



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



ResourceSync フレームワーク仕様 (ANSI / NISO Z39.99-2017)

2017 年 2 月 2 日

本バージョン:

<http://www.openarchives.org/rs/1.1/resourcesync>

最新バージョン:

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

旧バージョン:

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

概要

本 ResourceSync 仕様では、第三者システムが進化するサーバーのリソースとの同期を保つことを可能にする、さまざまな機能で構成される Web のための同期フレームワークについて説明しています。これらの機能は、ローカルまたはコミュニティの要件を満たすためにモジュール方式で組み合わせることができます。また、本仕様では、サーバーがサポートする同期機能を公表する方法と、第三者システムがこの情報を見つける方法についても説明しています。本仕様では、サイトマップ・プロトコル (Sitemap protocol) で定義されたドキュメント形式を再利用し、それらに対する拡張機能を取り入れています。

本ドキュメントのステータス

本ドキュメントは、米国情報保存検索センターが開発した米国標準規格 (ANS) [ANSI/NISO Z39.99-2017](#) の HTML バージョンです。これは米国規格協会 (ANSI) によって 2017 年 2 月 2 日に承認されました。本 PDF 版の [前付け](#)、[作成者および謝辞](#) は、付録 C に記載されています。

本仕様は『[ResourceSync Framework Specification](#) (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。

目次

1. [はじめに](#)
 - 1.1 [目的と範囲](#)
 - 1.2 [動機付けの例](#)
 - 1.3 [ウォークスルー](#)
2. [規範的な参考文献](#)
3. [定義](#)

4. [名前空間接頭辞バインディング](#)
5. [同期処理](#)
 - 5.1 [ソースの観点](#)
 - 5.2 [デスティネーションの観点](#)
 - 5.3 [まとめ](#)
6. [フレームワークの構成](#)
 - 6.1 [構造](#)
 - 6.2 [ナビゲーション](#)
 - 6.3 [検出](#)
 - 6.3.1 [概要](#)
 - 6.3.2 [ResourceSync の well-known URI](#)
 - 6.3.3 [リンク](#)
 - 6.3.4 [robots.txt](#)
7. [サイトマップ・ドキュメントの形式](#)
8. [ソースの記述](#)
9. [公表機能](#)
10. [リソースの記述](#)
 - 10.1 [Resource List](#)
 - 10.2 [Resource List Index](#)
11. [リソースのパッケージ化](#)
 - 11.1 [Resource Dump](#)
 - 11.1.1 [Resource Dump Manifest](#)
12. [変更の記述](#)
 - 12.1 [Change List](#)
 - 12.2 [Change List Index](#)
13. [変更のパッケージ化](#)
 - 13.1 [Change Dump](#)
 - 13.1.1 [Change Dump Manifest](#)
14. [関連リソースへのリンク](#)
 - 14.1 [概要](#)
 - 14.2 [ミラーリングされたコンテンツ](#)
 - 14.3 [代替表現](#)
 - 14.4 [コンテンツへのパッチの適用](#)
 - 14.5 [リソースとリソースに関するメタデータ](#)
 - 14.6 [リソースの旧バージョン](#)
 - 14.7 [コレクション・メンバーシップ](#)
 - 14.8 [リソースの再公開](#)
- 付録 A. (規範的) [時間属性の要件](#)
- 付録 B. [参考文献一覧](#)
- 付録 C. [前付け、作成者および謝辞](#)
- 付録 D. [変更ログ](#)

1. はじめに

1.1 目的と範囲

Web は非常に動的で、リソースは継続的に作成、更新、削除されています。そのため、リモートサーバーのリソースを使用するには、そのコンテンツの変化に対応していかなければならないという課題が伴います。多くの場合には、サーバーの進化するコンテンツを完全に反映する必要はなく、更新メカニズムとしては、クローリング (crawling) などの十分に確立されたリソース検出手法で十分です。ただし、リモートサーバーのコンテンツの変化の反映については、低遅延と高精度を必要とする重要なユースケースがあります。これまで、これらの要件については、共同システムの小さなグループ内で実装されたアドホックな技術的アプローチが取られてきました。今のところ、Web ベースで広く採用されているアプローチはありません。

本 **ResourceSync** 仕様では、サーバーが、リモートシステムが進化するリソースとより緊密に調和できるようにするためにサポートする可能性のある、さまざまな実装しやすい機能を紹介します。また、サーバーがサポートする機能を公開する方法についても説明します。リモートシステムは、この情報を確認することで、進化するデータとの整合性を保つ最善の方法を判断することができます。

各機能は、変更の性質 (作成、更新または削除) を含む、サーバーの「Resource List (リソース・リスト)」や最近変更されたリソースのリストなどの異なる同期機能を提供します。機能はすべて、[サイトマップ・プロトコル](#) によって取り入れられたドキュメント形式に基づいて実装されます。機能を組み合わせ、さまざまなレベルの機能を実現することで、ローカルまたはコミュニティの異なる要件を満たすことができます。このモジュール性の柔軟性により、ResourceSync は幅広いユースケースに適したものになっています。

1.2 動機付けの例

多くのプロジェクトやサービスが同期化へのニーズを抱えており、アドホックのソリューションが実装されています。ResourceSync は、標準化された同期方法を提供することで、実装の上での労力を減らし、リソースの再利用を簡単にするものです。本項では、異なるニーズや複雑さに関する動機付けの例を説明します。

まず、小さな博物館コレクションの Web サイトのケースを想定してみてください。その Web サイトには、数十個の静的 Web ページしかないかもしれません。メンテナーは、これらの Web ページの「[Resource List](#)」を作成し、ResourceSync を活用するサービスに公開できます。

Linked Data を介してサービスを構築する場合、アクセスと可用性を向上させるために、データのローカルコピーを維持することが望ましいことがよくあります。データセットの「[Resource List](#)」を公開することで、ハーベスティング (harvesting) を利用可能にすることができます。多くの場合、Linked Data として公開されるリソース表現が小さいために、少量のコンテンツに対する多数のラウンドトリップが原因となって、個々の HTTP GET 要求 (request) を介したリソース表現の取得が遅くなります。この点については、ZIP ファイルにパッケージ化・記述されたコンテンツを指す「[Resource Dump](#) (リソース・ダンプ)」を公開することで、クライアントにとってはより効率的に、サーバーにとってはより負担を軽減することができます。継続的な同期化は、最新の「[Resource List](#)」または「[Resource Dump](#)」を自動更新することで、あるいは、(より効率的には) リソースの変更のみに関する情報を提供する「[Change List](#) (変更リスト)」を公開することで可能になります。

学術論文およびデータのリポジトリは、通常、[OAI-PMH](#)を介してメタデータを共有しています。これらのリポジトリはリソース中心または Web 中心になるように再設計されているため、ResourceSync フレームワークを使用することで、メタデータとコンテンツの両方をアグリゲーターや汎用 Web 検索エンジンとも同じように共有できるようになります。[サイトマップ](#)と互換性のある「[Resource List](#)」を公開することで、基本レベルの相互運用性の提供と、汎用 Web 検索エンジンによるインデックスの作成が可能になります。「[Resource Dump](#)」、「[Change List](#)」および「[Change Dump](#) (変更ダンプ)」のような他の機能を使用すると、より簡単な初期同期と、その後のより効率的な更新がサポートされます。ResourceSync は、「[リソースとリソースに関するメタデータ](#) (Resources and Metadata about Resources)」の関係を明確に示しつつ、その両方の同期化をサポートします。また、「[ソースの記述](#) (Source Description)」を用いてさまざまなリソースのセットの表現を可能にし、リポジトリのコンテンツのサブセットの同期を必要とするユースケースをサポートします。

1.3 ウォークスルー

変化し続けるコンテンツを公開するソース (Source) (<http://example.com/>) との同期化を保ちたいと希望する人がいたと想定してみましょう。この作業をデスティネーション (Destinations) にとって簡単にするための最初のステップは、同期に使用できるリソースの URI が記載された「[Resource List](#)」をソースが公開することです。この「Resource List」はサイトマップとして表されます。[例 1](#)に示すように、ソースは各リソースの URI を `<url>` 要素の `<loc>` 子要素の値として伝えます。サイトマップが ResourceSync の「Resource List」機能を実装していることを示す `<urlset>` ルート要素の `<rs:md>` 子要素に注意してください。また、「Resource List」には、`at` 属性で指定された日時 (datetime) のソースのリソースの状態が反映されることも伝達されます。この日時により、デスティネーションは、この特定の「Resource List」を以前に処理したかどうかをすばやく判断することができます。

```
<?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:md capability="resourcelist"
    at="2013-01-03T09:00:00Z"/>
  <url>
    <loc>http://example.com/res1</loc>
  </url>
  <url>
    <loc>http://example.com/res2</loc>
  </url>
</urlset>
```

例 1. 「Resource List」

ソースは、「Resource List」に追加情報を提供することで、デスティネーションがコンテンツを収集し、その正確性を検証するプロセスを最適化することができます。たとえば、ソースがリソースの最新の変更の日時を表すことで、デスティネーションは現在のバージョンを保持しているかどうかを判断できるため、最新の状態を保つために発行する必要がある HTTP 要求の数を最小限に抑えることができます。[例 2](#)では、このサイトマップの任意の `<lastmod>` 要素を使用して伝達される情報が示されています。ソースが特定のビットストリーム (bitstream) のハッシュ (hash) も伝達する場合には、デスティネーションはそれを取得するプロセスが成功したかどうかを検証できます。例

では、この<rs:md>要素のハッシュ属性を使用して伝達される情報が示されています。さらに、ソースは<rs:ln>要素を使用して関連リソースへのリンクを提供できます。例では、リストされている2番目のリソースのミラーコピーへのリンクが示されており、ソースがデスティネーションに、ここからリソースを取得することを望んでいることが示されます。

```
<?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:md capability="resourcelist"
    at="2013-01-03T09:00:00Z"/>
  <url>
    <loc>http://example.com/res1</loc>
    <lastmod>2013-01-02T13:00:00Z</lastmod>
    <rs:md hash="md5:1584abdf8ebdc9802ac0c6a7402c03b6"/>
  </url>
  <url>
    <loc>http://example.com/res2</loc>
    <lastmod>2013-01-02T14:00:00Z</lastmod>
    <rs:md hash="md5:1e0d5cb8ef6ba40c99b14c0237be735e"/>
    <rs:ln rel="duplicate"
      href="http://mirror.example.com/res2"/>
  </url>
</urlset>
```

例 2. 追加情報を含む「Resource List」

変化するコンテンツをよりタイムリーに記述するために、ソースは最新の「Resource List」を公開する頻度を増やすことができます。ただし、変更が頻繁に行われたり、コンテンツ・コレクションのサイズが非常に大きかったりするために、完全な「Resource List」を定期的に更新することが実際的でない場合もあります。そのような場合には、ソースは変更に関する情報のみを伝達する機能を実装できます。このため、ResourceSync では、「Change List」が導入されています。「Change List」は、リソースの変更、変更の性質（作成、更新または削除）、および、リソースの新しい状態を記述するメタデータを列挙します。デスティネーションは、ソースから「Change List」を繰り返し取得し、リストされた変更を確認して、すでに実行された変更を見極め、残りの変更を処理することができます。「Change List」の変更は時系列順で提供されるため、デスティネーションがどの変更をすでに処理したかを判断するのに役立ちます。さらに、「Change List」には、「Change List」に含まれる時区間 (temporal interval) の開始時刻と終了時刻を伝える日時も含まれています。これらの時間は、当該の時区間中に発生したすべてのリソースの変更が「Change List」に記載されていることを伝えています。（ResourceSync は、「Change List」が作成された後、「Change List」をいつまで利用可能にしなければならないかについては指定しません。ソースが「Change List」を長く保持すればするほど、デスティネーションが、たとえばオフラインであったために見逃した変更をキャッチアップする確率は高まります。）

例 3 は「Change List」を示しています。<urlset> の<rs:md> 子要素の capability 属性の値は、ここではサイトマップが「Resource List」ではなく「Change List」であることを明確にしています。from 属性と until 属性は、「Change List」に含まれる時区間を指定しています。以下に示す「Change List」では、3つのリソースの変更が伝えられています（1つは更新、もう1つは削除、3つ目は作成）。変更タイプは、各<rs:md>要素の change 属性の値から確認できます。この例では、リソースの変更時刻を伝える<lastmod>要素の使用と、変更時刻を伝える<rs:md>の datetime 属性の使用も示されています。作成されたリソースの場合には、<lastmod>要素が伝える日時が

datetime 属性の日時よりも前になっており、これはおそらく、リポジトリが指定された datetime にのみ古いファイルを表示するように設定したためです。変更は、更新時間順(datetime 属性の値として指定できる)に時系列にリストされる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="changelist"
    from="2013-01-02T00:00:00Z"
    until="2013-01-03T00:00:00Z"/>
  <url>
    <loc>http://example.com/res1.pdf</loc>
    <lastmod>2013-01-02T13:00:00Z</lastmod>
    <rs:md change="updated" datetime="2013-01-02T13:00:00Z"/>
  </url>
  <url>
    <loc>http://example.com/res2.pdf</loc>
    <rs:md change="deleted" datetime="2013-01-02T14:00:00Z"/>
  </url>
  <url>
    <loc>http://example.com/res3.tiff</loc>
    <lastmod>2011-01-01T00:00:00Z</lastmod>
    <rs:md change="created" datetime="2013-01-02T15:00:00Z"/>
  </url>
</urlset>
```

例 3. 「Change List」

デスティネーションは、「Resource List」にリストされた各リソース URI に対して HTTP GET 要求を発行することができます。大きな「Resource List」の場合には、これらの要求をすべて発行するのは煩雑かもしれません。そのため、ResourceSync では、ソースがパッケージ化されたコンテンツを利用可能にするために使用できる機能を導入します。サイトマップとして実装される「Resource Dump」には、パッケージ化されたコンテンツへのポインタ(pointer)が含まれています。「Resource Dump」で参照される各コンテンツ・パッケージは、ソースのビットストリームと、それぞれを説明する「Resource Dump Manifest(リソース・ダンプ・マニフェスト)」を含む ZIP ファイルとなっています。「Resource Dump Manifest」自体もサイトマップとして実装されます。デスティネーションは、「Resource Dump」を読み出し、含まれるポインタを逆参照することでコンテンツ・パッケージを取得し、読み出したパッケージを解凍できます。「Resource Dump Manifest」には、ソースが各ビットストリームに関連付ける URI もリストされるため、デスティネーションは、「Resource List」にリストされた URI を逆参照してデータを取得するのと同じ結果を得ることができます。[例 4](#) は、単一のコンテンツ・パッケージを指す「Resource Dump」を示しています。このパッケージの URI を逆参照すると、[例 5](#) で示す、「Resource Dump Manifest」を含む ZIP ファイルにたどり着きます。ここでは、ソースの ZIP ファイルに 2 つのビットストリームが含まれることが示されています。

<rs:md>要素の path 属性は、ZIP ファイル内のビットストリームのファイルパス(ZIP が解凍されていた場合には、ビットストリームが存在する相対ファイルシステムパス)を伝えますが、<loc>要素はソースのビットストリームと関連する URI を伝えます。

追加機能である「Change Dump」は、「Resource Dump」に似た機能を提供しますが、特定の瞬間のリソース・ビットストリームのスナップショットをパッケージ化するのではなく、ある時区間の間に変更されたリソースのビットストリームをパッケージ化します。

<http://www.openarchives.org/rs/1.1/resourcesync>

```
<?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="resourcedump"
    at="2013-01-03T09:00:00Z"/>
  <url>
    <loc>http://example.com/resourcedump.zip</loc>
  </url>
</urlset>
```

例 4. 「Resource Dump」

```
<?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="resourcedump-manifest"
    at="2013-01-03T09:00:00Z"/>
  <url>
    <loc>http://example.com/res1</loc>
    <rs:md hash="md5:1584abdf8ebdc9802ac0c6a7402c03b6"
      path="/resources/res1"/>
  </url>
  <url>
    <loc>http://example.com/res2</loc>
    <rs:md hash="md5:1e0d5cb8ef6ba40c99b14c0237be735e"
      path="/resources/res2"/>
  </url>
</urlset>
```

例 5. ZIP ファイルのコンテンツを詳述する「Resource Dump Manifest」

ResourceSync では、ソースが 1 セットのリソースに対してサポートする機能を記述する方法である、「機能リスト (Capability List)」も取り入れています。その記述を例 6 に示します。ここでは、ソースが「Resource List」、「Resource Dump」および「Change List」の各機能をサポートし、それぞれの URI をリストすることが示されます。「機能リスト」に含まれるリソースのセットの記述と `describedby` 関係によってリンクされる `<urlset>` の `<rs:ln>` 子要素が含まれていることに注意してください。これらの機能は、同じ「機能リスト」の中で伝えられるため、このリソースのセットに均一に適用されます。たとえば、特定のリソースが「Resource List」に表示される場合には、「Resource Dump」にも表示され、さらに、リソースへの変更は「Change List」に報告されなければなりません。

```
<?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="describedby"
    href="http://example.com/info_about_set1_of_resources.xml"/>
  <rs:ln rel="up"
    href="http://example.com/resourcesync_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>
</urlset>
```

例 6. ソースがリソースのセットに対してサポートする ResourceSync 機能をリストする「機能リスト」

ソースが ResourceSync をサポートするかどうか、また、どのようにサポートするのかをデスティネーションが検出するには 3 つの方法があります。すなわち、ソース全体のアプローチ、リソース固有のアプローチ、および、サイトマップを検出するための既存の慣行を活用するアプローチです。ソース全体のアプローチは、公認の URI 仕様を活用するもので、例 7 に示すように、「ソースの記述」を作成するソースで構成されます (`/.well-known/resourcesync` で公開)。「ソースの記述」は、ソースが提供する「機能リスト」を列挙し、リソースのセットごとに 1 つの「機能リスト」が存在します。ソースにリソースのセットが 1 つしかないために、「機能リスト」が 1 つしかない場合には、必須となる「ソースの記述」にはポインタが 1 つだけ含まれます。リソース固有の検出のアプローチは、リンクを提供するリソースを含む「機能リスト」を指す HTML ドキュメント、または、HTTP Link ヘッダーのリンクを提供するソースで構成されます。例 6 では、`up` 関係によって「ソースの記述」とリンクする `<urlset>` の `<rs:ln>` 子要素が含まれていることにより、「機能リスト」から「ソースの記述」へのナビゲーションが可能になっていることに注意してください。さらに、もう 1 つのアプローチは、ソースの `robots.txt` ファイルを用いてサイトマップを検出する確立された慣行に従ったものです。「Resource List」はサイトマップであるため、`Sitemap` ディレクティブ (directive) の値として `robots.txt` ファイルにその URI を含めることにより、検出可能にすることができます。「Resource

List」に含まれるナビゲーションの `up` リンクを使用すると、その「Resource List」に含まれるリソース・セットに関連する「機能リスト」を検出でき、さらに、「機能リスト」の中の `up` リンクからは「ソースの記述」にたどり着けます。

