

# PTBurn 3 SDK

**PRIMERA**  
TECHNOLOGY, INC.

Network-capable, Text-based, Shared Folder Burning & Printing

Introduction

Job File Command Structure

Programming Examples

Robot Status Information

Client Commands

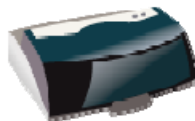
Compatible with:



**Bravo™ XRP**  
DISC PUBLISHER



**Bravo™ XR**  
DISC PUBLISHER



*Bravo II*  
DISC PUBLISHER



*Bravo Pro*  
DISC PUBLISHER



**Bravo<sup>se</sup>**  
Disc Publisher



**BravoProXi**  
Series



**Bravo 4100**  
Series

Primera Technology Inc.  
March 11, 2011 / Revision 3.23

1	Introduction .....	5
2	JRQ File Command Structure .....	6
2.1	Command Keys/Values .....	6
2.1.1	JobID.....	6
2.1.2	ClientID.....	6
2.1.3	Importance.....	6
2.1.4	Data.....	6
2.1.5	AudioFile .....	7
2.1.6	ImageFile.....	8
2.1.7	DeleteFiles.....	9
2.1.8	ImageType.....	9
2.1.9	DataImageType .....	10
2.1.10	CloseDisc .....	10
2.1.11	Copies.....	11
2.1.12	PrintLabel .....	11
2.1.13	VolumeName .....	11
2.1.14	BurnSpeed.....	11
2.1.15	VerifyDisc .....	11
2.1.16	RejectIfNotBlank .....	11
2.1.17	NotifyClient .....	11
2.1.18	PreserveISOVariations .....	12
2.1.19	ReadDataTo.....	12
2.1.20	ReadDataFormat.....	12
2.1.21	CreateSubFolders.....	12
2.1.22	CheckFileOnDisc.....	13
2.1.23	CheckSystemIDOnDisc .....	13
2.1.24	CheckVolumeIDOnDisc .....	13
2.1.25	CheckPubIDOnDisc.....	13
2.1.26	CheckPreparerIDOnDisc.....	13
2.1.27	CheckAppIDOnDisc .....	13
2.1.28	PVDSystemID.....	14
2.1.29	PVDPublisherID .....	14
2.1.30	PVDPreparerID .....	14
2.1.31	PVDApplicationID.....	14
2.1.32	LoadUnloadOverride.....	14
2.1.33	PreMasterData .....	14
2.1.34	DriveID .....	14
2.1.35	BinID.....	15
2.1.36	TestRecord .....	15
2.1.37	PrintQuality .....	15
2.1.38	PrintInnerDiam.....	16
2.1.39	PrintOuterMargin.....	16
2.1.40	MergeField .....	16
2.1.41	RobotName .....	17
2.1.42	PrintReject.....	17

2.1.43	CDTextDiscTitle .....	17
2.1.44	CDTextDiscPerformer.....	17
2.1.45	CDTextDiscComposer.....	18
2.1.46	CDTextTrackTitle .....	18
2.1.47	CDTextTrackPerformer .....	18
2.1.48	CDTextTrackComposer.....	18
2.2	Programming Examples .....	18
2.2.1	Simple Data Job.....	18
2.2.2	Simple Audio Job.....	19
2.2.3	Burn and Print job of 5 discs with many options specified .....	19
2.2.4	Print Only Job of 25 Discs.....	20
2.2.5	Read Data Job of 3 Discs .....	20
2.2.6	Read Global Image Job of 3 Discs .....	20
3	Status Information .....	21
3.1	SystemStatus.txt File.....	21
3.1.1	Example of SystemStatus.txt .....	21
3.1.2	Robot List .....	22
3.1.3	Robot Section .....	22
3.1.4	System Section .....	24
3.2	RobotName.txt Status Files .....	26
3.2.1	Example of RobotName.txt (e.g. Disc Publisher XRP.txt) .....	26
3.2.2	JobList Section.....	28
3.2.3	CompletedJobs Section.....	28
3.2.4	Job Details Section .....	29
3.2.5	System Section .....	32
3.2.6	Drive Statistics Section.....	36
4	Client Commands.....	38
4.1	Abort Command.....	38
4.2	Shutdown After Job Command.....	38
4.3	Shutdown Immediate Command.....	38
4.4	Align Printer Command.....	39
4.5	Ignore Ink Low Command .....	39
4.6	Process Disc Command .....	39
4.7	Reject Disc Command .....	40
4.8	Unload Command .....	40
4.9	Check Bins Command .....	40
4.10	Create Report Command .....	40
4.11	Generate Preview .....	41
4.12	Run Utility Command.....	41
4.13	USB Connect.....	41
4.14	USB Disconnect .....	41
4.15	Set/Clear Kiosk Mode.....	41
5	Coding Best Practices.....	42
5.1	Programming Tips .....	42
5.1.1	Reading the Status File.....	42
5.1.2	Determining if the service is Running.....	42

---

5.1.3	Starting the service.....	42
5.2	Merge Printing .....	42
6	Appendix A: PTSetup.ini.....	45
7	Appendix B: Deployment Details .....	50
8	Appendix C: Job, System and Burn Engine Error Codes.....	51
9	Appendix D: Loading/Unloading Override.....	65
10	Revision History .....	67

# 1 Introduction

Primera's **PTBurn** SDK service allows any client application to easily create discs with customized burn data and printed label information. It is ideal for use with customer-supplied "front-end" applications in networked or custom environments in markets such as medical imaging, digital photo processing, music/video/software on-demand and much more. The supported devices for this application are Primera's *Disc Publisher SE*, *Disc Publisher II*, *Disc Publisher PRO*, *Disc Publisher XR*, *Disc Publisher XRP*, *Disc Publisher Pro Xi-Series*, and *Disc Publisher 4100 Series*.

The client application creates an ASCII text file (Job Request File) in the Job Request Folder (shared "hot-folder"). The Job Request File specifies the image data to burn, the label to print, and the number of copies to create. The PTDisk service scans this Job Request Folder for new jobs submitted by any client application and creates the discs automatically on the Disc Publisher. There is both a Unicode version and a Multibyte (MBCS/ANSI) version of PTDisk SDK; use the Unicode version for burning and printing 2-byte characters.

The PTDisk Windows Service works on Windows XP, Windows Vista or Windows 7. Client machines can run any operating system that is able to read and write into a Windows shared folder. It can burn either data or audio discs, as well as create discs from a disc image.

Changes can be made to the PTSETUP.INI file to customize certain settings. See [Appendix A](#) for more details on the PTSETUP.INI file

## 2 JRQ File Command Structure

To create discs, the client application must create a Job Request File. This text file can be named anything the client chooses but it **MUST** reside within the Job Request Folder **AND** must have an extension of .JRQ (this can be changed in PTSETUP.INI). This .JRQ file created by the client contains information that specifies the data to burn, number of copies, etc.

The Job Request File must be a text document with certain Key values defined (many of which are optional). The format is:

### 2.1 Command Keys/Values

#### 2.1.1 JobID

Any ASCII string message up to 32 characters describing the job. When the job is being executed this JobID string will be shown in the PTBurn window. If this key is not provided then the job request filename will be used.

#### 2.1.2 ClientID

This ASCII string message up to 32 characters is a description for the Client who is submitting the job. The client can decide what string he wants to use. If this key is not provided then the ClientID will default to "Unknown."

#### 2.1.3 Importance

This tells PTBurn how important the job is. Possible values are **0** to **400** where **0** is most important and **400** is least important. If this key is not provided then a default value of **4** will be used. Jobs with lower values will be processed first unless the job has been waiting too long (this value can be changed in PTSETUP.INI). NOTE: When submitting jobs of different priorities it is important to submit jobs with higher priority before jobs with lower priority; jobs that are discovered by PTBurn will not be interrupted *once they actually begin*, even if a job with a higher priority is discovered later.

#### 2.1.4 Data

This is a string that specifies the either the filename to burn or the folder that contains all the data to be burned. All files and subfolders within a specified folder will be burned (the root folder itself will not be copied – only its contents). Alternatively, individual files can be specified here (the full path for the source file must be given, but the file will be copied to the root location on the disc). The format is as follows:

**Data = Filename, SessionX**

Where SessionX is *optional* and is used only if requesting a multi-session disc (a maximum of 2 sessions is allowed at this time). The value X is an integer value (1 or 2) that tells PTBurn for which session to burn this data.

If SessionX is not given then it will be burned into the first (and only) session.

e.g.

Data = C:\MyFolder1, Session2

(This tells PTDurn to burn the contents of C:\MyFolder1 to session 2 on the disc. In this example, there must also be an AudioFile= or ImageFile= entry for Session 1).

See the DataImageType key below to specify burning options including file system.

**The only way the Data key can be used with the AudioFile keys or the ImageFile key is if the SessionX value is used. The Data key must not be used with AudioFile or ImageFile with the same session value.**

For added flexibility you can also burn data from a source location and change the destination name and/or location of the data to be burned onto the disc. For example, you can copy files from the source hard drive and rename them. You can also place them into a new folder on the disc. Also, you can copy an entire source folder and place them into any specified destination folder on the disc.

The format for this is as follows:

**Data = Source Filename>\Destination Filename**

Or

**Data = Source Folder>\Destination Folder**

Note the destination filename and folder should be preceded by a “\” to denote the root of the disc.

Examples:

Data = C:\MyFolder1\MyFile.txt>\YourFile.txt

This will burn C:\MyFolder1\MyFile.txt to the root of the disc and change the name to YourFile.txt

Data = C:\MyFolder1\MyFile.txt>\YourFolder1\YourFile.txt

This will burn C:\MyFolder1\MyFile.txt as YourFile.txt into a folder on the disc called YourFolder1 (note YourFolder1 will automatically be created on the disc).

Data = C:\MyFolder1>\YourFolder1\YourFolder2

This will burn the entire contents of C:\MyFolder1 into a new folder on the disc called YourFolder2 (which is created as a subfolder within YourFolder1)

### 2.1.5 AudioFile

This is a string that specifies the name of an audio file to burn.

You can optionally specify a Pregap value (in sectors) of the track, the Session, or the ISRC code for the disc. So, the format is as follows:

AudioFile = Filename, PregapX, SessionX, ISRCxxxxxxxxxxx

Up to 99 AudioFile = statements are allowed (99 total if Data and ImageFile statements used also).

Acceptable audio filename types include .WAV, .MP3, and .WMA.

PregapX (audio silence) values are in sectors so 75 sectors = 1 second, 150 sectors = 2 seconds, and so on. The first audio track will always have a Pregap of 150 even if another value is specified. If Pregap is not specified then a value of 150 will be used as the default.

SessionX is optional and is used only if requesting a multi-session disc (a maximum of 2 sessions are allowed at this time). The value X is an integer value (1 or 2) and tells PTBurn for which session to burn this audio file. If SessionX is not given then it will be burned into the first (and only) session

e.g.

AudioFile = MySong.mp3, Session1

(This tells PTBurn to burn the audio file MySong.mp3 to session 1 on the disc as cd audio. In this example there may be a Data= or ImageFile= entry for Session 2).

**The only way this key can be used with the Data keys or the ImageFile key is if the SessionX value is used. AudioFile must not be used with Data or ImageFile with the same session value.**

ISRC is another option for each audio track. The ISRC value xxxxxxxxxxxx is a 12 character sequence that describes each audio track. See <http://en.wikipedia.org/wiki/Isrc> for details of valid values for the 12 characters.

e.g.

AudioFile = MySong1.wav, ISRCUSPR37300012

AudioFile = MySong2.wav, ISRCUSPR37300013

### 2.1.6 ImageFile

This specifies the name (including path) of an already-mastered disc image to be burned. The acceptable disc image types include .GI or .ISO.

The format is as follows:

ImageFile = Filename, SessionX



Where SessionX is optional and is used only if requesting a multi-session disc (a maximum of 2 sessions are allowed at this time). The value X is an integer value (1 or 2) that tells PTBurn for which session to burn this image.

If SessionX is not given then it will be burned into the first (and only) session.

e.g.

```
ImageFile = c:\MyImage.ISO, Session1
```

(This tells PTBurn to burn the disc image MyImage.ISO to session 1 on the disc. In this example, there may be Data= or AudioFile= entries for Session 2).

The only way this key can be used with the Data keys or the AudioFile keys is if the multi-session SessionX value is used. ImageFile must not be used with Data or AudioFile keys with the same session value.

### 2.1.7 DeleteFiles

This specifies if the files/folders/image for this particular job, should be deleted after the job is completed. Possible values:

YES

NO

The default value if this key is not given is NO. In order for the files/folders to be deleted they must reside within a subfolder of the Shared Job Folder. Everything within the first subfolder under the Shared Job Folder (including the subfolder) will be deleted.

If DeleteFiles is set to YES, the job data and print files will be deleted upon the job completing (Pass, Fail, or Abort). The job request file (now with a .don or .err extension) will not be deleted immediately. The deletion of this file will occur when the time set in PTSETUP.ini (Status\_Time) has expired after job completion.

### 2.1.8 ImageType

This specifies the mode (block length) to use when burning the ImageFile disc image. The possible values are as follows:

MODE\_1\_2048

MODE\_2\_2336

MODE\_2\_2352

This key only applies to .ISO disc images.

