



Open Archives Initiative
ResourceSync フレームワーク仕様



ResourceSync フレームワーク仕様 : 変更通知

2017年7月20日

本バージョン:

<http://www.openarchives.org/rs/notification/1.0.1/notification>

最新バージョン:

<http://www.openarchives.org/rs/notification>

旧バージョン:

<http://www.openarchives.org/rs/notification/1.0/notification>

編集者 :

Martin Klein, Herbert Van de Sompel、ロスアラモス国立研究所
Simeon Warner、コーネル大学
Graham Klyne、オックスフォード大学
Bernhard Haslhofer、ウィーン大学
Michael Nelson、オールドドミニオン大学
Carl Lagoze、ミシガン大学
Robert Sanderson、ゲティ財団

概要

ResourceSync [コア仕様](#)では、デスティネーション (Destination) が進化するリソースとの同期を維持できるようにするために、ソース (Source) が実装できるさまざまな機能で構成される Web のためのプルベース (pull-based) の同期フレームワークを導入しています。本『ResourceSync 変更通知』仕様では、ソースがサポートできる追加的なプッシュベースの機能を記述しています。これは同期の遅延を減らすことを目的としており、また、ソースをサブスクライブ (subscribe) するデスティネーションに、ソースが通知を送信することを課すものです。

本ドキュメントのステータス

本仕様は『[ResourceSync Framework Specifications](#) (ResourceSync フレームワーク仕様)』を構成するドキュメントのうちの1つです。[ResourceSync Google Group](#) (ResourceSync Google グループ) へのフィードバックを歓迎します。

目次

1. [はじめに](#)
 - 1.1 [動機付けの例](#)
 - 1.2 [表記規則](#)
2. [Change Notification チャンネル](#)
3. [Change Notification](#)
4. [トランスポート・プロトコル: WebSub](#)
 - 4.1 [ソースがハブに通知を送信](#)
 - 4.2 [デスティネーションがハブをサブスクライブして通知を受信](#)
 - 4.3 [ハブがデスティネーションに通知を送信](#)
 - 4.4 [デスティネーションがハブのサブスクライブを解除](#)
5. [Change Notification チャンネルの公表](#)
6. [参考文献](#)

付録

- A. [謝辞](#)
- B. [変更ログ](#)

1.はじめに

本仕様では、[ResourceSync フレームワーク](#)において定義される「[Change Notification](#) (変更通知)」機能について説明します。プッシュベースの通知機能は、ResourceSync [コア仕様](#)で定義されるプルベースの機能に固有のソースとデスティネーション間の同期の遅延を減らすことを目的としています。「[Change Notification](#)」機能は、たとえば、リソースの作成または削除など、あるソースがそのリソースに対する変更に関して通知を送信することで構成されます。

「[Framework Notification](#) (フレームワーク通知)」機能は、たとえば、新しい「Resource List (リソース・リスト)」の公開や「Change List (変更リスト)」の更新など、あるソースがその ResourceSync フレームワークの実装への変更に関する通知を送信することで構成されますが、これについては別の仕様で説明しています。

1.1. 動機付けの例

Linked Data に基づくアプリケーションでは、さまざまなデータセットのリソースが統合されますが、それらのリソースは異なる頻度で変更する可能性があります。その例として、とりわけ、Last.FM、DBPedia、MusicBrainz、GeoNames などからデータを統合する BBC Linked Data のアプリケーションがあげられます。このような統合されたリソース・コレクションに基づくサービスの正確さは、提供されたリソースが最新のものであるかどうかによって左右されます。たとえば、LiveDBPedia リソースの更新頻度は、平均して 1 秒あたり約 2 回の変更であることが確認されています。このことから、「Change Notification」機能が対応しようとしている同期が大きな課題となることがいえます。

1.2. 定義と名前空間接頭辞バインディング

本仕様では、「リソース(resource)」、「表現(representation)」、「要求(request)」、「応答(response)」、「コンテンツ・ネゴシエーション(content negotiation)」、「クライアント(client)」、および、「サーバー(server)」の各用語を、『[Architecture of the World Wide Web](#) (World Wide Web のアーキテクチャ)』の記述に従って使用します。

本ドキュメントでは、以下の名前空間接頭辞バインディングが使用されます。

接頭辞	名前空間 URI	記述
なし	http://www.sitemaps.org/schemas/sitemap/0.9	Sitemap protocol で定義されたサイトマップ XML 要素
rs	http://www.openarchives.org/rs/terms/	本仕様で導入された要素と属性の名前空間

表 1: 本ドキュメントで使用されている名前空間接頭辞バインディング

2. Change Notification チャンネル

「Change Notifications」は、リソースの変更イベントでも、特に、同期の対象となるソースのリソースが作成、更新または削除されたときに、デスティネーションに通知するために送信されます。これらの通知のペイロード (payload) については、[3 項](#)で説明します。通知は、[4 項](#)で解説するプッシュ技術が提供する 1 つ以上のチャンネル (channel) により、ソースからデスティネーションに送信されます。

[図 1](#) は、単一の [リソースのセット](#) (set of resource) からなり、上部に「[Source Description](#) (ソースの記述)」と「[Capability List](#) (機能リスト)」が配置された、ソースにとっての [ResourceSync フレームワークの構造](#)を示しています。

