

# Lab Manager SOAP API リファレンス

vCenter Lab Manager 4.0

JA-000175-00



最新のテクニカルドキュメントに関する情報は、当社の Web サイトから入手できます。

<http://www.vmware.com/jp/support/>

当社の Web サイトは、最新の製品アップデートも提供します。

本書に関するコメントがございましたら、次のメールアドレスまでフィードバックをお寄せください。

[docfeedback@vmware.com](mailto:docfeedback@vmware.com)

© 2006-2009 VMware, Inc. All rights reserved. この製品は、米国および国際的な著作権法および知的財産法によって保護されています。VMware 製品は、<http://www.vmware.com/go/patents> のリストに表示されている 1 つまたは複数の特許の対象です。

VMware、VMware "boxes" のロゴおよびデザイン、Virtual SMP、および VMotion は、米国およびその他の地域における VMware, Inc. の登録商標または商標です。その他のマークおよび名前は、それぞれの企業の商標です。

Lab Manager は、サードパーティ製品の AxpDataGrid と一緒に配布されます。AxpDataGrid の著作権は、ノルウェーのオスロに拠点を置く Axezz (<http://www.axezz.com/axpdatagrid>) に帰属します。

**VMware, Inc.**

3401 Hillview Ave.  
Palo Alto, CA 94304  
[www.vmware.com](http://www.vmware.com)

# 目次

本書について	7
<b>1 VMware vCenter Lab Manager SOAP API の概要</b>	<b>9</b>
Lab Manager と自動テスト ツールの統合	9
サポートされている操作	10
Lab Manager のデータオブジェクト	10
標準への準拠と互換性のある開発プラットフォーム	10
SOAP API のセキュリティ	10
ユーザー認証	10
<b>2 Lab Manager SOAP API スタート</b>	<b>11</b>
ガイド	11
アプリケーション開発の要件	11
WSDL ファイルの取得とインポート	11
開発環境への WSDL ファイルのインポート	11
Microsoft Visual Studio での Lab Manager WSDL ファイルの使用	12
C# コードの簡単なサンプルと高度なサンプル	13
C# での簡単なコンソールアプリケーションのサンプル	13
Lab Manager と Quality Center を統合する C# の高度なサンプル	15
<b>3 Lab Manager の API データ型</b>	<b>19</b>
プリミティブ XML データ型	19
Lab Manager のデータ型	19
AuthenticationHeader	20
サポートされている API 呼び出し	20
フィールド	20
C# サンプルコード	20
Configuration	21
Machine	21
<b>4 Lab Manager API メソッド リファレンス</b>	<b>23</b>
ConfigurationCapture	24
構文	24
引数	24
応答	24
C# サンプルコード	24
ConfigurationCheckout	25
構文	25
引数	25
応答	25
C# サンプルコード	25
ConfigurationClone	26
構文	26
引数	26
応答	26
C# サンプルコード	26
ConfigurationDelete	27

構文	27
引数	27
応答	27
C# サンプルコード	27
ConfigurationDeploy	28
構文	28
引数	28
応答	28
C# サンプルコード	28
ConfigurationPerformAction	29
構文	29
引数	29
応答	29
C# サンプルコード	29
ConfigurationSetPublicPrivate	30
構文	30
引数	30
応答	30
C# サンプルコード	30
ConfigurationUndeploy	31
構文	31
引数	31
応答	31
C# サンプルコード	31
GetConfiguration	32
構文	32
応答	32
C# サンプルコード	32
GetConfigurationByName	33
構文	33
引数	33
応答	33
C# サンプルコード	33
GetCurrentOrganizationName	34
構文	34
応答	34
C# サンプルコード	34
GetCurrentWorkspaceName	35
構文	35
応答	35
C# サンプルコード	35
GetMachine	35
構文	35
引数	35
応答	35
C# サンプルコード	36
GetMachineByName	37
構文	37
引数	37
応答	37
C# サンプルコード	37
GetSingleConfigurationByName	38
構文	38
引数	38
応答	38
C# サンプルコード	38

ListConfigurations	39
構文	39
引数	39
応答	39
C# サンプルコード	39
ListMachines	40
構文	40
引数	40
応答	40
C# サンプルコード	40
LiveLink	41
構文	41
引数	41
応答	41
C# サンプルコード	41
MachinePerformAction	42
構文	42
引数	42
応答	42
C# サンプルコード	42
SetCurrentOrganizationByName	43
構文	43
引数	43
C# サンプルコード	43
SetCurrentWorkspaceByName	44
構文	44
引数	44
C# サンプルコード	44
索引	45



# 本書について

---

VMware® 『vCenter Lab Manager SOAP API ガイド』は、Lab Manager Web サービス データを使用したり、タスクを自動化したり、Lab Manager を他のソフトウェア テスト ツールと統合したりするアプリケーションを開発する場合にご利用ください。

## 対象読者

このガイドは、Lab Manager データを使用してテストソリューションをカスタマイズすること、または環境内の Lab Manager と他のソフトウェア テストツールを統合することを考えている開発者を対象としています。たとえば、Lab Manager SOAP API を使用すると、Lab Manager と自動化されたソフトウェアテストツールを統合できます。

このガイドの情報を使用するには、次のことをよく理解している必要があります。

- 仮想マシン技術
- 分散した複数階層のシステムの概念
- 開発とテストの実務
- Windows または Linux オペレーティングシステム
- Web サービス、SOAP、および XML

## 本書へのフィードバック

ドキュメントの向上にご協力ください。本書に関するコメントがございましたら、メールアドレス [docfeedback@vmware.com](mailto:docfeedback@vmware.com) までフィードバックをお寄せください。

## テクニカルサポートおよび教育リソース

ここでは、お客様にご利用いただけるテクニカルサポートリソースについて説明します。本書およびその他の文書の最新バージョンは、<http://www.vmware.com/jp/support/pubs> でご覧いただけます。

## オンラインサポートおよび電話によるサポート

テクニカルサポートリクエストの提出や、製品および契約情報の確認、製品の登録をオンラインで行うには、<http://www.vmware.com/jp/support/> をご覧ください。

該当するサポート契約を結んでいるお客様の場合、迅速な対応が必要な Severity1 の問題に関しては電話でのサポートをご利用ください。詳細については、[http://www.vmware.com/support/phone\\_support](http://www.vmware.com/support/phone_support) をご覧ください。

## サポートサービス

お客様のビジネスニーズに適した各種サポートの詳細については、  
<http://www.vmware.com/jp/support/services/> をご覧ください。

## VMware 教育プロフェッショナルサービス

VMware 教育サービスのコースでは、広範なハンズオン ラボやケーススタディをご紹介します。また、業務の際のリファレンスとしてお使いいただける資料も提供しています。トレーニングは、オンサイト、講義形式、およびライブオンラインで受講できます。オンサイトのパイロットプログラムおよび実装のベストプラクティスについては、VMware コンサルティングサービスがご使用の仮想環境の評価、計画、構築、および管理に役立つサービスを提供しています。教育トレーニング、認定プログラム、およびコンサルティングサービスについては、<http://www.vmware.com/jp/services/> をご覧ください。



# VMware vCenter Lab Manager SOAP API の概要

# 1

Lab Manager SOAP アプリケーションプログラミング インターフェイス (API) を使用すると、Lab Manager システムにプログラムでアクセスできます。安全な API を使用することで、Lab Manager サーバに接続して、さまざまな操作を自動化したり実行したりできます。

Lab Manager SOAP API は、通信プロトコルとして SOAP などの XML ベースの技術を使用し、インターフェイス記述言語として Web Services Description Language (WSDL) を使用します。Lab Manager の WSDL ファイルでは、サービスの使用可能なメソッド (Web サービスでは操作と呼ばれます)、パラメータの型、およびサービスの SOAP エンドポイントが詳細に記述されています。

---

**注** 構成、仮想マシン、またはテンプレートオブジェクトの名前は、大文字と小文字を区別されません。

---

ここでは、以下のトピックについて説明します。

- 「[Lab Manager と自動テスト ツールの統合](#)」 (P. 9)
- 「[サポートされている操作](#)」 (P. 10)
- 「[Lab Manager のデータオブジェクト](#)」 (P. 10)
- 「[標準への準拠と互換性のある開発プラットフォーム](#)」 (P. 10)
- 「[SOAP API のセキュリティ](#)」 (P. 10)
- 「[ユーザー認証](#)」 (P. 10)

## Lab Manager と自動テスト ツールの統合

Lab Manager SOAP API を利用すると、選択した言語とプラットフォームを使用して、Lab Manager のデータと対話できます。このガイドの例では、C# プログラミング言語と Microsoft .NET Framework を使用しますが、他のプログラミング言語と開発環境もサポートされます。C# 以外の言語を使用する場合は、ご使用の開発環境のドキュメントで Web サービス アプリケーションの開発に関する情報を参照してください。

SOAP API を使用して Lab Manager を拡張またはカスタマイズするだけでなく、Lab Manager と自動テストシステムを統合することもできます。この統合の例については、「[Lab Manager と Quality Center を統合する C# の高度なサンプル](#)」 (P. 15) を参照してください。

Lab Manager ソリューション、開発者向けリソース、およびコミュニティリソースの詳細については、<http://www.vmware.com> を参照してください。

## サポートされている操作

好みの Web 対応開発環境を使用し、標準の Web サービスプロトコルを使用して Web サービスクライアントアプリケーションを作成することで、次のタスクをプログラムで実行できます。

- 仮想マシンおよび構成情報を照会する。
- マシンおよび構成に対するアクションを実行する。
- 構成を取得、チェックアウト、クローン作成、削除、およびデプロイする。
- 他のチーム メンバに電子メールで送信できる LiveLink 構成 URL を作成する。

