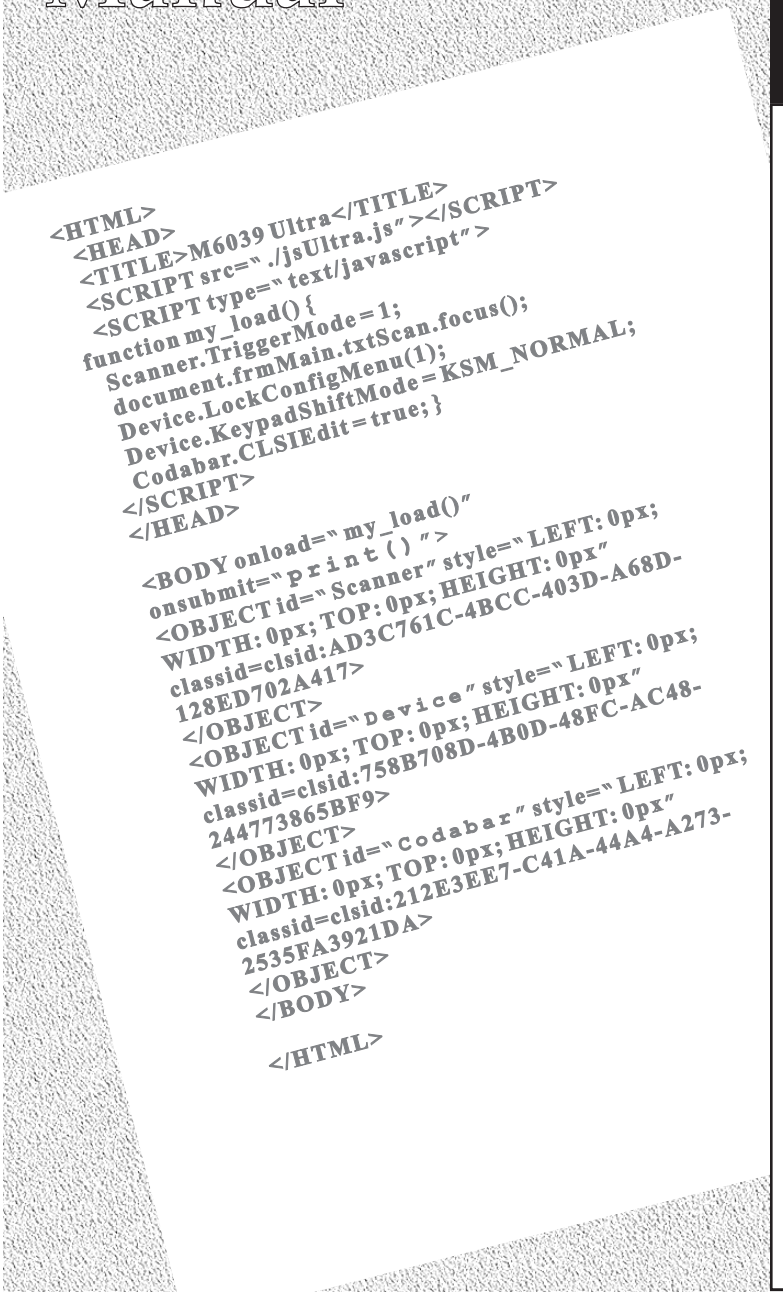
Programmer's Manual

Monarch® Pathfinder® Ultra® Platinum 6039™ Printer



```
<HTML>
 <HEAD>
 <TITLE>M6039 Ultra</TITLE>
 <SCRIPT src=" ./jsUltra.js"></SCRIPT>
 <SCRIPT type=" text/javascript">
function my_load() {
 Scanner.TriggerMode=1;
 document.frmMain.txtScan.focus();
 Device.LockConfigMenu(1);
 Device.KeypadShiftMode=KSM_NORMAL;
 Codabar.CLSIEdit=true;}
 </SCRIPT>
 </HEAD>

 <BODY onload=" my_load()"
 onsubmit=" print()">
 <OBJECT id=" Scanner" style=" LEFT: 0px;
 WIDTH: 0px; TOP: 0px; HEIGHT: 0px"
 classid=clsid:AD3C761C-4BCC-403D-A68D-
 128ED702A417>
 </OBJECT>
 <OBJECT id=" Device" style=" LEFT: 0px;
 WIDTH: 0px; TOP: 0px; HEIGHT: 0px"
 classid=clsid:758B708D-4B0D-48FC-AC48-
 244773865BF9>
 </OBJECT>
 <OBJECT id=" Codabar" style=" LEFT: 0px;
 WIDTH: 0px; TOP: 0px; HEIGHT: 0px"
 classid=clsid:212E3EE7-C41A-44A4-A273-
 2535FA3921DA>
 </OBJECT>
 </BODY>

 </HTML>
```

- Microsoft® Visual Basic® Scripting Edition-VBScript
- Microsoft® JScript®
- JavaScript™

AVERY DENNISON

TC6039COMPM Rev. AC 1/08

Each product and program carries a respective written warranty, the only warranty on which the customer can rely. Paxar reserves the right to make changes in the product and the programs and their availability at any time and without notice. Although Paxar has made every effort to provide complete and accurate information in this manual, Paxar shall not be liable for any omissions or inaccuracies. Any update will be incorporated in a later edition of this manual.

Trademarks

Monarch®, Pathfinder®, Ultra®, MPCL, and 6039 are trademarks of Paxar Americas, Inc.
Paxar® is a trademark of Paxar Corporation.
Avery Dennison® is a trademark of Avery Dennison Corporation.
JavaScript is a registered trademark of Sun Microsystems, Inc., used under license for technology invented and implemented by Netscape.

Avery Dennison Printer Systems Division
170 Monarch Lane
Miamisburg, OH 45342

TABLE OF CONTENTS

INTRODUCTION 1

The Monarch® Pathfinder® Ultra® Platinum 6039™ software development kit (SDK) helps developers write applications for the Monarch® Pathfinder® Ultra® Platinum 6039™ printer.

Note: This manual includes the library for developers using JavaScript™ language or other approved scripting language (Microsoft® Visual Basic® Scripting Edition-VBScript, Microsoft® JScript®).

Information in this document supercedes information in previous versions.

Audience

This manual is written for experienced programmers who write printer applications for the Microsoft® Windows® CE 5.0 platform. These programmers should also be familiar with the Monarch® Printer Control Language (MPCLII).

Using this Manual

Following is a summary of the contents of this manual.

Chapter		Contents
1	Introduction	Information you should know before using the SDK.
2	Developing Applications	Information about developing applications using the SDK.
3	Printing Functions	Contains syntax, definitions, and examples of each printing function.
4	Scanning Functions	Contains syntax, definitions, structures, and examples of each scanning function.
A	Sample Application	Sample application written using Javascript and HTML.

System Requirements

Following are the hardware and software requirements.

Software Requirements

- Windows® 2000 Professional Edition and Windows® 2000 Service Pack 3 or later; or Windows® XP Professional Edition and Windows® XP.
- JavaScript™, Microsoft® Visual Basic® Scripting Edition-VBScript, Microsoft® JScript®, or other scripting environment.

Minimum Hardware Requirements

- Desktop computer with Pentium® II, 450Mhz (Pentium® III, 600Mhz recommended)
- Super VGA or higher monitor
- CD-ROM or DVD-ROMdrive
- 96 MB (128 MB recommended) memory for Windows® 2000 Professional and 160 MB for Windows® XP Professional
- hard disk with 900 MB of free space

Related Documentation

The following table describes other documentation for the printer.

Item	Description
Quick Reference	Includes basic start-up information such as supply loading, cleaning and minor troubleshooting.
Operator's Handbook	Includes information about using the printer, charging the battery, loading supplies, and more.
Packet Reference Manual	Includes syntax descriptions of the MPCL printer language to design a format.
System Administrator's Guide	Includes information about printer diagnostics, configuring the scanner, and using scanner diagnostics.

About the Printer

There are several printer features that you must understand before you write an application, such as the speaker, memory, display, and keyboard.

Speaker

Applications can make the printer's speaker beep for different lengths of time and frequencies or play a .wav file. For example, you might use the speaker to bring an error to the operator's attention or to indicate a good scan. Refer to your programming language documentation to use the speaker.

Memory

The printer contains 64 MB of Flash Memory and 32 MB of RAM. The Flash memory contains the Kernel and permanent storage used for user applications.

Display

The printer has a touch screen display (with a backlight) similar to a hand held computer. Refer to your programming language documentation to write messages to the display.

Scanners

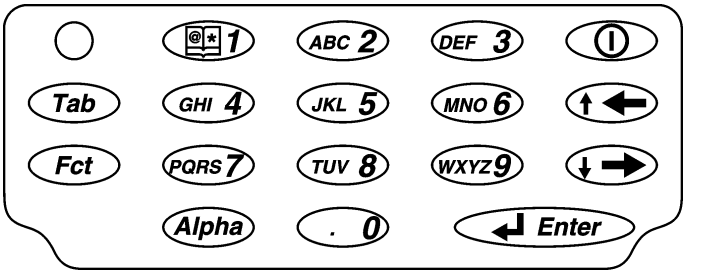
Each printer comes with the Symbol SE-955 bar code scanner.

See Chapter 4, “Scanning Functions” for functions and data structures for scanning.

Keyboard

The printer’s keypad appears to the right.

Refer to your programming language documentation to use the keyboard.



Key(s)	Description
Enter	Accepts data or a menu selection.
Tab	Tabs to the next tab stop or the next field. Pressing **Fct** + **Tab** backspaces a tab stop or returns to the previous field.
Fct	Performs an application-defined function when pressed with a single-digit number.
Alpha	Enters upper case or lower-case alphabetic mode.
Right Arrow	Moves the cursor to the right in a menu. Pressing **Fct** + **right arrow** scrolls the cursor down in a menu.
Left Arrow	Moves the cursor to the left in a menu. Pressing **Fct** + **left arrow** scrolls the cursor up in a menu. Backspaces in Alpha mode.
On/Off ⏻	Turns on and off the printer.
Numeric/ Alphabetic	Displays a numeric digit or letter.

Fonts

The printer has many resident fonts. You must load other fonts separately. Following is a list of these fonts and their IDs.