「Capability List」では、「[Resource List](#) (リソース・リスト)」「[Change List](#)

(変更リスト)」「[Resource Dump](#) (リソース・ダンプ) 」および「[Change Dump](#) (変更ダンプ) 」の4つの異なる機能が公表されます。この図には、「Change Notification Channel (変更通知チャンネル) 」(黄色の六角形) も表示されており、特定のリソースのセットのリソース変更に関する情報を送信するために使用されることが示されています。これらのリソースへの変更は、「Change Notification」チャンネルを介して変更通知として伝達されます。

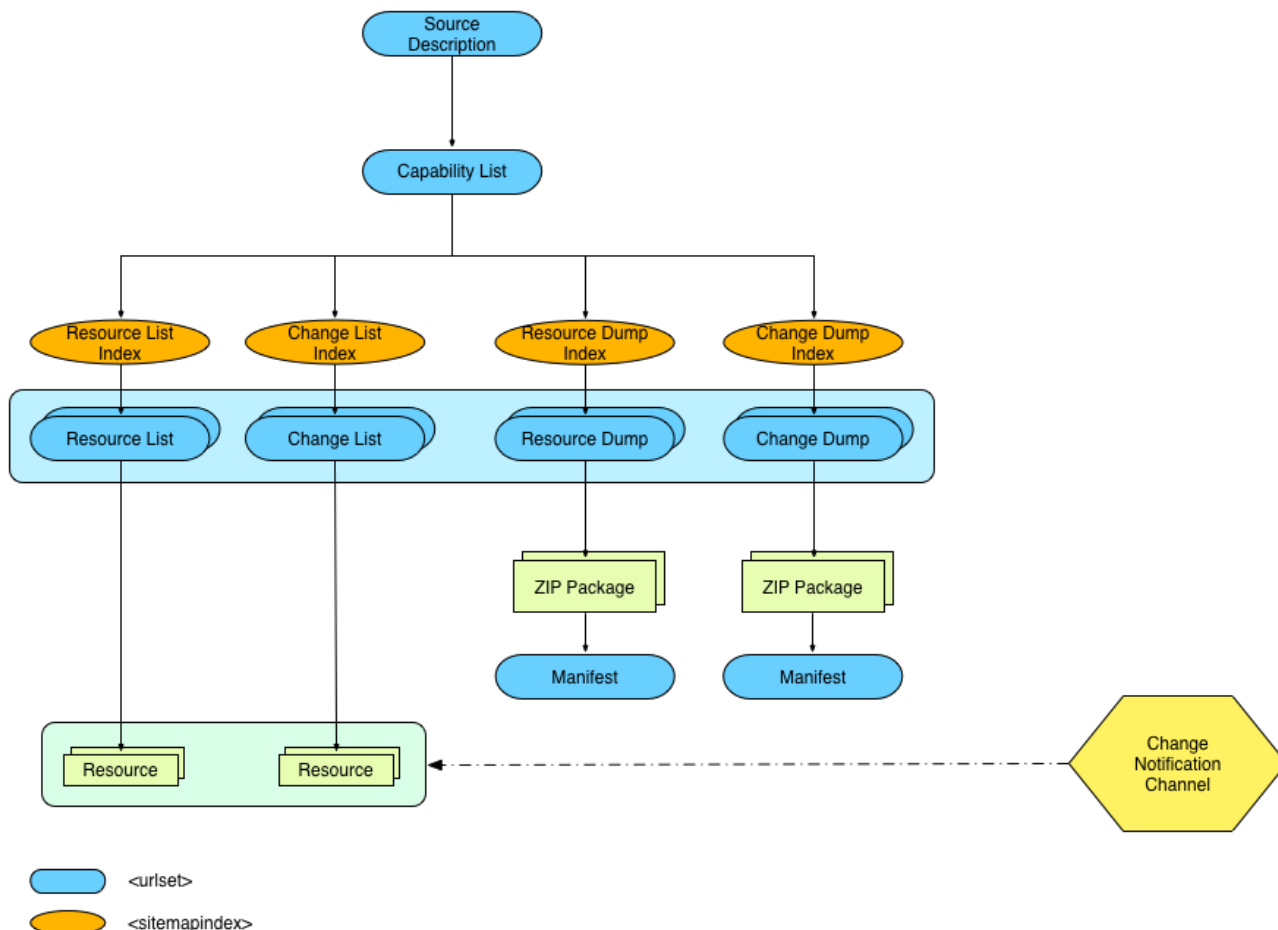


図 1 : ResourceSync フレームワーク構造における「Change Notification」チャンネル

ResourceSync フレームワークにより、ソースは複数のリソースのセットを提供できるようになりますが、その際、「Source Description」は、リソースのセットごとに1つずつ存在する複数の「Capability List」を指します。この場合、「Change Notification」がサポートされる個々のリソースのセットごとに、専用の「Change Notification」チャンネルが提供されなければなりません。リソースの変更に関する通知は、当該リソースが存在するリソースのセットに関連付けられた「Change Notification」チャンネルを介して送信されます。リソースが複数のリソースのセットに存在する場合には、通知は、それらのリソースのセットに関連付けられた各「Change Notification」チャンネルで送信されます。