Note that using the wrong value here can cause the image to not fit and lead to an unusable disc.

### 2.1.9 DataImageType

This specifies what kind of image should be created for a data burn (when DataFiles key is given). Multiple items are possible so the format is:

DataImageType = FileSystem, Option1, Option 2, ...

Possible values of FileSystem are the following:

**ISOLEVEL1** – for an ISO 9660 Level 1 compliant CD with folder and file names in the 8+3 format using only the A-:~Z, 0-~9 and “\_” character set.

**ISOLEVEL2** – for and ISO 9660 Level 2 compliant CD (DOS OEM characters). Filenames are 8+3 format (use ISOLEVEL3 for long filename support).

**ISOLEVEL3** – for an ISO 9660 Level 3 compliant CD (DOS OEM characters, long file names)

**JOLIET** – for a Microsoft Joliet compliant CD with filenames up to 106 characters.

**ISOLEVEL1+JOLIET** – for an ISO 9660 Level 1 with Joliet CD.

**ISOLEVEL2+JOLIET** – for an ISO 9660 Level 2 with Joliet CD.

**ISOLEVEL3+JOLIET** – for an ISO 9660 Level 3 with Joliet CD.

**UDF** – for a UDF 1.02 bridge file system

**UDF201** – for a UDF 2.01 bridge file system

**UDF250** – for a UDF 2.50 bridge file system

**UDF260** – for a UDF 2.60 bridge file system

Possible values of Option1, Option2, ... are the following: **SETNOW** – this will cause all files to be set to the time/date at which the disc is created. If this option is not given then the file's time/date will keep their original values.

**MODE2** – this will make a CD-ROM Mode 2 XA disc. If this option is not given then the disc created will be Mode 1.

This option is valid only for a CD-R, and is ignored for a DVD-R.

**SAO** -- if the disc must be written Session At Once. When SAO is not specified the recording is done Track At Once (TAO). TAO is required for DVD multi-border (DVD-R, DVD-RW) but is optional for CD-R, CD-RW.

(If this key is not provided, the default will be a Joliet, Mode1, Track at Once Disc).

Note: If choosing ISOLevel1 the filenames must adhere to the ISOLevel1 naming convention (8+3 using A-Z, 0-9, and \_ only). If any files/folders in the job do not adhere to this convention the job will fail.

**Note: If burning DVD Video (VIDEO\_TS), we recommend that you select UDF to allow for better playback compatibility with dvd players.**

### 2.1.10 CloseDisc

This specifies if the disc must be closed so that no other session can be added. Possible values:

**YES**

**NO**

The default value is NO if this key is not given.

### 2.1.11 Copies

This specifies how many copies you want created. The range of values is from **1** to **10,000**. The default is **1** if this key is not given.

### 2.1.12 PrintLabel

This specifies path and filename of the label to print on disc.

The possible file types are .STD (SureThing™), .jpg (JPEG), .bmp (Windows Bitmap), or .PRN (printed to file through any application). If this key is not given then no printing will be performed. For printing merge text and pictures (unique text and/or pictures on each disc) see the supporting SureThing documentation from Microvision (some tips are also given below).

### 2.1.13 VolumeName

This string specifies the volume label name. The string can be up to 32 characters, but usually only the first 11 characters are visible on a system. If this key is not given then the default will be the JobID string (if the JobID string is not given then the default will be "NewDisc")

### 2.1.14 BurnSpeed

This integer value specifies a requested recorder burn speed. The range is from **1** to **100**. This corresponds to 1x to 100x (even though 100x recorders are currently not available). If this key is not given then the recorder will burn at MAX speed (or you can set BurnSpeed=0 for MAX speed). If the recorder is not capable of the requested burn speed, then the nearest available speed will be used.

### 2.1.15 VerifyDisc

This specifies if the disc must be verified or not. Possible values:

**YES**

**NO**

The default value is NO if this key is not given.

### 2.1.16 RejectIfNotBlank

This specifies if a disc should be rejected if is not blank. Possible values:

**YES**

**NO**

The default value is NO if this key is not given. When a value if YES is given then the disc will be rejected unless blank media is inserted. NO means that it will attempt to add a session if one already exists.

### 2.1.17 NotifyClient

This key specifies if the client wants to be notified when their job is complete (pass, fail, or abort).

Possible values:

**Message**  
**Disabled**

The default value is **Disabled** if this key is not given. When a value of **Message** is given then the client will be notified via a net message. If this feature is used, the ClientID key must be set to the client's computer name. This function works for Windows XP but not Windows Vista or Windows 7.

#### 2.1.18 PreserveISOVariations

This key forces the recording engine to preserve the file name as is even if it is out of specification. Possible values:

**YES**  
**NO**

The default value is NO if this key is not given.

#### 2.1.19 ReadDataTo

This key specifies the location to which data from the disc(s) should be stored (this is useful for saving the contents of a disc for later use). If the ReadDataFormat key is set to ReadData, then this key contains the path of where the data from the disc should be written to the hard disk drive (ex. C:\MyDisc1). If ReadDataFormat is ReadGI, then ReadDataTo specifies the path and GI file name that should be created (ex. \\Computer\PTBurnJobs\ReadFile would create the global image file \\Computer\PTBurnJobs\ReadFile.gi ). When using ReadGI as the format you should NOT specify the .gi extension in the ReadDataTo key because PTBurn will append the .gi extension to the name.

#### 2.1.20 ReadDataFormat

This key specifies the format in which the disc(s) will be stored onto the hard disk drive (into the location specified by ReadDataTo). The possible values are as follows:

**ReadGI**  
**ReadData**

The default value is ReadData, which will read the data from the disc and store the same data onto the hard disk drive. ReadGI will create a Global Image file of the Disc. If the format is ReadGI and the number of copies is greater than 1, then a \_0, \_1... will be appended to the GI filename.

NOTE: any files, including .GI files will be overwritten if they already exist.

#### 2.1.21 CreateSubFolders

This key will cause PTBurn to create a subfolder below the ReadDataTo path for each disc read. The subfolders will be named Disc\_0, Disc\_1 ... The possible values are:

**YES**  
**NO**

The default value is NO if this key is not given. This key is only valid if ReadDataTo is specified, ReadDataFormat = ReadData, and the number of copies is greater than 1.

NOTE: Before starting the job, the subfolders Disc\_0, Disc\_1, ..., should be deleted, renamed, or moved if they already exist in the location specified by ReadDataTo.

### 2.1.22 CheckFileOnDisc

This key will cause PTBurn to check the disc prior to recording to see if the specified file exists. The value of this key should be the complete path (on the disc) of the file. For example:

```
CheckFileOnDisc=Server\myFile.txt
```

will cause PTBurn to check to see if the file “myFile.txt” is on the disc in the “Server” folder. If the file is not found then an Invalid Media Error (Job/Disc Error 10) will be set.

### 2.1.23 CheckSystemIDOnDisc

This key will cause PTBurn to check the disc prior to recording to see if the SystemID value in the PVD matches the value of this key.

```
CheckSystemIDOnDisc=PTBurn
```

will cause PTBurn to check to see if the system ID in the PVD on the disc is set to “PTBurn” . If the value is not found then an Invalid Media Error (Job/Disc Error 10) will be set.

### 2.1.24 CheckVolumeIDOnDisc

This key will cause PTBurn to check the disc prior to recording to see if the VolumeID value in the PVD matches the value of this key.

If the value is not found then an Invalid Media Error (Job/Disc Error 10) will be set

### 2.1.25 CheckPubIDOnDisc

This key will cause PTBurn to check the disc prior to recording to see if the PublisherID value in the PVD matches the value of this key.

If the value is not found then an Invalid Media Error (Job/Disc Error 10) will be set.

### 2.1.26 CheckPreparerIDOnDisc

This key will cause PTBurn to check the disc prior to recording to see if the DataPreparerID value in the PVD matches the value of this key.

If the value is not found then an Invalid Media Error (Job/Disc Error 10) will be set.

### 2.1.27 CheckAppIDOnDisc

This key will cause PTBurn to check the disc prior to recording to see if the ApplicationID value in the PVD matches the value of this key.

If the value is not found then an Invalid Media Error (Job/Disc Error 10) will be set.

### 2.1.28 **PVDSystemID**

This key will set the SystemID value of the PVD. This max number of characters for this key is 32.

### 2.1.29 **PVDPublisherID**

This key will set the PublisherID value of the PVD. This max number of characters for this key is 128.

### 2.1.30 **PVDPreparerID**

This key will set the PreparerID value of the PVD. This max number of characters for this key is 128.

### 2.1.31 **PVDApplicationID**

This key will set the ApplicationID value of the PVD. This max number of characters for this key is 128.

### 2.1.32 **LoadUnloadOverride**

This key will enable the manual load and unload functionality.

**YES**

**NO**

The default value is NO if this key is not given. This key allows you to Pause after the drive is loaded (before burning) and also allows you to Pause after burning (before the drive is unloaded). It also allows you to Load/Unload the drive without burning (e.g. read disc contents only).

Please see [Appendix D](#) for details regarding manual load and unload.

### 2.1.33 **PreMasterData**

This key will enable pre-mastering of data jobs. This means that when the job is started a temporary image will be created and then each disc in the job will be burned from that image. Otherwise each disc will be burned as a separate data disc. We recommend setting this to YES if you are burning a large number of files, or there is the possibility that a file could change during the burn process.

**YES**

**NO**

The default value is NO if this key is not given.

**IMPORTANT NOTE:** If you are burning to DVD, then you must specify this with another key:

**DiscType = DVDR**

If not specified, then it will be assumed that you want to pre-master a CD.

### 2.1.34 **DriveID**

This Key is used to specify a drive to be used in a job. There can be more than one DriveID key in a jrj file. The DriveID should be set to the drive index, which can be found in the Status file (See DriveDescX in Section 3.4.22). If no DriveID is specified then all drives in the robot will be used. For Example:

If the JRQ Specifies

DriveID = 0

And the Status file has the following entry

DriveDesc0=HL-DT-ST CD-RW GCE-8526B 1.02

DriveLocation0=Robotic Recorder

Then, in this example, only the top drive will be used in the job.

### 2.1.35 BinID

This key is used to specify from which bin to pick a disc and where to output the disc. The behavior of BinID depends on whether Kiosk mode is set or not. (See Section 3.1.2.11 to determine if Kiosk Mode is enabled or not. See Section 4.15 to enable/disable Kiosk Mode). Valid values for BinID are 0, 1, or 2

#### Kiosk Mode Enabled:

<u>BinID</u>	<u>Disc Picked from</u>	<u>Disc Output to</u>
0	Both Bins	Front Kiosk area
1	Left Bin	Front Kiosk area
2	Right Bin	Front Kiosk area

#### Kiosk Mode Disabled:

<u>BinID</u>	<u>Disc Picked from</u>	<u>Disc Output to</u>
0	Right Bin	Left Bin
1	Left Bin	Front Kiosk area
2	Right Bin	Front Kiosk area

### 2.1.36 TestRecord

This key is used to specify if you want the disc to be tested instead or recorded. This is useful when running tests where you do not want to waste media by burning it. The possible values are

**YES**

**NO**

The default value is NO if this key is not given.

### 2.1.37 PrintQuality

This key specifies the print quality. If this key is not specified then the default driver value will be used. The possible values are listed below.

Low = 0

Medium =1

Better =2

High =3

Best =4

The default driver value can be change in the Printer Preferences dialog when viewing the printers on Windows.

#### 2.1.38 **PrintInnerDiam**

This key specifies the inner diameter of the print. The units for this key are .1mm. The range of values is from 150 – 500 (.1mm). If this key is not specified then the default driver value will be used.

The default driver value can be change in the Printer Preferences dialog when viewing the printers on Windows.

#### 2.1.39 **PrintOuterMargin**

This key specifies the outer diameter of the print. The units for this key are .1mm. The range of values is from 0 – 20 (.1mm). If this key is not specified then the default driver value will be used.

The default driver value can be change in the Printer Preferences dialog when viewing the printers on Windows.

#### 2.1.40 **MergeField**

This key specifies a “Merge” field for SureThing printing (either merge text or merge picture). Note that the print file specified within the JRQ must be a SureThing file, and it must have been designed with a Merge File specified.

This key is not needed if the merge file exists. The reason for this key is convenience; by using this key the merge file does not need to saved. Instead you can specify the merge fields within the JRQ using this key.

Example:

MergeField=Kevin’s Merge Job Disc

MergeField=C:\images\logo.bmp

MergeField=C:\Documents and Settings\kevin\My Documents\My Pictures\sample.jpg

This example is for a disc that was designed with one merge text field and two merge pictures. Note that the fields should be specified in the correct order to match the SureThing design.

TIP: When designing the original SureThing image you must be sure to follow these steps:

1. Open a new SureThing design.
2. Insert the number of desired merge fields in the SureThing design. Both text and pictures can be merged. For each type, follow these steps in SureThing --
  - a. Merge Text: Tools/Insert Field, Merge Fields tab, Field One, Two...
  - b. Merge Photo: Tools/Object Tools/Picture Tool, enable the ‘Merge’ checkbox, Field One, Two....



