



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



ResourceSync フレームワーク仕様： フレームワーク通知

2017年7月20日

本バージョン:

http://www.openarchives.org/rs/notification/1.0.1/framework_notification

最新バージョン:

http://www.openarchives.org/rs/framework_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 フレームワーク通知』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。ソースが、その ResourceSync の実装への変更について、デスティネーションに通知することを目的としています。すなわち、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. [Framework Notification チャンネル](#)
3. [Framework 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. [Framework 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「[Framework Notification](#) (フレームワーク通知)」機能について説明します。このプッシュベースの通知機能は、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。「[Change Notification](#) (変更通知)」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ために、ソースがそのリソースの変更に関して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

ResourceSync [コア仕様](#)で指定されているプルベースの機能により、デスティネーションは、ソースの進化するリソースの状態について、常に情報を得ることができるようになっています。ただし、たとえば、ソースが新しい「Resource List」を発行したかどうか、「Change List」を更新したかどうかなどについて、いつデスティネーションが確認する必要があるかについては、回

答を提示していません。デスティネーションにとって実際的な解決策は、それまでの更新頻度の経験に基づいて設定した頻度で、ソースを繰り返しポーリング (poll) することでしょう。[Framework Notification](#) 機能とは、ソースの ResourceSync 環境への変更についてデスティネーションに通知することであり、それによってソースをポーリングするトリガーを明示的に提供し、また、そうすることで不確定な要素を取り除き、同期プロセスを最適化するものです。このアプローチでは、特に、ソースでの変更が頻繁でないにも関わらず、デスティネーションが低遅延での更新を求める場合に、その効率性が顕著に認められます。

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. Framework Notification チャンネル

「Framework Notifications」は、ソースの ResourceSync フレームワークの実装の機能への変更、たとえば、ソースの「Change List」や「Capability List」の作成、更新または削除について、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、「Framework Notification Channel (フレームワーク通知チャンネル)」(赤の六角形) も表示されており、さまざまな機能ドキュメント (例、

「Resource List」、「Change List」など)や、「Capability List」および「Source Description」への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。これらのドキュメントへの変更は、「Framework Notification」チャンネルを介してフレームワーク通知として伝達されます。

図1は、インデックス・ドキュメント (index document) のレベルでの変更について、フレームワーク通知が送信されないことも示しています。たとえば、「Resource List Index (リソース・リスト・インデックス)」の下に存在する新しい「Resource Lists」が作成された場合には、フレームワーク通知は、「Resource Lists」のいずれかの作成についてのみ送信され、それらを含む「Resource List Index」の作成または更新については送信されません。ソースには、通知時に、「Resource List Index」がコンポーネントとなるすべての新しい「Resource Lists」を確実に指すようにする責任があります。また、インデックスの下に存在するドキュメントへの変更について送信されるフレームワーク通知には、そのインデックスを指す関係タイプ `index` とのリンクが含まれ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は、インデックスに向けてナビゲートし、そこでさらなる変更を検出することができるようになります。たとえば、新しい「Resource Lists」の作成に関するフレームワーク通知には、「Resource List Index」を指す `index` リンクが含まれていなければなりません。

