPVERSION(*IPV4) OPTION(*STATE) STATE(*LSA) LSA(*EXTERNAL)
```

#### 関連概念

10 ページの『IPv6 用の OSPF』

IPv4 と IPv6 では、i5/OS OSPF セマンティクス、アドレッシング、および認証サポートに違いがあります。

## リンク状態レコードの経時処理

リンク状態通知 (LSA) は、OSPF ネットワーク内のすべてのルーターによって発信されます。

OMPROUTED サーバーでは i5/OS 環境における LSA の経時処理または管理についていくつかの考慮事項があります。

- 新しい LSA の経過時間は 0 になります。
- レコードの経過時間は、関連付けられているインターフェースに構成されている伝送遅延によって増加します。
- レコードの経過時間は、レコードが保守される毎秒ごとに増加します。
- LSA の最大経過時間は 1 時間で、その時点で OMPROUTED サーバーはもはやその LSA を経路テーブル計算には使用しません。レコードは最終的にデータベースから除去されます。
- OMPROUTED サーバーは、少なくとも 30 分に 1 回はレコードを再送する必要があります。

## OSPF のパケット・タイプ

OSPF は、隣接性を確立および維持し、要求を送受信し、隣接者間でリンク状態通知 (LSA) を高い信頼度で確実に配信し、リンク状態データベースを記述するために、隣接者にパケットを送信します。リンク状態データベースは、エリア・ルーターが送受信するすべての LSA を元に生成されます。その後で、リンク状態データベースは、Shortest Path First (SPF) アルゴリズムを使用して最短パス・スパンニング・ツリーを計算するために使用されます。

次のパケット・タイプは、OSPF 環境の i5/OS でサポートされます。

## Hello パケット

このパケットは、OSPF 隣接ルーターを発見し、それらとの間に双方向通信を確立するために OMPROUTED サーバーによって送信されます。

OSPF を実行しているいくつかのシステムまたはルーターが、共通のネットワークに接続されているインターフェイスを持っている場合、Hello プロトコルは代表ルーター (DR) を判別します。DR はネットワーク内のすべてのルーターに対して隣接しており、その役割はネットワークを代表して LSA を生成しフラッディング (全ポートにパケットを送出) することです。イーサネットなどのブロードキャスト・ネットワークにおいては、DR を設けると、生成されるルーター・プロトコル・トラフィックの量を削減することになります。

また DR は、ネットワーク内の同一 OSPF エリアの中にある他のすべてのルーターで複製される、ネットワーク・トポロジー・データベースの維持も行います。

DR の概念は、Point-to-Point 接続の場合には存在しません。

## データベース記述パケット

Hello パケットが交換されて両方向通信が確立されたら、i5/OS と他のエリア・ルーターは隣接者になります。この時点で、OMPROUTED サーバーはどの隣接者と隣接性を確立すべきかを認識し、OSPF 隣接性の形成を開始します。隣接性の立ち上げは、ルーター間のデータベース同期化を行う、重要な OSPF プロトコル機能です。データベース記述パケットは、交換データベース・プロトコルを使用して送信されます。交換データベース・プロトコルでは、データベース記述パケットを使用して、隣接パートナー間でリンク状態データベースの記述を交換します。

## リンク状態更新パケット

OMPROUTED データベースが完全に同期化されるまで、リンク状態要求パケットおよびリンク状態更新パケットを使用してさらに隣接ルーターからの情報が要求され交換されます。

より高い数値のルーター ID を持つルーターまたは i5/OS が 1 次的役割を持つと想定され、他方が 2 次的役割を持つとされます。1 次ルーターは自身のデータベース記述を 1 つずつ送信します。2 次ルーターは 1 つずつ応答し、その応答に自身のデータベース記述を組み込みます。タイプ、通知ルーター、およびリンク状態 ID に従ってレコードが比較されます。レコード内の順序番号によって、レコードの新旧が判別されます。新しい記述により、受信側が既にデータベース内に持っているよりも新しいレコードであることが示されれば、この記述が保管されます。

## リンク状態要求パケット

すべての記述が受信された後、隣接者は要求されたレコードに関する、より完全な情報を求めるデータベース要求を送信します。これらの要求に続いて、要求された情報を含むリンク状態更新のフラッディングが行われます。各リンク状態更新パケットは、リンク状態肯定応答パケットにより明示的に、またはリンク状態パケットにより暗示的に、応答されます。ルーターは、リンク状態データベースが完全に同期化された時点で、完全に隣接性を持ちます。

OSPF 隣接性状態には、Down (ダウン)、Init (初期)、Attempt (試行)、2-way (双方向)、Exstart (交換開始)、Exchange (交換)、Loading (ロード)、および Full (完全) があります。i5/OS では 2-way (双方向) は \*WAY2 で表され、そのルーターは、DR またはバックアップ代表ルーター (BDR) とのみ完全に隣接していることを意味します。隣接性状態が \*FULL であるときは、多数の隣接性状態が \*WAY2 であるために、個別のルーターは隣接者については認識していても完全な隣接性は持っていないことを示します。

データベース同期化処理が長過ぎる時は、データベース交換タイムアウト値を、非活動状態ルーター・インターバルで指定されるデフォルト値よりも高く設定することを検討してください。データベース交換タイムアウト値は、ADDOSPFIFC コマンドまたは CHGOSPFIFC コマンドを使用してインターフェースを追加または変更する時に指定します。データベース交換タイムアウトがデフォルトの時間よりも高く設定されると、次のようになります。

- 時間が足りないためにデータベース同期化が失敗する。
- 隣接者処理は隣接状態 \*WAY2 よりも堅固には決してならない。

## リンク状態肯定応答パケット

新しく受信される各 LSA は、配信を検証するために必ず受信側による応答が必要です。これは、1 つ以上の肯定応答を含むリンク状態肯定応答パケットを送信することによって行われます。肯定応答パケットは、指定されている時間間隔に基づいて、即時に、または遅延して送られます。

隣接性の状態を表示するには、次のように OSPF 表示 (DSPOSPF) コマンドで IPv4 か IPv6 かを指定し、STATE パラメーターを指定することにより OSPF 隣接者の状態を表示します。

```
DSPOSPF IPVERSION(*IPV4) OPTION(*STATE) STATE(*NGH)
```

### 関連概念

2 ページの『OSPF 経路指定ドメインおよびエリア』

OSPF 経路指定は、経路指定ドメイン内のエリア間に定義されている関係によって決まります。エリア・タイプおよび i5/OS 経路指定隣接者を定義するには、CL コマンドを使用します。

## IPv6 用の OSPF

IPv4 と IPv6 では、i5/OS OSPF セマンティクス、アドレッシング、および認証サポートに違いがあります。

### IPv6 用の OSPF のアドレッシング

IPv6 アドレッシング・セマンティクスは、ネットワークをプロトコルから独立させるために、OSPF パケットおよび主たる LSA タイプから除去されました。主たる変更は次の通りです。

- アドレッシング情報は LSA のみに含まれます。Hello パケットにはアドレス情報は含まれません。インターフェース ID と呼ばれる新しいフィールドが発信元ルーターによって割り当てられ、インターフェースをリンクに対して一意的に識別します。
- ルーター LSA タイプおよびネットワーク LSA タイプは、もはやネットワーク・アドレスを含まなくなりました。LSA アドレスは、IP アドレスとネットワーク・マスクではなく、プレフィックスおよびプレフィックス長として表されます。
- IPv6 用の OSPF では、隣接ルーターは必ずルーター ID で識別されます。IPv4 では、IP アドレスで識別されます。

Point-to-Point リンクは IPv6 ではサポートされません。

IPv6 はルーター LSA、ネットワーク LSA、AS 外部 LSA、およびその他の追加 LSA タイプのセットをサポートします。

### リンクがサブネットを置換

フラッドイング、指定ルーター選択、エリア・サポート、および Shortest Path First 計算などの OSPF の基本メカニズムは、IPv6 プロトコル使用時に変更されないままですが、セマンティクスにおいては一部変更が行われています。IPv6 の用語リンクは、IPv4 の用語サブネットを置換します。



IPv6 は、ノードがリンク層で通信を行う場合に使用する通信メディアを示すために、用語「リンク」を使用します。IPv6 では、同一の IPv6 プレフィックス・サブネットを共有していない場合にも、複数のサブネットは同一の単一リンク内にあり、システムはその同一リンクを介して他のノードと直接に通信します。

## リンク・ローカル・アドレス

IPv6 リンク・ローカル・アドレスは、本質的に単一物理インターフェースに対して割り当てられるため、自動アドレス構成において有用です。OSPF は、リンク・ローカル・アドレスを使用して、同一の物理インターフェース・リンクによって接続されるすべての隣接者を発見します。OMPROUTED サーバーは、リンク・ローカル・アドレスに接続されるすべての他のルーターまたはシステムのアドレスを発見し、これらのアドレスをネクスト・ホップ情報として使用します。標準のクラスレス・ドメイン間ルーティング・サブネット・マスク表記を使用して、サーバーは FE80/10 の範囲内のリンク・ローカル・アドレスを必要とします。

OSPF IPv6 インターフェース・パケットは、リンク・ローカル・ユニキャスト・アドレスを使用して送信されます。OSPF IPv6 インターフェースを構成するには、OSPF インターフェース追加 (ADDOSPFIFC) コマンドを使用し、リンク・ローカル・アドレスを持つ回線記述に対してそのインターフェースを関連付けます。

IPv6 とは異なり、OSPF 仮想リンク・パケットではリンク・ローカル・アドレスは使用せず、サイト・ローカル IP アドレスまたはグローバル範囲の IP アドレスを使用して送信されます。

## 認証

i5/OS における IPv6 用の OSPF インプリメンテーションでは認証はサポートされません。これは、IPv6 パケット・サポートに既に認証が組み込まれているためです。IPv6 用の OSPF における認証用のフィールドは、OSPF パケット・ヘッダーから除去されています。OSPF IPv6 エリアまたはインターフェースの追加時あるいは変更時には、すべての認証フィールドが無視されます。

### 関連概念

12 ページの『i5/OS OSPF 認証』

OSPF 環境内で i5/OS を認証するために TCP/IP サーバーを構成します。

7 ページの『リンク状態通知』

i5/OS ルーターは、OSPF ネットワークの参加者として、1 つ以上のリンク状態通知 (LSA) を発信し、自分自身に関する経路指定情報を送信したり、または隣接者からの経路指定情報を受信したりします。この情報は、リンク状態データベースの作成に使用します。

### 関連情報

IPv6 アドレス・タイプ

## OSPF インターフェース

i5/OS の場合、OSPF インターフェースは、既存の i5/OS 回線記述に関連付けられている論理 TCP/IP 接続を表す IP アドレスです。回線記述は OSPF インターフェースとして構成されます。

使用可能な OSPF 経路指定インターフェースとしてインターフェースを識別するには、ADDOSPFIFC コマンドを使用して、OMPROUTED サーバーが必要とするさまざまなパラメーターを設定します。i5/OS の場合、インターフェース ID は IPv4 アドレスまたは IPv6 アドレスです。

OSPF は、インターフェースがイーサネットなどのブロードキャスト・ネットワークに接続される場合は、マルチキャスト機能を使用します。すべての OSPF ルーターにおいて、これらのインターフェースはマル

チキャスト・アドレス 224.0.0.5 に結合されます。マルチキャスト IP アドレスに結合されているインターフェースは、224.0.0.5 IP アドレスに宛てられたパケットのみを受信します。IPv6 では、マルチキャスト・アドレスに ff02::5 を使用します。

注: ブロードキャスト以外のメディアまたはマルチキャスト対応のメディアでは、隣接者を定義してはなりません。OMPROUTED サーバーは、定義済みの隣接者に対してのみ OSPF 情報を通信し、追加の隣接者と隣接性を形成することはできません。

有効な OSPF インターフェースである i5/OS TCP/IP インターフェースを識別するには、IPv4 の場合 NETSTAT OPTION(\*IFC) を、IPv6 の場合 NETSTAT OPTION(\*IFC6) を使用します。共通のサブネットに属する複数の OSPF インターフェース ID が 1 つの OSPF 構成に追加される場合、実際には 1 つのインターフェースのみが OSPF 情報を交換します。

#### 関連情報

- OSPF インターフェース追加 (ADDOSPFIFC)
- OSPF インターフェース除去 (RMVOSPFIFC)
- OSPF インターフェース変更 (CHGOSPFIFC)

## OSPF の Point-to-Point リンク

Point-to-Point 接続は、2 つのルーターを接続するシリアル・リンクで、接続するルーターを OSPF (Open Shortest Path First) 隣接者にします。Point-to-Point 接続は、OSPF 環境内の i5/OS でサポートされます。

i5/OS に Point-to-Point リンクを介する隣接者を発見させるには、モデム・リソースへの接続を表す接続プロファイルまたは回線記述が必要です。モデムは専用モードまたはスイッチ・モードです。

OSPF インターフェース追加 (ADDOSPFIFC) コマンドを次のように使用し、IFC パラメーターおよび PPPCNNPRF パラメーターを指定することにより、OSPF 構成に Point-to-Point インターフェースを追加します。ここで、*connection\_profile\_name* は接続プロファイル・リスト内の既存のプロファイルを表します。