「Change Notifications」は、異なるチャンネルで送信されなければなりません。

図 2 は、ソースが複数のリソースのセットを提供し、そのため、「Source Description」が、リソースのセットごとに1つずつ複数の「Capability List」を指しているシナリオ（ここでは Capability List 1 および Capability List 2）を示しています。図 2 には、リソースのセットごとに、専用の「Change Notification」チャンネルが存在することが示されています。たとえば、Change Notification Channel 1 は、Capability List 1 でカバーされるリソースのセットの一部であるリソースへの変更に関する変更通知を送信するために使用されます。

「Change Notification」チャンネルの作成と削除が、更新された「Capability List」に反映されることに注意してください（5 項を参照のこと）。本仕様では、通知チャンネルに関する個別の通知については定義しないものとします。

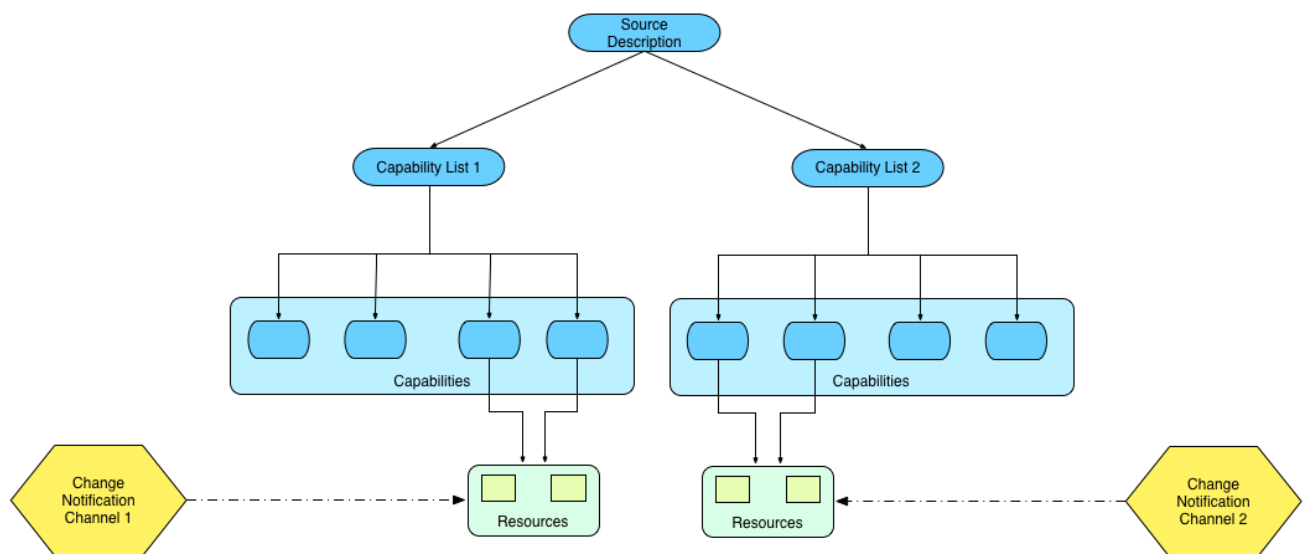


図 2 : 複数のリソースのセットの「Change Notification」チャンネル

3. Change Notification

変更通知（change notification）は、2 項で説明したように、同期の対象となる 1 つ以上のリソースが変更されたことをソースがデスティネーションに通知したい場合に、適切な「Change Notification」チャンネルを介して送信されます。

「Change Notification」チャンネルをサブスクライブすることにより、デスティネーションは同期の遅延を削減し、ソースの「Change Lists」が存在する場合には、定期的にポーリング（polling）してリソースに変更が発生したかどうかを判断する作業を回避することができます。

変更通知の形式は、コア仕様の 12 項で導入している「Change List」の形式に非常に類似しています。変更通知のすべてのエントリは、日付の前のものから時系列に提供されなければなりません。すなわち、最も古い変更日時のリソースは変更通知ペイロードの先頭に、最も新しい変更日時のリソースは末尾にリストされ

なければなりません。この形式は、サイトマップ・プロトコルによって導入された<urlset>ドキュメント形式に基づいています。これは<urlset>ルート要素と次の構造を持ちます。

- <urlset>の子要素である必須の<rs:md>には、change-notification の値を持つ capability 属性がなければなりません。また、必須の from 属性および until 属性も持ちます。from 属性は、変更通知が、属性の値として表現された日時以降にソースでリソースのセットに発生したすべての変更を含むことを示しています。until 属性は、変更通知が、属性の値として表現された日時までにソースでリソースのセットに発生したすべての変更を含むことを示しています。ソースが送信する連続した変更通知は、連続し重複しない from/until の時区間をカバーしていなければなりません。これにより、デスティネーションでは、すべての変更通知を受信したかどうかを確認し、ソースが送信した順序で受信されなかった場合には、適切な順序でそれら进行处理することができます。
- <urlset> の子要素である任意の<rs:ln>は、関係タイプ up を持つ「Capability List」を指します。
- リソースの変更ごとに、<url>の子要素である一つの<urlset>。この要素には属性はありませんが、子要素を使用して、変更されたリソースに関する情報を伝えます。<url>要素は、以下の子要素を持ちます。
 - 必須の子要素<loc>は、変更されたリソースの URI を提供します。
 - [コア仕様の 7 項](#)で説明されるセマンティクスを持つ任意の子要素<lastmod>。
 - 変更の性質を伝えるために、必須の子要素<rs:md>は、属性 change を持たなければなりません。その値は、created、updated または deleted となります。[コア仕様の 7 項](#)で説明されているように、変更の日時を伝える任意の datetime 属性を持ちます。子要素<rs:md>は、さらに、[コア仕様の 7 項](#)で説明されているように、hash、length および type の属性を持つことができます。
 - [コア仕様の 14 項](#)および [7 項](#)で説明されているように、任意の子要素<rs:ln>は関連リソースにリンクしています。