Standard (1), Reduced (2), Bold (3), OCRA (4), HR1 (5), and HR2 (6)	CG Triumvirate™ Typeface Bold 9 pt. (10)
EFF Swiss Bold (50)	CG Triumvirate™ Typeface 6 pt. (11)
CG Triumvirate™ Typeface Bold (Full Character Set) 6.5 pt. (1000)	CG Triumvirate™ Typeface Bold (Full Character Set) 8 pt. (1001)
CG Triumvirate™ Typeface Bold (Full Character Set) 10 pt. (1002)	CG Triumvirate™ Typeface Bold (Full Character Set) 12 pt. (1003)
CG Triumvirate™ Typeface Bold (Partial Character Set) 18 pt. (1004)	CG Triumvirate™ Typeface Bold (Partial Character Set) 22 pt. (1005)
CG Triumvirate™ Typeface Bold Condensed (Full Character Set) 6.5 pt. (1006)	CG Triumvirate™ Typeface Bold Condensed (Full Character Set) 8 pt. (1007)
CG Triumvirate™ Typeface Bold Condensed (Full Character Set) 10 pt. (1008)	CG Triumvirate™ Typeface Bold Condensed (Full Character Set) 12 pt. (1009)
CG Triumvirate™ Typeface Bold Condensed (Partial Character Set) 18 pt. (1010)	CG Triumvirate™ Typeface Bold Condensed (Partial Character Set) 22 pt. (1011)
Letter Gothic Bold (Full Character Set) 6 pt. (1012)	Letter Gothic Bold (Full Character Set) 9 pt. (1013)

Note: The partial character set fonts contain only numeric and special characters. With fonts 1012 and 1013, the space character is only 70% as wide as the other characters.

Using Non-Resident Fonts

Within your application, instantiate a new object, such as 6039Printer and call a function such as print to load the non-resident font or a font you have created with the MPCL® Toolbox Font Utility.

DEVELOPING APPLICATIONS 2

This chapter describes what you need to know to develop an application for the printer.

You will need to:

1. Create MPCL packets for your labels and tags, if needed.
2. Write the application.

Creating MPCL Packets

An application prints labels by submitting MPCL packets to the printer. Refer to the *6039 Packet Reference Manual* for more information.

Writing Applications

The SDK is designed to work with JavaScript™ language or other approved scripting language (Microsoft® Visual Basic® Scripting Edition-VBScript, Microsoft® JScript®).

Notes: To lock access to functions, such as the display, video, control panel, refer to your standard Microsoft® Windows® documentation.

As with other HTML or JavaScript applications, copy the application to the server when the application is complete.

Within your application, instantiate the Printer and Scanner classes such as Printer and call a method such as Print when you need to print.

Example

```
<html>
   <head>
   <title>OnDemand</title>
   <script src="./jsUltra.js"></script>
   <script type="text/javascript" language="javascript">
   function print() {
   var fmt = "{F,1,A,R,E,200,200,\"UPCA\"|
   C,150,49,0,50,8,8,A,L,0,0,\"M06039 Platinum\",1|
   B,1,12,F,25,28,1,4,100,7,L,0|
   T,2,22,V,133,1,0,1,1,1,O,C,0,0|}"
     }

Printer.ClearSystemError();             // Clear any errors
if (!Printer.IsBatteryOKToPrint)        // Check Battery
alert("Low Battery");
else {

PRINTER.Print = fmtUPCA;
PRINTER.Print = "{B,1,N,1|E,0,0,1,1,0,1|
PRINTER.Print = 1,\"" + txtUPC.Print + "\"|}";
}

</SCRIPT>
<body  onsubmit="print()">
   <object id="PRINTER" style="LEFT: 0px; WIDTH: 0px; TOP:
0px; HEIGHT: 0px"
   classid="clsid:19c1754d-ba97-43a1-a06d-496d2167400b">
   </object>
</body>
```

This example prints the data stored in the fmtUPCA string and the data stored in the txtUPC.Print field. The **`PRINTER.Print`** lines contain the batch data.

Scanner Function Overview

The scanner contains a buffer to hold the data from a scan. The application receives data from the system by one of two methods. The first method is by the standard keyboard input and second method is by a special Windows message. See Chapter 4, “Scanning Functions” for information about the two methods.

Using the Scanners

To use any of the scanners, the application must:

1. Include the code to instantiate the class. For example,

```
<SCRIPT src="./jsUltra.js"></SCRIPT>
<SCRIPT type="text/javascript">

SCANNER.Postamble = “\n\r”

GENERAL.SendScanStatus = false

</SCRIPT>
<BODY>
<OBJECT id=”SCANNER” style ="LEFT: 0px; WIDTH: 0px; TOP: 0px;
     HEIGHT: 0px"
     classid=clsid: AD3C761C-4BCC-403D-A68D-128ED702A417>
</OBJECT>                              //Instantiate Class
<OBJECT id=”GENERAL” style ="LEFT: 0px; WIDTH: 0px; TOP: 0px;
     HEIGHT: 0px"
     classid=clsid:B1EED6A7-2259-442D-B273-71EBE93C8338>
</OBJECT>                              //Instantiate Class
</BODY>
```

2. Configure the control scanning attributes and the attributes for each bar code. For example,

```
<SCRIPT src="./jsUltra.js"></SCRIPT>
<SCRIPT type="text/javascript">

UPCEAN.EnableUPCA = true
UPCEAN.EnableUPCE = false

</SCRIPT>
<BODY>
<OBJECT id=”UPCEAN” style="LEFT: 0px; WIDTH: 0px; TOP: 0px;
   HEIGHT: 0px"
   classid=clsid:67A4B8A3-A9E5-4757-97EC-2D052AEDE521>
</OBJECT>                              //Instantiate Class
</BODY>
```

3. Call the CommitConfigChanges function to save the new settings. For example,

```
<SCRIPT src="./jsUltra.js"></SCRIPT>
<SCRIPT type="text/javascript">

SCANNER.CommitConfigChanges()

</SCRIPT>
<BODY>
<OBJECT id=”SCANNER” style ="LEFT: 0px; WIDTH: 0px; TOP: 0px;
     HEIGHT: 0px"
     classid=clsid: AD3C761C-4BCC-403D-A68D-128ED702A417>
</OBJECT>                              //Instantiate Class
</BODY>
```

Programming Notes

- When the application is complete, test your application using different Browsers and verify the functionality of the application.
- As with other HTML or JavaScript applications, copy the application to the server when complete.
- The libraries included in this SDK are designed to support JavaScript™ language or other approved scripting language (Microsoft® Visual Basic® Scripting Edition-VBScript, Microsoft® JScript®); therefore, some function names changed. Many developers use the same name for Type Defines and functions, just by varying the case. However, varying the case creates a new name and your application may not function as designed.
- Train the end users (Operators) and/or their supervisor (System Administrator) on the application. They also must know how to perform procedures (loading supplies, for example) that may vary from the generic descriptions in the *Operator's Handbook.*
- When a file is saved in RAM, it is lost when the charge in the main battery and backup battery has been depleted. A file saved in the Onboard Flash folder or SD (Secure Digital) Memory Card is saved even when both batteries have been depleted.
- To lock access to functions, such as the display, video, control panel, refer to your standard Microsoft® Windows® documentation.

PRINTING FUNCTIONS 3

The SDK contains a library of functions you can call in your application. The functions are divided into two categories: Printing Interface and Scanning Interface.

This chapter describes the printer management functions and data structures. Refer to Chapter 4, “Scanning Functions” for scanning functions.

The function and data structure names are case-sensitive.

Note: Refer to your programming language documentation to configure the keyboard, sound, and display.

The libraries included in this SDK are designed to support the JavaScript™ language or other approved scripting language (Microsoft® Visual Basic® Scripting Edition-VBScript, Microsoft® JScript®). Many developers use the same name for Type Defines and functions, just by varying the case. However, varying the case creates a new name and your application may not function as designed.

Type	Functions	
Stock	CalibratePrinter	StockType
Battery	BatteryLevel CallBatteryDisplay	IsBatteryOKToPrint
Printing	PaperFeed FilePrint	Print LastPrintStatus
Sensors	BlackMark	OnDemand
Misc	ClearSystemError Status	LockCfgMenu ShiftMode

Instantiating Printer Classes

To instantiate the Printer class, use

```
<OBJECT id=”PRINTER” style="LEFT: 0px; WIDTH: 0px; TOP: 0px;
     HEIGHT: 0px"
     classid=clsid: 19C1754D-BA97-43A1-A06D-496D2167400B>
</OBJECT>
```

To instantiate the Device class, use

```
<OBJECT id=”DEVICE” style="LEFT: 0px; WIDTH: 0px; TOP: 0px;
     HEIGHT: 0px"
     classid=clsid:758B708D-4B0D-48FC-AC48-244773865BF9>
</OBJECT>
```

Refer to Chapter 4, “Scanning Functions” for information about instantiating the Scanning classes.

CalibratePrinter

Description

Calibrates the supplies in the printer and gives the supply information to the Print subsystem.

Operators can load supplies (as described in the *Operator's Handbook*) before running an application, but they cannot calibrate the supplies until the application calls this function. In general, you should display a prompt ("Load your supplies," for example) and require the operator to press a key, such as the trigger, to call this function.

Do not use this function when using fax paper because it has no black mark to detect.

Syntax

```
CalibratePrinter()
```

Parameters

None

Return Values

None

Example

```
<SCRIPT src="./jsUltra.js"></SCRIPT>
<SCRIPT type="text/javascript">

function calibratesupply()
      PRINTER.CalibratePrinter()      // Calibrate Supply

</SCRIPT>

<BODY  onload="calibratesupply()"
       onsubmit="print()">
<OBJECT id=”PRINTER” style="LEFT: 0px; WIDTH: 0px; TOP: 0px;
      HEIGHT: 0px"
      classid=clsid: 19C1754D-BA97-43A1-A06D-496D2167400B>
</OBJECT>                              //Instantiate Class

<FORM Name="frmMain" Action="javascript: print()">
       <INPUT TYPE=BUTTON value="Submit" name="btnSubmit"
       onClick="print()">
</FORM>
</BODY>
```

StockType

Description

Sets the supply type in the printer.

Syntax

```
StockType
```

Parameters

STOCK_PAPER	Paper
STOCK_FAX	Fax
STOCK_SYNTHETIC	Synthetic

Return Values

STOCK_PAPER	Paper
STOCK_FAX	Fax
STOCK_SYNTHETIC	Synthetic

Example

```
<SCRIPT src="./jsUltra.js"></SCRIPT>
<SCRIPT type="text/javascript">

PRINTER.StockType = STOCK_PAPER;        // Stock Type = Paper

</SCRIPT>
<BODY>
<OBJECT id=”PRINTER” style="LEFT: 0px; WIDTH: 0px; TOP: 0px;
     HEIGHT: 0px"
     classid=clsid: 19C1754D-BA97-43A1-A06D-496D2167400B>
</OBJECT>                               //Instantiate Class
</BODY>
```

IsBatteryOKToPrint

Description

Checks if the printer's LiIon battery (located in the handle) is charged enough to allow printing. Check the battery level before any printing session.

