

TR-1039

企業内
セッション開始プロトコル(SIP)
のための
プレゼンスイベントパッケージ
に関する技術資料

Technical Report on a Presence Event Package
for the Session Initiation Protocol (SIP)
in the Enterprise Network

第1版

2012年2月2日制定

一般社団法人
情報通信技術委員会

THE TELECOMMUNICATION TECHNOLOGY COMMITTEE

本書は、一般社団法人情報通信技術委員会が著作権を保有しています。
内容の一部又は全部を一般社団法人情報通信技術委員会の許諾を得ることなく複製、転載、
改変、転用及びネットワーク上での送信、配布を行うことを禁止します。

目次

<参考>	4
1. 概要	5
2. はじめに	5
3. 定義	5
4. 操作の概要	6
5. メッセージフロー例	9
6. プロトコル要素の定義	15
7. プレゼンス情報データフォーマット(PIDF)	16
8. プレゼンス情報の拡張	19
9. 規範となる参考文献	21
10. 有益な参考文献	22

<参考>

1. 国際勧告等の関連

本技術レポートは、The Internet Society において制定された「A Presence Event Package for Session Initiation Protocol(SIP)(RFC3856)」「Presence Information Data Format(PIDF)」「Session Initiation Protocol(SIP) Extension for Event State Publication」に準拠している。

2. 上記国際勧告等に対する追加項目等

なし

3. 改版の履歴

版数	制定日	改版内容
第1版	2012年 2月 2日	制定

4. 工業所有権

本標準に関わる「工業所有権等の実施の権利に係る確認書」の提出状況は、TTCホームページでご覧になれます。

5. 技術レポート作成部門

第1版 : 企業ネットワーク専門委員会

1. 概要

本文書では、プレゼンスの予約と通知に関するセッション開始プロトコル(Session Initiation Protocol/SIP)およびイベントフレームワーク内で使用される発行イベントステートの SIP 拡張の用法を説明する。プレゼンスは、ネットワーク上にいる他のユーザーと通信するユーザーの意思と能力が定義される。従来、プレゼンスは、「オンライン」と「オフライン」を示すものと限定されてきたが、本書で使用するプレゼンスの概念は、より広い。

プレゼンスの予約と通知は、通常の SIP イベント通知の枠組み内でイベントパッケージを定義することによって対応される。

本文書で説明される仕組みは、どんなイベントステートの発行にも対応するように拡張することができるが、その用途で他に適切なイベントパッケージが存在することがある。

本文書は、任意データを送信するための一般的な用途の仕組みを意図したものではない。その用途では、より適切な仕組みが存在する。

また、本文書では共通プレゼンスデータフォーマットとしてのプレゼンス情報データフォーマット(PIDF)についても説明を行う。

2. はじめに

プレゼンスサービスとは、プレゼンス情報を受け入れ、格納し、関係者(ウォッチャーと呼ばれる)に配信するシステムである。プレゼンスプロトコルは、インターネットまたは任意の IP ネットワーク上で、プレゼンスサービスを提供するプロトコルである。

本文書では、プレゼンスの一般的な SIP のイベント通知の枠組みを、RFC3856 および RFC3903 に基づいて、PUBLISH メソッド及び SUBSCRIBE メソッドと NOTIFY メソッドを利用して、例示する。

3. 定義

本文書は、RFC2778(参考文献[A-10])の、RFC 3265(参考文献[A-2])および RFC3261(参考文献[A-1])の定義に加え、本文書は、以下の新たな概念を取り入れ、新たに定義したり、より明確に説明を加える。

イベントステート(Event State): リソースのステート情報。イベントパッケージおよび Addresses-of-Record に関連付けられる。

ウォッチャー (Watcher) : PA に対して、プレゼンティティのプレゼンス情報を要求する。

プレゼンスユーザーエージェント(Presence User Agent/PUA): プレゼンスユーザーエージェントは、プレゼンティティのプレゼンス情報を操作する。この操作は、他の何らかのアクション(新規の Contact を追加する SIP REGISTER リクエストの送信など)によって引き起こされたり、プレゼンスドキュメントの発行によって明示的に実行される可能性がある。本文書では、1つのプレゼンティティに付き複数の PUA ということを示唆する。つまり、1人のユーザーは、多くのデバイス(携帯電話、PDA (Personal Digital Assistant)など)を持つことができ、それぞれのデバイスが、別々にプレゼンティティのプレゼンス情報全体のコンポーネントを生成している、ということの意味する。PUA は、プレゼンスシステムにデータをプッシュするが、システムの外部にあり、SUBSCRIBE メッセージの受信や NOTIFY メッセージの送信はシステム内で行わない。

イベント発行エージェント(Event Publication Agent/EPA): イベントステートを公開するときに PUBLISH リクエストを発行するユーザーエージェントクライアント(User Agent Client/UAC)。通常のプレゼンスユーザーエージェント(Presence User Agent/PUA)の役割を持つ。

プレゼンスエージェント(Presence Agent/PA): プレゼンスエージェントは、SUBSCRIBE リクエストの受信、SUBSCRIBE リクエストへの応答、プレゼンスステートの変更時に通知の生成、という機能を持つ SIP ユーザーエージェントである。プレゼンスエージェントは、プレゼンティティのプレゼンスステートを知っている必要がある。つまり、プレゼンスエージェントは、プレゼンティティの PUA が操作するプレゼンスデータに対して、アクセス権を持つ必要がある。一例として、PA をプロキシ/レジストラと同じ場所に置くという方法がある。他にも、PA をプレゼンティティのプレゼンスユーザーエージェントと同じ場所に置く方法がある。ただし、実現方法はこの2つだけではない。また、本仕様では、PA 機能を配置する場所に関して何ら推奨しない。PA は、プレゼンティティをユニークに識別する SIP URI で常にアドレス指定できる(言い換えると、sip:joe@example.com)。1つのプレゼンティティに対して複数の PA が存在する可能性がある。また、各 PA は、プレゼンティティの現在アクティブなサブスクリプション全体のうち、一部を操作する。また、PA は、プレ