3. Create a “dummy” merge text file. In the example above we would create a file called MyMergeFile.txt with the following contents:  
    “Text Field 1”, “MyImage1.jpg”, “MyImage2.jpg”
4. Associate the merge text file with the SureThing design – to do this, click Tools/Set Merge File and browse to the newly created MyMergeFile.txt.
5. Save the new SureThing design file.

Now, this newly created SureThing design file (.STD) can be specified in PTDisk and the values specified with “MergeField” will be inserted into the image at print time.

#### Additional notes:

The merge text file is only required at design time; during run-time it is no longer required. If the original merge file is not found at print time, then a new file will be created in the same folder as the SureThing design file. However, if the original merge text file is found, then the values in it are populated with values from the JRQ MergeField values (and any unused fields in the merge file will be removed).

#### 2.1.41 RobotName

This key specifies a particular robot to use for the job. The name must be one listed in [RobotList] as described in Section 3.1.2. This is useful in a multiple robot scenario when you want to specify a particular robot for the job.

#### Example:

RobotName=Disc Publisher XRP (Copy 1)

#### Note:

If AutoSwitchRobot=TRUE in PTSETUP.INI and the robot specified using RobotName has an error, PTDisk will automatically switch to a robot without any errors.

#### 2.1.42 PrintReject

This specifies if the disc should be printed with “Reject” if the burn is rejected.

**YES**

**NO**

The default value is NO if this key is not given.

#### 2.1.43 CDTextDiscTitle

This key specifies the CD-Text field representing the disc’s title. This entry is required when doing an audio disc with CD-Text (one entry per disc). The maximum number of characters is 100. The total number of characters for CDTextDiscTitle and all CDTextTrackTitle entries cannot exceed 2000 characters

#### 2.1.44 CDTextDiscPerformer

This key specifies the CD-Text field representing the disc’s performer. This entry is required when doing an audio disc with CD-Text (one entry per disc). The maximum number of

characters is 100. The total number of characters for CDTextDiscPerformer and all CDTextTrackPerformer entries cannot exceed 2000 characters.

#### 2.1.45 CDTextDiscComposer

This key specifies the CD-Text field representing the disc's composer. This entry is required when doing an audio disc with CD-Text (one entry per disc). The maximum number of characters is 100. The total number of characters for CDTextDiscComposer and all CDTextTrackComposer entries cannot exceed 2000 characters.

#### 2.1.46 CDTextTrackTitle

This key specifies the CD-Text field representing the track's title. This entry is required for each track when doing an audio disc with CD-Text. The maximum number of characters for each entry is 100. The CD-Text fields will be associated with their respective track based on the order it appears in the Job Request File. For example:

```
AudioFile=myfile1.mp3  
AudioFile=myfile2.mp3
```

```
CDTextTrackTitle = First track  
CDTextTrackTitle = Second track
```

Track 1 will be myfile1.mp3 with the CD-Text Track Title for the track set to "First track". Track 2 will be myfile2.mp3 with the CD-Text Track Title for the track set to "Second track".

#### 2.1.47 CDTextTrackPerformer

This key specifies the CD-Text field representing the track's performer. This entry is required for each track when doing an audio disc with CD-Text. The maximum number of characters for each entry is 100.

#### 2.1.48 CDTextTrackComposer

This key specifies the CD-Text field representing the track's composer. This entry is required for each track when doing an audio disc with CD-Text. The maximum number of characters for each entry is 100.

## 2.2 Programming Examples

### 2.2.1 Simple Data Job

Test1. JRQ in the Job Request Folder has the contents:

```
Data = c:\MyBackup
```

This will burn one CD-R disc with all the files and subfolders within the c:\MyBackup folder. It will be a Joliet file system, Mode 1, Track-at-once, not-closed disc (see other defaults above).

### 2.2.2 Simple Audio Job

Test2. JRQ in the Job Request Folder has the contents:

```
AudioFile = c:\MyMusic\MyTrack1.wav
AudioFile = c:\MyMusic\MyTrack2.mp3, Pregap75
AudioFile = c:\MyMusic\MyTrack3.wma, Pregap150
BurnSpeed = 8
CloseDisc = YES
Copies = 2
```

This will burn 2 CD-R audio discs with three tracks. The .wav, .mp3, and .wma will be converted and burned as CD-Audio automatically. MyTrack2 will have a 1 second pause before it and MyTrack3 will have a 2 second pause before it. The discs will be burned at 8x (or as close as possible) and the disc will be closed.

### 2.2.3 Burn and Print job of 5 discs with many options specified

Test3. JRQ in the shared-folder location have the contents:

```
JobID = Kevin's Photos
ClientID = COMPANYYX-KEVIN1
Data = \\SERVER\PTBurnJobs\KevinsJob\Photos\Data
Data = c:\MyBackup\File1.jpg
DeleteFiles = YES
DataImageType = ISOLEVEL1, SAO, SETNOW
CloseDisc = YES
Copies = 5
PrintLabel = \\SERVER\PTBurnJobs\KevinsJob\Print\Photos.std
VolumeName = My Photos
VerifyDisc = YES
NotifyClient = Message
```

This will burn 5 CD-R discs with all the files and subfolders within the subfolder \\SERVER\PTBurnJobs\KevinsJob\Photos\Data and it will also burn the file C:\MyBackup\File1.jpg. The disc will be verified after burning. The SureThing label file \\SERVER\PTBurnJobs\KevinsJob\Print\Photos.std will be printed on each disc. The discs will be closed-session Disc-at-once with ISO 9660 Level 1 file system, and the time/date of the files will be set to the time at which each disc is burned. After completing the job, PTBurn will delete all the files and subfolders within and including the subfolder \\SERVER\PTBurnJobs\KevinsJob. Note that the file C:\MyBackup\File1.jpg will not be deleted. Because the NotifyClient key is set to Message, when the job is completed a net message will be sent to the computer COMPANYYX-KEVIN1.

#### 2.2.4 Print Only Job of 25 Discs

JobID = My Print Job  
ClientID = MyWorkstation  
PrintLabel = \\SERVER\PTBurnJobs\MyJob\MyImage.STD  
Copies = 25

#### 2.2.5 Read Data Job of 3 Discs

ClientID=MyComputer  
JobID=Job0  
Importance=2  
Copies=3  
NotifyClient=Message  
ReadDataTo=\\MyComputer\PTBurnJobs\ReadData  
ReadDataFormat=ReadData  
CreateSubFolders=YES

#### 2.2.6 Read Global Image Job of 3 Discs

**ClientID=MyComputer**  
**JobID=Job0**  
**Importance=2**  
**Copies=3**  
**NotifyClient=Message**  
**ReadDataTo=\\MyComputer\PTBurnJobs\ReadData**  
**ReadDataFormat=ReadGI**

This will create a GI global image file for each of the 3 discs at the following locations:

\\MyComputer\PTBurnJobs\ReadData\_1.gi  
\\MyComputer\PTBurnJobs\ReadData\_2.gi  
\\MyComputer\PTBurnJobs\ReadData\_3.gi

## 3 Status Information

PTBurn will provide status for each submitted job in several different ways.

First of all, the filename extension for the submitted Job Request File will have four different values: the “new job request” extension, the “job discovered” extension, the “in process” extension, and the “job completed” extension. All four of these extensions can be specified in PTSETUP.INI. Using the default values here is an example description of the four filename extensions:

MyJob.JRQ → The client has submitted the job request  
MyJob.QRJ → PTFurn has discovered and put the job in its list of jobs.  
MyJob.INP → PTFurn is currently processing this job.  
MyJob.DON → The job has been completed.

An additional filename extension is possible when a job has an error or has been aborted.  
MyJob.ERR → The job has finished with errors or has been aborted.

Additional status can be retrieved from text files containing additional detailed status information.

### 3.1 SystemStatus.txt File

PTBurn will provide status for the entire system via the **SystemStatus.txt** file. This file will reside in a subfolder called **Status** within the Job Request Folder location defined in PTSETUP.INI (the default is C:\PTBurnJobs).

SystemStatus.txt contains at least three sections; the structure of the file will be similar to an .INI file with two fixed Sections of [RobotList] and [System], and one or more sections [RobotName] where RobotName is the name of the Robot (e.g. Disc Publisher XRP).

For more detailed information, PTFurn also creates one or more status files called RobotName.txt, where RobotName is the name of the robot (e.g. Disc Publisher XRP.txt). See section 3.2 for more details on this.

Note: When processing SystemStatus.txt from a client application, the client application should **NOT** access SystemStatus.txt with exclusive access.

#### 3.1.1 Example of SystemStatus.txt

**[RobotList]**

Robot0=Disc Publisher XRP

**[Disc Publisher XRP]**

SysErrorString=No Errors

SysErrorNumber=0

SystemDrives=2

RobotType=5  
NumJobs=0  
DiscsInLeftBin=6  
DiscsInRightBin=4  
StatusFile=Disc Publisher XRP.txt  
KioskModeEnabled=0  
CartridgeType0=2  
CartridgeFill0=96  
CartridgeType1=1  
CartridgeFill1=95  
**[System]**  
SysErrorString=No Errors  
SysErrorNumber=0  
PtburnSWVer=1.0.3.0  
JobProcSWVer=3.0.9  
PTRobotVer=1.3.5.0  
PxsdkSWVer=4.2.62.500  
PxengSWVer=4.2.62.500  
PxDrvSWVer=1.02.16a  
PxMasSWVer=4.2.62.500  
PxWaveSWVer=4.2.62.500  
PxHelp20SWVer=3.00.67a  
ServerTime=06/05/2008 17:44:27  
ServerTickCount=1749  
DriveSpace=123849

Below is a description of each section and key available in SystemStatus.txt.

### 3.1.2 Robot List

This section in SystemStatus.txt contains a list of each robot active in the system.

e.g.

[RobotList]

Robot0=Disc Publisher XRP

Robot1=Disc Publisher SE

Each robot entry point to a "Robot Section within the file.

### 3.1.3 Robot Section

This section in SystemStatus.txt contains some basic status on a particular robot in the system. It also contains the file names where more status and can be found.

e.g.

[Disc Publisher XRP]

### 3.1.3.1 SystemDrives

Number of drives on the robot.

### 3.1.3.2 RobotType

Type of robot the possible values are defined below.

Disc Publisher = 0

Disc Publisher II = 1

Disc Publisher PRO = 2

Disc Publisher XR = 4

Disc Publisher XRP =5

Disc Publisher SE = 6

Disc Publisher Xi = 7

Disc Publisher 4100 = 8

### 3.1.3.3 NumJobs

Number of jobs currently running on the robot.

### 3.1.3.4 Status

String containing the status of the robot (For display purposes)

### 3.1.3.5 StatusFile

Filename of that status file for this particular robot.

This "Robot Status" file is located in the same folder as this file.

### 3.1.3.6 DiscsInRightBin

Number that contains the number of discs in the Right Bin. If this value is -1 then the number is unknown. This value is updated each time a disc is picked from a bin. The user can also send the "Check Bins" command to force an update of this value.

### 3.1.3.7 DiscsInLeftBin

Number that contains the number of discs in the Left Bin. If this value is -1 then the number is unknown. This value is updated each time a disc is picked from a bin. The user can also send the "Check Bins" command to force an update of this value.

### 3.1.3.8 CartridgeTypeX

Contains the type of CartridgeX. The X will be replaced with a value. CartridgeType0 represents the leftmost cartridge. If the robot contains a second cartridge, CartridgeType1 represents the right cartridge.

The potential values for this key are listed below.

CARTRIDGE\_NONE 0

CARTRIDGE\_COLOR 1

CARTRIDGE\_BLACK 2  
CARTRIDGE\_YELLOW 3  
CARTRIDGE\_CYAN 4  
CARTRIDGE\_MAGENTA 5

**NOTES:**

Values 3, 4, & 5 are for robots that have individual color ink cartridges (ie. Robots with 4 separate ink tanks). For better backward compatibility, robots with four ink cartridges will actually record five CartridgeType values like this:

CartridgeType0 = 2 ← Black  
CartridgeType1 = 1 ← Color  
CartridgeType2 = 3 ← Yellow  
CartridgeType3 = 4 ← Cyan  
CartridgeType4 = 5 ← Magenta

**3.1.3.9 CartridgeFillX**

Number that contains the percent fill of the cartridge. The X will be replaced with a value. CartridgeFill0 represents the leftmost cartridge. If the robot contains a second cartridge, CartridgeFill1 represents the right cartridge fill percent.

This value is a percent with values from 0-100%

NOTE: For better backward compatibility, robots with four ink cartridges will actually record five CartridgeFill values like this example:

CartridgeFill0 = 91 ← Black  
CartridgeFill1 = 37 ← Color (this will be the lowest value of Yellow, Magenta, & Cyan)  
CartridgeFill2 = 59 ← Yellow  
CartridgeFill3 = 37 ← Cyan  
CartridgeFill4 = 100 ← Magenta

**3.1.3.10 KioskModeEnabled**

If kiosk mode is enabled on this robot.

1=enabled, 0=disabled

**3.1.4 System Section**

This section in *SystemStatus.txt* contains general information about the system.

ie.

