

ithaca[®]

a product of **TRANSACT**

Ithaca[®] Printer Windows Driver API (Applications Programming Interface)

This page intentionally left blank

Change History

Rev A Initial release

September 2007

Disclaimer

NOTICE TO ALL PERSONS RECEIVING THIS DOCUMENT:

The information in this document is subject to change without notice. No part of this document may be reproduced, stored or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of Transact Technologies, Inc. ("Transact"). This document is the property of and contains information that is both confidential and proprietary to Transact. Recipient shall not disclose any portion of this document to any third party.

TRANSACTION DOES NOT ASSUME ANY LIABILITY FOR DAMAGES INCURRED, DIRECTLY OR INDIRECTLY, FROM ANY ERRORS, OMISSIONS OR DISCREPANCIES IN THE INFORMATION CONTAINED IN THIS DOCUMENT.

Transact cannot guarantee that changes in software and equipment made by other manufacturers, and referred to in this publication, do not affect the applicability of information in this publication.

Copyright

© 2007 Transact Technologies, Inc. All rights reserved.

Revision Level A

September 2007

Printed in USA

Trademarks

Some of the product names mentioned herein are used for identification purposes only and may be trademarks and/or registered trademarks of their respective companies.

BANKjet, 50Plus, Insta-Load, Ithaca, "Made to Order. Built to Last", Magnetec, PcOS, POSjet, PowerPocket and TransAct are registered trademarks and Epic 950, Flex-Zone, imPort, ithaColor, iTherm, KITCHENjet, Momentum, QDT and TicketBurst are trademarks of Transact Technologies, Inc.

Table of Contents

Change History	i
Disclaimer	ii
Copyright.....	ii
Trademarks	ii
Table of Contents.....	iii
Transact Windows Driver Interface	1
Who Should Read This Guide?.....	1
What is Included In This Guide?	1
About the Ithaca Windows Printer Driver	2
Ithaca Printer Extended Interface.....	3
Typical Program Flow Flowchart.....	4
MSDN function definition.....	4
Transact Extension to GetPrinterData	7
Transact I/O pValueName Functions	7
ReadStatus	7
Defined Constant	7

This page intentionally left blank

Transact Windows Driver Interface

Who Should Read This Guide?

This document provides information useful to applications programmers and original equipment manufacturers (OEM) who will develop applications for all Ithaca® printers.

What is Included In This Guide?

This document describes extended functionality to the standard Windows printer applications programming interface (API) that is specific for Ithaca® printers.

Refer to the appropriate Programmer's Guide or OEM Integration Manual for your specific printer for complete information on the mechanical, electrical, and command language requirements of each printer, which is not covered in this supplement. For further technical information, visit Transact's on-line support center at www.transact-tech.com.



About the Ithaca Windows Printer Driver

This interface definition is provided give the application programmer additional functionality and control useful in developing applications for point of sale (POS), banking, and gaming printers. The need for an extension comes from the fact that our printers include functionality that the Windows operating system does not include for a “standard” (normally office / page) printer. Some of these functions include:

- Activate a cutter
- Open a cash drawer
- Accept an inserted form for printing/validation
- Eject an inserted form after printing/validation
- Clear the power cycled status
- Feed a specified number of lines
- Send raw data to the printer
- Get direct status reports

This extended interface represents a way to control the printer via applications programming, and provides a synchronous status return from the printer. Since print jobs sent via the standard Windows spooler service are a potentially concurrent source of output to the printer, the interface has to wait for those jobs when handling the additional functions.

These functions can be invoked in the same way that the standard Windows Graphics Device Interface (GDI) function calls are used to generate and check on a print job.



Note: There is no standard definition for the precise nature of how extensions should be defined and implemented in our industries. It is thus unknown how other printers will operate if our extended driver is used to operate them.

In general, extended functions that return no data can be invoked in the context of a “StartDoc” and “EndDoc,” as used for normal print jobs. If additional functionality for receiving status is also desired, this functionality should be invoked *outside* the context of a true “start document” and “end document” segment, with the GetPrinterData() function called to perform the IO. This may require that a print job be broken up into to smaller subsets, and the extended function called in-between these subsets. Consult the program flow flowchart on page 4 for an overview of an example sequence.