変更通知では、サイトマップ・プロトコルで導入されている<sitemapindex>ドキュメント形式は使用しません。ソースで非常に多数の同時的な変更が行われた場合には、通知は、<urlset>ドキュメントを使用して、一連の変更通知に分割されなければなりません。

例 1 は、2つのリソースへの変更の説明を含む変更通知のペイロードを示しています。

```
<?xml version="1.0" encoding="UTF-8"?>
<urlset xmlns="http://www.sitemaps.org/schemas/sitemap/0.9"
  xmlns:rs="http://www.openarchives.org/rs/terms/">
  <rs:ln rel="up"
    href="http://example.com/dataset1/capabilitylist.xml"/>
  <rs:md capability="change-notification"
    from="2013-01-03T00:00:00Z"
    until="2013-01-03T00:10:00Z"/>
  <url>
    <loc>http://example.com/res1</loc>
    <lastmod>2013-01-01T13:03:00Z</lastmod>
    <rs:md change="created"
      datetime="2013-01-03T00:07:22Z"
      hash="md5:1584abdf8ebdc9802ac0c6a7402c03b6"
      length="8876"
      type="application/pdf"/>
  </url>
  <url>
    <loc>http://example.com/res2</loc>
    <rs:md change="updated"
      datetime="2013-01-03T00:08:52Z"
      hash="md5:1e0d5cb8ef6ba40c99b14c0237be735e
      sha-
256:854f61290e2e197a11bc91063afce22e43f8ccc655237050ace766adc68dc784"
      length="14599"
      type="text/html"/>
  </url>
</urlset>
```

例 1 : 変更通知のペイロード

4. トランスポート・プロトコル : WebSub

ResourceSync フレームワークの通知機能を起動 (bootstrap) するために、単一のトランスポート・プロトコル (transport protocol) である [WebSub](#) が選択されています。WebSub は、シンプルな HTTP ベースのパブリッシュ/サブスクライブ (publish/subscribe) ・プロトコルであり、非常に高い頻度で変更通知を送信する必要がないユースケースでうまく機能することが期待されます。

以下の「Change Notifications」での WebSub の使用法の説明は、「[Framework Notifications](#)」での使用法の説明と基本的に同じです。ただし、それぞれの仕様を独立したものにするため、また、「Change Notifications」と「Framework Notifications」は異なるチャンネルで送信しなければならないため、両方の仕様でこの説明を繰り返します。

[表 2](#) は、ResourceSync および WebSub で使用される用語をマップ (map) しています。WebSub には、パブリッシュ/サブスクライブのパラダイムを実装するために、ソースとデスティネーションの間のパイプとして機能する hub が導入さ

れています。ハブは、ソース自体または第三者によって操作することが可能です。また、hub URI によって一意に識別されます。WebSub の topic は、本仕様で使用されるチャンネルの概念に対応しています。トピック (topic) は、その topic URI によって一意に識別されます。したがって、ソースは、リソースのセットごとに、変更通知のための専用のトピック (およびトピック URI) を持ちます。

ResourceSync	WebSub
ソース (Source)	パブリッシャー (Publisher)
デスティネーション (Destination)	サブスクライバー (Subscriber)
チャンネル (Channel)	トピック (Topic)
通知 (Notification)	通知 (Notification)
	ハブ (Hub)

表 2 : ResourceSync と WebSub 間の用語のマッピング

本項の残りの部分では、ResourceSync における WebSub の使用について説明します。メカニズム全体を適切に理解する上で不可欠な WebSub プロトコルに関する情報のみを提供します。WebSub プロトコルの詳細は、[WebSub](#) 仕様に記載されています。[図 3](#) は、ソース、ハブおよびデスティネーション間の HTTP の相互作用の概要を示しています。これらについては、本項の残りの部分で詳しく説明します。