> Use this function immediately prior to printing, but not during printing. If you use it during printing, the return value is not accurate.

Syntax

```
IsBatteryOKToPrint
```

Parameters

None

Return Values

false	The battery level is too low to allow printing.
true	The battery level is high enough to allow printing.

Example

```
<SCRIPT src="./jsUltra.js"></SCRIPT>
<SCRIPT type="text/javascript">

PRINTER.ClearSystemError():                    // Clear any errors
     if (PRINTER.IsBatteryOKToPrint)           // OK to Print
          alert("Low Battery");

</SCRIPT>
<BODY>
<OBJECT id=”PRINTER” style="LEFT: 0px; WIDTH: 0px; TOP: 0px;
     HEIGHT: 0px"
     classid=clsid: 19C1754D-BA97-43A1-A06D-496D2167400B>
</OBJECT>                                  //Instantiate Class
</BODY>
```

BatteryLevel

Description

Retrieves the LiIon battery's level. This battery is located in the printer's handle. This returns a value between 1 and 100, the percentage of charge left in the battery. Check the battery level before any processing.

Use this function immediately prior to printing, but not during printing. If you use this function during printing, the return value is not accurate.

Syntax

```
BatteryLevel
```

Parameters

None

Return Values

0	You must charge the battery.
1 –100	The battery level is high enough to run the printer and print.

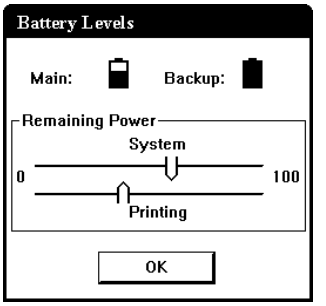
Example

```
<SCRIPT src="./jsUltra.js"></SCRIPT>
<SCRIPT type="text/javascript">

Stat;
Stat = PRINTER.BatteryLevel;
alert("Battery at " + Stat);

</SCRIPT>
<BODY>
<OBJECT id=”PRINTER” style="LEFT: 0px; WIDTH: 0px; TOP: 0px;
     HEIGHT: 0px"
     classid=clsid: 19C1754D-BA97-43A1-A06D-496D2167400B>
</OBJECT>                              //Instantiate Class
</BODY>
```

CallBatteryDisplay

Description

Retrieves the battery's display screen. This is the same screen that the user displays once they press on the status bar. Do not use this function while printing, the return value is not accurate.



Syntax

```
CallBatteryDisplay()
```

Parameters

None

Return Values

None

Example

```
<SCRIPT src="./jsUltra.js"></SCRIPT>
<SCRIPT type="text/javascript">

PRINTER.CallBatteryDisplay ();     // Get the Battery Display

</SCRIPT>
<BODY>
<OBJECT id=”PRINTER” style="LEFT: 0px; WIDTH: 0px; TOP: 0px;
     HEIGHT: 0px"
     classid=clsid: 19C1754D-BA97-43A1-A06D-496D2167400B>
</OBJECT>                                  //Instantiate Class
</BODY>
```

PaperFeed

Description

Feeds a label through the printer.

Syntax

```
PaperFeed()
```

Parameters

None

Return Values

None

Example

```
<SCRIPT src="./jsUltra.js"></SCRIPT>
<SCRIPT type="text/javascript">

PRINTER.PaperFeed();                    // Feed a Label

</SCRIPT>
<BODY>
<OBJECT id=”PRINTER” style="LEFT: 0px; WIDTH: 0px; TOP: 0px;
     HEIGHT: 0px"
     classid=clsid: 19C1754D-BA97-43A1-A06D-496D2167400B>
</OBJECT>                               //Instantiate Class
</BODY>
```

FilePrint

Description

Writes MPCL packets to the Print subsystem.

You can send more than one packet at a time in a file.

A batch packet starts a print job, which makes an asynchronous call to the Print subsystem. After submitting a print job, the application should call LastPrintStatus in a loop, waiting until the printer becomes free.

Syntax

```
FilePrint
```

Parameters

FileName to be sent for printing

Return Values

None

Example

```
<SCRIPT src="./jsUltra.js"></SCRIPT>
<SCRIPT type="text/javascript">

strScanData = "C39";
PRINTER.FilePrint = "\\Onboard Flash\\C39Fmt.txt";
PRINTER.Print = "{B,1,N,1|E,0,0,1,1,0,1|";
PRINTER.Print = "1,\"" + strScanData + "\"|2,\"" +
strScanData + "\"|}";

</SCRIPT>
<BODY>
<OBJECT id=”PRINTER” style="LEFT: 0px; WIDTH: 0px; TOP: 0px;
     HEIGHT: 0px"
     classid=clsid: 19C1754D-BA97-43A1-A06D-496D2167400B>
</OBJECT>                              //Instantiate Class
</BODY>
```

Print

Description

Writes a string to the Print subsystem. MPCL packets can be sent as a string. After sending a completed MPCL Packet use the LastPrintStatus function to get a status of the MPCL Packet.

Syntax

```
Print
```

Parameters

String to be sent to print

Return Values

None

Example

```
<SCRIPT src="./jsUltra.js"></SCRIPT>
<SCRIPT type="text/javascript">

fmtC39 = "{F,1,A,R,E,200,200,\"C39\"|"+
"C,150,49,0,50,8,8,A,L,0,0,\"M6039 Platinum\",1|" +
"T,1,22,V,6,1,0,1,1,1,O,C,0,0|" +
"B,2,20,V,23,2,4,12,100,8,C,0|}";
strScanData = "C39";

PRINTER.Print = fmtC39;
PRINTER.Print = "{B,1,N,1|E,0,0,1,1,0,1|";
PRINTER.Print = "1,\"" + strScanData + "\"|2,\"" +
strScanData + "\"|}";

</SCRIPT>
<BODY>
<OBJECT id=”PRINTER” style="LEFT: 0px; WIDTH: 0px; TOP: 0px;
     HEIGHT: 0px"
     classid=clsid: 19C1754D-BA97-43A1-A06D-496D2167400B>
</OBJECT>                              //Instantiate Class
</BODY>
```

LastPrintStatus

Description

Retrieves status of the last MPCL packet processed by the Print Engine.

Syntax

```
LastPrintStatus
```

Parameters

None

Return Values

0	Successful.
703-793	A motion control error occurred. After the operator corrects the printer condition, the application must call ClearSystemError to reset the Motion Control subsystem. Refer to the *6039 Packet Reference Manual* for more information.
Other non-zero	Error has occurred. Refer to the *6039 Packet Reference Manual* for more information.

Example

```
<SCRIPT src="./jsUltra.js"></SCRIPT>
<SCRIPT type="text/javascript">

   if (PRINTER.LastPrintStatus != 0)     // Print Status
   alert("Print Error");

</SCRIPT>
<BODY>
<OBJECT id=”PRINTER” style="LEFT: 0px; WIDTH: 0px; TOP: 0px;
     HEIGHT: 0px"
     classid=clsid: 19C1754D-BA97-43A1-A06D-496D2167400B>
</OBJECT>                              //Instantiate Class
</BODY>
```

BlackMark

Description

Retrieves the black mark sensor's latest state. This state is not necessarily the current state because it is updated only by the Print subsystem.

Syntax

```
BlackMark
```

Parameters

None

Return Values

true	The supplies are aligned on the black mark.
false	The supplies are not aligned on the black mark, or the Print subsystem is busy or uninitialized.

Example

```
<SCRIPT src="./jsUltra.js"></SCRIPT>
<SCRIPT type="text/javascript">

blnBlackMark;
blnBlackMark = Printer.BlackMark;
   if (blnBlackMark)                              // If true
   alert("Blocked);
     else                                         // If false
   alert("Not Blocked");

</SCRIPT>
<BODY>
<OBJECT id=”PRINTER” style="LEFT: 0px; WIDTH: 0px; TOP: 0px;
     HEIGHT: 0px"
     classid=clsid: 19C1754D-BA97-43A1-A06D-496D2167400B>
</OBJECT>                              //Instantiate Class
</BODY>
```

OnDemand

Description

Determines the On-demand sensor's current state. This sensor is an option for the printer.

Syntax

```
OnDemand
```

Parameters

None

Return Values

true	The sensor is blocked.
false	The sensor is not blocked.

Example

```
<SCRIPT src="./jsUltra.js"></SCRIPT>
<SCRIPT type="text/javascript">

blnOnDemand;
blnOnDemand = PRINTER.OnDemand;                    // If true
   if (blnOnDemand)
   alert("Blocked");
   else                                            // If false
   alert("Not Blocked");

</SCRIPT>
<BODY>
<OBJECT id=”PRINTER” style="LEFT: 0px; WIDTH: 0px; TOP: 0px;
     HEIGHT: 0px"
     classid=clsid: 19C1754D-BA97-43A1-A06D-496D2167400B>
</OBJECT>                                  //Instantiate Class
</BODY>
```

ClearSystemError

Description

Resets the Motion Control subsystem after an application receives a motion control error (703-793).

The operator must correct the printer condition (a supply jam, for example) before the application calls this function.

Syntax

```
ClearSystemError()
```

Parameters

None

Return Values

None

Example

```
<SCRIPT src="./jsUltra.js"></SCRIPT>
<SCRIPT type="text/javascript">

PRINTER.ClearSystemError();                    //Clear Error

</SCRIPT>
<BODY>
<OBJECT id=”PRINTER” style="LEFT: 0px; WIDTH: 0px; TOP: 0px;
     HEIGHT: 0px"
     classid=clsid: 19C1754D-BA97-43A1-A06D-496D2167400B>
</OBJECT>                              //Instantiate Class
</BODY>
```

LockConfigMenu

Description

Allows the application to control access to all configuration menus for the printer and scanner.

Syntax

```
LockConfigMenu(bool enable)
```

Instantiate Class

```
<OBJECT id=”DEVICE” style="LEFT: 0px; WIDTH: 0px; TOP: 0px;
     HEIGHT: 0px"
     classid=clsid: 758B708D-4B0D-48FC-AC48-244773865BF9>
</OBJECT>
```

Parameters

enable true to lock the menu.

false to release the lock on the menu.

