

# ULS Automation Interface Software User Guide

Revised 5/3/2013

The automation interface software provides access to the ULS Control Panel (UCP) by means of an interface that is defined by the user (i.e. a C# API). The instructions for installation, automation operation and development using the software modules provided by ULS are described in the sections 1, 2, and 3. A sample C# application interface is provided as a guide for development.

## 1. INSTALLATION

1. Unzip the ULSRemoteControl.zip file. A debug and release version of the remote control DLL can be found in the bin directory of the C# project.
2. Install UCP version 5.35.57 or later
3. Start the UCP, wait for the icon to appear and launch the application (see figure 1).

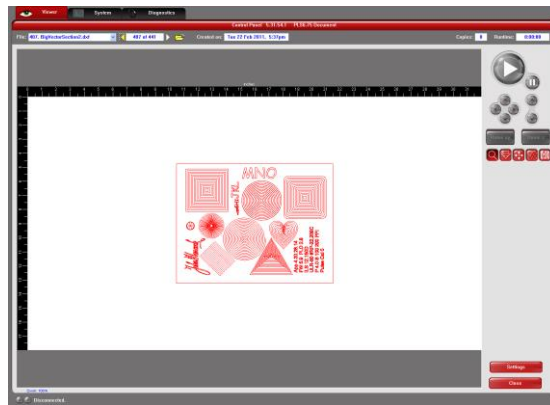


Figure 1: UCP view panel

4. Open the Remote Interface project in Visual Studio 8 (or later) and execute the code in either release or debug mode.
5. If the queue is empty; this will create problems for importing a DXF. The user must set up the material in the database before importing the DXF. A workaround is to create a job using a drawing package like Corel and printing the job using the “printer driver”.

## 2. OPERATION

### Remote Control Diagnostics GUI

The remote interface is a C# application (test GUI) that governs the UCP using a separate library, RemoteControl.dll. Its purpose is to be used as sample code for custom application software. The sample code’s events, timers, and commands are described below.

- Asynchronous signaling is available with two delegates which are invoked when the job stops, the door transitions state, the file was found in the queue or the DXF job was imported. Accompanying these events is machine status or interlock status.
- Synchronous signaling is available through the timer which updates the position of the machine, machine state, interlock status, and run time.

Note: copy the attached DLL's: remotectrl, materialdb, and fbelient to your project's release and/or debug directories. (The first two are ULS libraries. The third is the Firebird embedded database library, which is freely redistributable, and is needed to access the materials database files, UlsMaterials.fdb and UlsMatCustom.fdb, which are copied to the "ProgramData\ULDAT" directory by the UCP installer.) (Note: on Windows XP systems, this path will be C:\Documents and Setting\All Users\Application Data\ULSDAT.)

Figure 1 below shows the main screen of the Remote Control Diagnostics GUI.

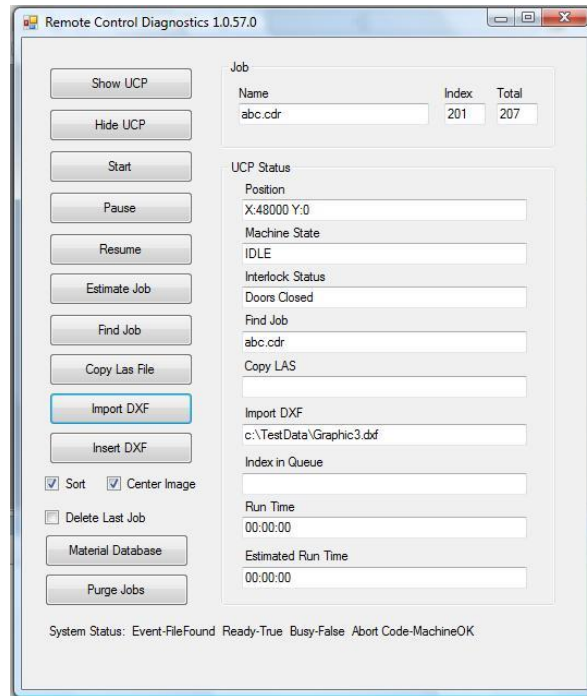


Figure 1: Remote interface when UCP is first connected to engraver.

