



CHAPTER 5

ハンドラ、ルート、およびサービス記述子の使用方法

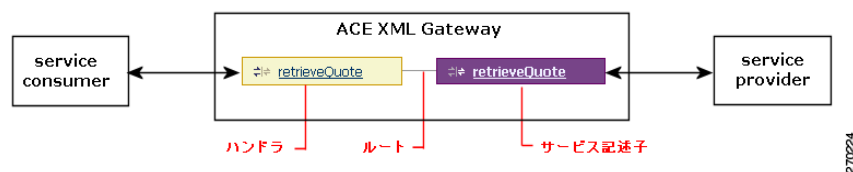
このセクションでは、拡張仮想サービスの使用方法について説明します。内容は次のとおりです。

- 「拡張仮想サービスについて」(P.5-41)
- 「基本仮想サービスから拡張仮想サービスへの変換」(P.5-42)
- 「WSDL 文書のインポートによる拡張仮想サービスの作成」(P.5-42)
- 「サービス記述子の作成」(P.5-43)
- 「ハンドラの作成」(P.5-44)
- 「ルートの使用法」(P.5-46)
- 「要求の引数の使用法」(P.5-53)
- 「次の手順」(P.5-58)

拡張仮想サービスについて

第 4 章「仮想サービスに関する作業」、で説明したように、基本仮想サービスは 1 つのポリシー オブジェクトで構成されています。ポリシー オブジェクトには、ACE XML Gateway でのユーザ インターフェイス設定と、最も遠くに位置するサービス プロバイダー設定の両方が含まれます。一方、拡張仮想サービスは、サービス ユーザのインターフェイスとサービス プロバイダーのインターフェイスを個別に定義するオブジェクトで構成されています。これらのインターフェイスは、ハンドラとサービス記述子のポリシー オブジェクトで表示されます。

図 5-1 拡張仮想サービス



拡張仮想サービスを使用して、複数の高度なメッセージ処理シナリオを実装します。その 1 つでは、分岐型のメッセージルートを作成できます（その中の複数の可能なルートから、指定したメッセージのパスを選択します）。また、拡張仮想サービスではプロトコル メディエーションのシナリオも可能で、このシナリオではあるプロトコルの受信メッセージが別のプロトコルのものに変換されます。他の仮想サービスと同様に、拡張仮想サービスにはメッセージ検証の仕様および指定したバックエンドサービスの処理に関する指示が含まれます。

この章では、拡張仮想サービスの作成および設定方法について説明します。

基本仮想サービスから拡張仮想サービスへの変換

基本仮想サービス オブジェクトは、拡張仮想サービス オブジェクトにいつでも変換できます。したがって、基本仮想サービス オブジェクトを拡張仮想サービス オブジェクトに後からいつでも変換できるため、最初に仮想サービスを作成する際、通常は基本仮想サービス オブジェクトから開始するのが適切です。逆に、ハンドラ、サービス記述子、およびルートを統合して基本仮想サービス オブジェクトにすることはできません。また、基本仮想サービス オブジェクトは通常、開発の初期段階で簡単に使用できます。たとえば、基本仮想サービスはワンクリックで複製または削除できますが、拡張仮想サービスは複数のコンポーネント オブジェクトを複製または削除しなければ同じ結果は得られません。

基本仮想サービスを拡張仮想サービスに変換すると、元の仮想サービスの設定が生成済みのオブジェクトに反映するため、元の仮想サービスと機能的には同じになります。変換後は、別のインターフェイス設定の指定や、分岐型のルーティングの設定などが行えます。

変換後に拡張仮想サービスを基本仮想サービスに戻すことはできません。仮想サービスを基本仮想サービスに復元する唯一の方法は、基本仮想サービスを再作成することです。

基本仮想サービスを拡張仮想サービスに変換するには、次の手順を実行します。

-
- ステップ 1** コンソールでの Routing ロールを持つ Administrator ユーザまたは Privileged ユーザとして、変換する仮想サービスを含むサブポリシーをアクティブなサブポリシーに設定します。
 - ステップ 2** [Virtual Services] ブラウザで、拡張仮想サービスに変換する仮想サービスの名前をクリックします。仮想サービスの設定ページが表示されます。
 - ステップ 3** ページ最下部の [Convert to Advanced] ボタンをクリックします（ボタンを表示するには、下方向へのスクロールが必要になる場合があります）。
 - ステップ 4** プロンプトが表示されたら、操作を確認します。
-

新しいハンドラの設定ページが表示されます。ここで、ハンドラの設定を変更するか、または [Virtual Services] ブラウザに戻り、変換によって生成された他のコンポーネントにアクセスできます。

WSDL 文書のインポートによる拡張仮想サービスの作成

拡張仮想サービスは、手動（ハンドラ、サービス記述子、およびルートが作成され、互いが個別に関連付けられる）または自動（オブジェクトが生成され、Web Services Description Language (WSDL) 文書のインポートによって設定される）で作成できます。このセクションでは、拡張仮想サービスを WSDL 文書のインポートによって自動的に作成する方法について説明します。

拡張仮想サービスを自動的に生成する手順は基本仮想サービスを生成する方法（「[WSDL インポートによるサービスの仮想化](#)」(P.4-23) を参照）と似ていますが、次の手順において異なる部分を強調表示しています。

-
- ステップ 1** WSDL 文書のインポート ウィザードの [Configure Interface & Access] ページで、ページ最下部の [Advanced Options] の項目を拡張します。
 - ステップ 2** 選択すると、[Condense SOAP Document operations into one virtual service per WSDL "Service" element, if possible] と記載されたオプションから選択内容が削除されます。

次の [Create a direct-mapping route between each handler and service instead of a pass-through route] というオプションが使用可能になります。

