



# Caché での SOAP と Web サービスの使用法

Version 5.1

2006-03-14

Caché での SOAP と Web サービスの使用法

Caché Version 5.1 2006-03-14

Copyright © 2006 InterSystems Corporation.

All rights reserved.

このドキュメントは、Sun Microsystems、RenderX Inc.、アドビ システムズ および ワールドワイド・ウェブ・コンソーシアム (www.w3c.org) のツールと情報を使用して、Adobe Portable Document Format (PDF) で作成およびフォーマットされました。主要ドキュメント開発ツールは、InterSystemsが構築したCaché と Javaを使用した特別目的のXML処理アプリケーションです。



Caché 製品とロゴは InterSystems Corporation の登録商標です。



Ensemble 製品とロゴは InterSystems Corporation の登録商標です。



InterSystems という名前とロゴは InterSystems Corporation の登録商標です

このドキュメントは、インターシステムズ社(住所: One Memorial Drive, Cambridge, MA 02142)あるいはその子会社が所有する企業秘密および秘密情報を含んでおり、インターシステムズ社の製品を稼動および維持するためにのみ提供される。この発行物のいかなる部分も他の目的のために使用してはならない。また、インターシステムズ社の書面による事前の同意がない限り、本発行物を、いかなる形式、いかなる手段で、その全てまたは一部を、再発行、複製、開示、送付、検索可能なシステムへの保存、あるいは人またはコンピュータ言語への翻訳はしてはならない。

かかるプログラムと関連ドキュメントについて書かれているインターシステムズ社の標準ライセンス契約に記載されている範囲を除き、ここに記載された本ドキュメントとソフトウェアプログラムの複製、使用、廃棄は禁じられている。インターシステムズ社は、ソフトウェアライセンス契約に記載されている事項以外にかかるソフトウェアプログラムに関する説明と保証をするものではない。さらに、かかるソフトウェアに関する、あるいはかかるソフトウェアの使用から起こるいかなる損失、損害に対するインターシステムズ社の責任は、ソフトウェアライセンス契約にある事項に制限される。

前述は、そのコンピュータソフトウェアの使用およびそれによって起こるインターシステムズ社の責任の範囲、制限に関する一般的な概略である。完全な参照情報は、インターシステムズ社の標準ライセンス契約に記載され、そのコピーは要望によって入手することができる。

インターシステムズ社は、本ドキュメントにある誤りに対する責任を放棄する。また、インターシステムズ社は、独自の裁量にて事前通知なしに、本ドキュメントに記載された製品および実行に対する代替と変更を行う権利を有する。

Caché および InterSystems Caché、Caché SQL、Caché ObjectScript および Caché Object は、インターシステムズ社の商標です。

ここで使われている他の全てのブランドまたは製品名は、各社および各組織の商標または登録商標です。

インターシステムズ社の製品に関するサポートやご質問は、以下にお問い合わせください:

InterSystems ワールドワイド カスタマサポート

Tel: +1 617 621-0700

Fax: +1 617 374-9391

Email: support@InterSystems.com

# 目次

1 はじめに .....	1
1.1 Caché SOAP サーバ・アーキテクチャ .....	1
1.2 Caché SOAP クライアント・アーキテクチャ .....	3
2 Web サービスの生成 .....	5
2.1 Web サービスの定義 .....	5
2.1.1 Web サービスのパラメータ .....	6
2.1.2 スタジオを使用した新規の Web サービスの生成 .....	7
2.2 Web メソッドの定義 .....	8
2.2.1 リテラルのメソッド引数 .....	8
2.2.2 オブジェクトのメソッド引数 .....	8
2.3 Web サービスの使用法 .....	9
2.3.1 ブラウザでの Web サービスのテスト .....	9
2.3.2 Java からの Web サービスの使用法 .....	11
2.3.3 .NET からの Web サービスの使用法 .....	11

# テーブル一覧

Web サービスのパラメータ .....	7
----------------------	---

# 1

## はじめに

Caché は、SOAP (Simple Object Access Protocol) に対する完全なサポートを提供します。このサポートは、使いやすく効率的で、SOAP 仕様と完全な互換性があります。このサポートは Caché に組み込まれているので、複雑なミドルウェアやオペレーティング・システムの拡張は必要ありません。Caché にサポートされるすべてのプラットフォームで使用できます。

SOAP は HTTP ベースのプロトコルであるため、アプリケーションはインターネット上でリモート・プロセスの呼び出しを実行できます。SOAP は、XML を使用して、要求、およびそれに伴う応答をコード化します。