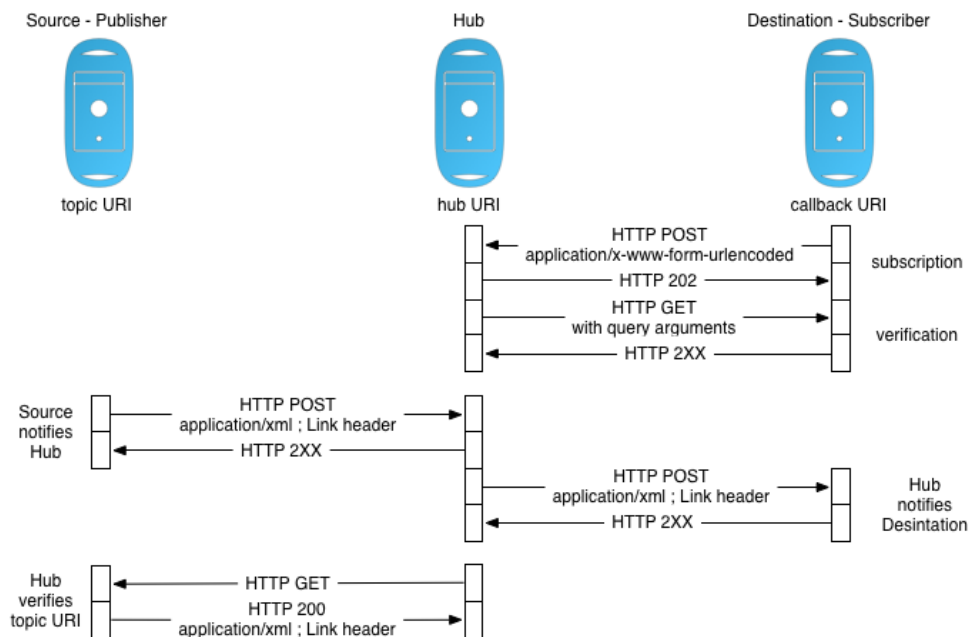


図 3 : ソース、ハブおよびデスティネーション間の HTTP 相互作用

4.1. ソースがハブに通知を送信

WebSub プロトコルは、ソースが通知をハブに通信する方法に関して、特定のガイドラインを提供していません。ResourceSync の変更通知のメカニズムは次のとおりです。

- リソースのセットごとに、ソースは、変更通知のための専用のトピック URI を持ちます。トピック URI に対して発行された HTTP HEAD/GET 要求に対する応答には、HTTP "200 OK" 応答コード、値が `application/xml` の HTTP `Content-Type` ヘッダー、および、次のリンクを持つ HTTP Link ヘッダーがなければなりません。
 - 送信された通知のトピック URI を `href` 属性の値として提供する `self` 関係タイプを持つ必須のリンク。
 - ハブ URI を `href` 属性の値として提供する `hub` 関係タイプを持つ必須のリンク。

トピック URI で識別されるリソースには、通知のペイロードと同じ `Content-Type (application/xml)` がなければなりません。トピック URI でコンテンツを提供することを意図している場合には、最新の通知のペイロードが適切でしょう。トピック URI でコンテンツが提供されていない場合には、値 0 の応答ヘッダー `Content-Length` が返されなければなりません。この通信は、[図 3](#) には、「Hub verifies topic URI (ハブがトピック URI を検証)」と示されています。

- ソースは、変更通知のペイロードをハブに送信します。通知ペイロードは、ハブ URI に対して発行される HTTP POST のボディ (body) として提供されます。この HTTP POST には、`application/xml` の値を持つ HTTP `Content-Type` ヘッダーと、次のリンクを持つ HTTP Link ヘッダーがなければなりません。
 - 送信された通知のトピック URI を `href` 属性の値として提供する `self` 関係タイプを持つ必須リンク。
 - ハブ URI を `href` 属性の値として提供する `hub` 関係タイプを持つ必須リンク。

これは、[図 3](#) には、「Source notifies Hub (ソースがハブに通知)」の右向き矢印として示されています。

- ハブが第三者によって運用されている場合には、HTTP POST が正当なソースによって送信されたものかどうかを検証した方がよいでしょう。ハブは、その通知をリレー (relay) する前に、たとえば、1 回限りの登録メカ

ニズムを介するなどの方法で、ソースとの信頼関係を確立することにより、これを行うことができます。ハブが通知をリレーした後でも、ソースが WebSub プロトコルに準拠しているかどうかを確認することが可能です。これは、次のすべてが成立する場合に当てはまります。

- HTTP POST に、前述のリンクを持つ HTTP Link ヘッダーが含まれている。
- トピック URI に対して発行された HTTP HEAD/GET への応答に、前述のリンクを持つ HTTP Link ヘッダーが含まれている。
- ハブは、変更通知のペイロードの送信に次のように応答します。
 - 送信が成功した場合、応答には、HTTP "200 OK"ステータス・コード (status code) が含まれていなければならない。
 - 送信が成功しなかった場合、応答には、HTTP 3XX、4XX または 5XX ステータス・コードが含まれていなければならない。

これは、[図 3](#)には、「Source notifies Hub (ソースがハブに通知)」の左向き矢印として示されています。

[例 2](#) は、[例 1](#) の変更通知ペイロードを送信するために、ソースがそのハブに対して発行した HTTP POST を示しています。簡潔にするために、ペイロードは完全には表示していません。第三者のハブ URI は `http://hub.example.org/websub/` で、dataset1 に関連する変更通知のためのソースのトピック URI (チャンネル) は `http://example.com/dataset1/change/` です。

```
POST /websub/ HTTP/1.1
Host: http://hub.example.org
Content-Type: application/xml
Link: <http://example.com/dataset1/change/> ; rel="self",
      <http://hub.example.org/websub/> ; rel="hub",
      <http://www.example.com/dataset1/capabilitylist.xml> ; rel="resourcesync"
Content-Length: 849

<?xml version="1.0" encoding="UTF-8"?>
<urlset ...
```

例 2：ソースがハブに変更通知ペイロードを送信するために使用する HTTP POST

4.2. デスティネーションがハブをサブスクライブして通知を受信

デスティネーションは、[WebSub](#) の「Subscribing and Unsubscribing (サブスクライブとサブスクライブ解除)」項で説明されているプロセスを使用して、ソースのトピックをサブスクライブします。このプロセスは、必須のサブスクリプション要求フェーズとサブスクリプション検証 (verification) フェーズで構成されています。

