



CHAPTER 13

XML Stylesheet Language Transformations (XSLT) を使用したメッセージ変換

この章では、ACE XML Gateway での Extensible Markup Language (XML) Transformations の使用方法について説明します。内容は次のとおりです。

- 「ACE XML Gateway での変換」(P.13-139)
- 「XSLT 変数」(P.13-140)
- 「LDAP 結果の変換例」(P.13-144)
- 「リモート XSLT による処理」(P.13-146)

ACE XML Gateway での変換

メッセージ処理の通常の過程では、ACE XML Gateway は通常、メッセージをさまざまな方法で変換します。たとえば、メッセージをあるプロトコルから別のプロトコルへ (Java Message Service (JMS) から HyperText Transfer Protocol (HTTP; ハイパーテキスト転送プロトコル) へなど) 変換する、メッセージを暗号化または復号化する、または、メッセージヘッダーの追加や消費などです。

これらの変換は ACE XML Gateway ポリシー設定の副産物です。しかし、メッセージに直接変換を適用することもできます。XSL Transformations (XSLT) を使用すると、ACE XML Gateway を通過するメッセージのコンテンツを制御できます。

XSLT バージョン サポート

ACE XML Gateway は、XSL Transformations World Wide Web Consortium (W3C) Recommendation Version 1.0 の指定に従って XSLT をサポートします。詳細については、次を参照してください。

<http://www.w3.org/TR/xslt>

この仕様に記述されるすべての言語機能を ACE XML Gateway に適用される XSLT ファイルに使用できます。

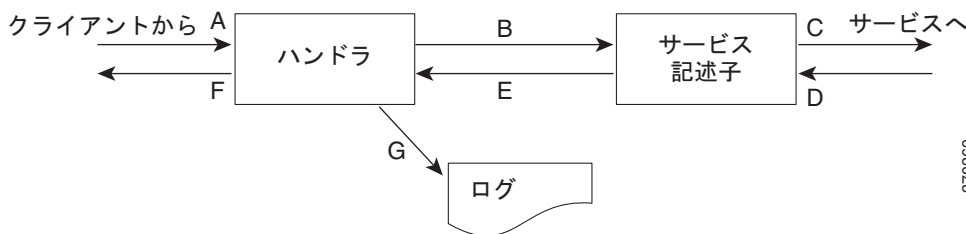
さらに、ACE XML Gateway は、ソース IP アドレス、要求時間、要求引数などの、メッセージコンテンツに関する情報にアクセスするためのシステム変数を定義します。

XSLT 処理の段階

図 13-1 に示すとおり、XSLT 処理は、処理フローのさまざまな段階でメッセージに適用できます。

- 要求メッセージの前処理 (A)
- 要求ルートでのメッセージ本文処理 (B)
- 要求の後処理 (C)
- 応答の前処理 (D)
- 応答ルートでのメッセージ本文処理 (E)
- 応答の後処理 (F)
- ログ前のメッセージ本文 (G)

図 13-1 メッセージ処理



変換は一般にメッセージに直接作用しますが、メッセージログ変換 (G) は、ログに保存されるためのメッセージのコピーに作用します。これは、そうしなければ記録される機密情報を隠すために、しばしば使用されます。

XSLT 変数を使用すると、この種類の変換が記録される情報を拡張し、要求に関する情報を追加して、トラブルシューティングや開発などに役立てることができるメカニズムを提供できます。

リモート XSL Transformations の適用

他の種類のリソースと同様に、ACE XML Gateway ポリシーで利用するために、XSL Transformations は通常、まず ACE XML Manager のポリシーにアップロードされ、必要に応じてサービス定義に適用されます。しかし、XSLT ファイルの場合、ACE XML Gateway がメッセージの処理の際に XSLT をリモートリソースから動的に取得する追加オプションがあります。

リモート XSLT の取得を使用することで、ACE XML Gateway ポリシーを編集、再導入することなく XSLT を変更、改善できます。一方、より大きな柔軟性、つまり XSLT ファイルの動的な選択を実現する、設定オプションの基盤が形成されます。XSLT ファイルの動的な選択では、特定のメッセージの変換に使用される XSLT が、メッセージ処理のときに評価される要因に基づいて複数の XSLT から選択されます。これによって、特定の受信者へのコンテンツに適応に非常に大きな柔軟性が実現します。