- ステップ 3** [Create a separate handler and service descriptor for each operation] のオプションをイネーブルにします。
- ステップ 4** [Create a direct-mapping route between each handler and service instead of a pass-through route] オプションで、ルートを直接マッピング方式で動作させるか、またはパススルー方式で動作させるかを選択します。このオプションにより、Gateway でのメッセージの処理方法が次のように決定します。
- 直接マッピングの場合、受信メッセージが ACE XML Gateway で分解され、ポリシーの設定および通常の正規化方法に従って再作成されます。メッセージプロトコルの標準でないか、またはポリシー設定用に特別に用意されたものではないヘッダーまたは他のメッセージ要素は、Gateway によって削除されます。このオプションでは、最も強化されたセキュリティを使用でき、メッセージの正規化という利点がありますが、予期しない統合問題が発生する可能性があります。
 - デフォルトで設定されているように、直接マッピングがディセーブルの場合はより寛容なパススルー マッピング モードが使用されます。メッセージを完全に分解したり、再作成する代わりに、ACE XML Gateway は予期しないヘッダーおよびその他のメッセージの属性を発信インターフェイスに反映します。
- ステップ 5** 「WSDL インポートによるサービスの仮想化」(P.4-23) の説明に従って、設定ウィザードの他の設定を行います。

サービス記述子の作成

Gateway で仮想化するサービスを記述する WSDL 文書が使用できない場合は、拡張仮想サービスを手作業で作成できます。このセクションでは、バックエンド サービス プロバイダーのインターフェイスの設定を定義する仮想サービスのコンポーネントである、サービス記述子の作成方法について説明します。サービス プロバイダーに送信される発信要求の仕様と受信応答の仕様が含まれます。

サービス記述子を作成するには、次の手順を実行します。

- ステップ 1** Routing ロールを持つ Administrator ユーザまたは Privileged ユーザとして ACE XML Manager の Web コンソールにログインし、操作メニューで [Virtual Services] リンクをクリックして [Virtual Services] ページを開きます。
- ステップ 2** [Virtual Services] メニューから、[Create a Service Descriptor] を選択して [Go] をクリックします ([Virtual Services] ブラウザのページ最上部にこのメニューが表示されます)。
[Step 1 of 5: Service Protocol] ページが表示されます。
- ステップ 3** [Select the protocol that this service uses] メニューからバックエンド サービスのプロトコルを選択します。このサービス記述子を使用して配信されたメッセージでは、選択したプロトコルが使用されます。Simple Object Access Protocol (SOAP)、Hyper Text Transfer Protocol (HTTP; ハイパーテキスト転送プロトコル)、ebXML メッセージング、TIBCO Rendezvous (TIB/RV)、Java Message Service (JMS)、および MQSeries のオプションがあります。



(注) ACE XML Gateway の Software Development Kit (SDK; ソフトウェア開発キット) で作成されたカスタム I/O 拡張機能がシステムにインストールされている場合は、その他のプロトコルが表示される場合があります。I/O 拡張機能の詳細については、シスコのサポートに問い合わせてください。

- ステップ 4** [Continue] ボタンをクリックします。
- ステップ 5** [General Information] ページで、このサービス記述子の名前（ポリシー内のサービス記述子で一意のもの）を入力します。
- ステップ 6** バックエンド サービスをホストするサーバを [Server] メニューで選択し、[Continue] をクリックします。
- [Server] メニューが表示されない、または正しいサーバがメニューのリストにない場合は、サーバを表すリソースがポリシー内にまだありません。サーバの定義をポリシーに追加するには、[Add a New Server] ボタンをクリックします。
- 詳細については、第 14 章「宛先サーバ設定の指定」を参照してください。
- SOAP WS-Addressing のサービス記述子は、あらかじめ設定されたサーバを使用しません。これらのタイプのサービス記述子の詳細については、「[ダイナミック サービス ルーティング WS-Addressing](#)」(P.11-127) を参照してください。
- ステップ 7** 次の 3 つの手順での設定は、サービス記述子のプロトコルによって異なります。プロトコルごとの手順の詳細については、次の表に記載されているドキュメントを参照してください。

表 5-1 各プロトコルの参照ドキュメント

プロトコル名	参照先
SOAP RPC または SOAP Document	「基本仮想サービスの手動作成」(P.4-30)
WS-Addressing による SOAP	「 ダイナミック サービス ルーティング WS-Addressing 」(P.11-127)
HTTP	「基本仮想サービスの手動作成」(P.4-30)
ebXML Message Service	第 18 章「 ebXML トラフィックに関する作業」
TIB/RV	第 17 章「 TIB/RV サービスおよび MQ サービスに関する作業」
MQSeries	第 17 章「 TIB/RV サービスおよび MQ サービスに関する作業」

- ステップ 8** 設定が完了したら、[Save Changes] をクリックして作業ポリシーのアクティブなサブポリシーに変更をコミットします。

ACE XML Manager に新しいサービス記述子の設定ページが表示されます。ここで、新しいサービス記述子で表されたサービスにユーザ メッセージをルーティングするためのハンドラを作成できます。

ハンドラの作成

ハンドラのポリシー オブジェクトは、拡張仮想サービスのユーザ インターフェイスの設定を定義します。他の設定では、ACE XML Gateway でサービスを呼び出すためのパス、ハンドラが表示されるリスニング ポート、および受信要求と送信応答のメッセージ仕様がハンドラによって指定されます。

ハンドラを作成するには、次の手順を実行します。

- ステップ 1** Routing ロールを持つ Administrator ユーザまたは Privileged ユーザとして ACE XML Manager の Web コンソールにログインし、新しいハンドラを提供するサブポリシーをアクティブなサブポリシーに設定します（まだアクティブでない場合）。
- ステップ 2** 操作メニューの [Policy] という見出しの横のクイック リンク アイコンをクリックし、[Create a new handler] を選択します。
- ステップ 3** [New Handler] ページで、このハンドラが受け入れるメッセージのプロトコルを [Select the protocol for this handler] メニューから選択し、[Continue] をクリックします。
- ステップ 4** [Step 2 of 5: General Information] ページで次の設定を行います。