サポートされる Web サービス操作についての詳細は、「[Lab Manager API メソッド リファレンス](#)」(P. 23) を参照してください。

## Lab Manager のデータオブジェクト

Lab Manager SOAP API は、オブジェクトを使用して、組織内のデータと対話します。オブジェクトとは、Lab Manager データのプログラムでの表現です。オブジェクトのプロパティは、データ エンティティでのフィールドを表します。たとえば、Lab Manager の構成は Configuration オブジェクトで表され、これには構成名、構成の数値 ID、デプロイ ステータス、共有状態などを表すフィールドが含まれます。

このドキュメントでは、Lab Manager のデータ オブジェクトを使用して、クエリ、クローン作成、取得、デプロイなどの操作を Lab Manager データに対して実行する方法を説明します。「[Lab Manager の API データ型](#)」(P. 19) を参照してください。

## 標準への準拠と互換性のある開発プラットフォーム

Lab Manager SOAP API は、SOAP 1.1、WSDL 1.1、および WS-I Basic Profile Version 1.1 で示されている他の標準に準拠します。Lab Manager SOAP API は、Basic Profile Version 1.1 標準に準拠する現在の SOAP 開発環境で動作します。このドキュメントの例では、Microsoft Visual Studio .NET 2005 開発環境および C# プログラミング言語を使用します。

---

**注** 特定の開発プラットフォームでの実装の違いにより、Lab Manager SOAP API の一部または全部の機能にアクセスできない場合があります。

---

.NET の開発に Visual Studio を使用する場合は、Visual Studio 2005 以降を使用することをお勧めします。

## SOAP API のセキュリティ

組織内の Lab Manager データにアクセスするクライアント アプリケーションは、Lab Manager Web コンソールで使用されるものと同じセキュリティ保護の対象になります。Lab Manager は SSL を使用してすべての SOAP API メソッドを公開します。

Web サービス URL を使用して Web サービス URL の SOAP API にアクセスすると、SSL 証明書の警告が表示される場合があります。証明書を受け入れて API を使用するか、または証明書を有効な署名の証明書に置き換えてください。

## ユーザー認証

クライアントアプリケーションは、Lab Manager Web サービスメソッドを呼び出すたびに、有効なユーザー認証情報を提供する必要があります。必要な認証情報は、Lab Manager のユーザーアカウント、パスワード、組織、およびワークスペース名です。操作を実行するオブジェクトを含む組織とワークスペースのどちらか一方または両方の名前を指定する必要があります。Lab Manager サーバはこれらの認証情報を認証します。

---

**注** 管理者以外の認証情報を使用できます。

---

# Lab Manager SOAP API スタートガイド

# 2

Lab Manager SOAP API を使用して、XML Web サービス クライアントを作成できます。XML Web サービス クライアントは、XML Web サービスを参照して使用するコンポーネントまたはアプリケーションです。これにはクライアントベースのアプリケーションは必要ありません。通常、XML Web サービス クライアントは、Web フォームや他の XML Web サービスなどの、他の Web アプリケーションです。

ここでは、以下のトピックについて説明します。

- 「アプリケーション開発の要件」 (P. 11)
- 「WSDL ファイルの取得とインポート」 (P. 11)
- 「C# コードの簡単なサンプルと高度なサンプル」 (P. 13)

## アプリケーション開発の要件

アプリケーションの開発を始める前に、次の要件を確認してください。

- 読者は基本的なプログラミング概念に習熟し、コンピュータにはプログラミング開発環境がすでに用意されているものとします。
- Lab Manager のインスタンスをインストールして構成し、ネットワークで実行している必要があります。
- 完全修飾ホスト名または IP アドレスで始まる Lab Manager サーバ インスタンスのアドレスがわかっている必要があります (例: <https://hostname.company.com/LabManager>)。
- ターゲットの Lab Manager サーバにアカウントが必要です。
- 次のセクションで示されているコード リストをコピーし、Microsoft Visual Studio 2005 環境に貼り付けてください。

## WSDL ファイルの取得とインポート

他の標準ベースの SOAP API 実装と同様に、Lab Manager API の定義は、XML 形式の WSDL ファイルとして Web サービスで使用できます。このファイルを取得するには、Internet Explorer 5.5 以降を開き、ご使用の Lab Manager サーバの <https://<hostname>/LabManager/SOAP/LabManager.asmx?WSDL> に移動します。

WSDL ファイルでは、Lab Manager API のすべての呼び出しとオブジェクトが定義されています。WSDL の詳細については、<http://www.w3.org/TR/wsdl> を参照してください。

## 開発環境への WSDL ファイルのインポート

WSDL ファイルを取得した後は、それを開発環境にインポートして、クライアント Web サービス アプリケーションの構築で使用するために必要なオブジェクトを生成します。プロセスは、開発環境、プログラミング言語、および関連するツールに依存します。たとえば、Microsoft Visual Studio 開発環境ではタスクは自動的に処理されます。他の開発プラットフォームに関しては、プラットフォームの製品ドキュメントを参照してください。

## Microsoft Visual Studio での Lab Manager WSDL ファイルの使用

Microsoft Visual Studio のプログラミング言語は、サーバ側の対応部分に対するプロキシとして機能するオブジェクトを通して Lab Manager SOAP API にアクセスします。管理対象コードが XML Web サービスにアクセスするときは、プロキシ クラスと .NET Framework がインフラストラクチャのすべてのコーディングを処理します。

Lab Manager SOAP API を Visual Studio で使用する場合は、最初に WSDL ファイルからプロキシ クラス オブジェクトを生成する必要があります。Visual Studio には、Web サービスに接続して必要な成果物を生成する Web 参照の追加ウィザードが用意されています。既存のアプリケーションに Web 参照を追加することも、Visual Studio で新しいアプリケーションを作成することもできます。

Visual Studio のドキュメントで「Adding and Removing Web References」を参照してください。

### Microsoft Visual Studio 2005 を使用して Web 参照を追加するには

- 1 Windows で、[スタート]>[Microsoft Visual Studio .NET 2005] を選択します。
- 2 [新しいプロジェクト] を選択して新しいプロジェクトを作成するか、[ファイルを開く] を選択して既存のプロジェクトを開きます。
- 3 [URL] テキスト ボックスに「<https://<hostname>/LabManager/SOAP/LabManager.asmx>」と入力して、Lab Manager Web サービスのサービス記述を取得します。
- 4 [実行] をクリックします。

Lab Manager サーバと開発環境クライアントの間で証明書の交換が開始します。サーバから送信された証明書の詳細がセキュリティアラートに表示されます。

---

**注** このセキュリティアラート メッセージは、Lab Manager サーバがデフォルトの自己署名証明書を使用している場合に生成されます。Lab Manager サーバで、これらの証明書を Verisign、Thawte、または他の認証局から入手した証明書に置き換えることができます。

---

- 5 [はい] をクリックします。
- 6 (オプション) Visual Studio 環境からのアラートが表示される場合は、[はい] をクリックします。  
Microsoft Visual Studio 環境が Web サービスのエンドポイントに接続し、Lab Manager Web サービスの WSDL ファイルで記述されている操作を表示します。
- 7 Web 参照名のテキスト ボックスでテキストを選択し、「**LabManager**」と入力します。これは、この Web 参照に使用するネームスペースです。  
**LabManager** は、スペースを含まない 1 語です。
- 8 [参照の追加] をクリックします。
- 9 (オプション) 証明書の警告メッセージが表示される場合は、[はい] をクリックします。
- 10 [はい] をクリックします。

Visual Studio は、サービスの記述を取得し、アプリケーションから Lab Manager Web サービスへのインターフェイスとして機能する LabManager プロキシ クラスを生成します。プロセスの最後に、クラスがプロジェクトの Web References フォルダに追加されます (ソリューションエクスプローラ をクリックし、Web References フォルダに表示される LabManagerSoap を確認してください)。

この基本的なセットアップ タスクが完了すると、Lab Manager SOAP API を使用するクライアントアプリケーションを構築できます。API に早く慣れるには、「[C# コードの簡単なサンプルと高度なサンプル](#)」(P. 13) で示されているコード サンプルを調べてください。

## C# コードの簡単なサンプルと高度なサンプル

「アプリケーション開発の要件」(P. 11) の設定要件を確認してください。

C# での簡単なコンソールアプリケーションのサンプルを使用して、開発ワークステーションと Lab Manager Web サービスの間の基本的な API プログラミング接続をテストできます。C# 以外のプログラミング言語を使用し、Microsoft Visual Studio 2005 以外の Web サービス開発環境を使用している場合は、該当するドキュメントで詳細を参照してください。

---

**注** 読者は基本的なプログラミング概念に習熟し、コンピュータにはプログラミング開発環境がすでに用意されているものとします。

---

このコードサンプルはいくつかの簡単なタスクを実行します。最初の 2 つのタスクは、Lab Manager Web サービスを呼び出すすべてのアプリケーションに必要なものです。

- Lab Manager SOAP API にバインドします。
- SOAP 呼び出しを行うためのユーザー名とパスワードを設定します。
- ServicePointManager 証明書ポリシーを設定し、SSL 証明書を受け入れます。接続するには、すべての証明書を受け入れるように証明書ポリシーを設定する必要があります。
- 呼び出しを行って、オブジェクト名に基づく構成オブジェクトを取得します。
- コンソールにすべての構成フィールドを表示します。