XSLT 変数

XSLT 変数を使用することで、実行時情報にアクセスできます。次の 2 種類の変数があります。

- ユーザ定義変数
- 組み込み変数

ユーザ定義変数は、Software Development Kit (SDK) 拡張機能（認証や変換拡張機能など）で作成され、SDK 拡張機能内の値に結び付けられた変数です。このような拡張機能内では、拡張機能間の情報共有を目的とする SDK データというものに情報を保存できます。他の拡張機能からだけでなく、XSLT もこの情報にアクセスできます。

この種類の変数の値にアクセスするには、「\$SDK_DATA」プレフィックスを付けて変数名を参照します。次に例を示します。

```
<xsl:copy-of select="$SDK_DATA-key"/>
```

この例では、「key」が、XSLT の適用前に、ACE XML Gateway 処理パスのメッセージを処理した拡張機能によって変数に与えられた名前です。拡張機能から変数を使用する方法の詳細については、『Cisco ACE XML Gateway Developer Guide』を参照してください。

組み込み変数はシステムによって定義され、設定されます。これらは、処理されているメッセージに関連するランタイム情報を提供します。copy-of XSLT エレメントを使用し、変数を select の対象とする、次のような形式でコンテンツにアクセスできます。

```
<xsl:copy-of select="$<prefix>-<var>"/>
```

ここで、prefix が変数の種類で、var が XSLT 変数の名前です。上述のとおり、組み込み変数には複数の種類があります。

- メタデータ (META) メッセージに関する一般的な情報。これらは主に、ACE XML Gateway の内部のメッセージ処理に関係します。
- 転送 (TRANSSRC)。発信元 IP などの転送関係のメッセージの特性。
- メッセージ ヘッダー (MSGSRC)。メッセージ ヘッダー値の情報。
- セキュリティ (SECSRC)。暗号化およびその他のセキュリティ関係のメッセージのプロパティ。
- LDAP (LDAPRESULT)。ACE XML Gateway がメッセージに関して発行した LDAP クエリーの結果。
- 引数 (ARGSRC)。要求内で渡された引数。多くの場合、Universal Resource Locator (URL) の name-value ペアです。XSLT がメッセージ処理チェーンのどの場所で適用されるか、要求処理か応答処理のいずれかによって、同じ要求引数を取得するために別の変数を使用する必要があることに注意してください。

変数の概要

次の表に、XSLT 変数を示します。

表 13-1 XSLT 変数

変数	説明
META-GUID	グローバルに一意的な、メッセージの識別子 17 文字 (16 進数)。この識別子は主に内部利用が目的です。 しかし、シナリオのトラブルシューティングに役立つことがあります。
META-Receive-Time	メッセージが受信された時間。形式: 2005-12-12T22:54:53Z
TRANSSRC-http-client-ip	クライアントの IP アドレス。ドット付き 10 進表記法 (n.n.n.n)。
TRANSSRC-http-server-port	ACE XML Gateway で要求が受信されたポートの番号 (例: 80)。
MSGSRC-http-method	GET、POST、DELETE などのクライアントの HTTP メソッド。