```
<?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="describedby"
    href="http://example.com/info-about-source.xml"/>
  <rs:md capability="description"/>
  <url>
    <loc>http://example.com/dataset1/capabilitylist.xml</loc>
    <rs:md capability="capabilitylist"/>
    <rs:ln rel="describedby"
      href="http://example.com/info_about_set1_of_resources.xml"/>
  </url>
</urlset>
```

例 7. ソースによって提供される単一のリソース・セットのための「機能リスト」へのポインタを含む「ソースの記述」

場合によっては、これまでに説明したドキュメントをいくつかに分割する必要があります。たとえば、サイトマップ・プロトコルは、現在、サイトマップごとに最大 50,000 個のリソースを規定しており、ソースには同期化の対象となるリソースがより多く存在する場合があります。ResourceSync フレームワークでは、これらのコミュニティで定義された制限に従っているため、そのような場合には、複数の「Resource List」と、それらの各々を指す「Resource List Index (リソース・リスト・インデックス)」が公開されます。「Resource List Index」は、サイトマップの `<sitemapindex>` ドキュメント形式を用いて表されます。[例 8](#) は、2 つの「Resource List」を指す「Resource List Index」を示しています。

```
<?xml version="1.0" encoding="UTF-8"?>
<sitemapindex xmlns="http://www.sitemaps.org/schemas/sitemap/0.9"
  xmlns:rs="http://www.openarchives.org/rs/terms/">
  <rs:md capability="resourcelist"
    at="2013-01-03T09:00:00Z"/>
  <sitemap>
    <loc>http://example.com/resourcelist-part1.xml</loc>
  </sitemap>
  <sitemap>
    <loc>http://example.com/resourcelist-part2.xml</loc>
  </sitemap>
</sitemapindex>
```

例 8. `<sitemapindex>` ドキュメント形式を使って表される「Resource List Index」

2. 規範的な参考文献

以下のドキュメントには、本標準の実装に必要とされる規定が含まれています。すべての標準は改訂の対象であり、これらの標準の最新バージョンが使用されなければなりません。

[ALE]

Snell, J. [*Atom Link Extensions*](#). Internet Draft. Internet Engineering Task Force (IETF), June 8, 2012. Available at: <http://tools.ietf.org/html/draft-snell-atompub-link-extensions-09>

[IANA MIME]

[*MIME Media Types*](#) [registry website]. Internet Assigned Numbers Authority (IANA). Available at: <http://www.iana.org/assignments/media-types>

[IANA Relation]

[*Link Relations*](#) [registry website]. Internet Assigned Numbers Authority (IANA). Available at: <http://www.iana.org/assignments/link-relations/link-relations.xml>

[RFC 2616]

Fielding, R. *et al.* [*Hypertext Transfer Protocol -- HTTP/1.1*](#). RFC 2616. Internet Engineering Task Force (IETF), June 1999. Available at: <http://www.ietf.org/rfc/rfc2616.txt>

[RFC 4287]

Nottingham, M., and R. Sayre, eds. [*The Atom Syndication Format*](#). RFC 4287. Internet Engineering Task Force (IETF), December 2005. Available at: <http://www.ietf.org/rfc/rfc4287.txt>

[RFC 5988]

Nottingham, M. [*Web Linking*](#). RFC 5988. Internet Engineering Task Force (IETF), October 2010. Available at: <http://www.ietf.org/rfc/rfc5988.txt>

[RFC 6249]

Bryan, A. *et al.* [*Metalink/HTTP: Mirrors and Hashes*](#). RFC 6249. Internet Engineering Task Force (IETF), June 2011. Available at: <http://www.ietf.org/rfc/rfc6249.txt>

[RFC 6906]

Wilde, E. [*The 'profile' Link Relation Type*](#). RFC 6906. Internet Engineering Task Force (IETF), March 2013. Available at: <http://www.ietf.org/rfc/rfc6906.txt>

[RFC 7089]

Van de Sompel, H., M. Nelson, and R. Sanderson. [*HTTP Framework for Time-Based Access to Resource States -- Memento*](#). RFC 7089. Internet Engineering Task Force (IETF), December 2013. Available at: <http://www.ietf.org/rfc/rfc7089.txt>

[Sitemap]

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

[W3C Datetime]

Wolf, Misha, and Charles Wicksteed. [*Date and Time Formats*](#). W3C Note. World Wide Web Consortium, August 27, 1998. Available at: <http://www.w3.org/TR/1998/NOTE-datetime-19980827>

[ZIP]

[*ZIP File Format Specification*](#). Application Note. Version 6.3.3. PKWARE Inc., September 1, 2012. Available at: <http://www.pkware.com/documents/casestudies/APPNOTE.TXT>

3. 定義

本標準では、以下の用語を以下に示す意味で使用します。

用語	定義
ソース (Source)	同期の対象となるリソースをホストするサーバー
デスティネーション (Destination)	ソースのリソースと自身を同期させるシステム
リソースのセット (set of resources)	ソースが同期化に使用できるようにしたリソースのコレクション。ソースは、このようなコレクションを 1 つまたはそれ以上公開し、各コレクションに対して個別の ResourceSync 機能をサポートできる。個々のリソースは、複数のリソース・セットに含まれる場合もある。

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。

4. 名前空間接頭辞バインディング

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
	http://www.sitemaps.org/schemas/sitemap/0.9	サイトマップ・プロトコル 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の名前空間

5. 同期処理

[1.3 項](#) では、ソースが実装できる機能の具体的なワークスルーを提供し、デスティネーションがこれらの機能を使用してソースの変化するデータとの同期を保つ方法について説明しています。本項では、さまざまな ResourceSync 機能について高レベルの概要を提供し、これらの変更に合わせて設計されたデスティネーションの処理にどのように適合するかを示します。

5.1 ソースの観点

デスティネーションが変化するデータとの同期を保つための処理が行えるようにサポートすることのできる ResourceSync 機能は、ソースの観点から、以下のように要約されます。

- **コンテンツの記述 (Describing Content)** : データを記述するために、ソースは最新の「Resource List」を維持することができます。基本的な「Resource List」では、ソースが同期のために提供するリソースの URI を最小限提供します。ただし、リソースの最新の変更時刻や、コンテンツベースのチェックサム (checksum) またはハッシュや長さ (length) など

の固定情報など、ソースのリソースを取得するデスティネーションの処理を最適化するための追加情報を、「Resource List」に追加しても構いません。[図 1](#)は、ソースが時刻 t2 および t4 で最新の「Resource List」を公開した様子を示しています。t4 では、1 つの「Resource List」に収まるようにリストする必要のあるリソースが多すぎるため、複数の「Resource List」が公開され、「Resource List Index」にグループ化されています。

- **コンテンツのパッケージ化(Packaging Content)**: データをダウンロード可能にするために、ソースは自身のコンテンツの最新の「Resource Dump」を繰り返し利用可能にすることができます。「Resource Dump」は、1 つまたはそれ以上のパッケージを指しており、各パッケージには、ソースがホストするリソースに関連付けられたビットストリームが含まれます。各パッケージには、そこに含まれるビットストリームに関するメタデータを提供する「Resource Dump Manifest」も含まれ、その ZIP ファイルには、関連付けられた URI とファイルパスが最小限含まれていなければなりません。[図 1](#)は、ソースが時刻 t1 および t3 で最新の「Resource Dump」を公開した様子を示しています。時刻 t3 では、複数の「Resource Dump」が発行され、「Resource Dump Index(リソース・ダンプ・インデックス)」にグループ化されています。
- **変更の記述(Describing Changes)**: 同期の待ち時間を短縮し、および/または、転送効率を改善するために、ソースはリソースの変更に関する情報を提供する「Change List」を公開することができます。「Change List」に含まれる時区間は、たとえば、過去 1 時間、当日、または、最新の「Resource List」の発行以降に発生したすべての変更など、ソースが決定します。「Change List」は、リソースの変更ごとに、変更されたリソースの URI と変更の性質(作成、更新、削除)を最小限伝えなければなりません。「Change List」は変更に基づいて編成されるため、変更ごとに同じリソースを複数回リストする場合があります。[図 2](#)は、3 つの「Change List」を示しています。最初の「Change List」は、t1 と t3 の間、2 番目は t3 と t5 の間、3 番目は t5 と t7 の間で発生したリソースの変更を網羅しています。t5 と t7 の間に発生した変更が多すぎて単一の「Change List」に収まらないため、複数の「Change List」が発行され、「Change List Index(変更リスト・インデックス)」にグループ化されています。
- **変更のパッケージ化(Packaging Changes)**: コンテンツの変更をダウンロード可能にするために、ソースは「Change Dump」を公開することができます。「Change Dump」は、1 つまたはそれ以上のパッケージを指し、各パッケージには、変更後のリソースの状態に対応するビットストリームが含まれます。また、各パッケージには、「Change Dump」で提供されるビットストリームに関するメタデータを提供する、「Change Dump Manifest(変更ダンプ・マニフェスト)」も含まれています。各ビットストリームについて、「Change Dump Manifest」には、関連する URI、変更の性質(作成、更新、削除)、および、必要に応じて ZIP ファイル内のビットストリームのファイルパスを最小限含まなければなりません。「Change Dump」に含まれる時区間は、たとえば、過去 1 時間、当日、または、最新の「Resource Dump」の発行以降に発生したすべてのリソース変更など、ソースが決定します。「Change Dump」は変更に基づいて編成されるため、「Change Dump」が指すパッケージには、変更ごとに 1 つずつ、特定のリソースに関連付けられた複数のビットストリームが含まれる場合があります。[図 2](#)は、3 つの「Change Dump」を示しています。最初の「Change Dump」は、t2 と t4 の間、2 番目は t4 と t6 の間、3 番目は t6 と t8 の間で発生したリソースの変更を網羅しています。t6 と t8 の間には、複数の「Change Dump」が発行され、「Change Dump Index(変更ダンプ・インデックス)」にグループ化されています。

- **関連リソースへのリンク(Linking to Related Resources)** : 関連リソースへの同期の対象となるリソースから追加のリンクを提供する理由はいくつかあります。
 - **代替コンテンツ転送(Alternate Content Transfer)** : デスティネーションがリソースのコンテンツを取得するデフォルトのメカニズムは、「Resource List」または「Change List」にある URI に対して HTTP GET を発行する、あるいは、「Resource Dump」または「Change Dump」を介して取得したパッケージを解凍する、の 2 つです。ただし、追加のアプローチがサポートされる場合もあります。たとえば、ソースは、同期化の目的に対して、ミラーサーバーから、つまり別の URI から、コンテンツが取得されることを望む場合があります。また、ソースでは、変更されたリソース全体ではなく、リソースが受けた変更のみの取得を許可する場合があります。これは、リソースのサイズがかなり大きい場合、および／または、変更の頻度が高い場合に望ましいといえます。このような「代替コンテンツ転送」アプローチでは、リソースから別のリソースへのリンクを使って表現することで、コンテンツを代替方法で利用できるようにします。特定のデスティネーションが特定の「代替コンテンツ転送」アプローチを認識しない可能性があります、その場合には、リンクを無視してリソースの URI を逆参照する予備のアプローチを利用します。
 - **リソースとリソースに関するメタデータ** : リソースとリソースに関するメタデータの両方を同期しなければならないケースがあります。たとえば、科学出版物のコレクションと、それぞれを説明するメタデータなどがあげられます。ResourceSync の観点からすれば、リソースとそれに関するメタデータは、両方とも、同期の対象となる個別の URI を持つリソースと見なされます。それらの相互関係は、適切な関係タイプ (relation type) を持つリンクによって表現されます。
 - **リソースの旧バージョン(Prior Versions of Resources)** : 場合によって、デスティネーションは最新バージョンのみでなく、リソースの各バージョンのコピーを必要とすることがあります。ソースでは、リンクを介して、リソースの以前のバージョンの検出とアクセスをサポートすることができます。これには、リソースのバージョンへのリンクに基づくアプローチと、リソースの状態への時間ベースのアクセスに [Memento](#) プロトコルの機能を活用する 2 つのアプローチの 3 つが提供されています。

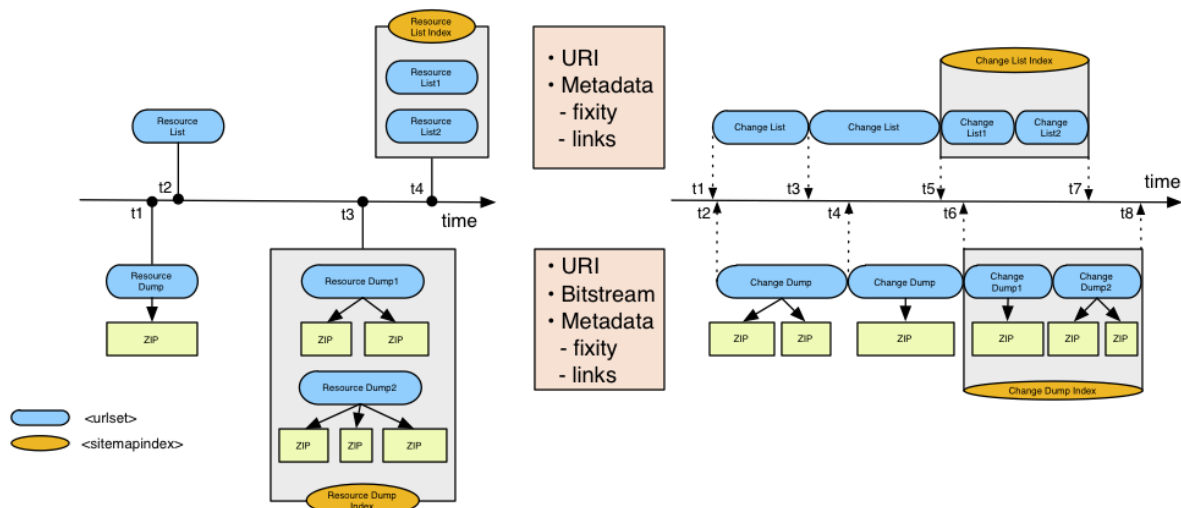


図 1. ソースの観点からみた ResourceSync のリソースの記述

図 2. ソースの観点からみた ResourceSync の変更の記述

5.2 デスティネーションの観点

デスティネーションの観点からは、ResourceSync の機能により、3 つの主要な処理が可能になります。図 3 は、その概要を示しています。

- ベースライン同期 (Baseline Synchronization)**: ソースと同期するためには、デスティネーションはソースのデータの初期コピーを作成しなければなりません。デスティネーションは、ソースのリソースの URI を伝える「Resource List」を取得し、その後、それらの URI を 1 つずつ逆参照することができます。デスティネーションは、また、ソースのリソースに関連付けられたビットストリームをそれぞれ含む 1 つまたはそれ以上のコンテンツ・パッケージの URI を伝える「Resource Dump」を取得することもできます。デスティネーションは、これらの URI を逆参照し、その後、そこに含まれた「Resource Dump Manifest」に基づいて、取得したコンテンツ・パッケージを解凍することができます。
- 増分同期 (Incremental Synchronization)**: デスティネーションは、ベースライン同期を繰り返し実行することにより、ソースと同期した状態を保つことができます。効率を高め、遅延を減らすために、ソースは「Change List」を介してリソースの変更に関する情報を伝達することができます。これにより、デスティネーションが、「Change List」にリストされた新しく作成および更新されたリソースの URI を逆参照して、最新のコンテンツを取得できるようにします。また、必要に応じて、デスティネーションが削除されたリソースのコピーを削除できるようにもします。また、ソースは、変更後のリソースの状態に対応するビットストリームをそれぞれ含む 1 つまたはそれ以上のパッケージを指す「Change Dump」を利用可能にすることもできます。その場合、デスティネーションは、最初に「Change Dump」を取得し、次に、「Change Dump」にリストされた URI を逆参照することによってパッケージを取得し、その後、そこに含まれた「Change Dump Manifest」に基づいて、それらを解凍します。
- 監査 (Audit)**: ソースと同期しているかどうかを検証するために、デスティネーションでは、取得したコンテンツが、範囲と精度の両面から、ソースがホストする現在のリソースと一致していることを確認できなければなりません。これにはソースがホストするリソースについて

の最新のリストが必要ですが、これは、「Resource List」および「Change List」に基づいてコンパイルすることができます。また、これらのリストには、最終更新時刻、長さ、コンテンツベースのハッシュなど、最新の状態を特徴付けるメタデータをリソースごとに含める必要があります。

	Baseline Synchronization	Incremental Synchronization	Audit
<ul style="list-style-type: none">• URI• Metadata<ul style="list-style-type: none">- fixity- links	Resource List	Change List	<ul style="list-style-type: none">Resource List<ul style="list-style-type: none">fixityChange List<ul style="list-style-type: none">fixity
<ul style="list-style-type: none">• URI• Bitstream• Metadata<ul style="list-style-type: none">- fixity- links	Resource Dump	Change Dump	<ul style="list-style-type: none">Resource Dump<ul style="list-style-type: none">fixityChange Dump<ul style="list-style-type: none">fixity

図 3. デスティネーションの観点からみた ResourceSync

5.3 まとめ

表 1 に 5 項の要約を示します。表には、デスティネーションの処理を列に、ソースの機能を行にリストし、セルには特定の処理に対する各機能の適用性を示します。

ソースの機能	デスティネーションの処理		
	ベースライン同期	増分同期	監査
ソースの記述	X	X	X
公表機能	X	X	X
リソースの説明			
Resource List	X		X
リソースのパッケージ化			
Resource Dump	X		
変更の記述			
Change List		X	X
変更のパッケージ化			
Change Dump		X	
関連リソースへのリンク			
ミラーリングされたコンテンツ	X	X	X
代替表現	X	X	X
コンテンツへのパッチの適用		X	X
リソースとリソースに関するメタデータ	X	X	X
リソースの旧バージョン		X	X
コレクション・メンバーシップ	X	X	X
リソースの再公開	X	X	X

表 1. ソースの機能とデスティネーションの処理

6. フレームワークの構成

6.1 構造

ResourceSync フレームワークのすべての機能は、<urlset>および<sitemapindex>サイトマップドキュメント形式に基づいて実装されます。使用されるドキュメントセットの全体的な構造を [図 4. ResourceSync フレームワーク構造](#) に示します。

- 図の上部には必須となる「ソースの記述」があります。デスティネーションの典型的なエントリーポイントは、ソースの ResourceSync 実装について確認することです。「ソースの記述」は、ソースが提供するすべての「機能リスト」を列挙しており、リソースのセットごとに 1

つの「機能リスト」が存在します。ソースが 1 セットのリソースのみを提供する場合には、「ResourceSync の記述 (ResourceSync Description)」は単一のポインタを含みます。「ソースの記述」は、`<urlset>`ドキュメントとして表され、「機能リスト」ごとに`<url>`要素が導入されます。`<url>`の`<loc>`子要素には「機能リスト」の URI が含まれ、`<url>`の`<rs:md>`子要素の `capability` 属性のための `capabilitylist` 値は、URI が「機能リスト」のそれであることを明確にしています。

- 「機能リスト」は、ソースのリソース・セットがサポートするすべての機能を列挙します。本 ResourceSync 仕様で定義する機能は、「Resource List」、「Change List」、「Resource Dump」および「Change Dump」になります。追加の機能が他の仕様で定義されることもあります。「機能リスト」は`<urlset>`ドキュメントとして表され、サポートされる機能ごとに`<url>`要素が導入されます。`<url>`の`<loc>`子要素には、機能を実装するドキュメントの URI が含まれ、機能のタイプは、`<url>`の`<rs:md>`子要素の `capability` 属性の値によって表されず (例. 「Resource List」のための `resourcelist`)。
- 「Resource List」と「Change List」は、リソースを指します。リソースの表現は、当該リソースの`<url>`要素の`<loc>`子要素の値としてリストされている URI を逆参照することで取得できます。
- 「Resource Dump」と「Change Dump」は、それぞれ、リソースに関連付けられたビットストリームと、パッケージで提供されるビットストリームを記述する「マニフェスト」を含むパッケージを指します。
- ビットストリームのパッケージに含まれるマニフェストは、`<urlset>`ドキュメントとして表されます。パッケージに含まれるビットストリームごとに、そのドキュメントには`<url>`要素が含まれます。つまり、`<url>`の`<loc>`子要素がビットストリームに対応する URI を提供しているのに対し、`<url>`の`<rs:md>`子要素の `path` 属性はパッケージ内のビットストリームのパスを提供します。
- 単一のドキュメントで「ソースの記述」、「Resource List」、「Change List」などを表現するのに十分な場合には、`<urlset>`ドキュメント形式が使用されます。複数のドキュメントが必要とされる場合には、各ドキュメントは`<urlset>`ドキュメント形式を使用して表され、すべての個々の`<urlset>`ドキュメントを指すインデックスとして`<sitemapindex>`ドキュメントが導入されます。その結果、たとえば、「機能リスト」で提供される「Resource List」の URI は、`<urlset>`または`<sitemapindex>`ドキュメントの URI のいずれかになります。特定の機能 (例. 「Resource List」) に利用される`<urlset>`または`<sitemapindex>`ドキュメントは、`capability` 属性 (例. `resourcelist`) と同じ値を持ちます。

図 4 の「Resource List」のブランチ (branch) は、既存のサイトマップ仕様と完全に互換性がありますが、他のブランチは、サイトマップ・ドキュメント形式を活用するリソース同期をサポートするために導入された拡張機能です。

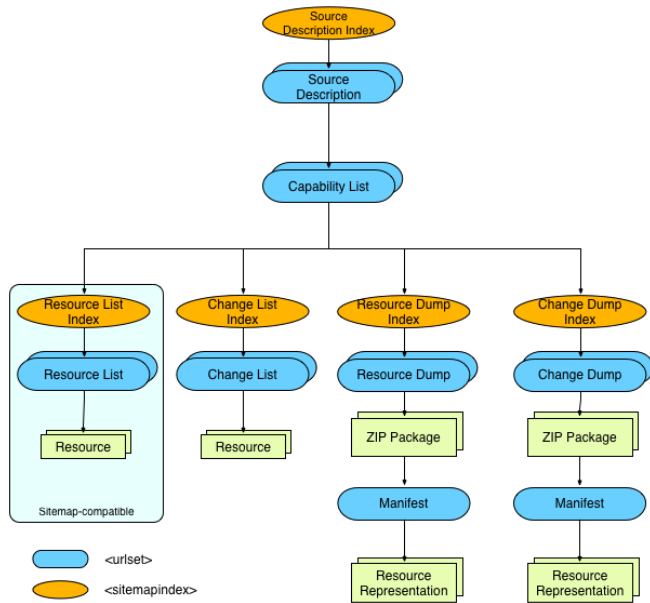


図 4. ResourceSync フレームワークの構造

6.2 ナビゲーション

6.1 項で説明したドキュメント階層のナビゲートをサポートするために、以下のメカニズムを紹介します。これらは図 5 および図 6 で図説します。

- 上方ナビゲーションのリンクは、<urlset>の<rs:ln>子要素、または、ドキュメントの<sitemapindex>要素によって提供されます。このポイントは、rel 属性の値として up を持っており、階層のより上位にあるドキュメントの URI は href 属性の値として提供されます。以下の連続した up リンクをたどると、最終的に「ソースの記述」が表示されます。
- ドキュメントからドキュメントが存在するインデックスへのナビゲーション用のリンクは、それが存在する場合には、ドキュメントの<urlset>要素の<rs:ln>子要素によって提供されます。このポイントは rel 属性の値として index を持っており、インデックス・ドキュメントの URI は href 属性の値として提供されます。
- 下方ナビゲーションのリンクは、<url>要素のコンテンツによって提供されます。すなわち、階層のより下位にあるドキュメントの URI は、その<loc>子要素の値として提供され、そのタイプは<rs:md>子要素の capability 属性の値として伝えられます。

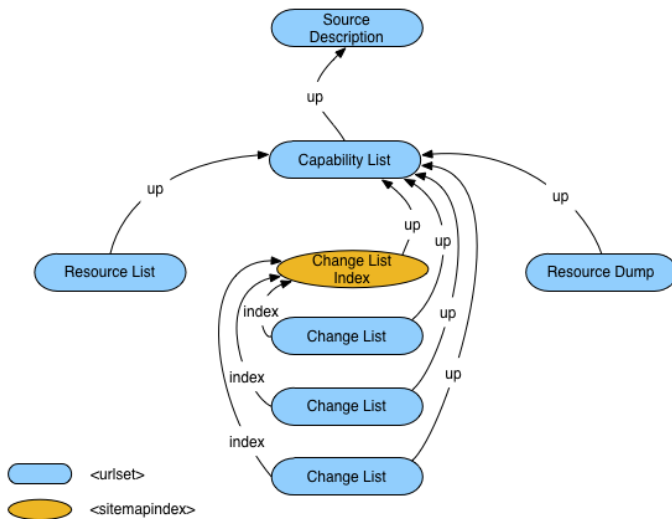


図 5. ResourceSync の上方ナビゲーション

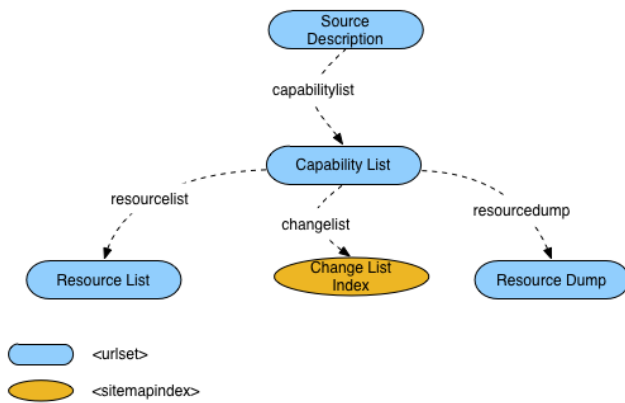


図 6. ResourceSync の下方ナビゲーション

6.3 検出

6.3.1 概要

ResourceSync では、ソースが ResourceSync をサポートするかどうか、また、どのようにサポートするのかをデスティネーションが検出するために、3つの方法を提供しています。それらは、[6.3.2 項](#)で詳述するソース全体のアプローチ、[6.3.3 項](#)で詳述するリソース固有のアプローチ、および、[6.3.4 項](#)で説明する robots.txt ファイルを介したサイトマップの検出という既存の慣行を活用するアプローチです。これらのすべてのアプローチを[図 7](#)にまとめます。

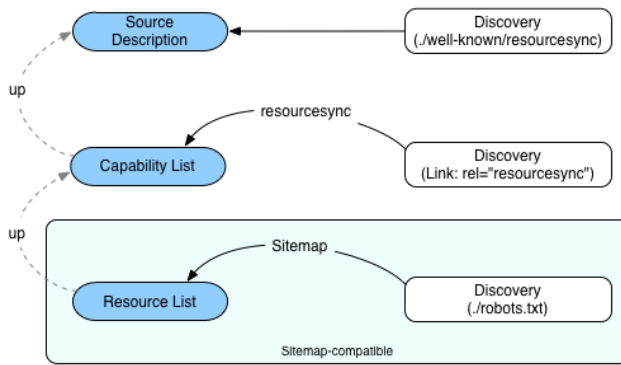


図 7. 「ソースの記述」と「機能リスト」の検出

6.3.2 ResourceSync の well-known URI

ソースは、[例 7](#)に示すような「ソースの記述」を公開しなければならず、また、ここで定義するように、公認の URI [\[RFC 5785\]](#) (`/.well-known/resourcesync`) で公開します。「ソースの記述」ドキュメントは、ソースの「機能リスト」を列挙しているため、ソースの機能を理解することに関心のあるデスクティネーションにとって適切なエントリーポイントになります。

6.3.3 リンク

「機能リスト」は、HTML ドキュメント [\[HTML Links, XHTML Links\]](#)、または、HTTP リンク・ヘッダー [\[RFC 5988\]](#) のいずれかで提供されるリンクによって検出可能にすることができます。

HTML ドキュメントに検出リンク (discovery link) を含めるために、「機能リスト」を指す `<link>` 要素がドキュメントの `<head>` に導入されます。この `<link>` には、値が `resourcesync` の `rel` 属性がなければなりません。この方法で検出可能にされた「機能リスト」は、当該リンクを提供するリソースに関係していなければなりません。これは、リンクされた「機能リスト」にリストされている機能でリソースを網羅しなければならないことを意味します。[例 9](#) は、「機能リスト」へのリンクを含む Web ページの構造を示しています。[例 6](#) に示すように、「ソースの記述」は、`<rs:ln>` 要素で提供される `up` 関係タイプのリンクをたどることで、「機能リスト」からを検出することができます。