Return Values

None

Example

```
<SCRIPT src="./jsUltra.js"></SCRIPT>
<SCRIPT type="text/javascript">

DEVICE.LockConfigMenu(true);           //Lock Config Menu

</SCRIPT>
<BODY>
<OBJECT id=”PRINTER” style="LEFT: 0px; WIDTH: 0px; TOP: 0px;
     HEIGHT: 0px"
     classid=clsid: 19C1754D-BA97-43A1-A06D-496D2167400B>
</OBJECT>                              //Instantiate Class
<OBJECT id=”DEVICE” style="LEFT: 0px; WIDTH: 0px; TOP: 0px;
     HEIGHT: 0px"
     classid=clsid: 758B708D-4B0D-48FC-AC48-244773865BF9>
</OBJECT>                              //Instantiate Class
</BODY>
```

Status

Description

Retrieves the Print subsystem's status.

After submitting a print job, the application should call Status in a loop, waiting until the printer becomes free.

Syntax

```
Status
```

Parameters

None

Return Values

0	The Print subsystem is ready.
1	The Print subsystem is busy.
406	MPCL parser error.
412	Could not communicate with the print engine.
601	Batch packet error.
756	Out of supplies or other motion error. Load supplies. The application must call ClearSystemError to reset the Motion Control subsystem.
762	Low battery. Printing is disabled. Replace with a fully charged battery. (Test labels can be printed.) The application must call ClearSystemError to reset the Motion Control subsystem.
763	Waiting to dispense label. The on-demand sensor may be blocked. The application must call ClearSystemError to reset the Motion Control subsystem.
791	Error pending. Operator intervention is required to clear the error. The application must call ClearSystemError to reset the Motion Control subsystem.
703-793 not listed above	A motion control error occurred. After the operator corrects the printer condition, the application must call ClearSystemError to reset the Motion Control subsystem. Refer to the *6039 Packet Reference Manual* for more information.

Example

```
<OBJECT id=”PRINTER” style="LEFT: 0px; WIDTH: 0px; TOP: 0px;
     HEIGHT: 0px"
     classid=clsid: 19C1754D-BA97-43A1-A06D-496D2167400B>
</OBJECT>                              //Instantiate Class

<SCRIPT src="./jsUltra.js"></SCRIPT>
<SCRIPT type="text/javascript">

   if (PRINTER.Status  > 1)                    // Check Status
   alert("Status");

</SCRIPT>
```

KeypadShiftMode

Description

Allows the application to the shift determine the current shift mode of the keypad and to set mode.

Syntax

```
KeypadShiftMode
```

Instantiate Class

```
<OBJECT id=”DEVICE” style="LEFT: 0px; WIDTH: 0px; TOP: 0px;
     HEIGHT: 0px"
     classid=clsid: 758B708D-4B0D-48FC-AC48-244773865BF9>
</OBJECT>
```

Parameters

KSM_NORMAL	Normal Mode -Numeric (unshifted)
KSM_FUNCTION	Function Mode (F1-F10)
KSM_LOWERALPHA	Lower-case Alpha Mode
KSM_UPPERALPHA	Upper-case Alpha Mode

Return Values

KSM_NORMAL	Normal Mode -Numeric (unshifted)
KSM_FUNCTION	Function Mode (F1-F10)
KSM_LOWERALPHA	Lower-case Alpha Mode
KSM_UPPERALPHA	Upper-case Alpha Mode

Example

```
<SCRIPT src="./jsUltra.js"></SCRIPT>
<SCRIPT type="text/javascript">

DEVICE.KeypadShiftMode = KSM_NORMAL;
                                    // Force Keypad to numbers
</SCRIPT>
<BODY>
<OBJECT id=”PRINTER” style="LEFT: 0px; WIDTH: 0px; TOP: 0px;
     HEIGHT: 0px"
     classid=clsid: 19C1754D-BA97-43A1-A06D-496D2167400B>
</OBJECT>                                  //Instantiate Class
<OBJECT id=”DEVICE” style="LEFT: 0px; WIDTH: 0px; TOP: 0px;
     HEIGHT: 0px"
     classid=clsid: 758B708D-4B0D-48FC-AC48-244773865BF9>
</OBJECT>                                  //Instantiate Class
</BODY>
```

SCANNING FUNCTIONS 4

The SDK contains a library of functions you can call in your application. The functions are divided into two categories: Printing Interface and Scanning Interface.

This chapter describes the scanning functions and data structures. Refer to Chapter 3, “Printing Functions” for printing.

The function and data structure names are case-sensitive.

Note: Refer to your programming language documentation to configure the keyboard, sound, and display.

The libraries included in this SDK are designed to support the JavaScript™ language or other approved scripting language (Microsoft® Visual Basic® Scripting Edition-VBScript, Microsoft® JScript®). Many developers use the same name for Type Defines and functions, just by varying the case. However, varying the case creates a new name and your application may not function as designed.

Type	Functions	
General Class	AimDuration BiDirRedundancy GoodScanWAVFile LinearSecurity	NoReadWAVFile Preamble Postamble Timeout
Bar Code Classes:	Codabar Code128 Code39 Code93 Code128	D2of5 I2of5 MSI RSS UPCEAN
Scanner/Control Class	CommitConfigChanges DisableAllScanCodes DoTrigger Enable	ScannerMode SendScanStatus Trigger TriggerMode

Instantiating Scanner Classes

To instantiate the Scanner or Control class, use

```
<OBJECT id=”SCANNER” style="LEFT: 0px; WIDTH: 0px; TOP: 0px;
     HEIGHT: 0px"
     classid=clsid: AD3C761C-4BCC-403D-A68D-128ED702A417>
</OBJECT>
```

To instantiate the General class, use

```
<OBJECT id=”GENERAL” style="LEFT: 0px; WIDTH: 0px; TOP: 0px;
     HEIGHT: 0px"
     classid=clsid:B1EED6A7-2259-442D-B273-71EBE93C8338>
</OBJECT>
```

Each bar code must also be instantiated. See the bar code sections for the class ID.

Refer to Chapter 3, “Printing Functions” for information about instantiating the Printing classes.

General Class

AimDuration

Description

Sets the AIM Duration, which is the duration of the aiming beam when the scanner is activated.

Syntax

`AimDuration`

Parameters

AimDuration	00 - 99 tenths of a second 0 disables. **Default**

Return Values

None

Example

```
<SCRIPT src="./jsUltra.js"></SCRIPT>
<SCRIPT type="text/javascript">

GENERAL.AimDuration =  0;            //Set Aim duration to 0
SCANNER.CommitConfigChanges();
                 //Save Changes to the Scanner Configuration

</SCRIPT>
<BODY>
<OBJECT id=”GENERAL” style="LEFT: 0px; WIDTH: 0px; TOP: 0px;
     HEIGHT: 0px"
     classid=clsid:B1EED6A7-2259-442D-B273-71EBE93C8338>
</OBJECT>                            //Instantiate Class
<OBJECT id=”SCANNER” style="LEFT: 0px; WIDTH: 0px; TOP: 0px;
     HEIGHT: 0px"
     classid=clsid: AD3C761C-4BCC-403D-A68D-128ED702A417>
</OBJECT>                            //Instantiate Class
</BODY>
```

BiDirRedundancy

Description

Enables Bi-Directional Redundancy, which specifies that good scans must occur in both directions (forward and reverse) for the scan to be complete.

Syntax

```
BiDirRedundancy
```

Parameters

BiDirRedundancy	true	Scans must occur in both directions
	false	**Default**

Return Values

None

Example

```
<SCRIPT src="./jsUltra.js"></SCRIPT>
<SCRIPT type="text/javascript">

GENERAL.BiDirRedundancy = false;
          //Scans do not have to occur in both directions
SCANNER.CommitConfigChanges();
               //Save Changes to the Scanner Configuration

</SCRIPT>
<OBJECT id=”GENERAL” style="LEFT: 0px; WIDTH: 0px; TOP: 0px;
     HEIGHT: 0px"
     classid=clsid:B1EED6A7-2259-442D-B273-71EBE93C8338>
</OBJECT>                              //Instantiate Class
<BODY>
<OBJECT id=”SCANNER” style="LEFT: 0px; WIDTH: 0px; TOP: 0px;
     HEIGHT: 0px"
     classid=clsid: AD3C761C-4BCC-403D-A68D-128ED702A417>
</OBJECT>                              //Instantiate Class
</BODY>
```

GoodScanWAVFile

Description

Sets the file for a Good Scan. This sound is heard whenever a bar code is successfully scanned. To have no sound, clear this field.

`GoodScanWAVFile`

Parameters

None

Return Values

None

Example

```
<SCRIPT src="./jsUltra.js"></SCRIPT>
<SCRIPT type="text/javascript">

GENERAL.GoodScanWAVFile = "ding.wav";      //Set WAV file

SCANNER.CommitConfigChanges();
                //Save Changes to the Scanner Configuration

</SCRIPT>
<BODY>
<OBJECT id=”GENERAL” style="LEFT: 0px; WIDTH: 0px; TOP: 0px;
     HEIGHT: 0px"
     classid=clsid:B1EED6A7-2259-442D-B273-71EBE93C8338>
</OBJECT>                              //Instantiate Class
<OBJECT id=”SCANNER” style="LEFT: 0px; WIDTH: 0px; TOP: 0px;
     HEIGHT: 0px"
     classid=clsid: AD3C761C-4BCC-403D-A68D-128ED702A417>
</OBJECT>                              //Instantiate Class
</BODY>
```

LinearSecurity

Description

Sets the Linear Security, which is how many times to scan the same bar code to determine a successful read. Select a higher level for lower quality bar codes.

Syntax

```
LinearSecurity
```

Parameters

LinearSecurity 1 – 4 scans
Default: 1

Return Values

None

Example

```
<SCRIPT src="./jsUltra.js"></SCRIPT>
<SCRIPT type="text/javascript">

GENERAL.LinearSecurity = 1;     //Scan bar code to scan once
SCANNER.CommitConfigChanges();
               //Save Changes to the Scanner Configuration

</SCRIPT>
<BODY>
<OBJECT id=”GENERAL” style="LEFT: 0px; WIDTH: 0px; TOP: 0px;
     HEIGHT: 0px"
     classid=clsid:B1EED6A7-2259-442D-B273-71EBE93C8338>
</OBJECT>                              //Instantiate Class
<OBJECT id=”SCANNER” style="LEFT: 0px; WIDTH: 0px; TOP: 0px;
     HEIGHT: 0px"
     classid=clsid: AD3C761C-4BCC-403D-A68D-128ED702A417>
</OBJECT>                              //Instantiate Class
</BODY>
```

NoReadWAVFile

Description

Sets the file for a No Scan. This sound is heard whenever a bar code is unsuccessfully scanned. To have no sound, clear this field.

