

Translate Manager SDK Documentation

v1.1.0

Larson Davis – PCB Piezotronics, Inc.

Confidential

Contents

Translate Manager SDK Documentation 1

 Change Log..... 3

Introduction 4

First Project 4

 Import References 4

 Prepare and Send File 5

SLM Record 6

 Measurement History 6

 Event History..... 7

 Time History 7

Change Log

Initial draft	4/28/2016
Revisions to clarify using SImRecord	8/18/16

Introduction

This document is meant to augment the sample code as part of the SDK for opening and using ldbin files. The Translate Manager SDK has the following parts:

1. This document which describes the functionality of the SDK and associated .dll's.
2. Sample code found in the CSharpSamples solution. Each of these demo projects show how to open an .ldbin file, send that file to the translator, and receive the response which holds one or more SlmRecord objects. Each SlmRecord object contains all of the data for a single Data Record from the translated .ldbin file.
 - a. TimeHistoryDemo
 - i. Demonstrates how to process any Time History records contained in the SlmRecord.
 - b. SoundRecordingDemo
 - i. Demonstrates how to process the Session Log entries, including how to access and play any sound recordings.

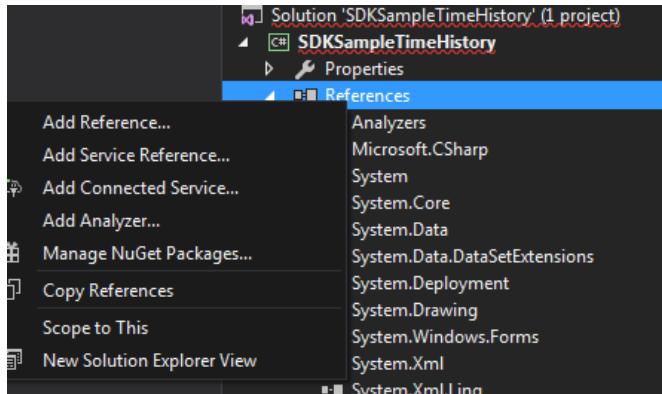
First Project

When building a project which will translate .ldbin files, you will need to do the following:

1. Import the 3 .dll references
 - a. LDTranslateManager
 - b. LDCommon
 - c. LDTranslator
2. Send the full path and filename of an .ldbin file to the translator.
 - a. Use the SlmRecord object(s) contained in the response to access the data for the associated Data Record downloaded in the .ldbin file.

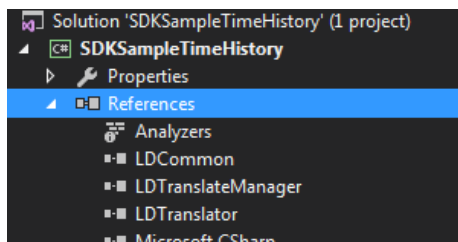
Import References

In your project you will need to right click on References then select: Add Reference...



This will open a dialog for you to find the references to LDCommon, LDTranslateManager and LDTranslator.

Choose to browse and navigate to find your .dlls from the SDK bin folder. After adding the references you should have them listed as below.



Prepare and Send File

Once you have added the .dlls, you are ready to access the functions that allow you to translate an .ldbin file. Make sure that you have a proper .ldbin file to translate. You have two options for sending an .ldbin file to be translated.

In this example we use TranslateManager to handle the call to the translator and return a simple IList<SlmRecord> list.

```
private void TranslateFile(string fileName)
{
    string tabName = Path.GetFileNameWithoutExtension(fileName);
    TranslateManager tm = new TranslateManager(tabName);
    IList<SlmRecord> slmRecs = tm.TranslateSlmBinaryData(fileName);

    foreach (var rec in slmRecs)
    {
        ProcessRecord(rec);
    }
}
```

In this example we call the translator directly and pull the SlmRecord object(s) out of the Tuple<> response.

```
private void TranslateFile(string fileName)
{
    SlmTranslator trans = new SlmTranslator();
    Tuple<ServiceReply, DataFileInfo> result = trans.TranslateFileComplete(fileName);
    IList<SlmRecord> slmRecs = new List<SlmRecord>();

    if (result.Item1.Response == ServiceResponse.Success)
    {
        foreach (var rec in result.Item2.RecordList)
        {
            slmRecs.Add(rec.SlmRec);
        }

        foreach (var rec in slmRecs)
        {
            ProcessRecord(rec);
        }
    }
}
```

SLM Record

As mentioned previously, the SlmRecord class holds all of the data from the associated Data Record in the .ldbin file. Reference the LDClassLibrary.chm in the Documentation folder for a complete description of the SlmRecord class. It is in the LarsonDavis.LDUtility.LDCommon.Legacy namespace.

Here are a few highlights:

Measurement History

Measurement History records are found in SlmRecord.Intervals.

```
private IList<IntervalInfo> GetMeasurementHistory(SlmRecord rec)
{
    IList<IntervalInfo> result = new List<IntervalInfo>();

    if (rec.Intervals.Count > 0)
    {
        result = rec.Intervals;

        foreach (var item in rec.Intervals)
        {
            ProcessMeasurementHistory(item);
        }
    }

    return result;
}

private void ProcessMeasurementHistory(IntervalInfo item)
{
    throw new NotImplementedException();
}
```

Event History

Event History records are found in `SlmRecord.Events`;

```
private IList<EventInfo> GetEventHistory(SlmRecord rec)
{
    IList<EventInfo> result = new List<EventInfo>();

    if (rec.Events.Count > 0)
    {
        result = rec.Events;

        foreach (var item in rec.Events)
        {
            ProcessEventHistory(item);
        }
    }

    return result;
}

private void ProcessEventHistory(EventInfo item)
{
    throw new NotImplementedException();
}
```

Time History

Time History records are accessed through `SlmRecord.THMngr`. (Example code is included in the `TimeHistoryDemo` project of the `CSharpSamples` solution.)