[System]



#### **3.1.4.1 SysErrorString**

String corresponding to the SysErrorNumber. Please see [Appendix C](#) for details on system errors.

#### **3.1.4.2 SysErrorNumber**

Integer value that details what error has occurred on the system. Please see [Appendix C](#) for more details on system errors.

#### **3.1.4.3 ServerTickCount**

Current tick count obtained by the PTFurn application on the server (computer on which PTFurn is running). This can be used to determine if the PTFurn Service is still running and updating status.

1 tick count = 1 second

#### **3.1.4.4 ServerTime**

Current time obtained by the PTFurn application on the server (computer on which PTFurn is running).

Format is:

MM/DD/YYYY hh:mm:ss

where:

MM is month (1-12)

DD is day (1-31)

YYYY is year (e.g. 2008)

hh is based on a 24-hour clock so the range is from 0 to 23 (e.g. 18:00:00 is 6PM)

mm is minutes (0-59)

ss is seconds (0-59)

#### **3.1.4.5 DriveSpace**

Integer value that tells how much free hard drive space is available. This is the drive on which the Shared Folder resides. The value is given in MB (e.g. for 12GB of free space – DriveSpace=12000)

#### **3.1.4.6 PtfurnSWVer**

String that contains the version of PTFurn.exe.

#### **3.1.4.7 JobProcSWVer**

String that contains the version of JobProcessor.dll.

#### **3.1.4.8 PTRobotVer**

String that contains the version of PTRobot.dll.

### 3.1.4.9 PxengSWVer

String that contains the version of Px.dll.

### 3.1.4.10 PxsdkSWVer

String that contains the version of PXSDKPLS.dll.

### 3.1.4.11 PxDrvSWVer

String that contains the version of PxDrv.dll.

### 3.1.4.12 PxMasSWVer

String that contains the version of PxMas.dll.

### 3.1.4.13 PxWaveSWVer

String that contains the version of PxWave.dll.

### 3.1.4.14 PxHelp20SWVer

String that contains the version of PxHelp20.sys.

## 3.2 RobotName.txt Status Files

Besides the *SystemStatus.txt* file, PTDurn will also create a more detailed status file for each robot; the name of the file(s) will be **RobotName.txt** (e.g. Disc Publisher SE.txt).

The structure of this status file will be similar to an .INI file with three fixed Sections of [JobList], [CompletedJobs], and [System]. Also, each job will have its own Section (section name is the name of the Job Request File without an extension) with detailed information of the job.

### 3.2.1 Example of RobotName.txt (e.g. Disc Publisher XRP.txt)

#### [System]

```
SystemStatus=System OK
RoboFWVer=4.07 03/24/2008
SerialNum=2091200229
DateManf=3/27/2010
SystemDesc=Disc Publisher XRP
SystemDrives=2
GoodDiscsLaunch=1
BadDiscsLaunch=0
GoodDiscsLife=5
BadDiscsLife=0
SysErrorString=No Errors
SysErrorNumber=0
DriveDesc0=PLEXTOR DVDR PX-716A 1.09
DriveLetter0=H
DriveLocation0=Bottom Recorder
```

DriveDesc1=PLEXTOR DVDR PX-716A 1.09

DriveLetter1=F

DriveLocation1=Top Recorder

DriveState0=0

DrivePercent0=0

DriveJob0=None

DriveDisc0=None

DriveState1=0

DrivePercent1=0

DriveJob1=None

DriveDisc1=None

DiscsInLeftBin=6

DiscsInRightBin=4

CartridgeType0=2

CartridgeFill0=96

CartridgeType1=1

CartridgeFill1=95

**[Top Recorder]**

BadDiscs=3

GoodDiscs=4

ReadingErrors=0

ReadingErrorsMedia=0

RecordingErrors=0

RecordingErrorsMedia=0

VerifyingErrors=0

VerifyingErrorsMedia=0

DriveNotReady=0

**[Bottom Recorder]**

GoodDiscs=0

BadDiscs=0

ReadingErrors=0

ReadingErrorsMedia=0

RecordingErrors=0

RecordingErrorsMedia=0

VerifyingErrors=0

VerifyingErrorsMedia=0

DriveNotReady=0

**[JobList]**

Job0=DataJob\_Test1

**[DataJob\_Test1]**

JobsAhead=0

DiscsAhead=0

CurrentStatus>Loading Disc 0

CurrentStatusState=4

TimeStarted=06/06/2008 14:06:12

GoodDiscs=0

```
BadDiscs=0
DiscsRemaining=1
JobState=1
LoadDiscState0=0
LoadDiscDrive0=F
[CompletedJobs]
Job0=DataJob_ClosedDisc
[DataJob_ClosedDisc]
JobsAhead=0
DiscsAhead=0
CurrentStatus=The job is complete...
CurrentStatusState=10
TimeStarted=06/05/2008 17:14:49
GoodDiscs=1
BadDiscs=0
DiscsRemaining=0
JobState=2
TimeCompleted=06/05/2008 17:16:13
```

Below is a description of each section and key available in RobotName.txt.

### 3.2.2 JobList Section

The first fixed section in RobotName.txt is [JobList]

Within this section are keys that can provide the list of the current jobs being processed by PTBurn. The Key value of the current job is always Job0, the next job is Job1, etc.

So, an example might be:

```
[JobList]
Job0=MyJob1
Job1=YourJob1
Job2=MyJob2
```

The client can display the list of jobs by sequentially trying to get the string values for Keys: Job0, Job1, Job2,... The end of the list of jobs can be determined by the client when the string value for the next key is not found. *This section may not exist if no jobs are currently running.*

### 3.2.3 CompletedJobs Section

The second fixed section in RobotName.txt is [CompletedJobs]

Within this section are keys that can provide a client the list of the recently completed jobs (including successful jobs, jobs with errors, and jobs that have been aborted). The Key value of the current job is always Job0, the next job is Job1, etc.

So, an example might be:

```
[CompletedJobs]
Job0=MyJob3
```

Job1=YourJob4

Job2=MyJob5

The client can display the list of jobs by sequentially trying to get the string values for Keys: Job0, Job1, Job2,... The end of the list of jobs can be determined by the client when the string value for the next key is not found.

### 3.2.4 Job Details Section

More detailed information about each job in [JobList] and [CompletedJobs] can be accessed in the job's own Section.

For example, if a submitted job is called MyJob1.jrq, then the section in the status file would be given as [MyJob1].

The keys under each job's details section are:

#### 3.2.4.1 JobID

String that describes this job

#### 3.2.4.2 ClientID

String that describes the client of this job

#### 3.2.4.3 JobsAhead

Integer value that tells how many jobs are in the PTBurn queue before this job.

#### 3.2.4.4 DiscsAhead

This integer value tells how many discs remain to be completed (in other jobs) before this job can begin.

#### 3.2.4.5 DiscsRemaining

Integer value that tells how many discs are remaining to be completed in this job.

#### 3.2.4.6 GoodDiscs

Integer value that tells how many discs have been successfully completed in this job

#### 3.2.4.7 BadDiscs

Integer value that tells how many discs that have failed to be completed successfully in this job (rejects)

#### 3.2.4.8 CurrentStatus

String that describes what is currently going on with this job.

#### 3.2.4.9 TimeCreated

String that gives the time and date of when the Job Request File was created

Format is:

MM/DD/YYYY hh:mm:ss XX

where XX is either AM or PM.

#### **3.2.4.10 TimeStarted**

String that gives the time and date of when the job was started.

Format is:

MM/DD/YYYY hh:mm:ss XX

where XX is either AM or PM.

If not started yet the value is blank space

#### **3.2.4.11 TimeCompleted**

String that gives the time and date of when the job was completed.

Format is:

MM/DD/YYYY hh:mm:ss XX

where XX is either AM or PM.

If not completed yet the value is blank space

#### **3.2.4.12 JobState**

Integer values that give the state of the job:

0 = Job not started

1 = Job running

2 = Job completed successfully

3 = Job failed

#### **3.2.4.13 JobErrorNumber**

Integer value that details what job error has occurred. Please see [Appendix C](#) for details on job errors.

#### **3.2.4.14 JobErrorString**

String corresponding to the JobErrorNumber. Please see [Appendix C](#) for details on job errors.

#### **3.2.4.15 DiscErrorIndexX**

Integer value that details which disc in the job had the error. In other words, the disc number within the job that the software is attempting to record (e.g. the first disc of the job = 1, second disc of the job = 2, etc.).

X is a value from 0 to 9 that provides a sequential unique key.

So, the first disc error that is encountered in the job will be given the key value of DiscErrorIndex0, the second disc error (within the same job) will be given the key value of DiscErrorIndex1, and so on. The number of disc errors reported is limited to 10 so the values for this key can range from DiscErrorIndex0 up to DiscErrorIndex9.

A disc may have more than 1 error reported if the disc was retried. Example:

A job is set to record 3 copies of a disc:

Disc 1 is recorded successfully.

Disc 2 is recorded successfully.

Disc 3 fails to record two times before successfully burning on its third attempt.

In this example, the entries for this job will be:

DiscErrorIndex0 = 3

DiscErrorIndex1 = 3

#### **3.2.4.16 DiscErrorNumberX**

Integer value that details what disc error has occurred. Please see [Appendix C](#) for details on disc errors and the potential values.

X is a value from 0 to 9 that provides a sequential unique key.

So, the first disc error that is encountered in the job will be given the key value of DiscErrorNumber0, the second disc error (within the same job) will be given the key value of DiscErrorNumber1, and so on. The number of disc errors reported is limited to 10 so the values for this key can range from DiscErrorNumber0 up to DiscErrorNumber9.

#### **3.2.4.17 DiscErrorStringX**

String corresponding to the DiscErrorNumber. Please see [Appendix C](#) for details on disc errors and the possible strings.

X is a value from 0 to 9 that provides a sequential unique key.

So, the first disc error that is encountered in the job will be given the key value of DiscErrorString0, the second disc error (within the same job) will be given the key value of DiscErrorString1, and so on. The number of disc errors reported is limited to 10 so the values for this key can range from DiscErrorString0 up to DiscErrorString9.

#### **3.2.4.18 LoadDiscStateX**

If the LoadUnloadOverride key is set in the jrj file then PTBurn will report the current state of this disc with this key. X is a value from 0 to the number of drives in the robot (ie. 0 for the first drive, 1 for the second drive).

See [Appendix D](#) for an example of use. The possible values for this key are:

- 1 = Not In Use
- 0 = Disc Loading
- 1 = Disc Loaded (hold state)
- 2 = Disc Processing
- 3 = Disc Processed (hold state)
- 4 = Disc Unloading/Unloaded

#### 3.2.4.19 LoadDiscDriveX

If the LoadUnloadOverride key is set in the jrj file then PTCBurn will report the drive letter that is processing the disc. X is a value from 0 to the number of drives in the robot (ie. 0 for the first drive, 1 for the second drive).

See [Appendix D](#) for an example of use.

#### 3.2.4.20 CurrentStatusState

This is a numerical value to describe what is going on with this job. This value can be used instead of CurrentStatus (section 3.2.4.8) if you would rather display your own status string based on this value. The possible numerical values for this key are:

- 0 = Recording Disc
- 1 = Reading Disc
- 2 = Verifying Disc
- 3 = Printing Disc
- 4 = Loading Disc
- 5 = Unloading Disc
- 6 = Waiting for Printer
- 7 = Waiting for Recorder
- 8 = Finishing Disc
- 9 = Rejecting Disc
- 10 = Job Complete
- 11 = Job Paused
- 12 = Job Resumed
- 13 = Job Aborted
- 14 = Job Initializing
- 15 = Job Failed
- 16 = Pre-Mastering
- 17 = System OK
- 18 = The Job has been moved to another disc publisher

### 3.2.5 System Section

The final fixed section in *RobotName.txt* is [System]

Within this section are keys that can provide more detailed system information for the robot in question. Some of these keys are duplicated from SystemStatus.txt.



### **3.2.5.1 SystemDesc**

String that describes the system (e.g. Disc Publisher DVDR)

### **3.2.5.2 SystemStatus**

String that gives general system status of the robotics (e.g. Offline, System OK, Out of Discs, Ink Low, ...).

If the software is running but the printer is offline (user pressed the power button) then the SystemStatus will be "Printer Offline". For this error, and other system errors, the SysErrorNumber key can be queried. See more information for SysErrorNumber below.

### **3.2.5.3 SystemDrives**

Integer value that tells how many drives are managed by the robotics. (for the Disc Publisher this will be 1).

### **3.2.5.4 GoodDiscsLife**

Integer value that tells how many discs have been successfully completed in the life of this system (value is stored on the computer – not within the robotics).

### **3.2.5.5 BadDiscsLife**

Integer value that tells how many discs have been rejected because of failure during the life of this system (value is stored on the computer – not within the robotics).

### **3.2.5.6 GoodDiscsLaunch**

Integer value that tells how many discs have been successfully completed since the PTBurn application was launched.

### **3.2.5.7 BadDiscsLaunch**

Integer value that tells how many discs have been rejected because of failure since the PTBurn application was launched.