Syntax

```
NoReadWAVFile
```

Parameters

None

Return Values

None

Example

```
<SCRIPT src="./jsUltra.js"></SCRIPT>
<SCRIPT type="text/javascript">

GENERAL.NoReadWAVFile = "noscan.wav";      //Set WAV file
SCANNER.CommitConfigChanges();
                //Save Changes to the Scanner Configuration

</SCRIPT>
<BODY>
<OBJECT id=”GENERAL” style="LEFT: 0px; WIDTH: 0px; TOP: 0px;
     HEIGHT: 0px"
     classid=clsid:B1EED6A7-2259-442D-B273-71EBE93C8338>
</OBJECT>                              //Instantiate Class
<OBJECT id=”SCANNER” style="LEFT: 0px; WIDTH: 0px; TOP: 0px;
     HEIGHT: 0px"
     classid=clsid: AD3C761C-4BCC-403D-A68D-128ED702A417>
</OBJECT>                              //Instantiate Class
</BODY>
```

Preamble

Description

Sets the **Preamble**, which specifies the characters to preface returned data from scanning. The Preamble can be up to twenty user-defined characters.

Syntax

```
Preamble
```

Parameters

Preamble	up to 20 characters **Default:** Empty

Return Values

None

Example

```
<SCRIPT src="./jsUltra.js"></SCRIPT>
<SCRIPT type="text/javascript">

GENERAL.Preamble = "[";
                //Adds [ at beginning of all scanned data
SCANNER.CommitConfigChanges();
                //Save Changes to the Scanner Configuration

</SCRIPT>
<BODY>
<OBJECT id=”GENERAL” style="LEFT: 0px; WIDTH: 0px; TOP: 0px;
     HEIGHT: 0px"
     classid=clsid:B1EED6A7-2259-442D-B273-71EBE93C8338>
</OBJECT>                              //Instantiate Class
<OBJECT id=”SCANNER” style="LEFT: 0px; WIDTH: 0px; TOP: 0px;
     HEIGHT: 0px"
     classid=clsid: AD3C761C-4BCC-403D-A68D-128ED702A417>
</OBJECT>                              //Instantiate Class
</BODY>
```

Postamble

Description

Sets the Postamble, which is the data to be sent after each scanned bar code. The Postamble can be up to twenty user-defined characters.

Syntax

```
Postamble
```

Parameters

Postamble	up to 20 characters **Default:** Empty

Return Values

None

Example

```
<SCRIPT src="./jsUltra.js"></SCRIPT>
<SCRIPT type="text/javascript">

GENERAL.Postamble = "]";  //Adds [ at end of all scanned data
SCANNER.CommitConfigChanges();
                //Save Changes to the Scanner Configuration

</SCRIPT>
<BODY>
<OBJECT id=”GENERAL” style="LEFT: 0px; WIDTH: 0px; TOP: 0px;
     HEIGHT: 0px"
     classid=clsid:B1EED6A7-2259-442D-B273-71EBE93C8338>
</OBJECT>                              //Instantiate Class
<OBJECT id=”SCANNER” style="LEFT: 0px; WIDTH: 0px; TOP: 0px;
     HEIGHT: 0px"
     classid=clsid: AD3C761C-4BCC-403D-A68D-128ED702A417>
</OBJECT>                              //Instantiate Class
</BODY>
```

Timeout

Description

Sets the scan Timeout in tenths of seconds, which is the amount of time the scanner beam is on before turning off when the trigger is pressed.

Syntax

`Timeout`

Parameters

Timeout	5 – 99 tenths of a second **Default:** 30

Return Values

None

Example

```
<SCRIPT src="./jsUltra.js"></SCRIPT>
<SCRIPT type="text/javascript">

GENERAL.Timeout = 30; ();       // Timeout after 30 seconds
SCANNER.CommitConfigChanges();
                //Save Changes to the Scanner Configuration

</SCRIPT>
<BODY>
<OBJECT id=”GENERAL” style="LEFT: 0px; WIDTH: 0px; TOP: 0px;
     HEIGHT: 0px"
     classid=clsid:B1EED6A7-2259-442D-B273-71EBE93C8338>
</OBJECT>                              //Instantiate Class
<OBJECT id=”SCANNER” style="LEFT: 0px; WIDTH: 0px; TOP: 0px;
     HEIGHT: 0px"
     classid=clsid: AD3C761C-4BCC-403D-A68D-128ED702A417>
</OBJECT>                              //Instantiate Class
</BODY>
```

Codabar

Description

Sets the scanner configuration values for CODABAR bar code.

Syntax

```
Enable
FixedLength
Length1
Length2
CLSIEdit
NOTISEdit
```

Instantiate Class

```
<OBJECT id=”CODABAR” style="LEFT: 0px; WIDTH: 0px; TOP: 0px;
     HEIGHT: 0px"
     classid=clsid: 212E3EE7-C41A-44A4-A273-2535FA3921DA>
</OBJECT>
```

Field	Description
Enable	Enables/disables the ability to scan Codabar bar codes. **Default:** false
FixedLength	If FixedLength is true, lengths 1 and 2 are fixed; if FixedLength is false, length 1 is the minimum and length 2 is the maximum. **Default:** false
Length1 *Length2*	Specifies lengths (including start and stop characters) for Codabar bar codes. Length 1: 0, 1-99, **Default:** 5 Length 2: 0, 1-99, **Default:** 55
CLSIEdit	Enables/disables the ability to strip the start and stop characters from 14-character Codabar bar codes and insert a space after the first, fifth, and tenth characters. **Default:** false
NOTISEdit	Enables/disables the ability to strip the start and stop characters from Codabar bar codes. **Default:** false

Return Values

None

Example

```
<SCRIPT src="./jsUltra.js"></SCRIPT>
<SCRIPT type="text/javascript">

CODABAR.Enable = true;              // Enable Codabar Symbology
CODABAR.CLSIEdit = true;            // Enable CLSIEDit
CODABAR.NOTISEdit = true;           // Enable NOTISEdit
CODABAR.FixedLength = false;        // Variable Length Enabled
CODABAR.Length1 = 2;                // Variable Length1 = 2
CODABAR.Length2 = 55;               // Variable Length2  = 55

SCANNER.CommitConfigChanges();
                //Save Changes to the Scanner Configuration

</SCRIPT>
<BODY>
<OBJECT id=”CODABAR” style="LEFT: 0px; WIDTH: 0px; TOP: 0px;
     HEIGHT: 0px"
     classid=clsid: 212E3EE7-C41A-44A4-A273-2535FA3921DA>
</OBJECT>                               //Instantiate Class
<OBJECT id=”SCANNER” style="LEFT: 0px; WIDTH: 0px; TOP: 0px;
     HEIGHT: 0px"
     classid=clsid: AD3C761C-4BCC-403D-A68D-128ED702A417>
</OBJECT>                               //Instantiate Class
</BODY>
```

Code128

Description

Sets the scanner configuration values for CODE128 bar code.

Syntax

```
Enable
UCCEAN128
ISBT128
```

Instantiate Class

```
<OBJECT id=”CODE128” style="LEFT: 0px; WIDTH: 0px; TOP: 0px;
     HEIGHT: 0px"
     classid=clsid: 85A860CB-9A63-4778-93C8-6C15F769B3DE>
</OBJECT>
```

Field	Description
Enable	Enables/disables the ability to scan Code 128 bar codes. **Default:** true
UCCEAN128	Enables/disables the ability to scan UCC/EAN-128 bar codes. **Default:** true
ISBT128	Enables/disables the ability to scan ISBT 128 bar codes. **Default:** true

Return Values

None

Example

```
<SCRIPT src="./jsUltra.js"></SCRIPT>
<SCRIPT type="text/javascript">

CODE128.Enable = true;          // Enable Code 128 Symbology
CODE128.ISBT128 = true;         // Enable ISBT128
CODE128.UCCEAN128 = true;       // Enable UCCEAN128

SCANNER.CommitConfigChanges();
               //Save Changes to the Scanner Configuration

</SCRIPT>
<BODY>
<OBJECT id=”CODE128” style="LEFT: 0px; WIDTH: 0px; TOP: 0px;
     HEIGHT: 0px"
     classid=clsid: 85A860CB-9A63-4778-93C8-6C15F769B3DE>
</OBJECT>                              //Instantiate Class
<OBJECT id=”SCANNER” style="LEFT: 0px; WIDTH: 0px; TOP: 0px;
     HEIGHT: 0px"
     classid=clsid: AD3C761C-4BCC-403D-A68D-128ED702A417>
</OBJECT>                              //Instantiate Class
</BODY>
```

Code39

Description

Sets the scanner configuration values for Code39 bar code.

Syntax

```
Enable;
Trioptic;
XlateToCode32;
Code32Prefix;
FixedLength;
Length1;
Length2;
VerifyCD;
XmitCD;
FullASCII;
```

Instantiate Class

```
<OBJECT id=”CODE39” style="LEFT: 0px; WIDTH: 0px; TOP: 0px;
     HEIGHT: 0px"
     classid=clsid: 339CD82F-F888-4500-B351-07A2B1CBEBBB>
</OBJECT>
```

Field	Description
Enable	Enables/disables the ability to scan Code 39 bar codes. **Default:** true
Trioptic	Enables/disables the ability to scan Trioptic Code 39 bar codes. Do not enable *Trioptic* and *FullASCII* at the same time. **Default:** false
XlateToCode32	Enables/disables the ability to convert Code 39 bar codes to Code 32 bar codes. You must enable *Enable* when enabling this parameter. **Default:** false
Code32Prefix	Enables/disables the ability to add “A” as a prefix to all Code 32 bar codes. You must enable *xlatetoCode32* when enabling this parameter. **Default:** false
FixedLength	If FixedLength is true, lengths 1 and 2 are fixed; if FixedLength is false, length 1 is the minimum and length 2 is the maximum. **Default:** false

Field	Description
VerifyCD	Enables/disables the ability to check the integrity of Code 39 bar codes. When this parameter is enabled, only Code 39 symbols with a modulo 43 check digit are decoded. **Default:** false
XmitCD	Enables/disables the ability to transmit check digits with the data. **Default:** false
FullASCII	Enables/disables the ability to scan Full ASCII Code 39 bar codes. The scanner cannot distinguish Code 39 bar codes from Full ASCII Code 39 bar codes. Do not enable *Trioptic* and *FullASCII* at the same time. **Default:** false

Return Values

None

Example

```
<SCRIPT src="./jsUltra.js"></SCRIPT>
<SCRIPT type="text/javascript">

CODE39.Enable = true;           // Enable Code 39 Symbology
CODE39.Trioptic = false;        // Disable Trioptic
CODE39.XlateToCode32 = false;   // Disable Convert to Code 39
CODE39.FixedLength = false;     // Variable Length Enabled
CODE39.Length1 = 2;             // Variable Length1 = 2
CODE39.Length2 = 55;            // Variable Length2 = 55
CODE39.VerifyCD = false;        // Disable Verify C/D
CODE39.XmitCD = false;          // Disable Transmit C/D
CODE39.FullASCII = false;       // Disable Full Ascii

SCANNER.CommitConfigChanges();
                //Save Changes to the Scanner Configuration

</SCRIPT>
<BODY>
<OBJECT id=”CODE39” style="LEFT: 0px; WIDTH: 0px; TOP: 0px;
     HEIGHT: 0px"
     classid=clsid: 339CD82F-F888-4500-B351-07A2B1CBEBBB>
</OBJECT>                               //Instantiate Class
<OBJECT id=”SCANNER” style="LEFT: 0px; WIDTH: 0px; TOP: 0px;
     HEIGHT: 0px"
     classid=clsid: AD3C761C-4BCC-403D-A68D-128ED702A417>
</OBJECT>                               //Instantiate Class
</BODY>
```

Code93

Description

Sets the scanner configuration values for CODE93 bar code.