### C# での簡単なコンソールアプリケーションのサンプル

このサンプルコードをコピーし、Microsoft Visual Studio 2005 環境に貼り付けてください。

```
using System;
using System.Net;

namespace LMConsoleApplication1
{
    class Class1
    {
        /// <summary>
        /// The main entry point for the application.
        /// </summary>
        [STAThread]
        static void Main(string[] args)
        {
            try
            {
                //
                /** Bind to the Lab Manager SOAP API
                //
                LabManagerSoap.LabManagerSOAPInterface binding =
                new LabManagerSoap.LabManagerSOAPInterface();
                //
                /** Enter the URL for your system here
                //
                binding.Url = "https://10.6.1.248/LabManager/SOAP/LabManager.asmx";
                binding.Timeout = 10 * 60 * 1000; // 10 minutes
                ServicePointManager.CertificatePolicy = new CertificateAcceptor();

                /**
                /** Allocate AuthenticationHeader object to hold caller's
                /** user name and password
                /**
                binding.AuthenticationHeaderValue = new
                LabManagerSoap.AuthenticationHeader();

                //
            }
        }
    }
}
```

```

    /** Substitute a real user's user name, password, organization, and workspace
        name here
    //
    binding.AuthenticationHeaderValue.username = "jaya";
    binding.AuthenticationHeaderValue.password = "Lab Manager";
    binding.AuthenticationHeaderValue.organizationname = "MyOrg";
    binding.AuthenticationHeaderValue.workspace = "My'Workspace";

    /**
    /** Call GetSingleConfigurationByName()
    /** Get default configuration that comes with Lab Manager
    /** installation and write all property values to console
    /**
    LabManagerSoap.Configuration defCfg=
        binding.GetSingleConfigurationByName("Sample Configuration");
    //
    /** Print out configuration properties to the Console
    //
    Console.WriteLine("Name = " + defCfg.name);
    Console.WriteLine("ID = " + defCfg.id.ToString());
    Console.WriteLine("Description = "+ defCfg.description);
    Console.WriteLine("isPublic = "+ defCfg.isPublic.ToString());
    Console.WriteLine("isDeployed = "+ defCfg.isDeployed.ToString());
    Console.WriteLine("fenceMode = "+ defCfg.fenceMode.ToString());
    Console.WriteLine("type = " + defCfg.type.ToString());
    Console.WriteLine("owner = " + defCfg.owner);
    Console.WriteLine("dateCreated = " +
        defCfg.dateCreated.ToString());
    Console.ReadLine();
    }
    catch (Exception e)
    {
        Console.WriteLine("Error:" + e.Message);
        Console.ReadLine();
    }
    } /** end Main
} /** end Class1

/// <summary>
/// This class is needed to automatically accept the SSL certificate
/// the Lab Manager sends on each API call.
/// </summary>

public class CertificateAcceptor :System.Net.ICertificatePolicy
{
    public CertificateAcceptor() {}

    public bool CheckValidationResult(
        System.Net.ServicePoint servicePoint,
        System.Security.Cryptography.X509Certificates.X509Certificate cert,
        System.Net.WebRequest webRequest, int iProblem)
    {
        return true;
    }
}
} /** end Namespace}

```

## Lab Manager と Quality Center を統合する C# の高度なサンプル

このセクションの C# .NET のサンプルは、Lab Manager SOAP API を使用するさらに包括的で実践的なサンプルです。このコードサンプルでは、Lab Manager SOAP API 呼び出しと Mercury Interactive Corporation Quality Center 製品の統合を示します。このサンプルコードは下記のタスクを実行します。

- Lab Manager API (Lab Manager SOAP API) の呼び出しを行って、ライブラリから構成をチェックアウトし、デプロイします。
- Mercury Quality Center を使用して、デプロイされた構成で事前に定義されている一連のテストを実行します。
- Lab Manager SOAP API 呼び出しを行って、構成を取得し、ワークスペースからアンデプロイします。

下記のタスクは、下記のメソッドを使用するサンプルコードで実行されます。

- **CheckoutDeployConfiguration()** — ライブラリから構成を取得し、Lab Manager ワークスペースにデプロイします。
- **RunQCTestset()** — Mercury Interactive Quality Center の定義済みの一連のテストを実行します。定義済みのテストの詳細については、Mercury Interactive Quality Center のドキュメントを参照してください。
- **CaptureUndeployConfiguration()** — 構成をアンデプロイし、ライブラリに取得します。

さらに、GetLMAPI() メソッドは、Lab Manager API への新しいバインドを作成し、認証パラメータを設定します。このメソッドは、すべての証明書をプログラムで受け入れるように、.NET サービス ポイントマネージャの証明書ポリシーを構成します。GetLMAPI() は、Lab Manager のバインドのインスタンスを返します。

このサンプルコードをコピーし、Microsoft Visual Studio 2005 環境に貼り付けてください。

```
using System;
using System.Configuration;
using System.Collections.Specialized;
using System.IO;
using System.Net;
using TDAPIOLELib; /** From Mercury Quality Center

namespace MATRun
{
    /// <summary>
    /// Class1 comprises methods to check out a configuration from the Lab
    /// Manager Library and deploy it to the Workspace; execute several
    /// tests; and capture a configuration.
    /// </summary>
    class Class1
    {
        /// <summary>
        /// The main entry point for the application.
        /// </summary>
        [STAThread]

        static void Main(string[] args)
        {
            NameValueCollection settings=ConfigurationSettings.AppSettings;
            string filename      = null;
            string buildlocation = null;
            string buildversion  = null;

            if ( args.Length > 0 )
            {
                buildlocation = args[0];
                buildversion  = args[1];
            }

            if ( buildlocation == null )
            {
                buildlocation =
                    @"\\fs.labmanger.com\public\build\outputdir\1423\artifacts";
                buildversion = "Lab Manager-2.0.4018";
            }
        }
    }
}
```

```

    }
    filename =
        @"\\fs.labmanager.com\public\build\build-to-test.bat";
    StreamWriter f = new StreamWriter(filename);
    f.WriteLine(String.Format(@"xcopy {0}\setup.exe c:\ /Y",
        buildlocation));
    f.Close();
    Console.WriteLine(String.Format("Testing {0} at location {1}",
        buildversion, buildlocation));

    string config = CheckoutDeployConfiguration(buildversion);
    RunQCTestset();
    CaptureUndeployConfiguration(config);

} /** End Main() method

//
/** Initialize parameters
//
static string library_config = "ProofOfBuild-R2";
static string storage_server = "LM Server";
static string perform_capture = "Yes";
static string soap_server = "LM Server";

///

```



```

        TDAPIOLELib.List testlist;
        testlist = tsf.NewList("");
        foreach ( TSTest test in testlist)
        {
            TDAPIOLELib.Run r= (Run) test.LastRun;
            if (r != null)
            {
                Console.WriteLine(test.Name + " " + r.Name + " " +
                    r.Status.ToString());
            }
        } /*** end foreach
        break;
    } /*** end if
} /*** end foreach
} /*** end if
} /*** end RunQCTestset

///

```

```

static void CaptureUndeployConfiguration(string configname)
{
    //
    /** Bind to Lab Manager SOAP Web Service
    //
    LabManagerSoap.LabManagerSOAPinterface binding = GetLMAPI();
    LabManagerSoap.Configuration config =
        binding.GetSingleConfigurationByName(configname);
    if ( perform_capture.Equals("Yes") )
    {
        Console.WriteLine("Capture configuration "+ configname);
        int newConfigCaptureID = binding.ConfigurationCapture(config.id,
            configname);
    }
    Console.WriteLine("Undeploy configuration "+ configname);
    binding.ConfigurationUndeploy(config.id);
    Console.WriteLine("Undeploy is completed");
}

/// <summary>
///The GetLMAPI() method creates a new binding to the Lab Manager API
///and sets up authentication and other basic parameters.This method
///returns a CertificateAcceptor object.
/// </summary>

static LabManagerSoap.LabManagerSOAPinterface GetLMAPI()
{
    //
    /** Bind to SOAP interface
    //
    LabManagerSoap.LabManagerSOAPinterface binding = new
    LabManagerSoap.LabManagerSOAPinterface();
    //
    /**Allocate caller login object
    //
    binding.AuthenticationHeaderValue = new
        LabManagerSoap.AuthenticationHeader();
    binding.Url = binding.Url.Replace("https://qa240.VMware.com",
        "https://demo44.VMware.com");
    binding.AuthenticationHeaderValue.username = "jaya";
    binding.AuthenticationHeaderValue.password = "vlm";
    binding.AuthenticationHeaderValue.organizationname = "MyOrg";
    binding.AuthenticationHeaderValue.workspaceName = "MyWorkspace";
    binding.Timeout = 10 * 60 * 1000; // 10 minutes

    ServicePointManager.CertificatePolicy = new CertificateAcceptor();
    return binding; /** return binding reference
}

/// <summary>
/// The CertificateAcceptor class automatically accepts the SSL
/// certificate sent by Lab Manager with each API call from a client
/// application.
/// </summary>

public class CertificateAcceptor :System.Net.ICertificatePolicy
{
    public CertificateAcceptor() {}

    public bool CheckValidationResult(
        System.Net.ServicePoint servicePoint,
        System.Security.Cryptography.X509Certificates.X509Certificate cert,
        System.Net.WebRequest webRequest, int iProblem)
    {
        return true;
    }
}
} //end CertificateAcceptor class declaration
} //end namespace declaration

```

## Lab Manager の API データ型

この章では、Lab Manager の API データ型に関する詳細情報を提供します。

ここでは、以下のトピックについて説明します。

- 「[プリミティブ XML データ型](#)」 (P. 19)
- 「[Lab Manager のデータ型](#)」 (P. 19)
- 「[AuthenticationHeader](#)」 (P. 20)
- 「[Configuration](#)」 (P. 21)
- 「[Machine](#)」 (P. 21)

### プリミティブ XML データ型

Lab Manager の SOAP API データ型は、[テーブル 3-1](#) に示すプリミティブ XML データ型に基づいています。これらのプリミティブ データ型は、Lab Manager の API 呼び出しに使用される Lab Manager データ型の構成要素です。