```
ADDOSPFIFC IFC(*PPPCNNPRF) PPPCNNPRF(connection_profile_name)
```

Point-to-Point 接続を構成に追加する時点で接続プロファイルが活動状態になっている必要はありません。Point-to-Point リンクを介して OSPF パケットを送信するために最終的に使用されるローカル・インターフェースが認識されない場合、インターフェース状況が \*DOWN を示すことがあります。プロファイルが活動状態になった時点で、プロファイルは OMPROUTED サーバーに、このインターフェースを使用してパケットを送信することを伝えます。

#### 関連情報

リモート・アクセス・サービス: PPP 接続

## i5/OS OSPF 認証

OSPF 環境内で i5/OS を認証するために TCP/IP サーバーを構成します。

システムの OSPF 認証は、OSPF ネットワーク環境内のパケットの引き渡しを制御します。OSPF 認証は、次のいずれかのコマンドで AUTHTYPE パラメーターを使用して、OSPF インターフェースまたは OSPF 仮想リンクを追加あるいは変更する時に指定されるタイプによって、制御されます。

- OSPF インターフェース追加 (ADDOSPFIFC)
- OSPF インターフェース変更 (CHGOSPFIFC)
- OSPF 仮想リンク追加 (ADDOSPFLNK)
- OSPF 仮想リンク変更 (CHGOSPFLNK)

前述のコマンドのいずれかで AUTHTYPE が指定されない場合は、コマンドが指定する OSPF エリアに対して定義されているタイプが継承されます。

AUTHTYPE パラメーターは OSPF IPv6 インターフェースでは無視されます。

OSPF ネットワークに参加しているシステムに対して OMPROUTED TCP/IP サーバーを認証するよう構成することは、次のような利点があります。

- 変更がトラステッド・システムまたはルーターのみに限定される。
- 複数の悪意のある外部ルーターを OSPF ネットワーク内に入れてしまう可能性を持つルーターの偶発的な導入に対して、保護が提供される。

OMPROUTED サーバーはメッセージ要約アルゴリズム 5 (MD5) 認証と、シンプル・パスワード認証の両方をサポートする。

## メッセージ要約アルゴリズム 5 (MD5) 認証

i5/OS プラットフォームでは、MD5 値は 32 文字の 16 進数または ASCII 値として表されます。

MD5 認証用の 16 進値を提供するには、次の要件に従います。

- 長さは正確に 32 文字でなければならず、先頭文字は **X** でなければなりません。
- 認証キー全体を単一引用符で囲まなければならない。

これは、MD5 認証用の 16 進値を提供する例です。

```
ADDOSPFIFC IFC('9.7.85.1') AREA('1.1.1.1') AUTHTYPE(*MD5) AUTHVAL (124
'Xffffffffffffffffffffffffffffffff')
```

MD5 認証用の ASCII 値を提供するには、次の要件に従います。

- キーは Cisco 互換性のあるものでなければなりません。
- 最大長は 16 文字でなければならず、先頭文字は **A** または **a** でなければなりません。
- 認証キー全体を単一引用符で囲まなければならない。

これは、MD5 認証用の ASCII 値を提供する例です。

```
ADDOSPFIFC IFC('9.7.85.1') AREA('1.1.1.1') AUTHTYPE(*MD5) AUTHVAL(124 'aACDZ')
```

## シンプル・パスワード認証

i5/OS プラットフォームでは、シンプル・パスワードは 16 文字の 16 進数または ASCII 値として表されます。

シンプル・パスワード認証用の 16 進値を提供するには、次の要件に従います。

- 長さは正確に 16 文字でなければならず、先頭文字は **X** でなければなりません。
- 認証キー全体を単一引用符で囲まなければならない。

これは、シンプル・パスワード認証用の 16 進値を提供する例です。

```
ADDOSPFIFC IFC('9.7.85.1') AREA('1.1.1.1') AUTHTYPE(*PASSWORD) AUTHVAL(124
'Xffffffffffffffff')
```

シンプル・パスワード認証用の ASCII 値を提供するには、次の要件に従います。

- キーは Cisco 互換性のあるものでなければなりません。

- 最大長は 8 文字。
- 認証キー全体を単一引用符で囲まなければならない。

これは、シンプル・パスワード認証用の ASCII 値を提供する例です。

```
ADDOSPFLNK NGRTR('2.2.2.2') LNKTMASARA('1.1.1.1') AUTHTYPE(*PASSWORD) AUTHVAL(*N
'del8gado')
```

#### 関連概念

10 ページの『IPv6 用の OSPF』

IPv4 と IPv6 では、i5/OS OSPF セマンティクス、アドレッシング、および認証サポートに違いがあります。

## i5/OS OSPF ジョブ・トレースの使用可能化

OSPF 経路指定の状態をトレースするために、関連する i5/OS ジョブ・ログにメッセージが記録されます。

OSPF 経路指定時には、メッセージが QTOOROUTE ジョブ・ログに送られてユーザーに次のような情報を伝えます。

- Hello プロトコルの処理中は、構成済み OSPF インターフェースの状態変更に関する情報。
- データベース交換プロセス中は、発見された隣接者の状態変更に関する情報。
- OSPF によって発見された経路の状態。

QTOOROUTE ジョブをトレースして、OSPF プロトコルの流れを検証します。トレースは、OSPF インターフェースを介して送受信されたパケットを表示します。

OSPF のトレースを使用可能にするには、TRCTYPE を \*ROUTING に設定し、さらにオプションで、他のトレース・レベルに設定します。次の例では、トレース開始 (STRTRC) コマンドを使用して OSPF のトレースを開始し、OSPF に関するトレースだけをフィルター操作します。

```
STRTRC SSNID(QTOOROUTE) JOB((*ALL/QTCP/QTOOROUTE)) JOBTRCTYPE(*TRCTYPE)
TRCTYPE((*ROUTING *VERBOSE))
```

活動状態の i5/OS セッションをトレースすることにより、OSPF CL コマンドのトレースがサポートされます。例として次を使用します。