```
<html>
  <head>
    <link rel="resourcesync"
          href="http://www.example.com/dataset1/capabilitylist.xml"/>
    ...
  </head>
  <body>...</body>
</html>
```

例 9. HTML リンクによる検出

「機能リスト」は、任意のコンテンツ・タイプ (content-type) のリソースの表現に含まれる HTTP リンク・ヘッダーによって検出可能にすることもできます。そうするには、HTTP リンク・ヘッダーにリンクを導入します。このリンクのターゲットは「機能リスト」の URI であり、その `rel` 属性の値は `resourcesync` です。この方法で検出可能にされた「機能リスト」は、当該リンクを提供するリソース

に関係していなければなりません。これは、リンクされた「機能リスト」にリストされている機能でリソースを網羅しなければならないことを意味します。[例 10](#) は、このアプローチを示す HTTP 応答ヘッダーの抜粋です。[例 6](#) に示すように、「ソースの記述」は、`<rs:ln>`要素で提供される `up` 関係タイプのリンクをたどることで、「機能リスト」からを検出することができます。

```
HTTP/1.1 200 OK
Date: Thu, 21 Jan 2010 00:02:12 GMT
Server: Apache
Link: <http://www.example.com/dataset1/capabilitylist.xml>;
      rel="resourcesync"
...
```

例 10. HTTP リンクによる検出

6.3.4 robots.txt

「Resource List」はサイトマップであるため、「Resource List」の URI を値として持つソースの `robots.txt` ファイルに `Sitemap` ディレクティブを追加するという確立されたアプローチを用いることで検出可能になります。ソースが複数のリソース・セットをサポートしている場合には、特定のリソース・セットに関連付けられた各「Resource List」に対して 1 つずつ、複数のディレクティブを追加することができます。ソースが通常のサイトマップと ResourceSync サイトマップ（「Resource List」）の両方をサポートしている場合には、[例 11](#) に示すように、複数の `Sitemap` ディレクティブを含めることで、それらを再び検出可能にすることができます。

この方法でリソース・セットの「Resource List」が検出されると、対応する「機能リスト」は、「Resource List」で提供される `up` 関係タイプのリンクをたどることで検出できるようになります。次に、「ソースの記述」は、「機能リスト」で提供される `up` 関係タイプの別のリンクをたどることで検出できます。

