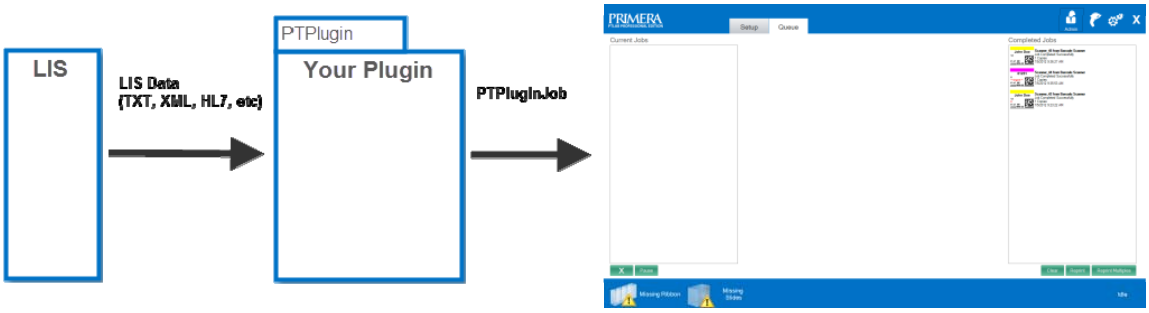
Extending PTLab PE using PTLabPlugin



Version 1.0

July 5, 2012

PRIMERA TECHNOLOGY, INC.

INTRODUCTION

This document will focus on how to develop a plugin for PTLab PE to allow for LIS integration. This is just one of a number of ways to integrate the Signature Slide Printer with your LIS. To see other ways to integrate the Signature printer into your LIS download Primera's Slide Printer Integration white paper at http://www.primerahealthcare.com/LISIntegration.html

LIS Integration using PTLabPlugin

- PTLabPlugin is a defined interface that allows 3rd party developers to create plugins for Primera's PTLab PE software.
- Jobs can be submitted through a plugin to PTLab PE
- PTLab PE will provide job status and system status back to the plugin.
- Primera provides complete C# source code for its own FlatFilePlugin that can used as a reference design for your own plugin.

PTLABPLUGIN INTERFACE

Description:

PTPluginInterface is a C# plugin (DLL) for Primera's PTLab PE software. PTLab will call into any plugins that reside in the plugin directory under the program data folder (C:\ProgramData\PTI\PTLab\Plugins on Windows 7/Vista). You can see the plugins that are recognized by PTLab via the Software Info tab in the settings dialog.

PTPluginInterface Definition:

```
namespace PTPluginInterface
{
    public enum PTJobType { Image, SVG, String }; //Only String is supported
    public enum PTSystemStatus { Idle=0, RunningJobs=1, InError=2 };

    public interface IPTPluginJob
    {
        PTJobType JobType { get; set; }
        String JobData { get; set; }
        String JobName { get; set; }
        String PluginFrom { get; set; }
        String PluginData { get; set; }
    }

    public interface IPTPluginSystemStatus
    {
        int StockRemaining { get; set; }
        int InkRemaining { get; set; }
        int Error { get; set; }
        PTSystemStatus Status { get; set; }
    }

    public interface IPTPluginJobStatus
    {
        string JobName { get; set; }
        String PluginFrom { get; set; }
        int Error { get; set; }
        String JobState { get; set; }
        String StatusString { get; set; }
    }

    public interface IPTPlugin
    {
        IPTPluginJob GetNextJob(string JobID);
        void StartPlugin();
        void ReportJobStatus(IPTPluginJobStatus status);
        void ReportSystemStatus(IPTPluginSystemStatus status);
        void StopPlugin();

        string Path { get; }
        string Name { get; }
        string Description { get; }
        string Author { get; }
        string Version { get; }
    }
}
```

SUBMITTING JOBS

Description:

PTLab will continue polling each plugin installed for jobs using a call to GetNextJob. PTLab will provide the plugin with a jobID string that will be used when PTLab reports job status.

```
IPTPluginJob GetNextJob(string JobID);

public interface IPTPluginJob
{
   PTJobType JobType { get; set; }
   String JobData { get; set; }
   String JobName { get; set; }
   String PluginFrom { get; set; }
   String PluginData { get; set; }
}
```

GETTING STATUS

Description:

PTLab will provide status on the system, including running jobs through calls to ReportJobStatus and ReportSystemStatus.

```
public enum PTSystemStatus { Idle=0, RunningJobs=1, InError=2 };

void ReportJobStatus(IPTPluginJobStatus status);

void ReportSystemStatus(IPTPluginSystemStatus status);

public interface IPTPluginSystemStatus
{
    int StockRemaining { get; }
    int InkRemaining { get; }
    int Error { get; }
    PTSystemStatus Status { get; }
}

public interface IPTPluginJobStatus
{
     string JobName { get; }
     String PluginFrom { get; }
     int Error { get;}
     String JobState { get; }
     String StatusString { get; }
}
```

Possible JobState values

Waiting, Running, CompletedCancelled, CompletedError, CompletedSuccessfully