- [Name] : このハンドラを説明する名前（ポリシー内のハンドラで一意的なもの）。
- [Handler Group] : このハンドラが属するハンドラ グループ。ハンドラ グループは関連するハンドラの構成に使用されます。
- [Default Message Logging] : 通常の（エラーのない）状況下での、このハンドラのアクティビティのメッセージ ログレベル。最初の開発時に、メッセージ本文の記録に役立つことがあります。
- [On Error] : エラー状態でこのハンドラが使用するメッセージ ログレベル。ログレベルの詳細については、[第 32 章「システム ステータスの監視」](#)を参照してください。

メッセージ本文のロギングがイネーブルの場合、メッセージの記録前にメッセージ処理に関する次のオプションが表示されます。

- [Request XSLT] : 記録前に、このハンドラが受信要求メッセージの本文に適用する XSL 変換。メッセージを変換するには、メニューから XSLT を選択するか、アップロードします。
- [Response XSLT] : メッセージ本文の記録前に、このハンドラが保護されたサービスからの応答メッセージ本文に適用する XSL 変換。デフォルト値は none です。メッセージを変換するには、メニューから XSL 変換を選択するか、アップロードします。
- [Send SNMP Traps] : このハンドラのこの Simple Network Management Protocol (SNMP; 簡易ネットワーク管理プロトコル) トラップの動作を設定します。デフォルト値の never にすると、ハンドラが SNMP トラップを送信しません。SNMP トラップを使用して、ハンドラに関連するステータスおよびイベントを管理者に警告するには、Gateway の各アプライアンス設定メニューから SNMP トラップをイネーブルにすることも必要です（『Cisco ACE XML Gateway Administration Guide』を参照）。

エラー用のみ、または任意のイベント用にトラップを生成できます。ハンドラ イベントが発生すると、SNMP エージェントは、softwareNotificationTrap という名前のトラップを送信します。

- [Flex Path] : ストリームベースの Reactor プロセッサをバイパスして、このサービスのメッセージ処理を Flex Path に指示します。ほとんどの場合、ACE XML Manager 自体が Flex Path にサービスを割り当てられるようにすることができます。通常は、仮想サービスで割り当てられたタスクが Reactor でサポートされていない場合（プロトコル メディエーションなど）、サービスは Flex Path 処理に自動的に割り当てられ、特別な設定は必要ありません。ただし、ポート機能による応答の GZIP 圧縮に例外が存在します。メッセージの圧縮がポートでイネーブルの場合、何らかの理由ですでに Flex Path で処理されているサービスに対してだけ、ACE XML Gateway は応答を評価して圧縮します。Reactor は圧縮を実行せず、Reactor が処理する圧縮されるはずだったメッセージは Flex Path に渡されません。このため、可能であれば圧縮されるサービスの応答を含んでいることがシステムにおいて重要な場合、サービスのハンドラまたは仮想サービス オブジェクトでこのオプションをイネーブルにする必要があります。
- [Description] : ACE XML Manager の他のユーザに対してこのハンドラについて記述したテキストメッセージ。[Virtual Services] ブラウザでこのハンドラの名前をクリックすると、ACE XML Manager で表示される情報ページの [General] セクションに表示されます。

この記述は、ACE XML Manager のサービス ディレクトリでも表示されます。このディレクトリは、サービス ユーザの開発者にサービスの可用性を通知することを目的としています。したがって、これらのユーザ向けにサービスを説明した記述を追加する必要があります。

ステップ 5 完了したら、[Continue] をクリックします。

ステップ 6 以降のページでの設定は、選択したプロトコルによって異なります。ハンドラに関するプロトコル固有の情報については、次の表に示すこのマニュアルのセクションを参照してください。

表 5-2 各プロトコルの参照ドキュメント

プロトコル名	参照先
SOAP	「基本仮想サービスの手動作成」 (P.4-30)
HTTP	「基本仮想サービスの手動作成」 (P.4-30)
ebXML Message Service	「ebXML トラフィックに関する作業」 (P.18-191)
TIB/RV	「TIB/RV サービスおよび MQ サービスに関する作業」 (P.17-185)
MQSeries	「TIB/RV サービスおよび MQ サービスに関する作業」 (P.17-185)

ステップ 7 ハンドラの設定が完了したら、[Save Changes] をクリックしてアクティブなサブポリシーに変更をコミットします。

ステップ 8 ハンドラのルートを追加するかどうかを確認するプロンプトが表示されます。ルートによってハンドラが特定のバックエンド サービスに関連付けられます (サービス記述子を使用)。現時点ではルートを追加せずに完了するか、または「ルートの使用方法」 (P.5-46) の説明に従ってルートを追加できます。

完了したら、ハンドラの設定ページが表示されます。デフォルトでは、ハンドラはまだサービス ユーザからの要求を受け入れるようにプロビジョニングされていません。ハンドラのプロビジョニングの詳細については、第 6 章「サービスへのアクセス コントロール」を参照してください。

ルートの使用方法

ルートによってハンドラとサービス記述子が接続されます。ルートでは、Gateway のユーザ インターフェイスとバックエンド サービスの間でメッセージが通過できるパスが定義されます。また、ルート自体でメッセージに関する要件や処理方法を課すことができます。

ルートは通常、ハンドラおよびサービス記述子の作成過程で作成されます。ただし、場合によってはルート ディレクトリの作成が必要になることがあります。これは、ハンドラまたはサービス記述子が複数のルートを持つように設定した場合に最も多く見られます。

複数のルートでは、決定ベースの分岐型のルーティングが可能になります。メッセージ パスに複数のルートが存在する場合、Gateway はセレクタを使用して、特定のメッセージを取得するルートを決めます。セレクタは、メッセージのヘッダー値またはその他のアトリビュートに基づいて条件を指定できます。メッセージがルートのすべてのセレクタ条件に合致している場合、メッセージはルートを通過します。


ハンドラに送信された条件に合致していないメッセージは、デフォルト ルートを指定しない限り、HTTP 404 エラーによって拒否されます。ハンドラのいずれのルートのセレクタにも合致しなかったメッセージは、デフォルト ルートに割り当てられます。