変数	説明
MSGSRC-http-full-path	引数 (例: /service?a=b) を含む、クライアントの要求するフル URL パス。
MSGSRC-http-path	引数を含まない、クライアントの要求するサービスのパス。
MSGSRC-http-host	クライアントが送信した Host ヘッダー。
MSGSRC-SOAPAction	クライアントが送信した SOAPAction ヘッダー。
MSGSRC-Content-Type	クライアントが送信した Content-Type ヘッダー。
MSGSRC-<Header_Name>	MSGSRC-<Header_Name> 形式を使用して特定される、クライアントが送信した HTTP ヘッダー。ここで、Header_Name は、メッセージに現れる実際のヘッダーの名前です。
SECSRC-encrypted	クライアントからの要求が、SSL を使用して暗号化されていたかどうか。次の値を返します。 <ul style="list-style-type: none"> 1: ACE XML Gateway によって暗号化された要求を受信した場合。 0: 暗号化されていない場合。
SECSRC-cert-digest	コロンで区切られた 20 個の 16 進数 2 文字である、クライアントの証明書のダイジェスト。
SECSRC-cert-verified	ACE XML Gateway がクライアント証明書を認証したかどうか。 <ul style="list-style-type: none"> 1: クライアントの証明書が ACE XML Gateway によって確認された場合。 0: 確認されていない場合。
SECSRC-cert-dn	クライアント証明書の認定者名。
SECSRC-cert-issuer-dn	クライアント証明書の発行元名。
SECSRC-cert-revoked	クライアント証明書が取り消されたかどうか。次の値を返します。 <ul style="list-style-type: none"> 1: クライアントの証明書が取り消された場合。 0: クライアントの証明書が取り消されなかった場合。
LDAPRESULT-<n>	LDIF テキスト形式の LDAP クエリーの結果。n は、クエリーの番号を示します。
ARGSRC-<Argument_Name>	要求処理に適用された XSLT 内でアクセスされた場合、要求内に送信された引数の値。Argument_Name をアクセスする値の引数の実際の名前に置き換えます。 たとえば、要求の引数が prodid として GET 要求の引数内で提供された場合、その値は要求処理の中で ARGSRC-prodid 変数を使用してアクセスされます。 応答処理の場合、オリジナルの要求で供給された引数に REQARGSRC プレフィクスを使用してアクセスします。
REQARGSRC-<Argument_Name>	要求で送信された引数の値。応答処理に適用される XSLT から変数にアクセスする場合は、この変数プレフィクスを使用します。 たとえば、要求の引数が prodid として GET 要求引数内で提供された場合、その値は応答処理の中で REQARGSRC-prodid 変数を使用してアクセスされます。

XSLT 変数の使用例

XSLT 変数の使用に慣れる 1 つの方法は、メッセージ ログ変換に使用してみることです。メッセージ ログ変換は、要求元およびバックエンドサービスのどちらに送信する場合でもメッセージに影響はありません。その代わりに、アプライアンスでメッセージが記録される前にメッセージに適用されます。したがって、外部のテスト コンポーネントに影響を与えることなく試すことができます。

次のサンプルに、メッセージ値を複数の変数の出力に置き換えるだけの簡単な XSLT を示します。

例 13-1 変数のある XSLT

```
<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="soap:Envelope">
      <xsl:element name="soap:Body">
        <xsl:element name="variableTestResults">
          <TRANSSRC-http-client-ip>
            <xsl:copy-of select="$TRANSSRC-http-client-ip" />
          </TRANSSRC-http-client-ip>
          <SECSRC-encrypted>
            <xsl:copy-of select="$SECSRC-encrypted" />
          </SECSRC-encrypted>
          <MSGSRC-http-path>
            <xsl:copy-of select="$MSGSRC-http-path" />
          </MSGSRC-http-path>
        </xsl:element>
      </xsl:element>
    </xsl:element>
  </xsl:template>
</xsl:stylesheet>
```

このリストには 3 つの変数があります。

- TRANSSRC-http-client-ip：要求者の IP アドレス。
- SECSRC-encrypted：要求は SSL を使用して暗号化されている。
- MSGSRC-http-path：クライアント要求でのリソース パス。

ACE XML Manager ログに書き込まれるメッセージに XSLT を適用するには、まず、例 13-1 のテキストをファイル名の拡張子が .xslt であるテキスト ファイルにコピーします。次に、次の手順を実行します。

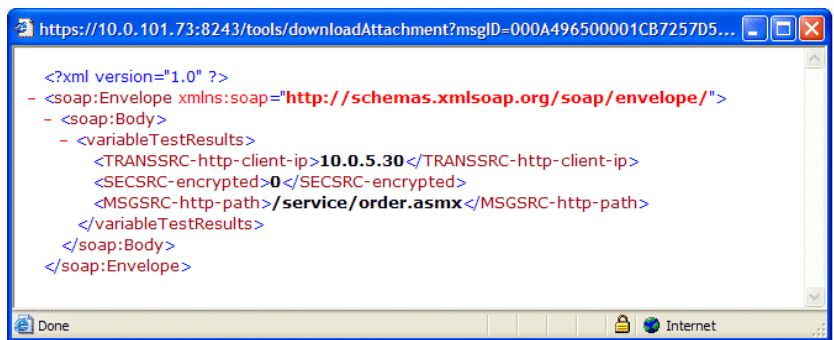
-
- ステップ 1** Web コンソールで、操作メニューで [Virtual Services] リンクをクリックします。
 - ステップ 2** テスト変換を適用する仮想サービス オブジェクトの名前をクリックします。
 - ステップ 3** ハンドラ ページの [General] 設定見出しの横にある [Edit] リンクをクリックします。
 - ステップ 4** [Default Message Logging] の選択が [log bodies of inbound and outbound messages] であることを確認してください。
 - ステップ 5** [Click the configure XSL transformation of message bodies before logging] という見出しの下の設定を拡張し、[Upload XSLT] ボタンをクリックします。
 - ステップ 6** [Upload XSL Transformation Resource] ダイアログ ウィンドウに、リソースの記述的名前が表示されます。
 - ステップ 7** [Browse] ボタンをクリックし、作成した XSLT ファイルへ移動して一覧から選択します。