## Materials Database Interface Test GUI

The remote interface has access to the Materials Database using a separate library, materialsdb.dll. The database-related controls can be launched by clicking the "Materials Database" near the bottom of the test GUI. Clicking the "Test Engraver Setup" button causes the application to configure the selected engraver for the specified material at the thickness entered in the edit box below. Subsequent documents printed to this particular engraver will be configured with settings appropriate to the selected material and thickness.

The Engraver selection control displays all installed printers. Those whose names begin with PLS, ILS, or VLS are listed first. All printers are included in case any of the standard names of any ULS engravers have been changed.

The Material ID must be a valid ULS-supported material, in the range 1000-9999. The material must be defined in either the ULS standard database or the custom database (materials defined by the user.) A list of ID's for ULS-supported materials (such as paper, cotton, acrylic, aluminum and wood) will be provided on request.

The application accepts mil values for the material thickness that are greater than 0 and less than or equal to 12000 (12 inches.). The application reports whether or not the operation succeeds (See Figure 2).

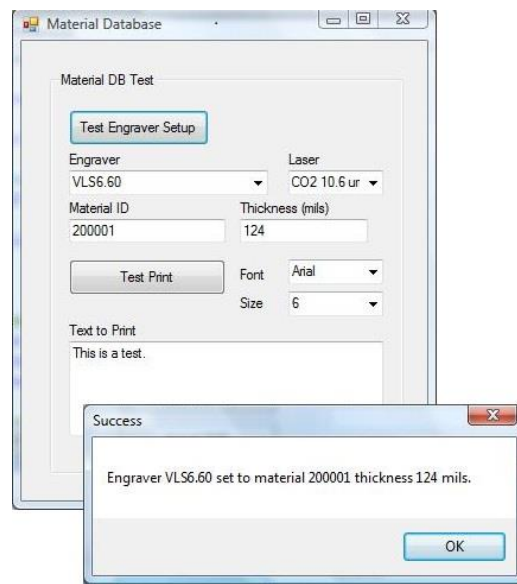


Figure 2: Remote interface when engraver settings are updated.

## ULS Remote Control GUI Details

The "Remote Control Diagnostics" GUI includes the following controls (left column)

- Show UCP - display UCP property sheet
- Hide UCP - hide UCP property sheet
- Start button - starts the job
- Pause button - stops the job
- Resume - resumes the job
- Estimate Job – command to estimate the completion time
- Find Job – Using the job in edit box, a command is sent to find it and index to its location in the queue. The Job name should include the extension.
- CopyLAS – Copy a standard LAS file using the UCP to ULS data directory. Call one time before importing/ inserting a batch of .emf or .dxf files.

- Import DXF - Using the path in edit box, a command is sent to automatically import the DXF file. The Dxf parameters will set sorting and center image, both = 3, sort only = 2, center image only = 1, neither = 0.
- Insert DXF - Inserts dxf job at Index in queue
- Sort Checkbox – sorting on/off selection for DXF import function
- Center Image Checkbox – image centering on/off selection for DXF import function
- Delete Last Job Checkbox – if checked, delete the last job when the job has finished engraving (useful for kiosk applications.)
- Material Database Button – launches Material Database GUI
- Purge - purge all of the jobs (accompanied by a warning)

The “Remote Control Diagnostics” GUI includes the following informational fields (right column)

- Name – Name of currently displayed print job
- Job Index – the index number in job queue
- Job Count – total number of jobs in queue
- X Position of the engraver in mils from left edge
- Y Position of the engraver in mils from top edge
- Machine state - state of the machine and the engraving process
- Interlock status - door open or closed
- Find Job – name of job to search for “Find Job” button
- Copy LAS – name of LAS file to copy for “Copy Las File” button
- Import DXF – name of DXF file for “Import DXF” or “Insert DXF”
- Index in Queue – numeric value or “Insert DXF” function
- Run Time - job completion time
- Estimated Run Time - estimated job completion time
- Status line - updates the status of the machine
- Events (in status line) - job stopped or door transition
- Machine Busy (in status line) - machine initializing or engraving
- Machine Ready (in status line) - this condition displays true when the engraver has power and is in the home position