```
User-agent: *
Disallow: /cgi-bin/
Disallow: /tmp/
Sitemap: http://example.com/dataset1/resourcelist.xml
```

例 11. 「Resource List」を指す robots.txt ファイル

7. サイトマップ・ドキュメントの形式

ResourceSync フレームワークのリソースに関する情報を伝達する上で、[サイトマップ・プロトコル](#) によって導入されたサイトマップ（ルート要素 `<urlset>`）と、サイトマップ・インデックス（ルート要素 `<sitemapindex>`）のドキュメント形式が、さまざまな目的に使用されます。`<sitemapindex>` ドキュメント形式は、複数の `<urlset>` 形式のドキュメントをグループ化する必要がある際に使用されます。`<urlset>` 形式の複数のドキュメントを公開する際、ResourceSync フレームワークでは、コミュニティ定義の制限に従っています。本仕様の公開時点では、制限はドキュメントあたり 50,000 アイテム、ドキュメントサイズは 50MB です。

ドキュメント形式と、ResourceSync 拡張要素を [表 2](#) に示します。`<rs:md>` と `<rs:ln>` 要素は、それぞれメタデータとリンクを表現するために導入されます。どちらも ResourceSync の XML 名前空間にあり、属性を持つ可能性があります。これらの ResourceSync が定義する要素の属性を [表 3](#)

にリストし、以下に詳述します。例に示されるように、これらの属性には XML 名前空間接頭辞を含めてはなりません。<rs:ln>要素だけでなく、ResourceSync 属性のいくつかは他の仕様に基づいており、そのような場合には、当該の仕様に定義されたセマンティクス(semantics)を継承します。表 3 の「仕様」列は、そのような仕様を示しています。コミュニティは必要に応じて追加の属性を導入することができますが、ResourceSync 以外の XML 名前空間を使用し、それらの属性に対して名前空間接頭辞を適切に使用しなければなりません。

サイトマップ	サイトマップ・インデックス
<pre><?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 /> <rs:ln /> <url> <loc /> <rs:md /> <rs:ln /> </url> <url> ... </url> </urlset></pre>	<pre><?xml version="1.0" encoding="UTF-8"?> <sitemapindex xmlns="..." xmlns:rs="..."> <rs:md /> <rs:ln /> <sitemap> <loc /> <rs:md /> <rs:ln /> </sitemap> <sitemap> ... </sitemap> </sitemapindex></pre>

表 2. ResourceSync 拡張機能を含むサイトマップ・ドキュメント形式

ResourceSync ドキュメントの全体的な構造は以下のとおりです。

<urlset>または<sitemapindex>:これらの要素は、ResourceSync ドキュメントのルート要素です。本仕様では、サイトマップ・ドキュメント形式の子要素に必須の子要素と任意の子要素を 1 つずつ追加します。

- <rs:md>:このコンテキストでは、この要素はドキュメント自体に関する情報を伝えます。その使用は必須であり、次の属性を持つ場合があります。
 - at:この属性は、「Resource List」、「Resource List Index」、「Resource Dump」、「Resource Dump Index」および「Resource Dump Manifest」に使用され、「Change List」、「Change List Index」、「Change Dump」、「Change Dump Index」および「Change Dump Manifest」には使用されません。この属性の必要とされる使用については、各ドキュメントについて説明する項に詳述し、付録 A にまとめています。at 属性は、リソースをこの属性が関連するドキュメントに含めるために、そのスナップショットを取る処理が始まった日時を伝えます。したがって、ドキュメントで表される各リソースの状態が、少なくとも、at 属性の値として表された日時までのリソースへのすべての変更の結果であることが保証されます。属性値は [W3C Datetime](#) として表現され、YYYY-MM-DDThh:mm:ss[.s]Z の形式を使用して UTC で表現された、完全な日付と時刻を使用することが推奨されます。この属性は、図 1 の t1、t2、t3、t4 などのスナップショットの時間を表します。
 - capability:この属性は、すべての ResourceSync ドキュメントで必須です。属性の値は、ドキュメントの性質を伝えています。例. ドキュメントが「Resource List」、

「Change List」、「マニフェスト」などかどうか、定義される値が、`resourcelist`、`changelist`、`resourcedump`、`changedump`、`resourcedump-manifest`、`changedump-manifest`、`capabilitylist` および `description` など。

- `completed`: この任意の属性は、「Resource List」、「Resource List Index」、「Resource Dump」、「Resource Dump Index」および「Resource Dump Manifest」に使用され、「Change List」、「Change List Index」、「Change Dump」、「Change Dump Index」および「Change Dump Manifest」には使用されません。`completed` 属性は、リソースをこの属性に関連するドキュメントに含めるために、そのスナップショットを取る処理が完了した日時を伝えます。`at` 属性と `completed` 属性で提供される日時の組み合わせは、リソースが `at` 属性で表された日時での状態を超えて変更された可能性のある区間を表します。`completed` の属性値は [W3C Datetime](#) として表現され、`YYYY-MM-DDThh:mm:ss[.s]Z` の形式を使用して UTC で表現された、完全な日付と時刻を使用することが推奨されます。
- `from`: この属性は、「Change List」、「Change List Index」、「Change Dump」、「Change Dump Index」および「Change Dump Manifest」に使用され、「Resource List」、「Resource List Index」、「Resource Dump」、「Resource Dump Index」および「Resource Dump Manifest」には使用されません。この属性の必要とされる使用については、各ドキュメントについて説明する項に詳述し、[付録 A](#) にまとめています。この属性は、表現された日時から（および、`until` 属性が存在する場合には、それで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が、この属性に関連するドキュメントに含まれることを示します。属性値は [W3C Datetime](#) として表現され、`YYYY-MM-DDThh:mm:ss[.s]Z` の形式を使用して UTC で表現された、完全な日付と時刻を使用することが推奨されます。たとえば、[図 2](#) の最初の「Change List」は `t1` の `from` 値を、2 番目の「Change List」は `t3` の `from` 値を持つでしょう。
- `until`: この任意の属性は、「Change List」、「Change List Index」、「Change Dump」、「Change Dump Index」および「Change Dump Manifest」に使用され、「Resource List」、「Resource List Index」、「Resource Dump」、「Resource Dump Index」および「Resource Dump Manifest」には使用されません。この属性は、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が、この属性に関連するドキュメントに含まれることを示します。

- ドキュメントに `until` 属性がある場合には、当該ドキュメントがそれ以上更新されないことを示します。
- 変更ドキュメントに `until` 属性がない場合には、対応するリソース・セットへのその後のすべての変更によって、ドキュメントは更新されます。

属性値は [W3C Datetime](#) として表現され、`YYYY-MM-DDThh:mm:ss[.s]Z` の形式を使用して UTC で表現された、完全な日付と時刻を使用することが推奨されます。たとえば、[図 2](#) の最初の「Change List」は `t3` の `until` 値を、2 番目の「Change List」は `t5` の `until` 値を持つでしょう。

- `<rs:ln>`: リンクによる他のドキュメントの検出をサポートするために使用される、繰り返し可能な要素。この要素に必要とされる使用は、`<rs:ln>` が使用されるドキュメントを説明する項で詳しく説明しています。複数の属性が存在する場合があります、ResourceSync で定義される属性は以下のとおりです。

- href:他のドキュメントの URI を伝えるための必須属性。
- rel:関係を表すための必須属性。本仕様では、以下の値が明示的に使用されます。
 - describedby:「機能リスト」から、それに含まれるリソースのセットを説明するドキュメントへのリンク用、および、「ソースの記述」からソースを説明するドキュメントへのリンク用。
 - up:「機能リスト」から「ソースの記述」へのリンク用、および、「リソースのリスト」のような機能を伝えるドキュメントからそれが存在する「機能リスト」へのリンク用。
 - index:機能(例、「Resource List」)を伝えるドキュメントから、親インデックスドキュメント(例、「Resource List Index」)へのリンク用。
- <url>または<sitemap>:<urlset>要素には 0 またはそれ以上の<url>子要素があり、<sitemapindex>要素には 0 またはそれ以上の<sitemap>子要素があります。そのような各子要素は、ResourceSync フレームワークで役割を果たすリソースに関する情報を伝達するために使用されます。それらは以下の子要素を持つ場合があります。
 - <loc>:ResourceSync フレームワークで役割を果たすリソースの URI を伝える必須要素。
 - <lastmod>:[RFC 2616, 14.29 項](#)の定義のように、<loc>で提供される URI を使用してリソースの最終変更時刻を伝える要素。本項で前述したように、値は [W3C Datetime](#) として表されます。<lastmod>の使用は、すべての ResourceSync ドキュメントでオプションです。最終更新時間の使用と保守についての慣行には大きなばらつきがあること、表現された時間が最後の ResourceSync 変更イベントの `datetime` と一致していないことがあること、または、リソースを含む「Change List」または「Change Dump Manifest」の `from` および `until` 日付内にあることがあることに注意してください。
 - <changefreq>:<loc>で提供される URI を使用して、リソースの変更頻度に関するヒントを提供する任意の要素。定義された値は、`always`、`hourly`、`daily`、`weekly`、`monthly`、`yearly` および `never`。`always` の値は、変化するリソースがアクセスされるたびに使用します。`never` の値は、アーカイブされたリソースに使用すべきではありません。
 - <rs:md>:このコンテキストでは、この要素は<loc>で提供される URI を持つリソースに関連するメタデータを伝えます。該当する項で説明しているように、この要素は反復可能でなく、一部のドキュメントでは必須、その他のドキュメントでは任意です。複数の属性が存在する場合があります、ResourceSync で定義される属性は以下のとおりです。
 - `at` および `completed`:これらの属性のセマンティクスと値は、この項で前述したとおり。「Resource List Index」、「Resource Dump」および「Resource Dump Index」にのみ使用されます。この属性の必要とされる使用については、各ドキュメントについて説明する項に詳述し、[付録 A](#) にまとめています。
 - `capability`:この属性は、「ソースの記述」と「機能リスト」で必須です。その値は、<loc>属性の URI で識別されるリソースの性質を示します(例、「Resource List」、「Change List」、「Change Dump」など)。本仕様で定義

- される値は、`resourcelist`、`changelist`、`resourcedump`、`changedump` および `capabilitylist` です。
- `change`:この属性の値は、リソースに行われた変更のタイプを伝えます。本仕様で定義される値は、`created`、`updated` および `deleted` で、それぞれ、リソースの作成、更新および削除を伝えます。この属性は、[「Change List」\(12.1 項\)](#)および[「Change Dump Manifest」\(13.2 項\)](#)で使用されます。
 - `datetime`:この属性の値は、リソースの変更イベントの日時を伝えます。本項で前述したように、値は [W3C Datetime](#)として表されます。この属性は、[「Change List」\(12.1 項\)](#)および[「Change Dump Manifest」\(13.2 項\)](#)で使用される場合があります。値は、`from` 日時と同じまたはそれ以降、および、指定されている場合には、`until` 日時より前または同じでなければなりません。
 - `encoding`:この属性の値は、どのコンテンツ・コーディングがリソースに適用されているかを伝えます。`encoding` 属性の値は、[RFC 2616, 14.11 項](#)で定義されるように、HTTP 応答の `content-encoding` ヘッダの値に等しくなければなりません。
 - `from` および `until`:セマンティクスとこれらの属性の値は、本項で前述した定義のとおりです。これらは、「Change List Index」、「Change Dump」および「Change Dump Index」にのみ使用されます。これらの属性の必要とされる使用については、各ドキュメントについて説明する項に詳述し、[付録 A](#)にまとめています。
 - `hash`:この属性の値は、`<loc>`の URI が逆参照された際に返されるリソース表現のための固定情報を伝えます。この属性値は、空白で区切られたハッシュ値のリストの形式で表されます。各ハッシュ値は 16 進数でエンコードされたダイジェスト(digest)で表され、使用されたハッシュアルゴリズムを識別するトークン(token) (例. `md5:`、`sha-256:`) が前に付きます。
 - `length`:この属性の値は、`<loc>`の URI が逆参照された際に返されるリソース表現のためのコンテンツの長さを伝えます。`length` 属性の値は、HTTP 応答の `Content-Length` ヘッダの値に等しい必要があり、[RFC 2616, 4.4 項](#)で定義されるように計算されなければなりません。
 - `path`:この属性は、[「Resource Dump Manifest」\(11.2 項\)](#)および[「Change Dump Manifest」\(13.2 項\)](#)のみで使用されます。その値は、ZIP ファイルの `<loc>`の URI に関連付けられたビットストリームのファイルパスを伝えます。これは、ZIP が解凍された場合に、ビットストリームが存在する相対ファイル・システムパスです。
 - `type`:この属性の値は、`<loc>`の URI が逆参照された時に返されるリソース表現のメディアタイプ(Media Type)を伝えます。登録された値は、「[IANA MIME Media Type registry](#) (IANA MIME メディアタイプ・レジストリ)」に記載されています。
- `<rs:ln>`:このコンテキストでは、`<loc>`に提供された URI を持つものの 1 つ、たとえばミラーサイト上のコピー、リソースの旧バージョンなど([5.1 項](#)の「関連リソースへのリンク」を参照)に関連するリソースへのリンクに使用される、任意および反復可能な要素。複数の属性が存在する場合があります、ResourceSync で定義される属性は以下のとおりです。

- href: 関係するリソースのドキュメントの URI を伝えるための必須属性。
- rel:<loc>内の URI を持つリソースと href 内の URI を持つリソースの間の関係を伝えるための必須属性。rel 属性の値は、『[Relation Types Used in the ResourceSync Framework](#) (ResourceSync フレームワークで 사용되는関係タイプ)』ドキュメントにリストされています。本仕様で取り上げるものは以下のとおりです。
 - contents: 1つのビットストリーム・パッケージを指す「Resource Dump」または「Change Dump」内のエントリから、それぞれ、当該のビットストリーム・パッケージのための「Resource Dump Manifest」または「Change Dump Manifest」へのリンク用。
 - duplicate: リソースのミラーロケーションへのリンク用。
 - alternate および canonical: リソースの代替表現へのリンク用。
 - <http://www.openarchives.org/rs/terms/patch>: リソースの旧バージョンと現在のバージョンの違いを詳述するリソースへのリンク用。
 - describedby および describes: リソースに関する追加情報を提供するリンク用。
 - memento および timegate: リソースの旧バージョンにアクセスするためのリンク用。
 - collection: コレクション・メンバーシップを表すリンク用。
 - via: 出所情報を提供するリンク用。
- encoding、hash、length、modified、path、type: 本項の前述の定義および関連リソースに関連するような意味を持つ任意の属性。
- pri: 同じ関係タイプを持つリンク間の優先順位を表すために使用される任意の属性。属性値は 1~999,999 の整数で、整数が小さいほど優先順位が高く、属性がない場合は 999,999 の値を示します。

[表 3](#) に、ResourceSync ドキュメントで使用される要素をリストし、それぞれについて、それらと使用できる ResourceSync が定義する属性を示します。「仕様」列は、ResourceSync の同等物がセマンティクスを継承する元となった要素または属性が導入された仕様を指しています。属性の「表示」列のマークは、特定のリソース表現が関係する場合にのみ使用すべきであることを示し、「リソース」列のマークは、それがリソースの全般に使用可能であることを示しています。「[W3C XML スキーマ](#)」(<http://www.openarchives.org/rs/resourcesync.xsd> から入手可能)は、ResourceSync によって導入された要素の妥当性を確認するために設けられています。

ResourceSync フレームワークでは、上記以外の関係タイプを使用することができます。有効な関係タイプは、「[IANA Link Relation Type Registry](#) (IANA リンク関係タイプ・レジストリ)」に登録するか、[RFC 5988、4.2 項](#)で指定された URI として表現しなければなりません。『[Relation Types Used in the ResourceSync Framework](#)』ドキュメントでは、最新の概要を提供するように努めています。

要素／属性	仕様	リソース	表現
<urlset>または<sitemapindex>	サイトマップ・プロトコル		
<rs:md>	本仕様		
at	本仕様		
capability	本仕様		
completed	本仕様		
from	本仕様		
until	本仕様		
<rs:ln>	RFC4287		
href	RFC4287		
rel	RFC4287		
<url> OR <sitemap>	サイトマップ・プロトコル		
<loc>	サイトマップ・プロトコル		
<lastmod>	サイトマップ・プロトコル		
<changefreq>	サイトマップ・プロトコル		
<rs:md>	本仕様		
at	本仕様		
capability	本仕様		
change	本仕様	X	X
completed	本仕様		
datetime	本仕様	X	X
encoding	RFC2616		X
from	本仕様		
hash	Atom Link Extensions		X
length	RFC4287		X
path	本仕様		X
type	RFC4287		X
until	本仕様		
<rs:ln>	本仕様		
encoding	RFC2616		X
hash	Atom Link Extensions		X
href	RFC4287	X	X
length	RFC4287		X
modified	Atom Link Extensions	X	X
path	本仕様		X
pri	RFC6249	X	X
rel	RFC4287	X	X
type	RFC4287		X

表 3. ResourceSync ドキュメントのために定義された要素と関連属性

8. ソースの記述

「ソースの記述」は、ソースが提供する「機能のリスト」を列挙する必須のドキュメントです。ソースには、自身が区別するリソースのセットごとに1つの「機能リスト」が存在するため、「ソースの記述」には、ソースが区別するリソース・セットの数だけ「機能リスト」が列挙されます。

「ソースの記述」は、`<urlset>`形式に基づいています。これには`<urlset>`ルート要素と以下の構造が存在します。

- `<urlset>`の必須`<rs:md>`子要素には、`description` 値を持つ `capability` 属性がなければなりません。
- `describedby` 関係タイプを持つ推奨される`<urlset>`の`<rs:ln>`子要素は、ソースに関する情報を提供するドキュメントを指します。
- ソースが提供する各「機能リスト」には、`<urlset>`の`<url>`子要素が1つ含まれる必要があります。この要素には属性はありませんが、子要素を使用して「機能リスト」に関する情報を伝えます。`<url>`要素には、以下の子要素があります。
 - 必須の`<loc>`子要素は、それぞれの「機能リスト」のURIを提供します。
 - 任意の`<rs:md>`子要素には、URIが「機能リスト」を指していることを伝えるために、`capabilitylist` の値を持つ `capability` 属性がなければなりません。
 - `describedby` 関係タイプを持つ推奨される`<rs:ln>`子要素は、「機能リスト」で記述されたリソースのセットを記述するドキュメントを指します。

`<lastmod>`要素は、ソースが「機能リスト」の1つを更新するたびに「ソースの記述」を更新しない限り、「ソースの記述」から省略されなければなりません。

例 12 は、ソースが 3 つの「機能リスト」を提供する「ソースの記述」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="describedby"
    href="http://example.com/info_about_source.xml"/>
  <rs:md capability="description"/>
  <url>
    <loc>http://example.com/capabilitylist1.xml</loc>
    <rs:md capability="capabilitylist"/>
    <rs:ln rel="describedby"
      href="http://example.com/info_about_set1_of_resources.xml"/>
  </url>
  <url>
    <loc>http://example.com/capabilitylist2.xml</loc>
    <rs:md capability="capabilitylist"/>
    <rs:ln rel="describedby"
      href="http://example.com/info_about_set2_of_resources.xml"/>
  </url>
  <url>
    <loc>http://example.com/capabilitylist3.xml</loc>
    <rs:md capability="capabilitylist"/>
    <rs:ln rel="describedby"
      href="http://example.com/info_about_set3_of_resources.xml"/>
  </url>
</urlset>
```

例 12. 「ソースの記述」

ソースが複数の「ソースの記述」を公開する必要がある場合、または、公開することを選択した場合、「ソースの記述インデックス (Source Description Index)」を使用して、それらをグループ化しなければなりません。

9. 公表機能

「機能リスト」は、ソースが特定のリソース・セットのためにサポートするすべての機能を列挙したドキュメントです。ソースは、どのリソースが「機能リスト」に記述されたリソース・セットの一部であるかを定義します。そのようなセットが複数ある場合には、ソースは異なる機能リストでそれらを区別しなければなりません。どのリソースがどのセットの一部かについての選択は、メディアタイプ、コレクション・メンバーシップ、変更頻度、リソースの主題、その他多くの条件を含む、さまざまな基準を元に行われます。

「機能リスト」は、そのリソース・セットの機能ドキュメントを指します。すなわち、「Resource List」(10.1 項)、「Resource Dump」(11.1 項)、「Change List」(12.1 項)、および、「Change Dump」(13.1 項)があげられます。「機能リスト」には、機能ごとに 1 つのエントリのみが含まれなければなりません。

同じ「機能リスト」で伝達される機能は、その「機能リスト」に含まれるリソースのセットに均一に適用されます。たとえば、「機能リスト」が「Resource List」、「Resource Dump」および「Change List」を列挙する場合には、「Resource List」に表示される特定のリソースは「Resource Dump」にも表示され、リソースへの変更は「Change List」で伝達されなければなりません。

「機能リスト」は、`<urlset>`形式に基づいています。これには`<urlset>`ルート要素と以下の構造が存在します。

- `<urlset>`の必須`<rs:md>`子要素には、`capabilitylist` 値を持つ `capability` 属性がなければなりません。
- 関係タイプ `up` を持つ`<urlset>`の必須`<rs:ln>`子要素は、ソースが提供するすべての「機能リスト」を列挙する「ソースの記述」ドキュメントを指します。
- 関係タイプ `describedby` を持つ`<urlset>`の推奨される`<rs:ln>`子要素は、「機能リスト」に含まれるリソースのセットに関する情報を提供するドキュメントを指します。
- ソースが提供する機能ごとに、`<urlset>`の`<urlset>`子要素は1つ。この要素には属性はありませんが、機能に関する情報を伝えるために子要素を使用します。`<url>`要素には、以下の子要素があります。
 - 必須の`<loc>`子要素は、それぞれの機能ドキュメントのURIを提供します。
 - 必須の`<rs:md>`子要素には、それぞれの機能のタイプを伝える `capability` 属性がなければなりません。

`<lastmod>`要素は、ソースが機能ドキュメントの1つを更新するたびに「機能リスト」を更新しない限り、「機能リスト」から省略されなければなりません。

例 13 は、ソースが 4 つの機能を提供する「機能リスト」、すなわち、「Resource List」、「Resource Dump」、「Change List」および「Change Dump」を示しています。デスティネーションは、ソースが、たとえば「Resource List Index」または単一の「Resource List」を提供するかどうかを「機能リスト」から判断することはできません。その判断をするには、機能ドキュメントをダウンロードしなければなりません。<sitemapindex>ルート要素の付いたドキュメントがインデックスであり、<urlset>ルート要素の付いたドキュメントはそうではありません。

```
<?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="describedby"
    href="http://example.com/info_about_set1_of_resources.xml"/>
  <rs:ln rel="up"
    href="http://example.com/resourcesync_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>
</urlset>
```