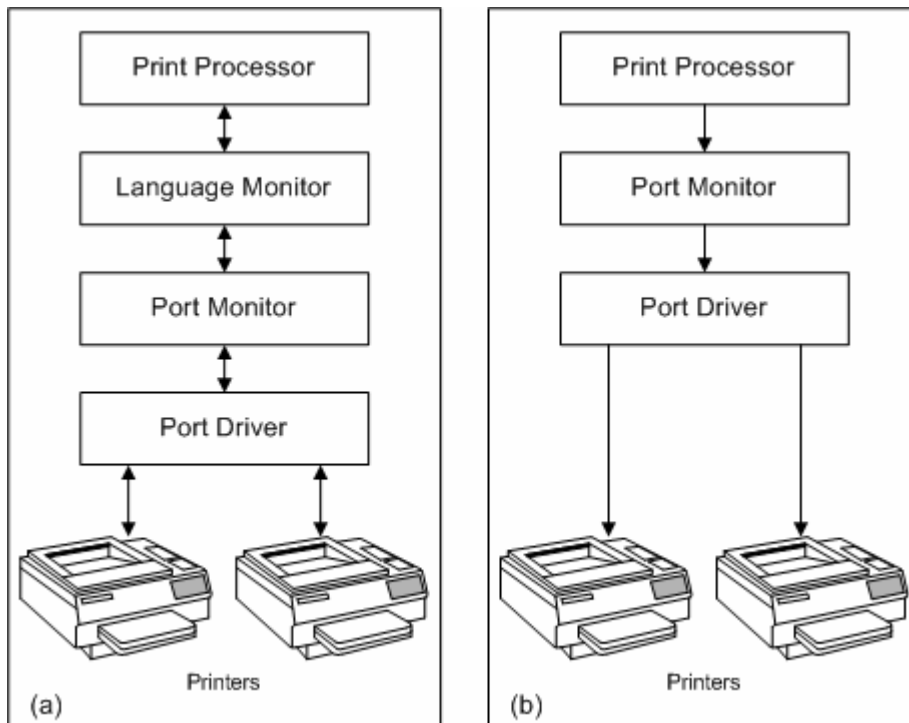
Notes: If a Transact printer’s Properties window Advanced tab has spooling selected to start printing after the last page is spooled (vs. start immediately), then it may be necessary to enclose GetPrinterData calls within a start document – end document sequence.

For best results when using the Ithaca printer extensions, do not specify any commands in the Job Control tab of the Printing Preferences menu, available from the General tab of printer properties menu, which can be brought up by right-clicking a printer in the Printers and Faxes window.

Ithaca Printer Extended Interface

The Ithaca printer extended interface is implemented in a module defined in the Windows printer architecture as the print job “Language Monitor”.

The following diagram, taken from Microsoft documentation, shows the difference between a printer installed (a) with, and (b) without a language monitor. The Print Processor is the largest module here and includes the print spooling function and the Graphical User interface widow, “Printers and Faxes” available as a selection from the Control Panel window.

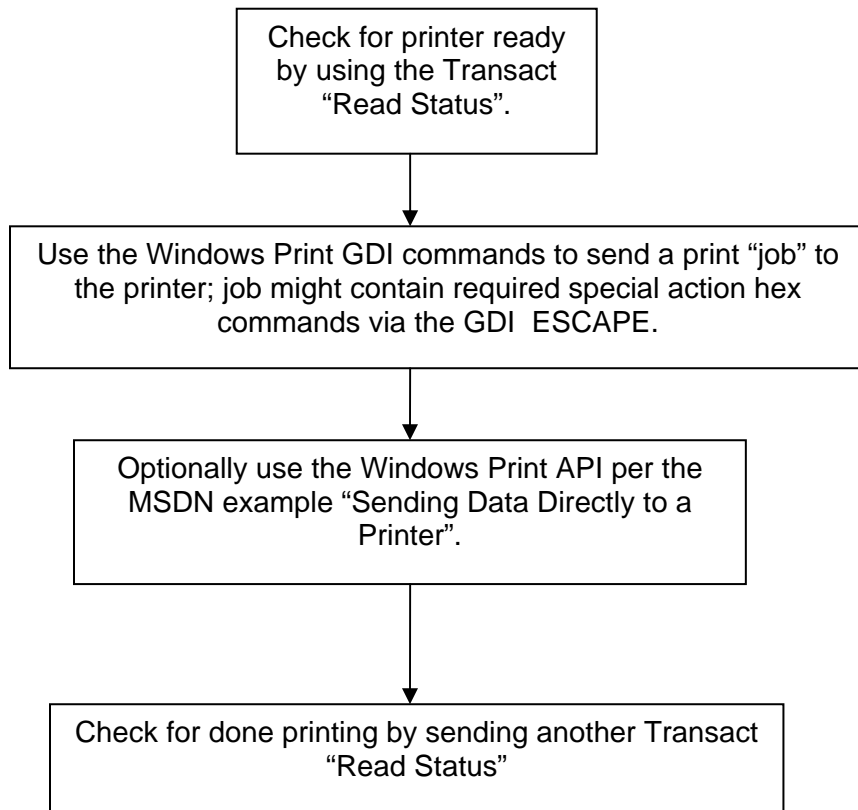


Before you can use the extended printer interface functions, you must install the standard Windows provided printer drivers that are part of the appropriate Transact provided Windows Print Driver install package, which includes a Language Monitor along with the specific printer model “mini driver” that works in conjunction with the Windows printing system.