Caché SOAP を使用すると、以下を実行できます。

- ・ Web サービス (SOAP プロトコルを使用してクライアント・アプリケーションが呼び出す関連メソッドの集合) を定義し、発行します。他の SOAP と互換性のあるアプリケーションから、これらのメソッドを検索して呼び出すことができます。Caché は、(外部のミドルウェアからではなく) データベースから直接 SOAP メソッドを実行するため、非常に効率的に実行できます。
- ・ ご使用のアプリケーションが、標準 SOAP プロトコルを使用して他のアプリケーションと相互運用する方法を提供します。
- ・ (新規に開発したり、既存のアプリケーション・コードを利用したりして) Caché がサポートするあらゆるプラットフォームで展開できる、非常に効率的な Web サービスを開発します。特定のベンダのオペレーティング・システムやプラットフォームに限定されません。

### 1.1 Caché SOAP サーバ・アーキテクチャ

Caché SOAP サーバは、以下のように動作します。

## はじめに

---

- ・ 実装する Web サービス (関連する SOAP メソッドのセット) に対し、Caché ライブラリで提供される `%SOAP.WebService` クラスを拡張する (つまり、`%SOAP.WebService` から派生する)、新規の Caché クラス定義を作成します。
- ・ このクラス定義内で、ユーザの Web サービスのメソッドに対応する 1 つ以上のクラス・メソッドを定義します。その定義に “WebMethod” キーワードを追加することにより、それぞれを “WebMethod” として指定できます。

また、(基本的には、SQL 文かユーザ定義のコードに基づく組み込まれた結果セットである) クラス・クエリを定義したり、その定義に “WebMethod” キーワードを追加することによって、“objects” の配列を返す Web メソッドを定義することもできます。

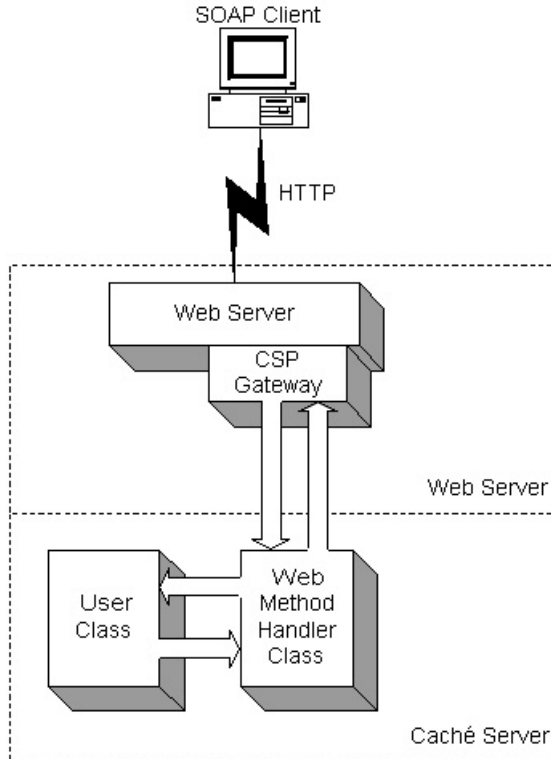
SOAP はステートレス・プロトコルなので、インスタンス・メソッドの呼び出しはサポートしません。したがって、Web メソッドとして使用できるのはクラス・メソッドのみです。しかし、Web メソッド内でも、オブジェクト・インスタンスを生成するコードや、Caché データベース・アクセス、Caché SQL ゲートウェイ経由のリモート・データベース・アクセスのコードなども含め、あらゆるコードを実行できます。

- ・ Web サービス・クラスをコンパイルする際は、Caché クラス・コンパイラは SOAP サービスのコンテンツについてのカタログ情報を自動的に集め、各 Web メソッドに対し SOAP インタフェースを構築します。Web メソッドの SOAP インタフェースは、SOAP 要求を Web メソッドへの特定の呼び出しに変換する作業を実行する生成クラスです。これには、Caché の XML からオブジェクトへの変換技術を使用します。
- ・ Caché は、各 Web サービスに対して、自動的に WSDL (Web Services Description Language) ドキュメントを発行します。WSDL ドキュメントは、XML ドキュメントで、利用可能なメソッドのリストやそのシグニチャ、およびそれらを SOAP クライアントから呼び出す方法の詳細を提供します。WSDL ドキュメントは、CSP (Caché Server Pages) を使用して Web (HTTP) サーバ経由で発行されます。