Syntax

```
Enable;
FixedLength;
Length1;
Length2;
```

Instantiate Class

```
<OBJECT id=”CODE93” style="LEFT: 0px; WIDTH: 0px; TOP: 0px;
      HEIGHT: 0px"
      classid=clsid: C63FC12E-7F89-42BF-9BDA-B2E797632D0F>
</OBJECT>
```

Field	Description
Enable	Enables/disables the ability to scan Code 93 bar codes. **Default:** false
FixedLength	If FixedLength is true, lengths 1 and 2 are fixed; if FixedLength is false, length 1 is the minimum and length 2 is the maximum. **Default:** false
Length1 *Length2*	Specifies lengths (including start and stop characters) for Code 93 bar codes. Length 1: 0, 2-99, **Default:** 4 Length 2: 0, 2-99, **Default:** 55

Return Values

None

Example

```
<SCRIPT src="./jsUltra.js"></SCRIPT>
<SCRIPT type="text/javascript">

CODE93.Enable = true;                  // Enable Code93 Symbology
CODE93.FixedLength = false;            // Variable Length Enabled
CODE93.Length1 = 4;                    // Variable Lenght1 = 4
CODE93.Length2 = 20;                   // Variable Lenght1 = 20

SCANNER.CommitConfigChanges();
                //Save Changes to the Scanner Configuration

</SCRIPT>
<BODY>
<OBJECT id=”CODE93” style="LEFT: 0px; WIDTH: 0px; TOP: 0px;
     HEIGHT: 0px"
     classid=clsid: C63FC12E-7F89-42BF-9BDA-B2E797632D0F>
</OBJECT>                              //Instantiate Class
<OBJECT id=”SCANNER” style="LEFT: 0px; WIDTH: 0px; TOP: 0px;
     HEIGHT: 0px"
     classid=clsid: AD3C761C-4BCC-403D-A68D-128ED702A417>
</OBJECT>                              //Instantiate Class
</BODY>
```

D2of5

Description

Sets the scanner configuration values for D2of5 bar code.

Syntax

```
Enable;
FixedLength;
Length1;
Length2;
```

Instantiate Class

```
<OBJECT id=”D2of5” style="LEFT: 0px; WIDTH: 0px; TOP: 0px;
      HEIGHT: 0px"
      classid=clsid: 1796AB7E-36A2-4113-AA1F-58C580E44786>
</OBJECT>
```

Field	Description
Enable	Enables/disables the ability to scan D2of5 bar codes. **Default:** false
FixedLength	If FixedLength is true, lengths 1 and 2 are fixed; if FixedLength is false, length 1 is the minimum and length 2 is the maximum. **Default:** true
Length1 *Length2*	Specifies lengths (including start and stop characters) for D2of5 bar codes. Length 1: 0, 2-99, **Default:** 12 Length 2: 0, 2-99, **Default:** 0

Return Values

None

Example

```
<OBJECT id=”D2OF5” style="LEFT: 0px; WIDTH: 0px; TOP: 0px;
     HEIGHT: 0px"
     classid=clsid:1796AB7E-36A2-4113-AA1F-58C580E44786>
</OBJECT>                              //Instantiate Class
<OBJECT id=”SCANNER” style="LEFT: 0px; WIDTH: 0px; TOP: 0px;
     HEIGHT: 0px"
     classid=clsid: AD3C761C-4BCC-403D-A68D-128ED702A417>
</OBJECT>                              //Instantiate Class

<SCRIPT src="./jsUltra.js"></SCRIPT>
<SCRIPT type="text/javascript">

D2OF5.Enable = true;                  // Enable D2OF5 Symbology
D2OF5.FixedLength = false;            // Variable Length Enabled
D2OF5.Length1 = 12;                   // Variable Lenght1 = 12
D2OF5.Length2 = 0;                    // Variable Lenght1 = 0

SCANNER.CommitConfigChanges();
                //Save Changes to the Scanner Configuration

</SCRIPT>
```

I2of5

Description

Sets the scanner configuration values for I2of5 bar code.

Syntax

```
Enable;
FixedLength;
Length1;
Length2;
VerifyCD;
XmitCD;
XlateToEAN13;
```

Instantiate Class

```
<OBJECT id=”I2of5” style="LEFT: 0px; WIDTH: 0px; TOP: 0px;
     HEIGHT: 0px"
     classid=clsid: 370F6557-9281-4EAD-9F2A-3DBF260BD501>
</OBJECT>
```

Field	Description
Enable	Enables/disables the ability to scan I2of5 bar codes. **Default:** true
FixedLength	If FixedLength is true, lengths 1 and 2 are fixed; if FixedLength is false, length 1 is the minimum and length 2 is the maximum. **Default:** true
Length1 *Length2*	Specifies lengths (including start and stop characters) for I2of5 bar codes. Length 1: 0, 2-99, **Default:** 14 Length 2: 0, 2-99, **Default:** 0
VerifyCD	Specifies whether the scanner should check the integrity of 2of5 bar codes to ensure they comply with either the Uniform Symbology Specification (USS) or Optical Product Code Council (OPCC) algorithms. **Default:** false
XmitCD	Enables/disables the requirement to transmit check digits with the data. **Default:** false

Field	Description
XlateToEAN13	Enables/disables the requirement to convert 14-character I2of5 bar codes into EAN13 bar codes, and transmit them as EAN13 bar codes. To use this parameter, enable Enable, set the length to 14, and ensure the data has a leading zero and a valid EAN 13 check digit. **Default:** false

Return Values

None

Example

```
<SCRIPT src="./jsUltra.js"></SCRIPT>
<SCRIPT type="text/javascript">

I2OF5.Enable = true;                // Enable I2OF5 Symbology
I2OF5.FixedLength = true;           // Variable Length Enabled
I2OF5.Length1 = 14;                 // Variable Lenght1 = 14
I2OF5.Length2 = 0;                  // Variable Lenght1 = 0
I2of5.VerifyCD = true;              // Enable Verify C/D
I2of5.XmitCD = true;                // Enable Transmit C/D
I2of5.XlateToEAN13= false;          // Disable Convert to EAN13

SCANNER.CommitConfigChanges();
                //Save Changes to the Scanner Configuration

</SCRIPT>
<BODY>
<OBJECT id=”I2of5” style="LEFT: 0px; WIDTH: 0px; TOP: 0px;
     HEIGHT: 0px"
     classid=clsid: 370F6557-9281-4EAD-9F2A-3DBF260BD501>
</OBJECT>
<OBJECT id=”SCANNER” style="LEFT: 0px; WIDTH: 0px; TOP: 0px;
     HEIGHT: 0px"
     classid=clsid: AD3C761C-4BCC-403D-A68D-128ED702A417>
</OBJECT>                              //Instantiate Class
</BODY>
```

MSI

Description

Sets the scanner configuration values for MSI bar code.

Syntax

```
Enable;
FixedLength;
Length1;
Length2;
Use2CDs;
XmitCD;
UseMod10Mod11CDAlg;
```

Instantiate Class

```
<OBJECT id=”MSI” style="LEFT: 0px; WIDTH: 0px; TOP: 0px;
     HEIGHT: 0px"
     classid=clsid: 795F5CAE-189C-40F6-AF41-3604B108FF6A>
</OBJECT>
```

Field	Description
Enable	Enables/disables the ability to scan MSI bar codes. **Default:** false
FixedLength	If FixedLength is true, lengths 1 and 2 are fixed; if FixedLength is false, length 1 is the minimum and length 2 is the maximum. **Default:** false
Length1 *Length2*	Specifies lengths (including start and stop characters) for MSI bar codes. Length 1: 0, 1-99, **Default:** 6 Length 2: 0, 1-99, **Default:** 55
Use2CDs	Indicates the bar code contains two check digits instead of one. If true, enable UseMod10Mod11CDAlg. **Default:** false
XmitCD	Enables/disables the requirement to transmit data with the check digit. **Default:** false
UseMod10Mod11CDAlg	Uses the Mod10/Mod11 check digit algorithm instead of Mod10/Mod10 algorithm. **Default:** false

Return Values

None

Example

```
<SCRIPT src="./jsUltra.js"></SCRIPT>
<SCRIPT type="text/javascript">

MSI.Enable = true;                  // Enable MSI Symbology
MSI.FixedLength = true;             // Variable Length Enabled
MSI.Length1 = 6;                    // Variable Lenght1 = 6
MSI.Length2 = 55;                   // Variable Lenght1 = 55
MSI.Use2CDs = false;                // Disable two check digits
MSI.XmitCD = false;                 // Disable Transmit C/D
MSI.UseMod10Mod11CDAlg= false;      // Enable Mod10/Mod10

SCANNER.CommitConfigChanges();
                //Save Changes to the Scanner Configuration

</SCRIPT>
<OBJECT id=”MSI” style="LEFT: 0px; WIDTH: 0px; TOP: 0px;
     HEIGHT: 0px"
     classid=clsid: 795F5CAE-189C-40F6-AF41-3604B108FF6A>
</OBJECT>                               // Instantiate class
<BODY>
<OBJECT id=”SCANNER” style="LEFT: 0px; WIDTH: 0px; TOP: 0px;
     HEIGHT: 0px"
     classid=clsid: AD3C761C-4BCC-403D-A68D-128ED702A417>
</OBJECT>                               //Instantiate Class
</BODY>
```

RSS

Description

Sets the scanner configuration values for RSS bar code.