例 13. 「機能リスト」

ResourceSync では少数の機能のみが定義され、それらを列挙しても単一の「機能リスト」の制限には達しません。したがって、<sitemapindex>ドキュメント構造は不要であり、ソースは、そのような構造を生成すべきではありません。

10. リソースの記述

ソースは、同期に使用できるようにするリソースに関する記述を公開できます。この情報により、デスティネーションでは、これらのリソースの一部またはすべての初期コピーを作成したり、ローカルコピーを更新して変更との同期を保ったりすることができます。

10.1 Resource List

「Resource List」は、ソースが同期のために利用できるようにしたリソースについてリストし記述するために導入されています。これは、特定の時点でのソースのリソースのスナップショットを提示します。

「Resource List」は、サイトマップ・プロトコルで導入された<urlset>ドキュメント形式に基づいています。これには<urlset>ルート要素と以下の構造が存在します。

- <urlset>の必須<rs:md>子要素には、resourcelist の値を持つ capability 属性がなければなりません。リソースを「Resource List」に含めるために、そのスナップショットを取得する処理が開始された日時を伝える at 属性もなければなりません。また、その処理が完了した日時を伝える completed 属性がある場合もあります。
- <urlset>の必須<rs:ln>子要素は、関係タイプ up の「機能リスト」を指します。
- 「[Resource List Index](#)」が存在する場合には、<urlset>の推奨される<rs:ln>子要素は、関係タイプ index でそれを指します。
- 各リソースには、<urlset>の<url>子要素が 1 つ含まれる必要があります。この要素には属性はありませんが、リソースに関する情報を伝えるために子要素を使用します。<url>要素には、以下の子要素があります。
 - 必須の<loc>子要素は、リソースの URI を提供します。
 - [7 項](#)で記述されたセマンティクスを持つ任意の<lastmod>および<changefreq>子要素。
 - 任意の<rs:md>子要素は、リソースに関する追加のメタデータを提供します。これには [7 項](#)で記述された属性が存在する場合があります。
 - 任意の<rs:ln>子要素は、[7 項](#)で説明され、[14 項](#)で詳述されているように、関連するリソースにリンクします。

例 14 は、2つのリソースを持つ「Resource List」を示しています。at 属性により、デスティネーションは、リストされたリソースのどちらも、それぞれ 2013-01-02T13:00:00Z と 2013-01-02T14:00:00Z の最終変更日時に変更が行われなかったことを判断することができ、また、at 属性の値である日時は、2013-01-03T09:00:00Z です。

```
<?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="resourcelist"
    at="2013-01-03T09:00:00Z"
    completed="2013-01-03T09:01:00Z"/>
  <url>
    <loc>http://example.com/res1</loc>
    <lastmod>2013-01-02T13:00:00Z</lastmod>
    <rs:md hash="md5:1584abdf8ebdc9802ac0c6a7402c03b6"
      length="8876"
      type="text/html"/>
  </url>
  <url>
    <loc>http://example.com/res2</loc>
    <lastmod>2013-01-02T14:00:00Z</lastmod>
    <rs:md hash="md5:1e0d5cb8ef6ba40c99b14c0237be735e
      sha-
256:854f61290e2e197a11bc91063afce22e43f8ccc655237050ace766adc68dc784"
      length="14599"
      type="application/pdf"/>
  </url>
</urlset>
```

例 14. 「Resource List」

10.2 Resource List Index

ResourceSync フレームワークでは、<urlset>形式のドキュメントを公開するためのコミュニティ定義の制限を採用し、複数の「Resource List」をグループ化するための「Resource List Index」を導入しています。「Resource List Index」で参照される「Resource List」の集合体は、ソースが同期に使用できるようにしたリソースのセット全体を表します。このリソース・セットは、単一の「Resource List」で、あるいは、「Resource List Index」を介して複数の「Resource List」で伝達されるかに関係なく、ある時点でのソースのデータの状態を表します。

「Resource List Index」は、サイトマップ・プロトコルで導入された<sitemapindex>ドキュメント形式に基づいています。これには<sitemapindex>ルート要素と以下の構造が存在します。

- <sitemapindex>の必須の<rs:md>子要素には、resourcelist の値のある capability 属性がなければなりません。また、リソースを「Resource List Index」に含めるために、そのスナップショットを取得する処理が開始された日時を伝える at 属性もなければなりません。また、その処理が完了した日時を伝える completed 属性がある場合もあります。
- <sitemapindex>の必須<rs:ln>子要素は、関係タイプ up の「機能リスト」を指します。

- 各リソースには、<sitemapindex>の<sitemap>子要素が1つ含まれる必要があります。この要素には属性はありませんが、「Resource List」に関する情報を伝えるために子要素を使用します。<sitemap>要素には、以下の子要素があります。
 - 必須の<loc>子要素は、「Resource List」のURIを提供します。
 - [7項](#)で記述されたセマンティクスを持つ任意の<lastmod>子要素。
 - リソースを「Resource List」に含めるために、そのスナップショットを取得する処理がそれぞれ開始および終了した日時を伝える `at` 属性、および、場合によっては、`completed` 属性を持つ任意の<rs:md>子要素。

デスティネーションは、ルート要素がそれぞれ<urlset>または<sitemapindex>であるかどうかに基づいて、「Resource List」または「Resource List Index」に到達したかどうかを判断できます。[例 15](#)に、3つの「Resource List」を指す「Resource List Index」を示します。

```
<?xml version="1.0" encoding="UTF-8"?>
<sitemapindex 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="resourcelist"
    at="2013-01-03T09:00:00Z"
    completed="2013-01-03T09:10:00Z"/>
  <sitemap>
    <loc>http://example.com/resourcelist1.xml</loc>
    <rs:md at="2013-01-03T09:00:00Z"/>
  </sitemap>
  <sitemap>
    <loc>http://example.com/resourcelist2.xml</loc>
    <rs:md at="2013-01-03T09:03:00Z"/>
  </sitemap>
  <sitemap>
    <loc>http://example.com/resourcelist3.xml</loc>
    <rs:md at="2013-01-03T09:07:00Z"/>
  </sitemap>
</sitemapindex>
```

例 15. 「Resource List Index」

例 16 は、URI (<http://example.com/resourcelist1.xml>) で識別される「Resource List」の内容を示しています。構造的には、**例 14** に示される「Resource List」と同じですが、**例 15** に示されるように、関係タイプ `index` とともに、親の「Resource List Index」へのナビゲーション・リンクを提供する追加的な `<urlset>` の子要素 `<rs:ln>` が含まれています。このリンクは、デスティネーションのナビゲーションを簡単にするためのものであるため、その採用が推奨されます。

```
<?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:ln rel="index"
    href="http://example.com/dataset1/resourcelist-index.xml"/>
  <rs:md capability="resourcelist"
    at="2013-01-03T09:00:00Z"/>
  <url>
    <loc>http://example.com/res3</loc>
    <rs:md hash="md5:1584abdf8ebdc9802ac0c6a7402c8753"
      length="4385"
      type="application/pdf"/>
  </url>
  <url>
    <loc>http://example.com/res4</loc>
    <rs:md hash="md5:4556abdf8ebdc9802ac0c6a7402c9881"
      length="883"
      type="image/png"/>
  </url>
</urlset>
```

例 16: 親の「Resource List Index」へのナビゲーション・リンクを持つ「Resource List」

11. リソースのパッケージ化

少数の HTTP 要求を使用してデスティネーションがソースのデータをコピーする効率的な方法を提供するために、ソースはリソースのパッケージ化されたビットストリームを提供することができます。

11.1 Resource Dump

ソースは、リソースのビットストリームのパッケージへのリンクを提供する「Resource Dump」を公開することができます。「Resource Dump」は、ある時点でのソースの状態を表します。これを利用することで、デスティネーションが多くの個別の要求を行う代わりに、ソースからリソースを一括して転送することができます。

ResourceSync フレームワークでは、パッケージングの形式として [ZIP ファイル形式](#) の使用を推奨しています。コミュニティは独自のパッケージング形式を定義する場合がありますが、それにより、コミュニティ間の相互運用性に悪影響が及ぼされることに注意する必要があります。「Resource Dump」は、同じ形式のパッケージのみを指している必要があります。

「Resource Dump」は、サイトマップ・プロトコルで導入された `<urlset>` ドキュメント形式に基づいています。これには `<urlset>` ルート要素と以下の構造が存在します。

- `<urlset>` の必須 `<rs:md>` 子要素には、`resourcedump` の値のある `capability` 属性がなければなりません。また、リソースを「Resource Dump」に含めるために、そのスナップショットを取得する処理が開始された日時を伝える `at` 属性もなければなりません。また、その処理が完了した日時を伝える `completed` 属性がある場合もあります。
- `<urlset>` の必須 `<rs:ln>` 子要素は、関係タイプ `up` の「機能リスト」を指します。
- [「Resource Dump Index」](#) が存在する場合には、`<urlset>` の推奨される `<rs:ln>` 子要素は、関係タイプ `index` でそれを指します。
- 各ビットストリーム・パッケージには、`<urlset>` の `<url>` 子要素が 1 つ含まれる必要があります。この要素には属性はありませんが、パッケージに関する情報を伝えるために子要素を使用します。`<url>` 要素には、以下の子要素があります。
 - 必須の `<loc>` 子要素は、パッケージの URI を提供します。
 - [7 項](#) で記述されたセマンティクスを持つ任意の `<lastmod>` 子要素。
 - `type` および `length` 属性をそれぞれ使用して、メディアタイプおよびパッケージの長さを伝えるための推奨される `<rs:md>` 子要素。また、この要素は、リソースをパッケージに含めるために、そのスナップショットを取得する処理がそれぞれ開始および終了した日時を伝える `at` 属性、および、場合によっては、`completed` 属性を持つ場合があります。さらに、[7 項](#) の説明のように、子要素は `hash` などの属性を持つ場合があります。
 - ビットストリーム・パッケージに関連付けられた「Resource Dump Manifest」を指す関係タイプ `contents` を持つ任意の `<rs:ln>` 子要素。

[例 17](#) は、3 つの ZIP ファイルを指す「Resource Dump」を示しています。各 `<url>` 要素には、パッケージに関連付けられた [「Resource Dump Manifest」](#) へのポインタが含まれています。このポインタは任意であり、デスティネーションのための利便性を目的としていますが、提供される場合に

は、ソースは参照されるマニフェストがビットストリーム・パッケージに含まれるマニフェストに対応することを確認する必要があります。

```
<?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="resourcedump"
    at="2013-01-03T09:00:00Z"
    completed="2013-01-03T09:04:00Z"/>
  <url>
    <loc>http://example.com/resourcedump-part1.zip</loc>
    <rs:md type="application/zip"
      length="4765"
      at="2013-01-03T09:00:00Z"
      completed="2013-01-03T09:02:00Z"/>
    <rs:ln rel="contents"
      href="http://example.com/resourcedump_manifest-part1.xml"
      type="application/xml"/>
  </url>
  <url>
    <loc>http://example.com/resourcedump-part2.zip</loc>
    <rs:md type="application/zip"
      length="9875"
      at="2013-01-03T09:01:00Z"
      completed="2013-01-03T09:03:00Z"/>
    <rs:ln rel="contents"
      href="http://example.com/resourcedump_manifest-part2.xml"
      type="application/xml"/>
  </url>
  <url>
    <loc>http://example.com/resourcedump-part3.zip</loc>
    <rs:md type="application/zip"
      length="2298"
      at="2013-01-03T09:03:00Z"
      completed="2013-01-03T09:04:00Z"/>
    <rs:ln rel="contents"
      href="http://example.com/resourcedump_manifest-part3.xml"
      type="application/xml"/>
  </url>
</urlset>
```

例 17. 「Resource Dump」

ソースが複数の Resource Dump を公開する必要がある、または、公開することを選んだ場合には、[11.2 項](#)の説明と同様の方法で、「Resource Dump Index」を使用してグループ化しなければなりません。

11.2 Resource Dump Manifest

「Resource Dump」から参照される各 ZIP パッケージには、パッケージの構成ビットストリームを記述する「**Resource Dump Manifest**」ファイルが含まれていなければなりません。ファイルは `manifest.xml` と名付け、ZIP パッケージの最上位に配置しなければなりません。

「Resource Dump Manifest」は、<urlset>形式に基づいています。これには<urlset>ルート要素と以下の構造が存在します。

- <urlset>の必須<rs:md>子要素には、resourcedump-manifest の値のある capability 属性がなければなりません。また、リソースを ZIP パッケージに含めるために、そのスナップショットを取得する処理が開始された日時を伝える at 属性もなければなりません。また、その処理が完了した日時を伝える completed 属性がある場合もあります。
- <urlset>の必須<rs:ln>子要素は、関係タイプ up の「機能リスト」を指します。
- 各ビットストリームにつき、<urlset>の1つの<url>子要素。この要素には属性はありませんが、ビットストリームに関する情報を伝えるために子要素を使用します。<url>要素には、以下の子要素があります。
 - 必須の<loc>子要素は、ソースがビットストリームに関連付ける URI を提供します。
 - [7 項](#)で記述されたセマンティクスを持つ任意の<lastmod>および<changefreq>子要素。
 - 必須の<rs:md>子要素には、パッケージ内のビットストリームの場所を伝える path 属性がなければなりません。この属性の値は、パッケージのルートに対して相対的であり、先頭にスラッシュ(/)を付けて表現されます。デスティネーションがビットストリームのメディアタイプを決定できるように、<rs:md>要素で type 属性を使用することが推奨されます。<rs:md>要素は、さらに、[7 項](#)の記述のように、hash および length 属性を持っている場合があります。
 - 任意の<rs:ln>子要素は、[7 項](#)で説明され、[14 項](#)で詳述されているように、関連するリソースにリンクします。

例 18 は、2つのビットストリームを含む ZIP ファイルの「Resource Dump Manifest」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="resourcedump-manifest"
    at="2013-01-03T09:00:00Z"
    completed="2013-01-03T09:02:00Z"/>
  <url>
    <loc>http://example.com/res1</loc>
    <lastmod>2013-01-02T13:00:00Z</lastmod>
    <rs:md hash="md5:1584abdf8ebdc9802ac0c6a7402c03b6"
      length="8876"
      type="text/html"
      path="/resources/res1"/>
  </url>
  <url>
    <loc>http://example.com/res2</loc>
    <lastmod>2013-01-02T14:00:00Z</lastmod>
    <rs:md hash="md5:1e0d5cb8ef6ba40c99b14c0237be735e
      sha-
256:854f61290e2e197a11bc91063afce22e43f8ccc655237050ace766adc68dc784"
      length="14599"
      type="application/pdf"
      path="/resources/res2"/>
  </url>
</urlset>
```

例 18. 「Resource Dump Manifest」

12. 変更の記述

ソースは、リソースへの変更の記録を公開することができます。これにより、デスティネーションは変更について効率的に確認し、増分的に同期することができます。

12.1 Change List

「Change List」は、ソースによるリソースへの変更についての記述を含んだドキュメントです。「Change List」の公開頻度と、カバーする時区間は、ソースが決定します。たとえば、ソースは、「Change List」ごとに一定数の変更を公開、あるいは、1 時間、1 日、1 週間などの固定長の期間内のすべての変更を公開することを選択できます。「Change List」のすべてのエントリは、時系列順で提供しなければなりません。すなわち、「Change List」の最初には、最も長い間変更されていないリソースをリストし、ドキュメントの最後には、ごく最近変更されたリソースをリストしなければなりません。「Change List」の対象期間中にリソースに複数の変更が加えられた場合には、変更ごとに 1 回ずつ、リソースが複数回リストされます。

「Change List」は、サイトマップ・プロトコルで導入された<urlset>ドキュメント形式に基づいています。これには<urlset>ルート要素と以下の構造が存在します。

- `<urlset>`の必須`<rs:md>`子要素には、`changelist` の値のある `capability` 属性がなければなりません。また、「Change List」に含まれる時区間を含めるために、必須の `from` 属性と任意の `until` 属性も持っています。これらの属性のセマンティクスの詳細は以下に記します。
- `<urlset>`の必須`<rs:ln>`子要素は、関係タイプ `up` の「機能リスト」を指します。
- 「[Change List Index](#)」が存在する場合には、`<urlset>`の推奨される`<rs:ln>`子要素は、関係タイプ `index` でそれを指します。
- 各リソースの変更につき、`<urlset>`の1つの`<url>`子要素。この要素には属性はありませんが、変更されたリソースに関する情報を伝えるために子要素を使用します。`<url>`要素には、以下の子要素があります。
 - 必須の`<loc>`子要素は、変更されたリソースの URI を提供します。
 - [7 項](#)で記述されたセマンティクスを持つ任意の`<lastmod>`および`<change freq>`子要素。
 - 必須の`<rs:md>`子要素には、変更の性質を伝える `change` 属性がなければなりません。`change` 属性の値は、`created`、`updated` または `deleted` です。[7 項](#)の記述のように、要素には、`datetime`、`hash`、`length` および `type` 属性がある場合もあります。
 - 任意の`<rs:ln>`子要素は、[7 項](#)で説明され、[14 項](#)で詳述されているように、関連するリソースにリンクします。

「Change List」でカバーされる時区間は、`<urlset>`ルート要素の`<rs:md>`子要素の `from` および `until` 属性によって伝達されます。`from` 属性は、属性の値として表される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が「Change List」に含まれることを示します。`until` 属性が存在する場合には、属性の値として表される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が「Change List」に含まれることを示します。その使用は「Change List」ではオプションです。

- ドキュメントに `until` 属性が含まれている場合には、ドキュメントがそれ以上更新されないことが示されます。「Change List」は閉じられます (closed)。デスティネーションは、閉じられた「Change List」の処理を終了した後、添付の「Change List Index」(`index` 関係タイプでリンクをたどる)、または、該当する場合には、「機能リスト」(`up` 関係タイプでリンクをたどる)を参照し、閉じられた「Change List」の `until` 値として表される日時の後に発生した変更を報告する「Change List」の URI を判断する必要があります。
- ドキュメントに `until` 属性が含まれていない場合には、ドキュメントがさらなる変更によって更新されることが示されます。「Change List」は開いたままです (open)。デスティネーションは、開いている「Change List」のポーリング (poll) を継続して、さらなる変更について確認する必要があります。添付の「Change List Index」や、該当する場合でも、「機能リスト」を参照する必要はありません。

`from` および `until` 属性は、デスティネーションが「Change List」を完全に処理したかどうかを判断する助けになります。「Change List」の変更の時系列順、リソース変更の日時 (提供されている場合)、および、変更されたリソースの URI は、デスティネーションが完全に処理されていない「Change List」の中の最初の未処理の変更を判断するのに役立ちます。デスティネーションは、こ

こから処理を開始する必要があります。つまり、変更を伝える<url>要素の<loc>子要素で提供される URI を逆参照することにより、変更されたリソースの表現を取得することができます。

例 19 は、2013-01-03T00:00:00Z 以降に 4 つのリソースの変更 (1 つの作成、2 つの更新、1 つの削除) が発生したことを示す「Change List」を表しています。リソースの 1 つには、これらの変更のうち 2 つが行われたため、2 回リストされています。「Change List」には `until` 属性がありませんが、それにより、さらなる変更が報告される場合があることが示されます。そのため、デスティネーションでは、この「Change List」をポーリングし続ける必要があります。この例は、任意の <lastmod> 要素と任意の `datetime` 属性の違いを示しています。作成イベントは、イベントの `datetime` より早く、「Change List」の `from` 日時より前の <lastmod> を持つリソースのためのものです。最初の更新イベントは、一致する <lastmod> 要素と `datetime` 属性を示します。2 番目の更新イベントには <lastmod> 要素も `datetime` 属性もありませんが、「Change List」内のその位置により、それがより最近のリソースの更新だったことが示されます。ソースは、各イベントに `datetime` 属性を提供することにより、デスティネーションが最後に処理した変更イベントを簡単に判断できるようにすることができます。