WSDL ドキュメントは、動的に 供給され、ユーザの Web サービス・クラスのインタフェースのすべての変更を、自動的に反映します。

- ・ SOAP クライアントは、Web サーバから WSDL ドキュメントを要求することによって使用可能な Web サービスを見つけ、次に Web サーバが Caché サーバにこれを要求します。この情報を WSDL で使用すると、SOAP クライアントは (XML エンベロープと呼ばれる) XML メッセージを作成して特定のメソッドを呼び出し、これを (HTTP 経由で) WSDL ドキュメントに指定されている通りに、SOAP サーバに書き込みます。
- ・ Caché SOAP サーバは、Caché (CSP) HTTP ゲートウェイ経由で SOAP 要求を受信します。サーバはメッセージを開封し、検証した後、指定された Web (クラス) メソッドを呼び出します。Caché SOAP サーバは、Web メソッドを呼び出す前に、すべての入力パラメータを Caché に適合する表記に変換します。複雑なタイプの場合、複雑なタイプを表すオブジェクトのインスタンスが生成され、これらを Web メソッドの入力引数として使用します。
- ・ Web メソッドはそのコードを実行し、現在の出力デバイスに XML 応答を書き込みます。この応答は、単純なリテラル値 (文字列など)、XML にコード化されたオブジェクトやオブジェクトのセット、あるいは (SQL クエリなどの) クエリの結果のいずれかです。

以下の図は、SOAP メッセージの実行方法を表します。



## 1.2 Caché SOAP クライアント・アーキテクチャ

Caché は、SOAP クライアントを生成する機能を提供します。SOAP クライアントは、呼び出されたときに SOAP プロトコルを使用して、他のシステム上の Web サービス・メソッドを呼び出すメソッドを含むクラスです。

Caché SOAP クライアントは、以下のように動作します。

- ・ 実装する Web サービス (関連する SOAP メソッドのセット) に対し、Caché ライブラリで提供される `%SOAP.WebClient` を拡張する (つまり、`%SOAP.WebClient` から派生する)、新規の Caché クラス定義を作成します。
- ・ SOAP クライアント・クラスには、接続先の Web サービスのメソッドに対応する 1 つ以上のクラス・メソッドが含まれます。これらのクライアント・メソッドは、“WebMethod” キーワードを追加することにより、それぞれを “WebMethod” として指定できます。

SOAP はステートレス・プロトコルなので、インスタンス・メソッドの呼び出しはサポートしません。したがって、Web メソッドとして使用できるのはクラス・メソッドのみです。しかし、Web メソッド内でも、オブジェクト・インスタンスを生成するコードや、Caché データベース・アクセス、Caché SQL ゲートウェイ経由のリモート・データベース・アクセスのコードなども含め、あらゆるコードを実行できます。

- ・ SOAP クライアント・クラスをコンパイルする際、Caché クラス・コンパイラは SOAP サービスのコンテンツについてのカタログ情報を自動的に集め、各 Web メソッドに対し SOAP クライアント・インタフェースを構築します。Web メソッドの SOAP クライアント・インタフェースは、SOAP 要求を Web メソッドへの特定の呼び出しに変換する作業を実行する生成クラスです。これには、Caché の XML からオブジェクトへの変換技術を使用します。
- ・ SOAP クライアントは、Web サーバから WSDL ドキュメントを要求することによって使用可能な Web サービスを見つけ、次に Web サーバが Caché サーバにこれを要求します。この情報を WSDL で使用すると、SOAP クライアントは (XML エンベロープと呼ばれる) XML メッセージを作成して特定のメソッドを呼び出し、これを (HTTP 経由で) WSDL ドキュメントに指定されている通りに、SOAP サーバに書き込みます。
- ・ SOAP サーバは SOAP 要求を受信し、メッセージをアンパックし、検証し、指定した演算を呼び出します。

# 2

## Web サービスの生成

この章では、Caché で Web サービスを定義する方法、および、このサービスを SOAP クライアントから使用する方法について説明します。

### 2.1 Web サービスの定義

Caché では、すべての Web サービスは、**%SOAP.WebService** クラスのサブクラスによって定義されます。**%SOAP.WebService** ベース・クラスは、SOAP プロトコル経由で呼び出すことのできる 1 つ以上のメソッドを作成するために必要な、すべての機能を提供します。さらに、このクラスは SOAP に関連する“ブックキーピング”（サービスについて説明している WSDL のドキュメントの保持など）の管理を自動化します。