Note: When you distribute your application you will need to distribute and install the Windows Print Driver install directory with your application. Most install package tools will take care of this for you.

Typical Program Flow Flowchart



MSDN function definition

The following definition is used as the API for the Transact extended function and is taken from Microsoft documentation. Coding is done to the standard Windows Print spooler service GDI-based interface. Our extended function is defined by a printer specific pValueName, and this Transact value is defined as a constant which can be copied directly from this document. All other values are left as-is and thus passed directly to the standard Windows port monitor.

GetPrinterData

The **GetPrinterData** function retrieves configuration data for the specified printer or print server.

Windows 2000/XP: Calling **GetPrinterData** is equivalent to calling the [GetPrinterDataEx](http://msdn2.microsoft.com/en-us/library/ms535646.aspx) [<http://msdn2.microsoft.com/en-us/library/ms535646.aspx>] function with the *pKeyName* parameter set to "PrinterDriverData".

```
DWORD GetPrinterData(  
HANDLE hPrinter, // handle to printer or print server  
LPTSTR pValueName, // value name  
LPDWORD pType, // data type  
LPBYTE pData, // pointer to data buffer  
DWORD nSize, // size of data buffer  
LPDWORD pcbNeeded // bytes received or required );
```

Parameters

hPrinter

[in] Handle to the printer or print server for which the function retrieves configuration data. Use the [OpenPrinter](http://msdn2.microsoft.com/en-us/library/ms536027.aspx) [<http://msdn2.microsoft.com/en-us/library/ms536027.aspx>] or [AddPrinter](http://msdn2.microsoft.com/en-us/library/ms535500.aspx) [<http://msdn2.microsoft.com/en-us/library/ms535500.aspx>] function to retrieve a printer handle.

pValueName

[in] Pointer to a null-terminated string that identifies the data to retrieve.

For printers, this string is the name of a registry value under the printer's "PrinterDriverData" key in the registry. Note that Transact keys for the Language Monitor are stored in a Print\Monitors section and are not retrieved by this function.

For printers with the LM installed, uniquely named *pValueName* strings are used to invoke the extended functions.

For print servers, this string is one of the predefined strings listed in the following Remarks section.

pType

[out] Pointer to a variable that receives the type of data retrieved. The function returns the type specified in the [SetPrinterData](http://msdn2.microsoft.com/en-us/library/ms535657.aspx) [<http://msdn2.microsoft.com/en-us/library/ms535657.aspx>] or [SetPrinterDataEx](http://msdn2.microsoft.com/en-us/library/ms535693.aspx) [<http://msdn2.microsoft.com/en-us/library/ms535693.aspx>] call when the data was stored.

Note that for our use this must have a value which the LM will ignore when one of the *pValueName* strings is a Transact extended function.

pData

[out] Pointer to a buffer that receives the data. For Transact extended status, the size of this buffer is at least 8 bytes long.

nSize

[in] Specifies the size, in bytes, of the buffer pointed to by *pData*.

pcbNeeded

[out] Pointer to a variable that receives the size, in bytes, of the received data. If the buffer size specified by *nSize* is too small, the function returns ERROR_MORE_DATA, and *pcbNeeded* indicates the required buffer size.

For Transact extended status, *pcbNeeded* indicates the size, in bytes, of data from the printer.

Windows 95/98/Me: If *nSize* is greater than zero, the behavior is as documented above. If *nSize* is zero, the return value is ERROR_SUCCESS and *pcbNeeded* returns the required buffer size.

Return Values

If the function succeeds, the return value is ERROR_SUCCESS. If the function fails, the return value is an error value.

Remarks

GetPrinterData retrieves printer-configuration data set by the [SetPrinterDataEx](http://msdn2.microsoft.com/en-us/library/ms535693.aspx) [<http://msdn2.microsoft.com/en-us/library/ms535693.aspx>] or [SetPrinterData](http://msdn2.microsoft.com/en-us/library/ms535657.aspx) [<http://msdn2.microsoft.com/en-us/library/ms535657.aspx>] function.