ゼンスイベントパッケージに対応する通知者(RFC 3265(参考文献 [A-2])で定義)でもある。

イベントステートコンポジタ(Event State Compositor/ESC) : PUBLISH リクエストを処理し、イベントステートを完全なリソースの複合イベントステート(composite event state)へと合成する役割を持つ、ユーザーエージェントサーバー(User Agent Server/UAS)。通常のプレゼンスエージェント(Presence Agent/PA)の役割を持つ。

プレゼンスサーバー(Presence Server): プレゼンスサーバーは、SUBSCRIBE リクエストに対して、プレゼンスエージェントまたはプロキシサーバーとして動作することができる物理エンティティである。PA として動作する場合、何らかのプロトコルを使用してプレゼンティティのプレゼンス情報を認識している。プロキシとして動作する場合、SUBSCRIBE リクエストは、PA として動作している可能性がある別のエンティティにプロキシされる。

エッジプレゼンスサーバー(Edge Presence Server): PUA と同じ場所にあるプレゼンスエージェント。プレゼンス情報を操作するエンティティと同居しているため、プレゼンティティのプレゼンス情報を認識している。

発行(Publication): イベントステートを発行するときに PUBLISH リクエストを ESC に送信する、EPA の動作。

イベントハードステート(Event Hard State): リソースの定常ステートまたはデフォルトのイベントステート。ソフトステートの発行がない場合、またはソフトステートの発行とは別に、ESC が使用できる。

イベントソフトステート(Event Soft State): PUBLISH メカニズムを使用する。EPA によって発行されるイベントステート。ESC で特定のソフトステートエンティティを識別するために、プロトコル要素(つまりエンティティタグ)が使用される。ソフトステートには、定義済みの生存時間があり、また、ネゴシエートされた時間後に期限切れとなる。

4. 操作の概要

このセクションでは、このイベントパッケージの操作概要を説明する。SIP イベントパッケージ(参考文献[A-2])と SIP 仕様(参考文献[A-1])について、あまり知識のない読者向けでもこのパッケージを理解しやすいように動作の概要を説明する。ただし、このパッケージの詳細な意味については、SIP イベントと SIP 仕様を理解する必要がある。

イベントステート発行のために PUBLISH を定義する。PUBLISH と REGISTER とは、ユーザーの代わりにステートを管理する他のエンティティにおいて、ユーザーがステートを作成、修正および削除できるようになる、という点で似ている。PUBLISH リクエストのアドレス指定は、SUBSCRIBE リクエストのアドレス指定と同じである。PUBLISH の Request-URI は、イベントステート発行先リソースのアドレスを設定する。言い替えると、ユーザーは、イベントステートを発行するユーザーエージェント(User Agent)またはエンドポイントを複数持つことができる。各エンドポイントは、イベントステートコンポジタがリソースの複合イベントステートを生成するのは別に、自身の一意なステートを発行できる。特定のリソースに加え、すべての発行済みイベントステートは特定のイベントパッケージに関連付けられる。そのイベントパッケージに対するサブスクリプションによって、ユーザーは、すべてのアクティブな発行の複合イベントステートを検出できる。

PUBLISH リクエストは ESC においてソフトステートを作成する。このイベントソフトステートは、定義済みの生存時間を持ち、ネゴシエートされた時間の後に期限切れとなる。また、次の PUBLISH リクエスト発行による更新を必要とする。特定のイベントパッケージの各リソースごとに、組み込まれたイベントハードステートも存在する可能性がある。イベントステートとは、常時存在し、期限切れしないリソースステートを示す。ESC は、PUBLISH の仕組みによって提供されるイベントソフトステートがない場合、あるいはイベントソフトステートに加えて、イベントハードステートを使用する可能性がある。このイベントハードステートの設定方法、または多様なイベントステートの集積に関する ESC ポリシーは、本仕様の範囲外である。

PUBLISH リクエストのボディは、発行済みのイベントステートを伝達する。正常に送信されたすべての PUBLISH リクエストに対する応答に、ESC は、エンティティタグの形式で識別子を割り当てる。この識別子は、後続の PUBLISH リクエスト(発行イベントステート

を修正、更新、または削除)で、EPA によって使用される。イベントステートが期限切れになるか、明示的に削除された場合、それに関連付けられていたエンティティタグは無効になる。無効なエンティティタグの発行は当然ながら失敗するため、EPA は、以前のエンティティタグを参照することなく、イベントステートを再び開始し、再送信する必要がある。

エンティティ(サブスクリバ)があるユーザーのプレゼンス情報を知りたい場合、SUBSCRIBE リクエストを作成する。このリクエストでは、SIP URI、SIPS URI(参考文献[A-1])、またはプレゼンス(pres)URI(参考文献[A-3])を使用して、Request-URIにある目的のプレゼンティティが特定される。SUBSCRIBE リクエストは、他の SIP リクエストと同様に、SIP プロキシ間を伝達される。ほとんどの場合、このリクエストは最終的にプレゼンスサーバーへ到達し、そこからリクエストへの応答を生成するか(この場合、プレゼンティティのプレゼンスエージェントとして動作する)、エッジプレゼンスサーバーへプロキシする。エッジプレゼンスサーバーは、サブスクリプションを処理する場合、プレゼンティティのプレゼンスエージェントとして動作している。プレゼンスサーバーで行われる SUBSCRIBE をプロキシするか終了するかは決定は、ローカルの問題である。