- ステップ 8** ファイルを選択した後で [Upload] をクリックします。ACE XML Manager はアップロードされる XSLT を検証します。エラーが見つかった場合、ACE XML Manager は、[Upload] ウィンドウにエラーを表示し、操作をキャンセルします。
- ステップ 9** [From the Request XSLT] メニューから、ロードした XSLT をリソース名で選択します。
- ステップ 10** [Save Changes] をクリックし、ポリシーを導入します。

ここで、変換リソースを使用して設定したサービスヘトラフィックを向けることで変換をテストできます。サービスにトラフィックを送信した後、[Message Traffic Log] の中で、該当する要求の [View] カラムの中の [req/resp pair] リンクをクリックします。

最後に、[Logged Message Content] ウィンドウの本文のエンコードリンク ([text/xml; charset=utf-8] など) をクリックして、記録されたメッセージを表示します。メッセージは次のように表示されます。

図 13-2 XSLT によって書き換えられたメッセージ ログの出力



変換によって追加された項目は、対応するランタイム値が代入されることに注意してください。

LDAP 結果の変換例

「XSLT 変数の使用例」の例は、Gateway 変換の概略の理解に役立ちますが、この例の実際の適用範囲は広くありません。次に、変数のある XSLT 使用方法の実用的な例を示します。ここでは、LDAP 認証クエリーの結果が発信要求メッセージに追加されています。これは、ACE XML Gateway が LDAP ディレクトリに対して実行したクライアント認証クエリーの結果をバックエンドサービスが知る必要があるシナリオで役立ちます。

1 つのクライアント要求によって、複数の要求を ACE XML Gateway が LDAP サーバに対して発行することがあります。LDAP クエリーが実行されるごとに、LDAPRESULT-x の形式の変数として結果が追加されます (X は 1 から始まる整数)。

したがって、ユーザがセットアップクエリーを使用して認証されることをポリシーが指定する場合、セットアップクエリー結果は、LDAPRESULT-1 であり、認証クエリーの結果は、LDAPRESULT-2 となります。

例 13-2 に、発信メッセージに LDAP 変数情報を追加する変換の例を示します。

例 13-2 LDAP クエリー結果を使用する XSLT

```

<?xml version="1.0"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:l="http://www.example.com/LDAPStuff">

<!-- If the envelope already has a header, copy over and add it -->
<xsl:template match="soap:Envelope[count(*[local-name()='Header'])=1]">
  <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>
    <soap:Header>
      <xsl:apply-templates select="soap:Header/*"/>
      <xsl:call-template name="CreateHeader" />
    </soap:Header>
    <xsl:apply-templates select="soap:Body" />
  </xsl:element>
</xsl:template>

<!-- If the envelope does not have a header, create one -->
<xsl:template match="soap:Envelope[count(*[local-name()='Header'])=0]">
  <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>
    <soap:Header>
      <xsl:call-template name="CreateHeader" />
    </soap:Header>
    <xsl:apply-templates/>
  </xsl:element>
</xsl:template>

<!-- Create our header -->
<xsl:template name="CreateHeader">
  <l:UserLDAP>
    <Result1><xsl:copy-of select="$LDAPRESULT-1" /></Result1>
    <Result2><xsl:copy-of select="$LDAPRESULT-2" /></Result2>
  </l:UserLDAP>
</xsl:template>

<!-- identity transform -->
<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:stylesheet>

```

この変換は、LDAP クエリー結果を使用して Simple Object Access Protocol (SOAP) メッセージにヘッダーを作成します。例 13-2 メッセージにすでにヘッダーがある場合は、結果は既存のヘッダーに追加されます。そうでない場合は、ヘッダーが作成されます。