Syntax

```
Enable14;
EnableLimited;
EnableExpanded;
ConvertUPCEAN;
```

Instantiate Class

```
<OBJECT id=”RSS” style="LEFT: 0px; WIDTH: 0px; TOP: 0px;
      HEIGHT: 0px"
      classid=clsid: 7EC88950-001C-4519-B6A4-D3F780A98A74>
</OBJECT>
```

Field	Description
Enable14	Enables/disables the ability to scan PDF417 bar codes. **Default:** false
EnableLimited	Enables/disables the ability to scan RSS Limited bar codes. **Default:** false
EnableExpanded	Enables/disables the ability to scan RSS Expanded bar codes. **Default:** false
ConvertUPCEAN	Enables/disables the ability to convert to UPC/EAN bar codes. **Default:** false

Return Values

None

Example

```
<SCRIPT src="./jsUltra.js"></SCRIPT>
<SCRIPT type="text/javascript">

RSS.Enable14 = false;        // Disable RSS Symbology
RSS.EnableLimited = false;   // Disable RSS Limited
RSS.EnableExpanded = false;  // Disable RSS Expanded
RSS.ConvertUPCEAN = false;   // Disable Conversions to
UPC/EAN

SCANNER.CommitConfigChanges();
                //Save Changes to the Scanner Configuration

</SCRIPT>
<OBJECT id=”RSS” style="LEFT: 0px; WIDTH: 0px; TOP: 0px;
     HEIGHT: 0px"
     classid=clsid: 7EC88950-001C-4519-B6A4-D3F780A98A74>
<BODY>
</OBJECT>
<OBJECT id=”SCANNER” style="LEFT: 0px; WIDTH: 0px; TOP: 0px;
     HEIGHT: 0px"
     classid=clsid: AD3C761C-4BCC-403D-A68D-128ED702A417>
</OBJECT>                              //Instantiate Class
</BODY>
```

UPCEAN

Description

Sets the scanner configuration values for UPC and EAN bar codes.

Syntax

```
EnableUPCA;
EnableUPCE;
EnableUPCE1;
EnableEAN8;
EnableEAN13;
EnableEANBklnd;
Supplemental;
SupRedundancy;
XmitUPCACD;
XmitUPCECD;
XmitUPCE1CD;
XmitUPCAPre;
XmitUPCEPre;
XmitUPCE1Pre;
XlateEtoA;
XlateE1toA;
EANZeroExtend;
Xlate8to13;
Security;
CouponCode;
```

Instantiate Class

```
<OBJECT id=”UPCEAN” style="LEFT: 0px; WIDTH: 0px; TOP: 0px;
     HEIGHT: 0px"
     classid=clsid: 67A4B8A3-A9E5-4757-97EC-2D052AEDE521>
</OBJECT>
```

Field	Description
EnableUPCA	Enables/disables the ability to scan UPCA bar codes. **Default:** true
EnableUPCE	Enables/disables the ability to scan UPCE0 bar codes. **Default:** true
EnableUPCE1	Enables/disables the ability to scan UPCE1 bar codes. **Default:** false
EnableEAN8	Enables/disables the ability to scan EAN8 bar codes. **Default:** true
EnableEAN13	Enables/disables the ability to scan EAN13 bar codes. **Default:** true
EnableEANBkInd	Enables/disables the ability to scan Bookland EAN bar codes. **Default:** false
Supplemental	Specifies how to treat UPC and EAN bar codes with supplemental characters (UPCA+2, for example). Values are: USM_REQUIRED The scanner scans bar codes with supplemental characters only. For example, it scans a UPCA+2 bar code, but not UPCA. USM_IGNORE The scanner ignores supplemental characters. For example, it scans a UPCA+2 bar code as a UPCA. USM_AUTO Uses scanning information as specified in SupRedundancy. **Default:** USM_IGNORE
SupRedundancy	If using USM_AUTO for Supplement, this sets the number of times a symbol without supplemental information is decoded. Values are 2-20. **Default:** 7
XmitUPCACD	Enables/disables the requirement to transmit UPCA bar codes with the check digit. **Default:** true
XmitUPCECD	Enables/disables the requirement to transmit UPCE0 bar codes with the check digit. **Default:** true
XmitUPCE1CD	Enables/disables the requirement to transmit UPCE1 bar codes with the check digit. **Default:** true

Field	Description
XmitUPCAPre	Specifies how to transmit UPCA Preamble characters. Values are: 1 = XUP_NONE Do not transmit. 2 = XUP_SYSCHAR Transmit system character. 3 = XUP_SYSCOUNT Transmit system character and country code. **Default:** XUP_SYSCHAR
XmitUPCEPre	Specifies how to transmit UPCE Preamble characters. Values are: 1 = XUP_NONE Do not transmit. 2 = XUP_SYSCHAR Transmit system character. 3 = XUP_SYSCOUNT Transmit system character and country code. **Default:** XUP_SYSCHAR
XmitUPCE1Pre	Specifies how to transmit UPCE1 Preamble characters. Values are: 1 = XUP_NONE Do not transmit. 2 = XUP_SYSCHAR Transmit system character. 3 = XUP_SYSCOUNT Transmit system character and country code. **Default:** XUP_SYSCHAR
XlateEtoA	Translates a UPCE bar codes to a UPCA bar code. **Default:** false
XlateE1toA	Translates a UPCE1 bar codes to a UPCA bar code. **Default:** false
EANZeroExtend	Adds five leading zeros to scanned EAN8 bar codes to make them compatible with EAN13 bar codes. **Default:** false
Xlate8TO13	Translates an EAN8 bar code to an EAN13 bar code. **Default:** false
Security	Sets the scan security level, which is how many times to scan the same bar code to determine a successful read. Choose the minimum-security level needed. Values are 0 Use when most scans are successful. 1 Use when the unsuccessful scans are related to characters 1, 2, 7, and 8. 2 Use when the unsuccessful scans are not limited to characters 1, 2, 7, and 8. 3 Choose 4 if unsuccessful scans still occur at level 2. **Default:** 0

Field	Description
CouponCode	Enables the ability to scan UPCA/EAN Coupon Code bar codes. **Default:** false

Return Values

None

Example

```
<SCRIPT src="./jsUltra.js"></SCRIPT>
<SCRIPT type="text/javascript">

UPCEAN.EnableUPCA = true;       // Enable UPCA Symbology
UPCEAN.EnableUPCE = false;      // Disable UPCA Symbology
UPCEAN.EnableUPCE1 = false;     // Disable UPCE1 Symbology
UPCEAN.EnableEAN8 = false;      // Disable EAN8 Symbology
UPCEAN.EnableEAN13 = false;     // Disable EAN13 Symbology
UPCEAN.EnableBklnd = false;     // Disable Bklnd Symbology
UPCEAN.Supplemental = USM_IGNORE;
                           // Ignore Supplemental characters
UPCEAN.SupRedundancy = 7;       // Number of times
                                // Supplemental is decoded
UPCEAN.XmitUPCACD = true;       // Enable Transmit UPCA C/D
UPCEAN.XmitUPCECD = false; // Disable Transmit UPCE C/D
UPCEAN.XmitUPCE1CD = false;     // Disable Transmit UPCE1 C/D
UPCEAN.XmitUPCAPre = XUP_SYSCHAR;
                                // Transmit UPCA Preamble
UPCEAN.XmitUPCEPre = XUP_SYSCHAR;
                                // Transmit UPCE Preamble
UPCEAN.XmitUPCE1Pre = XUP_SYSCHAR;
                                // Transmit UPCE1 Preamble
UPCEAN.XlateEtoA = false;     // Disable Convert UPCE to UPCA
UPCEAN.XlateE1toA = false;   // Disable Convert UPCE1 to UPCA
UPCEAN.EANZeroExtend = false;   // Disable adding zeros
                                // for EAN13
UPCEAN.Xlate8TO13= false;       // Disable Convert to
                                // EAN8 to EAN13
UPCEAN.Security = 1;            // Enable Scan Security level
UPCEAN.CouponCode = true;       // Enable Coupon Scanning
```

```
SCANNER.CommitConfigChanges();
                //Save Changes to the Scanner Configuration

</SCRIPT>
<BODY>
<OBJECT id=”UPCEAN” style="LEFT: 0px; WIDTH: 0px; TOP: 0px;
     HEIGHT: 0px"
     classid=clsid: 67A4B8A3-A9E5-4757-97EC-2D052AEDE521>
</OBJECT>                          // Instantiate class
<OBJECT id=”SCANNER” style=""LEFT: 0px; WIDTH: 0px; TOP: 0px;
     HEIGHT: 0px"
     classid=clsid: AD3C761C-4BCC-403D-A68D-128ED702A417>
</OBJECT>                                  //Instantiate Class
</BODY>
```

CommitConfigChanges

Description

Retrieves the scanner configuration data from the application and saves the reconfigured settings to the scan engine.

Syntax

```
CommitConfigChanges();
```

Parameters

None

Return Values

None

Example

```
<SCRIPT src="./jsUltra.js"></SCRIPT>
<SCRIPT type="text/javascript">

SCANNER.CommitConfigChanges();
                //Save Changes to the Scanner Configuration

</SCRIPT>
<BODY>
<OBJECT id=”SCANNER” style="LEFT: 0px; WIDTH: 0px; TOP: 0px;
     HEIGHT: 0px"
     classid=clsid: AD3C761C-4BCC-403D-A68D-128ED702A417>
</OBJECT>                              //Instantiate Class
</BODY>
```

Enable

Description

Allows the scanner to scan the defined bar codes.

Syntax

```
Enable();
```

Parameters

None

Return Values

None

Example

```
<SCRIPT src="./jsUltra.js"></SCRIPT>
<SCRIPT type="text/javascript">

SCANNER.Enable();                    // Enable Scanning
SCANNER.CommitConfigChanges();
               //Save Changes to the Scanner Configuration

</SCRIPT>
<BODY>
<OBJECT id=”SCANNER” style="LEFT: 0px; WIDTH: 0px; TOP: 0px;
     HEIGHT: 0px"
     classid=clsid: AD3C761C-4BCC-403D-A68D-128ED702A417>
</OBJECT>                            //Instantiate Class
</BODY>
```

DisableAllScanCodes

Description

Disables all bar codes. This function will cause all bar codes to be disabled from being scanned. Use this function when you want to enable only a few selected codes.