プレゼンスエージェント(プレゼンスサーバーでもエッジプレゼンスサーバーでも)は、まずサブスクリプションを認証してから、認可する。認可方法はこのプロトコルの範囲外であり、多くの方法が使用されると予想している。認可されると、200 OK 応答が返される。この時点で認可されないと、サブスクリプションは「保留(pending)」と見なされ、202 応答が返される。どちらの場合でも、PA は、プレゼンティティとサブスクリプションのステートを含む NOTIFY メッセージをすぐに送信する。サブスクリプションが保留された場合、プレゼンティティのステートは偽造されている可能性がある。たとえば、プレゼンティティの実際のステートにかかわらず、オフラインを示すなど。これは、プレゼンティティのプライバシーを保護するためのものである。プレゼンティティは、認可を付与していないサブスクリバに対して、プライバシーを公開することを望まない場合がある。プレゼンティティのステートが変わると、PA は、サブスクリプションが認可済みの全サブスクリバに対して、ステートの変更内容を含む NOTIFY を生成する。サブスクリプション自体のステート変更によって、NOTIFY リクエストのトリガとなる場合もある。このステートは NOTIFY の Subscription-State ヘッダーフィールドで伝達され、通常、サブスクリプションがアクティブか保留中かを示す。

SUBSCRIBE メッセージによって、プレゼンスエージェントで「ダイアログ」が確立する。ダイアログは、RFC 3261(参考文献[A-1])で定義されているが、ピアツーピアメッセージを容易に交換するためのエンティティペア間の SIP ステートを示す。このステートには、ルートセットとリモートターゲット URI だけでなく、双方向メッセージ(サブスクリバからの SUBSCRIBE、プレゼンスエージェントからの NOTIFY)のシーケンス番号が含まれる。ルートセットは、SUBSCRIBE の更新または NOTIFY リクエストのパスに沿って到達する SIP プレゼンスサーバーを特定する SIP(SIPS) URI のリストである。リモートターゲット URI は、メッセージのターゲットを特定する SIP または SIPS URI である。たとえば、NOTIFY の場合はサブスクリバ、SUBSCRIBE 更新の場合はプレゼンスエージェントである。

SIP には、NOTIFY メッセージと SUBSCRIBE 更新のパス上にあるようにプレゼンスサーバーがリクエストすることを許容する、レコードルートという手順がある。最初の SUBSCRIBE リクエストの Record-Route ヘッダーフィールドに URI を入力することで、これを実行できる。

最初の SUBSCRIBE の一部で交渉された有効期間に従って、サブスクリプションは継続する。サブスクリバが、その後もサブスクリプションを保持する場合は、期限切れ前にサブスクリプションを更新する必要がある。これは、最初の SUBSCRIBE で確立した同じダイアログ内で、SUBSCRIBE 更新を送信することで実行できる。この SUBSCRIBE は、最初の SUBSCRIBE とほぼ同一だが、To ヘッダーフィールドのタグ、より高い CSeq ヘッダーフィールド値、場合によっては Route ヘッダーフィールド値のセット(リクエストが取るべきプロキシパスが指定されたもの)が含まれる。

サブスクリバは、ダイアログ内で値がゼロの Expires ヘッダーフィールド(サブスクリプションの有効期間を指定)が指定された SUBSCRIBE を送信することで、サブスクリプションを終了することができる。この結果、サブスクリプションは直ちに終了する。次に、プレゼンスエージェントによって、最新のステートを含む NOTIFY リクエストが生成される。実際は、Expires がゼロの SUBSCRIBE リクエストを処理するプレゼンスエージェントの動作は、他の期限切れ値の場合とは異なる。SUBSCRIBE リクエストの保留や認可によって、最新のプレゼンティティステートとサブスクリプションステートで NOTIFY がトリガされる結果になる。

プレゼンスエージェントは、サブスクリプションを任意のタイミングで終了できる。この場合、サブスクリプションが終了したことを示す `Subscription-State` ヘッダーフィールドを含んだ `NOTIFY` リクエストを送信する。理由を示すために、`reason` パラメータを指定することもできる。

