APPENDIX F

# サービスに情報を追加する方法

この付録では、Prime Provisioning で追加情報機能がどのようにサポートされるかについて説明します。次の事項について説明します。



## 概要

追加情報機能によって、ユーザは、一連の属性（名前と値のペア）を XML ファイルで定義できます。ファイルは、後にポリシーに関連付けられます。追加情報の属性は、サービス要求に関連付けられる値を定義します。これらは、GUI のラベルおよび外観を定義します。サービス要求ワークフローで、これらの値は、ユーザが入力できます。テンプレートから、または xDE プロビジョニング ロジックからこれらの属性値にアクセスして、サービスの一部として設定されるデータ値を提供することもできます。これらの値でデータ ファイルを作成する代わりに、テンプレートと組み合わせて追加属性を使用することにより、ポリシーおよびサービス要求 GUI に対して、テンプレート属性値の入力を要求できます。この付録では、Prime Provisioning の追加情報機能を理解し、使用するために必要な情報を提供します。

## 前提条件と制限事項

追加情報機能の次の前提条件および制約事項に注意してください。

- 追加情報機能は、MPLS、L2VPN、VPLS、および EVC サービスだけでサポートされます。
- MPLS-TP および TEM ポリシーおよびサービス要求は、追加情報をサポートしません。
- VRF サービス要求にはポリシーが存在しないため、追加情報をサポートしません。

- サポートされるポリシーとサービス要求タイプでこの機能を使用する前に、追加情報の定義ファイルを作成する必要があります。これは、ユーザ定義の属性と値のペアを定義する XML ファイルです。ポリシー ワークフロー内の手順で、この定義ファイルを後にロードします。このコマンドの詳細については、「追加情報の定義ファイルの作成」（P.F-7）を参照してください。

# 追加情報 GUI ワークフローの概要

次の手順では、Prime Provisioning で追加情報を実装するために実行する必要がある作業の概要を示します。この付録の残りの項では、これらのトピックの詳細情報を示します。

1. 追加情報の定義ファイルを作成します（通常、付属の XSD を使用して検証します）。このファイルは、追加情報の属性を定義します。
2. 追加の属性値を参照する、または、xDE プロビジョニングのロジックを拡張するテンプレートを作成します。
3. テンプレートに対して、1 つのデフォルトのデータ ファイルを作成します。
4. 任意で、ネゲート テンプレートとネゲート データ ファイルを追加します。
5. 適切なポリシー タイプのポリシーを作成します。
6. ポリシー作成のワークフローの [Additional Information] ウィンドウに移動します。
7. 作成した追加情報の定義ファイルをロードします。ファイルが解析され、検証され、エラーが GUI に表示されます。
8. 提供されるフィールドの値を、必要に応じて入力します。変更が必要ない標準値である場合は、追加情報の定義ファイルでこれらの値を定義できます。
9. ポリシー ワークフローで、追加情報属性を編集可能または編集不可能としてマークします。これにより、ポリシーに基づいて、サービス要求内のこれらの値を編集できるかどうかが決まります。
10. ポリシー ワークフローでは、テンプレートをイネーブルにし、追加の値にアクセスするテンプレートを参照します。
11. ポリシーを保存します。追加情報が解析され、検証され、エラーが GUI に表示されます。
12. ポリシーに基づいて、サービス要求を作成します。
13. サービス要求ワークフローの [Service Request Editor] ウィンドウには、追加情報の属性が表示され、それらを編集することができます（編集可能な場合）。
14. サービス要求を保存します。追加情報が解析され、検証され、エラーが GUI に表示されます。

# ポリシー ワークフローでの追加情報の設定

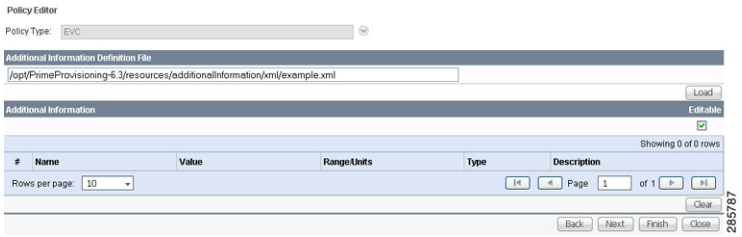
サポートされるポリシー タイプ内で追加情報機能を使用するには、次の手順を実行します。