ハンドラは複数のルートを持つことができますが、指定したメッセージが受け入れられるルートは1つだけです。ハンドラへ送信されるメッセージが複数のデフォルト以外のルートの条件に合致する場合、1つのルートだけが選択されます（つまり、メッセージは複数のサービス記述子にはブロードキャストされません）。条件に合致するルートが複数ある場合は、セクタが多いルートが優先されます。ただし、条件に合致する各ルートの制限が等しい場合は、いずれかのルートが ACE XML Gateway によって選択されます。この曖昧性を解決する方法は、ACE XML Gateway では保証されません。通常は、セクタの条件が曖昧な可能性がある複数のルートを作成しないようにする必要があります。

ルートでは、プロトコルが異なるサービス記述子にハンドラを接続できます。メッセージメディエーションは、ルートで透過的に実行されます。ただし、ルートでの変換を直接指定して、メッセージの内容または構成を変更することもできます。

ルートの設定

ハンドラにルートを追加するには、次の手順を実行します。

- ステップ 1** Routing ロールを持つ Administrator ユーザまたは Privileged ユーザとして Web コンソールにログインし、ルートを設定するサブポリシーをアクティブなサブポリシーに設定します（まだアクティブでない場合）。
- ステップ 2** 操作メニューで [Virtual Services] リンクをクリックします。
- ステップ 3** [Virtual Services] ブラウザで、ルートを作成するハンドラの名前をクリックします。
- ステップ 4** 必要に応じて、サービス設定ページの [Routes] セクションが見えるまでページ最下部へスクロールし、[Add New Route] ボタンをクリックします。
- ステップ 5** このルートの宛先のサービス記述子を [Please select the service to which you wish to route requests] と記載されたメニューから選択します。

(注) 現在アクティブなサブポリシーで使用可能なサービス記述子だけがこのメニューに表示されます。
- ステップ 6** ルートをデフォルトルートにするか、またはセクタを使用してメッセージの条件を指定するかを選択します。複数のルートがある場合、デフォルトルートでは他のルートの条件に合致しないメッセージ用のパスが提供されます（デフォルトルートの使用はオプションですが、複数のデフォルトルートを存在させることはできません）。このルートにセクタを追加するには、「[セクタの追加](#)」(P.5-48)の手順に従います。
- ステップ 7** [Save Changes] をクリックして新しいルートを作成します。

新しいルートが作成され、ハンドラに接続されます。ここで、このルートに対してカスタム例外および追加の要求/応答処理を任意に設定できます。

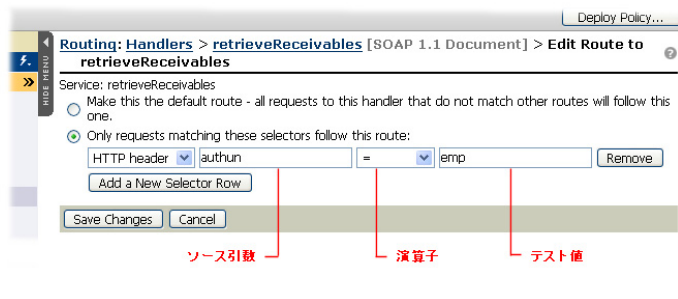
セレクトタの追加

特定のハンドラから分岐型のルーティングを実装するには、セレクトタを使用してルートを通過するメッセージの条件を設定します。セレクトタは、メッセージの内容、ヘッダー値、またはその他のアトリビュートに基づいて条件を課すことができます。

既存のルートにセレクトタを追加するには、次の手順を実行します。

- ステップ 1** ハンドラ設定の [Routes] セクションで、セレクトタを追加するルートの [Selectors] ヘッダーの横の [Edit] リンクをクリックします。
- [Edit Route to <ServiceDescriptor>] ページが表示されます。
- ステップ 2** このルートを設定して 1 つ以上のセレクトタに対する受信メッセージを設定するには、[Only requests matching these selectors] ボタンをイネーブルにします。
- [Add a New Selector Row] ボタンがイネーブルになります。
- ステップ 3** [Add a New Selector Row] ボタンをクリックします。
- セレクトタ設定用のコントロールが、3 つのフィールドの形式でページに表示されます。左側のメニューは、条件が適用されるメッセージ機能を示しています。この機能はハンドラのプロトコルによって異なることに注意してください。
- たとえば、HTTP POST Arguments ハンドラの場合、指定したヘッダーまたは引数の値に対する基準を設定できます（メッセージの仕様で引数を設定している場合）。
- ステップ 4** セレクトタと合致させるメッセージ機能をページの左側のメニューから選択します。
- テストするメッセージ機能を選択すると、そのタイプの機能の基準を指定するために追加コントロールが表示されます。図 5-2 は、メッセージ引数の形式を示しています。

図 5-2 セレクトタ設定



- ステップ 5** コントロールを使用して、メッセージがルートを通過するために合致する必要がある条件を指定します。
- たとえば、カスタム HTTP ヘッダーによってスクリーニングするには、条件のアトリビュートとして HTTP ヘッダーを選択し、ヘッダー名を示します。テスト値で、受信メッセージで合致するとメッセージが受け入れられる値を指定します。
- ステップ 6** 目的の数だけ条件を追加します。メッセージがルートを通過するためには、指定したすべてのセレクトタ条件に合致している必要があります。
- ステップ 7** [Save Changes] をクリックして、変更内容を作業ポリシーにコミットします。

新しいセレクトタ情報が、ハンドラ情報ページの [Routes] セクションの [Selectors] ペインに表示されません。

デフォルト ルートの指定

複数のルートを持つハンドラに対して、デフォルト ルートを指定できます。デフォルト ルートは、その他のルートの条件に合致しないメッセージに使用できます。



(注)

デフォルト ルートの使用は必須ではありません。メッセージがハンドラのいずれのルートの条件にも合致しない場合は、HTTP 404 エラー応答がクライアントに返されます。