この変換は、2 つの LDAP クエリー結果を取り込むように設計されています。ACE XML Gateway で設定可能な、特定の種類の LDAP 要件によって、2 つの異なるクエリーが LDAP サーバに対して発行されます。1 番目はセットアップクエリーと呼ばれ、2 番目はセットアップクエリーからの情報を組み入れたクエリーです。この変換は LDAPRESULT-1 変数、および、LDAPRESULT-2 変数に結果が示されます。

セットアップクエリーの結果は、次の文で作成されたエレメントに含まれます。

```
<Result1>
  <xsl:copy-of select="$LDAPRESULT-1" />
</Result1>
```

ACE XML Gateway は実行時に LDAP ディレクトリが返した値を使用して Result1 エレメントに代入します。発信要求に適用される場合は、バックエンドサービスが必要に応じて結果を受け取り、消費します。

XSLT の最後のテンプレートに注意してください。これは、変換スクリプトで一般的に使用されるパターン、かつ、変換開発作業の基礎として使用できるパターンを示しています。テンプレートは、入力文字列から出力文字列へコンテンツを単にパススルーしています（しばしば、同一変換 (identity transformation) と呼ばれます）。これは、メッセージの小さい部分だけを変更し、残りの部分を変更せずに残す場合に有効です。

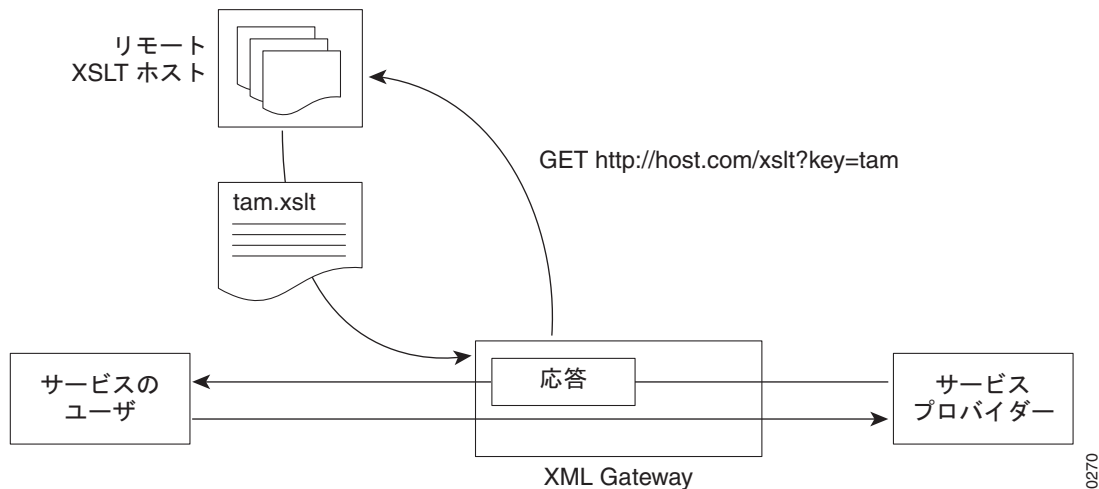
リモート XSLT による処理

他の種類のリソースと同様に、ACE XML Gateway ポリシーで利用するために、XSL Transformations は通常、まず Manager にアップロードされ、必要に応じてサービス定義に適用されます。しかし、XSLT の場合、ACE XML Gateway がメッセージの処理の際に XSLT をリモートリソースから動的に取得する追加オプションがあります。

リモート XSLT 取得によって、非常に動的な XSLT が可能になります。つまり、ポリシーを変更または導入することなく XSLT を追加または変更できます。外部リソースが XSLT を提供するため、XSLT の動的な選択も可能です。

動的な XSLT 選択を使用すると、ACE XML Gateway は、XSLT をホストするサーバに鍵の値を渡します。リモートサーバは鍵を使用して、メッセージ変換での使用のためにどの XSLT を ACE XML Gateway に提供するかを決定します。

図 13-3 変換リソースのリモート取得



リモート XSLT を使用することで、サービス要求が返す情報の表現方法に関して、パートナーおよびサービス エクステンダに効果的な制御を提供できます。たとえば、コンテンツの表示を個々のエンドユーザに特化した XSLT をパートナーが開発できます。受信者に最も適切な形式でデータを表示でき、高度に適応させたユーザ中心の応答コンテンツが可能になります。

