



EPC NT Printer Driver Documentation

The following document describes the NT parallel printer interface service for the EPC GSP-1086 family, 9802 THERMAL GRAPHIC RECORDER, Model 9205 THERMAL GRAPHIC RECORDER, ADS-640 models and related family of high-speed thermal printers. In addition to supporting NT applications with a kernel mode device driver, the included library also supports normal port IO under Windows 95. By linking a WIN32 application with the provided EPCLIB library a single executable can communicate with the EPC family of printers under either operating system. The library detects which operating system it is currently executing under and calls the NT service functions when running under NT. The library will call the port IO functions when running under Windows 95.

Installation

A simple InstallShield installation script installs the EPC NT Printer Driver service. This script will also install two sample programs and their related source code and Visual C++ project files.

Programming Overview

Included in this distribution is a high level library compatible with WIN32 console, MFC and WIN32 SDK applications. The library contains a set of functions that ease the communications with the NT printer service. Following is a description of the current functionality of this library and how best to use this tool.

In general, the sequence of programming functions to the EPC printer library would be pseudo coded as follows:

Win32 Static Library - API

Following each library function is described. All return codes as simple enumerated types (integer). A return value of eEPC_OK indicates success for all of the functions. See the error code table in appendix A for a list of possible error return values and their meanings.

```
int EpcOpenPrinter(const char * pSetupStr = EPC_1086_SETUP);
```

pPrinterSetupStr:

specifies the initialization string to send to place the printer in digital mode. The library defines two constants that should be used in almost all cases. When initializing a 9800 family printer use the constant EPC_9800_SETUP. When initializing any other EPC printer use the EPC_DEFAULT_SETUP constant.

Windows NT:

This function first tests to see if the printer service is available. If the service is available, the next test is to see if the service has actually been started. If the service is not running, this function will start the service. The library uses a single dynamically allocated buffer to communicate with the printer service. The size of this buffer is determined at "open-time" by the argument headSize. This will be the maximum

size of a raster that the library will pass to the printer service. Should a caller pass less than headSize bytes in a call to EpcSendRaster() this buffer will be padded with white space to complete the full raster. If no argument is supplied by the caller then the headSize argument is defaulted to 2048 pixels wide.

Windows 95:

The library will detect the operating system at printer initialization time. If Windows 95 is the current operating system, all subsequent IO will be routed to the standard printer IO port (0x378) using standard Win32 programmed IO calls.

```
int EpcGetPrinterStatusByte(ULONG * pStatus);
```

EpcGetPrinterStatusByte() returns the current ready state of the parallel port by reading the status byte at the printer port. A hexadecimal value of 0xDF indicates the printer is on-line and ready to accept data. See the EPC printer documentation for full details of the status byte.

```
int EpcSendRaster(T_EPC_PACKET * pPacket, DWORD dwPadToSize /*=
DEFAULT_HEAD_SIZE*/);
```

EpcSendRaster is a higher level function for sending a complete raster to the printer driver. It is the most efficient way of communicating with the driver since a single call can pass a full raster to the printer driver. The "C" language structure T_EPC_PACKET is described below. The second argument *dwPadToSize* should usually be set to match the current printer width. If the size of the packet passed to the library is less than the size of the width specified in the *dwPadToSize* argument, then the raster will be padded out with white pixels to make up the full *dwPadToSize* raster width. If the size of the raster (as specified in the T_EPC_PACKET) is greater than *dwPadToSize* argument, then the raster will be truncated to the *dwPadToSize* argument size.

```
int EpcReset();
```

EpcReset() toggles the control port line low then high. This action should reset any truly compatible Centronics device.

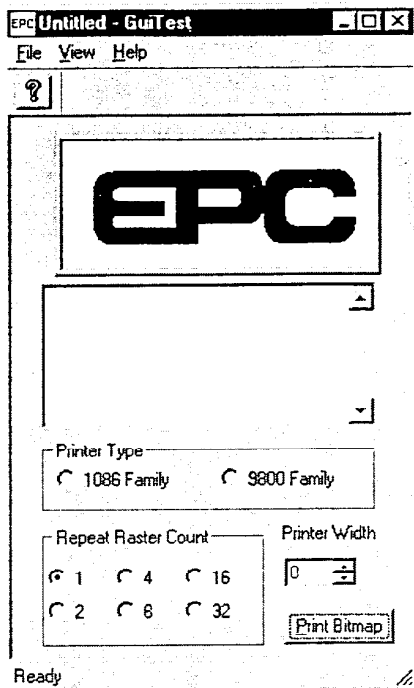
```
int EpcSendByte(UCHAR aByte);
```

EpcSendByte() provides the lowest level of control directly to user programs. The user can send any value between 0 and 255 directly to the data port of the parallel port.

```
int EpcClosePrinter();
```

It is essential to call EpcClosePrinter() after using the EPC library to avoid resource and memory leaks. EpcClosePrinter() closes the file handle to the EPC printer service and releases memory allocated by the library.