**表 3-1.** Lab Manager SOAP API のプリミティブ XML データ型

値	説明
xsd:Boolean	true、false、0、および 1 を取る論理値。
xsd:date	日付値。
xsd:dateTime	日付 / 時刻値 (タイムスタンプ)。
xsd:double	標準の IEEE 754-1985 で定義されている IEEE 倍精度 64 ビット浮動小数点数型に対応する数値。
xsd:int	-2147483648 ~ 2147483647 の数値。
xsd:string	任意の文字データ。

### Lab Manager のデータ型

クライアント アプリケーションを記述するときには、使用するプログラミング言語および開発環境向けに定義されたデータ型指定規則に従ってください。プログラミング言語で型指定したデータは、開発ツールによって上記の SOAP データ型にマッピングされます。

Lab Manager のデータ型は、Lab Manager WSDL ファイルで定義されます。詳細については、「[Lab Manager の SOAP API データ型](#)」 (P. 20) を参照してください。

表 3-2. Lab Manager の SOAP API データ型

データ型	説明
AuthenticationHeader	呼び出し元のユーザー名、パスワード、組織、およびワークスペース名を格納します。このデータ型は、Lab Manager Web サービス メソッドのすべての SOAP ヘッダーの一部です。
Configuration	構成オブジェクト。
Machine	マシン オブジェクト。

## AuthenticationHeader

このデータ型は、呼び出し元のユーザー名、パスワード、組織、およびワークスペース名を、すべての Lab Manager SOAP API メソッドに渡します。

### サポートされている API 呼び出し

このデータ型は、すべての API 呼び出しをサポートします。

### フィールド

表 3-3. AuthenticationHeader フィールド

フィールド	データ型	説明
organizationname	string	Lab Manager の組織の名前。
password	string	Lab Manager のアカウントのパスワード。
username	string	Lab Manager のアカウントのユーザー名。
workspacename	string	Lab Manager のワークスペースの名前。

## C# サンプルコード

```
/**
** Visual Studio Console application in C#
** LMsoap = Web reference name for LM Web service
** Set up login code for all LM Web service method calls
**
**/
try
{
    LabManagerSoap.LabManagerSOAPinterface binding = new
        LabManagerSoap.LabManagerSOAPinterface();
    binding.AuthenticationHeaderValue =new
        LabManagerSoap.AuthenticationHeader();
    binding.AuthenticationHeaderValue.username = "hedley";
    binding.AuthenticationHeaderValue.password = "Lab Manager";
    binding.AuthenticationHeaderValue.organizationname = "MyOrg";
    binding.AuthenticationHeaderValue.workspacename = "MyWorkspace";

    ServicePointManager.CertificatePolicy = new CertificateAcceptor();
}
catch (Exception e)
{
    Console.WriteLine("Error:" + e.Message);
    Console.ReadLine();
}
```

## Configuration

このデータ型は、Lab Manager の構成ライブラリまたはワークスペースの構成ごとに存在します。構成は、仮想マシンとそのオペレーティングシステム、アプリケーション、およびデータのグループで、Lab Manager によって1つの単位として制御されます。

整数の ID フィールドは、構成を一意に識別します。構成名は、必ずしも一意ではありません。

表 3-4. Configuration フィールド

フィールド	データ型	説明
dateCreated	dateTime	構成の作成日。
description	string	構成の説明。
fenceMode	int	1 = フェンスなし。 2 = フェンスあり。送受信トラフィックをブロックします。 3 = フェンスあり。送信トラフィックのみを許可します。 4 = フェンスあり。送受信トラフィックを許可します。
id	int	構成 ID。
isDeployed	boolean	デプロイされている場合は true。デプロイされていない場合は false。
isPublic	boolean	他のユーザーが表示およびアクセスできる場合は true。できない場合は false。
name	string	構成名。
owner	string	所有者のユーザー名。
type	int	構成タイプ。 1 = ワークスペース構成。 2 = キャプチャ ライブラリ構成。 3 = ゴールド マスタ キャプチャ ライブラリ構成。 4 = アーカイブ ライブラリ構成。
autoDeleteInMiliSeconds	double	構成が削除されるまでの時間。0 を入力した場合、構成はいつまでも削除されません。
bucketName	string	構成がワークスペースに存在する場合はワークスペース名を表示し、構成がライブラリに存在する場合は組織名を表示します。
mustBeFenced	SOAPUtils.SOAPMustBeFenced	「フェンスあり」または「フェンスなし」のいずれかの場合は NotSpecified。 True = 「フェンスあり」のみの場合。 False = 「フェンスなし」のみの場合。
autoDeleteDateTime	dateTime	構成が削除されるまでの時間。

## Machine

このデータ型は、Lab Manager の構成ライブラリまたはワークスペースの仮想マシンごとに存在します。整数の ID フィールドは、マシンを一意に識別します。マシン名は、構成内を除いて必ずしも一意ではありません。

表 3-5. Machine フィールド

フィールド	データ型	説明
configID	int	仮想マシンが属している構成の ID。
DatastoreNameResidesOn	string	仮想マシンが格納されているデータストアの名前。
description	string	マシンの説明。
externalIP	string	フェンス内にあるときの一時的な IP アドレス。

表 3-5. Machine フィールド (続き)

フィールド	データ型	説明
HostNameDeployedOn	string	仮想マシンがデプロイされている ESX ホストの名前。仮想マシンがデプロイされていない場合、このフィールドは空です。
id	int	マシン ID。
internalIP	string	固定で割り当てられている IP アドレス。
isDeployed	boolean	デプロイされている場合は true。
macAddress	string	プライマリ NIC に割り当てられている MAC アドレス。
memory	int	メモリ サイズ (MB 単位)
name	string	マシン名。
OwnerFullName	string	仮想マシンの所有者のフルネーム。
status	int	1 = オフ。 2 = オン。 3 = サスペンド。 4 = 停止。 128 = 無効。

# Lab Manager API メソッド リファレンス

テーブル 4-1 は、vCenter Lab Manager SOAP API メソッドの一覧です。メソッドについての詳細および C# .NET コード サンプルを使用してこれらのメソッドを呼び出す方法については、個別のリンクをクリックしてください。

表 4-1. Lab Manager SOAP API メソッド

メソッド	説明
<a href="#">「ConfigurationCapture」</a> (P. 24)	ワークスペースの構成を取得し、指定された Lab Manager のデータストアに保存します。
<a href="#">「ConfigurationCheckout」</a> (P. 25)	構成ライブラリから構成をチェックアウトし、それをワークスペースに移動します。
<a href="#">「ConfigurationClone」</a> (P. 26)	ワークスペースでアクティブな構成のクローンを作成し、それをストレージに保存します。
<a href="#">「ConfigurationDelete」</a> (P. 27)	構成をワークスペースから削除します。
<a href="#">「ConfigurationDeploy」</a> (P. 28)	構成をワークスペースにデプロイします。
<a href="#">「ConfigurationPerformAction」</a> (P. 29)	構成に対してアクションを実行します。
<a href="#">「ConfigurationSetPublicPrivate」</a> (P. 30)	構成の状態をパブリックまたはプライベートに設定します。パブリックな構成は、他のユーザーもアクセスして使用できます。プライベートな構成は、所有者だけが使用できます。
<a href="#">「ConfigurationUndeploy」</a> (P. 31)	ワークスペースの構成をアンデプロイして、その状態を破棄します。
<a href="#">「GetConfiguration」</a> (P. 32)	構成 ID に一致する Configuration オブジェクトを返します。
<a href="#">「GetConfigurationByName」</a> (P. 33)	構成名に一致する Configuration オブジェクトを返します。構成名は、必ずしも一意ではありません。
<a href="#">「GetCurrentOrganizationName」</a> (P. 34)	現在の組織の名前を返します。
<a href="#">「GetCurrentWorkspaceName」</a> (P. 35)	現在のワークスペースの名前を返します。
<a href="#">「GetMachine」</a> (P. 35)	マシン ID に一致する Machine オブジェクトを返します。
<a href="#">「GetMachineByName」</a> (P. 37)	マシン名に一致する Machine オブジェクトを返します。
<a href="#">「GetSingleConfigurationByName」</a> (P. 38)	構成名に一致する 1 つの Configuration オブジェクトを返します。
<a href="#">「ListConfigurations」</a> (P. 39)	ワークスペース内または構成ライブラリ内の Configuration オブジェクトの配列を返します。
<a href="#">「ListMachines」</a> (P. 40)	構成の数値 ID に対応する Machine オブジェクトの配列を返します。
<a href="#">「LiveLink」</a> (P. 41)	電子メールで送信でき、クリックして構成を再作成できる、構成への URL を作成します。
<a href="#">「MachinePerformAction」</a> (P. 42)	マシンに対してアクションを実行します。
<a href="#">「SetCurrentOrganizationByName」</a> (P. 43)	以降の各ログインに使用する組織を設定します。
<a href="#">「SetCurrentWorkspaceByName」</a> (P. 44)	以降の各ログインに使用するワークスペースを設定します。