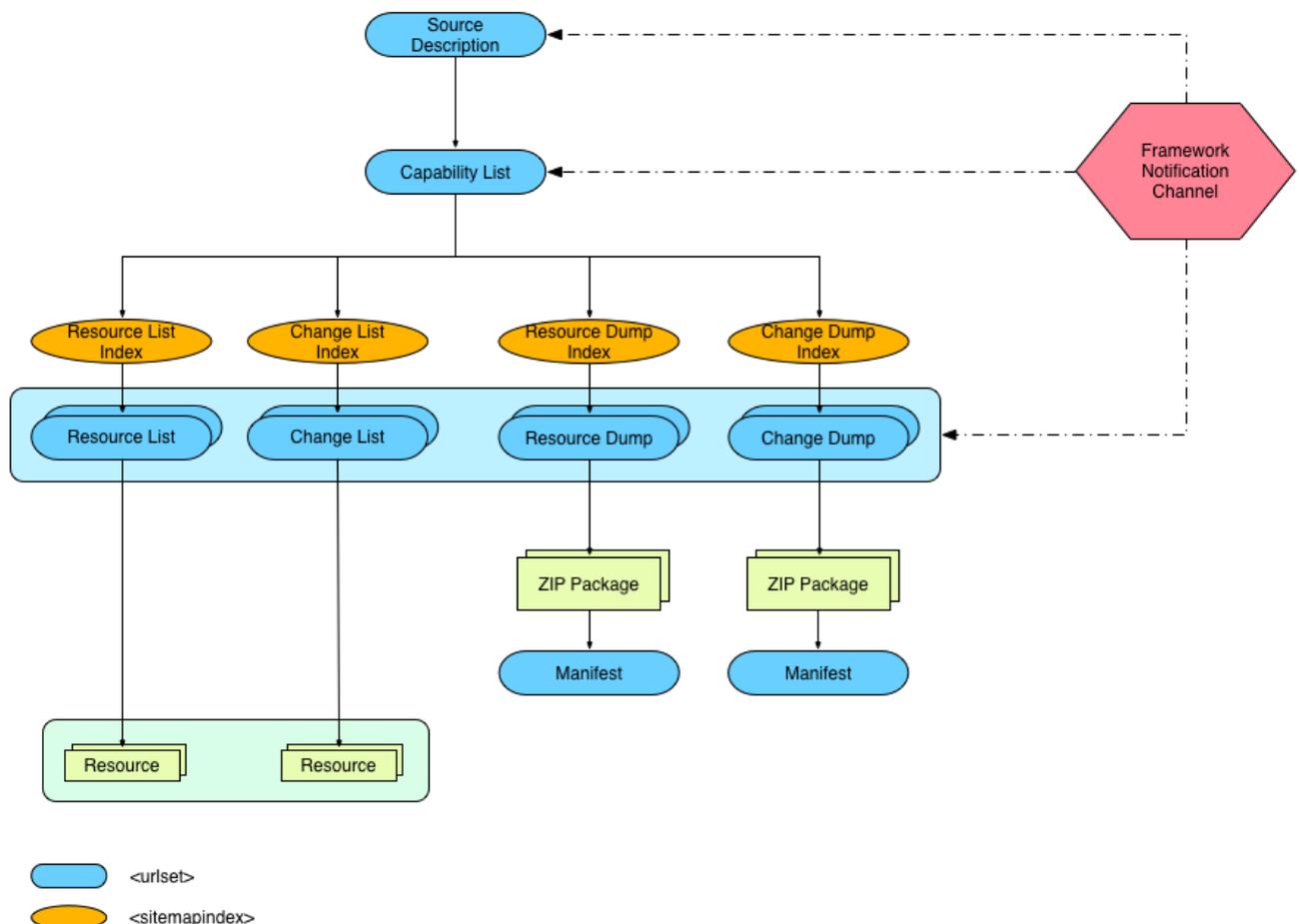


図1 : ResourceSync フレームワーク構造における「Framework Notification」チャンネル

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。

「Framework Notification」がサポートされている個々のリソースのセットごとに、専用の「Framework Notification」チャンネルを提供しなければなりません。つまり、リソースのセットには独自の「Framework Notification」チャンネルがあり、これを介して機能ドキュメントへの変更と、リソースのセットに関連付けられた「Capability List」への変更に関する通知が送信されます。ただし、

「Source Description」への変更に関する通知（例、新しい「Capability List」が作成された場合）は、すべての「Framework Notification」チャンネルを介して送信されます。このようにして、ソースは、デスティネーションが、そのサブスクライブする「Framework Notification」チャンネルに関係なく、ソースの ResourceSync 実装の全体的な組織に対する変更についての情報を確実に得られるようにすることができます。

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「Framework Notification」チャンネルが存在することが示されています。つまり、Framework Notification Channel 2 は、Capability List 2 によって公表された機能ドキュメントへの変更、および、Capability List 2 自体への変更に関する通知を送信するために使用されます。「Source Description」の変更に関する通知は、Framework Notification Channel 1 および Framework Notification Channel 2 を介して送信されます。

