Chapter 11
Print Core Design Specification

Abstract

There are two separate print NLMs in Client32: PrntCore, which is designed to be portable across all supported platforms and implements the new 32-bit printing API, and NetxPrnt, which handles backward-compatibility with Netx printing. This document describes PrntCore.

Introduction

PrntCore was designed to make writing to a printer as much like writing to a file as possible. It is no longer necessary to redirect output to a specific port, but only to open a print job, which in turn in tied to a print device and queue.

In the VLM model, the user captured to a port, and then wrote data to that port. This model divorces the LPTx and PRN devices from a network printer. Instead of redirecting LPTx to a network print queue, the user simply opens a print device. Then, using the print device handle returned, the user opens a print job and uses the returned print job handle for subsequent writes, and closes. Because the network printer is no longer associated with an LPTx or PRN device, there is no longer any limit to the number of print devices that can be opened and accessed.

The PrntCore module provides several new features to the client:

- Unlimited number of printers devices.
- Unlimited number of open and/or active print jobs. A process or NLM will be able to open any number of print jobs per print device.
- Complete support for global and private print devices.
- No low conventional memory usage.
- Greater ease in portability to other platforms (PrntCore is written entirely in C).

Design Description

PrntCore deals primarily with print devices and print jobs. A print device is attached to exactly one print queue, which in turn is serviced by any number of print servers.

Print Devices

A print device is a structure that maintains information on the print queue to which it is bound, and the attributes of the print jobs that are sent to it. Following are the two structures which store information about print devices.

```
typedef struct PrintDevInfoTag {
   UINT32 pDevAliasHandle;          // Print device handle
   UINT32 modHandle;                // Module handle
   UINT32 pgID;                     // Process group ID
   UINT32 processID;                // Process ID
   CONN_HANDLE qConnHandle;         // Conn handle of queue server
   UINT32 tsIDNumber;               // Target server id number
   UINT8  queueName[MAX_QUEUE_NAME];// Queue name
   UINT32 pQueueID;                 // Bindery id of print queue
} PrintDevInfo;

typedef struct PDevInfoBlockTag {
   struct ResourceLinkTag;
   struct PJobAttributesTag attr;// Default print job attributes
   UINT32 pDevHandle             // Handle DEVRegisterDeviceName
   UINT32  modHandle;            // NLM module handle of owner
   UINT32  pgID;                 // Process group ID of owner
   UINT32  processID;            // Process ID of owner
   UINT8   queueName[QUEUE_NAME_SIZE];   // Name of queue
   CONN_HANDLE   qConnHandle;    // Conn handle of queue server
   UINT32  tsIDNumber;           // Target server id number
} PDevInfoBlock;
```

```
typedef struct PJobInfoBlockTag {
    struct ResourceLinkTag;
    struct PJobAttributesTag attr;      // Attributes of print job
    UINT32    modHandle;    // NLM module handle of owner
    UINT32    pgID;              // Process group ID of owner
    UINT32    processID;      // Process ID of owner
    UINT32    stateFlags;          // Print device state flags
    UINT32    fibHandle;          // FIB handle linking print device to file
    BuildFIB  bFib;
    UINT8 targetTime[6]; // Scheduled time to print
    UINT32    pJobNum;        // Job number returned from createqjob
    UINT8 jobHandle[6];       // Handle for redirected file or queue
    UINT32    pDevAliasHandle;// Print device handle
} PJobInfoBlock;



typedef struct PJobAttributesTag {
    UINT32    setupFlags;          // Print device setup flags
    UINT32    copies;          // Number of copies to print
    UINT32    maxLines;            // Number of lines per page
    UINT32    maxCols;             // Number of columns per page
    UINT32    tabSize              // Spaces per tab
    UINT32    formType             // (Continous, single sheet)
    UINT8 banner[48];          // Banner text
    UINT8 formName[48];     // Form name
    UINT8 bannerName[48];  // Banner name
    UINT8 description[48];
    UINT16   control;          // Default SERVICE_RESTART
    PSetupString pSetup;       // Printer setup string
    PSetupString pReset;            // Printer reset string
} PJobAttributes;
```

The print device and print job must be opened, and subsequent writes, flushes and closes need the job handle in order to function.