デフォルトでは、新しいルートにセレクトアの条件は含まれません。これらはデフォルト ルートとして作成されます。ルート条件の設定後、デフォルト ルートとしてルートを復元することが必要になる場合があります。復元を行うには、「セレクトアの追加」(P.5-48) の説明に従ってセレクトア設定にアクセスします。セレクトアの [Edit Route] ページで、[Make this the default route] と記載されたオプションをクリックします。

複数のデフォルト ルートを持つハンドラを含むポリシーは導入できません。導入すると、「ERROR: This Handler defines multiple default routes.」というエラーがコンソールに表示されます。このエラーを修正するには、追加のデフォルト ルートに対して条件を削除または追加する必要があります。

ルートでのメッセージの内容の変更

ルートでは、複数の方法でメッセージの内容を変更できます。たとえば、ルートを通過するメッセージに適用する XSLT をルート上で設定します。ルートでメッセージを変更する機能は、ハンドラやサービス記述子で提供されるものと同様です。ただし、ルートでの内容の処理を指定すると、この動作をハンドラ上のルート間の特定のルートだけに設定でき、サービス記述子に動作を割り当てる必要はありません。処理の適用に最適な場所（ルート、サービス記述子、またはハンドラ）は、使用例によって異なります。

デフォルトでは、ルートはハンドラとサービス記述子の間でメッセージ本文と引数のコンテンツをパススルー モードで通過させます。ACE XML Gateway によって受信メッセージの HTTP ヘッダーが削除され、送信メッセージ用の新しい HTTP ヘッダーが作成されます。

ルートでは、メッセージ本文および引数のコンテンツをパススルー方式または直接マッピングで処理できます。

- 直接マッピングでは、受信メッセージが Gateway で分解され、ポリシーのサービス設定および Gateway の通常の正規化方法に従って再作成されます。したがって、ポリシー設定用に特別に用意されたものではないヘッダーまたはその他の要素は、Gateway によって削除されます。このオプションでは、最も強化されたセキュリティを使用でき、メッセージの正規化という利点がありますが、予期しない統合問題が発生する可能性があります。
- デフォルトのパススルー マッピングは、許容度がより高くなります。メッセージを完全に分解したり、再作成する代わりに、Gateway は予期しないヘッダーおよびその他のメッセージのアトリビュートを発信インターフェイスに反映します。

また、メッセージ引数のクロスマッピングを設定することもできます。この場合、ある引数の値を使用して別の引数に入力します。

コンテンツ マッピングの作成

ルート内で入力メッセージから出力メッセージへフィールドをマッピングするには、最初にハンドラで引数と対応するサービス記述子の仕様を設定する必要があります。その後、次の手順でマッピングを設定できます。

これらの作業はメッセージ トランザクションの応答区間における引数のマッピングに適用されますが、その手順は他のコンテキストでも同様です。

ステップ 1 [Virtual Services] ブラウザで、引数のマッピングを設定するサービス記述子の名前をクリックします。

ステップ 2 [Incoming Response] 設定で、[Response Message Specification] の横の [edit] リンクをクリックします。



(注) 引数設定は、引数と無関係のコンテキストでは使用できません。たとえば、HTTP 応答処理では引数設定を使用できません。一般的に、応答メッセージでの引数処理は、メッセージングベースのソース (JMS など) に関連している可能性が高くなります。

ステップ 3 [Request Message Specification] ページで、[Add A New Row] をクリックします。

ステップ 4 引数に関する次の設定を行い、完了したら [Save Changes] をクリックします。

- [Name] : 引数の名前。
- [Req] : フィールドが必要かどうか。
- [Type] : 引数のタイプ。
- [Validation] : 正確性を保つために ACE XML Gateway による引数の確認が必要かどうか。タイプおよびコンテンツが検証対象となります。コンソールでタイプとコンテンツを選択すると、選択したタイプに適した検証方法を設定できるフィールド (XML 引数の XML スキーマ セレクタなど) が表示されます。

サービス記述子の応答仕様では、これらの設定はソースの引数、つまり、受信応答のマッピング ソースの引数を参照します。ハンドラの応答仕様では、これらの設定は送信応答での引数の作成方法を制御します。

ステップ 5 サービス記述子に接続されているハンドラで、マッピング ターゲットの引数を設定します。これは、クライアントが応答メッセージで見つけることが予期される引数であり、この応答メッセージにはサービス記述子で設定した引数のコンテンツが入力されることに注意してください。

ステップ 6 完了したら、ハンドラ情報ページの [Routes] 設定で、次のいずれかの見出しの横にある [edit] リンクをクリックします。

- [Req.Processing] : 受信要求から発信要求へフィールドをマッピングします。
- [Resp.Processing] : 受信応答から送信応答へフィールドをマッピングします。

ステップ 7 [Argument Processing] メニューから、引数のマッピングを指定するオプション ([Map body and properties to body and properties] など) を選択します。

ステップ 8 [From Service] で、[To Consumer] 引数に対応するメニュー (要求をマッピングする場合は、[From Consumer] で [To Service] 引数に対応するメニュー) からソースの引数を選択します。または、値を選択しないか、引数の入力に使用する定数値を選択します。

図 5-3 引数のマッピング



ステップ 9 引数のマッピング後、[Save Changes] をクリックして作業サブポリシーに変更をコミットします。

引数のマッピングの詳細については、「[要求の引数の使用方法](#)」(P.5-53) を参照してください。メッセージメディアエーションの詳細については、「[JMS メディアエーションの設定](#)」(P.16-181) を参照してください。

HTTP ヘッダーの処理

デフォルトでは、受信メッセージの HTTP ヘッダーは送信メッセージに反映されません。ACE XML Gateway は、受信メッセージのヘッダーを消費して、送信メッセージ用の新しいヘッダーをポリシーで指定されたとおりに作成します。

ヘッダーが通過する、ヘッダーに定数値が割り当てられる、または正規表現の照合と置換によってヘッダーが変更されるよう、この動作を変更できます。変更するには、ルートでの HTTP ヘッダーの処理仕様を定義します。仕様では、受信ヘッダーを名前で特定する必要があります。