### **3.2.5.8 SysErrorString**

String corresponding to the SysErrorNumber. Please see [Appendix C](#) for details on system errors.

### **3.2.5.9 SysErrorNumber**

Integer value that details what error has occurred on the system. Please see [Appendix C](#) for more details on system errors.

### **3.2.5.10 RoboFWVer**

String that contains the version of the firmware on the duplicator.

### 3.2.5.11 DriveDescX

String that contains the description of robotically controlled drives on the system. The X represents an index if multiple robotically controlled drives are present. Below is an example of the drive description key :

DriveDesc0 = PIONEER DVD-RW DVR-105 1.30

### 3.2.5.12 DriveLetterX

String that contains the system drive letter of the robotically controlled drive(s). The X represents an index if multiple robotically controlled drives are present. Below is an example of the drive letter key :

DriveLetter0 = F

DriveLetter1 = G

### 3.2.5.13 DriveLocationX

String that contains the location of robotically controlled drives on the system. The X represents an index if multiple robotically controlled drives are present. Below is an example of the drive location key :

DriveLocation0 = Top Recorder

DriveLocation1 = Bottom Recorder

### 3.2.5.14 DriveStateX

Number that contains the state of the robotically controlled drive(s) on the system. The X represents an index if multiple robotically controlled drives are present. The available values are:

0 = Idle

1= Recording

2= Reading

3= Verifying

4= Disc Loaded

5= Verify Failed

6= Verify Complete

7= Record Failed

8= Record Complete

### 3.2.5.15 DrivePercentX

Number that contains the percent complete (0-100) of the drive state that the recorder is currently processing. The X represents an index if multiple robotically controlled drives are present.

### 3.2.5.16 DriveJobX

String that contains the name of the job that the recorder is currently assigned to. The X represents an index if multiple robotically controlled drives are present.

### 3.2.5.17 DriveDiscX

Number that contains the disc number in the job that the recorder is currently assigned to. The X represents an index if multiple robotically controlled drives are present.

### 3.2.5.18 DiscsInRightBin

Number that contains the number of discs in the Right Bin. If this value is -1 then the number is unknown. This value is updated each time a disc is picked from a bin. The user can also send the "Check Bins" command to force an update of this value.

### 3.2.5.19 DiscsInLeftBin

Number that contains the number of discs in the Left Bin. If this value is -1 then the number is unknown. This value is updated each time a disc is picked from a bin. The user can also send the "Check Bins" command to force an update of this value.

### 3.2.5.20 CartridgeTypeX

Contains the type of CartridgeX. The X will be replaced with a value. CartridgeType0 represents the leftmost cartridge. If the robot contains a second cartridge, CartridgeType1 represents the right cartridge.

The potential values for this key are listed below.

CARTRIDGE\_NONE 0  
CARTRIDGE\_COLOR 1  
CARTRIDGE\_BLACK 2  
CARTRIDGE\_YELLOW 3  
CARTRIDGE\_CYAN 4  
CARTRIDGE\_MAGENTA 5

#### NOTES:

Values 3, 4, & 5 are for robots that have individual color ink cartridges (ie. Robots with 4 separate ink tanks). For better backward compatibility, robots with four ink cartridges will actually record five CartridgeType values like this:

CartridgeType0 = 2 ← Black  
CartridgeType1 = 1 ← Color  
CartridgeType2 = 3 ← Yellow  
CartridgeType3 = 4 ← Cyan  
CartridgeType4 = 5 ← Magenta

### 3.2.5.21 CartridgeFillX

Number that contains the percent fill of the cartridge. The X will be replaced with a value. CartridgeFill0 represents the leftmost cartridge. If the robot contains a second cartridge, CartridgeFill1 represents the right cartridge fill percent.

This value is a percent with values from 0-100%

NOTE: For better backward compatibility, robots with four ink cartridges will actually record five CartridgeFill values like this example:

CartridgeFill0 = 91 ← Black  
 CartridgeFill1 = 37 ← Color (this will be the lowest value of Yellow, Magenta, & Cyan)  
 CartridgeFill2 = 59 ← Yellow  
 CartridgeFill3 = 37 ← Cyan  
 CartridgeFill4 = 100 ← Magenta

### 3.2.5.22 SerialNum

Serial number of the unit. This key is only valid on Disc Publisher se, Disc Publisher PRO and Disc Publisher XRP

### 3.2.5.23 DateManf

Date the unit was manufactured. This key is only valid on Disc Publisher SE, Disc Publisher PRO and Disc Publisher XRP

## 3.2.6 Drive Statistics Section

There is a drive statistics section for each robotic drive. The name of the section can be obtained from the DriveLocationX key (where X = 0 or 1).

e.g.

DriveLocation0 = Top Recorder

DriveLocation1 = Bottom Recorder

In this example, the RobotName.txt file would contain two sections for drive statistics:

[Top Recorder]

[Bottom Recorder]

The contents of each drive statistics section is described here:

#### 3.2.6.1 GoodDiscs

The number of good discs produced by this drive.

#### 3.2.6.2 BadDiscs

The number of bad discs produced by this drive.

### **3.2.6.3 ReadingErrors**

The number of bad discs (produced by this drive) that were a result of reading errors that weren't media related (ie. not Sense 03)

### **3.2.6.4 ReadingErrorsMedia**

The number of bad discs (produced by this drive) that were a result of reading errors that were media related (ie. Sense 03)

### **3.2.6.5 RecordingErrors**

The number of bad discs (produced by this drive) that were a result of recording errors that were not media related (ie. not Sense 03)

### **3.2.6.6 RecordingErrorsMedia**

The number of bad discs (produced by this drive) that were a result of recording errors that were media related (ie. Sense 03)

### **3.2.6.7 VerifyingErrors**

The number of bad discs (produced by this drive) that were a result of verifying errors that were not media related (ie. not Sense 03).

### **3.2.6.8 VerifyingErrorsMedia**

The number of bad discs (produced by this drive) that were a result of verifying errors that were media related (ie. Sense 03)

### **3.2.6.9 DriveNotReady**

The number of bad discs (produced by this drive) that were a result of the drive not becoming ready.

## 4 Client Commands

PTBurn clients can send commands to the PTFurn server.

To do this, the client should create a new file in the Job Request Shared Folder with the filename extension of .PTM with the same filename as the JRQ file you are referencing.

### 4.1 Abort Command

The abort command will abort a particular job. The .ptm file must have the same filename as an existing Job Request File and the valid contents are shown below (note the filenames must match except for the filename extensions).

```
Message = ABORT
ClientID = XXX
```

Where XXX is the same ClientID that is specified in the corresponding Job Request File. You can abort any job by specifying “Administrator” for the ClientID.

For example, if the client submits a job with the Job Request Filename of:

```
MyJob1.JRQ
```

and then the client decides that this job should be aborted, the client must create a new file called:

```
MyJob1.PTM
```

Note that the ClientID must be specified and it MUST match the ClientID in the corresponding Job Request File – the abort request will be rejected if the ClientID does not match.

For the MyJob1 example (see above) the contents of MyJob1.PTM would be:

```
Message = ABORT
ClientID = Kevin
```

### 4.2 Shutdown After Job Command

The “Shutdown After Job” command will shutdown the Server software after any in-process jobs are completed (have a .inp extension). This command will prevent any new jobs from being put into an “in process” state. The unit will finish all actions on the “in process” jobs and then shutdown. The .ptm file can have any name and should be formatted as follows:

```
Message=SHUTDOWN_AFTERJOB
ClientID=Administrator
```

The ClientID should ALWAYS be fixed to “Administrator”.

### 4.3 Shutdown Immediate Command

The “Shutdown Immediate” command will shutdown the Server software immediately after the message is processed. Messages are processed approximately every 10 seconds. It is only advised to use this message in an emergency. Upon using this command the system will be left

in an unknown state. There could be discs left on the picker, in the recorder, or in the printer. The .ptm file can have any name and should be formatted as follows:

```
Message=SHUTDOWN_IMMEDIATE
ClientID=Administrator
```

The ClientID should ALWAYS be fixed to "Administrator".

#### **4.4 Align Printer Command**

**\*\* This command is only valid on the Disc Publisher PRO, XRP, SE \*\***

The Align Printer Command will align the printer when the system error is set to 22 or there are no jobs on the system. If the alignment fails, the system error will be set to 31 otherwise the system error will be set to 0. See [Appendix C](#) for system error codes.

```
Message= ALIGN_PRINTER
ClientID=Administrator
RobotName = zzzz
```

The ClientID should ALWAYS be fixed to "Administrator".

*RobotName* zzzz specifies which printer/robot to align (e.g. Disc Publisher SE). If *RobotName* is not specified it will align the first robot found.

#### **4.5 Ignore Ink Low Command**

**\*\* This command is only valid on the Disc Publisher PRO, XRP, SE \*\***

The Ignore Ink Low command will ignore an ink low condition when the system error is set to 5, 6, or 7.

```
Message= IGNORE_INKLOW
ClientID=Administrator
```

The ClientID should ALWAYS be fixed to "Administrator".

#### **4.6 Process Disc Command**

The Process Disc Command will cause the unit to leave the wait state prior to recording the disc and continue processing the disc. This command is only valid when LoadUnloadOverride was set to YES in the .JRQ Job Request File.

```
Message= PROCESS_DISC
DiscID=X
```

The DiscID should contain the id of the disc that is being processed (ie. the X value from LoadDiscStateX = in the Status file). See [Appendix D](#) for an example of how to use this message with the load/unload override functionality.

### **4.7 Reject Disc Command**

The Reject Disc Command will cause the unit to leave one of the two wait states (prior to recording or once recording is finished) and reject the disc. This command is only valid when LoadUnloadOverride was set to YES in the jobs jrj file.

```
Message= PROCESS_DISC  
DiscID=X
```

The DiscID should contain the id of the disc that is being processed (ie. the X value from LoadDiscStateX = in the Status file). See [Appendix D](#) for an example of how to use this message with the load/unload override functionality.

### **4.8 Unload Command**

The Unload Disc Command will cause the unit to leave one of the two wait states (prior to recording or once recording is finished) and unload the disc. This command is only valid when LoadUnloadOverride was set to YES in the jobs jrj file.

```
Message= PROCESS_DISC  
DiscID=X
```

The DiscID should contain the id of the disc that is being processed(ie. the X value from LoadDiscStateX = in the Status file). See [Appendix D](#) for an example of how to use this message with the load/unload override functionality.

### **4.9 Check Bins Command**

The “Check Bins” command will cause the Disc Publisher to check each bin for the number of discs remaining. It will then update the DiscsInLeftBin and DiscsInRightBin values of the status file. The .ptm file can have any name and should be formatted as follows:

```
Message= CHECK_DISCSINBIN  
ClientID=Administrator  
RobotName = zzzz
```

The ClientID should ALWAYS be fixed to “Administrator”. This command will only work if there are no jobs in the system.

*RobotName* zzzz specifies which printer/robot to check bins (e.g. Disc Publisher SE). If RobotName is not specified it will check bins on the first robot found.

### **4.10 Create Report Command**

This command causes a tech support report to be created and place in the PTBurnJobs\Reports folder. This report can be sent to Primera’s technical support department if an issues arises.

```
Message= CREATE_REPORT
```



### **4.11 Generate Preview**

This command generates an image preview of a Surething File. You must specify the path of the surething file, the path you want the image file to be placed, and the DPI of the image.

Message=GENERATE\_PREVIEW

PathFrom = Path of the source SureThing .std file (Required)

PathTo = Path to the destination image file (Required). .JPG, .BMP, or .PNG image files allowed.

DPI = 50 - 600 (Default Value = 72)

### **4.12 Run Utility Command**

The run utility command is used to run any .exe file on the server. You are required to provide the command line of the exe you want to run.

Message=RUN\_UTILITY

PathFrom = Path that the utility to run is at (Required)

### **4.13 USB Connect**

This command is used to simulate a new USB Disc publisher being connected. This is mainly used for testing and demonstration. The PTBurn service will automatically detect this when running.

Message = USB\_CONNECT

### **4.14 USB Disconnect**

This command is used to simulate a robot being disconnected. This is mainly used for testing and demonstration. The PTBurn service will automatically detect this when running.

Message=USB\_DISCONNECT

RobotName = Robot that was disconnected (Required)

### **4.15 Set/Clear Kiosk Mode**

**\*\* This command is only valid on the Disc Publisher II, XR, PRO, XRP, Xi, 4100 \*\***

This command is used to put or remove the robot from kiosk mode. Kiosk mode uses both bins as inputs and outputs all discs down the front slide. The current Kiosk mode state of all robots is stored in the "KioskModeEnabled" key, which resides in the Robot section of the SystemStatus file. (See Section 2.1.36 to see the behavior of BinID which tells the robot which bin to pick from and where to output the disc).

This message has the following structure:

Message= CHANGE\_KIOSK\_MODE

```
Value=X  
ClientID=Administrator  
RobotName=zzzz
```

where the value of X will either Set or Clear Kiosk Mode.

1 = Set Kiosk Mode

0 = Clear Kiosk Mode

*ClientID* should be set to Administrator

*RobotName* zzzz specifies which printer/robot to set/clear (e.g. Disc Publisher SE). If

RobotName is not specified it will set/clear kiosk mode for the first robot found.