The “Materials Database” GUI includes the following controls

- Test Engraver Setup button – clicking this button initializes the selected engraver with the ULS material ID and thickness in mils, as specified below.
- Engraver selection control – this displays all installed printers
- Laser selection control – for multi-laser engraver models, allows selection of active laser type
- Material ID – accepts a ULS material ID
- Thickness – thickness of material in mils.
- Test Print button – when clicked, generates a test print job containing rastered text which is sent to the engraver
- Font selector – selects font style for test print

- Size selector – selects font size (in points) for test print
- Text to Print – field for entering text to use in test print job

## Start Up

When the Remote Interface starts up, it will display the position, the machine state, interlock status, run time, est. run time, machine busy and event. The start, pause, resume and estimate job buttons are disabled until the engraver is connected or in IDLE state (Figure 4). When the engraver is connected and ready to accept commands, the machine ready will display “yes”. The X-Y position of the machine, machine state, interlock status, run time, and est. run time will be updated periodically.

## Print Queue Setup

The Find Job button will locate a job in the queue and if the function is successful then both job names in the Find Job text box and Jobs Name text box will be the same. The DXF import function provides options to import with sorting On/OFF and Center Image On/Off. To demonstrate the change to the display name, view panel and job count after a DXF job is imported successfully see Figure 3, 4, 5, and 6.