受信ヘッダーを名前で指定する代わりに、ワイルドカード演算子 (*) を使用して任意のヘッダーに一致させることができます。これにより、ACE XML Gateway がメッセージで受信したすべてのヘッダーを通過させることができます。特に、バックエンドのアプリケーションまたはネットワーク要素が要求内の [Set-Cookie] または [User-Agent] などのヘッダーに依存する場合、この設定の指定が必要になることがあります。

特定タイプのヘッダーの値は、ポリシーで設定できません。通常、これらのヘッダーにはメッセージに適したランタイム値が入力されます ([Date] および [Content-Length] ヘッダーなど)。さらに、ヘッダーのパススルーは「hop-by-hop」ヘッダーと見なされるものには適用されません。このようなヘッダーは、単一のトランスポート層接続のコンテキストだけにおいて重要であり、次のものがあります。

- Accept-Encoding
- Connection
- Keep-Alive
- Proxy-Authenticate
- Proxy-Authorization
- TE
- Trailers
- Transfer-Encoding
- Upgrade



(注) ACE XML Gateway で設定される Server HTTP ヘッダー値を設定できます。設定にアクセスするには、[System Management] ページで [I/O process settings] をクリックします。[Server Name] フィールドで、HTTP サーバおよび Reactor HTTP プロセッサの値を個別に設定できます。

カスタム ヘッダーを含むその他のタイプの HTTP ヘッダーの場合、HTTP ヘッダーの処理を次のように設定できます。拡張仮想サービスと基本仮想サービスでは手順が異なることに注意してください。拡張仮想サービスの場合は、次の手順に従います。

-
- ステップ 1** HTTP ベースのハンドラの設定ページで、次のいずれかの横にある [edit] をクリックします。
- [Req.Processing] : 発信要求のヘッダーを設定します。
 - [Resp.Processing] : 送信応答のヘッダーを設定します。
- ステップ 2** ページの [HTTP Header Processing] セクションで、[Add a New HTTP Header] ボタンをクリックします。このオプションは、HTTP ベースのルート (SOAP または HTTP GET など) でしか表示されません。
- ステップ 3** [Header Name] フィールドで、メッセージに表示される正確なヘッダー名を入力します。
- このフィールドにワイルドカード演算子 (*) を入力して、受信メッセージで受け取ったすべてのヘッダーを ACE XML Gateway に通過させることができます。ワイルドカード照合はパススルー ヘッダーのオプションだけで使用し、コンテンツ置換を実行するオプションでは使用しないことに注意してください。また、一部のワイルドカード照合はサポートされません (custom-*-header など)。
- ステップ 4** [Action] では、次のいずれかを選択します。
- [pass through] : ヘッダーおよびヘッダー値が、変更されずに着信インターフェイスから発信インターフェイスへ反映されます (前述のとおり、これは [Server] ヘッダーまたは [Date] ヘッダーには適用されません)。
 - [send constant value] : ヘッダー値に指定した値を再書き込みします。受信要求に存在しない場合は、ヘッダーが追加されます。
 - [use regular expression match] : 出力パターンで指定したテキストとともに、受信ヘッダー値の変数部分を使用します。バックスラッシュと変数 (\1、\2 など) を使用して、受信ヘッダーの一致した値を送信ヘッダーに反映します。
- ステップ 5** 完了したら [Save Changes] をクリックします。
-

基本仮想サービス オブジェクトでは、サポートされているヘッダーのパススルーしか設定できません (ヘッダーの再書き込みなどの機能の場合は、オブジェクトを拡張仮想サービスに変換する必要があります)。

基本仮想サービス オブジェクトの HTTP ヘッダー パススルーをイネーブルにするには、次の手順を実行します。

-
- ステップ 1** サービス設定の要求処理または応答処理のセクションで、[HTTP Header Processing] というラベルの横の [enable] リンクをクリックします。
- ヘッダーを受信要求からバックエンド サービスへパススルーするには、[Request Processing] セクションの横の [enable] リンクをクリックします。バックエンド サービスから返されたヘッダーをクライアントへパススルーするには、[Response Processing] セクションの横の [enable] リンクをクリックします。

ステップ 2 オプションをイネーブルにしてすべての受信 HTTP ヘッダーを通過させます。

ポリシーを導入して ACE XML Gateway で変更内容を有効にします。

要求の引数の使用方法

通常、プロトコルが異なるハンドラとサービス記述子の間のルーティングは、メッセージの変換方法の詳細を気にせずに設定できます。これは、ACE XML Gateway によってメディエーションに適したメッセージに自動的に変換されるためです。このデフォルト変換は、場合によっては特定の結果を得るために変更が必要になることがあります。多くの場合はデフォルトの結果で十分です。ただし、URL 要求の引数を含むメッセージのメディエーションには例外が存在します。この場合、有用な変換を確実に行うには、いくつかの特定の設定手順を実行する必要があります。

HTTP Get Argument ハンドラの場合、要求の引数は要求 URL の疑問符（「?」）デリミタの後にアンパサンド（「&」）で区切られた複数の引数とともに示されます。次に例を示します。

```
http://sample.com/quote/agent?sid=2345&loc=ca
```

HTTP Post Arguments ハンドラの場合、要求の引数は要求の本文で伝達されますが、HTTP Get 引数と同じ形式の **name** 値を使用します。

ACE XML Gateway に引数のデータを処理させるには、次の複数の方法があります。

- 引数のパススルー：変更されずに URL 引数が要求 URL へ反映されます。引数のパススルーは、ユーザ インターフェイスとバックエンド インターフェイスの両方が HTTP Get Arguments プロトコルの場合に自動的に行われます。HTTP Post ハンドラの場合は、この動作を直接設定する必要があります。
- 引数のマッピング：特定の引数のコンテンツが発信要求の要素（別の引数または本文）に反映されます。
- 引数の変換：ACE XML Gateway によって受信要求の引数が XML 文書として表示されます。XSLT を文書に適用してコンテンツをさらに処理します。このオプションでは、要求内の複数の任意の引数を処理できます。