```
STRTRC SSNID(OSPF) JOBTRCTYPE(*TRCTYPE) TRCTYPE((*ROUTING *VERBOSE))
```

---

## OSPF (Open Shortest Path First) サポートの作業

このトピックでは、ご使用の i5/OS を OSPF ネットワークに参加させるために必要となる構成のステップを説明します。TCP/IP インターフェースを始動、終了、および再始動するために使用できるさまざまなコマンドも説明します。

### OSPF ネットワーキング用の i5/OS の構成

OSPF 構成には、ルーター ID、エリア、およびエリア内に接続する OSPF 登録インターフェースが含まれます。i5/OS は、自身を OSPF ネットワークの参加者として設定するための CL コマンドのセットを提供します。

有効な OSPF インターフェースである i5/OS TCP/IP インターフェースを識別するには、IPv4 の場合 NETSTAT OPTION(\*IFC) を、IPv6 の場合 NETSTAT OPTION(\*IFC6) を使用します。TCP/IP の開始と停止については、『i5/OS 上で OSPF 用に TCP/IP を使用可能化する』のトピックを参照してください。

ご使用の i5/OS を構成して OSPF ネットワークに参加するには、以下のステップに従います。

1. OSPF 属性変更 (CHGOSPFA) ルーター ID を設定します。

CHGOSPFA コマンドはルーター ID を設定します。ルーター ID は、OSPF ネットワーク内で固有なシステム ID です。小数点付き 10 進数フォーマットの IP アドレスです。CHGOSPFA の例は次のようになります。

```
CHGOSPFA ROUTER('1.1.1.1')
```

2. OSPF エリア追加 (ADDOSPFARA) エリア ID を構成します。

ADDOSPFARA コマンドは、OSPF が稼働するイントラネットまたは自律システム (AS) 内で OSPF エリアを識別します。OSPF エリアは、OSPF インターフェースが接続される IP ネットワークの集合です。エリアは、インターフェースが OSPF 構成に追加される時点で存在していなければなりません。エリア ID は小数点付き 10 進数フォーマットで表されます。ADDOSPFARA の例は次の通りです。

```
IPv4 の場合、ADDOSPFARA AREA('1.1.1.1')
```

```
IPv6 の場合、ADDOSPFAREA ('1.1.1.1') IPVERSION(*IPV6)
```

3. OSPF インターフェース追加 (ADDOSPFIFC) インターフェース ID を構成します。

OSPF インターフェースは、構成済みの論理 TCP/IP インターフェースを表す IP アドレスで、システム上に存在する回線記述に関連付けられます。インターフェースを経路指定プロトコル・インプリメンテーションに追加するには、ADDOSPFIFC コマンドを使用して、OMPROUTED サーバーが必要とするさまざまなパラメーターを設定します。ADDOSPFIFC の例は次の通りです。

```
IPv4 の場合、ADDOSPFIFC IFC('9.7.85.1') AREA('1.1.1.1')
```

```
IPv6 の場合、ADDOSPFIFC IFC('2000::38') AREA('1.1.1.1')
```

#### 関連概念

18 ページの『シナリオ: OSPF インターフェースおよび隣接者の構成』

このシナリオでは、サンプル・ネットワークで i5/OS システムの OSPF 構成を示します。

#### 関連情報

OSPF 属性変更 (CHGOSPFA)

OSPF エリア追加 (ADDOSPFARA)

OSPF インターフェース追加 (ADDOSPFIFC)

## i5/OS 上で OSPF 用に TCP/IP を使用可能化する

i5/OS TCP/IP インターフェースが活動状態であることは、OSPF を使用するための必要条件です。

1. TCP/IP 構成 (CFGTCP) コマンドのさまざまなオプションを使用して、TCP/IP OSPF インターフェースを設定します。
2. TCP/IP サーバー開始 (STRTCPSVR) コマンドを次のように使用して、サーバーを始動します。

```
STRTCPSVR SERVER(*OMPROUTED) INSTANCE(*OSPF)
```

このコマンドは、QSYSWRK サブシステムに QTOOROUTE ジョブをサブミットします。

QTOOROUTE ジョブが活動状態の時にこのコマンドが出されると、ユーザー・ジョブ・ログにメッセ



ージが送られます。INSTANCE オプションを指定すると、OSPF 経路指定機能だけが開始されます。オプションで、このパラメーターを省略することができます。

TCP/IP インターフェースが活動状態になった後は、必要に応じて、次の情報を使用してサーバーを終了させたり再始動させたりします。

- 次のコマンドを使用して、OSPF 経路指定機能を終了します。

```
ENDTCPSVR SERVER(*OMPROUTED) INSTANCE(*OSPF)
```

このコマンドは、QTOOROUTE ジョブから OSPF 機能のみを終了させます。ジョブは引き続き QSYSWRK サブシステムのもとで活動状態のままです。

- 次のコマンドを使用して、OMPROUTED サーバーを再始動します。

```
STRTCPSVR SERVER(*OMPROUTED) RESTART(*OMPROUTED)
```

このコマンドは、QSYSWRK サブシステムに QTOOROUTE ジョブを再サブミットします。構成および索引ファイルの両方が再処理され、同期化されます。これらのファイルが同期化されていないと、予測不能な結果が生じます。例えば、別のシステムから構成ファイルだけをインポートすると、ファイルが同期化されていないことになり、サーバーの再始動が必要になります。

#### 関連情報

TCP/IP セットアップ

OSPF (Open Shortest Path First)

TCP/IP サーバーの終了 (ENDTCPSVR)

TCP/IP サーバーの始動 (STRTCPSVR)

---

## OSPF (Open Shortest Path First) サポートの参照

ユーザーの参照用に、OSPF 経路指定で使用される i5/OS API および CL コマンドについて、このトピックで説明します。またこのトピックでは、OSPF 経路指定環境の設定時に参照できるシナリオも記載します。

## OSPF (Open Shortest Path First) API およびコマンド

OSPF 属性、インターフェース、エリア、範囲、リンク、統計、状態情報を管理し、OSPF アクティビティをログに記録するには、制御言語 (CL) コマンドを使用します。OSPF 統計を検索するには、OSPF 状態情報検索 API を使用します。

### CL コマンド

経路指定プロトコル構成 (CFGRTG) コマンドは、システム上のすべての OSPF コマンドのリストを表示します。

表 1. CL コマンドおよびサポートされる機能

CL 名	CL コマンド	関数
OSPF 属性変更	CHGOSPFA	一般的な OSPF 属性を設定します。
OSPF 表示	DSPOSPF	現行の OSPF 構成または経路指定プロトコルの状態を表示します。
OSPF インターフェース追加	ADDOSPFI	OSPF のインターフェースを追加します。
OSPF インターフェース除去	RMVOSPFI	OSPF のインターフェースを除去します。
OSPF インターフェース変更	CHGOSPFI	OSPF のインターフェースを変更します。

表 1. CL コマンドおよびサポートされる機能 (続き)

CL 名	CL コマンド	関数
OSPF エリア追加	ADDOSPFARA	OSPF のエリアを追加します。
OSPF エリア除去	RMVOSPFARA	OSPF のエリアを除去します。
OSPF エリア変更	CHGOSPFARA	OSPF のエリアを変更します。
OSPF 範囲追加	ADDOSPFRRNG	OSPF の範囲を追加します。
OSPF 範囲除去	RMVOSPFRRNG	OSPF の範囲を除去します。
OSPF 範囲変更	CHGOSPFRRNG	OSPF の範囲を変更します。
OSPF 仮想リンク追加	ADDOSPFLNK	OSPF の仮想リンクを追加します。
OSPF 仮想リンク除去	RMVOSPFLNK	OSPF の仮想リンクを除去します。
OSPF 仮想リンク変更	CHGOSPFLNK	OSPF の仮想リンクを変更します。

## OSPF 表示コマンド

OSPF 表示 (DSPOSPF) コマンドは、構成ファイルに設定されている OSPF 構成を表示します。OSPF バージョンが IPv4 か IPv6 かは、DSPOSPF コマンドの必須パラメーターです。

OPTION パラメーターは、構成情報と状態情報のいずれを表示するかを指定します。インターフェースの状況および実行時統計を示す状態情報を表示するには、次のようにして OPTION パラメーターを \*STATE に設定します。

```
DSPOSPF IPVERSION(*IPV4) OPTION(*STATE)
```

構成情報を表示するには、OPTION パラメーターを \*CFG に設定し、CONFIG パラメーターを設定することにより、表示される構成情報のタイプを指定します。エリア、インターフェース、隣接者、または仮想リンクに関する情報を検索するには、CONFIG パラメーターにこれらのいずれかを指定します。ルーター ID と OSPF 経路指定プロトコル状況を表示するには、CONFIG パラメーターに \*GLOBAL を指定します。

```
DSPOSPF IPVERSION(*IPV4) OPTION(*CFG) CONFIG(*GLOBAL)
```

注:

\*GLOBAL を指定するには、Simple Network Management Protocol (SNMP) サーバーが始動していることが必要です。次のようなコマンドを入力して、サーバーを開始するかまたはサーバーが活動状態かどうか検証します。

- サーバーを開始するには、SNMP サーバーを指定して TCP/IP サーバー開始 (STRTCPSVR) コマンドを使用します。

```
STRTCPSVR *SNMP
```

- サーバーが活動状態であることを検証するには、次のように QTMSNMP ジョブ名を指定して WRKJOB コマンドを使用し、続いてオプション 1 を選択します。