## ConfigurationCapture

このメソッドは、ワークスペースの構成を取得して保存します。

### 構文

```
int newConfigId = ConfigurationCapture(10,
    "Config10Capture");
```

### 引数

フィールド	データ型	説明
configurationID	int	構成 ID。
newLibraryName	string	取得名。

### 応答

フィールド	データ型	説明
configurationID	int	新しい取得の構成 ID。

### C# サンプルコード

```
try
{
    /**
    /** LabManagerSoap is the name of the Web reference in Visual Studio
    /**
    LabManagerSoap.LabManagerSOAPinterface binding = new
        LabManagerSoap.LabManagerSOAPinterface();

    /** Create login
    binding.AuthenticationHeaderValue = new
        LabManagerSoap.AuthenticationHeader();
    binding.AuthenticationHeaderValue.username = "jaya";
    binding.AuthenticationHeaderValue.password = "Lab Manager";
    binding.AuthenticationHeaderValue.organizationname = "MyOrg";
    binding.AuthenticationHeaderValue.workspacename = "MyWorkspace";
    ServicePointManager.CertificatePolicy = new CertificateAcceptor();

    /** Get Configuration object
    LabManagerSoap.Configuration Config =
        binding.GetSingleConfigurationByName("Config26");

    /** Get configuration identifier and deployed status from object
    int configurationId = Config.id;
    bool deployed = Config.isDeployed;

    /** Capture configuration if it's deployed
    if (deployed)
    {
        /** Save capture with date and time stamp
        string captureName=Config.name + DateTime.Now.ToString();
        string LMStorageServer = "LM Server";
        binding.ConfigurationCapture(configurationId, captureName);
    }
}
catch (Exception e)
{
    Console.WriteLine("Error:" + e.Message);
    Console.ReadLine();
}
```



## ConfigurationCheckout

このメソッドは、構成をライブラリからチェックアウトし、それを別の名前でワークスペースに移動します。

### 構文

```
int result = ConfigurationCheckout(7, "Config7May10");
```

### 引数

フィールド	データ型	説明
configurationID	int	構成ライブラリ内の構成の数値 ID。
workspaceName	string	チェックアウトする構成の名前。

### 応答

フィールド	データ型	説明
configurationID	int	ワークスペース内の構成の数値 ID。

## C# サンプルコード

```
try
{
    //
    /** LMSOap is the name of the Web reference in Visual Studio.
    //
    LabManagerSoap.LabManagerSOAPinterface binding = new
        LabManagerSoap.LabManagerSOAPinterface();

    //
    /** Create login
    //
    binding.AuthenticationHeaderValue = new
        LabManagerSoap.AuthenticationHeader();
    binding.AuthenticationHeaderValue.username = "jaya";
    binding.AuthenticationHeaderValue.password = "Lab Manager";
    binding.AuthenticationHeaderValue.organizationname = "MyOrg";
    binding.AuthenticationHeaderValue.workspacename = "MyWorkspace";
    binding.Url =
        "https://demo18.LabManager.com/LabManager/SOAP/LabManager.asmx";
    binding.Timeout = 10 * 60 * 1000; // 10 minutes
    ServicePointManager.CertificatePolicy = new CertificateAcceptor();

    //
    /** Get Configuration object
    //
    LabManagerSoap.Configuration Config =
        binding.GetSingleConfigurationByName("Win2K3Exchange");
    int configurationId = Config.id;

    //
    /** Timestamp library configuration name as new Workspace name
    //
    string checkoutName=Config.name + DateTime.Now.ToString();

    //
    /** Check out and move to Workspace
    //
    int newConfigID = binding.ConfigurationCheckout(Config.id,
        checkoutName);
    Console.WriteLine("New Config ID=" + newConfigID.ToString());
    Console.ReadLine();
}
catch (Exception e)
{
    Console.WriteLine("Error="+e.Message);
    Console.ReadLine();
}
```

## ConfigurationClone

このメソッドは、ワークスペース構成のクローンを作成し、それをデータストアに保存して、新しい名前を使用してワークスペース内で参照できるようにします。

### 構文

```
int result = ConfigurationClone(6, "Config6Clone");
```

### 引数

フィールド	データ型	説明
configurationId	int	構成ライブラリ内の構成の数値 ID。
newWorkspaceName	string	新しいワークスペース構成の名前。

### 応答

フィールド	データ型	説明
configurationID	int	新しいワークスペース構成の数値 ID。

### C# サンプルコード

```
try
{
    /**
    /** LabManagerSoap is the name of the Web reference in Visual Studio
    /**
    LabManagerSoap.LabManagerSOAPinterface binding = new
        LabManagerSoap.LabManagerSOAPinterface();

    /** Create login
    binding.AuthenticationHeaderValue = new
        LabManagerSoap.AuthenticationHeader();
    binding.AuthenticationHeaderValue.username = "jaya";
    binding.AuthenticationHeaderValue.password = "Lab Manager";
    binding.AuthenticationHeaderValue.organizationname = "MyOrg";
    binding.AuthenticationHeaderValue.workspacename = "MyWorkspace";
    ServicePointManager.CertificatePolicy = new CertificateAcceptor();

    /** Clone Configuration
    int newConfigId = binding.ConfigurationClone(24, "ClonedConfig24");
    Console.WriteLine("New Config ID=" + newConfigId.ToString());
}
catch (Exception e)
{
    Console.WriteLine("Error:" + e.Message);
    Console.ReadLine();
}
```

## ConfigurationDelete

このメソッドは、構成をワークスペースから削除します。デプロイされている構成は削除できません。

### 構文

```
ConfigurationDelete(6);
```

### 引数

フィールド	データ型	説明
configurationID	int	ワークスペース構成の数値 ID。

### 応答

応答はありません。

### C# サンプルコード

```
try
{
    LabManagerSoap.LabManagerSOAPInterface binding = new
        LabManagerSoap.LabManagerSOAPInterface();

    /** Create login
    binding.AuthenticationHeaderValue = new
        LabManagerSoap.AuthenticationHeader();
    binding.AuthenticationHeaderValue.username = "jaya";
    binding.AuthenticationHeaderValue.password = "Lab Manager";
    binding.AuthenticationHeaderValue.organizationname = "MyOrg";
    binding.AuthenticationHeaderValue.workspacename = "MyWorkspace";
    ServicePointManager.CertificatePolicy = new CertificateAcceptor();

    /** Get Configuration object
    LabManagerSoap.Configuration Config=binding.GetSingleConfigurationByName(
        "Config24");

    /** Get configuration identifier and deployed status from object
    int configurationId = Config.id;
    bool deployed = Config.isDeployed;

    /** Delete configuration if it isn't deployed
    if (!deployed)
    {
        binding.ConfigurationDelete(configurationId);
    }
    else
    {
        /**
        /** Must undeploy configuration before deleting it
        /**
        binding.ConfigurationUndeploy(configurationId);
        binding.ConfigurationDelete(configurationId);
    }
}
catch (Exception e)
{
    Console.WriteLine("Error:" + e.Message);
    Console.ReadLine();
}
```

## ConfigurationDeploy

このメソッドを使用すると、ワークスペース内に存在するアンデプロイされた構成をデプロイできます。

### 構文

```
ConfigurationDeploy(6, false, 1);
```

### 引数

フィールド	データ型	説明
configurationID	int	ワークスペース内の構成の数値 ID。
isCached	boolean	常に false 値に設定します。
fenceMode	int	1 = Nonfenced 2 = FenceBlockInAndOut 3 = FenceAllowOutOnly 4 = FenceAllowInAndOut

### 応答

応答はありません。

### C# サンプルコード

```
try
{
    LabManagerSoap.LabManagerSOAPinterface binding = new
        LabManagerSoap.LabManagerSOAPinterface();

    /** Create login
    binding.AuthenticationHeaderValue = new
        LabManagerSoap.AuthenticationHeader();
    binding.AuthenticationHeaderValue.username = "jaya";
    binding.AuthenticationHeaderValue.password = "Lab Manager";
    binding.AuthenticationHeaderValue.organizationname = "MyOrg";
    binding.AuthenticationHeaderValue.workspaceName = "MyWorkspace";
    ServicePointManager.CertificatePolicy = new CertificateAcceptor();

    /** Get Configuration object
    LabManagerSoap.Configuration Config =
        binding.GetSingleConfigurationByName("Config24");

    /** Get configuration identifier and deployed status from object
    int configurationId = Config.id;
    bool deployed = Config.isDeployed;

    /** Deploy configuration if it isn't already.
    if (!deployed)
    {
        /** Deploy in fenced mode and run from ESX hosts
        binding.ConfigurationDeploy(configurationId, false, 1);
    }
}
catch (Exception e)
{
    Console.WriteLine("Error:" + e.Message);
    Console.ReadLine();
}
```

## ConfigurationPerformAction

このメソッドは、アクション ID で指定されている次のいずれかの構成アクションを実行します。

- 1=パワーオン。構成をオンにします。
- 2=パワーオフ。構成をオフにします。何も保存されません。
- 3=サスペンド。CPU および構成の状態をフリーズします。
- 4=レジューム。サスペンドしている構成をレジュームします。
- 5=リセット。構成を再起動します。
- 6=スナップショット。特定の時点での構成の状態を保存します。
- 7=復元。構成をスナップショットの状態に復元します。
- 8=シャットダウン。オフにする前に構成をシャットダウンします。

### 構文

```
ConfigurationPerformAction(int configurationID, int action);
```

### 引数

フィールド	データ型	説明
action	int	構成で実行するアクション。
configurationID	int	構成 ID。

### 応答

応答はありません。

### C# サンプルコード

```
try
{
    LabManagerSoap.LabManagerSOAPInterface binding = new
        LabManagerSoap.LabManagerSOAPInterface();
    binding.AuthenticationHeaderValue = new
        LabManagerSoap.AuthenticationHeader();
    binding.AuthenticationHeaderValue.username = "jaya";
    binding.AuthenticationHeaderValue.password = "Lab Manager";
    binding.AuthenticationHeaderValue.organizationname = "MyOrg";
    binding.AuthenticationHeaderValue.workspacename = "MyWorkspace";
    ServicePointManager.CertificatePolicy = new CertificateAcceptor();

    int configurationType = 1; /** Get workspace configuration

    /**
    /** Get array of all configurations
    /**
    LabManagerSoap.Configuration [] configurations =
        binding.ListConfigurations(configurationType);

    /**
    /** Loop through all configurations.
    /**
    for (int j=0; j < configurations.Length; j++)
        {
            binding.ConfigurationPerformAction(configurations[j].id,4/*Resume*/);
        }
}
catch (Exception e)
{
    Console.WriteLine("Error:" + e.Message);
    Console.ReadLine();
}
```