リモート リソースへの接続は常に信頼できるわけではないため、リモート サーバが非応答の場合の代替アクションを設定できます。選択肢として、ユーザにエラーを返す、変換せずにメッセージの処理を続ける、および、デフォルト XSLT の使用があります。

この方法で使用する XSLT は ACE XML Gateway でキャッシュされないことに留意してください。Gateway は、XSLT リソースを必要とときに毎回、リモート サーバから取得します。これは、使用するケース、および、ネットワークの特性によってトラフィック処理パフォーマンスに重大な影響を与えることがあります。

次のセクションで、リモート リソースからの XSLT 取得の設定方法および動的 XSLT 選択の設定方法について説明します。

リモート XSLT 取得の使用

次の手順は、メッセージ処理に使用する XSLT を実行時に取得するための ACE XML Gateway の設定方法を説明します。この機能を使用するためには、XSLT をホストするサーバはネットワークで ACE XML Gateway へ実行時にアクセスできる必要があります。また、XSLT が URL ロケーションに存在する必要があります。

リモート XSLT を設定するには、次の作業を実行します。

- ステップ 1** 操作メニューで、[XSL Transformations] リンクをクリックします。
- ステップ 2** [XSL Transformations] ページで、[Add a New Remote XSLT Resource] をクリックします。
- ステップ 3** XSLT リソースで一意の名前をリソースに付けます。
- ステップ 4** [Remote Source] メニューで [Fixed URL] が選択されていることを確認します。
- ステップ 5** [URL] フィールドに、リソースの完全な URL パスを入力します。ACE XML Gateway は、この URL への GET 要求を解決します。次に例を示します。

`http://swan.example.com/docs/remoteIdXform.xsl`

- ステップ 6** リモート サーバが非応答の場合に ACE XML Gateway がどのアクションを実行するかを指定します。オプションは次のとおりです。
- [return an error message]。クライアントは、メッセージの処理中に変換エラーが発生したことを示すエラーメッセージを受け取ります。
 - [continue processing]。変換なしで、メッセージの処理が継続されます。
 - [apply local XSLT]。選択されたリモート XSLT が使用できない場合は、ACE XML Manager にアップロードされた XSLT が使用されます。
- ステップ 7** [Save Changes] をクリックします。

リソースは、ローカル XSLT をサポートするメッセージ処理のどの段階にも適用できます。

リモート リソースをサービス定義に適用し、ポリシーを導入したら、試験メッセージをサービスに送信してログをチェックし、設定が期待どおりに機能していることを確認します。動作している場合、情報レベル ログ イベントは、リモート リソースが使用された時間、および、次のようなイベント説明を示します。

Downloading remote resource from <http://swan.example.com:80//docs/remoteIdXform.xml>

動的に選択されたリモート XSLT 取得の利用

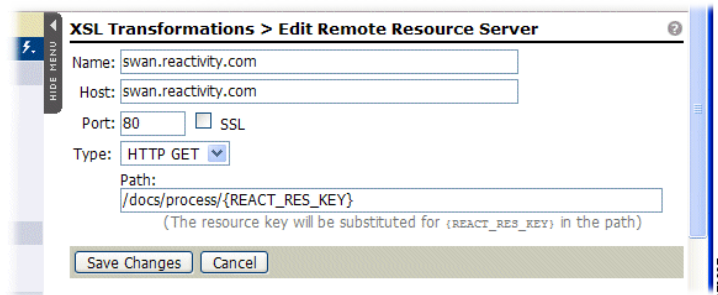
特定の XSLT を変換リソースとして指定する代わりに、動的に選択された XSLT を使用して、リモート XSLT ホスト、および、要求処理時の XSLT の選択にホストが使用する「鍵」を設定できます。前述のとおり、この機能の使用は、ポリシー設定の外部の次の条件を意味します。

- 受信ユーザ要求は、XSLT 選択の鍵として使用できる値を含んでいる必要があります。この値は、HTTP ヘッダー、要求引数、または、XPath で特定されるメッセージのロケーションの値です。
- リモート サーバが使用可能でなければならず、ACE XML Gateway の HTTP GET 要求に応答内で XSLT を返すことがあります。サーバは、ACE XML Gateway が渡した鍵の値を認識できなければなりません (GET 要求の URL に設定可能な形式で表されます)。