```
WRKJOB QTMSNMP
```

## OSPF (Open Shortest Path First) 状態情報検索 API

OSPF (Open Shortest Path First) 状態情報検索 (QtooRtvOSPFData) API は、OMPROUTED TCP サーバーから OSPF (Open Shortest Path First) 統計を検索します。

関連概念

24 ページの『シナリオ: OSPF 状態情報検索 API』

このシナリオでは、i5/OS の OSPF 状態情報検索 (QtooRtvOSPFdta) API を使用して OMPROUTED サーバーの情報を検索します。

### 関連情報

OSPF 状態情報検索 (QtooRtvOSPFdta) API  
OSPF 属性変更 (CHGOSPFA)  
OSPF エリア追加 (ADDOSPFARA)  
OSPF インターフェース追加 (ADDOSPFIFC)  
OSPF 仮想リンク追加 (ADDOSPFLNK)  
OSPF 範囲追加 (ADDOSPFRRNG)  
OSPF エリア変更 (CHGOSPFARA)  
OSPF インターフェース変更 (CHGOSPFIFC)  
OSPF 仮想リンク変更 (CHGOSPFLNK)  
OSPF 範囲変更 (CHGOSPFRRNG)  
OSPF 表示 (DSPOSPF)  
TCP/IP サーバーの終了 (ENDTCPSVR)  
OSPF エリア除去 (RMVOSPFARA)  
OSPF インターフェース除去 (RMVOSPFIFC)  
OSPF 仮想リンク除去 (RMVOSPFLNK)  
OSPF 範囲除去 (RMVOSPFRRNG)  
TCP/IP サーバーの始動 (STRTCPSVR)  
トレース開始 (STRTRC)

## シナリオ: OSPF の構成

OSPF 構成の項目をサンプル・ネットワークで示すシナリオ、OSPF マルチパス経路を説明するシナリオ、そして i5/OS OSPF API について説明するシナリオを示します。

注: コーディング例をご使用になることで、28 ページの『コードに関するライセンス情報および特記事項』の条件に同意されたものとみなされます。

## シナリオ: OSPF インターフェースおよび隣接者の構成

このシナリオでは、サンプル・ネットワークで i5/OS システムの OSPF 構成を示します。

サンプル構成では、システム A およびシステム B のインターフェースと隣接状態を示します。Hello プロトコルおよびデータベース交換プロトコルが隣接ルーターを発見し、隣接ルーター間の隣接性を形成します。OSPF プロトコルは、両方のルーターでリンク状態型のデータベース同期を維持します。

注: コーディング例をご使用になることで、28 ページの『コードに関するライセンス情報および特記事項』の条件に同意されたものとみなされます。





図2. OMPROUTED 状態

システム A は次のステップおよび情報を使用して構成されました。

1. ルーター ID は、次の OSPF 属性変更 (CHGOSPFA) コマンドを使用して設定されました。

```
CHGOSPFA ROUTER('1.1.1.1')
```

2. OSPF エリア ID は、次の OSPF エリア追加 (ADDOSPFARA) コマンドを使用して追加されました。

```
ADDOSPFARA AREA('1.1.1.1')
```

3. OSPF インターフェースは、インターフェース 9.7.85.1 が有効な TCP/IP インターフェースであることを確認した後に、次の OSPF インターフェース追加 (ADDOSPFIFC) コマンドを使用して構成されました。

```
ADDOSPFIFC IFC('9.7.85.1') AREA('1.1.1.1')
```

システム B は次のステップおよび情報を使用して構成されました。

1. ルーター ID は、次の CHGOSPFA コマンドを使用して設定されました。

```
CHGOSPFA ROUTER('2.2.2.2')
```

2. OSPF エリア ID は、次の ADDOSPFARA コマンドを使用して追加されました。

```
ADDOSPFARA AREA('2.2.2.2')
```

3. OSPF インターフェースは、インターフェース 9.7.85.2 が有効な TCP/IP インターフェースであることを確認した後に、次の ADDOSPFIFC コマンドを使用して構成されました。

```
ADDOSPFIFC IFC('9.7.85.2') AREA('2.2.2.2')
```

## サーバーの状況

OSPF グローバル情報表示コマンドを使用して、OSPF プロトコル状況が表示されました。OSPF プロトコル状況とは、OSPF プロトコルを実行している OMPROUTED サーバーの状況です。TCP/IP サーバー開始 (STRTCPSVR) コマンドが使用され、この例の残りの部分での状況は使用不可から使用可能に変更されました。

```

                                OSPF グローバル情報の表示
                                システム :   BS28A
OSPF プロトコル状況 . . . . . :   使用不可
ルーター ID . . . . . :   1.1.1.21
境界機能 . . . . . :   使用可能
エリア境界ルーターの機能 . . . . . :   使用不可
サーバーの自動開始 . . . . . :   *YES
サブエージェント . . . . . :   *YES

```

図3. サーバー状況を調べるために表示される OSPF グローバル情報。

### インターフェースの状況

OMPROUTED サーバー・ジョブが双方のシステムで開始された後で、OSPF 表示 (DSPOSPF) コマンドが使用され、各 OSPF インターフェースの状態が検査されました。システム A とシステム B の両方について、コマンドで次の通り IP バージョンを IPv4 と指定することにより、状況が検索されました。

```

DSPOSPF IPVERSION(*IPV4) OPTION(*STATE) STATE(*IFC)

                                OSPF 構成済みインターフェースの状態の表示
                                システム :   DS30A
ルーター ID . . . . . :   1.1.1.1
構成済みインターフェースの数 . . . . . :   1

                                構成済みインターフェース・リスト
オプションを入力して、実行キーを押してください。
5= 詳細の表示

OPT インターフェースID   エリア ID   インターフェース   インターフェース   隣接局
                              .タイプ   状況   の数
          9.5.85         1.1.1.1         *BCST   T   *BDR         1

                                終わり

F3= 終了   F5= 最新表示   F6= 印刷   F12= 取り消し

```

図4. 検査された、システム A の OSPF インターフェースの状況。



リンク状態公示の表示					
リンク状態公示の数 . . . . . : 3					システム : CS29A
合計チェックサム . . . . . : 1e561					
リンク状態公示リスト					
リンク 状態 タイプ	リンク 状態 宛先	リンク 状態 発信元	リンク 状態開始 順序番号	リンク 状態 時間	リンク 状態 チェックサム
*LSTRL	1.1.1.1	1.1.1.1	80000024	1270	91ef
*LSTRL	2.2.2.2	2.2.2.2	80000086	1275	91ef
*LSTRL	2.2.2.2	2.2.2.2	8000001e	820	8625

図7. 表示されたシステム B の LSA。

**関連タスク**

14 ページの『OSPF ネットワーキング用の i5/OS の構成』

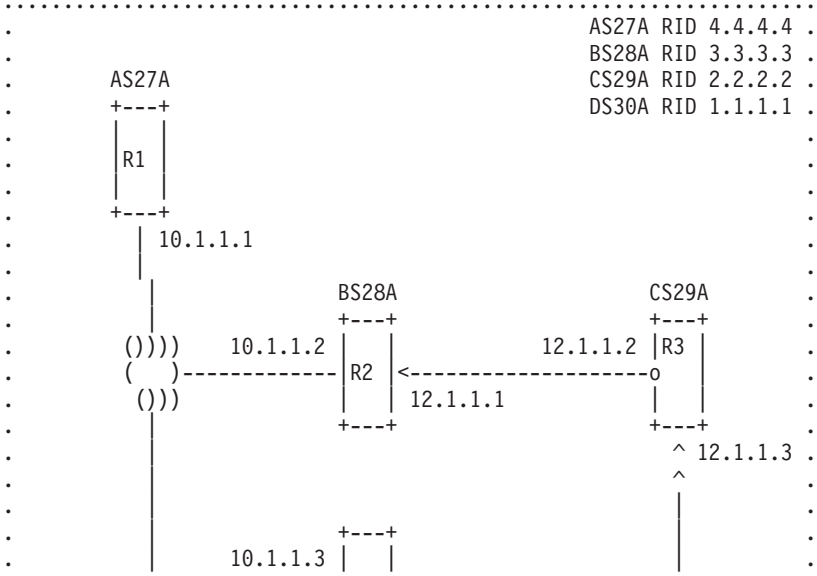
OSPF 構成には、ルーター ID、エリア、およびエリア内に接続する OSPF 登録インターフェースが含まれます。i5/OS は、自身を OSPF ネットワークの参加者として設定するための CL コマンドのセットを提供します。

**シナリオ: OSPF マルチパス経路**

このシナリオでは、同一の i5/OS に対する異なる OSPF 経路の使用について説明します。OSPF の利点は、同一の宛先またはシステムへのマルチパス経路、等価コスト経路を計算できることにあります。

**サンプル構成**

この例では 4 つの System i 物理装置があります。システム R1、R2、および R4 には、同一のイーサネット・ネットワーク (10.1.1.1) に接続されている OSPF インターフェースがあります。システム R2 および R4 には、システム R3 への OSPF インターフェース・タイプ Point-to-Point 接続があります。各インターフェースに 1 つのコストがあり、すべてのシステムが同一の OSPF エリア (10.10.10) 内にあります。



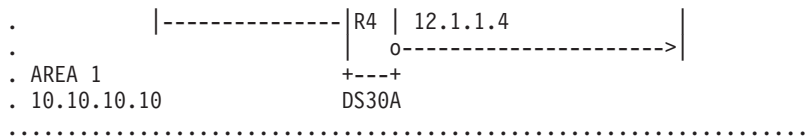


図8. マルチパス構成

### 検索された TCP/IP 経路情報

経路情報を表示するために、i5/OS コマンド行で次のように NETSTAT コマンドを入力した後、オプション 2「TCP/IP 経路情報表示」を選択しました。