- **図 3**に「subscription」と示されているサブスクリプション要求フェーズでは、デスティネーションは、ソースのトピック URI（チャンネル）に関連する通知のサブスクライブを希望することを示します。これを行うには、デスティネーションは、ハブ URI に対して発行された HTTP POST のボディとしてフォーム（form）を提供します。この HTTP POST は、値が `application/x-www-form-urlencoded` の HTTP `Content-Type` ヘッダーを持たなければならない、また、フォームには次の情報が含まれます。
 - 値としてデスティネーションのコールバック URI を持つ必須の `hub.callback` パラメータ。これは、ハブがソースのトピック URI に関する通知を送信する先の URI になります。
 - 値として `subscribe` を持つ必須の `hub.mode` パラメータ。
 - デスティネーションがサブスクライブを希望するソースのトピック URI を値として持つ必須の `hub.topic` パラメータ。
 - デスティネーションがサブスクリプションをアクティブにしておくことを望む秒数を値として持つ任意の `hub.lease_seconds` パラメータ。値を提供するデスティネーションは、ハブがこれに従う場合とそうでない場合があることに注意する必要があります。

ハブは、要求が受信され、検証されることを示す、HTTP "202 Accepted" 応答コードでサブスクリプション要求に応答しなければなりません。

- **図 3**に「verification」と示されているサブスクリプション検証フェーズでは、ハブは、デスティネーションが本当にサブスクライブするつもりであったかどうか、また、デスティネーションが提供したコールバック URI を制御できるかどうかを検証します。これを行うには、ハブは、次のクエリ文字列引数（query string arguments）が追加されたデスティネーションのコールバック URI に対して、HTTP GET を発行します。
 - 値として `subscribe` を持つ必須の `hub.mode` パラメータ。
 - サブスクリプション要求で指定されたトピック URI を値として持つ必須の `hub.topic` パラメータ。
 - ハブが生成したランダムな文字列を値として持つ必須の `hub.challenge` パラメータ。
 - ハブがサブスクリプションをアクティブに保つ秒数を値として持つ必須の `hub.lease_seconds` パラメータ。この実際のサブスクリプション期間は、デスティネーションが要求したものと任意に異なる場合があります。付与されるサブスクリプションの期間は、300 秒（5

分)以下でなく、かつ、2678400秒(1か月)以上でないことが推奨されます。これらの推奨される制限はやや恣意的ですが、下限が頻繁なサブスクリプションの更新による過負荷を防ぐことを目的としている一方で、上限はキャンセルされていないサブスクリプションが予測可能な期間内に期限切れになるように選択されたものです。継続的なサブスクリプションを維持するために、デスティネーションは、許可されたサブスクリプション期間を記録し、通知の受信を継続する場合には、指定された期間が終了する前に、新しいサブスクリプション要求を発行しなければなりません。

サブスクライバー(subscriber)は、HTTP 2XXステータス・コードを持ち、ボディにhub.challengeの値を持つ応答を送信することで、サブスクライブする意思を確認します。その他の応答は、サブスクライブする意図がなかったことを示します。

例 3は、デスティネーションがハブ URI <http://hub.example.org/websub/>に対して発行した HTTP POST が、コールバック URI

<http://destination.example.net/callback/>で dataset1 に関連する変更通知を受信する手段として、ソースのトピック URI (チャンネル)

<http://example.com/dataset1/change/>へのサブスクリプションを要求していることを示しています。

```
POST /websub/ HTTP/1.1
Host: http://hub.example.org
Content-Type: application/x-www-form-urlencoded
Content-Length: 141

hub.mode=subscribe&hub.topic=http%3A%2F%2FAexample.com%2Fdataset1%2Fchange%2F&hub.callback=http%3A%2F%2Fdestination.example.net%2Fcallback%2F&hub.lease_seconds=3600
```

例 3 : デスティネーションがソースの通知チャンネルをサブスクライブするためにハブに行く要求

例 4は、ハブがデスティネーションのコールバック URI に対して発行した HTTP GET が、デスティネーションにサブスクライブする意図があったことを検証することを示しています。

```
GET
/callback/?hub.mode=subscribe&hub.topic=http%3A%2F%2FAexample.com%2Fdataset1%2Fchange%2F
&hub.challenge=c0cc4630-5116-11e3-8f96-0800200c9a66&hub.lease_seconds=2400
HTTP/1.1
Host: http://destination.example.net
Connection: Close
```

例 4 : デスティネーションのサブスクライブへの意図を検証するためのハブの要求

例 5 は、**例 4** のハブのサブスクリプション検証要求に対するデスティネーションの応答を示しています。デスティネーションがサブスクリプションを要求していることを示しています。

```
HTTP/1.1 200 OK
Date: Tue, 19 Nov 2013 12:42::13 GMT
Content-Type: text/plain; charset=UTF-8
Content-Length: 36
Connection: Close

c0cc4630-5116-11e3-8f96-0800200c9a66
```