これらのリソースが使用できる場合、動的な XSLT 選択を次のように設定できます。

- ステップ 1** 操作メニューで [XSL Transformations] リンクをクリックします。
- ステップ 2** [Add a New Resource Server] ボタンをクリックします。
- ステップ 3** [Name] フィールドに、リソース サーバで一意的な名前を入力します。
- ステップ 4** [Host] フィールドに、host.mycompany.com などの XSLT リソースのあるサーバのホスト名または IP アドレスを入力します。
- ステップ 5** ホストの Web アプリケーションが XSLT リソースの要求をリスンするリスニング ポートを入力します。
- ステップ 6** 接続がセキュア ソケット レイヤを使用する場合は、[SSL] チェックボックスを選択します。
- ステップ 7** [Path] フィールドに XSLT を提供する Web アプリケーションのパスを入力します。変数 REACT_RES_KEY を使用して URL に鍵を指定します。次のように、中カッコが必要です。
- ```
/resource/xslt/{REACT_RES_KEY}
```
- 次のようにページが表示されます。

図 13-4 リモート変換リソース サーバ設定

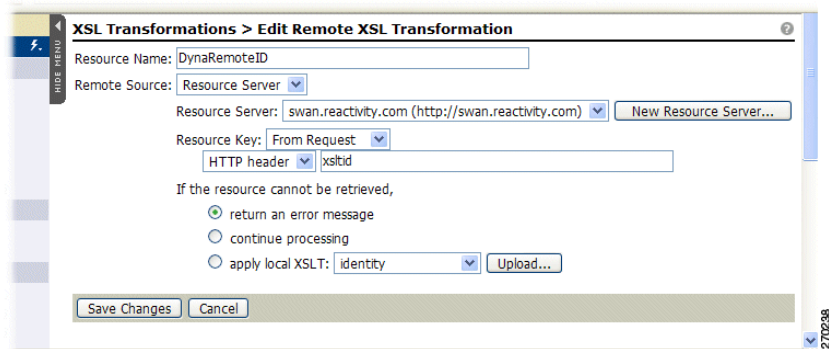


**ステップ 8** [Save Changes] をクリックします。

次のように、リモート XSLT リソースを設定します。このプロセスは、ターゲット リソースとして固定 URL を指定する代わりにリソース サーバを設定することを除いて「[リモート XSLT 取得の使用](#)」(P.13-147) に記載されたものとほぼ同じです。

- ステップ 1** 操作メニューで、[XSL Transformations] リンクをクリックします。
- ステップ 2** [XSL Transformations] ページで、[Add a New Remote XSLT Resource] をクリックします。
- ステップ 3** リモート XSLT ファイルおよびローカル XSLT ファイルの両方で一意な名前をリソースに付けます。
- ステップ 4** [Remote Source] メニューから [Resource Server] を選択します。
- ステップ 5** [Resource Server] リストから定義済みのリソース サーバを選択するか、または、[New Resource Server] ボタンをクリックして、新しいリソース サーバを追加します。
- ステップ 6** 受信要求またはバックエンド サーバからの応答のいずれかの HTTP ヘッダーとして、または Xpath によって、あるいは、URL 引数としてメッセージのリソース鍵を特定します。
- この値は、リモート リソースへのパスで設定した {REACT\_RES\_KEY} 値の代替値として使用されます。
- すべてのサービス定義で、すべてのリソース鍵タイプが使用できるわけではないことに注意してください。たとえば、XPath によって特定されるリソース鍵は、HTTP GET 要求に対しては使用できません。
- ステップ 7** リモート サーバが非応答の場合に ACE XML Gateway がどのアクションを実行するかを指定します。オプションは次のとおりです。
- [return an error message]。クライアントは、メッセージの処理中に変換エラーが発生したことを示すエラーメッセージを受け取ります。
  - [continue processing]。変換なしで、メッセージの処理が継続されます。
  - [apply local XSLT]。選択されたリモート XSLT が使用できない場合は、ACE XML Manager にアップロードされた XSLT が使用されます。

図 13-5 リモート変換設定



**ステップ 8** [Save Changes] をクリックします。

リソースはサービス定義に適用できます。ポリシーによって、コンテンツ処理に XSLT の適用が許可されている段階であればどの段階でもリソースが使用できます。