Test Applications



Two test applications are provided to illustrate how to use the EPC printer library and parallel port service. Both applications exercise the library functions to `EpcOpenPrinter()`, `EpcGetPrinterStatusByte()`, `EpcSendRaster()` and `EpcClosePrinter()`.

ConsoleTest: DOS console application.

The sample `ConsoleTest` sample application is a simple "DOS" style application that prints a gray scale test pattern to the printer using the EPC library. Error conditions are reported directly to the DOS console window.

GUI Test: an MFC/Windows Printer Driver Sample application

GUI Test is a simple MFC application that demonstrates how to print a bitmap image using the EPC Printer Driver library. A screen shot of the application is shown above. GUI Test loads a bitmap from an embedded resource file and prints this bitmap as a gray scale image using the statically linked EPCLib library described below. The library supports a repeat raster function when "stretching" an image along the paper feed direction is required. GUI Test demonstrates the `EpcSendRaster()` function. When testing with the 1086 family of printers, set the "Printer Width" control to match the current setting of the printer. For the 9800 family, the library will always pad the raster to the full capacity of the printer.

The source code for the MFC GUI Test application is installed under the installation directory in the folder ".\SampleCode\MFC\GUI Test".

Appendix A – EPCLIB Function Return Codes

Error Code	Description
eEPC_OK	Indicates successful call in all library functions
eEPC_UNABLETOOPENDRIVER	The service was detected as running but the driver could not be opened.
eEPC_FAILED_TO_OPEN_SERVICE	The service could not be started. Possible causes: 1) the service has not yet been installed; 2) the service is installed but the machine has not yet been restarted or 3) the service is installed but the registry has been corrupted.
eEPC_NULL_ARGUMENT	An expected valid argument was NULL and the function failed.
eEPC_PRINTER_NOT_OPEN	A call was made to access the printer service before the printer was opened.
eEPC_GET_STATUS_FAILED	An attempt to read the status byte from the parallel port failed.
eEPC_SEND_RASTER_FAILED,	Indicates general failure of the driver to send a raster to the printer.
eEPC_SEND_BYTE_FAILED,	Indicates a general failure of the driver to send a single byte to the printer service.
eEPC_DEVICE_TIMED_OUT	Indicates that the printer driver timed out while attempting to send data to the printer. Possible causes: 1) Cabling too long or faulty, 2) Printer off-line; 3) Parallel interface not selected from printer control panel.
eEPC_SERVICE_NOT_INSTALLED	NT service control manager did not detect EPC parallel printer service. Re-install the service and re-boot the machine.
eEPC_ACCESS_DENIED	Administrator rights are required to start the EPC parallel printer service. Have the system administrator change the start-up options of the EPC printer service to <i>Automatic</i> using the control panel applet <i>Devices</i>
eEPC_SERVICE_FAILED_TO_START	The service was installed and detected but failed to start.
eEPC_UNSUPPORTED_HEADSIZE	The head size must be either 2048 or 4096.
eEPC_BAD_RASTER_SIZE	The raster size passed in a call to EpcSendRaster() was larger than the headSize specified in the EpcOpenPrinter() call
eEPC_NOT_A_9800	A 9800 operation was attempted but the printer was opened as a non-9800 device.

Appendix B – T_EPC_PACKET definition

The T_EPC_PACKET structure is used to pass an entire raster to the printer service in one function call. The structure is defined as follows:

```
typedef struct epc_packetStruct
{
  ULONG packetSizeBytes;
  ULONG repeatCnt;
  UCHAR * pData;
} T_EPC_PACKET;
```

Field	Type	Description
PacketSizeBytes	Unsigned long	The number of bytes in the data portion of this packet.
RepeatCnt	Unsigned long	Number of times to repeat this raster
Pdata	Unsigned char pointer	Pointer to memory containing raster data.

Last Updated: 3/16/98 10:09:50 PM