**ステップ 1**　属性の追加先として、サポートされるポリシー タイプを編集または作成します。

**ステップ 2**　ポリシー ワークフロー ウィンドウをナビゲートし、必要に応じて属性値を設定します。

ワークフローでは複数のウィンドウが表示され、図 F-1 に示すように、[Additional Information] ウィンドウが表示されます。このウィンドウは、追加情報機能をサポートするすべてのポリシー タイプで、同様に表示され、動作します。

**図 F-1　[Additional Information] ウィンドウ**




このウィンドウには、ポリシー ワークフローの最後から 2 番目のウィンドウで、[Template Association] ウィンドウの前に表示されます。

[Additional Information] ウィンドウの使用は任意です。

**ステップ 3**　追加情報の属性と値のペアを定義する XML 定義ファイルをロードするには、[Load] ボタンをクリックします。

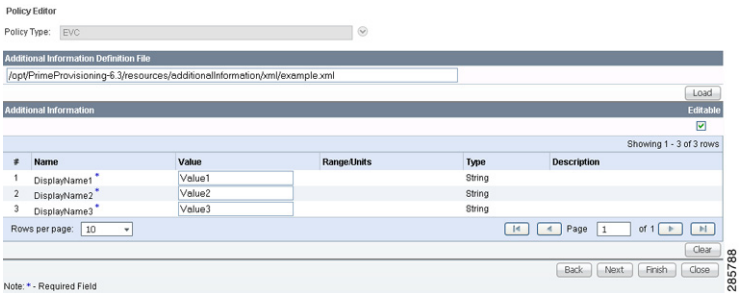
**（注）**　このファイルの作成方法については、「追加情報の定義ファイルの作成」（P.F-7）を参照してください。

定義ファイルのデフォルトのパスおよび名前は、次のとおりです。

$PRIMEP_HOME/resources/additionalInformation/xml/example.xml

ウィンドウが更新され、定義ファイルの属性と値のペアは、[Additional Information] セクションに表示されます（図 F-2 を参照）。

**図 F-2　外部 XML 定義ファイルからロードされる属性**




**ステップ 4**　必要に応じて、[Clear] ボタンをクリックして、ウィンドウの [Display] セクションに表示される属性をクリアできます。

**ステップ 5**　[Editable] チェックボックスをオンまたはオフにして、すべての追加情報属性を編集可能または編集不可能に設定します。

個々の属性を編集可能または編集不可能にすることはできません。

**ステップ 6**　ポリシーに対して、必要に応じて追加情報属性の値を設定します。

ウィンドウのこのセクションの内容と動作については、以下の説明を参照してください。

**ステップ 7**　[Next] をクリックして、ポリシー ワークフローの次のステップに進みます。

**ステップ 8**　Prime Provisioning の標準の手順に従って、ポリシー ワークフローを完了します。

ウィンドウの [Additional Information] セクションの内容と動作に関して、次の点に注意してください。

- 追加情報属性は、追加情報の定義ファイルで定義されている方法に基づいて、GUI 内でグループ化されます。
- グループが定義されている場合、各グループに対して、追加情報属性を含むページング テーブルの上部に、グループ名が表示されます。
- グループが定義ファイルで定義されていない場合、追加属性情報を含むページング テーブルだけが表示されます。
- 各属性は、ページング テーブル内の各行に表示されます。
- [Name] 列には、定義ファイルで定義されている属性の DisplayName が含まれます。属性が定義ファイルで必須としてマークされると、上付き文字のアスタリスクが DisplayName に付けられます。これは、ポリシー内で属性に値が必要なことを示すものではありません。これは、定義ファイルで値が定義される方法と、このポリシーを使用したサービス要求で、この属性に対して値が要求されることを示します。
- [Value] 列には、定義ファイルで定義されている属性の値が含まれます。
- [Range/Units] 列には、属性の範囲と単位の組み合わせが含まれます。
- [Description] 列には、定義ファイルで定義されている属性の説明が含まれます。

## ポリシー ワークフローで定義ファイルに対して実行される検証チェック

XSD の検証に加えて、追加情報があるポリシーを Prime Provisioning データベースに保存するときに、解析チェックが実行され、追加情報の定義ファイルに対して検証が実行され、次の追加の検証チェックが実行されます。[Additional Information] セクションが編集不可能としてとマークされている場合（[Editable] チェックボックスがオフのままになっている場合）、必須としてマークされているすべての属性には、値が定義されている必要があります。これに該当しない場合は、検証エラーが発生します。この制限が課されるのは、ポリシーに基づくサービス要求で、すべての必須の追加情報属性に値が必要であるためです。（追加情報が編集可能ではないために）値を編集できない場合は、ポリシーに基づいてサービス要求を作成することはできません。