以下は、単純な Caché Web サービスの定義です。この場合、**SOAP.Demo** クラスは、Caché SAMPLES ネームスペースに提供されています。

```
/// A simple demonstration Web Service
Class SOAP.Demo Extends %SOAP.WebService
{
  /// This is the name of our service.
  Parameter SERVICENAME = "SOAPDemo";

  /// This is the URL used to access our service.
  /// Change localhost to an IP address to make this service
  /// available over a network.
  Parameter LOCATION = "http://localhost/csp/samples";

  /// Change this to your own namespace when you deploy a service.
  Parameter NAMESPACE = "http://tempuri.org";

  /// Returns a corporate mission statement.
  ClassMethod Mission() As %String [WebMethod]
  {
    Quit ##class(%PopulateUtils).Mission()
  }

  /// Returns the city name for the given U.S. ZIP Code.
  ClassMethod LookupCity(zip as %String) As %String [WebMethod]
  {
    New SQLCODE, city
    Set city = ""

    // Use embedded SQL to perform the query
    &sql(SELECT City INTO :city
        FROM Sample.USZipCode
        WHERE ZipCode = :zip )

    Quit city
  }
}
```

このクラスは、“SOAPDemo” という名前の Web サービスを定義します。Web サービスには、Mission と LookupCity という 2 種類のメソッドがあります。Mission は、(%PopulateUtils データ生成クラスを使用して実装された) 会社の綱領を含む文字列を返します。一方、LookupCity はアメリカの郵便番号コードのデータベースで、市の名前を検索します (Cache SAMPLES データベースには、デモンストレーション用に、アメリカの北西部の市のデータベースが含まれています)。

この Web サービスをお試しになる場合は、下記の [Web サービスの使用法](#) で説明されている技術をご利用ください。

### 2.1.1 Web サービスのパラメータ

%SOAP.WebService クラスは、Web サービスの特性を指定する、一連のパラメータを定義します。以下のパラメータがあります。

## Web サービスのパラメータ

パラメータ	説明
LOCATION	クライアントが Web サービスを使用する際の URL。この URL は、Web サービスの WSDL ドキュメントに含まれています。ユーザがネットワーク・アドレスとして “localhost” (127.0.0.1) を使用している場合は、リモート・クライアントはそのサービスを受けることができないので、ユーザのサーバがインストールされているマシンの IP アドレスを使用してください。
NAMESPACE	ユーザの Web サービスのネームスペースを定義するのに使用する URI。ネームスペースは、ユーザのサービスやそのコンテンツが、確実に他のサービスと衝突しないようにするために使用します。既定では、“http://tempuri.org” に設定されています。これは、SOAP 開発者が開発中に一時的なネームスペースとして使用する URI です。実行時には、これを変更してください。
SERVICENAME	Web サービスの名前。この名前は、有効な識別子である必要があります。つまり、文字から開始し、英数字でない文字は含まれません。

## 2.1.2 スタジオを使用した新規の Web サービスの生成

Caché スタジオで、簡単に新規 Web サービスを作成できます。

1. スタジオで、[ファイル] メニューの [新規作成] コマンドを呼び出します。
2. [新規作成] ダイアログが表示されます。“新規 Web サービス” を選択し、[OK] ボタンをクリックします。
3. Web サービス・ウィザードが表示されます。クラス名や Web サービスのパラメータの値を入力し (“Web サービス・パラメータ” 参照)、[OK] ボタンをクリックします。

この時点で、以下のような新規の Web サービス・クラスが作成されます。

```
Class MyApp.StockService Extends %SOAP.WebService
{
    Parameter SERVICENAME = "MyStockService";
    Parameter LOCATION = "http://localhost/csp/user";
    Parameter NAMESPACE = "http://tempuri.org";
}
```

ここから、ユーザのクラスに 1 つ以上の Web メソッドを追加できます。これは、以下のセクションで説明します。このサービスをネットワーク全体で使用する前に、LOCATION パラメータのネットワーク・アドレスを、Web サーバの実際のアドレスに変更する必要がありますので注意してください。