## 5 Coding Best Practices

### 5.1 Programming Tips

#### 5.1.1 Reading the Status File

The status file should NOT be opened in a locked state. To ensure this you should either open the file with FileShare set to FileShare.ReadWrite or use GetPrivateProfileString (and/or GetPrivateProfileInt).

C# example:

```
FileStream fs = File.Open("C:\PTBurnJobs\Status\Disc Publisher Pro Xi.txt",  
    FileMode.Open, FileAccess.Read, FileShare.ReadWrite);
```

#### 5.1.2 Determining if the service is Running.

To determine if the service is running you can use the ServiceController class in .NET. Otherwise if you read the status file and the ServerTime and ServerTickCount are NOT incrementing then you know the service is NOT running.

#### 5.1.3 Starting the service

We advise that when your application starts it should ensure the service is running and if not your application should start the service. This can be done easily with .NET using the ServiceController class.

### 5.2 Merge Printing

More information regarding merge printing can be found in Section 2.1.40.

This is another example of creating a SureThing .STD document template for replaceable text & photos.

Below are the steps used to create the file MySureThingFile.STD and its associated Merge text file called MyMerge.CSV (these files are located in the \Client\SureThingMergeSample\ folder):

1. Create a comma-delimited text file with a title/name for each replaceable item. You can use any word processing application or Excel. Be sure to save the text file in a text format (not in a proprietary word processing format). Save with Unicode encoding if you have 2-byte characters or you can use ANSI encoding if there are not any 2-byte characters.  
In this example, the names/titles of the replaceable items are: MyText, MyPic1, MyPic2, MyPic3. Then, add the actual text and photo names you want to be inserted into the .STD document at print time. In this example, I wanted it to say "Joe's Photos", and I wanted three images to be inserted: Pic1.jpg, Pic2.jpg, and Pic3.jpg. Save this file. (Note that you can modify the Merge file just before print time to allow you to replace the text & photos with anything you want, or you can use the MergeField key in your JRQ to let PTBurn replace the text). The file extension does not have to be .csv; it could be something else like .txt.
2. Open Surething and start a new design. You can add any static information at this time -- for example, text, pictures, and backgrounds that do not change from disc to disc.
3. Click on Insert/Set Merge File. Browse and select the file you created in step 1.
4. Set up the print date to be automatically set at print time:  
Click on Insert/Insert Merge Field. Then click the Special Fields tab. Then choose Print Date from the Field Type.
5. Set up the replaceable text field:  
Click on Insert/Insert Merge Field. You should see the titles created in step 1. Choose any of the replaceable text values -- MyText in this example (we are only creating one replaceable text item in this example).
6. Set up the replaceable Photos:  
Click on Tools/Object Tools/Picture Tool. Marquis a rectangle (you can change size/position later) and then check the Merge: checkbox.  
You will see a list of Field Names with the titles that you created in step 1. Choose the image you want: MyPic1 in this example.  
Repeat this for MyPic2 and MyPic3.
7. Click on the Preview... button in the upper-right corner of SureThing. You should see the correct text and photos.
8. Save the .STD file. This can be a "template" for you to use. Unique data can be printed using step 9.
9. Now each time you want create a unique disc you should overwrite the text value for MyText --- so, you would overwrite "Joe's Photos" with "Amy's Photos" for example. Also, you would overwrite the three .JPG files with new .JPG files -- they could still have the same names if you want: Pic1.jpg, Pic2.jpg, and Pic3.jpg. In this case, you would not need to modify these values in the Merge file. However, if you did want to use new photos with alternate names (PicX.jpg, PicY.jpg, and PicZ.jpg for example) then you need to modify the

Merge file to call out these new filenames. Again, you can have PTBurn do the work for you by specifying MergeField keys in the .JRQ as described in Section 2.1.40.

**NOTE:**

One issue with SureThing is that it will always try to use the Merge file from its original location -- if you copy the .STD file and the Merge file to a new folder, then the .STD file will still try to use the Merge file from the original location -- if it can find the original Merge file then it will use it instead of the new one. If you don't want this to happen then you must delete or move the original Merge file (or copying files to a server machine without the original path works also). When specifying a merge picture within a Merge file, it is a good idea to NOT put in the path information -- just put the .STD file, Merge file, and the images in the same folder.

## 6 Appendix A: PTSetup.ini

The file PTSETUP.INI is a “loose” file that can be modified before deployment. Below is a copy of the default PTSETUP.INI file. See the description below for each key. It is advised that you modify this file (in the PTBurn directory on the installation disc) before you deploy your solution; the installer will then install your modified version which PTBurn server can use. If you do wish to modify the PTSETUP.INI file after installation through your own application then you can find the location of the PTSETUP.INI directory by querying the Registry key called: DataPath which is located in the Registry at:

HKEY\_LOCAL\_MACHINE\SOFTWARE\Primera\PTBurn

Note that that PTBurn will parse PTSETUP.INI when launched each time, so your program should modify PTSETUP.INI before launching PTBurn.exe. To find the directory where the PTBurn executable is located you can query the Registry key called: AppPath which is located in the same location in the Registry.

```
; PTSETUP.INI
;
; Setup file for the Primera Text-Based Shared-Folder Burning application PTBurn.
;
```

```
; NOTE: This is the default PTSETUP.INI file
; To change the defaults, you should edit this file before installing PTBurn.
; After installation, this file will reside in the PTBurnJobs\Settings folder
```

### [PTBurn\_Settings]

```
; For debugging purposes. Should be kept at FALSE unless advised to change.
RecordEngineLog=FALSE
```

```
; How long before a lower priority job will be processed over a higher priority job (in minutes).
; For example, if a job with a priority of 9 (lowest) has been waiting for over 120 minutes it will
; be processed
; before a job with a priority of 0 (highest) that has been waiting for less than 120 minutes. This
; key is only
; relevant for jobs request files that haven't been changed to "IN PROCESS" (.inp).
LowPriority_Time = 120
```

```
; How long the status information stays around (in status file) before being deleted (in minutes)
; This is also specifies how long the job request file (now .don or .err) will remain before being
; deleted.
Status_Time = 60
```

; What is the level of logging that PTDurn should track:

; 0 = No logging

; 1 = errors only

; 2 = warnings & errors

; 3 = some details

; 4 or 5 = more and more details...

LogLevel=4

; Number of rejects in a row before PTDurn will abort the job

Rejects\_InARow=3

; Should PTDurn attempt to complete all the copies requested

; or should it only process the number of discs requested

; TRUE will finish all copies requested (regardless of the number of rejects)

; FALSE will only deal with the number of disc requested (if any rejects occur then the number of good discs will be less than the number of copies requested)

CompleteAllCopies=TRUE

; Specifies what to do with "in process" (.inp) jobs on startup. This key has no effect on jobs that were completed, aborted or errored out (.DON or .ABT or .ERR) .

; Kill will automatically kill the INP job/s without prompting the user

; Retry will automatically retry the job/s without prompting the user

; KillAllIncludingJRQ will automatically kill all jobs INCLUDING JRQ and QRJ

JobActionOnStartup=Retry

; Specifies if the admin (set up through the "Setup Admin Notifications" button in the

; Settings dialog on the server) is to receive net messages when system errors occur.

; The AdminComputer key specifies the admin, this MUST be a computer name on the network.

; By default this notification is disabled; To enable net messages set to: NotifyAdmin=Message

NotifyAdmin=Disabled

AdminComputer=COMPANYX-NAME1

; Added to disable the power button, default is TRUE

DisablePowerButton=TRUE

; Used only when you want to specify a bin for each type of media (DVD and CD). The Possible values are

; 0 = Dont Care (Default)

```
;      1 = DVDs in left bin, CDs in right bin
;      2 = CDs in left bin, DVDs in right bin
; NOTE: As of PTBurn 3.0, this method of bin selection still works, but it is not the preferred
method.
;      The preferred method of specifying a particular bin is done by setting BinID=x in the
.JRQ file.
BinConfiguration=0
```

```
;We found that on some Bravo I units (Firewire interface with LG drives) the drive will fail to
open, but
;will still respond like it is open causing a mispick error. We have found that by adjusting the
time between
;when we stop the drive from spinning and when we open the drive we can prevent this from
happening.
;The default value is 2 seconds, we have found on units that exhibit this problem a 10sec delay
will usually fix
;the problem.
DriveDelay=2
```

```
;This string will set the temp path that the recording engine uses to store temporary data
necessary to
;burn the disc. The folder must exist or the job will fail with an Internal Recording Engine error.
; If not specified, it will use the default system temp folder.
;TempPath=C:\
```

```
;This key will specify if the disc publisher will print before recording. This is necessary for
business card
;discs. The default value is False.
PrintFirst=False
```

```
;This key determines how much time PTBurn will wait between getting a coveropen error and
reporting it. This is set to
;30 by default because we want to prevent getting an error when the user just opens and shuts
the cover.
CoverOpenWait=30
```

```
; TempSize sets the size (in KB) of the files that are copied in the temporary.
; For example, 4 means that all the files under 4096 bytes in size will be
; copied into a swap file generated in the temp path. This file will be destroyed after burning.
; A value of 0 means that no swap file is generated. The caching maximum value is 256.
TempSize=16
```

; SingleRobotOnly specifies whether or not to allow multiple robots.  
; When set to FALSE it will allow multiple robots; this also means that there will at least  
; two status files created (one master status file SystemStatus.txt and one status file for each  
; robot connected). This means an application developed for PTCBurn SDK 2.x will need to  
change slightly.  
; Set to TRUE to only allow a single robot - this will also create the same PTStatus.txt file as  
generated  
; in PTCBurn SDK 2.x  
SingleRobotOnly=FALSE

; Setting AutoSwitchRobot to TRUE tells the SDK to switch jobs from one robot to another robot  
; if one robot goes into an error condition, and there are multiple robots attached.  
AutoSwitchRobot=FALSE

;;;;;;;;;; SHARED FOLDER ;;;;;;;;;;;  
; This is the local path of where the Shared Folder "PTBurnJobs" will be created.  
; The shared folder will be created on first startup of the service.  
LocalPath=c:\PTBurnJobs\

; This is the name of the Shared Folder.  
; If not specified, the default is "PTBurnJobs"  
SharedFolderName=PTBurnJobs

; This is the remark/comment that is given to the Shared Folder.  
; If not specified, the default is "PTBurnJobs Shared Folder"  
SharedFolderRemark=PTBurnJobs Shared Folder

;;;;;;;;;;

; This is how often the Status file(s) are updated  
; Value is given in milliseconds. Minimum value is 1000  
StatusFrequency=2000

; DrivesToUse allows you to disable a particular drive in a two drive system. This key is ignored  
in

; a one drive system. The possible values are  
;  
;                                   0 - all drives are used (Default mode)  
;  
;                                   1 - Only the top drive is used  
;  
;                                   2 - Only the bottom drive is used  
;  
;

;       !! If you specify 1 or 2 for this key then the DriveID in the jrj is ignored !!  
DrivesToUse=0



```
; KioskMode allows you to set kiosk mode system wide (on all 2 bin robots)
;                               0 - kiosk mode set by ptm message (Default mode)
;                               1 - Kiosk mode always on
;                               2 - Kiosk mode always off
;
; !! If you specify 1 or 2 for this key then the Kiosk mode .ptm is ignored !!
KioskMode=0
```

```
; LightState allows you to set the interior light on robots with that feature.
;                               0 - PTBurn controls light state (Default mode)
;                               1 - Light is always on
;                               2 - Light is always off
;
; !! If you specify 1 or 2 for this key then the Kiosk mode .ptm is ignored !!
LightState=0
```

```
; end of PTSETUP.INI
```

---

## 7 Appendix B: Deployment Details

Primera includes an installer that can be used to install the PTFurn server and the PxEngine. Simply launch the file **SETUP.EXE** in the PTFurn directory and complete the installation. Note that there are two separate setup folders – one for Unicode (2-byte fonts) and one MBCS (1-byte fonts) – choose the one appropriate for your application. Each vendor can perform some customization by editing the loose file PTSETUP.INI before deployment. See the PTSETUP.INI file in the PTFurn directory for customization choices in the file.

NOTE: If you require PTFurnService to install quietly you can run the PTFurnServiceInstall.msi file with the following command line:

***msiexec /i PTFurnServiceInstall.msi /quiet***

After installation, you should have the user reboot the computer. If you want to force a reboot use 'REBOOT=Force' in the command line like this:

***msiexec /i PTFurnServiceInstall.msi REBOOT=Force***

## 8 Appendix C: Job, System and Burn Engine Error Codes

### System Errors

System errors are defined in the table below. These are errors that occur that require administrator attention. PTCBurn provides these error numbers and messages in the the Status file (See section IV of this document) with key values of SysErrorNumber and SysErrorString. You may choose to use/display the error string provided, or develop your own, for each respective error number.