```
<?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="changelist"
    from="2013-01-03T00:00:00Z"/>
  <url>
    <loc>http://example.com/res1.html</loc>
    <lastmod>2000-01-01T01:01:00Z</lastmod>
    <rs:md change="created" datetime="2013-01-03T11:00:00Z"/>
  </url>
  <url>
    <loc>http://example.com/res2.pdf</loc>
    <lastmod>2013-01-03T13:00:00Z</lastmod>
    <rs:md change="updated" datetime="2013-01-03T13:00:00Z"/>
  </url>
  <url>
    <loc>http://example.com/res3.tiff</loc>
    <rs:md change="deleted" datetime="2013-01-03T18:00:00Z"/>
  </url>
  <url>
    <loc>http://example.com/res2.pdf</loc>
    <rs:md change="updated"/>
  </url>
</urlset>
```

例 19. 4 つのリソースの変更を説明する開かれた「Change List」

12.2 Change List Index

ソースが複数の「Change List」を公開する必要がある場合には、それらを「Change List Index」にグループ化しなければなりません。「Change List Index」は、「Change List」を時系列で列挙しなければなりません。

「Change List Index」は、サイトマップ・プロトコルで導入された <sitemapindex> ドキュメント形式に基づいています。これには <sitemapindex> ルート要素と以下の構造が存在します。

- `<sitemapindex>`の必須`<rs:md>`子要素には、`changelist` の値のある `capability` 属性がなければなりません。また、「Change List」に含まれる時区間を含めるために、必須の `from` 属性と任意の `until` 属性も持っています。これらの属性のセマンティクスは、[「Change List」](#)で詳述されたとおりです。
- `<sitemapindex>`の必須`<rs:ln>`子要素は、関係タイプ `up` の「機能リスト」を指します。
- 各「Change List」につき、`<sitemapindex>`の`<sitemap>`子要素が1つ含まれている必要があります。この要素には属性はありませんが、「Change List」に関する情報を伝えるために子要素を使用します。`<sitemap>`要素には、以下の子要素があります。
 - 必須の`<loc>`子要素は、「Change List」の URI を提供します。
 - [7 項](#)で記述されたセマンティクスを持つ任意の`<lastmod>`子要素。
 - 推奨される`<rs:md>`子要素には、「Change List」でカバーされる時区間を伝えるための `from` 属性と、場合によっては、`until` 属性が含まれます。`until` の使用は次のとおりです。
 - 「Change List」が閉じている場合には、`<until>`の使用が必要です。
 - 「Change List」が開いている場合には、`<until>`を指定してはなりません。

デスティネーションは、ルート要素がそれぞれ`<urlset>`または`<sitemapindex>`であるかどうかに基づいて、「Change List」または「Change List Index」に到達したかどうかを判断する必要があります。

[例 20](#)に、3つの「Change List」を指す「Change List Index」を示します。「Change List」の2つは、`until` 属性の存在によって示されるように閉じられており、もう1つは、その不在によって示されるように開かれています。閉じられた「Change List」(<http://example.com/20130102-changelist.xml>) を [例 21](#)に示します。開かれた「Change List」には、[例 19](#)に示すケースのように、そのリストに「Change List Index」を指す `index` 関係タイプを持つ追加のリンクがあります。

```
<?xml version="1.0" encoding="UTF-8"?>
<sitemapindex 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="changelist"
    from="2013-01-01T00:00:00Z"/>
  <sitemap>
    <loc>http://example.com/20130101-changelist.xml</loc>
    <rs:md from="2013-01-01T00:00:00Z"
      until="2013-01-02T00:00:00Z"/>
  </sitemap>
  <sitemap>
    <loc>http://example.com/20130102-changelist.xml</loc>
    <rs:md from="2013-01-02T00:00:00Z"
      until="2013-01-03T00:00:00Z"/>
  </sitemap>
  <sitemap>
    <loc>http://example.com/20130103-changelist.xml</loc>
    <rs:md from="2013-01-03T00:00:00Z"/>
  </sitemap>
</sitemapindex>
```

例 20. 「Change List Index」

```
<?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:ln rel="index"
    href="http://example.com/dataset1/changelist.xml"/>
  <rs:md capability="changelist"
    from="2013-01-02T00:00:00Z"
    until="2013-01-03T00:00:00Z"/>
  <url>
    <loc>http://example.com/res7.html</loc>
    <rs:md change="created" datetime="2013-01-02T12:00:00Z"/>
  </url>
  <url>
    <loc>http://example.com/res9.pdf</loc>
    <rs:md change="updated" datetime="2013-01-02T13:00:00Z"/>
  </url>
  <url>
    <loc>http://example.com/res5.tiff</loc>
    <rs:md change="deleted" datetime="2013-01-02T19:00:00Z"/>
  </url>
  <url>
    <loc>http://example.com/res7.html</loc>
    <rs:md change="updated" datetime="2013-01-02T20:00:00Z"/>
  </url>
</urlset>
```

例 21. インデックスを指す閉じられた「Change List」

13. 変更のパッケージ化

リソースの変更を取得するために必要な要求の数を減らすために、ソースは変更されたリソースのパッケージ化されたビットストリームを提供することができます。

13.1 Change Dump

コンテンツの変更をダウンロード可能にするために、ソースは「**Change Dump**」を公開することができます。「Change Dump」は、ソースの変更されたリソースのビットストリームを含むパッケージを指すドキュメントです。ResourceSync フレームワークでは、[ZIP ファイル形式](#)の使用をパッケージ化の形式として指定しています。コミュニティは独自のパッケージ形式を定義することができます。「Change Dump」は、同じ形式のパッケージのみを指している必要があります。

これらのパッケージの公開頻度、および、カバーする時区間は、ソースが決定します。たとえば、ソースは、パッケージごとに一定数の変更を公開、あるいは、1 時間、1 日、1 週間などの固定長の期間内のすべての変更を公開することを選択できます。1 つのパッケージの対象期間内にリソースに複数の変更が加えられた場合には、パッケージには変更ごとに 1 つずつ、リソースの複数のビットストリームが含まれます。新しいパッケージが公開されると、新しいエントリがそれらを指す「Change Dump」に追加されます。「Change Dump」のすべてのエントリは、時系列順で提供される必要があります。「Change Dump」の最初にリストされているパッケージは最も長い間公開されていないもの、ドキュメントの最後にリストされているパッケージはごく最近公開されたものです。

「Change Dump」は、サイトマップ・プロトコルで導入された<urlset>ドキュメント形式に基づいています。これには<urlset>ルート要素と以下の構造が存在します。

- <urlset>の必須<rs:md>子要素には、changedump の値のある capability 属性がなければなりません。また、「Change Dump」に含まれる時区間を含めるために、必須の from 属性と任意の until 属性も持っています。これらの属性のセマンティクスは「[Change List](#)」に詳述されたとおりです。
- <urlset>の必須<rs:ln>子要素は、関係タイプ up の「機能リスト」を指します。
- 「[Change Dump Index](#)」が存在する場合には、<urlset>の推奨される<rs:ln>子要素は、関係タイプ index でそれを指します。
- 各ビットストリーム・パッケージにつき、<urlset>の<url>子要素が 1 つ含まれている必要があります。この要素には属性はありませんが、パッケージに関する情報を伝えるために子要素を使用します。<url>要素には、以下の子要素があります。
 - 必須の<loc>子要素は、パッケージの URI を提供します。
 - [7 項](#)で記述されたセマンティクスを持つ任意の<lastmod>子要素。
 - type および length 属性をそれぞれ使用して、メディアタイプおよびパッケージの長さを伝えるための推奨される<rs:md>子要素。また、パッケージに含まれる時区間を伝えるために from および until 属性を持つ場合もあります。さらに、[7 項](#)の説明のように、子要素は hash などの属性を持つ場合があります。
 - 各パッケージに関連付けられた「Change Dump Manifest」のコピーを指す関係タイプ contents を持つ任意の<rs:ln>子要素。

例 22 は、変更されたリソースに関連付けられた 3 つのビットストリーム・パッケージへのポインタを含む「Change Dump」を示しています。until 属性の不在は、さらにパッケージが追加されることを示しています。この例には、各<url> 要素内の、パッケージに関連付けられた「[Change Dump Manifest](#)」のコピーへのポインタも含まれています。このポインタはオプションであり、デスティネーションがパッケージをダウンロードする必要があるかどうかを判断できるようにすることを目的としています。そのようなポインタが提供されている場合には、ソースは、参照されるマニフェストがビットストリーム・パッケージに含まれるマニフェストと一致することを確認する必要があります。

```
<?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="changedump"
    from="2013-01-01T00:00:00Z"/>
  <url>
    <loc>http://example.com/20130101-changedump.zip</loc>
    <lastmod>2013-01-01T23:59:59Z</lastmod>
    <rs:md type="application/zip"
      length="3109"
      from="2013-01-01T00:00:00Z"
      until="2013-01-02T00:00:00Z"/>
    <rs:ln rel="contents"
      href="http://example.com/20130101-changedump-manifest.xml"
      type="application/xml"/>
  </url>
  <url>
    <loc>http://example.com/20130102-changedump.zip</loc>
    <lastmod>2013-01-02T23:59:59Z</lastmod>
    <rs:md type="application/zip"
      length="6629"
      from="2013-01-02T00:00:00Z"
      until="2013-01-03T00:00:00Z"/>
    <rs:ln rel="contents"
      href="http://example.com/20130102-changedump-manifest.xml"
      type="application/xml"/>
  </url>
  <url>
    <loc>http://example.com/20130103-changedump.zip</loc>
    <lastmod>2013-01-03T23:59:59Z</lastmod>
    <rs:md type="application/zip"
      length="8124"
      from="2013-01-03T00:00:00Z"
      until="2013-01-04T00:00:00Z"/>
    <rs:ln rel="contents"
      href="http://example.com/20130103-changedump-manifest.xml"
      type="application/xml"/>
  </url>
</urlset>
```

例 22. 「Change Dump」

ソースが複数の「Change Dump」を公開する必要がある場合には、[12.2 項](#)の説明と同様の方法で、「Change Dump Index」にグループ化しなければなりません。

13.2 Change Dump Manifest

「Change Dump」から参照される各 ZIP パッケージには、パッケージの構成ビットストリームを記述する「Change Dump Manifest」ファイルが含まれていなければなりません。ファイルは `manifest.xml` と名付けられ、ZIP パッケージの最上位に配置されなければなりません。「Change Dump Manifest」のすべてのエントリは、時系列順で提供される必要があります。最も前のリソースの変更に関連付けられたビットストリームが最初にリストされ、最新の變更に関連付けられたビットストリームが最後にリストされます。

「Change Dump Manifest」は、`<urlset>` 形式に基づいています。これには `<urlset>` ルート要素と以下の構造が存在します。

- `<urlset>` の必須 `<rs:md>` 子要素には、`changedump-manifest` の値のある `capability` 属性がなければなりません。また、「Change Dump Manifest」が含まれる ZIP パッケージに含まれる時区間を含めるために、必須の `from` 属性と任意の `until` 属性も持っています。指定された期間中に発生したすべてのリソース變更のビットストリームが ZIP パッケージに含まれなければならないが、また、「Change Dump Manifest」は、それぞれを参照しなければなりません。
- `<urlset>` の必須 `<rs:ln>` 子要素は、関係タイプ `up` の「機能リスト」を指します。
- 各リソースの變更、つまり、各ビットストリームにつき、`<urlset>` の `<url>` 子要素が 1 つ含まれている必要があります。この要素には属性はありませんが、リソースの變更と関連するビットストリームに関する情報を伝えるために子要素を使用します。`<url>` 要素には、以下の子要素があります。
 - 必須の `<loc>` 子要素は、ソースがビットストリームに関連付ける URI を提供します。
 - [7 項](#) で記述されたセマンティクスを持つ任意の `<lastmod>` および `<change freq>` 子要素。
 - 必須の `<rs:md>` 子要素には、リソースに加えられた變更のタイプを伝える `change` 属性がなければなりません。`change` 属性値は、`created`、`updated` または `deleted` です。`change = "deleted"` の場合を除き、要素には、ZIP パッケージ内のビットストリームの場所を伝える `path` 属性もなければなりません。パス (`path`) はパッケージのルートと相対的で、先頭にスラッシュ (`/`) を付けて表現されます。デスティネーションがビットストリームのメディアタイプを判断できるようにするために、`<rs:md>` 要素で `type` 属性を使用することが強く推奨されます。[7 項](#) の記述のように、`<rs:md>` 要素には、さらに `datetime`、`hash` および `length` 属性がある場合もあります。
 - 任意の `<rs:ln>` 子要素は、[7 項](#) で説明され、[14 項](#) で詳述されているように、関連するリソースにリンクします。

[例 23](#) は、[例 22](#) の「Resource Dump」の 2 番目のエントリに関連付けられた「Change Dump Manifest」を示しています。マニフェストは、ZIP パッケージの最上部で `manifest.xml` と名付けられなければならない。マニフェストのコピーは、[例 22](#) の「Change Dump」 (`http://example.com/20130102-changedump-manifest.xml`) の中の関係タイプ `contents` を持つ任意の `<rs:ln>` 要素によって示される場所に提供される場合があります。マニフェストは、[例 21](#) の閉じられた「Change List」で伝えられたのと同じ變更を含んでいます。リソース `http://example.com/res7.html` は 2 回リストされており、1 回は作成されたため、もう 1 回は更

新されたためとなっています。両者のエントリの URI は同じです。このマニフェストが含まれる ZIP パッケージには、このリソースの 2 つのビットストリームがあり、パッケージ内の異なるパスで利用可能です。

```
<?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="changedump-manifest"
    from="2013-01-02T00:00:00Z"
    until="2013-01-03T00:00:00Z"/>
  <url>
    <loc>http://example.com/res7.html</loc>
    <lastmod>2013-01-02T12:00:00Z</lastmod>
    <rs:md change="created"
      datetime="2013-01-02T12:00:00Z"
      hash="md5:1c1b0e264fa9b7e1e9aa6f9db8d6362b"
      length="4339"
      type="text/html"
      path="/changes/res7.html"/>
  </url>
  <url>
    <loc>http://example.com/res9.pdf</loc>
    <lastmod>2013-01-02T13:00:00Z</lastmod>
    <rs:md change="updated"
      datetime="2013-01-02T13:00:00Z"
      hash="md5:f906610c3d4aa745cb2b986f25b37c5a"
      length="38297"
      type="application/pdf"
      path="/changes/res9.pdf"/>
  </url>
  <url>
    <loc>http://example.com/res5.tiff</loc>
    <rs:md change="deleted"
      datetime="2013-01-02T19:00:00Z"/>
  </url>
  <url>
    <loc>http://example.com/res7.html</loc>
    <lastmod>2013-01-02T20:00:00Z</lastmod>
    <rs:md change="updated"
      datetime="2013-01-02T20:00:00Z"
      hash="md5:0988647082c8bc51778894a48ec3b576"
      length="5426"
      type="text/html"
      path="/changes/res7-v2.html"/>
  </url>
</urlset>
```

例 23. 「Change Dump Manifest」

14. 関連リソースへのリンク

14.1 概要

同期の対象となるリソースのコンテンツを取得したり、それに関する追加情報を提供したりするための代替アプローチを促進するために、ソースは、そのリソースから関連リソースへのリンクを提供することができます。そのようなリンクは、「Resource List」、「Resource Dump Manifest」、「Change List」および「Change Dump Manifest」で発生する場合があります。本項の残りの部分では、次のケースを考察し、(「Change List」の例の中で) 詳述します。

- ソースが、ミラーロケーションでリソース・コンテンツを利用できるようにする。
- ソースが、リソースの代替表現を提供する。
- ソースに、リソースの現在のバージョンと旧バージョンの違いが詳しく記載される。
- ソースが、リソースおよびリソースに関するメタデータを同期に使用できるようにする。
- ソースが、リソースの旧バージョンへのアクセスを提供する。
- ソースが、リソースに関するコレクション・メンバーシップ情報を提供する。
- デスティネーションが、ソースから取得したリソースを再公開し、その再公開したリソースを同期に使用できるようにし(つまり、ソース自体として機能する)、再公開した元のリソースにリンクする。

これまでのように、`<url>`の`<loc>`子要素は、同期の対象となるリソースの URI を伝えます。関連するリソースは、`<url>`の`<rs:ln>`子要素によって提供されます。`<rs:ln>`の可能な属性、および、前述のユースケースに対処するために使用されるリンク関係タイプについては、[7 項](#)で詳述しています。リストされているもの以外のニーズを満たすためのリンクを提供してもよく、また、適切な関係タイプが [IANA Link Relation Type Registry](#) から選択されるか、[RFC 5988, 4.2 項](#)で指定された URI として表現されても構いません。

デスティネーションは、`<rs:ln>`要素で伝達される情報を適切に解釈できない場合には、関連するリソースへのアクセスを控え、`<loc>`で提供される URI を使用してリソースを取得する必要があります。

14.2 ミラーリングされたコンテンツ

主要なアクセス・メカニズムの負荷を軽減するために、ソースはリソースの 1 つまたはそれ以上のミラーロケーションを伝達することができます。`<rs:ln>`要素は、リソースの各ミラーロケーションを表現するために導入されます。この要素には以下の属性があります。

- `duplicate` の値を持つ必須の `rel` 属性。
- ミラーリングされたリソースの URI を伝える必須の `href` 属性。
- 複数のミラーロケーション間の優先順位を表す任意の `pri` 属性。各ロケーションは個別の `<rs:ln>`要素によって表されます。`pri` の使用については、[7 項](#)で詳述しています。
- その他の属性は、[7 項](#)で`<url>`の`<rs:ln>`子要素について説明したとおり。

例 24 は、ソースがリソースの優先されるミラーロケーションに関する情報をどのように伝えるかを示しています。<rs:ln>要素によって伝えられる 3 つの場所が、<loc>で指定されたリソースの複製を指しているため、<rs:md>の各属性の値は、リソースとそのミラーとで同一であることが予想されます。そのため、それらは<rs:ln>要素から省略される必要があります。最後の<rs:ln>要素は、URI スキームから分かるように、HTTP 以外のプロトコルを介してリソースにアクセスできるミラーロケーションを指します。リソースが重複していても、最終更新日時は異なる場合があります。