## ConfigurationSetPublicPrivate

このメソッドは、構成の状態をパブリックまたはプライベートに設定します。構成の状態がパブリックの場合は、すべての組織のすべての Lab Manager ユーザーがこの構成にアクセスできます (読み取りのみ)。構成がプライベートの場合は、所有者と管理者のみがその構成を表示できます。

### 構文

```
ConfigurationSetPublicPrivate(10, false);
```

### 引数

フィールド	データ型	説明
configurationID	int	構成 ID。
isPublic	boolean	true = パブリック、false = プライベート。

### 応答

応答はありません。

### C# サンプルコード

```
try
{
    //
    /** LabManagerSoap is the name of the Web reference in Visual Studio
    //
    LabManagerSoap.LabManagerSOAPinterface binding = new
        LabManagerSoap.LabManagerSOAPinterface();

    /** Create login
    binding.AuthenticationHeaderValue = new
        LabManagerSoap.AuthenticationHeader();
    binding.AuthenticationHeaderValue.username = "jaya";
    binding.AuthenticationHeaderValue.password = "Lab Manager";
    binding.AuthenticationHeaderValue.organizationname = "MyOrg";
    binding.AuthenticationHeaderValue.workspacename = "MyWorkspace";
    ServicePointManager.CertificatePolicy = new CertificateAcceptor();

    /** Get Configuration object
    LabManagerSoap.Configuration Config =
        binding.GetSingleConfigurationByName("Config24");

    /** Get configuration identifier and shared status from object
    bool shared = Config.isPublic;

    /** Make configuration public if it isn't already.
    if (!shared)
    {
        binding.ConfigurationSetPublicPrivate(Config.id, true);
    }
}
catch (Exception e)
{
    Console.WriteLine("Error:" + e.Message);
    Console.ReadLine();
}
```

## ConfigurationUndeploy

ワークスペースの構成をアンデプロイして、その状態を破棄します。

### 構文

```
ConfigurationUndeploy(10);
```

### 引数

フィールド	データ型	説明
configurationID	int	構成の数値 ID。

### 応答

応答はありません。

### C# サンプルコード

```
try
{
    LabManagerSoap.LabManagerSOAPInterface binding = new
        LabManagerSoap.LabManagerSOAPInterface();
    binding.Url =
        "https://demo18.LabManager.com/LabManager/SOAP/LabManager.asmx";
    binding.Timeout = 10 * 60 * 1000; // 10 minutes
    ServicePointManager.CertificatePolicy = new CertificateAcceptor();
    binding.AuthenticationHeaderValue = new
        LabManagerSoap.AuthenticationHeader();
    binding.AuthenticationHeaderValue.username = "jaya";
    binding.AuthenticationHeaderValue.password = "Lab Manager";
    binding.AuthenticationHeaderValue.organizationname = "MyOrg";
    binding.AuthenticationHeaderValue.workspacename = "MyWorkspace";
    //
    /** Get configurations in Workspace, not Library
    //
    int configurationType= 1;
    LabManagerSoap.Configuration[] configurations =
        binding.ListConfigurations(configurationType);
    //
    /** Undeploy all deployed configurations I own
    //
    for (int i=0; i < configurations.Length; i++)
    {
        if (configurations[i].owner.Equals("jaya"))
        {
            binding.ConfigurationUndeploy(configurations[i].id);
        }
    }
}
catch (Exception e)
{
    Console.WriteLine("Error:" + e.Message);
    Console.ReadLine();
}
```

## GetConfiguration

このメソッドは、渡された構成 ID に一致する Configuration 型のオブジェクトを返します。

### 構文

```
Configuration config = GetConfiguration(10);
```

フィールド	データ型	説明
configurationID	int	構成 ID。

### 応答

フィールド	データ型	説明
configuration	Configuration	渡された構成 ID に一致する Configuration オブジェクト。

## C# サンプルコード

```
try
{
    /** LabManagerSoap is the name of the Web reference in Visual Studio
    LabManagerSoap.LabManagerSOAPinterface binding = new
        LabManagerSoap.LabManagerSOAPinterface();

    /** Create login
    binding.AuthenticationHeaderValue = new
        LabManagerSoap.AuthenticationHeader();
    binding.AuthenticationHeaderValue.username = "jaya";
    binding.AuthenticationHeaderValue.password = "Lab Manager";
    binding.AuthenticationHeaderValue.organizationname = "MyOrg";
    binding.AuthenticationHeaderValue.workspacename = "MyWorkspace";
    ServicePointManager.CertificatePolicy = new CertificateAcceptor();

    /** Get Configuration object
    LabManagerSoap.Configuration Config =
        binding.GetConfiguration(26);

    /** Write to the console configuration properties
    Console.WriteLine("Config name = " + Config.name);
    Console.WriteLine("Config id = " + Config.id.ToString());
    Console.WriteLine("Config description = " + Config.description);
    Console.WriteLine("Config isPublic = " + Config.isPublic.ToString());
    Console.WriteLine("Config isDeployed = " + Config.isDeployed.ToString());
    Console.WriteLine("Config fenceMode = " + Config.fenceMode.ToString());
    Console.WriteLine("Config type = " + Config.type.ToString());
    Console.WriteLine("Config owner = " + Config.owner);
    Console.WriteLine("Config dateCreated = " +
        Config.dateCreated.ToString());
    Console.ReadLine();
}
catch (Exception e)
{
    Console.WriteLine("Error:" + e.Message);
    Console.ReadLine();
}
```



## GetConfigurationByName

このメソッドは、構成の名前を受け取り、その名前に一致する構成の配列を返します。ワークスペースの構成名は一意です。指定された名前でも複数の構成が存在する場合があります。指定された名前の構成が存在しない場合、空の配列を返します。

### 構文

```
Configuration [] config = GetConfigurationByName( "Config9" );
```

### 引数

フィールド	データ型	説明
name	string	構成名。

### 応答

フィールド	データ型	説明
configuration[]	Configuration	同じ名前の Configuration オブジェクトの配列。

### C# サンプルコード

```
try
{
    //
    /** LabManagerSoap is the name of the Web reference in Visual Studio
    //
    LabManagerSoap.LabManagerSOAPInterface binding = new
        LabManagerSoap.LabManagerSOAPInterface();
    /** Create login
    binding.AuthenticationHeaderValue = new
        LabManagerSoap.AuthenticationHeader();
    binding.AuthenticationHeaderValue.username = "jaya";
    binding.AuthenticationHeaderValue.password = "Lab Manager";
    binding.AuthenticationHeaderValue.organizationname = "MyOrg";
    binding.AuthenticationHeaderValue.workspacename = "MyWorkspace";
    ServicePointManager.CertificatePolicy = new CertificateAcceptor();

    //
    /** Get Configuration objects
    //
    LabManagerSoap.Configuration [] Configs =
        binding.GetConfigurationByName("Config24Capture");

    //
    /** Write to the console all configurations and their properties.
    //
    for (int i=0; i < Configs.Length; i++)
    {
        Console.WriteLine("Config name = " + Configs[i].name);
        Console.WriteLine("id = " + Configs[i].id.ToString());
        Console.WriteLine("description = " + Configs[i].description);
        Console.WriteLine("isPublic = " +
            Configs[i].isPublic.ToString());
        Console.WriteLine("isDeployed = " +
            Configs[i].isDeployed.ToString());
        Console.WriteLine("fenceMode = " +
            Configs[i].fenceMode.ToString());
        Console.WriteLine("type = " + Configs[i].type.ToString());
        Console.WriteLine("owner = " + Configs[i].owner);
        Console.WriteLine("dateCreated = " +
            Configs[i].dateCreated.ToString());
    }
}
```

```

        Console.WriteLine();
        Console.ReadLine();
    }
}
catch (Exception e)
{
    Console.WriteLine("Error:" + e.Message);
    Console.ReadLine();
}
}

```

## GetCurrentOrganizationName

このメソッドは、現在の組織の名前を返します。

### 構文

```
string organizationName = GetCurrentOrganizationName();
```

### 応答

フィールド	データ型	説明
organization name	string	現在の組織の名前を返します。

### C# サンプルコード

```

try
{
    LabManagerSoap.LabManagerSOAPinterface binding = new
        LabManagerSoap.LabManagerSOAPinterface();
    binding.AuthenticationHeaderValue = new
        LabManagerSoap.AuthenticationHeader();
    binding.AuthenticationHeaderValue.username = "jaya";
    binding.AuthenticationHeaderValue.password = "Lab Manager";
    ServicePointManager.CertificatePolicy = new CertificateAcceptor();

    string organizationName = binding.GetCurrentOrganizationName();

    Console.WriteLine("Current organization I am logged in:" + organizationName);

    Console.ReadLine();
}
catch (Exception e)
{
    Console.WriteLine("Error:" + e.Message);
    Console.ReadLine();
}
}

```

## GetCurrentWorkspaceName

このメソッドは、現在のワークスペースの名前を返します。

### 構文

```
string workspaceName = GetCurrentWorkspaceName();
```

### 応答

フィールド	データ型	説明
workspace name	string	現在のワークスペースの名前を返します。

### C# サンプルコード

```
try
{
    LabManagerSoap.LabManagerSOAPinterface binding = new
        LabManagerSoap.LabManagerSOAPinterface();
    binding.AuthenticationHeaderValue = new
        LabManagerSoap.AuthenticationHeader();
    binding.AuthenticationHeaderValue.username = "jaya";
    binding.AuthenticationHeaderValue.password = "Lab Manager";
    binding.AuthenticationHeaderValue.organizationname;
    ServicePointManager.CertificatePolicy = new CertificateAcceptor();

    string workspaceName = binding.GetCurrentWorkspaceName();

    Console.WriteLine("Current workspace I am logged in:" + workspaceName);

    Console.ReadLine();
}
catch (Exception e)
{
    Console.WriteLine("Error:" + e.Message);
    Console.ReadLine();
}
```