```
NETSTAT *RTE
```

次のように情報がディスプレイに示されます。

- 12.1.1.4 への 2 つの重複経路。1 つはネクスト・ホップ 10.1.1.2 によるもので、他方は 10.1.1.3 によるものです。
- 12.1.1.1 への 2 つの重複経路。2 つの異なるネクスト・ホップを経由するパスがあり、1 つは 10.1.1.2 によるもので他方は 10.1.1.3 によるものです。

TCP/IP 経路情報の表示

システム : AS27A

オプションを入力して、実行キーを押してください。  
5= 明細の表示

OPT	経路宛先	サブネット・マスク	ネクスト・ホップ	使用可能な経路
	9.5.130.0	255.255.255.0	*DIRECT	*YES
	10.1.1.0	255.255.255.0	*DIRECT	*YES
	12.1.1.4	*HOST	10.1.1.2	*YES
	12.1.1.4	*HOST	10.1.1.3	*YES
	12.1.1.3	*HOST	10.1.1.3	*YES
	12.1.1.2	*HOST	10.1.1.2	*YES
	12.1.1.1	*HOST	10.1.1.2	*YES
	12.1.1.1	*HOST	10.1.1.3	*YES
	127.0.0.0	255.0.0.0	*DIRECT	*YES
	224.0.0.0	240.0.0.0	*DIRECT	*YES
	224.0.0.0	240.0.0.0	*DIRECT	*YES
	224.0.0.0	240.0.0.0	*DIRECT	*YES
	*DFTRROUTE	*NONE	9.5.130.1	*YES

終わり

F3= 終了      F5= 最新表示      F9= コマンド入力行      F11= 経路タイプの表示  
F12= 取り消し      F13= 欄による分類      F20=IPV6 経路の表示      F24= キーの続き

図9. NETSTAT コマンドで検索される経路情報

## TCP/IP 経路情報の表示

システム : AS27A

オプションを入力して、実行キーを押してください。

5= 明細の表示

OPT	経路宛先	TYPE OF SERVICE	経路 MTU	Route タイプ	経路ソース
	9.5.130.0	*NORMAL	4100	*DIRECT	*CFG
	10.1.1.0	*NORMAL	1492	*DIRECT	*CFG
	12.1.1.4	*NORMAL	1492	*HOST	*OSPF
	12.1.1.4	*NORMAL	1492	*HOST	*OSPF
	12.1.1.3	*NORMAL	1492	*HOST	*OSPF
	12.1.1.2	*NORMAL	1492	*HOST	*OSPF
	12.1.1.1	*NORMAL	1492	*HOST	*OSPF
	12.1.1.1	*NORMAL	1492	*HOST	*OSPF
	127.0.0.0	*NORMAL	576	*DIRECT	*CFG
	224.0.0.0	*NORMAL	1492	*DIRECT	*CFG
	224.0.0.0	*NORMAL	4100	*DIRECT	*CFG
	224.0.0.0	*NORMAL	576	*DIRECT	*CFG
	*DFTRROUTE	*NORMAL	4100	*DFTRROUTE	*CFG

終わり

F3= 終了      F5= 最新表示      F9= コマンド入力行      F11= 経路タイプの表示  
 F12= 取り消し      F13= 欄による分類      F20=IPV6 経路の表示      F24= キーの続き

図 10. NETSTAT コマンドで検索される経路情報

## シナリオ: OSPF 状態情報検索 API

このシナリオでは、i5/OS の OSPF 状態情報検索 (QtooRtvOSPFdta) API を使用して OMPROUTED サーバーの情報を検索します。

注: コーディング例をご使用になることで、28 ページの『コードに関するライセンス情報および特記事項』の条件に同意されたものとみなされます。

```

/** START HEADER FILE SPECIFICATIONS *****/
/*
/* Source File Name: H/OSPFAPITEST
/*
/* Descriptive Name: Retrieve OSPF State Information
/*
/* Description: The Retrieve OSPF State Information(QtooRtvOSPFdta) */
/* API retrieves information about OSPF that runs */
/* under the OMPROUTED server.
/*
/* Header Files Included: None.
/*
/* Macros List: None.
/*
/*Structure List:
/*Qtoo_SPFI0100_t (OSPF General info)
/*Qtoo_IPv4_OSPF_Area_Entry_t (OSPF IPv4 Area List Entry)
/*Qtoo_IPv6_OSPF_Area_Entry_t (OSPF IPv6 Area List Entry)
/*Qtoo_IPv4_OSPF_Ifc_Entry_t (OSPF IPv4 Interface Entry)
/*Qtoo_IPv6_OSPF_Ifc_Entry_t (OSPF IPv6 Interface Entry)
/*Qtoo_IPv4_OSPF_Neighbor_Entry_t (OSPF IPv4 Neighbor Entry)
/*Qtoo_IPv6_OSPF_Neighbor_Entry_t (OSPF IPv6 Neighbor Entry)
/*Qtoo_IPv4_OSPF_Vt1_Link_Entry_t (OSPF IPv4 Virtual Link Entry)
/*Qtoo_IPv6_OSPF_Vt1_Link_Entry_t (OSPF IPv6 Virtual Link Entry)
/*Qtoo_IPv4_OSPF_LSA_Entry_t (OSPF IPv4 LSA Entry)
/*Qtoo_IPv6_OSPF_LSA_Entry_t (OSPF IPv6 LSA Entry)

#include <QSYSINC/H/QTOOSPF1>
#include <stdio.h>
#include <signal.h>
#include <string.h>

```

```

#include <stdlib.h>
#include <qusrctug.h>
#include <qusdltug.h>
#include <qusec.h>

/*****
/*          Structures
*****/

typedef struct {
    Qus_EC_t ec_fields;
    char    exception_data[100];
} error_code_struct;

/*****
/*          Print Main (General) Info
*****/
void prtMainInfo(Qtoo_SPFI0100_t *OSPF_Ptr)
{
    printf("Router Id: %s \n", OSPF_Ptr->RouterID);
    printf("Attached Areas: %d \n", OSPF_Ptr->Attached_Areas);
    printf("Dijkstra Runs: %d \n", OSPF_Ptr->Dijkstra_Runs);
    printf("Packages Received: %d \n", OSPF_Ptr->OSPFpkt_Received);
    printf("LSA's Allocated: %d \n", OSPF_Ptr->LSA_Allocated);
    printf("Number Ipv4 Areas: %d \n", OSPF_Ptr->Number_Of_IPv4_Area_Lst_Ent);
    printf("Number Ipv6 Areas: %d \n", OSPF_Ptr->Number_Of_IPv6_Area_Lst_Ent);
    printf("Number Ipv4 Interfaces: %d \n", OSPF_Ptr->Number_Of_IPv4_Ifc_Lst_Ent);
    printf("Number Ipv6 Interfaces: %d \n", OSPF_Ptr->Number_Of_IPv6_Ifc_Lst_Ent);
    printf("Number Ipv4 Neighbors: %d \n", OSPF_Ptr->Number_Of_IPv4_Neigh_Lst_Ent);
    printf("Number Ipv6 Neighbors: %d \n", OSPF_Ptr->Number_Of_IPv6_Neigh_Lst_Ent);
    printf("Number Ipv4 Virtual Links: %d \n", OSPF_Ptr->Number_Of_IPv4_Vtl_Lst_Ent);
    printf("Number Ipv6 Virtual Links: %d \n", OSPF_Ptr->Number_Of_IPv6_Vtl_Lst_Ent);
    printf("Number Ipv4 LSA's: %d \n", OSPF_Ptr->Number_Of_IPv4_Vtl_Lst_Ent);
    printf("Number Ipv6 LSA's: %d \n", OSPF_Ptr->Number_Of_IPv6_Vtl_Lst_Ent);
}

/*****
/*          Main
*****/

int main()
{
    /*Reserving the enough bytes for the header in the ospf structure*/
    char* OSPF_Receiver = (char*)malloc (1000);
    int Length_Receiver = 1000;
    int i;
    error_code_struct error_code;

    /*****
    /* Initialize the error code parameter.
    *****/

    error_code.ec_fields.Bytes_Provided=sizeof(error_code_struct);

    /*****
    /* Prototype for calling Retrieve OSPF State Information API
    /* (QtooRtvOSPFdta)
    *****/

    QtooRtvOSPFdta( OSPF_Receiver, /* Receiver variable */
                   &Length_Receiver, /* Length of receiver variable */
                   "SPFI0100", /* Format name */
                   (char*)&error_code); /* Error code */

    /*****
    /* If an exception occurred, the API would have returned the
    /* exception in the error code parameter. The bytes available
    /* field will be set to zero if no exception occurred and greater
    /* than zero if an exception did occur.
    *****/

    if (error_code.ec_fields.Bytes_Available > 0)
    {
        printf("FAILED WITH EXCEPTION:%s",
              error_code.ec_fields.Exception_Id);
        exit(1);
    }

    Qtoo_SPFI0100_t *OSPF_Ptr = (Qtoo_SPFI0100_t *) OSPF_Receiver;

    if( OSPF_Ptr->Bytes_Returned < OSPF_Ptr->Bytes_Available)
    {
        printf("Recalculating space for receiver");
        /*Recalculating the enough bytes for the header in the ospf structure*/
        OSPF_Receiver = (char*)realloc (OSPF_Receiver,OSPF_Ptr->Bytes_Available);
    }

    /*****

```

```

/* Prototype for calling Retrieve OSPF State Information API */
/* (QtooRtvOSPFData) */
/*****

QtooRtvOSPFData( OSPF_Receiver, /* Receiver variable */
                 Length_Receiver, /* Length of receiver variable */
                 "SPFI0100", /* Format name */
                 (char *)&error_code); /* Error code */

/*Qtoo_SPFI0100_t *OSPF_Ptr = (Qtoo_SPFI0100_t *) OSPF_Receiver;*/
OSPF_Ptr = (Qtoo_SPFI0100_t *) OSPF_Receiver;
}

/* print general info */
printf("Router Id: %s \n", OSPF_Ptr->RouterID);
prtMainInfo(OSPF_Ptr);
printf("Router Id: %s \n", OSPF_Ptr->RouterID);

char* IPv4_Ptr = (char*) OSPF_Receiver;

/*Cast of the Qtoo_IPv4_OSPF_Area_Entry_t structure is made to the IPv4_Ospf_Area_Ptr */
/*pointer and the displacement will be Offset_To_IPv4_Area_Lst bytes, to aim at the */
/*Offset_To_IPv4_Area_Lst of the list in the structure */