使用可能なオプションは、使用しているハンドラおよびサービス記述子のプロトコルによって異なります（以降のセクションを参照）。

引数のマッピングの詳細については、「[ルートでのメッセージの内容の変更](#)」(P.5-49) を参照してください。その他のオプションについては次で説明しています。

URL 引数のパススルーの使用

受信要求で受け取った引数は HTTP Get Argument の発信要求用であるため、HTTP Post Argument サービス記述子に転送された場合に発信要求の URL には自動的に反映されません。

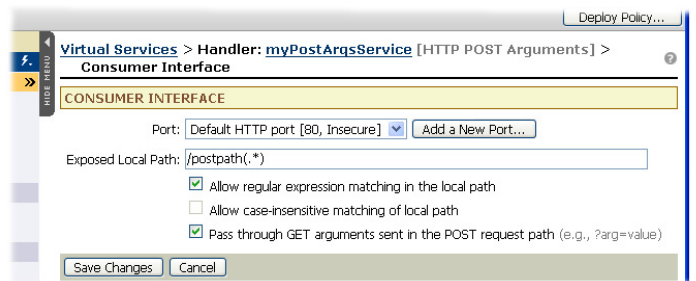
デフォルトでは、HTTP Get Argument ハンドラまたは HTTP Post Argument ハンドラに対する受信要求で受け取った引数は、HTTP Post Argument の発信要求の本文で引数にマッピングされます（HTTP Get Argument サービス記述子にルーティングされている HTTP Get Argument ハンドラにアドレス指定された引数は、発信要求 URL へ自動的にパススルーされます）。

次の手順を使用して、本文ではなく発信要求 URL に受信要求の URL 引数を反映させることができます。バックエンド サービスにおいて要求内の HTTP Post 本文と URL 引数の両方が求められる場合は、この設定を使用します。

最初に URL 引数のパススルーを次の手順で設定します。

- ステップ 1** 引数のパススルーを設定するハンドラの設定ページを開きます。
- ステップ 2** [Consumer Interface] という見出しの横の [Edit] リンクをクリックします。
- ステップ 3** まだ選択されていない場合は、[Allow regular expression matching in the local path] と記載されたチェックボックスをクリックします。URL 引数のパススルーではこの機能をイネーブルにする必要があります。
- ステップ 4** [Exposed Local Path] フィールドに、受信要求のクエリーパラメータに合致させる正規表現を追加します。任意の数の任意のクエリーに合致させるには、ワイルドカード表現を追加します ((.*) など)。
- ステップ 5** [Pass through GET arguments sent in the POST request path] と記載されたオプションを確認します。次のようなページが表示されます。

図 5-4 ユーザ インターフェイスの引数のパススルー

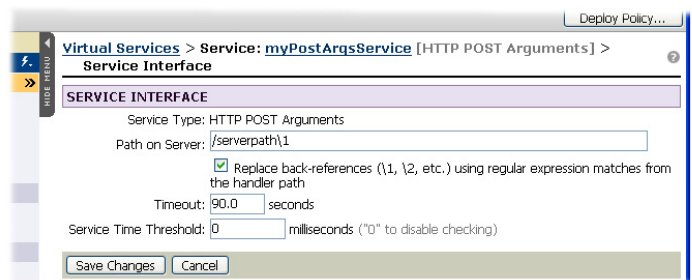


- ステップ 6** [Save Changes] をクリックし、次の説明に従ってサービス記述子を設定します。

このハンドラがメッセージを転送するサービス記述子ごとに次の手順を実行します。

- ステップ 1** [Virtual Services] ブラウザで、サービス記述子の名前をクリックしてサービス記述子の設定を開きます。
- ステップ 2** [Service Interface] という見出しの横の [Edit] リンクをクリックします。
- ステップ 3** [Path on Server] フィールドで、既存のパスに後方参照を追加します。ユーザパスのフィールドにある正規表現が 1 つの場合、追加する必要がある後方参照は 1 つ (\1) だけです。
- ステップ 4** [Replace back-references (\1, \2, etc.) using regular expression matches from the handler path] と記載されたチェックボックスをオンにして、変更内容を保存します。

図 5-5 サービス インターフェイスの引数のパススルー



設定はこれで完了しました。ポリシーの導入後、ハンドラにアドレス指定された要求の引数が、発信要求 URL にパススルーされます。

引数の処理

送信メッセージへの受信メッセージ引数の簡単なマッピングを作成する手順は、「[ルートでのメッセージの内容の変更](#)」(P.5-49) で説明しています。このセクションで説明されている引数のマッピングの場合、引数の名前はポリシーで設定する必要があります。ポリシーで指定されていない引数はドロップされます。

一部のアプリケーションでは、さらに大きな柔軟性が求められます。特に、引数の動的マッピングは、ますます一般的になりつつある使用例に必要です。その使用例とは、クライアント側の従来の Web アプリケーションまたは REST ベースの Web アプリケーションからバックエンドの SOAP サービスへのメディエーションです。

クライアント要求の URL 引数で渡されたデータは通常、バックエンドアプリケーションにとって重要であり、発信要求に反映される必要があります。

ルートの要求処理設定で [Convert incoming arguments to XML in outgoing request body] オプションを選択すると、ACE XML Gateway によって要求の引数から XML 文書が生成されます。このオプションを使用すると、引数のソース文書を処理する XSLT を ACE XML Gateway のデフォルト形式から目的のサービスで求められる形式へとさらに指定できます。

引数の XML 文書の処理に必要な知識は、SOAP RPC と SOAP Document のどちらのサービスにマッピングするかによってわずかに異なります (以降のセクションを参照)。

SOAP Document サービスの要求の引数の処理

引数を HTTP Get 要求または Post Body 要求から SOAP Document サービス記述子に変換する際、ACE XML Gateway によってソース文書が作成されます。各要求の引数は、このソース文書で param 要素にマッピングされます。

```
<param name="arg_name" value="arg_value" />
```