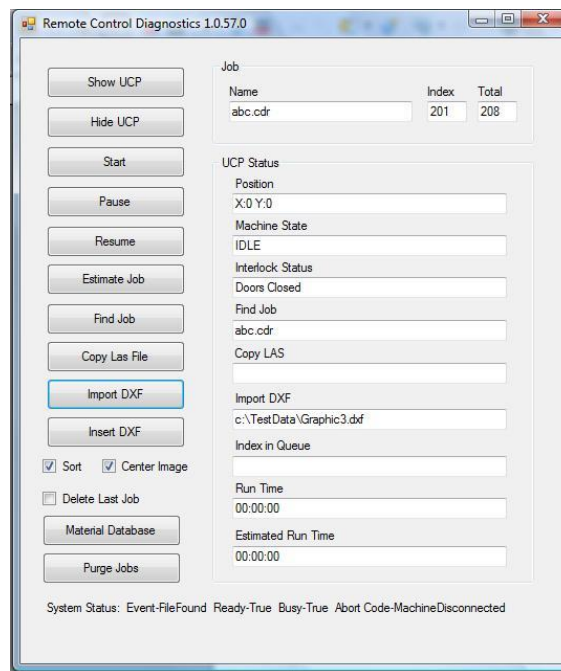


Figure 3: Job Count = 208 Job Name: Graphic3.dxf

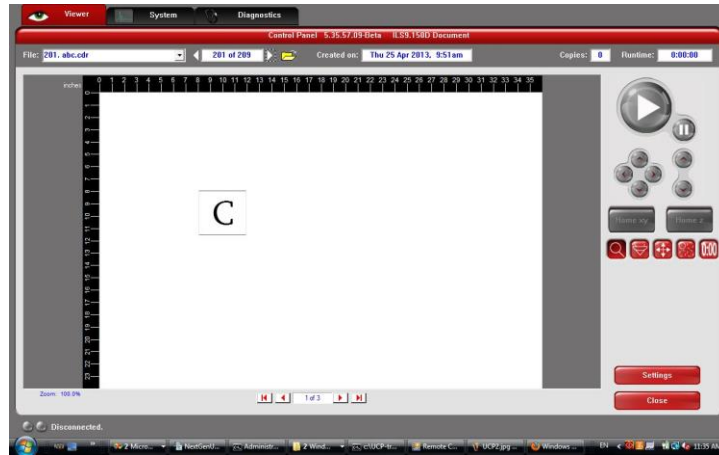


Figure 4: Selected Job in queue abc.cdr, Index in queue 201 of 208 jobs

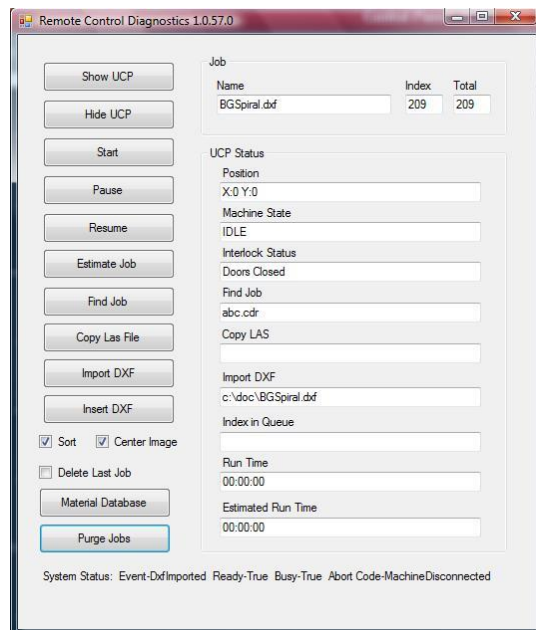


Figure 5: Imported DXF, display name changed and count incremented to 209 jobs

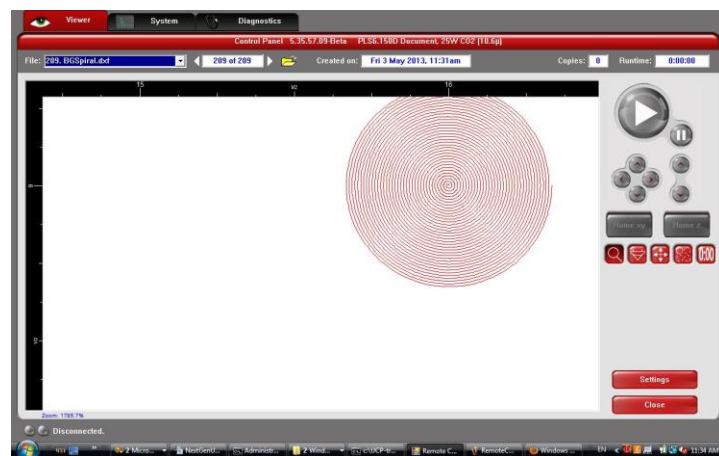


Figure 6: Imported DXF, UCP display name changed and count incremented to 209 jobs

When the user inserts a dxf job, he provides an index number, and an indicator that the job is 'last in first out' or 'first in first out.'

- 'last in first out' will append the job to the end of the queue which is the first job to run.
- 'first in first out' will insert the job at the start of the queue which is the first job to run.

## Engraving

Press start to begin engraving the current job in the queue. The position of the machine, machine state, interlock status, and run time will update while the machine is running. Press pause to stop engraving the job. Press resume to resume engraving the job from the stopping point. When the engraver completes the current job, an event will be fired from the DLL to notify the C# interface that the engraver has stopped, at which time the completion time of the current job will be updated.

## 3. DEVELOPMENT

### ULS Remote Control DLL functions

1. HideUCP() - hides UCP view panel
2. ShowUCP()- shows UCP view panel
3. GetCarriageLocation(POINT& XY) - dynamically provides the location of the engraver carriage in mils.
4. StartJob() - start the currently selected job
5. StopJob() - stop the curently printing job
6. ResumeJob() - resume the paused job
7. CopyLAS(LPCTSTR szFileName, int nLength) - Copy a standard LAS file into data directory from which UCP can import it.
8. DeleteLastJob(**bool** bDelete); Delete last job on job complete (true or false).
9. EstimateJob() - runs estimate command for curent job.
10. EstEngravingTime() - command to get estimated completion time derived from previous EstimateJob command.
11. FindJobName(LPCTSTR m\_strPrintFileName, int nLength)- finds job with the given name in queue and displays it in the UCP for run. (Note: there may be more than one job in queue with same name.)
12. GetAbortCode() - command to get abort code
13. GetEvent() - get most recent event code
14. GetJobName(LPCTSTR szFileName, int nLength) - Get the name of the job at the end of the queue.
15. ImportDXF(LPCTSTR m\_strPrintFileName, int nLength, unsigned int nDXFParameters) - Imports dxf job and adds it to the end of the queue. The Dxf parameters will set sorting and center image, both = 3, sort only = 2, center image only = 1, neither = 0.
16. InsertDXF(**Char**[] strDXF, **int** nLength, **uint** nDxfParamters, **int** nIndex, **bool** bLifo); Inserts dxf job at DXF index number, boolean bLifo for scenario A 'last in first out' and scenario B 'first in first out'.
  - For scenario A; insert 'last in first out' by calling InsertDXF with Index set to 0 and "true". This will append the job to the end of the queue which is the first job to run. Calling InsertDXF with