---

System Error	
Number	String
0	No Errors
1	Administrator attention: Printer tray movement error.
2	Administrator attention: Invalid cartridge.
3	Administrator attention: Input bin is empty.
4	Administrator attention: Internal printer communication error.
5	Administrator attention: Color ink cartridge is low.
6	Administrator attention: Black ink cartridge is low.
7	Administrator attention: Both ink cartridges are low.
8	Administrator attention: Failure picking disc.
9	Administrator attention: Mechanical error moving robotic arm.
10	Administrator attention: Mechanical error moving printer cartridge.
11	Administrator attention: Server lacks local admin rights on the machine.
12	Administrator attention: Internal software error (PTBurn.exe).
13	Administrator attention: No robotically controlled drives are present.
14	Administrator attention: The unit is offline.
15	Administrator attention: The cover on the unit has been open for an extended period of time.
16	Administrator attention: Failure picking disc.
17	Administrator attention: Multiple discs were picked.
18	Administrator attention: Disc was dropped while moving into the Recorder.
19	Administrator attention: Disc was dropped while moving into the Printer.
20	Administrator attention: Disc was dropped while moving into the Left Bin.
21	Administrator attention: Disc was dropped while moving to Reject (Front Slide).
22	Administrator attention: The printer needs alignment.
23	Administrator attention: Invalid color cartridge.
24	Administrator attention: Invalid black cartridge.
25	Administrator attention: Both cartridges are invalid.
26	Administrator attention: No cartridges are installed.
27	Administrator attention: The Black Cartridge is in the Color Cartridge holder.
28	Administrator attention: The Color Cartridge is in the Black Cartridge holder.
29	Administrator attention: The Color and Black cartridges are swapped.
30	Administrator attention: This printer is not compatible with a pigment based cartridge.
31	Administrator attention: The alignment print failed.
32	Administrator attention: The disc was dropped moving into the printer.
33	Administrator attention: Multiple discs were placed into the right bin.
34	Administrator attention: Multiple discs were placed into the left bin.
35	Administrator attention: The color cartridge is empty.
36	Administrator attention: The black cartridge is empty.
37	Administrator attention: Both ink cartridges are empty.
38	Administrator attention: Timed out waiting for printer.
39	Administrator attention: The disc was dropped.
40	Administrator attention: The disc is missing from the printer tray.
41	Administrator attention: The disc was dropped while moving into the Right Bin.
42	Administrator attention: Timed out waiting for robot response.

The following listing contains suggested alternative error strings that may provide the user with more information. This list contains only those errors that have a possible recover action apart from restarting the unit and server software. In the case of a user trying to clear the error (by the method suggested in the strings below), your software can find out if the retry was successful by seeing if the SysErrorNumber changes back to 0.

**System Error 1**

“Tray movement error. Press the Cartridge button (left button) on the unit to try again.”

**System Error 2**

“There was a problem finding the ink cartridges. Open the cover and press the Cartridge button (left button). Make sure the color cartridge is installed on the left and the black is on the right. Then close the lid.”

**System Error 3**

“The input bin is empty. Add more discs and push the Cartridge button (left button) on the unit.”

**System Error 4**

“There was an internal printer communications error. Press the Cartridge button (left button) on the unit to try again.”

**System Error 5**

“WARNING: The COLOR cartridge (left side) is LOW on ink. To replace the cartridge, open the cover on the unit and press the Cartridge button (left button), install the new cartridge, and close the cover. To ignore the warning press the Cartridge button (left button).”

**System Error 6**

“WARNING: The BLACK cartridge (right side) is LOW on ink. To replace the cartridge, open the cover on the unit and press the Cartridge button (left button), install the new cartridge, and close the cover. To ignore the warning press the Cartridge button (left button).”

**System Error 7**

“WARNING: BOTH cartridges are LOW on ink. To replace the cartridge, open the cover on the unit and press the Cartridge button (left button)., install the new cartridges, and close the cover. To ignore the warning press the Cartridge button (left button).”

**System Error 8**

“The disc was not picked. Push the Cartridge button (left button) on the unit to try again.”

**System Error 9**

“There was an arm movement error. Press the Cartridge button (left button) on the unit to try again.”

Note: This error is specific to Disc Publisher I, and not Disc Publisher II.

**System Error 10**

“Arm picker error (unable to home the picker or unable to hook up the picker). Press the Cartridge button (left button) on the unit to try again.”

**System Error 11**

“The current user does not have local administrator rights. Please login with local administrator rights and re-start the software.”

**System Error 12**

“There was an internal software error. Please re-start the software.”

**System Error 13**

“No external recorder drives were found. Please unplug all cables from the Disc Publisher (including power). Then plug all cables back into the Disc Publisher, re-boot the computer, and then re-start the software.”

Note: This error will occur only at the start of the software.

**System Error 14**

“The Disc Publisher is offline. Please ensure the unit is connected and powered on, and then shutdown and restart the software.”

Note: In order to consistently recover from the unit going offline the cause must be determined and corrected, and the software restarted.

**System Error 15**

“The Disc Publisher cover is open. Please close the cover.”

**System Error 16**

“The disc was not picked from the printer. Please manually remove the disc from the printer. Then close the cover and press the power button (right button) several times to reset the printer.”

**System Error 17**

“Multiple discs were picked up and moved. Please manually remove any extra discs that were moved, keeping a single disc in place. Then close the cover and press the left button.”

**System Error 18**

“The disc was dropped while moving into the Recorder. Please manually place

the disc into the Recorder tray. Then close the cover and press the left button.”

**System Error 19**

“The disc was dropped while moving into the Printer. Please manually place the disc into the Printer tray. Then close the cover and press the left button.”

**System Error 20**

“The disc was dropped while moving into the Left Bin. Please manually place the disc into the Left Bin. Then close the cover and press the left button.”

**System Error 21**

“The disc was dropped while moving to the Reject area (Front Slide). Please remove the dropped disc from the printer. Then close the cover and press the left button.”

**System Error 22**

**\*\* Not used in Disc Publisher II \*\***

“The printer needs to be aligned. Would you like to align the printer now? Click “Yes” to align the printer, or “No” to quit the application.

**System Error 23**

**\*\* Not used in Disc Publisher II \*\***

“The color cartridge is invalid. To change the cartridge open the cover, press the “Cartridge” button, change the cartridge, close the cover and press “Ok” to continue.

**System Error 24**

**\*\* Not used in Disc Publisher SE \*\***

“The black cartridge is invalid. To change the cartridge open the cover, press the “Cartridge” button, change the cartridge, close the cover and press “Ok” to continue.

**System Error 25**

**\*\* Not used in Disc Publisher SE \*\***

“Both cartridges are invalid. To change the cartridges open the cover, press the “Cartridge” button, change the cartridge, close the cover and press “Ok” to continue.

**System Error 26**

**\*\* Not used in Disc Publisher II \*\***

“No cartridges are installed. To install cartridges open the cover, press the “Cartridge” button, change the cartridge, close the cover and press “Ok” to continue.

**System Error 27**

**\*\* Only used on the DiscPublisher PRO/XRP \*\***

“The black cartridge is in the color cartridge holder. To change the cartridges open the cover, press the “Cartridge” button, change the cartridge, close the cover and press “Ok” to continue.

**System Error 28**

**\*\* Only used on the DiscPublisher PRO/XRP \*\***

“The color cartridge is in the black cartridge holder. To change the cartridges open the cover, press the “Cartridge” button, change the cartridge, close the cover and press “Ok” to continue.

**System Error 29**

**\*\* Only used on the DiscPublisher PRO/XRP \*\***

“The cartridges are swapped (color in black and black in color holders). To change the cartridges open the cover, press the “Cartridge” button, change the cartridge, close the cover, and press “Ok” to continue.

**System Error 30**

**\*\* Only used on the DiscPublisher PRO/XRP \*\***

“This printer is not compatible with a pigment based cartridge. To change the cartridges open the cover, press the “Cartridge” button, change the cartridge, close the cover, and press “Ok” to continue.

**System Error 31**

**\*\* Not used in Disc Publisher II \*\***

“The alignment print failed. Would you like to retry the alignment now? Click “Yes” to align the printer, or “No” to quit the application.

**System Error 32**

**\*\* Not used in the SDK \*\***

**System Error 33**

**\*\* Not used in the SDK \*\***

**System Error 34**

**\*\* Not used in the SDK \*\***

**System Error 35**

“WARNING: The color cartridge is Empty. To replace the cartridge, open the cover on the unit and press the left button. Then install the new cartridge and close the cover. To ignore the warning, press the left button.”

**System Error 36**

“WARNING: The black cartridge is Empty. To replace the cartridge, open the cover on the unit and press the left button. Then install the new cartridge and close the cover. To ignore the warning, press the left button.”

**System Error 37**



“WARNING: Both cartridges are Empty. To replace the cartridge, open the cover on the unit and press the left button. Then install the new cartridge and close the cover. To ignore the warning, press the left button.”

**System Error 38**

“The system timed out waiting for the printer to finish. The disc may not have been printed.”

**System Error 39**

“The disc was dropped.”

**System Error 40**

“The disc to be printed is missing from the printer tray. Please place the disc back into the printer tray and close the tray. NOTE: You can open/close the printer tray by pressing the left button with the cover closed.”

**System Error 41**

**\*\* Not used in the SDK \*\***

**System Error 42**

“The robot is not responding.”

**Job/Disc Errors**

Job/Disc errors are defined in the table below. These are errors that either cause a job to fail, or cause a disc within a job to fail. A job error means the job did not complete successfully, whereas disc errors within a job will not cause the job to fail unless too many rejected discs in a row are encountered. PTCBurn provides these error numbers and messages in each job's "Job Details" section of the Status file (See section IV of this document) with key values of JobErrorNumber and JobErrorString for job errors, and DiscErrorNumberX and DiscErrorStringX for disc errors (where X is an integer from 0 to 9). You may choose to use/display the error string provided, or develop your own, for each respective error.

<b>Job/Disc Errors</b>		
<b>Number</b>	<b>String</b>	<b>Error Type</b>
0	No Errors	Job
1	Internal recording engine error.	Job
2	Internal Job Processor error.	Job
3	Internal robotics error.	Job
4	The file/s you are trying to burn are invalid.	Job
5	The number of files you are trying to burn exceeds the limit.	Job
6	There is no burn file defined for this job.	Job
7	The print file is invalid.	Job
8	The job request file is invalid.	Job
9	The status file is corrupted or invalid.	Job
10	The media (CD/DVD) is invalid.	Disc
11	The media (CD/DVD) has insufficient space.	Disc
12	The media (CD/DVD) is not blank.	Disc
13	The drive failed to open/close.	Job
14	The drive failed to become ready in a reasonable time.	Disc
15	The drive selected is not a robotically controlled drive.	Job
16	The job was aborted.	Job
17	The DVD structure was invalid.	Job
18	Error recording the disc, see the log file for further information.	Disc
19	Error verifying the disc, see the log file for further information.	Disc
20	The number of allowable rejects was exceeded.	Job
21	The session of a multisection disc is invalid.	Job
22	This client (You) are not authorized to use the server and there are no licenses available.	Job
23	Error processing client message.	Job
24	An unknown job type was attempted.	Job
25	A critical key or value in the job request was invalid	Job
26	The temporary folder has exceeded its limit.	Job
27	The CDText you attempted to use is invalid.	Job
28	The print application is not installed.	Job
29	The print file does not exist.	Job
30	Invalid cartridge/s for the print file specified.	Job
31	Insufficient space in the read disc location.	Disc
32	Error reading the disc, see the log file for further information.	Disc
33	The PVD data is invalid	Job
34	Invalid job type for specifying PVD Data	Job
35	Error creating disc image	Job
36	Unable to get status	Job
37	Feature not supported	Job
38	Invalid Job ID	Job
39	String too long	Job

**Job/Disc Error 1**

This error indicates a problem with the recording engine module. This error will cause the job to fail.

**Job/Disc Error 2**

This error indicates a problem with the JobProcessor.dll module. This error will cause the job to fail.

**Job/Disc Error 3**

This error indicates a problem with the PxRobo.dll module. This error will cause the job to fail.

**Job/Disc Error 4**

This error will occur when an invalid burn file/s is specified. An invalid file error can be caused by missing files, corrupt files, or files that are not in the correct format for the job requested. This error will cause the job to fail.

**Job/Disc Error 5**

The maximum files/folders that can be burned is 99. However, files that are contained within a folder that is specified do not count towards the maximum. This error will cause the job to fail.

**Job/Disc Error 6**

The error will occur when there is no data files/folder in a job. This error will cause the job to fail.

**Job/Disc Error 7**

This error will occur when an invalid print file is specified. An invalid print file could be a file that is not one of the following types: .std, .bmp, .jpg, or .prn. This can also occur with a .prn file that was not created for the printer being used. This error will cause the job to fail.

**Job/Disc Error 8**

This error will occur when an invalid Job Request File is specified. This could be the result of a missing or locked file. This error will cause the job to fail.

**Job/Disc Error 9**

This error will not occur in the SDK version of PTDisk.

**Job/Disc Error 10**

This error will occur when attempting to burn to invalid media. This could be the result of a flipped disc or an unreadable disc. This error will cause the disc to fail.

**Job/Disc Error 11**

This error will occur when attempting to burn more data than is available on the recordable media. This error will cause the disc to fail.