## GetMachine

このメソッドは、マシンの数値 ID を受け取り、対応する Machine オブジェクトを返します。

### 構文

```
Machine mach = GetMachine(10);
```

### 引数

フィールド	データ型	説明
machineID	int	マシン ID。

### 応答

フィールド	データ型	説明
machine	Machine	マシン ID に一致する Machine オブジェクト。

## C# サンプルコード

```
try
{
    LabManagerSoap.LabManagerSOAPInterface binding = new
        LabManagerSoap.LabManagerSOAPInterface();
    binding.AuthenticationHeaderValue = new
        LabManagerSoap.AuthenticationHeader();
    binding.AuthenticationHeaderValue.username = "jaya";
    binding.AuthenticationHeaderValue.password = "Lab Manager";
    binding.AuthenticationHeaderValue.organizationname = "MyOrg";
    binding.AuthenticationHeaderValue.workspacename = "MyWorkspace";
    ServicePointManager.CertificatePolicy = new CertificateAcceptor();

    LabManagerSoap.Machine machine = binding.GetMachine(35);

    /** Write to the console all machines in configuration.
    Console.WriteLine("Machine = " + machine.name);
    Console.WriteLine("id = " + machine.id.ToString());
    Console.WriteLine("description = " + machine.description);
    Console.WriteLine("internalIP = " + machine.internalIP);
    Console.WriteLine("externalIP = " + machine.externalIP);
    Console.WriteLine("status = " + machine.status.ToString());
    Console.WriteLine("isDeployed = " + machine.isDeployed.ToString());
    Console.ReadLine();
}
catch (Exception e)
{
    Console.WriteLine("Error:" + e.Message);
    Console.ReadLine();
}
```

## GetMachineByName

このメソッドは、構成 ID とマシン名を受け取り、一致する Machine オブジェクトを返します。

### 構文

```
Machine mach = GetMachineByName(10, "Config9VM1");
```

### 引数

フィールド	データ型	説明
configurationId	int	構成 ID。
name	string	マシン名。

### 応答

フィールド	データ型	説明
machine	Machine	Machine オブジェクト。

### C# サンプルコード

```
try
{
    LabManagerSoap.LabManagerSOAPInterface binding = new
        LabManagerSoap.LabManagerSOAPInterface();
    binding.AuthenticationHeaderValue = new
        LabManagerSoap.AuthenticationHeader();
    binding.AuthenticationHeaderValue.username = "jaya";
    binding.AuthenticationHeaderValue.password = "Lab Manager";
    binding.AuthenticationHeaderValue.organizationname = "MyOrg";
    binding.AuthenticationHeaderValue.workspacename = "MyWorkspace";
    ServicePointManager.CertificatePolicy = new CertificateAcceptor();

    LabManagerSoap.Machine machine = binding.GetMachineByName(10,
        "Config9VM1");

    /** Write to the console all machines fields
    Console.WriteLine("Machine = " + machine.name);
    Console.WriteLine("id = " + machine.id.ToString());
    Console.WriteLine("description = " + machine.description);
    Console.WriteLine("internalIP = " + machine.internalIP);
    Console.WriteLine("externalIP = " + machine.externalIP);
    Console.WriteLine("status = " + machine.status.ToString());
    Console.WriteLine("isDeployed = " + machine.isDeployed.ToString());
    Console.ReadLine();
}
catch (Exception e)
{
    Console.WriteLine("Error:" + e.Message);
    Console.ReadLine();
}
```

## GetSingleConfigurationByName

このメソッドは、構成名を受け取り、構成ライブラリとワークスペースの両方でそれを検索して、対応する Configuration オブジェクトを返します。

### 構文

```
Configuration config = GetSingleConfigurationByName( "Config9" );
```

### 引数

フィールド	データ型	説明
name	string	構成名。

### 応答

フィールド	データ型	説明
configuration	Configuration	構成オブジェクト。

### C# サンプルコード

```
try
{
    /** LabManagerSoap is the name of the Web reference in Visual Studio
    LabManagerSoap.LabManagerSOAPinterface binding = new
        LabManagerSoap.LabManagerSOAPinterface();

    /** Create login
    binding.AuthenticationHeaderValue = new
        LabManagerSoap.AuthenticationHeader();
    binding.AuthenticationHeaderValue.username = "jaya";
    binding.AuthenticationHeaderValue.password = "Lab Manager";
    binding.AuthenticationHeaderValue.organizationname = "MyOrg";
    binding.AuthenticationHeaderValue.workspacename = "MyWorkspace";
    ServicePointManager.CertificatePolicy = new CertificateAcceptor();

    /** Get Configuration object
    LabManagerSoap.Configuration Config =
        binding.GetSingleConfigurationByName("Config24Capture");

    /** Write to the console configuration properties.
    Console.WriteLine("Config name = " + Config.name);
    Console.WriteLine("Config id = " + Config.id.ToString());
    Console.WriteLine("Config description = " + Config.description);
    Console.WriteLine("Config isPublic = " + Config.isPublic.ToString());
    Console.WriteLine("Config isDeployed = " + Config.isDeployed.ToString());
    Console.WriteLine("Config fenceMode = " + Config.fenceMode.ToString());
    Console.WriteLine("Config type = " + Config.type.ToString());
    Console.WriteLine("Config owner = " + Config.owner);
    Console.WriteLine("Config dateCreated = " +
        Config.dateCreated.ToString());
    Console.ReadLine();
}
catch (Exception e)
{
    Console.WriteLine("Error:" + e.Message);
    Console.ReadLine();
}
```

## ListConfigurations

このメソッドは、`Configuration` 型の配列を返します。要求された構成タイプに応じて、構成ライブラリ内の各構成またはワークスペース内の各構成に対して 1 つのオブジェクトを返します。

### 構文

```
Configuration [] config = ListConfigurations(1);
```

### 引数

フィールド	データ型	説明
configurationType	int	1 = ワークスペース構成、2 = ライブラリ構成。

### 応答

フィールド	データ型	説明
configurations[]	Configuration Array	Configuration オブジェクトの配列。

### C# サンプルコード

```
try
{
    /**
    /** LabManagerSoap is the name of the Web reference in Visual Studio.
    /**
    LabManagerSoap.LabManagerSOAPinterface binding = new
        LabManagerSoap.LabManagerSOAPinterface();

    /** Create login
    binding.AuthenticationHeaderValue = new
        LabManagerSoap.AuthenticationHeader();
    binding.AuthenticationHeaderValue.username = "jaya";
    binding.AuthenticationHeaderValue.password = "Lab Manager";
    binding.AuthenticationHeaderValue.organizationname = "MyOrg";
    binding.AuthenticationHeaderValue.workspacename = "MyWorkspace";
    ServicePointManager.CertificatePolicy = new CertificateAcceptor();

    /** Get Configurations in Workspace.
    int configurationType = 1; /** 1=Workspace
    LabManagerSoap.Configuration [] WSconfigurations =
        binding.ListConfigurations(configurationType);

    /** Write to the console all configurations
    for (int i=0; i < WSconfigurations.Length; i++)
    {
        Console.WriteLine("Configuration name = " +
            WSconfigurations[i].name);
        Console.WriteLine("id = " + WSconfigurations[i].id.ToString());
        Console.WriteLine("description = " +WSconfigurations[i].description);
        Console.WriteLine("isPublic = " +
            WSconfigurations[i].isPublic.ToString());
        Console.WriteLine("isDeployed = "+
            WSconfigurations[i].isDeployed.ToString());
        Console.WriteLine("fenceMode = " +
            WSconfigurations[i].fenceMode.ToString());
        Console.WriteLine("type = " + WSconfigurations[i].type.ToString());
        Console.WriteLine("owner = " + WSconfigurations[i].owner);
        Console.WriteLine("dateCreated = " +
            WSconfigurations[i].dateCreated.ToString());
        Console.WriteLine();
    }
    Console.ReadLine();
}
catch (Exception e)
{
    Console.WriteLine("Error:" + e.Message);
    Console.ReadLine();
}
```

## ListMachines

このメソッドは、Machine 型の配列を返します。構成内の仮想マシンごとに 1 つの Machine オブジェクトを返します。

### 構文

```
Machine [] machines = ListMachines(1);
```

### 引数

フィールド	データ型	説明
configurationID	int	構成の数値 ID。

### 応答

フィールド	データ型	説明
machine[]	Machine array	Machine オブジェクトの配列。

## C# サンプルコード

```
try
{
    LabManagerSoap.LabManagerSOAPinterface binding = new
        LabManagerSoap.LabManagerSOAPinterface();
    binding.Url =
        "https://demo18.LabManager.com/LabManager/SOAP/LabManager.asmx";
    binding.Timeout = 10 * 60 * 1000; // 10 minutes
    ServicePointManager.CertificatePolicy = new CertificateAcceptor();

    binding.AuthenticationHeaderValue = new
        LabManagerSoap.AuthenticationHeader();
    binding.AuthenticationHeaderValue.username = "jaya";
    binding.AuthenticationHeaderValue.password = "Lab Manager";
    binding.AuthenticationHeaderValue.organizationname = "MyOrg";
    binding.AuthenticationHeaderValue.workspacename = "MyWorkspace";
    int configurationType = 1;

    /** Get workspace configuration
    LabManagerSoap.Configuration [] configurations =
        binding.ListConfigurations(configurationType);
    for (int j=0; j < configurations.Length; j++)
    {
        Console.WriteLine("Configuration = " +
            configurations[j].name.ToString());
        LabManagerSoap.Machine [] machines =
            binding.ListMachines(configurations[j].id);
        /** Write to the console all machines in configuration
        for (int i=0; i < machines.Length; i++)
        {
            Console.WriteLine("Machine = " + machines[i].name);
            Console.WriteLine("id = " + machines[i].id.ToString());
            Console.WriteLine("description = " + machines[i].description);
            Console.WriteLine("internalIP = " + machines[i].internalIP);
            Console.WriteLine("externalIP = " + machines[i].externalIP);
            Console.WriteLine("status = " + machines[i].status.ToString());
            Console.WriteLine("isDeployed = " +
                machines[i].isDeployed.ToString());
            Console.WriteLine();
        }
        Console.ReadLine();
    }
}
catch (Exception e)
{
    Console.WriteLine("Error:" + e.Message);
    Console.ReadLine();
}
```