ポリシー ワークフローで追加情報に対して実行する検証チェックの詳細については、「XSD の検証方法」（P.F-10）を参照してください。

# サービス要求ワークフローでの追加情報の設定

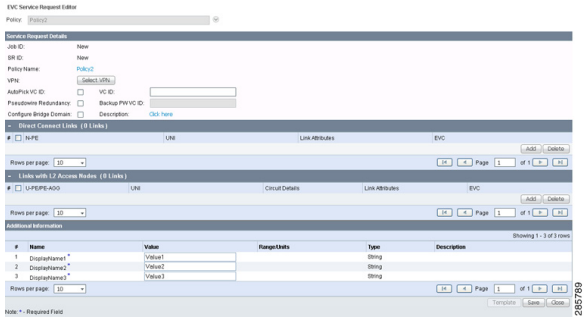
サービス要求ワークフローで追加情報機能を使用するには、次の手順を実行します。

**ステップ 1**　追加情報機能を使用して作成されたポリシーに基づいて、サービス要求を作成または編集します。

**ステップ 2**　サービス要求ワークフロー内の [Service Request Editor] ウィンドウに移動します。

サービス要求の基になるポリシーに定義された追加情報の属性がある場合、これらの属性は、図 F-3 に示すように表示されます。

図 F-3　[EVC Service Request] ウィンドウの追加情報属性




属性は、[Service Request Editor] ウィンドウですべての既存の属性の下に表示されます。属性情報の属性の形式は、ポリシーの対応するセクションと同じです。

**ステップ 3**　設定の要件に基づいて、[Service Request Editor] 内の属性を設定します。

属性の [Additional Information] セクションについて、次の点に注意してください。

- 追加情報の属性を編集できる場合は、属性の値を変更できます。
- 追加情報の属性が編集可能でない場合、値はグレーになり、変更できません。
- サービスの基になっているポリシーに追加情報属性がない場合、[Service Request Editor] ウィンドウに [Additional Information] セクションは表示されません。
- 必須としてマークされた属性には、値を設定する必要があります。この作業を行わないと、サービス要求を保存しようとしたときに、検証エラーが発生します。

**ステップ 4**　この段階では、追加情報属性にテンプレート変数をマッピングするために、テンプレートをデバイスに追加することもできます。詳細については、「テンプレートおよびデータ ファイルの追加属性の使用」（P.F-5）を参照してください。

**ステップ 5**　[Save] をクリックして、サービス要求を保存します。

# テンプレートおよびデータ ファイルの追加属性の使用

Prime Provisioning の 2 か所で、追加情報属性にテンプレート変数をマッピングできます。

- テンプレートの作成時：これを行うには、次の手順を実行します。
  1. マッピングするテンプレート変数を編集し、文字列型として定義します。
  2. テンプレート変数のデフォルト値として、追加情報の属性名を入力します。追加情報の定義ファイルに定義されている正確な名前を使用する必要があります。

（注） テンプレートで使用される属性の先頭は、$ にする必要があります（*$name* など）。これは、この値が展開時に別の値に置き換えられることを示します。この属性のデフォルト値またはデータ ファイルの作成時には、追加情報属性の正確な名前を指定します。追加情報の属性名の先頭は $ にする必要があります。これは、この属性が、実際の値によって置き換えられ、固定の文字列ではないことをテンプレート マネージャに示します。

- テンプレート データ ファイルの作成時：これを行うには、テンプレート変数の値として、追加情報の属性名を入力します。追加情報の定義ファイルに定義されている正確な名前を使用する必要があります。

上記の 2 つのいずれかを実行した後、ポリシーまたはサービス要求にテンプレートやテンプレート データ ファイルを関連付けたときに、テンプレート変数は、ポリシーまたはサービス要求の対応するユーザ定義の追加情報属性の値に置き換えられます。

# xDE プロビジョニングでの追加属性の使用

追加情報属性は、xDE プロビジョニング エンジンに渡される XML ドキュメントに追加されるため、xDE の手順でアクセスできます。

この XML ドキュメントには、次の XML ブロックを追加します。

```
<additionalInformation>
<attribute>
<name>Name1</name>
<value>123</value>
</attribute>
</additionalInformation>
```

