# MultiSoft Corporation



**Legtricity Integration Guideline**

**Web Service Signature Overview**

**Version 1.0**

# Table of Contents

# Web Service Signature Overview

## A.  Document Objectives

This document provides an overview of the web service signature for Legtricity.  It is the blueprint that defines the steps for integrating Legtricity product with any MLM Software application.

URL of the web-service in the test environment:

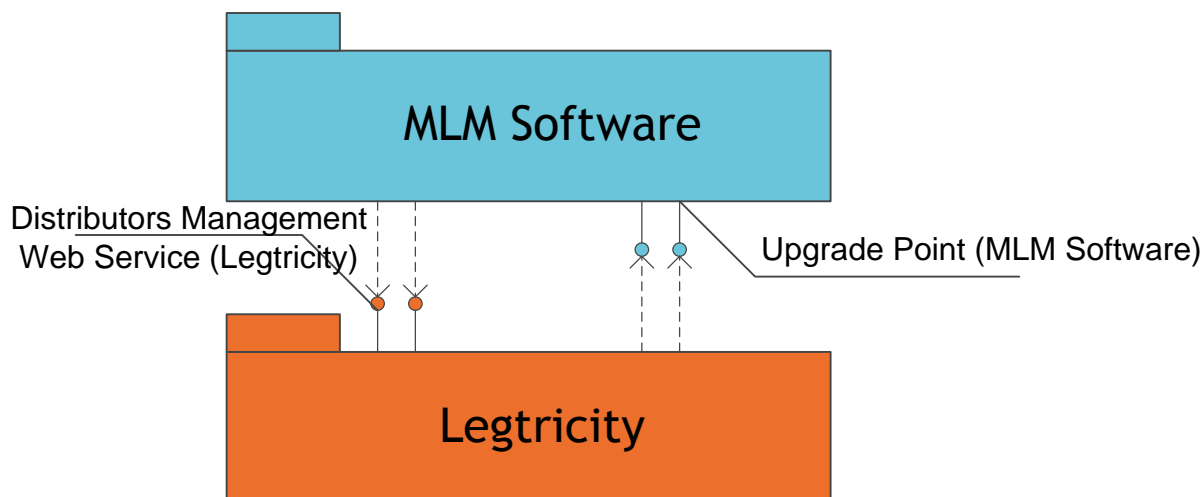Variable; please obtain from the MultiSoft technical resource assigned to assist you.

## B.  Main integration principles

Integration between the MLM Software and Legtricity is limited to two main functions:

- Populating distributors into Legtricity from the MLM Software;

- Upgrading pre-enrollees to Distributor status from Legtricity into the MLM Software.

*Chart 1. Main integration points of Legtricity.*

## C. Distributors Management Web Service

## 1.     Authentication and initialization

Authentication and initialization will be performed by the following fields:

1. Guid applicationId – Legtricity application identifier

2. mlmID – Distributor ID in MLM Software

Each of the methods returns a result as follows:

<Success>boolean</Success> – result of the executed method

<Message>string</Message> – additional message about the result or error of the method execution

## 2.     Adding distributor and default leg

To add a distributor record and its default leg the following parameters should be used:

AddDistributor (string accessCode, Guid applicationId, string mlmId, string hashPassword, string eMail, string firstName, string lastName, string phone, string city, string replicatedURL, string legFriendlyName)

1. mlmId – unique ID within the MLM Software

2. hashPassword – hash password (see "9. Hash Password")

3. eMail – unique e-mail of the distributor

4. firstName, lastName, phone, city – contact information of the distributor

5. replicatedURL – unique value, a short URL of a distributor's personal site (i.e. company.legtricity.com/replicatedURL)

6. legFriendlyName – can be blank, a friendly name of the leg; e.g. if a Binary, you may want friendly leg names of "Left" and "Right"

## 3.     Update fields

The web service should allow the following parameters to be updated – First Name, Last Name, Phone, City, Replicated URL:

UpdateDistributor web method (Guid applicationId, string mlmId, string firstName, string lastName, string phone, string city, string replicatedURL)

## 4.      Distributor Password Update

Distributor password update: public ResInfo UpdateDistributorPassword (Guid applicationId, string mlmId, string hashPassword)

## 5.      Distributor Status Update

Deactivation/Activation of a distributor in the system

UpdateDistributorStatus (Guid applicationId, string mlmId, bool isActive)

## 6.      Distributor Leg Add (Anchor Legs)

public ResInfo AddDistributorLeg (Guid applicationId, string distributorMlmId, string placementDistributorMlmId, string legFriendlyName)

*legFriendlyName – can be blank, friendly name of the leg*

*distributorMlmId – ID of the distributor who owns the leg*

*placementDistributorMlmId –ID of the distributor who is serving as an anchor leg for the owner*

## 7.      Remove leg

public ResInfo RemoveDistributorLeg(Guid applicationId, string distributorMlmId, string placementDistributorMlmId)

*distributorMlmId – ID of the distributor who owns the leg*

*placementDistributorMlmId –ID of the distributor who is serving as an anchor leg for the owner*

## 8.      Upgrade Pre-enrollee

UpgradePreEnrollee(string accessCode, Guid applicationId, Guid userId)

Upon execution of the Pre-enrollee upgrade method, the user is compressed from the leg

and can no longer access the system unless added again as a distributor via the AddDistributor method

## 9.     Hash Password

MD5 Hash function for the user password. Coding algorythm:

Base64String(MD5Hash(UTF8_password))

Implementation sample on .NET:

```
string GetHash(string password)

{

        byte[] bPassword = System.Text.UTF8Encoding.Unicode.GetBytes(password);

        System.Security.Cryptography.MD5 md5 = new
                System.Security.Cryptography.MD5CryptoServiceProvider();

        byte[] bHash = md5.ComputeHash(bPassword);

    return Convert.ToBase64String(bHash);

}
```

## D.  Upgrade Point

The upgrade point is a company-specific setting in Legtricity which directs the pre-enrollee to the MLM Software's distributor enrollment form when the pre-enrollee chooses to upgrade.

After completing the upgrade, the MLM software must return data to Legtricity per the Upgrade Pre-enrollee method (see 8 above).