```
<?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="changelist"
    from="2013-01-03T00:00:00Z"/>
  <url>
    <loc>http://example.com/res1</loc>
    <lastmod>2013-01-03T18:00:00Z</lastmod>
    <rs:md change="updated"
      hash="md5:1584abdf8ebdc9802ac0c6a7402c03b6"
      length="8876"
      type="text/html"/>
    <rs:ln rel="duplicate"
      pri="1"
      href="http://mirror1.example.com/res1"
      modified="2013-01-03T18:00:00Z"/>
    <rs:ln rel="duplicate"
      pri="2"
      href="http://mirror2.example.com/res1"
      modified="2013-01-03T18:00:00Z"/>
    <rs:ln rel="duplicate"
      pri="3"
      href="gsiftp://gridftp.example.com/res1"
      modified="2013-01-03T18:00:23Z"/>
  </url>
</urlset>
```

例 24. ミラーリングされたコンテンツ

14.3 代替表現

リソースには、異なる URI から利用可能な複数の表現があります。たとえば、リソースは `http://example.com/res1` のような汎用 URI で識別できます。サーバーとコンテンツ・ネゴシエーションを実行した後、クライアントは、たとえば、特定の URI

(`http://example.com/res1.html`) から利用可能なリソースの HTML 表現を取得できます。別のクライアントは、特定の URI (`http://example.com/res1.pdf`) からリソースの PDF 表現を要求して取得することもできます。クライアントがどの表現を取得するかは、メディアの種類と言語、地理的な場所、および、デバイスの種類などに関するクライアントの選択によって決まります。

ソースは、<loc>で汎用 URI を伝達することにより、リソースが同期の対象であることを表現できます。その場合、ソースが公表を希望する代替表現ごとに、<rs:ln>要素が導入されます。この要素には以下の属性があります。

- `alternate` の値を持つ必須の `rel` 属性。

- リソースの代替表現の特定の URI を伝える必須の `href` 属性。
- 代替表現のメディアタイプを伝える推奨される `type` 属性。
- [7 項](#)の`<url>`の`<rs:ln>`子要素についての説明のような他の属性。

リソースの汎用 URI がなく、特定の URI のみが存在するケースもあります。これは、たとえば、リソースがさまざまなデバイスで利用可能なさまざまな表現を持っている場合に発生する場合があります。その場合、`<loc>`の URI は特定の URI になり、`alternate` 関係タイプを持つ`<rs:ln>`要素は、その他の特定の URI から利用可能な代替表現を参照するために引き続き使用されます。

[例 25](#) は、`<loc>` で汎用 URI を記述する方法を示していますが、たとえば、コンテンツ・ネゴシエーションなど、特定の URI から利用可能な代替表現も指しています。

```
<?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="changelist"
    from="2013-01-03T11:00:00Z"/>
  <url>
    <loc>http://example.com/res1</loc>
    <rs:md change="updated"
      datetime="2013-01-03T18:00:00Z"/>
    <rs:ln rel="alternate"
      href="http://example.com/res1.html"
      type="text/html"/>
    <rs:ln rel="alternate"
      href="http://example.com/res1.pdf"
      type="application/pdf"/>
  </url>
</urlset>
```

例 25. 汎用 URI と特定の URI による代替表現

特定の表現が同期の対象と見なされる場合、その特定の URI は`<loc>`で提供されます。関連する汎用 URI (存在する場合は)、`<rs:ln>`要素を使用して提供できます。この要素には以下の属性があります。

- `canonical` の値を持つ必須の `rel` 属性。
- `<loc>`で提供される特定の URI に関連付けられた汎用 URI を伝える必須の `href` 属性。
- その他の属性は、[7 項](#)の`<url>`の`<rs:ln>`子要素について説明したとおり。

このアプローチは、ZIP ファイルに含まれるビットストリームを記述する「Resource Dump Manifest」および「Change Dump Manifest」に最適といえるでしょう。

例 26 は、`<loc>` の特定の URI を示すと同時に、`<rs:ln>` 要素を使用してリソースの汎用 URI を指すソースを示しています。その特定の URI から利用可能な表現に関するメタデータは、`<rs:md>` 要素の属性によって伝達されます。

```
<?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="changelist"
    from="2013-01-03T00:00:00Z"/>
  <url>
    <loc>http://example.com/res1.html</loc>
    <rs:md change="updated"
      datetime="2013-01-03T18:00:00Z"
      hash="md5:1584abdf8ebdc9802ac0c6a7402c03b6"
      length="8876"/>
    <rs:ln rel="canonical"
      href="http://example.com/res1"
      modified="2013-01-03T18:00:00Z"/>
  </url>
</urlset>
```

例 26. 特定の URI と汎用 URI による代替表現

14.4 コンテンツへのパッチの適用

リソースを更新する効率を高めるために、ソースは、変更されたリソース全体に加えて、リソースが受けた変更の記述を利用可能にすることができます。特に、頻繁な小さな変更、および／または、大きなリソースへの変更が関わる場合には、そのようなアプローチが魅力的でしょう。ただし、それには、デスティネーションが変更の記述を利用して旧バージョンに適切にパッチを適用することで、リソースの最新バージョンを構築することができるように、変更を明確に記述する方法が必要になります。

ソースは、通常どおり `<loc>` でリソースの URI を提供し、以下の属性を持つ `<rs:ln>` 要素を導入することにより、リソース変更の記述を利用可能にする旨を表現することができます。

- 値が <http://www.openarchives.org/rs/terms/patch> の必須の `rel` 属性。
- リソース変更の記述の URI を伝える必須の `href` 属性。
- 変更の記述のメディアタイプを伝える必須の `type` 属性。そのメディアタイプは、リソースの旧バージョンに記述された変更を明確に適用することで、現在のリソースを構築できるようにするものでなければなりません。
- その他の属性は、[7 項](#) の `<url>` の `<rs:ln>` 子要素について説明したとおり。

例 27 は、[JSON Patch](#) (JSON パッチ) で導入された `application/json-patch` メディアタイプを使用して表現された JSON リソースが受けた変更を表現するソースを示しています。また、ソースが、たとえば、コミュニティ仕様に記述された実験的なメディアタイプを使用して、大きな TIFF ファイルへの変更を伝達していることも示しています。メディアタイプを理解しないデスティネーションは、変更の記述を無視し、`<loc>` の URI を使用してリソースの最新バージョンを取得する必要があります。

す。XML ドキュメントへの変更を表現するための明確に指定されたメディアタイプの別の例は、[RFC 5261](#) に指定されている `application/patch-ops-error+xml` です。

このようにリソースの変更を表現する方法は、「Change List」([例 27](#) のように) および「Change Dump」にのみ適用できます。この方法を「Change Dump」で行う場合には、「Change Dump Manifest」のエントリには、コンテンツ・パッケージに含まれる変更の記述を指す `<rs:ln>` 要素の `path` 属性がなければなりません。

```
<?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="changelist"
    from="2013-01-03T00:00:00Z"/>
  <url>
    <loc>http://example.com/res4</loc>
    <rs:md change="updated"
      hash="sha-256:f40xZX_x_DFGFDgghgdfb6rtSx-iosjyf6735432nklj"
      length="56778"
      type="application/json"/>
    <rs:ln rel="http://www.openarchives.org/rs/terms/patch"
      href="http://example.com/res4-json-patch"
      hash="sha-256:y66dER_t_HWEIKpesdkeb7rtSc-ippjf9823742oplD"
      length="73"
      type="application/json-patch"/>
  </url>
  <url>
    <loc>http://example.com/res5-full.tiff</loc>
    <rs:md change="updated"
      hash="sha-256:f40xZX_x_FO5LcGBSKHWXfwtSx-j1ncoSt3SABJtkGk"
      length="9788456778"
      type="image/tiff"/>
    <rs:ln rel="http://www.openarchives.org/rs/terms/patch"
      href="http://example.com/res5-diff"
      hash="sha-256:h986gT_t_87HTkjHYE76G558hY-jdfgy76t55sadJUyT"
      length="4533"
      type="application/x-tiff-diff"/>
  </url>
</urlset>
```

例 27. リソースにパッチを適用する方法を詳述するドキュメントへのリンクを含む「Change List」

14.5 リソースとリソースに関するメタデータ

リソースとリソースに関するメタデータの両方を同期する必要がある場合があります。ResourceSync の観点に立てば、リソースとそれに関するメタデータは、その両方が、同期の対象となる個別の URI を持つリソースと見なされます。これまでのように、それぞれが個別の `<url>` ブロックを取得し、また、各 URI はそれぞれのブロックの `<loc>` 子要素で伝達されます。必要に応じて、リソース間の相互関係は、適切な関係タイプが各ブロックに追加された `<rs:ln>` 要素によって表現されます。`<rs:ln>` 要素には以下の属性があります。

- 必須の `rel` 属性。リソースからそれを記述するメタデータを指す場合、`rel` 属性の値は `describedby` です。メタデータからメタデータで記述されたリソースを指す場合、その値は `describes` です。
- 必須の `href` 属性。リソースからそれを記述するメタデータを指す場合、`href` 属性の値はメタデータリソースの URI です。メタデータからメタデータで記述されたリソースを指す場合、値は記述されたリソースの URI です。
- その他の属性は、[7項](#)の<url>の<rs:ln>子要素について説明したとおり。

例 28 は、ソースがどのようにこの 2 つのリソース間の相互関係を表現するかを示しています。デスティネーションは、メタデータベースの選択基準を満たすリソースと同期するためにのみ、リソースをフィルタリング・メカニズムとして記述するメタデータを使用することに注意してください。メタデータレコードの更新を伝える<url>要素の中で、リソースを記述するために使用されるメタデータの種類を表現するために `profile` 関係タイプ [[RFC 6906](#)] を持つリンクが使用されることにも注意してください(この場合、この種のメタデータは XML 名前空間を使用して表現されます)。リンクは、デスティネーションがメタデータ情報を解釈するサポートになる URI を提供する必要があります。たとえば、名前空間、XML スキーマ、または、MARC の記述を参照することができます。

```
<?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="changelist"
    from="2013-01-03T00:00:00Z"/>
  <url>
    <loc>http://example.com/res2.pdf</loc>
    <rs:md change="updated"
      datetime="2013-01-03T18:00:00Z"
      hash="md5:1584abdf8ebdc9802ac0c6a7402c03b6"
      length="8876"
      type="application/pdf"/>
    <rs:ln rel="describedby"
      href="http://example.com/res2_dublin-core_metadata.xml"
      type="application/xml"/>
  </url>
  <url>
    <loc>http://example.com/res2_dublin-core_metadata.xml</loc>
    <rs:md change="updated"
      datetime="2013-01-03T19:00:00Z"
      type="application/xml"/>
    <rs:ln rel="describes"
      href="http://example.com/res2.pdf"
      hash="md5:1584abdf8ebdc9802ac0c6a7402c03b6"
      length="8876"
      type="application/pdf"/>
    <rs:ln rel="profile"
      href="http://purl.org/dc/elements/1.1/">
  </url>
</urlset>
```

例 28. 「Change List」内のリソースとリソースに関するメタデータ間のリンク

14.6 リソースの旧バージョン

ソースは、デスティネーションが最新バージョンと単に同期しているだけでなく、履歴の観点を持つことができるようにするために、リソースの旧バージョンへのアクセスを提供することができます。このアプローチには、以下で構成される一般的なリソース・バージョン管理パラダイムを活用します。

- いつでもリソースの現在の表現が利用可能な、時間汎用 (time-generic) URI の存在。
- 特定の時間のリソースのバージョンが利用可能な、各リソース・バージョンの時間固有 (time-specific) URI の存在。

リソースについて通信する場合、時間汎用 URI は `<loc>` 要素で提供されます。`<lastmod>` 要素は、リソースの最終更新時刻を提供するために使用することができます。

最初のアプローチは、`<lastmod>` 要素で指定されるように、最後の変更時刻に対応するリソースの時間固有 URI を伝達することです。これは、以下の属性を持つ単一の `<rs:ln>` 要素を導入することにより実現されます。

- `memento` の値を持つ必須の `rel` 属性。
- 最終変更時刻に対応するリソースの時刻固有 URI を伝える必須の `href` 属性。この URI を使用すると、デスティネーションは、たとえば、オフライン中にリソースが再び変更された場合でも、キャッチアップ操作中にその特定のバージョンを取得することができます。
- 時間固有 URI で識別されるリソース・バージョンの最終変更日時は、`modified` 属性の値によって伝達されます。
- その他の属性は、[7 項](#)の `<url>` の `<rs:ln>` 子要素について説明したとおり。

2 番目のアプローチは、時間汎用リソースに関連付けられた TimeGate を指すことです。TimeGate は、Memento プロトコル [[RFC 7089](#)] で導入された日時ディメンションでのネゴシエーションをサポートし、指定された瞬間に存在するリソースのバージョンを取得します。これにより、デスティネーションは、過去の 1 点、たとえば、最終変更時刻 ([例 29](#) に示すように、`<lastmod>` 値が含まれている場合)、または、その他の任意の時刻に存在したバージョンを取得することができます。TimeGate へのポインタは、以下の属性を持つ `<rs:ln>` 要素を使用して導入されます。

- `timegate` の値を持つ必須の `rel` 属性。
- 時間汎用リソースに関連付けられた TimeGate の URI を伝える必須の `href` 属性。
- [7 項](#)の `<url>` の `<rs:ln>` 子要素で説明されたその他の属性は、TimeGates では無意味なため、使用すべきではありません。

3 番目のアプローチは、時間汎用リソースに関連付けられた TimeMap を指すことです。Memento プロトコル [[RFC 7089](#)] で導入された TimeMap により、デスティネーションは、サーバーが認識するすべての時間固有リソースの包括的なリストを取得できます。これにより、デスティネーションはそのリストの中からリソースの特定のバージョンを選択することができます。TimeMap へのポインタは、以下の属性を持つ `<rs:ln>` 要素を使用して導入されます。

- `timemap` の値を持つ必須の `rel` 属性。
- 時間汎用リソースに関連付けられた TimeMap の URI を伝える必須の `href` 属性。
- その他の属性は、[7 項](#)の `<url>` の `<rs:ln>` 子要素について説明したとおり。

例 29 は、リソースの旧バージョンへのリンク、TimeGate へのリンク、および、TimeMap へのリンクを含む「Change List」を示しています。hash、length および type 属性の値が、<rs:md> 要素と旧バージョンを指す<rs:ln>子要素とで同じ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:ln rel="up"
    href="http://example.com/dataset1/capabilitylist.xml"/>
  <rs:md capability="changelist"
    from="2013-01-03T00:00:00Z"/>
  <url>
    <loc>http://example.com/res1</loc>
    <lastmod>2013-01-03T18:00:00Z</lastmod>
    <rs:md hash="md5:1584abdf8ebdc9802ac0c6a7402c03b6"
      length="8876"
      type="text/html"
      change="updated"/>
    <rs:ln rel="memento"
      href="http://example.com/20130103070000/res1"
      modified="2013-01-02T18:00:00Z"
      hash="md5:1584abdf8ebdc9802ac0c6a7402c03b6"
      length="8876"
      type="text/html"/>
    <rs:ln rel="timegate"
      href="http://example.com/timegate/http://example.com/res1"/>
    <rs:ln rel="timemap"
      href="http://example.com/timemap/http://example.com/res1"
      type="application/link-format"/>
  </url>
</urlset>
```

例 29. リソース・バージョン、および Memento の TimeGate と TimeMap へのリンク

14.7 コレクション・メンバーシップ

ソースは、リソースが [OAI-ORE](#) アグリゲーション (Aggregation) や [OAI-PMH](#) セットなどのコレクションのメンバーであることを表現できます。ソースは、通常どおり、そのリソースの URI を<loc>で提供し、以下の属性を持つ<rs:ln>要素を導入することにより、同期の対象となるリソースのコレクション・メンバーシップを表現できます。