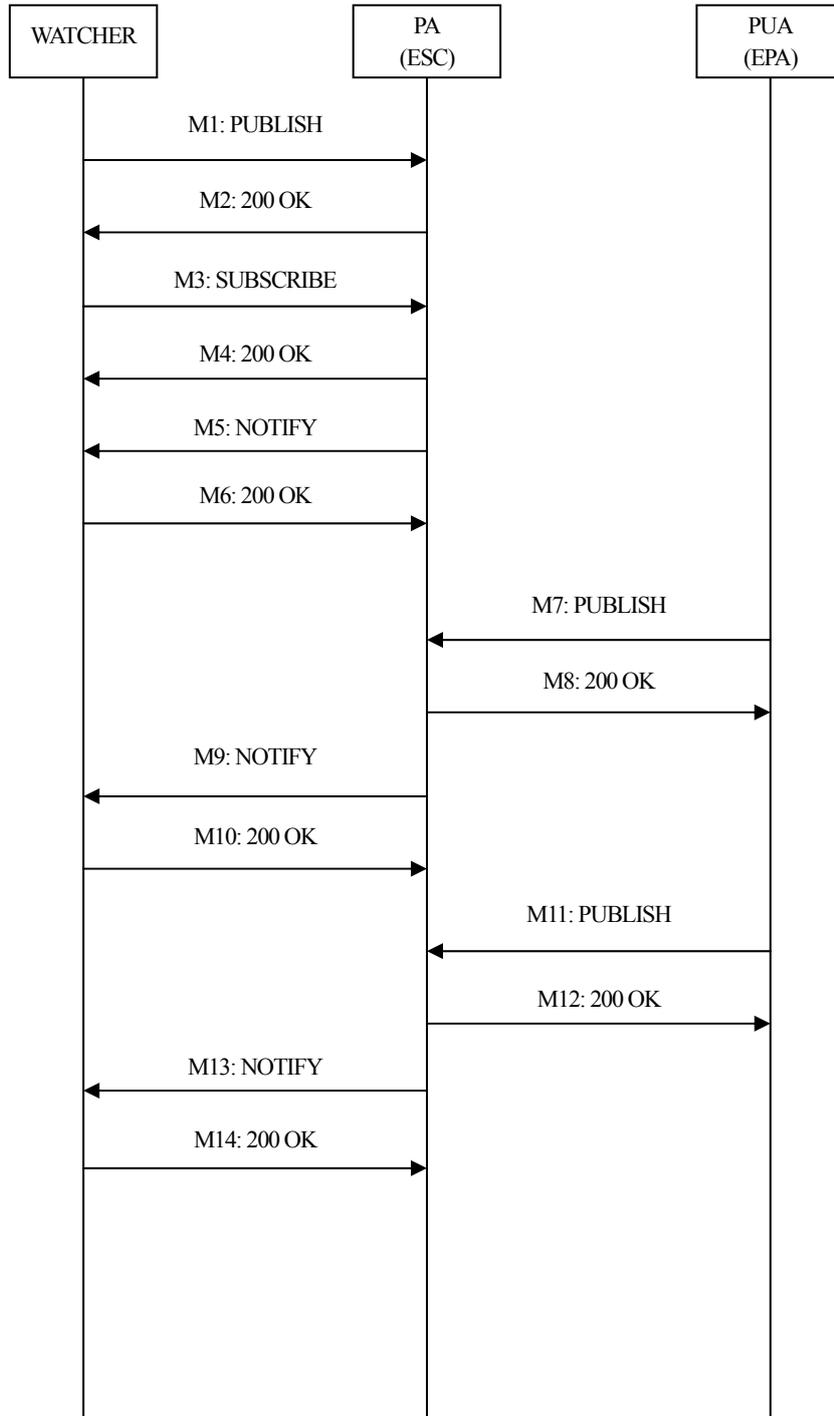
また、最新のプレゼンスステータスを取得して、現在のステータスを含むワンタイム通知を実行することもできる。この場合、すぐに期限切れになる `SUBSCRIBE` リクエストを送信する。

5. メッセージフロー例

このメッセージフローでは、プレゼンスの一連の流れを図示する。

PUA は、サーバーに対し、PUBLISH メソッドを使用する例を示す。この例のウォッチャーは、PA から、プレゼンティティのプレゼンス情報にサブスクライブしている。また、PUA は、自身のプレゼンスにSUBSCRIBE して、PA から公開された合成プレゼンスステートを表示してもよい。これは、オプションだが PUA にとってよくある手順である。この例では示していない。

Content-Length ヘッダーフィールドの値が「...」の場合、この値は、任意の長さを持つボディであることを示す。



メッセージフロー:

M1 : レジスト後、プレゼンス情報を伝えるために PUBLISH リクエストをプレゼンティティのプレゼンスエージェントに対して開始する。Expires ヘッダーフィールドは、このイベントソフトステートに、有効期限案を示す。PIDF ドキュメント部分については 6 節で説明する。

```
PUBLISH sip: watcher@example.com SIP/2.0
Via: SIP/2.0/UDP pua.example.com;branch=z9hG4bK652hsge
To: <sip: watcher@example.com>
From: <sip: watcher@example.com>;tag=1234wxyz
Call-ID: 81818181@pua.example.com
CSeq: 1 PUBLISH
Max-Forwards: 70
Expires: 3600
Event: presence
Content-Type: application/pidf+xml
Content-Length: ...
```

[発行された PIDF ドキュメント]

M2 : プレゼンスエージェントは、プレゼンス発行を許容して受信する。発行されたデータは、プレゼンティティのプレゼンス情報に組み込まれる。

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP pua.example.com;branch=z9hG4bK652hsge
;received=192.0.2.3
To: <sip: watcher@example.com>;tag=1a2b3c4d
From: <sip: watcher@example.com>;tag=1234wxyz
Call-ID: 81818181@pua.example.com
CSeq: 1 PUBLISH
SIP-ETag: dx200xyz
Expires: 1800
```

M3 : ウォッチャーは、新規のサブスクリプションを presentity@example.com のユーザーエージェントに対して開始する。

```
SUBSCRIBE sip:presentity@example.com SIP/2.0
Via: SIP/2.0/UDP host.example.com;branch=z9hG4bKnashds7
To: <sip:presentity@example.com>
From: <sip:watcher@example.com>;tag=12341234
Call-ID: 12345678@host.example.com
CSeq: 1 SUBSCRIBE
Max-Forwards: 70
Expires: 3600
Event: presence
Contact: sip:user@host.example.com
```

Content-Length: 0

M4 : presentity@example.com のプレゼンスエージェントは、サブスクリプションリクエストを処理し、新規のサブスクリプションを作成する。サブスクリプションを確認するために、200 (OK)応答が送信される。

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP host.example.com;branch=z9hG4bKnashds7
;received=192.0.2.1
To: <sip:presentity@example.com>;tag=abcd1234
From: <sip:watcher@example.com>;tag=12341234
Call-ID: 12345678@host.example.com
CSeq: 1 SUBSCRIBE
Contact: sip:pa.example.com
Expires: 3600
Content-Length: 0
```