Both the print device and the print job resources have a handle associated with the resource. The handle is a 32-bit value and access to the associated resource is limited by a combination of NLM module handle, process group ID, and/or process ID. Prior to allowing access to the resource, the calling module, process group ID, and process ID is checked and must match before access is allowed. A concept of a "wild card" for any of these owning identifiers in the context of creating the resource exists to allow

other modules, process groups, or processes to access that resource. This wild card is the reserved value 0xFFFFFFFF (all 32 bits set) and may only be used when creating the resource or when enumerating resources. When this wild card is used in the context of enumerating resources, its definition means to return any for that identifier. In this way the PrntCore module provides global as well as private printing support.

Print Device Functions

A print device is tied directly to a single network print queue. It must be opened before any print jobs can be opened and assigned to it. When an open is requested, PrntCore first validates that queue. If the queue is valid, PrntCore registers the device as a system device (refer to device registry document), and then allocates, links and initializes the data structures pertaining to print devices. At this point, any print jobs that will subsequently be opened will have default settings for the following print job attributes:

Default Tabs (on/off)
Auto Form Feed (on/off)
Banner (on/off)
Print job released for printing if server is lost. (on/off)
Number of copies
Maximum number of lines per page
Maximum number of columns per page
Number of spaces a tab character represents
Form type
Banner text
Banner name
Description
Size of allocated buffer for printer setup string
Number of bytes in printer setup string
Buffer for printer setup string (contents and memory location)
Size of allocated buffer for printer reset string
Number of bytes in printer reset string
Buffer for printer reset string (contents and memory location)

An unlimited number of print jobs can be opened for any given print device. A print device can be opened global to the system, private to an NLM, private to a process group or private to a process. When a print device is closed, all open print jobs associated with that device are closed and submitted to the queue.

Print Device APIs

PRINTOpenPrintDevice	Validates queueID, registers device as system device, allocates and initializes the internal data structures.
PRINTClosePrintDevice	Checks the callers rights, deregisters the device, deallocates the internal data structures.
PRINTEnumPrintDevices	Returns a buffer containing information concerning the print device. Enumerates all available print devices.
PRINTGetDeviceData	Fills and returns a buffer containing requested print device data from the specified print device structure.
PRINTSetDeviceData	Sets specified elements of the specified print device structure.

Miscellaneous Information

There will be a single print device registered when the PrntCore module is started. This print device will be used whenever a file redirection is desired. The PrntCore module needs to behave differently for file redirection versus queue redirecting. Whenever an open job for a file redirection is requested, the device handle used in the open job will be ignored, and the file print device will be used. There is no need to open a device when redirecting to a file, since that device is automatically opened when the system is started.

Print Job Functions

A print job is tied directly to a print device. It must be opened before any data can be written to a file or to a queue. When an open is requested it first checks the callers access rights to the specified print device. If the rights are sufficient, the print job data structures are allocated and initialized.

When a print job is opened, it will fill out the defaults that have been established by the print device, such as number of copies, banner name, max number of lines, columns etc. These data elements can be changed through calls to PRINTSetJobData.

A file or queue job is not created until a PRINTWriteToPrintJob occurs. This saves on any extraneous activity on the wire.

There is no limit (except memory) to the number of print jobs per print device. All print jobs are private. There is no concept of process or process groups sharing access to a single print job.

When a print device is closed, all open print jobs associated with that device are closed and submitted to the queue.

Print Job APIs

PRINTOpenPrintJob	Validates the print device and determines if the caller has sufficient rights to the print device. Allocates and initializes the print job structure.
PRINTWriteToPrintJob	Checks the callers rights to the print job. If first write, creates a file or a queue job, writes the requested data to the file or queue job.
PRINTFlushPrintJob	Checks the callers rights to the print job. Flushes all buffers associated with the specified print job.
PRINTClosePrintJob	Checks the callers rights to the print job. Submits the queue job, closes all files associated with the print job, deallocates the related print job structure.
PRINTAbortPrintJob	Checks the callers rights to the print job. Closes, without flushing, all files associated with the print job. Does not submit the queue job, deallocates the related print job structure.
PRINTGetJobData	Checks the callers rights to the print job. Fills and returns a buffer containing requested print job data from the specified print job structure.
PRINTSetJobData	Checks the callers rights to the print job. Sets specified elements of the specified print job structure.