- collection の値を持つ必須の rel 属性。
- コレクションを識別する URI を伝える必須の href 属性。
- その他の属性は、[7 項](#)の<url>の<rs:ln>子要素について説明したとおり。

例 30 は、OAI-ORE アグリゲーションのメンバーである 1 つのリソースを持つ「Change List」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="changelist"
    from="2013-01-03T00:00:00Z"/>
  <url>
    <loc>http://example.com/res1</loc>
    <rs:md hash="md5:1584abdf8ebdc9802ac0c6a7402c03b6"
      length="8876"
      type="text/html"
      change="updated"/>
    <rs:ln rel="collection"
      href="http://example.com/aggregation/0601007"/>
  </url>
</urlset>
```

例 30. コレクションのメンバーとしてのリソース

14.8 リソースの再公開

これ以後アグリゲーターと呼ばれる特別な種類のデスティネーションは、ソースからコンテンツを取得して再公開し、その結果、再公開されたコンテンツのソースとして機能することができます。このようなアグリゲーターのシナリオでは、アグリゲーターと同期するデスティネーションがコンテンツの出所を理解し、その正確性をコンテンツの元のソースと検証できることが重要になることもあります。再公開されたリソースについて通信する場合には、アグリゲーターは以下の属性を持つ<rs:ln>要素を導入することにより、そのような出所情報を提供することができます。

- via の値を持つ必須の rel 属性。
- 元のソースのリソースの URI を伝える必須の href 属性。
- その他の属性は、[7 項](#)の<url>の<rs:ln>子要素について説明したとおり。

例 31 は、ソースが単一のリソースへの変更に関する情報を公開する「Change List」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="changelist"
    from="2013-01-03T00:00:00Z"/>
  <url>
    <loc>http://original.example.com/res1.html</loc>
    <lastmod>2013-01-03T07:00:00Z</lastmod>
    <rs:md hash="md5:1584abdf8ebdc9802ac0c6a7402c03b6"
      length="8876"
      type="text/html"
      change="updated"/>
  </url>
</urlset>
```

例 31. ソースが公開するオリジナル

例 32 は、元のソースのリソースを参照する第 1 アグリゲーター (primary aggregator) の「Change List」を示しています。ここには、リソースの起源に関する情報を伝達する href などの属性を持つ関係タイプ via とのリンクが含まれます。この情報は、**例 31** に示す「Change List」の <url> ブロックで提供されるデータに対応しています。たとえば、**例 32** の href 属性の値は、**例 31** の <url> ブロックの <loc> 子要素の値と同等です。

```
<?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://aggregator1.example.com/dataset1/capabilitylist.xml"/>
  <rs:md capability="changelist"
    from="2013-01-03T11:00:00Z"/>
  <url>
    <loc>http://aggregator1.example.com/res1.html</loc>
    <lastmod>2013-01-03T20:00:00Z</lastmod>
    <rs:md hash="md5:1584abdf8ebdc9802ac0c6a7402c03b6"
      length="8876"
      type="text/html"
      change="updated"/>
    <rs:ln rel="via"
      href="http://original.example.com/res1.html"
      modified="2013-01-03T07:00:00Z"
      hash="md5:1584abdf8ebdc9802ac0c6a7402c03b6"
      length="8876"
      type="text/html"/>
  </url>
</urlset>
```

例 32. 第 1 アグリゲーターによる再公開

第 2 アグリゲーター (secondary aggregator) が、第 1 アグリゲーターの「Change List」を使用して変更されたリソースを取得し、その「Change List」を再公開すると、アグリゲーション

(aggregations)のチェーンが作成されます。この場合、各アグリゲーターは、リソースの起源に関する情報を伝えるために、既存の `via` リンクのみを維持する必要があります。

[例 33](#) は、変更されたリソースと[例 32](#)と同じ `via` リンクに関する情報を含む、第 2 アグリゲーターによる「Change List」を示しています。このリンクで伝達されるデータは、[例 31](#)の`<url>`ブロックで提供されるデータに対応しています。

```
<?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://aggregator2.example.com/dataset1/capabilitylist.xml"/>
  <rs:md capability="changelist"
    from="2013-01-03T12:00:00Z"/>
  <url>
    <loc>http://aggregator2.example.com/res1.html</loc>
    <lastmod>2013-01-04T09:00:00Z</lastmod>
    <rs:md hash="md5:1584abdf8ebdc9802ac0c6a7402c03b6"
      length="8876"
      type="text/html"
      change="updated"/>
    <rs:ln rel="via"
      href="http://original.example.com/res1.html"
      modified="2013-01-03T07:00:00Z"
      hash="md5:1584abdf8ebdc9802ac0c6a7402c03b6"
      length="8876"
      type="text/html"/>
  </url>
</urlset>
```

例 33. 第 2 アグリゲーターによる再公開

`at`、`completed`、`from` および `until` 属性の値は、常に、それらを含むドキュメントを公開するソースの観点から表現されなければなりません。そのため、「Change List」の `from` 日時が、`via` 関係タイプを持つ`<rs:ln>`リンクを使って伝えられた「Change List」に記述された、元のソースのリソースの`<lastmod>`日時より新しい可能性もあります。

アグリゲーターが、アグリゲーション・チェーンでリンクの前にあるソースから、`via` 関係タイプを持つリンク以外のリンクを継承する場合には、注意が必要です。そのようなリンクは、それぞれの独自の観点から適切なままとし、そうでない場合には、継承しないようにする必要があります。たとえば、元のソースで表現された関係タイプが `collection` または `canonical` のリンクは、アグリゲーターのコピーのコンテキストでは適切でない可能性があるため、アグリゲーターの機能ドキュメントの変更されたリソースの記述に含めるべきではありません。

A. 時間属性の要件

(規範的)

表 4 に、ResourceSync ドキュメントの `at` および `from` 属性を使用するための要件の概要を示します。列見出しのトップのラベルは、インデックス・ドキュメントの `<sitemapindex>` ルート要素、および、その他のすべてのドキュメントの `<urlset>` ルート要素を表しています。列見出しの子ラベルは、インデックス・ドキュメントの `<sitemap>` 子要素、および、その他のすべてのドキュメントの `<url>` 子要素を表しています。

任意の `completed` および `until` 属性は表に示されていませんが、対応する `at` および `from` 属性が必須、推奨または任意の場合には、いつでも追加することができます。

表 4 は、たとえば、「Change List」に含まれる時区間を伝えるために、「Change List」に属性 `from` を持つ `<urlset>` ルート要素の `<rs:md>` 子要素が含まれなければならないことを示しています。

機能ドキュメント	<code>/top/rs:md/@at</code>	<code>/top/rs:md/@from</code>	<code>/top/child/rs:md/@at</code>	<code>/top/child/rs:md/@from</code>
Resource List	必須	X	X	X
Resource List Index	必須	X	任意	X
Resource Dump	必須	X	任意	X
Resource Dump Index	必須	X	任意	X
Resource Dump Manifest	必須	X	X	X
Change List	X	必須	X	X
Change List Index	X	必須	X	推奨
Change Dump	X	必須	X	推奨
Change Dump Index	X	必須	X	推奨
Change Dump Manifest	X	必須	X	X

表 4. ResourceSync ドキュメントの `at` および `from` 属性の必須および任意の使用

B. 参考文献一覧

(この参考文献一覧は、ResourceSync フレームワーク仕様、ANSI/NISO Z39.99-2017 の一部ではありません。情報提供のみを目的として含めています。)

[HTML Links]

Raggett, Dave, Arnaud Le Hors, and Ian Jacobs, eds. [*"Links." Section 12 in: HTML 4.01 Specification.*](#) W3C Recommendation. World Wide Web Consortium, December 24, 1999. Available at: <http://www.w3.org/TR/html401/struct/links.html>

[JSON-Patch]

Bryan, P., and M. Nottingham, eds. [*JSON Patch.*](#) Internet Draft. Internet Engineering Task Force (IETF), January 20, 2013. Available at: <http://tools.ietf.org/html/draft-ietf-appsawg-json-patch-10>

[OAI-PMH]

Lagoze, Carl, Herbert Van de Sompel, Michael Nelson, and Simeon Warner, eds. [*The Open Archives Initiative Protocol for Metadata Harvesting.*](#) Version 2.0. Open Archives Initiative, December 7, 2008. Available at: <http://www.openarchives.org/OAI/openarchivesprotocol.html>

[ORE]

Lagoze, Carl, Herbert Van de Sompel, Pete Johnston, Michael Nelson, Robert Sanderson, and Simeon Warner, eds. [*ORE Specification - Abstract Data Model.*](#) Open Archives Initiative, October 17, 2008. Available at: <http://www.openarchives.org/ore/1.0/datamodel>

[Relation Types]

Klein, Martin, Herbert Van de Sompel, Simeon Warner, Graham Klyne, Bernhard Haslhofer, Michael Nelson, Carl Lagoze, and Robert Sanderson, eds. [*Relation Types Used in the ResourceSync Framework.*](#) Open Archives Initiative. Available at: <http://www.openarchives.org/rs/relationtypes>

[RFC 5261]

Urpalainen, J. [*An Extensible Markup Language \(XML\) Patch Operations Framework Utilizing XML Path Language \(XPath\) Selectors. RFC 5261.*](#) Internet Engineering Task Force (IETF), September 2008. Available at: <http://www.ietf.org/rfc/rfc5261.txt>

[RFC 5785]

Nottingham, M., and E. Hammer-Lahav. [*Defining Well-Known Uniform Resource Identifiers \(URIs\). RFC 5785.*](#) Internet Engineering Task Force (IETF), April 2010. Available at: <http://www.ietf.org/rfc/rfc5785.txt>

[RFC 6596]

Ohye, M., and J. Kupke. [*The Canonical Link Relation. RFC 6596.*](#) Internet Engineering Task Force (IETF), April 2012. Available at: <http://www.ietf.org/rfc/rfc6596.txt>

[rsync]

[*rsync.*](#) From Wikipedia, the free encyclopedia. Available at: <http://en.wikipedia.org/wiki/Rsync>

[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/>

[XHTML Links]

McCarron, Shane, et al., eds. ["Link Module." Section 5.19 in: XHTML Modularization 1.1. Second Edition.](#) World Wide Web Consortium, July 29, 2010. Available at: http://www.w3.org/TR/xhtml-modularization/abstract_modules.html#s_linkmodule

C. 前付け、作成者および謝辞

(この付録には、本仕様の PDF バージョンからの前付け、作成者および謝辞が含まれています。本 Web ドキュメントの読みやすさを保つために、ここに移動したものです。)

NISO 規格について

NISO 規格は、米国情報標準化機構 (National Information Standards Organization) の作業グループによって開発されています。その開発プロセスは、各 NISO 投票メンバーやその他の利害関係者に対して開かれた、提案された標準規格の厳格なピアレビューを含む、厳しいプロセスです。標準規格の最終承認には、米国規格協会 (the American National Standards Institute) により、NISO が適正手続き、コンセンサス、および、その他の承認基準の要件を満たしているかどうかを検証するプロセスが含まれます。ひとたび検証され、承認されると、NISO 規格は米国規格にもなります。

これらの標準規格はいつでも改訂または撤回されることがあります。本規格のステータスに関する最新情報については、NISO オフィスに問い合わせるか、NISO の Web サイト (www.niso.org) にアクセスしてください。

出版元
NISO
3600 Clipper Mill Road
Suite 302
Baltimore, MD 21211
www.niso.org

ISSN: 1041-5653 (National Information standards series)

ISBN: 978-1-937522-74-2 (HTML)

ISBN: 978-1-937522-73-5 (PDF)

序文

(この序文は ResourceSync フレームワーク仕様、ANSI/NISO Z39.99-2017 の一部ではありません。情報提供のみを目的として含めています。)

本規格について

本ドキュメントの構造は以下のとおりです。

- [1 項](#)の「はじめに」では、目的と範囲について説明、ユースケースの例を示し、ResourceSync フレームワークのコアコンポーネントを例を使って紹介するウォークスルーが含まれています。
- [2 項](#)の「規範的参考文献」には、本標準への適合に必要とされる他の標準／仕様への参照がリストされています。
- [3 項](#)の「定義」では、本標準の理解のために重要な用語を紹介しています。
- [4 項](#)の「名前空間接頭辞バインディング」では、本仕様に使用される名前空間を識別しています。
- [5 項](#)の「同期処理」では、さまざまな ResourceSync 機能の高レベルの概要を示し、これらがどのようにリソースの変更との一致を保つことを目的とした処理に適合するかを示しています。
- [6 項](#)の「フレームワークの構成」では、ResourceSync フレームワークの全体的な構成(構造、ナビゲーションおよび検出)について説明しています。
- [7 項](#)の「サイトマップ・ドキュメントの形式」では、サイトマップ・プロトコルによって導入されたドキュメント形式を使用して、すべての ResourceSync 機能で同期の関連情報を伝達する方法について説明しています。
- [8 項](#)の「ソースの記述」、および、[9 項](#)の「公表機能」では、サーバーがサポートする ResourceSync 機能をどのように伝達するかについて説明しています。
- [10 項](#)の「リソースの記述」、[11 項](#)の「パッケージ・リソース」、[12 項](#)の「変更の記述」、[13 項](#)の「変更のパッケージ化」、および、[14 項](#)の「関連リソースへのリンク」では、それぞれ、リモートシステムがサーバーの進化するデータとの一致を保つことができるようにするために、サーバーが実装できる機能に関して詳細を提供しています。
- [付録 A](#)の「時間属性の要件」では、ResourceSync ドキュメントの `at` および `from` 属性の使用に関する要件の概要を示しています。

2017 年の改定

標準の改訂は、リソースの最終変更日付とリソースへの変更通知の日時の融合に関連する問題を修正するための変更を反映しています。

商標、サービスマーク

本標準で使用されている場合には、商標またはサービスマークであるすべての用語は、それぞれの所有者の財産です。

NISO 投票メンバー

以下は、本標準を承認した ANSI/NISO Z39.99-2017 の ResourceSync 投票プールのメンバーです。本標準の NISO 承認は、必ずしも投票プールのすべてのメンバーが承認に投票したことを意味するものではありません。

米国物理学会 (AIP)

Kevin Ruthen

米国図書館協会 (ALA)

Jill Emery

米国物理学協会 (APS)

Mark Doyle
米国情報科学技術協会 (ASIS&T)
Mark Needleman
研究図書館協会 (ARL)
Sue Baughman
Cenveo Publisher Services
Evan Owens
EBSCO Information Services
Oliver Pesch
Ex Libris, Inc.
Mike Dicus
Index Data
Nassib Nassar
Innovative Interfaces, Inc.
Tim Auger
ITHAKA/JSTOR/Portico
Amy Kirchhoff
米国議会図書館
Sally McCallum
ロスアラモス国立研究所
Dianna Magnoni
Minitex
Paul Swanson
米国現代語学文学協会
Gregory Grazevich
アメリカ音楽図書館協会
Nara Newcomer
北米逐次刊行物分科会 (NASIG)
Beverly Geckle
全米高度情報サービス連合 (NFAIS)
Gregory Grazevich
フィンランド国立図書館
Juha Hakala
国立医学図書館 (NLM)
Barbara Rapp
ProQuest
Carol Brent
カリフォルニア州電子図書館コンソーシアム (SCELC)
Rick Burke
Scholarly iQ
Gary Van Overborg
The Library Corporation (TLC)
Sam Brenizer

NISO D2D 議題委員会 (Topic Committee)

本標準を承認した時点で、以下の方々が Discovery to Delivery (D2D) 議題委員会に参加しました。

Kristin Antelman
カリフォルニア工科大学
Scott Bernier
EBSCO Information Services
Christine Stohn
Ex Libris, Inc.
Lucy Harrison
GeorgiA 図書館ラーニングオンライン (GALILEO)
Julie Zhu
IEEE
Peter Murray
Index Data
John Dove
Paloma & Associates
Pascal Calarco
ウインザー大学、Leddy 図書館

NISO ResourceSync 作業グループメンバー

本標準を開発および承認した NISO ResourceSync ワーキンググループには、以下の方々が参加しました。

Todd Carpenter、共同議長
NISO
Michael Nelson*
オールドドミニオン大学
Bernhard Haslhofer*
ウィーン工科大学
Shlomo Sanders
Ex Libris, Inc.
Richard Jones
Cottage Labs
Robert Sanderson*
ロスアラモス国立研究所
Martin Klein*
ロスアラモス国立研究所
Herbert Van de Sompel、共同議長*
ロスアラモス国立研究所
Graham Klyne*
オックスフォード大学
Paul Walk
EDINA
Carl Lagoze*
ミシガン大学

Simeon Warner*
コーネル大学
Stuart Lewis
Jisc Collections
Zhiwu Xie
バージニア工科大学図書館
Peter Murray
Lyris
Jeff Young
OCLC オンラインコンピュータ図書館センター

*2017 年の更新を担当。

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。Martin Hays (カリフォルニアデジタル図書館)、David Rosenthal (スタンフォード大学図書館)、Ed Summers (米国議会図書館)、Vincent Wehren (Microsoft) など、多くの個人の貢献者にも感謝します。

D. 変更ログ

(この付録は、ResourceSync フレームワーク仕様、ANSI/NISO Z39.99-2017 の一部ではありません。情報提供のみを目的として含めています。)

日付	編集者	記述
2017-02-02	martin, herbert, simeon	バージョン 1.1, ANSI/NISO Z39.99-2017
2016-08-26	martin, herbert, simeon	バージョン 1.0.9, DRAFT REVISION
2014-04-21	martin, herbert, simeon	バージョン 1.0, ANSI/NISO Z39.99-2014
2013-08-05	martin, herbert, rob, simeon	バージョン 0.9.1
2013-06-07	martin, herbert, rob, simeon	バージョン 0.9
2013-05-01	martin, herbert, rob, simeon	バージョン 0.6
2013-02-01	martin, herbert, rob, simeon	ベータ仕様ドラフト
2012-08-13	martin, herbert, simeon, bernhard	最初のアルファ仕様ドラフト



この著作物は、[Creative Commons Attribution-Share Alike 4.0 International License](#) (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。