## LiveLink

このメソッドを使用すると、ライブラリの構成に対する LiveLink URL を作成できます。

### 構文

```
string url = LiveLink( "LiveLinkWin2K" );
```

### 引数

フィールド	データ型	説明
configurationName	string	ライブラリ構成の名前。

### 応答

フィールド	データ型	説明
URL	string	ライブラリ内の構成の URL を含む文字列。URL を電子メールで送信できます。その URL をクリックすると構成が再作成されます。

## C# サンプルコード

```
try
{
    LabManagerSoap.LabManagerSOAPinterface binding = new
    LabManagerSoap.LabManagerSOAPinterface();

    /** Create login
    binding.AuthenticationHeaderValue = new
        LabManagerSoap.AuthenticationHeader();
    binding.AuthenticationHeaderValue.username = "jaya";
    binding.AuthenticationHeaderValue.password = "Lab Manager";
    binding.AuthenticationHeaderValue.organizationname = "MyOrg";
    binding.AuthenticationHeaderValue.workspacename = "MyWorkspace";
    binding.Url =
        "https://demo18.LabManager.com/LabManager/SOAP/LabManager.asmx";
    binding.Timeout = 10 * 60 * 1000; // 10 minutes
    ServicePointManager.CertificatePolicy = new CertificateAcceptor();

    /** Get Configuration object
    LabManagerSoap.Configuration Config =
        binding.GetSingleConfigurationByName("Win2kBEA");

    /** If configuration is deployed, livelink it
    if (Config.isDeployed)
    {
        string captureName= "Win2kBEA" + DateTime.Now.ToString();
        string url = binding.LiveLink(Config.name);
        Console.WriteLine("LiveLink URL="+url);
        Console.ReadLine();
    }
}

catch (Exception e)
{
    Console.WriteLine("Error="+e.Message);
    Console.ReadLine();
}
```

## MachinePerformAction

このメソッドは、アクション ID で指定されている次のいずれかのマシン アクションを実行します。

- 1=パワーオン。マシンをオンにします。
- 2=パワーオフ。マシンをオフにします。何も保存されません。
- 3=サスペンド。マシンの CPU と状態をフリーズします。
- 4=レジューム。サスペンドしているマシンをレジュームします。
- 5=リセット。マシンを再起動します。
- 6=スナップショット。特定の時点でのマシンの状態を保存します。
- 7=復元。マシンをスナップショットの状態に復元します。
- 8=シャットダウン。オフにする前にマシンをシャットダウンします。

### 構文

```
MachinePerformAction(1, 3);
```

### 引数

フィールド	データ型	説明
action	int	マシンで実行するアクション。
machineID	int	マシン ID。

### 応答

応答はありません。

### C# サンプルコード

```
try
{
    LabManagerSoap.LabManagerSOAPInterface binding = new
        LabManagerSoap.LabManagerSOAPInterface();
    binding.AuthenticationHeaderValue = new
        LabManagerSoap.AuthenticationHeader();
    binding.AuthenticationHeaderValue.username = "jaya";
    binding.AuthenticationHeaderValue.password = "Lab Manager";
    binding.AuthenticationHeaderValue.organizationname = "MyOrg";
    binding.AuthenticationHeaderValue.workspacename = "MyWorkspace";
    ServicePointManager.CertificatePolicy = new CertificateAcceptor();

    int configurationType = 1; /** Get workspace configuration

    /**
    /** Get array of all configurations
    /**
    LabManagerSoap.Configuration [] configurations =
        binding.ListConfigurations(configurationType);

    /**
    /** Loop through all configurations.
    /**
    for (int j=0; j < configurations.Length; j++)
    {
        /**
        /** Get array of all machines in configurations
        /**
        LabManagerSoap.Machine [] machines =
```

```

        binding.ListMachines(configurations[j].id);
        /**
        /** Loop through all machines
        /**
        for (int i=0; i < machines.Length; i++)
        {
            /**
            /** Check status of machine is suspended, then resume it
            /**
            if (machines[i].status == 3)
            {
                binding.MachinePerformAction(machines[i].id, 4);
            }
        }
    }
}
catch (Exception e)
{
    Console.WriteLine("Error:" + e.Message);
    Console.ReadLine();
}
}

```

## SetCurrentOrganizationByName

以降の各ログインに使用する組織を設定します。このメソッドは、許可ヘッダーで組織名が空のときに機能します。

### 構文

```
SetCurrentOrganizationByName( "MyOrganization" );
```

### 引数

フィールド	データ型	説明
orgName	string	ユーザーがメンバである組織。

### C# サンプルコード

```

try
{
    LabManagerSoap.LabManagerSOAPinterface binding = new
        LabManagerSoap.LabManagerSOAPinterface();
    binding.AuthenticationHeaderValue = new
        LabManagerSoap.AuthenticationHeader();
    binding.AuthenticationHeaderValue.username = "jaya";
    binding.AuthenticationHeaderValue.password = "Lab Manager";
    // This field must be empty for SetCurrentOrganization to work
    binding.AuthenticationHeaderValue.organizationname = "";
    ServicePointManager.CertificatePolicy = new CertificateAcceptor();

    SetCurrentOrganizationByName("MyOrganization");
    string organizationName = GetCurrentOrganizationName();

    Console.WriteLine("Current organization I am logged in:" + organizationName);

    Console.ReadLine();
}
catch (Exception e)
{
    Console.WriteLine("Error:" + e.Message);
    Console.ReadLine();
}
}

```

## SetCurrentWorkspaceByName

以降の各ログインに使用するワークスペースを設定します。このメソッドは、許可ヘッダーでワークスペース名が空のときに機能します。

### 構文

```
SetCurrentWorkspaceByName( "MyOrganization ", "MyWorkspace" );
```

### 引数

フィールド	データ型	説明
orgName	string	ユーザーがメンバである組織。
workspaceName	string	ユーザーがメンバであるワークスペース。

### C# サンプルコード

```
try
{
    LabManagerSoap.LabManagerSOAPinterface binding = new
        LabManagerSoap.LabManagerSOAPinterface();
    binding.AuthenticationHeaderValue = new
        LabManagerSoap.AuthenticationHeader();
    binding.AuthenticationHeaderValue.username = "jaya";
    binding.AuthenticationHeaderValue.password = "Lab Manager";
    // The following authorization header fields must be empty or absent
    // for the SetCurrentOrganization and SetCurrentWorkspace to work
    binding.AuthenticationHeaderValue.organizationname = "";
    binding.AuthenticationHeaderValue.workspacename = "";
    ServicePointManager.CertificatePolicy = new CertificateAcceptor();

    SetCurrentOrganizationByName("MyOrganization");
    SetCurrentWorkspaceByName("MyOrganization", "MyWorkspace");

    string organizationName = GetCurrentOrganizationName();
    string workspaceName = GetCurrentWorkspaceName();

    Console.WriteLine("Current organization I am logged in:" + organizationName);
    Console.WriteLine("Current workspace I am logged in:" + workspaceName);

    Console.ReadLine();
}
catch (Exception e)
{
    Console.WriteLine("Error:" + e.Message);
    Console.ReadLine();
}
```

# 索引

## C

CaptureUndeployConfiguration, **18**  
CertificateAcceptor(), **18**  
CheckoutDeployConfiguration, **17**

## G

GetLMAPI(), **18**

## L

Lab Manager SOAP API、定義, **9**

## R

RunQCTestset(), **16**

## S

SOAP API メソッド

ConfigurationCapture, **24**  
ConfigurationCheckout, **25**  
ConfigurationClone, **26**  
ConfigurationDelete, **27**  
ConfigurationDeploy, **28**  
ConfigurationPerformAction, **29**  
ConfigurationSetPublicPrivate, **30**  
ConfigurationUndeploy, **31**  
GetConfiguration, **32**  
GetConfigurationByName, **33**  
GetCurrentOrganizationName, **34**  
GetCurrentWorkspaceName, **35**  
GetMachine, **23, 35**  
GetMachineByName, **37**  
GetSingleConfigurationByName, **38**  
ListConfigurations, **39**  
ListMachines, **40**  
LiveLink, **41**  
MachinePerformAction, **42**  
SetCurrentOrganizationByName, **43**  
SetCurrentWorkspaceByName, **44**

SOAP API、WSDL ファイルの取得, **11**

## か

開発環境、サポート対象, **9**

## こ

言語、サポート対象, **9**

## せ

セキュリティ、SSL の使用, **10**

## そ

操作、サポート対象, **10**

## て

データ型

Lab Manager 用, **19**

Machine, **21**

データがた AuthenticationHeader, **20**

プリミティブ XML, **19**

構成, **21**

## ゆ

ユーザー、認証, **10**