例 5：デスティネーションのサブスクライブへの意図を検証するためのハブの要求

4.3. ハブがデスティネーションに通知を送信

ハブは、ソースから変更通知を受け取ると、それをサブスクライブする（複数の）デスティネーションに渡します。**図 3**に「Hub notifies Destination（ハブがデスティネーションに通知）」と示されているプロセスは、次のとおりです。

- ハブは、ソースから受信した通知ペイロードを、デスティネーションのコールバック URI に対して発行された HTTP POST のボディとして提供します。この HTTP POST には、`application/xml` の値を持つ HTTP `Content-Type` ヘッダー、および、次のリンクを持つ HTTP Link ヘッダーがなければなりません。
 - 通知のトピック URI（チャンネル）を `href` 属性の値として提供する `self` 関係タイプを持つ必須のリンク。
 - ハブ URI を `href` 属性の値として提供する `hub` 関係タイプを持つ必須のリンク。
- デスティネーションのコールバック URI からの成功した応答には、HTTP 2XX コードが含まれていなければなりません。ハブはその他のすべてのサブスクライバー応答コードを失敗と見なさなければなりません。
- デスティネーションは、ハブから受信したペイロードについて、たとえば、有効な XML であることや、ルート要素 `<urlset>` があることなどの確認をすることで、検証した方がよいかもしれません。

例 6 は、ハブがデスティネーションのコールバック URI に対して発行する HTTP POST が、**例 2** のソースから受信した通知をリレーすることを示しています。簡潔にするために、ペイロードは完全には表示していません。

```
POST /callback/ HTTP/1.1
Host: http://destination.example.net
Content-Type: application/xml
Link: <http://example.com/dataset1/change/> ; rel="self",
      <http://hub.example.org/websub/> ; rel="hub",
      <http://www.example.com/dataset1/capabilitylist.xml> ; rel="resourcesync"
Content-Length: 849

<?xml version="1.0" encoding="UTF-8"?>
<urlset ...
```

例 6：ハブがソースの変更通知ペイロードをデスティネーションに送信するために使用する HTTP POST

4.4. デスティネーションがハブのサブスクライブを解除

デスティネーションがソースのトピック URI からサブスクライブを解除するメカニズムは、[4.1 項](#)で説明したとおりですが、`hub.mode` の値として `unsubscribe` の代わりに `unsubscribe` を使用します。

5. Change Notification チャンネルの公表

「Change Notification」機能は、[ResourceSync コア仕様](#)で定義されている機能と同様に、「Capability List」を介して公表されます。各リソースのセットが専用の「Change Notification」チャンネルを持つため、該当するチャンネルは、それぞれのリソースのセットに対応する「Capability List」で公表されます。

[図 4](#) は、「Capability List」で公表された「Change Notification」チャンネルを示しています。この図では、専用の「Change Notification」チャンネルを公表する「Capability List」が 1 つのみ存在する構造が示されています。その他の「Capability List」は、それぞれ異なるリソースのセットに関連しており、それぞれの通知チャンネルを公表します。「Capability List」は、「Change Notifications」の他にも、[コア仕様](#)で導入された「Resource List」や「Change List」、[アーカイブ仕様 \(archiving specification\)](#)で導入されたアーカイブ機能など、その他の機能を公表することができます。