M5 : 処理を完了するために、プレゼンスエージェントは、ウォッチャーに、プレゼンティティの現在のプレゼンスステートを持つ NOTIFY を送信する。

```
NOTIFY sip:user@host.example.com SIP/2.0
Via: SIP/2.0/UDP pa.example.com;branch=z9hG4bK8sdf2
To: <sip:watcher@example.com>;tag=12341234
From: <sip:presentity@example.com>;tag=abcd1234
Call-ID: 12345678@host.example.com
CSeq: 1 NOTIFY
Max-Forwards: 70
Event: presence
Subscription-State: active; expires=3599
Contact: sip:pa.example.com
Content-Type: application/pidf+xml
Content-Length: ...
```

[PIDF ドキュメント]

M6 : ウォッチャーは、NOTIFY リクエストの受信を確認する。

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP pa.example.com;branch=z9hG4bK8sdf2
;received=192.0.2.2
To: <sip:watcher@example.com>;tag=12341234
From: <sip:presentity@example.com>;tag=abcd1234
Call-ID: 12345678@host.example.com
CSeq: 1 NOTIFY
```

M7 : (プレゼンティティのために動作する)プレゼンスユーザーエージェントは、新規のプレゼンス情報を更新するために PUBLISH リクエストをプレゼンティティのプレゼンスエージェントに対して開始する。Expires ヘッダーフィールドは、このイベントソフトステート

に、有効期限案を示す。

```
PUBLISH sip:presentity@example.com SIP/2.0
Via: SIP/2.0/UDP pua.example.com;branch=z9hG4bK652hsge
To: <sip:presentity@example.com>
From: <sip:presentity@example.com>;tag=1234wxyz
Call-ID: 81818181@pua.example.com
CSeq: 1 PUBLISH
Max-Forwards: 70
Expires: 3600
Event: presence
Content-Type: application/pidf+xml
Content-Length: ...
```

[発行された PIDF ドキュメント]

M8 : プレゼンスエージェントは、プレゼンス発行を許容して受信する。発行されたデータは、プレゼンティティのプレゼンス情報に組み込まれる。

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP pua.example.com;branch=z9hG4bK652hsge
;received=192.0.2.3
To: <sip:presentity@example.com>;tag=1a2b3c4d
From: <sip:presentity@example.com>;tag=1234wxyz
Call-ID: 81818181@pua.example.com
CSeq: 1 PUBLISH
SIP-ETag: dx200xyz
Expires: 1800
```

M9 : プレゼンスエージェントは、プレゼンティティのプレゼンス情報に対してレポート可能な変更が加えられたことを判断し、新しいプレゼンス通知をウォッチャーに送信する。

```
NOTIFY sip:user@host.example.com SIP/2.0
Via: SIP/2.0/UDP pa.example.com;branch=z9hG4bK4cd42a
To: <sip:watcher@example.com>;tag=12341234
From: <sip:presentity@example.com>;tag=abcd1234
Call-ID: 12345678@host.example.com
CSeq: 2 NOTIFY
Max-Forwards: 70
Event: presence
Subscription-State: active; expires=3400
Contact: sip:pa.example.com
Content-Type: application/pidf+xml
Content-Length: ...
```

[新しいPIDF ドキュメント]

M10 : ウォッチャーは、NOTIFY リクエストの受信を確認する。

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP pa.example.com;branch=z9hG4bK4cd42a
;received=192.0.2.2
To: <sip:watcher@example.com>;tag=12341234
From: <sip:presentity@example.com>;tag=abcd1234
Call-ID: 12345678@host.example.com
CSeq: 2 NOTIFY
Content-Length: 0
```

M11 : プレゼンティティの PUA は、ユーザーのプレゼンスステートの変更を検出する。PUBLISH リクエストをプレゼンスエージェントに対して開始し、発行済みのプレゼンス情報を最新の変更に合わせて修正する。

```
PUBLISH sip:presentity@example.com SIP/2.0
Via: SIP/2.0/UDP pua.example.com;branch=z9hG4bKcdad2
To: <sip:presentity@example.com>
From: <sip:presentity@example.com>;tag=54321mm
Call-ID: 5566778@pua.example.com
CSeq: 1 PUBLISH
Max-Forwards: 70
SIP-If-Match: kwj449x
Expires: 3600
Event: presence
Content-Type: application/pidf+xml
Content-Length: ...
```

[発行された PIDF ドキュメント]

M12: プレゼンスエージェントは、修正の発行を許容して受信する。発行されたデータは、プレゼンティティのプレゼンス情報に組み込まれ、同じ PUA による前の発行が更新される。

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP pua.example.com;branch=z9hG4bKcdad2
;received=192.0.2.3
To: <sip:presentity@example.com>;tag=effe22aa
From: <sip:presentity@example.com>;tag=54321mm
Call-ID: 5566778@pua.example.com
CSeq: 1 PUBLISH
SIP-ETag: qwi982ks
Expires: 3600
```

M13: プレゼンスエージェントは、プレゼンティティのプレゼンスドキュメントに対してレポート可能な変更が加えられたことを判断し、新しいプレゼンス通知をアクティブなサブスクリプションすべてに送信する。

```
NOTIFY sip:user@host.example.com SIP/2.0
Via: SIP/2.0/UDP pa.example.com;branch=z9hG4bK32defd3
To: <sip:watcher@example.com>;tag=12341234
From: <sip:presentity@example.com>;tag=abcd1234
Call-ID: 12345678@host.example.com
CSeq: 2 NOTIFY
Max-Forwards: 70
Event: presence
Subscription-State: active; expires=3400
Contact: sip:pa.example.com
Content-Type: application/pidf+xml
Content-Length: ...
```