要素において、arg_name は要求内で渡される引数の名前であり、arg_value はその値です。

同じ名前の複数の引数が文書内に存在する可能性があります。引数は要求内で表示される順に表示されます。また、param 要素が 1 つの params 親要素に含まれます。

たとえば、次の要求 URL では、[例 5-1](#) に示す SOAP の中間メッセージが ACE XML Gateway によって作成されます。

http://example.com/quote?custid=9239&loc=ca&loc=us

例 5-1 引数のソース XML 文書の例

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

<soap:Envelope
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <params>
      <param name="custid" value="9239"/>
      <param name="loc" value="ca"/>
      <param name="loc" value="us"/>
    </params>
  </soap:Body>
</soap:Envelope
```

したがって、ルート用に設定された一般的な XSLT は、param 要素に一致し、必要に応じて引数を処理します。

次の例は、XSLT がこのメッセージを処理できることを示しています。この例では、Envelope 要素と Body 要素が維持されています。purchaseOrder 要素内では、param ごとに要素が作成されます。要素は name アトリビュートを使用して命名され、value アトリビュートのコンテンツを含みます。

例 5-2 XSLT による引数処理の例

```
<xsl:stylesheet
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  version="1.0"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" >

  <xsl:output method="xml"/>
  <xsl:template match="*">
    <xsl:element name="{name(.)}" namespace="{namespace-uri(.)}">
      <xsl:for-each select="@*">
        <xsl:attribute name="{name(.)}"
          namespace="{namespace-uri(.)}">
          <xsl:value-of select="."/>
        </xsl:attribute>
      </xsl:for-each>
      <xsl:apply-templates/>
    </xsl:element>
  </xsl:template>

  <xsl:template match="params">
    <xsl:element name="purchaseOrder">
      <xsl:for-each select="param">
        <xsl:element name="{@name}">
          <xsl:value-of select="@value"/>
        </xsl:element>
      </xsl:for-each>
    </xsl:element>
  </xsl:template>

</xsl:stylesheet>
```

このタイプの引数処理を実装するには、次の手順に従います。

- ステップ 1** ターゲットの SOAP Document サービス記述子にルーティングされている HTTP Get Arguments ハンドラまたは HTTP Post Arguments ハンドラの [Req Processing] 設定を開きます。



(注) 詳細については、「[ルートでのメッセージの内容の変更 \(P.5-49\)](#)」を参照してください。

- ステップ 2** 引数処理のオプションで、[Convert incoming arguments to XML in outgoing request body] を選択します。このオプションは、要求の引数を中間 XML 文書に変換するよう ACE XML Gateway に指示します。この中間 XML 文書に対して、最終処理用の XSLT を適用できます。
- ステップ 3** [Transform with XSLT] メニューで、XML 文書に適用する XSLT を指定します。XSLT がまだポリシーに読み込まれていない場合、オプションメニューは表示されません。[Upload] ボタンを使用して、XSLT をポリシーに追加します。
- ステップ 4** [Save Changes] をクリックして、変更内容を作業ポリシーにコミットします。

変換の詳細については、[第 13 章「XML Stylesheet Language Transformations \(XSLT\) を使用したメッセージ変換」](#)を参照してください。

SOAP RPC サービス向けの要求の引数の処理

HTTP Get Argument サービスと SOAP RPC サービスの間のルートでは、ACE XML Gateway がターゲットの SOAP Document サービスと同様に引数を処理します。ただし、ACE XML Gateway が生成する引数のソース文書はわずかに異なります。

SOAP RPC サービス記述子の URL 引数を変換すると、次のようになります。

- サービス記述子用に設定された SOAP Method 値は、SOAP Body 要素の最初の子要素として使用されます。
- 受信要求の各 GET 引数は発信要求の要素として、引数の値は要素のコンテンツとして表示されません。

ACE XML Gateway で実行される RPC エンコードは、SOAP 1.1 および 1.2 の仕様で定義されている SOAP RPC の規則に適合していることに注意してください。特に、RPC エンコードは Web Service Interoperability Basic Profile 1.0 (WSI BP 1.0) に準拠するサブセットに適合しています。

詳細は、次のサイトを参照してください。

- SOAP 1.1 エンコードの説明
(http://www.w3.org/TR/2000/NOTE-SOAP-20000508/#_Toc478383512)
- SOAP 1.2 (<http://www.w3.org/TR/2003/REC-soap12-part2-20030624/#soapforrpc>)
- WSI BP 1.0 (<http://www.ws-i.org/Profiles/BasicProfile-1.0-2004-04-16.html>)

したがって、SOAP RPC サービス記述子にルーティングされている HTTP Get ハンドラに対する要求で、そのサービス記述子が submitPO の SOAP Method 値を持つ場合、次の要求が例 5-3 に示す引数の文書になります。

`http://example.com/get2?item=1231&loc=CA`

例 5-3 SOAP RPC 要求として表示される URL 引数の例

```
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soap:Body
    soap:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
    <submitPO>
      <item>123251</item>
      <location>CA</location>
    </submitPO>
  </soap:Body>
</soap:Envelope>
```

ターゲットの SOAP RPC サービスの場合、このメッセージはデフォルト形式で十分な場合があります。ただし、SOAP Document サービスについては、XSLT をメッセージに適用してさらに処理できます。

引数処理の詳細については、「[SOAP Document サービスの要求の引数の処理](#)」(P.5-55) を参照してください。

次の手順

ポリシーの導入後、ACE XML Manager の Web コンソールに内蔵のテスト ブラウザを使用して初期設定をテストできます。ハンドラのコンテキストで、テスト ブラウザがユーザ インターフェイスに要求を送信します ([Test Handler] ボタンを使用)。サービス記述子で、テスト ブラウザがバックエンド サービスへ要求を直接送信します ([Test This Service] ボタンを使用)。詳細については、「[テストメッセージの送信](#)」(P.4-30) を参照してください。

ハンドラ、ルート、およびサービス記述子を作成して外部サービスのインターフェイスおよびユーザ インターフェイスを定義した後、この章の説明に従って、サービスのルールおよび処理タスクを追加できます。