（注） XML 属性ブロックは、各 *additionalInformation* 属性に対して繰り返す必要があります。

プロビジョニングのための現在の xDE の手順では、要求属性は、入力 XML ファイルを含むすべての手順に渡されます。xDE の手順で *additionalInformation* 属性値を使用するために、次のように MPLS SR XML 要求ドキュメントから属性 Name1 の値を抽出できます。

```
xml.xpathreference($serviceRequest,
"/MplsSR/additionalInformation/attribute[name=\"Name1\"]/value/text()")
```

また、すべての xDE の手順に渡される $additionalInformation 属性を通じて、追加情報の属性値にアクセスできます。この属性には、すべての追加情報属性の名前と値のペアのマップが含まれます。たとえば、次のように入力します。

```
map.get($additionalInformation, "Name1")
```

Name1 属性に関連付けられた値を返します

# 追加情報の定義ファイルの作成

ここでは、追加情報の定義ファイルの作成に使用できる参照情報を提供します。これは、必須 XML 要素の最小セットと追加のオプション要素が格納された XML ファイルです。このファイルは、「ポリシー ワークフローでの追加情報の設定」（P.F-2）の項で説明するように後でポリシーにロードされます。

## 必要最小限の XML 要素

例 F-1 は、追加情報の属性を定義するために、最低限必要な情報が含まれている追加情報のサンプル定義ファイルです。

**例 F-1　　最小 XML 要素を備えた追加情報の定義ファイル**

```
<additionalInformation>
<attribute>
        <name>Name1</name>
        <value>Value1</value>
    </attribute>
</additionalInformation>
```

必須 XML 要素の説明：

- *additionalInformation : additionalInformation* ブロックは、定義ファイルを開始および終了します。
- *attribute : attribute* ブロックは、定義する必要がある数の属性に対して、繰り返すことができます。各 *attribute* ブロックには、1 つの *name* 要素と 1 つの *value* 要素だけが必要です。
- *name : name* 要素には、ヌル以外の値を割り当てる必要があります。この値は、追加情報の定義ファイルで他の *name* 要素の値に対して一意である必要があります。
- *value : value* 要素には、任意の値（ヌルを含む）を割り当てることができます。この値は、追加情報の定義ファイルで他の *value* 要素の値に対して一意である必要はありません。

## 任意の XML 要素

追加情報の定義ファイルには、任意の XML 要素も含まれることがあります。ここでは、次の任意の要素について説明します。

- *group*
- *attribute/displayName*
- *attribute/description*
- *attribute/required*
- *attribute/type*
- *attribute/type/string*
- *attrribute/type/integer*
- *attribute/type/ipv4Address*
- *attribute/type/ipv6Address*
- *attribute/type/enumeration*

各要素がどのように解析され、どの条件でエラーが生成されるかについて説明します。

設定可能な任意の要素を含む追加情報のサンプル定義ファイルについては、「追加情報機能の例」（P.F-11）を参照してください。

## group

0 以上の *group* 要素が存在することがあります。

各 *group* には、少なくとも 1 つの属性ブロックが必要です。*group* 内に属性が存在しない場合は、ファイルがロードされたときにエラーが発生します。

*group* が定義されている場合は、同じレベル（*group* 外）で属性を定義できません *groups* と *attributes* が同じレベルにあると、ファイルがロードされたときに、解析エラーが発生します。

*group* 要素には *name* が必要ですが、*name* は空白にすることができます。

*group* の *name* は、空白の名前を含め、一意にする必要があります（つまり、空白の *name* は、1 回だけ使用できます）。一意ではない *group* の *name* があると、ファイルがロードされたときに、重複名エラーが発生します。

## attribute/displayName

*displayName* 要素には、ポリシーとサービス要求ワークフローの [Additional Information] テーブルで、属性の [Name] 列に表示されるテキストが含まれます。

*displayName* が定義されていない場合は、*name* 要素のテキストにデフォルト設定されます。

## attribute/description

*description* 要素には、ポリシーとサービス要求ワークフローの [Additional Information] テーブルで、属性の [Description] 列に表示されるテキストが含まれます。*description* が定義されていない場合は、空の文字列にデフォルト設定されます。

## attribute/required

*required* 要素には、属性が必要かどうかを示すブール値が含まれます。true に設定した場合、*name* テキストの横に上付きのアスタリスクが付けられます。このテキストは、ポリシーとサービス要求ワークフローの [Additional Information] テーブルで、属性の [Name] 列に表示されます。