[新しい PIDF ドキュメント]

M14: ウォッチャーは、NOTIFY リクエストの受信を確認する。

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP pa.example.com;branch=z9hG4bK32defd3
;received=192.0.2.3
To: <sip:watcher@example.com>;tag=12341234
From: <sip:presentity@example.com>;tag=abcd1234
Call-ID: 12345678@host.example.com
CSeq: 2 NOTIFY
Content-Length: 0
```

6. プロトコル要素の定義

ヘッダーフィールド	場所	PUBLISH	SUBSCRIBE	NOTIFY
Accept	R	o	o	o
Accept	2xx	-	-	-
Accept	415	m*	o	o
Accept-Encoding	R	o	o	o
Accept-Encoding	2xx	-	-	-
Accept-Encoding	415	m*	o	o
Accept-Language	R	o	o	o
Accept-Language	2xx	-	-	-
Accept-Language	415	m*	o	o
Alert-Info		-	-	-
Allow	R	o	o	o
Allow	r	o	o	o
Allow	405	m	m	m
Allow-Events	R	o	-	-
Allow-Events	489	m	-	-
Authentication-Info	2xx	o	o	o
Authorization	R	o	o	o
Call-ID	c	m	m	m
Call-Info		o	-	-
Contact	R	-	m	m
Contact	1xx	-	o	o
Contact	2xx	-	m	o
Contact	3xx	o	m	m
Contact	485	o	o	o
Content-Disposition		o	o	o
Content-Encoding		o	o	o
Content-Language		o	o	o
Content-Length		t	t	t
Content-Type		*	*	*
CSeq	c	m	m	m
Date		o	o	o
Event	R	m	-	-
Error-Info	300-699	o	o	o
Expires		o	o	-
Expires	2xx	m	m	-
From	c	m	m	m
In-Reply-To	R	-	-	-
Max-Forwards	R	m	m	m
Min-Expires	423	m	m	-
MIME-Version		o	o	o
Organization		o	o	-

表 1.ヘッダーフィールドの要約 (A-O)

ヘッダーフィールド	場所	PUBLISH	SUBSCRIBE	NOTIFY
Priority	R	o	o	-
Proxy-Authenticate	407	m	m	m
Proxy-Authenticate	401	o	o	o
Proxy-Authorization	R	o	o	o
Proxy-Require	R	o	o	o
Record-Route		-	-	-
Reply-To		-	-	-
Require		o	o	o
Retry-After	404,413,480,486	o	o	o
Retry-After	500,503	o	o	o
Retry-After	600,603	o	o	o
Route	R	c	c	c
Server	r	o	o	o
Subject	R	o	-	-
Supported	R	o	o	o
Supported	2xx	o	o	o
Timestamp		o	o	o
To	c(1)	m	m	m
Unsupported	420	o	o	o
User-Agent		o	o	o
Via	R	m	m	m
Via	rc	m	m	m
Warning	r	o	o	o
WWW-Authenticate	401	m	m	m
WWW-Authenticate	407	o	m	m

表 2.ヘッダーフィールドの要約 (P-W)

7. プレゼンス情報データフォーマット(PIDF)

このセクションでは RFC3863 に定義されている、プレゼンス情報データフォーマット(PIDF)について説明する。

PIDF のために、プレゼンス情報を含む XML MIME エンティティのために新コンテンツタイプ`application/pidf+xml`が定義されている。この仕様は RFC3023 に記載されている推奨に基づいて命名規則(接尾語`+xml`)と`charset`パラメータの使用法を含む。

プレゼンス情報は XML エンコーディングされるため、XML 中のエンコーディング宣言として次のような宣言を含むべきである。”
 <?xml version=’1.0’ encoding=’UTF-8?’>”。全てのアプリケーションは最小限の相互接続性のために UTF-8 キャラクターセットを保証しなければならない。

続いて各要素について説明する。

The <presence> element

PIDF 要素は XML ネームスペース 'urn:ietf:params:xml:ns:pidf' と関連付けられる。

"application/pidf+xml" オブジェクトのルートはプレゼンス情報ネームスペースに関連付けられた <presence> 要素である。これには 0 以上の <tuple> 要素を含み、0 以上の <note> 要素が続き、複数の OPTIONAL 要素が続く。

<presence> 要素は 'entity' 属性を持たなければならない。'entity' 属性の値は PRESENTITY の 'pres' URL である。

The <tuple> element

<tuple> 要素は PRESENCE TUPLE を運ぶ。この中には必須である <status> 要素、いくつかの OPTIONAL 拡張要素(おそらく他のネームスペースの)、1 つの OPTIONAL <contact> 要素、複数の OPTIONAL <note> 要素、1 つの OPTIONAL <timestamp> 要素からなる。

tuple はプレゼンス情報の分割を行う手段を提供する。プロトコルやアプリケーションは幾つかの理由によりプレゼンティティと関連させたプレゼンス情報を分割することを選ぶかもしれない。たとえば、同一デバイスの異なるアプリケーションや区別すべきデバイスを持っているプレゼンティティ向けのプレゼンス情報があるコンポーネントのためである。タプルは複数の PIDF インスタンスを分割プレゼンス情報とする方法より好まれる。

<tuple> 要素は 'id' 属性を含み、それによって同一プレゼンティティのほかのタプルと区別できるようにする。同一プレゼンティティのほかのタプルのなかでユニークに 'id' 属性は付与されなければならない [MUST]。'id' は同一プレゼンティティの前にとられた PRESENCE INFORMATION タプルにしたがってプレゼンスドキュメントを識別するためにアプリケーションによって処理される。'id' 属性は任意の文字列であり、前述したように <tuple> 値を区別するための手段以上のものではない。

<contact> 要素は OPTIONAL であり、PRESENTITY が COMMUNICATION ADDRESS を隠す必要がある場合や複数の COMMUNICATION MEANS にタプルが関連付けられないようにするためである。<basic> ステータス要素をもつタプルは <contact> アドレスを含むべきである [SHOULD]。タプルは衝突するプレゼンス状態を持つかもしれない -- 1 つが <basic><status> の OPEN、そしてもう 1 つが <basic><status> の CLOSED などである。これは同一 <contact> を持つ場合でもありえる。

WATCHER USER AGENT が分割されたプレゼンス情報を理解するためのルールは、WATCHER USER AGENT やプレゼンスアプリケーションの能力に大きく依存する。タプルの意味を知るためのアプリケーション依存、プロトコル依存の手段がない場合、WATCHER USER AGENT は次のガイドラインに従ってもよい。WATCHER USER AGENTS は各タプルの 'id' を以前の通知で受け取ったものと関連付け、<status> 値と <timestamp> 要素があれば両者を比較することで、最後の通知以降に PIDF にあるどのタプルが変化したか注意すべきである。

The <status> element

<status> 要素は、1 つの OPTIONAL <basic> 要素、複数の OPTIONAL 拡張要素(おそらく他のネームスペースのもの)からなる。これらの <status> の子要素はタプルの status 値を含む。1 つのタプル中で複数の status 値を許容することで、異なるタイプの status 値(reachability や location)がタプルで表現できる。

本文書では <basic> ステータス値の要素のみ定義する。他のステータス値は標準拡張フレームワークに含まれる。アプリケーションが <status> 内で認識できない要素に遭遇した場合、それらが mustUnderstand="true" あるいは mustUnderstand="1" 属性がなければ無視してもよい。

The <basic> element

<basic> 要素は "open" あるいは "closed" のうち 1 つの要素を含む。

タプルがインスタントメッセージのアドレスである場合、“open”と“closed”の値はインスタントメッセージを受信する能力を示す。これらは一般的なほかの通信手段能力を示すこともあるが、本文章では詳細に規定しない。

The <contact> element

<contact>要素はコンタクトアドレスの URL を含む。オプションで‘priority’属性を含み、他のコンタクトアドレスとの優先度を関連付けることができる。属性の値は0から1の10進数で多くて小数点3桁を含む数値となるべきである[MUST]。

大きな値は大きな優先度を意味する。例として0, 0.021, 0.5, 1.00を優先度とする。priority属性が省略された場合、アプリケーションはそのコンタクトアドレスを最低優先度に分類すべきである[MUST]。もし‘priority’値が範囲外の場合、アプリケーションは値を無視すべきで、属性が無いように処理すべきである。

アプリケーションは低い優先度より高い優先度をもつコンタクトを優位となるように処理するべきである。どのように扱うかは本仕様の範囲外である。また同一優先度の扱いは実装に依存する。

The <note> element

<note>要素は文字列を含み、可読性の高いコメントに使われる。<note>要素は<presence>の子要素、あるいは<tuple>の子要素として現れる[MAY]。前者ではコメントはプレゼンティティに関するコメントであり、後者の場合、特定のタプルのコメントとなる。

<note>要素は参考文献[C-1] 2.12 に定義されているように特殊な属性xml:langを持ち、この要素のコンテンツで使用される言語を指定するべきである。この属性値は[RFC3066]に定義されている言語識別子である。使用される言語がコンテンツのエンコーディング情報など、囲まれている XML 要素の xml:lang 属性、あるいは囲まれている MIME ラッパーの Content-language ヘッダ[RFC3282]など、より大きなコンテキストで示される場合、省略してよい[MAY]。

The <timestamp> element

<timestamp>要素は、タプルのステータスが変更された日時を示す文字列を含む。この要素の値はIMPP datetime フォーマット[RFC3339]に従わなければならない[MUST]。‘T’や‘Z’を含むタイムスタンプは大文字を使用しなければならない[MUST]。

セキュリティのため、正確なステータス変更時刻が決められない場合を除き、<timestamp>要素は全てのタプルに含めるべきである[SHOULD]。

プレゼンティティは同じタイムスタンプを含む連続した<presence>要素を生成するべきではない。

PIDF の例

以下にデフォルトネームスペースを使った PIDF の例を示す。