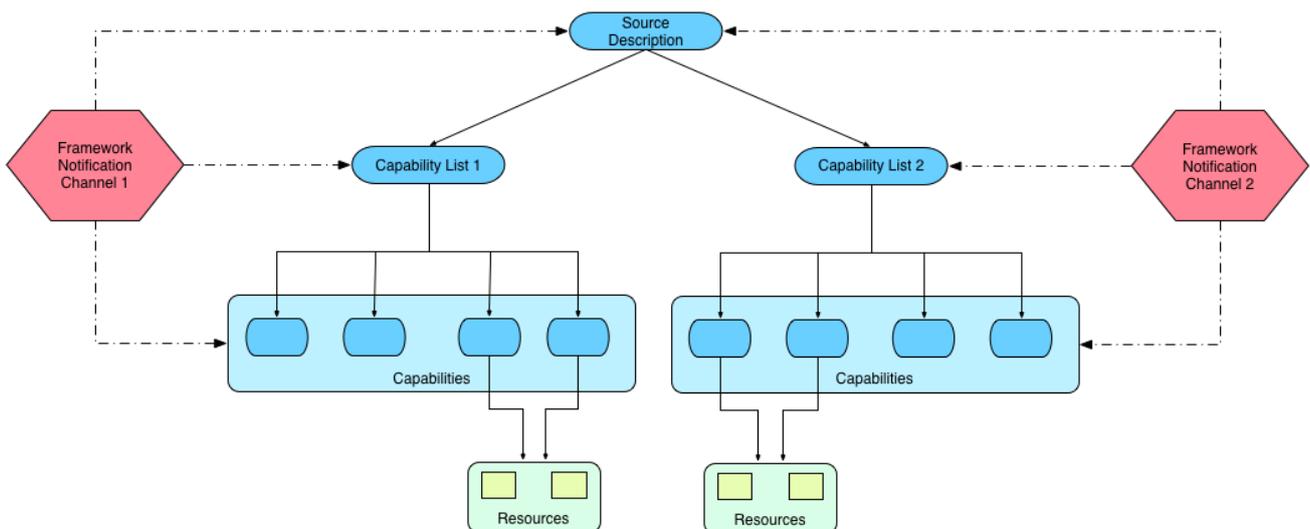


図2：複数のリソースのセットの「Framework Notification」チャンネル

2.1. 通知変更タイプ

次の表は、「Framework Notifications」が ResourceSync フレームワーク内で通知することが可能な変更タイプの概要を示しています。

機能	変更タイプ		
	作成	更新	削除
Framework Notification (フレームワーク通知)			
Resource List (リソース・リスト)	X	X	
Resource Dump (リソース・ダンプ)	X		
Change List (変更リスト)	X	X	
Change Dump (変更ダンプ)	X	X	
Capability List (機能リスト)	X	X	X
Source Description (ソースの記述)	X	X	X

表 2: 「Framework Notification」の変更タイプ

「Framework Notification」チャンネルの作成と削除については、更新された「Capability Lists」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。

3. Framework Notification

フレームワーク通知 (framework notification) は、[2 項](#)で説明したように、ソースが「Resource Lists」、「Change Lists」、「Resource Dumps」、「Change Dumps」、「Capability Lists」および「Source Descriptions」に対する変更についてデスティネーションに通知したい場合に、適切な「Framework Notification」チャンネルを介して送信されます。「Framework 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、これらのドキュメントを定期的にポーリングして変更されたかどうかを判断することができます。