GetPrintData may trigger a Windows call to **GetPrinterDataFromPort**. (See the Windows Development Kit, formerly Driver Development Kit, for more information about **GetPrinterDataFromPort**.) The latter function may write to the registry. If it does there may be side effects like triggering an update/upgrade printer event ID 20 in the client, if the printer is shared in a network.

If *hPrinter* is a handle to a print server, *pValueName* can specify one of the following predefined values.

SPLREG_DEFAULT_SPOOL_DIRECTORY
SPLREG_PORT_THREAD_PRIORITY_DEFAULT
SPLREG_PORT_THREAD_PRIORITY
SPLREG_SCHEDULER_THREAD_PRIORITY_DEFAULT
SPLREG_SCHEDULER_THREAD_PRIORITY
SPLREG_BEEP_ENABLED
SPLREG_NET_POPUP
SPLREG_EVENT_LOG
SPLREG_MAJOR_VERSION
SPLREG_MINOR_VERSION
SPLREG_ARCHITECTURE
SPLREG_OS_VERSION

Windows 2000/XP: SPLREG_OS_VERSIONEX

SPLREG_DS_PRESENT (On successful return, *pData* contains 0x0001 if the machine is on a DS domain, 0 otherwise.)

SPLREG_DS_PRESENT_FOR_USER (On successful return, *pData* contains 0x0001 if the user is logged onto a DS domain, 0 otherwise.)

SPLREG_REMOTE_FAX (On successful return, *pData* contains 0x0001 if the FAX service supports remote clients, 0 otherwise.)

SPLREG_NET_POPUP_TO_COMPUTER (On successful return, *pData* contains 1 if job notifications should be sent to the client computer, or 0 if job notifications are to be sent to the user.)

SPLREG_RETRY_POPUP (On successful return, *pData* contains 1 if server is set to retry PopUps for all jobs, or 0 if server does *not* retry PopUps for all jobs.)

In addition, the following values indicate pool printing behavior when an error occurs.

SPLREG_RESTART_JOB_ON_POOL_ERROR (Indicates the time, in seconds, when a job is restarted on another port after an error occurs. This is used with

SPLREG_RESTART_JOB_ON_POOL_ENABLED.)

SPLREG_RESTART_JOB_ON_POOL_ENABLED (A nonzero value indicates that SPLREG_RESTART_JOB_ON_POOL_ERROR is enabled.)

The time specified in SPLREG_RESTART_JOB_ON_POOL_ERROR is a minimum time. The actual time can be longer, depending on the following port monitor settings, which are found under HKLM\SYSTEM\CurrentControlSet\Control\Print\Monitors\MonitorName\Ports.

Port monitor setting	Meaning
StatusUpdateEnabled	If a nonzero value, enables the port monitor to update the spooler with the port status.
StatusUpdateInterval	Specifies the interval, in seconds, when the port monitor updates the spooler with the port status.

Transact Extension to GetPrinterData

The above method has been extended using the defined parameters, with values defined by Transact, for the Language Monitor (“LM”) to first interpret the pValueName parameter and take special actions for values defined in this document. All other parameter values are passed to a standard, Microsoft supplied, port monitor.

An application invokes these method extensions by providing parameter values given below when calling GetPrinterData.

Transact I/O pValueName Functions

ReadStatus

This function returns the 8 byte status defined in the PcOS command: Inquire all Printer Status. Refer to the printer Programmer’s Guide for a definition of the meaning of each bit in the returned data. Response to this call will immediately return that last status if the printer has been installed with periodic status polling; otherwise the command will be sent to the printer with a 2 second timeout, and the result, which may be a timeout – indicated by all 0 data, will be written to the provided LPBYTE address.

Parameters:

HANDLE	<i>hPrinter,</i>	// Handle returned from opening the printer
LPTSTR	<i>pValueName,</i>	// the string “TARReadStatus”, null terminated
LPDWORD	<i>pType,</i>	// NULL
LPBYTE	<i>pData,</i>	// I/O data buffer address (such as &pData[0])
DWORD	<i>nSize,</i>	// >=8
LPDWORD	<i>pcbNeeded</i>	// the provided buffer should receive 8 bytes

Defined Constant

/* This pValueName parameter value should be used to invoke GetPrinterData actions proprietary to Transact printers when the Windows Print Driver is installed: */

TSTR Const TARReadStatus = “TARReadStatus”;

[end of document]