```
<?xml version="1.0" encoding="UTF-8"?>
<presence xmlns="urn:ietf:params:xml:ns:pidf"
  entity="pres:someone@example.com">
  <tuple id="bs35r9">
    <status>
      <basic>open</basic>
    </status>
    <contact priority="0.8">im:someone@mobilecarrier.net</contact>
    <note xml:lang="en">Don't Disturb Please!</note>
    <note xml:lang="fr">Ne derangez pas, s'il vous plait</note>
```

```

    <timestamp>2001-10-27T16:49:29Z</timestamp>
  </tuple>
  <tuple id="eg92n8">
    <status>
      <basic>open</basic>
    </status>
    <contact priority="1.0">mailto:someone@example.com</contact>
  </tuple>
  <note>I'll be in Tokyo next week</note>
</presence>

```

8. プレゼンス情報の拡張

プレゼンス情報の拡張フレームワークは XML Namespace に基づいて行われる。

RFC2779 は<basic>以上の<status>拡張手段を必要とする。これらの拡張は<basic>が解釈できる手段を変更してはならない(MUST NOT)し、PIDF ボディの構造体や文法自体を変更してはならない。

要素の Namespace URI(URN)は参考文献[C-2]で定義されている。Namespace 識別子 'ietf'は参考文献[C-3]および[C-4]で定義されている。

Um:ietf:params:xml:ns:pidf

簡単な例は次のとおり。