フレームワーク通知の形式は、[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。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。これはルート要素である<urlset>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、`framework-notification`の値を持つ`capability`属性がなければなりません。また、任意の`from`属性および`until`属性も持ちます。`from`属性は、フレームワーク通知が、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、フレームワーク通知が、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。
- 関係タイプ`up`を持つ、`<urlset>`の子要素である任意の`<rs:ln>`
 - フレームワーク通知が「Resource List」などの機能ドキュメントに関連している場合、リンクのターゲットは「Capability List」とします。
 - フレームワーク通知が「Capability List」に関連している場合、リンクのターゲットは「Source Description」とします。
- フレームワーク通知ごとに、`<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、フレームワークへの変更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、変更された機能ドキュメント、「Capability List」または「Source Description」のURIを提供します。
 - 必須の子要素`<rs:md>`は、2つの属性を持たなければならないが、また、3つの属性を持つ場合もあります。必須の`change`属性は、変更の性質を伝えるために使用されます。可能な値は`created`、`updated`または`deleted`であり、その使用法は表2に示すとおりです。必須の`capability`属性は、変更が加えられたフレームワークのコンポーネントを示すために使用されます。可能な値は、`changelist`、`resourcelist`、`changedump`、`resourcedump`、`capabilitylist` および `description` です。アーカイブ機能の変更に関する通知には、[アーカイブ仕様 \(archiving specification\)](#) で定義されている `capability` 属性の値が使用されます。追加機能については、他の仕様で追加の値を定義する可能性があります。3番目の任意の`datetime`属性は、[コア仕様の7項](#)で説明されているように、変更の日時を伝えます。
 - 変更されたドキュメントがインデックス・ドキュメントの下に存在する場合には、子要素`<rs:ln>`と共に、インデックス・ドキュメントを指す関係タイプ`index`を提供しなければなりません。

フレームワーク通知では、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。

例 1 は、新しい「Resource List」が利用可能であることをデスティネーションに通知するフレームワーク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="framework-notification"/>
  <url>
    <loc>http://example.com/resourceset1/resourcelist.xml</loc>
    <rs:md change="created" datetime="2013-01-03T00:07:22Z"
      capability="resourcelist"/>
  </url>
</urlset>
```

例 1：新しい「Resource List」が利用可能であることを通知するフレームワーク通知のペイロード

図 1 および **図 2** に示すように、フレームワーク通知はインデックスレベルで送信されることはありません。ソースが、インデックスの下に存在するドキュメント（例、「Resource List」）への変更に関するフレームワーク通知を送信する場合には、その変更が伝達される<url>要素に子要素<rs:ln>を提供しなければなりません。そのリンクの関係タイプは `index` でなければならず、また、リンクのターゲットは、変更されたドキュメントが存在するインデックス（例、「Resource List Index」）でなければなりません。

フレームワーク通知には、フレームワークへの単一の変更に関する情報のみが含まれている可能性が高いといえます。ただし、複数の変更を1つのフレームワーク通知に集約することも可能です。**例 2** は、新しい「Resource List」、新しい「Resource Dump」について、および、更新された「Change List」について、デスティネーションに通知するフレームワーク通知のペイロードを示しています。「Resource List」はインデックスの下に存在するため、対応する<url>要素は、関係タイプ `index` を持つ子要素<rs:ln>を持ちます。フレームワーク通知には、インデックスが他の新しい「Resource Lists」を指している可能性がある場合でも、インデックスの下に存在する1つの新しい「Resource List」について、1つのエントリしか含まれないことに注意してください。