ポリシーで、ある属性が必須として設定され、追加情報が編集不可能として設定されている場合にだけ、その属性に値を割り当てる必要があります。それ以外の場合は、属性に値を割り当てる必要はありません。

サービス要求で、ある属性が必須として設定されている場合、その属性には値セットを割り当てる必要があります。

*required* が定義されていない場合は、true にデフォルト設定されます。

## attribute/type

*type* 要素には、定義している属性のタイプを記述します。

使用できるタイプは次のとおりです。

- *string*

- *integer*
- *ipv4Address*
- *ipv6Address*
- *enumeration*

*type* 要素が定義されていない場合、デフォルト タイプは *string* です（ranges または regex は定義されません）。

*type* 要素が定義されているが、サブ要素として使用可能ないずれかのタイプがない（タイプが存在しない、またはタイプがサポートされていない）場合、ファイルがロードされたときに、解析エラーが発生します。

属性に複数の *type* 要素がある場合は、ファイルがロードされたときに解析エラーが発生します。

## attribute/type/string

*string* タイプには、次のような範囲および単位を記述する任意のパラメータがいくつかあります。

- *minLength*：文字列の最小長を定義します。属性の文字列値の長さは、検証に合格するために、この値以上にする必要があります。*minLength* が定義されていない場合、デフォルトは 1 です。
- *maxLength*：文字列の最大長を定義します。属性の文字列値の長さは、検証に合格するために、この値以下にする必要があります。
- *rangeUnits*：定義されている場合は range パラメータとともに、[Range]/[Units] 列に表示する単位を定義します。*rangeUnits* が定義されていない場合、デフォルトは「characters」です。
- *regex*：属性の文字列値を検証するために使用する正規表現を定義します。文字列値は、検証に合格するために、regex と一致する必要があります。また、*regex* が定義されている場合、*rangeDescription* には、「Pattern: regex」が追加されます。

## attribute/type/integer

*integer* タイプには、次のような範囲および単位を記述する任意のパラメータがいくつかあります。

- *lower*：範囲の下限値を定義します。属性の *integer* 値は、検証に合格するために、この値以上にする必要があります。
- *upper*：範囲の上限値を定義します。属性の *integer* 値は、検証に合格するために、この値以下にする必要があります。
- *rangeUnits*：定義されている場合は range パラメータとともに、[Range]/[Units] 列に表示する単位を定義します。*rangeUnits* が定義されていない場合、デフォルトは空の文字列です。

## attribute/type/ipv4Address

*ipv4Address* タイプには、次のような範囲および単位を記述する任意のパラメータがいくつかあります。

- *ipv4Lower*：範囲の *ipv4Lower* 値を定義します。属性の *ipv4Address* 値は、検証に合格するために、この値以上にする必要があります。
- *ipv4Upper*：範囲の *ipv4Upper* 値を定義します。属性の *ipv4Address* 値は、検証に合格するために、この値以下にする必要があります。

### attribute/type/ipv6Address

*ipv6Address* タイプには、次のような範囲および単位を記述する任意のパラメータがいくつかあります。

- *ipv6Lower*：範囲の *ipv6Lower* 値を定義します。属性の *ipv6Address* 値は、検証に合格するために、この値以上にする必要があります。
- *ipv6Upper*：範囲の *ipv6Upper* 値を定義します。属性の *ipv6Address* 値は、検証に合格するために、この値以下にする必要があります。
- *rangeUnits*：定義されている場合は range パラメータとともに、[Range]/[Units] 列に表示する単位を定義します。*rangeUnits* が定義されていない場合、デフォルトは空の文字列です。

### attribute/type/enumeration

*enumeration* タイプには、次のような範囲および単位を記述する任意のパラメータがいくつかあります。

- *enumOptions*：属性の列挙オプションを定義します。
  - 1 つ以上の *enumOptions* 要素を定義できます。
  - 少なくとも 1 つの *enumOption* 要素が定義されていない場合は、ファイルがロードされたときに解析エラーが発生します。
  - 空の文字列は、有効な *enumOption* 値ではありません。いずれかの *enumOption* 要素に空の文字列が割り当てられている場合は、ファイルがロードされたときに解析エラーが発生します。
- *rangeUnits*：定義されている場合は range パラメータとともに、[Range]/[Units] 列に表示する単位を定義します。*rangeUnits* が定義されていない場合、デフォルトは空の文字列です。