**Job/Disc Error 12**

This error will occur when attempting to burn data to a disc that is not blank and the key RejectIfNotBlank (in the jrj file) is set to YES. This error will cause the disc to fail.

**Job/Disc Error 13**

This error will occur when the drive fails to respond to open/close commands. This error will cause the job to fail.

**Job/Disc Error 14**

This error will occur when a drive fails to become ready after the disc has been inserted. This is usually the result of poor/unreadable media (making it similar to Job/Disc Error 10). This error will cause the disc to fail.

**Job/Disc Error 15**

This error will occur when the drive that is being used for a job is not the drive in the Disc Publisher. This error can occur when the Disc Publisher drive is not seen by the server, and there is another external recorder on the system. This error will cause the job to fail.

**Job/Disc Error 16**

This error will occur if the job is aborted by the user.

**Job/Disc Error 17**

This error will occur when the VIDEO\_TS or AUDIO\_TS structures do not respect the DVD-Video or DVD-Audio rules. This error will cause the job to fail.

**Job/Disc Error 18**

This error will occur when the engine reports an error recording the disc. This error will also be accompanied by Sector, Sense, ASC, and ASCQ values. This error will cause the disc to fail.

**Job/Disc Error 19**

This error will occur when the engine reports an error verifying the disc. This error will also be accompanied by Sector, Sense, ASC, and ASCQ values. This error will cause the disc to fail.

**Job/Disc Error 20**

This error will occur when the maximum allowable reject limit has been reached. This error will cause the job to fail.

**Job/Disc Error 21**

This error will occur when the session of a multisession disc is invalid. This error will cause the job to fail.

**Job/Disc Error 22**

This error will not occur in the SDK version of PTDisk.

**Job/Disc Error 23**

This error will occur if there is an error processing a client message. This error will cause the job to fail.

**Job/Disc Error 24**

This error will occur if an unknown job type is specified in the .jrj file. This error will cause the job to fail.

**Job/Disc Error 25**

This error will occur if a critical key in the jrj file is invalid. This error will cause the job to fail.

**Job/Disc Error 26**

This error will occur if the temporary storage area for the application runs out of space. This error will cause the job to fail.

**Job/Disc Error 27**

This error will occur if the CD Text is in an invalid format or exceeds the maximum CD Text length (6000 bytes). This error will cause the job to fail.

**Job/Disc Error 28**

This error will occur if the CD printing application is not installed and the job requires the printing application. This error will cause the job to fail.

**Job/Disc Error 29**

This error will occur if the print file specified in the jrj file does not exist. This error will cause the job to fail.

**Job/Disc Error 30**

This error will occur if an improper ink cartridge is installed for the print file you are attempting to print (e.g. only the Black cartridge is installed but you are attempting to print with Color).

**Job/Disc Error 31**

This error will occur if the location the you specify to read the data to in a read job does not have sufficient space for the data. This error will cause the disc to fail.

**Job/Disc Error 32**

This error will occur when the engine reports an error reading the disc. This error will also be accompanied by Sector, Sense, ASC, and ASCQ values. This error will cause the disc to fail.

**Job/Disc Error 33**

This error will occur if a PVD field is invalid. This is usually the result of a field being too large. This error will cause the job to fail.

**Job/Disc Error 34**

This error will occur if PVD fields are specified for any jobs except Data Disc Jobs. This error will cause the job to fail.

**Job/Disc Error 35**

This error will occur if the disc image cannot be created. This error will cause the job to fail.

**Job/Disc Error 36**

This error will occur if status cannot be obtained from the unit.

**Job/Disc Error 37**

This error will occur if a robot action is requested, but the robot does not support the action.

**Job/Disc Error 38**

This error will occur if an incorrect Job ID is specified. This error will cause the job to fail.

**Job/Disc Error 39**

This error will occur if a merge field string is too long (>260 characters). This error will cause the job to fail.

**Burn Engine Errors**

Errors from the burn engine are also recorded in the Log file. These errors are differentiated from Job and System errors by their surrounding syntax. For example “Error 33 returned from PrimoSDK\_RunningStatus”

PRIMOSDK_OK	0	The operation completed successfully.
PRIMOSDK_CMDSEQUENCE	1	The function was used in the incorrect sequence. Another PrimoSDK API is required before calling this function.
PRIMOSDK_NOASPI	2	The ASPI layer is not loading or is in error.
PRIMOSDK_INTERR	3	An internal error occurred.
PRIMOSDK_BADPARAM	4	The function was passed an invalid parameter.
PRIMOSDK_ALREADYEXIST	6	The function was passed a pointer to a directory element that already exists in the target data structure.
PRIMOSDK_NOTREADABLE	7	The function was passed a pointer to source file that either cannot be found or is not readable.
PRIMOSDK_NOSPACE	8	Completion of the operation would result in too many files for the system memory.
PRIMOSDK_INVALIDMEDIUM	9	The function was passed a pointer to a target unit containing media that is not blank.
PRIMOSDK_RUNNING	10	The operation whose status is being queried is currently running.
PRIMOSDK_BUR	11	The unit whose status is being queried went into buffer underrun.
PRIMOSDK_SCSIERROR	12	The unit whose status is being queried experienced a communication error.
PRIMOSDK_UNITERROR	13	The SCSI command sent by the function returned a check condition.
PRIMOSDK_NOTREADY	14	The function was passed a pointer to a unit that is not ready.
PRIMOSDK_INVALIDSOURCE	16	The function was passed a pointer to a disc or file that is not valid.
PRIMOSDK_INCOMPATIBLE	17	The function was passed a pointer to an image from a type of disc that is not compatible with the capabilities of the recorder.
PRIMOSDK_FILEERROR	18	The function was passed a pointer to a file that cannot be found.
PRIMOSDK_ITSADEMO	23	The operation requested by the function would exceed the limits allowed by the Demo version of PrimoSDK.
PRIMOSDK_USERABORT	24	The operation whose status is being queried was aborted because of a call with the PRIMOSDK_ABORT flag.
PRIMOSDK_BADHANDLE	25	The function was passed an invalid PrimoSDK handle.
PRIMOSDK_BADUNIT	26	The function was passed a pointer to a unit that does not exist.
PRIMOSDK_ERRORLOADING	27	An error occurred while reading the directory of the

		specified session/border.
PRIMOSDK_NOAINCONTROL	29	AIN control cannot be activated, typically because PrimoSDK is running under WinASPI instead of PxHelper.
PRIMOSDK_READERROR	30	The unit whose status is being queried reported a reading error.
PRIMOSDK_WRITEERROR	31	The unit whose status is being queried reported a writing error.
PRIMOSDK_TMPOVERFLOW	32	A temporary file went into overflow.
PRIMOSDK_DVDSTRUCTERROR	33	A data structure includes a VIDEO_TS or AUDIO_TS folder that is not compliant with DVD-Video or DVD-Audio rules.
PRIMOSDK_FILETOOLARGE	34	The function was passed a pointer to a file that is bigger than 9.99 GB for UDF or 4 GB for ISO.
PRIMOSDK_CACHEFULL	35	Not currently used.
PRIMOSDK_FEATURE_NOT_SUPPORTED	36	The device does not support the requested feature.
PRIMOSDK_FEATURE_DISABLED	37	Use of the requested feature is not included in the license under which the SDK was provided.
PRIMOSDK_CALLBACK_ERROR	38	Returned from the caller's callback function to terminate stream.
PRIMOSDK_PROTECTEDWMA	39	Returned if application does not have permission to burn specified WMA file.



## 9 Appendix D: Loading/Unloading Override

PTBurn now allows an external program to override disc load/unload functionality. This is useful if you want to do some external processing of the disc, either before recording or after recording. This also allows the ability to load discs into the drive so an external program can read information from the drive (without performing any recording). An example of one use of this functionality is detailed below.

*NOTE: If you do not need burning capabilities, then you may want to consider using the PTRobot DLL for moving discs around instead of using the PTBurn SDK.*

PTBurn SDK Load/Unload Override example:

The user wants to run their own “special” verification procedure on the disc. They want this procedure to run prior to unloading the disc to the printer and want to be able to control whether the disc passes or fails based on their verification.

- User submits a Recording/Printing job with LoadUnloadOverride = YES.
- Once the disc is loaded, PTBurn will set the following values in the Job specific status section:

```
[JobXYZ]
LoadDiscState0 = 1
LoadDiscDrive0 = F
```

- This notifies the client that the disc is loaded in drive F: and in a hold state until the client releases the hold state. The user could do some special procedure on the disc at this point if they desire.
- The user releases the hold state by sending the “Process Disc” message. For this example the message file must be called JobXYZ.ptm and the contents are shown below. The DiscID is the value of X from the LoadDiscStateX in the status file.

```
Message= PROCESS_DISC
DiscID=0
```

- Once PTBurn processes the “Process Disc” message the .ptm file is deleted and the Job specific status section is updated as follows:

```
[JobXYZ]
LoadDiscState0 = 2
LoadDiscDrive0 = F
```

- At this point PTBurn will process the recording portion of the job as it was specified in the .JRQ file. If no recording is specified in the JRQ file, the PTBurn will skip over the recording phase.

- Once completed, PTCBurn will enter a hold state with the job status section as follows:  
[JobXYZ]  
LoadDiscState0 = 3  
LoadDiscDrive0 = F
- At this point, the client can perform his “special” verification procedure on the disc.
- If the user’s special verification procedure fails, the client can send the “Reject Disc” message as detailed below. This will cause the hold state to end, and the disc will fail and be rejected appropriately. The contents of JobXYZ.ptm would be:

```
Message= REJECT_DISC  
DiscID=0
```

- However, if the user’s special verification passes, the client can send the “Unload Disc” message as detailed below. This will cause the hold state to end. The job will continue with printing if the JRQ file specifies that the disc be printed on. Otherwise, the disc will be unloaded to the output. Here are the contents of JobXYZ.ptm:

```
Message= UNLOAD_DISC  
DiscID=0
```

## 10 Revision History

### 3/11/11 – document version 3.23

- Added new CurrentStatusState to section 3.2.4.20

### 2/4/11 – document version 3.22

- Updated ink level information for 4-Cartridge robots (sections 3.1.3.8 & 3.1.3.9 & 3.2.5.20 & 3.2.5.21)
- Added new RobotType to section 3.1.3.2
- Added ISRC capability for Audio discs (section 2.1.5)
- Added CD-Text capability for Audio discs (Sections 2.1.43 to 2.1.48)
- Added missing documentation for CurrentStatusState in section 3.2.4.20
- Updated information for ReadDataFormat (2.1.20) and CreateSubFolders (2.1.21)
- Moved Section 2.3 and consolidated with 5.1. Updated section 5.2
- Updated Section 1 to notify the existence of both Unicode and MBCS versions of PTDisk

### 9/16/10 – document version 3.2.1

- Updated options for JobStartup

### 8/10/10 – document version 3.2.0

- Added new keys to PTSetup.in in 5.0
  - DrivesToUse
  - KioskMode
  - LightState
- Added best practices section.

### 2/17/10 – document version 3.1.0

- PathTo, PathFrom and DPI keys were incorrect in 4.11 and 4.12. Added Burn Engine Error Code Table in Appendix C.

### 11/12/09 – document version 3.0.9

- Updated Introduction to include Disc Publisher Pro Xi-Series
- Updated Introduction to include Windows 7

### 9/21/09 – document version 3.0.8

- Added Section 2.1.42 for PrintReject Key.

### 4/24/09 – document version 3.0.7

- Added Section 2.1.41 for key RobotName. The functionality already existed for specifying a particular robot, but the key had not been documented.

### 12/31/08 – document version 3.0.6

- Updated Section 2.1.33 to note the need for the DiscType key when doing pre-mastering of a DVD.

#### 11/7/08 – document version 3.0.5

- Updated Appendix C to define more system and job/disc errors

#### 6/20/08 – document version 3.0.4

- Updated Section 2.1.3 to note that the Importance key now allows values up to 400.
- Updated Section 2.1.41 to clarify how merging works with SureThing
- Updated Section 3.1.2.9 with CartridgeType values
- Updated Section 3.2.3.18 for LoadDiscState
- Added Section 3.2.4.21 DriveLetter to get the drive letter for the recorder.
- Updated several of the Status file sections to clarify.
- Added UDF250 & UDF260 options to DataImageType.

#### 1/28/08 – document version 3.0.3

- Added Section 3.1.3
- Added Set/Clear Kisok mode message (4.15) and corresponding status (3.1.2.11)
- Changed “Error” key to “SysErrorNumber” (3.1.2.5)
- Added Server Tick Count (3.1.3.1)
- Added StatusFrequency key to PTSetup.INI in case the developer wants to change how often the status is updated. (Appendix A)

#### 9/20/07 - document version 3.0.2

- Updated Appendix 6 to reflect the changes in installing the service versus the old PTBurn application.
- Removed the CDText Keys. They will not be supported in the first version of PTBurn 3.0 and will be added in a future release.

#### 9/13/07 - document version 3.0.1

- Updated section 2.1.5 to show the new feature of copying source files/folders while changing the destination file and/or folder names.
- Added section 2.1.47 MergeField to document how a user can specify Surething merge fields within the JRQ file.

#### 8/31/07 - document version 3.0.0

-