Qtoo_IPv4_OSPF_Area_Entry_t *IPv4_Ospf_Area_Ptr=(Qtoo_IPv4_OSPF_Area_Entry_t*)(IPv4_Ptr+OSPF_Ptr->Offset_To_IPv4_Area_Lst);
for(unsigned long i=0;i < OSPF_Ptr->Number_Of_IPv4_Area_Lst_Ent; i++)
{
    printf("\nPrinting IPv4 Area Number %d",i);
    printf("\nArea ID: %s",IPv4_Ospf_Area_Ptr->AreaID);
    printf("\nOSPF_Runs: %d",IPv4_Ospf_Area_Ptr->SPF_Runs);
    printf("\nArea_BR_Count: %d",IPv4_Ospf_Area_Ptr->Area_BR_Count);
    printf("\nTotal_Number_of_LSA_Area: %d",IPv4_Ospf_Area_Ptr->Total_Number_of_LSA_Area);
    /*Increment IPv4_Ospf_Area_Ptr bytes in order to point to the next structure*/
    IPv4_Ospf_Area_Ptr+=sizeof(Qtoo_IPv4_OSPF_Area_Entry);
}

char *IPv6_Ptr = (char*)(OSPF_Receiver);

/*Cast of the Qtoo_IPv6_OSPF_Area_Entry_t structure is made to the IPv6_Ospf_Area_Ptr */
/*pointer and the displacement will be Offset_To_IPv4_Area_Lst bytes, to aim at the */
/*Offset_To_IPv6_Area_Lst of the list in the structure */

Qtoo_IPv6_OSPF_Area_Entry_t *IPv6_Ospf_Area_Ptr=(Qtoo_IPv6_OSPF_Area_Entry_t*)(IPv6_Ptr+OSPF_Ptr->Offset_To_IPv6_Area_Lst);
for(unsigned long i=0;i < OSPF_Ptr->Number_Of_IPv6_Area_Lst_Ent; i++)
{
    printf("\nPrinting IPv6 Area Number %d",i);
    printf("\nArea ID: %s",IPv6_Ospf_Area_Ptr->AreaID);
    printf("\nOSPF_Runs: %d",IPv6_Ospf_Area_Ptr->SPF_Runs);
    printf("\nArea_BR_Count: %d",IPv6_Ospf_Area_Ptr->Area_BR_Count);
    printf("\nTotal_Number_of_LSA_Area: %d",IPv6_Ospf_Area_Ptr->Total_Number_of_LSA_Area);
    /*Increment IPv6_Ospf_Area_Ptr bytes in order to point to the next structure*/
    IPv6_Ospf_Area_Ptr+=sizeof(Qtoo_IPv6_OSPF_Area_Entry);
}

char *IPv4_Ifc_Ptr = (char*)(OSPF_Receiver);

/*Cast of the Qtoo_IPv4_OSPF_Ifc_Entry_t structure is made to the IPv4_Ifc_Entry_Ptr */
/*pointer and the displacement will be Offset_To_IPv4_Ifc_Lst bytes, to aim at the */
/*Offset_To_IPv4_Ifc_Lst of the list in the structure */

Qtoo_IPv4_OSPF_Ifc_Entry_t *IPv4_Ifc_Entry_Ptr = (Qtoo_IPv4_OSPF_Ifc_Entry_t*)(IPv4_Ifc_Ptr+OSPF_Ptr->Offset_To_IPv4_Ifc_Lst);
for(unsigned long i=0;i < OSPF_Ptr->Number_Of_IPv4_Ifc_Lst_Ent; i++)
{
    printf("\nPrinting IPv4 Interface Entry %d",i);
    printf("\nIp_Address: %s",IPv4_Ifc_Entry_Ptr->IP_Address);
    printf("\nArea_ID: %s",IPv4_Ospf_Area_Ptr->AreaID);
    printf("\nInterface_Type: %d",IPv4_Ifc_Entry_Ptr->Interface_Type);
    printf("\nDesignated_Router_Pri: %d",IPv4_Ifc_Entry_Ptr->Designated_Router_Pri);
    printf("\nTransmission_Dly: %d",IPv4_Ifc_Entry_Ptr->Transmission_Dly);
    printf("\nRetransmission_Dly: %d",IPv4_Ifc_Entry_Ptr->Retransmission_Dly);
    printf("\nHello_Interval: %d",IPv4_Ifc_Entry_Ptr->Hello_Interval);
    printf("\nInactive_Router_Interval: %d",IPv4_Ifc_Entry_Ptr->Inactive_Router_Interval);
    printf("\nPoll_Interval: %d",IPv4_Ifc_Entry_Ptr->Poll_Interval);
    printf("\nInterface_State: %d",IPv4_Ifc_Entry_Ptr->Interface_State);
    printf("\nDesignated_Router: %d",IPv4_Ifc_Entry_Ptr->Designated_Router);
    printf("\nBackup_Designated_Router: %d",IPv4_Ifc_Entry_Ptr->Backup_Designated_Router);
    printf("\nCost: %d",IPv4_Ifc_Entry_Ptr->Cost);
    printf("\nPPP_Connection_Profile: %s",IPv4_Ifc_Entry_Ptr->PPP_Connection_Profile);
    /*Increment IPv4_Ifc_Entry_Ptr bytes in order to point to the next structure*/
    IPv4_Ifc_Entry_Ptr+=sizeof(Qtoo_IPv4_OSPF_Ifc_Entry);
}

char *IPv6_Ifc_Ptr = (char*)(OSPF_Receiver);

/*Cast of the Qtoo_IPv6_OSPF_Ifc_Entry_t structure is made to the IPv6_Ifc_Entry_Ptr */
/*pointer and the displacement will be Offset_To_IPv6_Ifc_Lst bytes, to aim at the */
/*Offset_To_IPv6_Ifc_Lst of the list in the structure */

Qtoo_IPv6_OSPF_Ifc_Entry_t *IPv6_Ifc_Entry_Ptr=(Qtoo_IPv6_OSPF_Ifc_Entry_t*)(IPv6_Ifc_Ptr+OSPF_Ptr->Offset_To_IPv6_Ifc_Lst);
for(unsigned long i=0;i < OSPF_Ptr->Number_Of_IPv6_Ifc_Lst_Ent; i++)

```



```

{
    printf("\nPrinting IPv6 Interface Entry %d",i);
    printf("\nIPv6 Address: %s",IPv6_Ifc_Entry_Ptr->IPv6_Address);
    printf("\nArea_ID: %s",IPv6_Ifc_Entry_Ptr->AreaID);
    printf("\nInterface_Type: %d",IPv6_Ifc_Entry_Ptr->Interface_Type);
    printf("\nDesignated_Router_Pri: %d",IPv6_Ifc_Entry_Ptr->Designated_Router_Pri);
    printf("\nTransmission_Dly: %d",IPv6_Ifc_Entry_Ptr->Transmission_Dly);
    printf("\nRetransmission_Dly: %d",IPv6_Ifc_Entry_Ptr->Retransmission_Dly);
    printf("\nHello_Interval: %d",IPv6_Ifc_Entry_Ptr->Hello_Interval);
    printf("\nInactive_Router_Interval: %d",IPv6_Ifc_Entry_Ptr->Inactive_Router_Interval);
    printf("\nPoll_Interval: %d",IPv6_Ifc_Entry_Ptr->Poll_Interval);
    printf("\nInterface_State: %d",IPv6_Ifc_Entry_Ptr->Interface_State);
    printf("\nDesignated_Router: %d",IPv6_Ifc_Entry_Ptr->Designated_Router);
    printf("\nBackup_Designated_Router: %d",IPv6_Ifc_Entry_Ptr->Backup_Designated_Router);
    printf("\nCost: %d",IPv6_Ifc_Entry_Ptr->Cost);
    /*Increment IPv6_Ifc_Entry_Ptr bytes in order to point to the next structure*/
    IPv6_Ifc_Entry_Ptr+=sizeof(Qtoo_IPv6_OSPF_Ifc_Entry);
}

char *IPv4_Neig_Ptr = (char*)(OSPF_Receiver);

/*Cast of the Qtoo_IPv4_OSPF_Neighbor_Entry_t structure is made to the IPv4_Ifc_Entry_Ptr */
/*pointer and the displacement will be Offset_To_IPv4_Neigh_Lst bytes, to aim at the */
/*Offset_To_IPv4_Neigh_Lst of the list in the structure */

Qtoo_IPv4_OSPF_Neighbor_Entry_t *IPv4_Neig_Entry_Ptr=(Qtoo_IPv4_OSPF_Neighbor_Entry_t*)(IPv4_Neig_Ptr+OSPF_Ptr->Offset_To_IPv4_Neigh_Lst);
for(unsigned long i=0;i < OSPF_Ptr->Number_Of_IPv4_Neigh_Lst_Ent;i++)
{
    printf("\nPrinting IPv4 Neighbor Entry %d",i);
    printf("\nIfc_IP_address: %s",IPv4_Neig_Entry_Ptr->Ifc_IP_address);
    printf("\nNeighbor_IP_Address: %s",IPv4_Neig_Entry_Ptr->Neighbor_IP_Address);
    printf("\nRouterID: %s",IPv4_Neig_Entry_Ptr->RouterID);
    printf("\nNeighbor_Options: %d",IPv4_Neig_Entry_Ptr->Neighbor_Options);
    printf("\nNeighbor_Priority: %d",IPv4_Neig_Entry_Ptr->Neighbor_Priority);
    printf("\nNeighbor_State: %d",IPv4_Neig_Entry_Ptr->Neighbor_State);
    printf("\nNeighbor_Events: %d",IPv4_Neig_Entry_Ptr->Neighbor_Events);
    /*Increment IPv4_Neig_Entry_Ptr bytes in order to point to the next structure*/
    IPv4_Neig_Entry_Ptr+=sizeof(Qtoo_IPv4_OSPF_Neighbor_Entry);
}

char *IPv6_Neig_Ptr = (char*)(OSPF_Receiver);