## XSD の検証方法

追加情報 XML は、XML スキーマ定義（XSD）を使用して検証されます。XSD は、メイン JAR ファイルで定義されており、ユーザは編集できません。ただし、追加情報の定義ファイルを作成する必要があるユーザは、次の場所でファイルのコピーを入手できます。

$PRIMEP_HOME/resources/additionalInformation/extAttrs.xs

ユーザが、XSD 検証をオンまたはオフにすることができる DCPL プロパティがあります。DCPL プロパティは、**additionalInformation.XML.validateWithXSD** です。デフォルトではオンになります。

## 追加情報の定義ファイルの検証方法

実行される XSD 検証と解析チェックに加えて、追加情報の定義ファイルがポリシーにロードされたときに、次の追加の検証チェックが実行されます。

- *Enumeration* 型：属性値が定義されているが、いずれかの *enumeration* オプションと一致しないと、認証エラーが発生します。重複 *enumeration* オプションがある場合は、認証エラーが発生します。
- *integer*、*ipv4Address*、および *ipv6Address* 型：属性値が定義されている場合、その値は範囲に照らしてチェックされます（範囲が定義されていない場合は、デフォルトが使用されます）。この範囲外の場合は、認証エラーが発生します。

- *string* 型：属性値が定義されている場合は、（上記の）範囲チェックに加えて、*regex* と一致する必要があります（定義されている場合）。

# 追加情報機能の例

ここでは、追加情報機能のエンドツーエンドの例を示します。この例では、次の情報を提供します。

- テンプレート
- テンプレート データ ファイル
- 追加情報の定義ファイル
- GUI に表示される属性のリスト
- サンプル GUI 入力および生成されるコンフィグレット

## テンプレート

次に、ポリシー テンプレート本体の例を示します。テンプレートは、非常に汎用的です。これは、アクセス ポートの E-Line サービスを示しています。これは、Cisco 3400 ルータの着信トラフィック用です。

```
policy-map qos-in-$Interface_Name
class class-default
#if($PIR_in_mbps==0)
     police cir $CIR_in_mbps m
#elseif($PIR_in_mbps!=0)
     police cir $CIR_in_mbps m pir $PIR_in_mbps m
#end

!
interface $Interface_Name
service-policy input qos-in-$Interface_Name
```

## テンプレート データ ファイル

次に、ポリシーに添付されるテンプレート データ ファイルを示します。

```
CIR_in_mbps:   $CIR_in_mbps
PIR_in_mbps:   $PIR_in_mbps
Interface_Name: $UNI_INTERFACE_NAME
```

## 追加属性の定義ファイル

次に、追加情報の定義ファイルを示します。

```
<additionalInformation>
<group name="QoS">
        <attribute>
                <name>$CIR_in_mbps</name>
                <value></value>
                <displayName>Committed Bandwidth</displayName>
                <type>
```

```
                        <integer>
                                <lower>1</lower>
                                <upper>32000</upper>
                                <rangeUnits>Mbps</rangeUnits>
                        </integer>
                </type>
                <description>CIR value in Mbps</description>
                <required>true</required>
        </attribute>
        <attribute>
                <name>$PIR_in_mbps</name>
                <value></value>
                <displayName>Peak Bandwidth</displayName>
                <type>
                        <integer>
                                <lower>1</lower>
                                <upper>32000</upper>
                                <rangeUnits>Mbps</rangeUnits>
                        </integer>
                </type>
                <description>PIR value in Mbps</description>
                <required>false</required>
        </attribute>
</group>
</additionalInformation>
```

# サービス要求ワークフローで表示される追加属性

この例に基づいて、2 つの新しい属性が、サービス要求ワークフローで表示されます。

- Committed Bandwidth
- Peak Bandwidth

[Committed Bandwidth] は必須フィールドで、[Peak Bandwidth] は任意のフィールドです。

# ユーザ入力とサンプル コンフィグレット

次の例は、新しい属性に対するユーザ入力と、その結果生成されるコンフィグレットを示しています。

## 例 1

ユーザ入力：

- Committed Bandwidth：25

生成されるコンフィグレット：

```
policy-map qos-in-<uni interface>
class class-default
    police cir 25m
!
interface <uni interface>
service-policy input qos-in-<uni interface>
```

## 例 2

ユーザ入力：

- Committed Bandwidth：25
- Peak Bandwidth：50

生成されるコンフィグレット：

```
policy-map qos-in-<uni interface>
class class-default
    police cir 25 m pir 50 m
!
interface <uni interface>
service-policy input qos-in-<uni interface>
```