```

<?xml version="1.0" encoding="UTF-8"?>
<impp:presence xmlns:impp="um:ietf:params:xml:ns:pidf"
  entity="pres:someone@example.com">
  <impp:tuple id="sg89ae">
    <impp:status>
      <impp:basic>open</impp:basic>
    </impp:status>
    <impp:contact priority="0.8">tel:+09012345678</impp:contact>
  </impp:tuple>
</impp:presence>

```

default XML namespace を使う場合次のとおりとなる

```

<?xml version="1.0" encoding="UTF-8"?>
<presence xmlns="um:ietf:params:xml:ns:pidf"
  entity="pres:someone@example.com">
  <tuple id="sg89ae">
    <status>
      <basic>open</basic>
    </status>
    <contact priority="0.8">tel:+09012345678</contact>
  </tuple>
</presence>

```

これに標準ネームスペースに基づいた拡張ルールによって他のステータスバリューを利用できる。
例として、ロケーションステータス値を次のように含めることができる

```
<?xml version="1.0" encoding="UTF-8"?>
<presence xmlns="urn:ietf:params:xml:ns:pidf"
  xmlns:local="urn:example-com:pidf-status-type"
  entity="pres.someone@example.com">
  <tuple id="ub93s3">
    <status>
      <basic>open</basic>
      <local:location>home</local:location>
    </status>
    <contact>im.someone@example.com</contact>
  </tuple>
</presence>
```

9. 規範となる参考文献

- [A-1] Rosenberg, J., Schulzrinne, H., Camarillo, H., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, June 2002.

- [A-2] Roach, A., "Session Initiation Protocol (SIP)-Specific Event Notification", RFC 3265, June 2002.

- [A-3] Peterson, J., "Common Profile for Presence (CPP)", RFC 3859, August 2004.

- [A-4] Bradner, S., "Key Words for Use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

- [A-5] Peterson, J., "Address Resolution for Instant Messaging and Presence", RFC 3861, August 2004.

- [A-6] Sugano, H., Fujimoto, S., Klyne, G., Bateman, A., Carr, W., and J. Peterson, "Presence Information Data Format (PIDF)", RFC 3863, August 2004.

- [A-7] Dierks, T. and C. Allen, "The TLS Protocol Version 1.0", RFC 2246, January 1999.

- [A-8] Rosenberg, J., "A Watcher Information Event Template-Package for the Session Initiation Protocol (SIP)", RFC 3857, August 2004.

- [A-9] Schulzrinne, H. Rosenberg, J., and P. Kyzivat, "Indicating User Agent Capabilities in the Session Initiation Protocol (SIP)", RFC 3840, August 2004.

- [B-1] Roach, A., "Session Initiation Protocol (SIP)-Specific Event Notification", RFC 3265, June 2002.

- [B-2] Rosenberg, J., "A Presence Event Package for the Session Initiation Protocol (SIP)", RFC 3856, August 2004.

- [B-3] Day, M., Rosenberg, J., and H. Sugano, "A Model for Presence and Instant Messaging", RFC 2778, February 2000.

- [B-4] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, June 2002.

[B-5] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[B-6] Sugano, H., Fujimoto, S., Klyne, G., Bateman, A., Carr, W., and J. Peterson, "Presence Information Data Format (PIDF)", RFC 3863, August 2004.

[B-7] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", RFC 2234, November 1997.

[B-8] Dierks, T. and C. Allen, "The TLS Protocol Version 1.0", RFC 2246, January 1999.

[B-9] Ramsdell, B., Ed., "Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.1 Message Specification", RFC 3851, July 2004.

10. 有益な参考文献

[A-10] Day, M., Rosenberg, J., and H. Sugano, "A Model for Presence and Instant Messaging", RFC 2778, February 2000.

[A-11] Peterson, J., "Enhancements for Authenticated Identity Management in the Session Initiation Protocol (SIP)", Work in Progress, May 2004.

[A-12] Calhoun, P., Loughney, J., Guttman, E., Zorn, G., and J. Arkko, "Diameter Base Protocol", RFC 3588, September 2003.

[A-13] Day, M., Aggarwal, S., Mohr, G., and J. Vincent, "Instant Messaging / Presence Protocol Requirements", RFC 2779, February 2000.

[A-14] Gutmann, P., "Password-Based Encryption for CMS", RFC 3211, December 2001.

[B-10] Campbell, B., "SIMPLE Presence Publication Requirements", Work in Progress, February 2003.

[B-11] Mahy, R., "A Message Summary and Message Waiting Indication Event Package for the Session Initiation Protocol (SIP)", RFC 3842, August 2004.

[B-12] Rosenberg, J., Schulzrinne, H., and P. Kyzivat, "Indicating User Agent Capabilities in the Session Initiation Protocol (SIP)", RFC 3840, August 2004.

[B-13] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L.,
Leach, P., and T. Berners-Lee, "Hypertext Transfer Protocol –
HTTP/1.1", RFC 2616, June 1999.

[C-1] Bray, T., Paoli, J., Sperberg-McQueen, C., and E.
Maler, "Extensible Markup Language (XML) 1.0 (Second
Edition)", W3C Recommendation, October 2000,
<<http://www.w3.org/TR/2000/REC-xml-20001006>>

[C-2] Moats, R., "URN Syntax", RFC 2141, May 1997.

[C-3] Moats, R., "A URN Namespace for IETF Documents", RFC
2648, August 1999.

[C-4] Mealling, M., "The IETF XML Registry", RFC 3688,
January 2004.