- Index set to JobCount and "true". This will insert the job at the start of the queue which is the last job to run.
- For scenario B; insert 'first in first out' by calling InsertDXF with Index set to 0 and "false". This will insert the job at the start of the queue which is the last job to run. Calling InsertDXF with Index set to JobCount and "false". This will insert the job at the end of the queue which is the first job to run.
17. InsertJob(LPCTSTR szFileName, in nLength, int nIndex); Insert job from files in Universal Laser's EMF format.
  18. InterlockStatus() - status of the doors (open/closed) using the UCP. Call this function once before importing/inserting a batch of .emf or .dxf files.
  19. JobCount() - Get the number of jobs in the Queue.
  20. JobIndex() - Get job index in the Queue.
  21. JobRunTime() - command to get completion time
  22. MachineBusy() - returns true if machine initializing or engraving;
  23. MachineReady() - returns true if machine is ready to engrave
  24. MachineState() - return engraver control virtual machine's state code
  25. PurgeJobs() - deletes all the jobs in the queue
  26. SelectJob(int nIndex) - The job will be selected for display and run. Setting the Index to JobCount will select the Number 1 job and setting the Index to 0 will select the job at the end of the queue.
  27. SetAppName() - returns current virtual machine state code
  28. SendTextCommand(const WCHAR\* cmd, WCHAR \*szReply, int nMaxReplySize) Send text command to program, sending szReply as place to store reply.
  29. UpdateDeviceInfo - Get current device state info from the controller
  30. KillPipe() - disconnects the pipe to UCP
  31. ReconnectPipe() - reconnects the pipe to UCP
  32. SetRemoveEventsCallback(MulticastDelegate callback) - designates a callback function which will be called on one of the possible events: JobCompleted, DoorChanged, FileFound, DxfImported, PipeConnected, PipeDisconnected, or JobAborted.

Note: the following C# Import DLL Functions are set to Unicode

```
[DllImport("RemoteCtrl.dll", CharSet = CharSet.Unicode)]
public static extern bool CopyLAS(Char [] FileName, int nLength);
[DllImport("RemoteCtrl.dll", CharSet = CharSet.Unicode)]
public static extern bool FindJobName(Char [] strJob, int nLength);
[DllImport("RemoteCtrl.dll", CharSet = CharSet.Unicode)]
public static extern bool ImportDXF(Char [] strDXF, int nLength, uint
nDXFParameters);
[DllImport("RemoteCtrl.dll", CharSet = CharSet.Unicode)]
public static extern bool InsertDXF(Char [] strDXF, int nLength, uint
nDXFParameters);
[DllImport("RemoteCtrl.dll", CharSet = CharSet.Unicode)]
public static extern bool GetJobName(IntPtr JobName, int nLength);
```

Note: the C# Default DXF Import Path string

```
static string strDxf = "C:\\TestData\\Graphic3.dxf";
```

## ULS Material Database DLL functions

1. GetMaterial(int nMatID, int nThick, char\* r szEngName, char\* Settings); - set material ID and thickness (in mils) for engraver with printer name specified by string szEngName.



2. `SelectMaterial(char* pSettings, char* pDevMode);` - set specific engraver parameters from `pSettings` into `DEVMODE` structure for a particular engraver.
3. `SetEngraverMaterial(int nMatID, int nThick, char* szEngName);` - retrieves material by ID and configures specified engraver's `DEVMODE` structure. On failure, returns zero. On success, returns ID that was set (ID may have been converted from legacy value to new value.)
4. `MatSettingsLen();` - returns length in bytes of the ULS `pSettings` structure.