```
<?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:md capability="framework-notification"/>
  <url>
    <loc>http://example.com/dataset1/resourcelist.xml</loc>
    <rs:md change="created"
      capability="resourcelist"/>
    <rs:ln rel="index"
      href="http://example.com/dataset1/resourcelist-index.xml"/>
  </url>
  <url>
    <loc>http://example.com/dataset1/resourcedump.xml</loc>
    <rs:md change="created"
      capability="resourcedump"/>
  </url>
  <url>
    <loc>http://example.com/dataset1/changelist.xml</loc>
    <rs:md change="updated"
      capability="changelist"/>
  </url>
</urlset>
```

例 2 : 複数のフレームワークの変更について通知するフレームワーク通知

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「Framework Notifications」での WebSub の使用法の説明は、「[Change 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 3](#) 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 3 : 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。

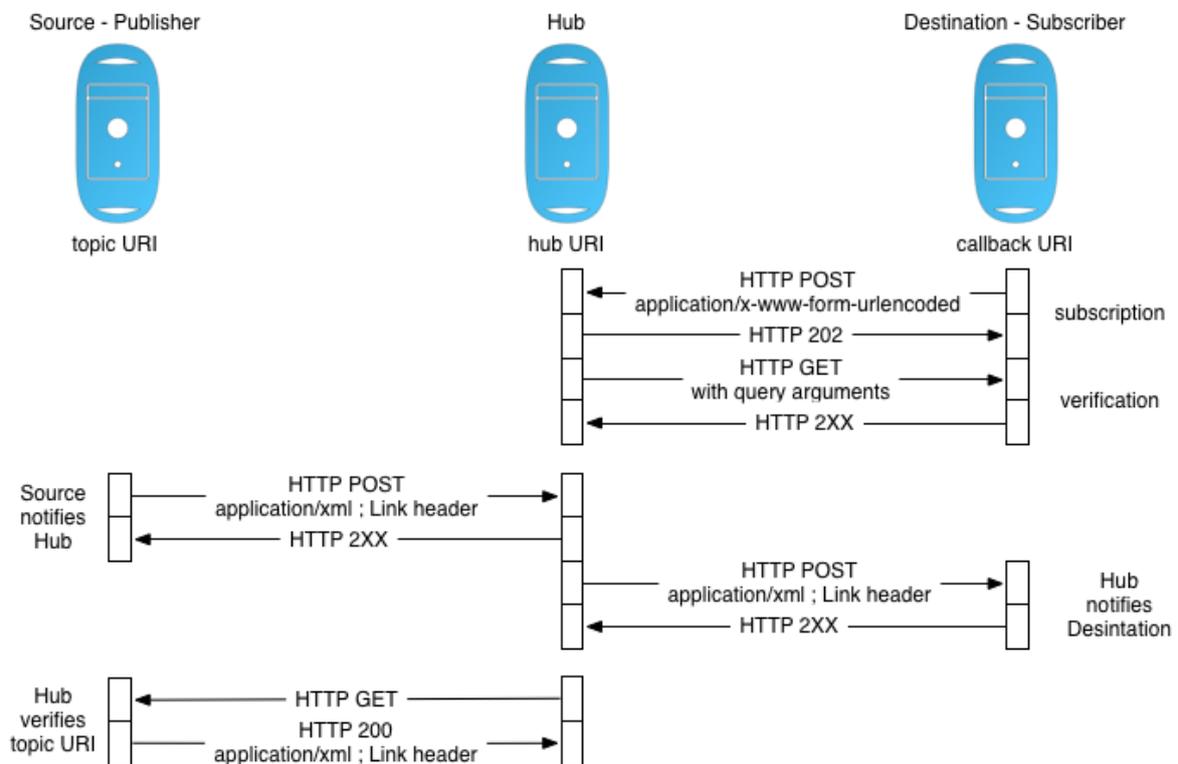


図 3 : ソース、ハブおよびデスティネーション間の HTTP 相互作用

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 のフレームワーク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、フレームワーク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関係タイプ `self` 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関係タイプ `hub` 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、フレームワーク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、最新のフレームワーク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、フレームワーク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関係タイプ `self` を持つ必須リンク。
 - ハブ URI を `href` 属性の値として提供する関係タイプ `hub` 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、フレームワーク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 3](#) は、[例 1](#) のフレームワーク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 に関連するフレームワーク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/framework/> です。

```
POST /websub/ HTTP/1.1
Host: http://hub.example.org
Content-Type: application/xml
Link: <http://example.com/dataset1/framework/> ; 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 3 : ソースがハブにフレームワーク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 4 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` に関連するフレームワーク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/framework/> 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: 144

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

例 4 : デスティネーションがソースのフレームワーク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 5 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%2Fexample.com%2Fdataset1%2Fframework%2F&hub.challenge=c0cc4630-5116-11e3-8f96-0800200c9a66&hub.lease_seconds=2400
HTTP/1.1
Host: http://destination.example.net
Connection: Close
```

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

例 6 は、**例 5** 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 6：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、ソースからフレームワーク通知を受け取ると、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関係タイプ `self` 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関係タイプ `hub` 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 7](#) 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 3](#) 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/framework/> ; rel="self",
      <span<http: hub.example.org="" websub="" >="" ;="" rel="hub" ,<="" span="">
      <http://www.example.com/dataset1/capabilitylist.xml> ; rel="resourcesync"
Content-Length: 849

<?xml version="1.0" encoding="UTF-8"?>
<urlset ...
</span<http:>
```