/*Cast of the Qtoo_IPv6_OSPF_Neighbor_Entry_t structure is made to the IPv6_Ifc_Entry_Ptr */
/*pointer and the displacement will be Offset_To_IPv6_Neigh_Lst bytes, to aim at the */
/*Offset_To_IPv6_Neigh_Lst of the list in the structure */

Qtoo_IPv6_OSPF_Neighbor_Entry_t *IPv6_Neig_Entry_Ptr=(Qtoo_IPv6_OSPF_Neighbor_Entry_t*)(IPv6_Neig_Ptr+OSPF_Ptr->Offset_To_IPv6_Neigh_Lst);
for(unsigned long i=0;i < OSPF_Ptr->Number_Of_IPv6_Neigh_Lst_Ent;i++)
{
    printf("\nPrinting IPv6 Neighbor Entry %d",i);
    printf("\nIfc_IP_address: %s",IPv6_Neig_Entry_Ptr->Ifc_IP_address);
    printf("\nNeighbor_IP_Address: %s",IPv6_Neig_Entry_Ptr->Neighbor_IP_Address);
    printf("\nRouterID: %s",IPv6_Neig_Entry_Ptr->RouterID);
    printf("\nNeighbor_Options: %d",IPv6_Neig_Entry_Ptr->Neighbor_Options);
    printf("\nNeighbor_Priority: %d",IPv6_Neig_Entry_Ptr->Neighbor_Priority);
    printf("\nNeighbor_State: %d",IPv6_Neig_Entry_Ptr->Neighbor_State);
    printf("\nNeighbor_Events: %d",IPv6_Neig_Entry_Ptr->Neighbor_Events);
    /*Increment IPv6_Neig_Entry_Ptr bytes in order to point to the next structure*/
    IPv6_Neig_Entry_Ptr+=sizeof(Qtoo_IPv6_OSPF_Neighbor_Entry);
}

char *IPv4_Vt1_Ptr = (char*)(OSPF_Receiver);

/*Cast of the Qtoo_IPv4_OSPF_Vt1_Entry_t structure is made to the IPv4_Ifc_Entry_Ptr */
/*pointer and the displacement will be Offset_To_IPv4_Vt1_Lst bytes, to aim at the */
/*Offset_To_IPv4_Vt1_Lst of the list in the structure */

Qtoo_IPv4_OSPF_Vt1_Link_Entry_t *IPv4_Vt1_Entry_Ptr=(Qtoo_IPv4_OSPF_Vt1_Link_Entry_t*)(IPv4_Vt1_Ptr+OSPF_Ptr->Offset_To_IPv4_Vt1_Lst);
for(unsigned long i=0;i < OSPF_Ptr->Number_Of_IPv4_Vt1_Lst_Ent;i++)
{
    printf("\nPrinting IPv4 Virtual Link Entry %d",i);
    printf("\nTransist_Area: %s",IPv4_Vt1_Entry_Ptr->Transist_Area);
    printf("\nRouterID: %s",IPv4_Vt1_Entry_Ptr->RouterID);
    printf("\nVt1_Link_Transit_Dly: %s",IPv4_Vt1_Entry_Ptr->Vt1_Link_Transit_Dly);
    printf("\nVt1_Link_Retransmission_Dly: %s",IPv4_Vt1_Entry_Ptr->Vt1_Link_Retransmission_Dly);
    printf("\nVt1_Link_Hello_Interval: %s",IPv4_Vt1_Entry_Ptr->Vt1_Link_Hello_Interval);
    printf("\nVt1_Link_State: %s",IPv4_Vt1_Entry_Ptr->Vt1_Link_State);
    /*Increment IPv4_Vt1_Entry_Ptr bytes in order to point to the next structure*/
    IPv4_Vt1_Entry_Ptr+=sizeof(Qtoo_IPv4_OSPF_Vt1_Link_Entry);
}

char *IPv6_Vt1_Ptr = (char*)(OSPF_Receiver);

/*Cast of the Qtoo_IPv6_OSPF_Vt1_Entry_t structure is made to the IPv6_Ifc_Entry_Ptr */
/*pointer and the displacement will be Offset_To_IPv4_Vt1_Lst bytes, to aim at the */
/*Offset_To_IPv6_Vt1_Lst of the list in the structure */

Qtoo_IPv6_OSPF_Vt1_Link_Entry_t *IPv6_Vt1_Entry_Ptr=(Qtoo_IPv6_OSPF_Vt1_Link_Entry_t*)(IPv6_Vt1_Ptr+OSPF_Ptr->Offset_To_IPv6_Vt1_Lst);
for(unsigned long i=0;i < OSPF_Ptr->Number_Of_IPv6_Vt1_Lst_Ent;i++)
{

```

```

printf("\nPrinting IPv6 Virtual Link Entry %d",i);
printf("\nTransist_Area: %s",IPv6_Vt1_Entry_Ptr->Transist_Area);
printf("\nRouterID: %s",IPv6_Vt1_Entry_Ptr->RouterID);
printf("\nVt1_Link_Transit_Dly: %s",IPv6_Vt1_Entry_Ptr->Vt1_Link_Transit_Dly);
printf("\nVt1_Link_Retransmission_Dly: %s",IPv6_Vt1_Entry_Ptr->Vt1_Link_Retransmission_Dly);
printf("\nVt1_Link_Hello_Interval: %s",IPv6_Vt1_Entry_Ptr->Vt1_Link_Hello_Interval);
printf("\nVt1_Link_State: %s",IPv6_Vt1_Entry_Ptr->Vt1_Link_State);
/*Increment IPv6_Vt1_Entry_Ptr bytes in order to point to the next structure*/
IPv6_Vt1_Entry_Ptr+=sizeof(Qtoo_IPv6_OSPF_Vt1_Link_Entry);
}

char *IPv4_LSA_Ptr = (char*)(OSPF_Receiver);

/*Cast of the Qtoo_IPv4_OSPF_LSA_Entry_t structure is made to the IPv4_Ifc_Entry_Ptr */
/*pointer and the displacement will be Offset_To_IPv4_LSA_Lst bytes, to aim at the */
/*Offset_To_IPv4_LSA_Lst of the list in the structure */

Qtoo_IPv4_OSPF_LSA_Entry_t *IPv4_LSA_Entry_Ptr=(Qtoo_IPv4_OSPF_LSA_Entry_t*)(IPv4_LSA_Ptr+OSPF_Ptr->Offset_to_IPv4_LSA_Lst);
for(unsigned long i=0;i < OSPF_Ptr->Number_Of_IPv4_LSA_Lst_Ent;i++)
{
printf("\nPrinting IPv4 LSA Entry %d",i);
printf("\nLSA_Area_ID: %s",IPv4_LSA_Entry_Ptr->LSA_Area_ID);
printf("\nLSA_Type: %d",IPv4_LSA_Entry_Ptr->LSA_Type);
printf("\nLSA_State_ID: %s",IPv4_LSA_Entry_Ptr->LSA_State_ID);
printf("\nLSA_Router_ID: %s",IPv4_LSA_Entry_Ptr->LSA_Router_ID);
printf("\nLSA_Sequence: %d",IPv4_LSA_Entry_Ptr->LSA_Sequence);
printf("\nLSA_Age: %d",IPv4_LSA_Entry_Ptr->LSA_Age);
printf("\nLSA_Checksum: %d",IPv4_LSA_Entry_Ptr->LSA_Checksum);
/*Increment IPv4_LSA_Entry_Ptr bytes in order to point to the next structure*/
IPv4_LSA_Entry_Ptr+=sizeof(Qtoo_IPv4_OSPF_LSA_Entry_t);
}

char *IPv6_LSA_Ptr = (char*)(OSPF_Receiver);

/*Cast of the Qtoo_IPv6_OSPF_LSA_Entry_t structure is made to the IPv6_Ifc_Entry_Ptr */
/*pointer and the displacement will be Offset_To_IPv6_LSA_Lst bytes, to aim at the */
/*Offset_To_IPv6_LSA_Lst of the list in the structure */