## 2.2 Web メソッドの定義

Web メソッドは、WebMethod と示された Caché クラスのクラス・メソッドです。例えば、以下のようになります。

```
Class MyApp.StockService Extends %SOAP.WebService
{
    Parameter SERVICENAME = "MyStockService";
    Parameter LOCATION = "http://localhost/csp/user";
    Parameter NAMESPACE = "http://tempuri.org";

    /// This method returns tomorrow's price for the requested stock
    ClassMethod Forecast(StockName As %String) As %Integer [WebMethod]
    {
        // apply patented, nonlinear, heuristic to find new price
        Set price = $Random(1000)
        Quit price
    }
}
```

このクラスがコンパイルされると、“Forecast” という 1 つのメソッドを含む、“MyStockService” と呼ばれる Web サービスを発行します。

Web メソッドには、データベース・アクセスや埋め込み SQL クエリ、オブジェクト・アクセスなど、あらゆる実装コードを使用できます。唯一の制約条件は、非リテラルの入力パラメータや戻り値が、XML 対応のクラス、つまりスーパークラスとして **%XML.Adaptor** を含むクラスとして表されることです。

### 2.2.1 リテラルのメソッド引数

単純なリテラルのメソッド・パラメータ、および戻り値は、Caché データ型クラスを使用して表します。例えば、入力値として 2 つの整数を持ち、文字列を返す Web メソッドは、以下のよう定義します。

```
ClassMethod MyMethod(p1 As %Integer,p2 As %Integer) As %String [WebMethod]
{
    Quit "A string"
}
```

### 2.2.2 オブジェクトのメソッド引数

複雑で、各種の要素からなるメソッド引数、および戻り値は、オブジェクトを使用して表されます。例えば、非データ型、XML 対応 Caché クラスです。

例えば、入力値として (実際の、および仮想のコンポーネント) 2 つの複雑な数字を持ち、複雑な数字を返す Web メソッドは、以下のよう定義します。

```
ClassMethod Add(a As ComplexNumber,b As ComplexNumber)
    As ComplexNumber [WebMethod]
{
    Set sum = ##class(ComplexNumber).%New()
    Set sum.Real = a.Real + b.Real
    Set sum.Imaginary = a.Imaginary + b.Imaginary

    Quit sum
}
```

この場合、**ComplexNumber** は以下のように定義します。

```
/// A complex number
Class MyApp.ComplexNumber Extends (%RegisteredObject,%XML.Adaptor)
{
    Property Real As %Float;
    Property Imaginary As %Float;
}
```

## 2.3 Web サービスの使用法

Caché で Web サービスを定義した後、様々な方法で試すことができます。そのうちの一部をこの章で説明します。

この章では、Caché に含まれているサンプルの Web サービス・クラスを使用します。

### 2.3.1 ブラウザでの Web サービスのテスト

Caché Web サービスはすべて、単純な HTTP ベースのインタフェースを自動的に提供します。これにより、Web ブラウザを使用してユーザが作成した Web サービスを見つけてテストできます。内部的には、以下のように機能します。各 Web サービス・クラスは、**%SOAP.WebService** クラスから派生します。この **%SOAP.WebService** は、同様に **%CSP.Page** から派生します。したがって、すべての Web サービス・クラスは (CSP 経由の) HTTP 要求に応答できます。**%SOAP.WebService** クラスは、以下を実行するための HTTP イベントに応答するメソッドを実装します。

- Web サービスとそのメソッドについて説明するカタログ・ページを、(HTML を使用して) 人が読める形式で発行します。
- Web サービスの WSDL ドキュメントを、XML ドキュメントで発行します。
- 各 Web サービス・メソッドに対して HTML ベースのテスト・ページを発行することにより、インタラクティブに各メソッドをテストできます。

#### 2.3.1.1 Web サービス・カタログ・ページ

ブラウザで特定の Web サービスの Web サービス・カタログ・ページを表示するには、(CSP の **.cls** ファイル拡張子を使用して) それに対応する Web サービス・クラスの URL を要求します。例えば、

表示したい Web サービス・クラスが **MyApp.StockService** で、使用している Caché サーバの USER ネームスペース内に定義されている場合、以下の URL を使用します。

```
http://localhost/csp/user/MyApp.StockService.cls
```

これは、HTML ページとして **MyApp.StockService** クラスのカタログ・ページを表示します。

Caché に含まれるサンプルの Web サービスのカタログ・ページを見るには、ブラウザで URL `http://localhost/csp/samples/SOAP.Demo.cls` を指定してください。