例 7 : ハブがソースのフレームワーク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 `subscribe` の代わりに `unsubscribe` を使用します。

5. Framework Notification チャンネルの公表: WebSub

「Framework 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「Framework 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「Framework Notification」チャンネルを示しています。この図では、専用の「Framework 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、「Framework 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。

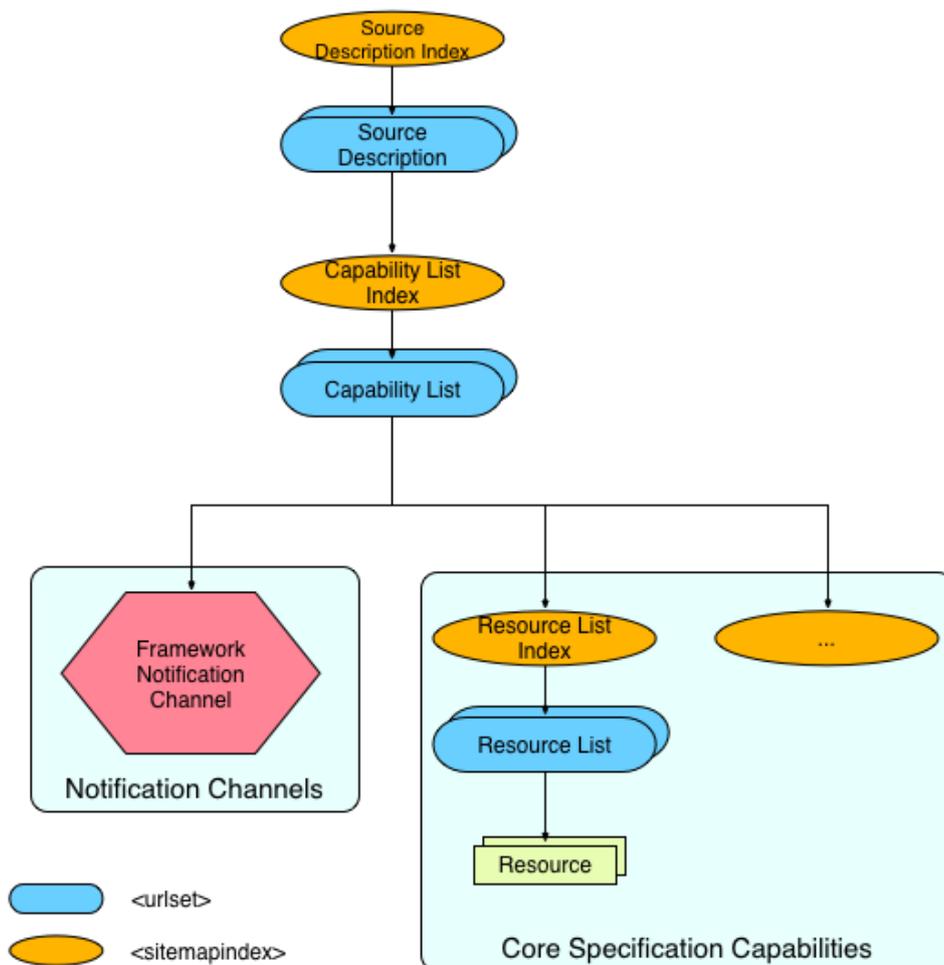


図 4 : 「Framework Notification」チャンネルの検出

例 8 は、[コア仕様の例 13](#) から、「Framework 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。「Framework Notification」機能を示すために、capability 属性に framework-notification 値が導入されていることに注意してください。

http://www.openarchives.org/rs/notification/1.0.1/framework_notification

```
<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/framework/</loc>
    <rs:ln rel="hub" href="http://hub.example.org/websub/" />
    <rs:md capability="framework-notification"/>
  </url>
</urlset>
```

例 8 : WebSub のフレームワーク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/webach/>

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

Date	Editor	Description
2017-07-20	simeon, martin, herbert	PubSubHubbub の代わりに WebSub を使用するよう更新
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。個別の Framework Notification 仕様の作成、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。