Qtoo_IPv6_OSPF_LSA_Entry_t *IPv6_LSA_Entry_Ptr=(Qtoo_IPv6_OSPF_LSA_Entry_t*)(IPv6_LSA_Ptr+OSPF_Ptr->Offset_to_IPv6_LSA_Lst);
for(unsigned long i=0;i < OSPF_Ptr->Number_Of_IPv6_LSA_Lst_Ent;i++)
{
printf("\nPrinting IPv6 LSA Entry %d",i);
printf("\nLSA_Area_ID: %s",IPv6_LSA_Entry_Ptr->LSA_Area_ID);
printf("\nLSA_Type: %d",IPv6_LSA_Entry_Ptr->LSA_Type);
printf("\nLSA_State_ID: %s",IPv6_LSA_Entry_Ptr->LSA_State_ID);
printf("\nLSA_Router_ID: %s",IPv6_LSA_Entry_Ptr->LSA_Router_ID);
printf("\nLSA_Sequence: %d",IPv6_LSA_Entry_Ptr->LSA_Sequence);
printf("\nLSA_Age: %d",IPv6_LSA_Entry_Ptr->LSA_Age);
printf("\nLSA_Checksum: %d",IPv6_LSA_Entry_Ptr->LSA_Checksum);
/*Increment IPv6_LSA_Entry_Ptr bytes in order to point to the next structure*/
IPv6_LSA_Entry_Ptr+=sizeof(Qtoo_IPv6_OSPF_LSA_Entry);
}
}

```

## 関連概念

16 ページの『OSPF (Open Shortest Path First) API およびコマンド』

OSPF 属性、インターフェース、エリア、範囲、リンク、統計、状態情報を管理し、OSPF アクティビティをログに記録するには、制御言語 (CL) コマンドを使用します。OSPF 統計を検索するには、OSPF 状態情報検索 API を使用します。

---

## コードに関するライセンス情報および特記事項

IBM は、お客様に、すべてのプログラム・コードのサンプルを使用することができる非独占的な著作使用権を許諾します。お客様は、このサンプル・コードから、お客様独自の特別のニーズに合わせた類似のプログラムを作成することができます。

強行法規で除外を禁止されている場合を除き、IBM、そのプログラム開発者、および供給者は「プログラム」および「プログラム」に対する技術的サポートがある場合にはその技術的サポートについて、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任を負わないものとします。

いかなる場合においても、IBM および IBM のサプライヤーならびに IBM ビジネス・パートナーは、その予見の有無を問わず発生した以下のものについて賠償責任を負いません。

1. データの喪失、または損傷。

2. 直接損害、特別損害、付随的損害、間接損害、または経済上の結果的損害
3. 逸失した利益、ビジネス上の収益、あるいは節約すべかりし費用

国または地域によっては、法律の強行規定により、上記の責任の制限が適用されない場合があります。



---

## 付録. 特記事項

本書は米国 IBM が提供する製品およびサービスについて作成したものです。

本書に記載の製品、サービス、または機能が日本においては提供されていない場合があります。日本で利用可能な製品、サービス、および機能については、日本 IBM の営業担当員にお尋ねください。本書で IBM 製品、プログラム、またはサービスに言及していても、その IBM 製品、プログラム、またはサービスのみが使用可能であることを意味するものではありません。これらに代えて、IBM の知的所有権を侵害することのない、機能的に同等の製品、プログラム、またはサービスを使用することができます。ただし、IBM 以外の製品とプログラムの操作またはサービスの評価および検証は、お客様の責任で行っていただきます。

IBM は、本書に記載されている内容に関して特許権 (特許出願中のものを含む) を保有している場合があります。本書の提供は、お客様にこれらの特許権について実施権を許諾することを意味するものではありません。実施権についてのお問い合わせは、書面にて下記宛先にお送りください。

〒106-8711  
東京都港区六本木 3-2-12  
日本アイ・ビー・エム株式会社  
法務・知的財産  
知的財産権ライセンス渉外

以下の保証は、国または地域の法律に沿わない場合は、適用されません。IBM およびその直接または間接の子会社は、本書を特定物として現存するままの状態を提供し、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任を負わないものとします。国または地域によっては、法律の強行規定により、保証責任の制限が禁じられる場合、強行規定の制限を受けるものとします。

この情報には、技術的に不適切な記述や誤植を含む場合があります。本書は定期的に見直され、必要な変更は本書の次版に組み込まれます。IBM は予告なしに、随時、この文書に記載されている製品またはプログラムに対して、改良または変更を行うことがあります。

本書において IBM 以外の Web サイトに言及している場合がありますが、便宜のため記載しただけであり、決してそれらの Web サイトを推奨するものではありません。それらの Web サイトにある資料は、この IBM 製品の資料の一部ではありません。それらの Web サイトは、お客様の責任でご使用ください。

IBM は、お客様が提供するいかなる情報も、お客様に対してなら義務も負うことのない、自ら適切と信ずる方法で、使用もしくは配布することができるものとします。

本プログラムのライセンス保持者で、(i) 独自に作成したプログラムとその他のプログラム (本プログラムを含む) との間での情報交換、および (ii) 交換された情報の相互利用を可能にすることを目的として、本プログラムに関する情報を必要とする方は、下記に連絡してください。

IBM Corporation  
Software Interoperability Coordinator, Department YBWA  
3605 Highway 52 N  
Rochester, MN 55901  
U.S.A.

本プログラムに関する上記の情報は、適切な使用条件の下で使用することができますが、有償の場合もあります。

本書で説明されているライセンス・プログラムまたはその他のライセンス資料は、IBM 所定のプログラム契約の契約条項、IBM プログラムのご使用条件、IBM 機械コードのご使用条件、またはそれと同等の条項に基づいて、IBM より提供されます。

この文書に含まれるいかなるパフォーマンス・データも、管理環境下で決定されたものです。そのため、他の操作環境で得られた結果は、異なる可能性があります。一部の測定が、開発レベルのシステムで行われた可能性があります。その測定値が、一般に利用可能なシステムのものと同じである保証はありません。さらに、一部の測定値が、推定値である可能性があります。実際の結果は、異なる可能性があります。お客様は、お客様の特定の環境に適したデータを確かめる必要があります。

IBM 以外の製品に関する情報は、その製品の供給者、出版物、もしくはその他の公に利用可能なソースから入手したものです。IBM は、それらの製品のテストは行っておりません。したがって、他社製品に関する実行性、互換性、またはその他の要求については確認できません。IBM 以外の製品の性能に関する質問は、それらの製品の供給者をお願いします。

IBM の将来の方向または意向に関する記述については、予告なしに変更または撤回される場合があります、単に目標を示しているものです。

表示されている IBM の価格は IBM が小売り価格として提示しているもので、現行価格であり、通知なしに変更されるものです。卸価格は、異なる場合があります。

本書はプランニング目的としてのみ記述されています。記述内容は製品が使用可能になる前に変更になる場合があります。

本書には、日常の業務処理で用いられるデータや報告書の例が含まれています。より具体性を与えるために、それらの例には、個人、企業、ブランド、あるいは製品などの名前が含まれている場合があります。これらの名称はすべて架空のものであり、名称や住所が類似する企業が実在しているとしても、それは偶然にすぎません。

#### 著作権使用許諾:

本書には、様々なオペレーティング・プラットフォームでのプログラミング手法を例示するサンプル・アプリケーション・プログラムがソース言語で掲載されています。お客様は、サンプル・プログラムが書かれているオペレーティング・プラットフォームのアプリケーション・プログラミング・インターフェースに準拠したアプリケーション・プログラムの開発、使用、販売、配布を目的として、いかなる形式においても、IBM に対価を支払うことなくこれを複製し、改変し、配布することができます。このサンプル・プログラムは、あらゆる条件下における完全なテストを経ていません。従って IBM は、これらのサンプル・プログラムについて信頼性、利便性もしくは機能性があることをほのめかしたり、保証することはできません。

それぞれの複製物、サンプル・プログラムのいかなる部分、またはすべての派生的創作物にも、次のように、著作権表示を入れていただく必要があります。

© (お客様の会社名) (西暦年). このコードの一部は、IBM Corp. のサンプル・プログラムから取られています。 © Copyright IBM Corp. \_年を入れる\_. All rights reserved.

この情報をソフトコピーでご覧になっている場合は、写真やカラーの図表は表示されない場合があります。

---

## プログラミング・インターフェース情報

この「ネットワーク OSPF (Open Shortest Path First) サポート」資料には、プログラムを作成するユーザーが IBM i5/OS のサービスを使用するためのプログラミング・インターフェースが記述されています。

---

## 商標

以下は、IBM Corporation の商標です。

i5/OS

IBM

IBM(ロゴ)

System i

Adobe、Adobe ロゴ、PostScript、PostScript ロゴは、Adobe Systems Incorporated の米国およびその他の国における登録商標または商標です。

IT Infrastructure Library は英国 Office of Government Commerce の一部である the Central Computer and Telecommunications Agency の登録商標です。

Intel、Intel ロゴ、Intel Inside、Intel Inside ロゴ、Intel Centrino、Intel Centrino ロゴ、Celeron、Intel Xeon、Intel SpeedStep、Itanium、および Pentium は、Intel Corporation または子会社の米国およびその他の国における商標または登録商標です。

Linux は、Linus Torvalds の米国およびその他の国における商標です。

Microsoft、Windows、Windows NT および Windows ロゴは、Microsoft Corporation の米国およびその他の国における商標です。

ITIL は英国 Office of Government Commerce の登録商標および共同体登録商標であって、米国特許商標庁にて登録されています。

Java およびすべての Java 関連の商標およびロゴは、Sun Microsystems, Inc. の米国およびその他の国における商標または登録商標です。

UNIX は、The Open Group の米国およびその他の国における登録商標です。

他の会社名、製品名およびサービス名等はそれぞれ各社の商標です。

---

## 使用条件

これらの資料は、以下の条件に同意していただける場合に限りご使用いただけます。

**個人使用:** これらの資料は、すべての著作権表示その他の所有権表示をしていただくことを条件に、非商業的な個人による使用目的に限り複製することができます。ただし、IBM の明示的な承諾をえずに、これらの資料またはその一部について、二次的著作物を作成したり、配布 (頒布、送信を含む) または表示 (上映を含む) することはできません。



**商業的使用:** これらの資料は、すべての著作権表示その他の所有権表示をしていただくことを条件に、お客様の企業内に限り、複製、配布、および表示することができます。ただし、IBM の明示的な承諾をえずにこれらの資料の二次的著作物を作成したり、お客様の企業外で資料またはその一部を複製、配布、または表示することはできません。

ここで明示的に許可されているもの以外に、資料や資料内に含まれる情報、データ、ソフトウェア、またはその他の知的所有権に対するいかなる許可、ライセンス、または権利を明示的にも黙示的にも付与するものではありません。

資料の使用が IBM の利益を損なうと判断された場合や、上記の条件が適切に守られていないと判断された場合、IBM はいつでも自らの判断により、ここで与えた許可を撤回できるものとさせていただきます。

お客様がこの情報をダウンロード、輸出、または再輸出する際には、米国のすべての輸出入関連法規を含む、すべての関連法規を遵守するものとします。

IBM は、これらの資料の内容についていかなる保証もしません。これらの資料は、特定物として現存するままの状態を提供され、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任なしで提供されます。







Printed in Japan