### 2.3.1.2 WSDL ドキュメント

特定の Web サービスの WSDL ドキュメントを表示するには、(CSP の `.cls` ファイル拡張子を使用して) それに対応する Web サービス・クラスの URL を、URL パラメータ・リストに “WSDL” を含めて要求します。例えば、表示したい Web サービス・クラスが **MyApp.StockService** で、使用している Caché サーバの USER ネームスペース内に定義されている場合、以下の URL を使用します。

```
http://localhost/csp/user/MyApp.StockService.cls?WSDL
```

これは、**MyApp.StockService** クラスの WSDL ドキュメントを、XML ドキュメントで表示します。

Caché に含まれるサンプルの Web サービスの WSDL ドキュメントを見るには、ブラウザで URL `http://localhost/csp/samples/SOAP.Demo.cls` を指定してください。

ユーザの Web サービス用の WSDL ドキュメントを含む静的な XML ファイルを作成するには、Web サービス・クラスの (`%SOAP.WebService` クラスから継承した) `FileWSDL` メソッドを使用します。例えば、以下のようになります。

```
Do ##class(MyApp.StockService).FileWSDL("/wsdl/StockService.xml")
```

### 2.3.1.3 Web メソッドのテスト・ページ

Web メソッドのテスト・ページは、特定の Web メソッドに対する Web ページで、Web メソッドの入力パラメータの値を記入するフォームを表示します。その後、**[送信]** ボタンを使用して、Web メソッドを呼び出します。Web メソッドから返されるデータは、XML としてブラウザ内に表示されます。

ブラウザで特定の Web サービスの Web メソッド・テスト・ページを表示するには、(CSP の `.cls` ファイル拡張子を使用して) それに対応する Web サービス・クラスの後、“/” とメソッド名を付けた URL を要求します。メソッド名の後に、URL パラメータとしてあらゆるメソッド引数を追加できます。

例えば、表示したいメソッドが `Forecast` であり、ユーザの Web サービス・メソッドが **MyApp.StockService** で、使用している Caché サーバの USER ネームスペース内に定義されている場合、以下の URL を使用します。

```
http://localhost/csp/user/MyApp.StockService.cls/Forecast
```

これは、**MyApp.StockService** クラスの `Forecast` メソッドに対する Web メソッド・テスト・ページを表示します。

Caché に含まれるサンプルの Web サービスの Web メソッド・テスト・ページを見るには、ブラウザで URL `http://localhost/csp/samples/SOAP.Demo.cls` を指定し、それぞれの Web メソッド・テスト・ページへのリンクをクリックしてください。

## 2.3.2 Java からの Web サービスの使用法

Java で利用できる SOAP クライアント・クラスを使用して、Java から Web サービス・メソッドを呼び出すことができます。

## 2.3.3 .NET からの Web サービスの使用法

.NET フレームワークに含まれる SOAP クライアント・クラスを使用して、.NET から Web サービス・メソッドを呼び出すことができます。SOAP は公開された標準プロトコルなので、Web サービスが実装されている場所や方法、プラットフォームは、SOAP クライアントには関係ありません。

例えば、(SAMPLES データベースに含まれる) サンプルの Web サービス “SOAPDemo” を、Visual Basic .NET から使用するとします。

最初に、ユーザの Visual Basic プロジェクトに Web 参照を追加します。これは、(WSDL ドキュメント経由で) SOAP クライアントに、任意の Web サービスと相互に作用するために必要な情報を与えます。Visual Studio .NET 内に Web 参照を追加するには、[プロジェクト] メニューの [Web 参照の追加] コマンドを呼び出します。Web サービスに対する WSDL ドキュメントのアドレス (URL) を入力するように指示されます。この場合は、以下の URL を [URL] フィールドに入力します。

```
http://localhost/csp/samples/SOAP.Demo.cls?WSDL
```

これは、SOAPDemo サービス用の WSDL ドキュメントを提供します。

次に、SOAP クライアント・インタフェース・オブジェクトを経由して、ご使用のクライアント・アプリケーションで Web サービスを使用します。この場合、SOAP クライアント・オブジェクトは **SOAPDemo** クラスのインスタンスです。以下の Visual Basic のコードは、SOAP クライアント・インタフェースである `tMyService` のインスタンスを生成し、“SOAPDemo” Web サービスのメソッドを呼び出し、メッセージ・ボックスにその結果を表示します。

```
'tMyService is an instance of SOAPDemo
'Show a mission statement
MsgBox.Show(tMyService.Mission())

'Loopkup up a city for a zipcode
MsgBox.Show(tMyService.LookupCity("02139"))
```

同様に、.NET 言語からも同じことを実行できます。