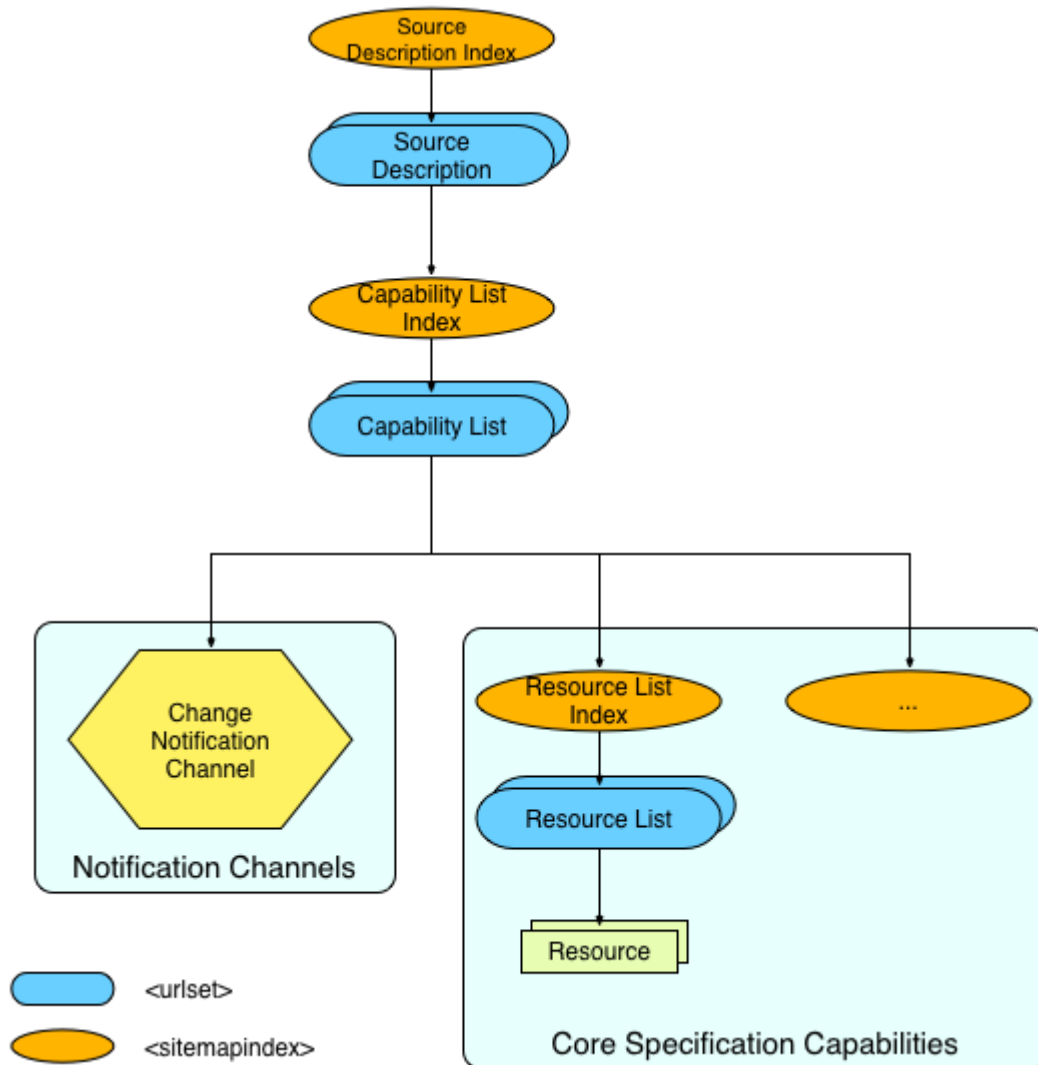


図 4 : 「Change Notification」チャンネルの検出

例 7 は、[コア仕様の例 13](#) から、「Change Notification」チャンネルの検出リンクが追加された「Capability List」を示しています。WebSub のトピック URI は <loc> 要素で提供されており、ハブ URI は <loc> の子要素である <rs:ln> を使用して提供されています。その <rs:ln> は、rel 属性の値として hub を、href 属性の値としてハブ URI を持たなければなりません。「Change Notification」機能を示すために、capability 属性に change-notification 値が導入されていることに注意してください。

```
<urlset xmlns="http://www.sitemaps.org/schemas/sitemap/0.9"
  xmlns:rs="http://www.openarchives.org/rs/terms/">
  <rs:ln rel="describedby"
    href="http://example.com/info_about_set1_of_resources.xml"/>
  <rs:ln rel="up"
    href="http://example.com/source_description.xml"/>
  <rs:md capability="capabilitylist"/>
  <url>
    <loc>http://example.com/dataset1/resourcelist.xml</loc>
    <rs:md capability="resourcelist"/>
  </url>
  <url>
    <loc>http://example.com/dataset1/resourcedump.xml</loc>
    <rs:md capability="resourcedump"/>
  </url>
  <url>
    <loc>http://example.com/dataset1/changelist.xml</loc>
    <rs:md capability="changelist"/>
  </url>
  <url>
    <loc>http://example.com/dataset1/changedump.xml</loc>
    <rs:md capability="changedump"/>
  </url>
  <url>
    <loc>http://example.com/dataset1/change/</loc>
    <rs:ln rel="hub" href="http://hub.example.org/websub"/>
    <rs:md capability="change-notification"/>
  </url>
</urlset>
```

例 7 : WebSub の変更通知チャンネルを検出するためのエントリーを含む「Capability List」

6. 参考文献

[WebSub]

Genestoux, Julien and Aaron Parecki, eds. [WebSub](#), W3C Candidate Recommendation, April 11, 2017.

[Sitemaps]

[Sitemaps XML Format](#). sitemaps.org, last updated February 27, 2008. Available at: <http://www.sitemaps.org/protocol.html>

[Web Architecture]

Jacobs, Ian, and Norman Walsh, eds. [Architecture of the World Wide Web, Volume One](#). W3C Recommendation. World Wide Web Consortium, December 15, 2004. Available at: <http://www.w3.org/TR/webarch/>

A. 謝辞

本仕様は [NISO](#) と [Open Archives Initiative](#) (オープン・アーカイブス・イニシアチブ) の共同作業です。ResourceSync の初期資金は、[Alfred P. Sloan Foundation](#) (アルフレッド P スローン 基金) によって提供されました。英国の参加は [Jisc](#) によってサポートされました。

B. 変更ログ

日時	編集者	記述
2017-07-20	simeon, martin, herbert	PubSubHubbub の代わりに WebSub を使用し、一貫して <code>change-notification</code> 機能の値を使用するように更新
2017-01-18	herbert, simeon	PubSubHubbub の Internet Archive (インターネット・アーカイブ) のコピーにリンク、コンテンツへの変更はなし
2016-08-10	herbert, martin, simeon	バージョン 1.0。仕様から Framework Notification を削除、Core Framework (コア・フレームワーク) の変更に関して更新
2014-03-24	graham, herbert	バージョン 0.9。ソースとハブ間の通信から ResourceSync 固有の要件を削除
2013-12-18	herbert, martin, rob, simeon	バージョン 0.8.1。PubSubHubbub を使用
2013-11-12	martin, herbert, rob, simeon	バージョン 0.8。WebSocket を使用



この著作物は、[Creative Commons Attribution-Share Alike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/) (クリエイティブ・コモンズ[表示-継承]4.0 国際ライセンス)の下でライセンスされています。