Syntax

```
DisableAllScanCodes();
```

Parameters

None

Return Values

None

Example

```
<SCRIPT src="./jsUltra.js"></SCRIPT>
<SCRIPT type="text/javascript">

SCANNER.DisableAllScanCodes();       // Disable all bar codes
                                  // Enable needed bar codes here
SCANNER.CommitConfigChanges();
                //Save Changes to the Scanner Configuration

</SCRIPT>
<BODY>
<OBJECT id=”SCANNER” style="LEFT: 0px; WIDTH: 0px; TOP: 0px;
     HEIGHT: 0px"
     classid=clsid: AD3C761C-4BCC-403D-A68D-128ED702A417>
</OBJECT>                              //Instantiate Class
</BODY>
```

ScannerMode

Description

Retrieves the current scanner mode data containing the default values and saves the reconfigured settings to the scan engine.

Syntax

```
ScannerMode;
```

Parameters

None

Return Values

SOM_MOMENTARY	The scanner is on when the trigger is pressed and goes off when the trigger is released.
SOM_CONTINUOUS	The scanner is always on. A good scan causes the scanner to reset and continue scanning.
SOM_COMPATIBLE	The scanner operates in Monarch® 6037™ compatible mode, which means the scanner is on when the trigger is pressed and goes off after a successful scan or a predetermined timeout period. **Default**

Example

```
<SCRIPT src="./jsUltra.js"></SCRIPT>
<SCRIPT type="text/javascript">

SCANNER.ScannerMode = SOM_COMPATIBLE;
                        // Enable Compatibility with 6037
SCANNER.CommitConfigChanges();
               //Save Changes to the Scanner Configuration

</SCRIPT>
<BODY>
<OBJECT id=”SCANNER” style="LEFT: 0px; WIDTH: 0px; TOP: 0px;
     HEIGHT: 0px"
     classid=clsid: AD3C761C-4BCC-403D-A68D-128ED702A417>
</OBJECT>                              //Instantiate Class
</BODY>
```

SendScanStatus

Description

Enable Send Scan Status to return the data after any scan. This data precedes the bar code and includes the length of data and bar code type.

Syntax

`SendScanStatus`

Parameters

SendScanStatus	true	Enable Send Scan Status
	false	**Default**

Return Values

None

Example

```
<SCRIPT src="./jsUltra.js"></SCRIPT>
<SCRIPT type="text/javascript">

SCANNER.SendScanStatus = false;
                                        // Send Scan Status
SCANNER.CommitConfigChanges();
               //Save Changes to the Scanner Configuration

</SCRIPT>
<BODY>
<OBJECT id=”SCANNER” style="LEFT: 0px; WIDTH: 0px; TOP: 0px;
     HEIGHT: 0px"
     classid=clsid: AD3C761C-4BCC-403D-A68D-128ED702A417>
</OBJECT>                               //Instantiate Class
</BODY>
```

DoTrigger

Description

Initiates a scan, placing the scanned data in the scanner buffer. If the LED on the keypad turns green, the scan was successful. This function works with each scanner.

Syntax

```
DoTrigger();
```

Parameters

None

Return Values

None

Example

```
<SCRIPT src="./jsUltra.js"></SCRIPT>
<SCRIPT type="text/javascript">

SCANNER.DoTrigger();                        // Start Scanning
SCANNER.CommitConfigChanges();
               //Save Changes to the Scanner Configuration

</SCRIPT>
<BODY>
<OBJECT id=”SCANNER” style="LEFT: 0px; WIDTH: 0px; TOP: 0px;
     HEIGHT: 0px"
     classid=clsid: AD3C761C-4BCC-403D-A68D-128ED702A417>
</OBJECT>                                  //Instantiate Class
</BODY>
```

TriggerMode

Description

Sets the trigger mode configuration values the application set.

Syntax

```
TriggerMode;
```

Parameters

TriggerMode	The supply type. Values are	
	TM_SCAN	Pressing the trigger turns on the scanner.
	TM_DROP	The printer ignores the trigger press and does not turn on the scanner.
	TM_FORWARD	The printer passes the trigger press to the application as an F4, allowing more control of the application. You can code a custom application to perform a special function whenever it receives an F4. **Default**

Return Values

None

Example

```
<SCRIPT src="./jsUltra.js"></SCRIPT>
<SCRIPT type="text/javascript">

SCANNER.TriggerMode = TM_SCAN;
                                  // Enable scanning by trigger
SCANNER.CommitConfigChanges();
                //Save Changes to the Scanner Configuration

<BODY>
<OBJECT id=”SCANNER” style="LEFT: 0px; WIDTH: 0px; TOP: 0px;
     HEIGHT: 0px"
     classid=clsid: AD3C761C-4BCC-403D-A68D-128ED702A417>
</OBJECT>                              //Instantiate Class
</BODY>
```

SendScanStatus Codes

If you have enabled SendScanStatus in your General structures definitions, use the following table to interpret the data returned from every scan. This data precedes the bar code and includes the length of data and bar code type. See “General” in this chapter for information about enabling SendScanStatus. Refer to Appendix A, “Sample Applications” for information on receiving the scan status and extracting the length and bar code type.

Value	Bar Code Type
0	No Scan
1	Code 39
2	Codabar
3	Code 128
4	Discrete 2of5
5	IATA 2of5
6	Interleaved 2of5
7	Code 93
8	UPC A
9	UPC E
10	EAN 8
11	EAN 13
14	MSI Plessey
15	EAN 128
16	UPC E1
17	PDF417
21	Trioptic Code 39
22	Bookland EAN
23	Coupon Code

Value	Bar Code Type
48	RSS 14
49	RSS Limited
50	RSS Expanded
72	UPC A with 2 Supplements
73	UPC E with 2 Supplements
74	EAN 8 with 2 Supplements
75	EAN 13 with 2 Supplements
80	UPC E1 with 2 Supplements
136	UPC A with 5 Supplements
137	UPC E with 5 Supplements
138	EAN 8 with 5 Supplements
139	EAN 13 with 5 Supplements
144	UPC E1 with 5 Supplements

SAMPLE APPLICATION

A

This chapter contains a sample application written in Javascript and HTML that can be used as a demo.

JSULTRA.JS

```
var STOCK_PAPER = 1;      // Indicates type of stock loaded is
//paper.
var STOCK_FAX = 2;        // Indicates the type of stock is
//non-indexed fax paper.
var STOCK_SYNTHETIC = 3; // Indicates the type of stock is
//synthetic.
var SOM_MOMENTARY = 1;    // Scanner Operation Mode type that
//indicates the scanner is to be active as long as the trigger
//is held down or timeout has expired.
var SOM_CONTINUOUS = 2;  // Scanner Operation Mode type that
//indicates the scanner is to remain on until a timeout
//occurs.  Good scan will cause scanner to reset and continue
//scanning.
var SOM_COMPATIBLE = 3;  // Scanner Operation Mode type that
//indicates the scanner is to operate in M6037 compatible
//mode. Scanner on until timeout or successful scan.
var TM_SCAN = 1;          // Trigger Mode type that indicates
//the current trigger mode is to activate scanner when trigger
//is pressed / released.
var TM_DROP = 2;          // Trigger Mode type that indicates
//the current trigger mode is to ignore and drop the trigger
//press / release.
var TM_FORWARD = 3;       // Trigger Mode type that indicates
//the current trigger mode is forward the trigger press to the
//application as a F11 key press.
var USM_REQUIRED = 1;     // UPC/EAN Supplemental is required.
//Barcodes without supplemental are ignored.
var USM_IGNORE = 2;       // UPC/EAN Supplemental is ignored
//when scanned.
var USM_AUTO = 3;         // UPC/EAN Supplemental is returned
//if scanned.
```

```
var XUP_NONE = 1;         // Do not send the preamble for UPC /
//EAN barcodes.
var XUP_SYSCHAR = 2;      // Transmit a system character along
//with the decoded data.
var XUP_SYSCOUNT = 3;     // Transmit a system character and
//country code along with the decoded
//data.
var KSM_NORMAL = 1;       // keyboard shift mode - normal
var KSM_FUNCTION = 2;     // keyboard shift mode - function
var KSM_LOWERALPHA = 3;   // keyboard shift mode - lower alpha
var KSM_UPPERALPHA = 4;   // keyboard shift mode - upper alpha
function print(code, qty){var fmt =
   "{F,1,A,R,E,200,200,\"UPCA\"|C,150,49,0,50,8,8,A,L,0,0,\"M6
   0//39 Platinum\",1|
   B,1,12,F,25,28,1,4,100,7,L,0|T,2,22,V,133,1,0,1
   ,1,1,O,C,0,0|}";
var labelHead = "{B,1,N,1|E,0,0,1,1,0,1|1,\""
var labelTail = "\"|2,\"Toothpaste\"|}";
var count = 0;
Printer.Print(fmt);

qty = parseInt(qty);
if (qty > 0) {
   while (count < qty) {
      if (Printer.IsBatteryOKToPrint == false) {
         alert("Low Battery");
      }
      else {
         count = count + 1;
         Printer.Print(labelHead + code + labelTail);
      }
   }
}
else {
   Printer.Print(labelHead + code + labelTail);
   }

var status = Printer.LastPrintStatus;
   if ((status != 0) && (status != 1) && (status != 412)) {
      alert("Printer Error # " + status);
      }
}
```

ULTRA.HTML

```
<HTML>
   <HEAD>
   <TITLE>M6039 Ultra</TITLE>
   <SCRIPT src="./jsUltra.js"></SCRIPT>
   <SCRIPT type="text/javascript">

   function print() {
      if (document.frmMain.txtScan.value.length == 12) {
         window.location = ("./" +
         document.frmMain.txtScan.value + ".html")
         window.reload()
      }
   }

   function my_load() {
      Scanner.TriggerMode = 1;
      document.frmMain.txtScan.focus();
      Device.LockConfigMenu(1);
      Device.KeypadShiftMode = KSM_NORMAL;
      UPCEAN.EnableUPCA = false;
      UPCEAN.EnableUPCE = false;
      UPCEAN.EnableUPCE1 = false;
      Codabar.CLSIEdit = true;
      Codabar.NOTISEdit = true;
   }

   function my_keypress() {
      if (event.keyCode == 13) {
         print()
      }
   }

   </SCRIPT>
   </HEAD>
```

```
<BODY onLoad="my_load()"
   onSubmit="print()">
   <OBJECT id=”Scanner”
   STYLE="LEFT: 0px; WIDTH: 0px; TOP: 0px; HEIGHT: 0px"
   CLASSID=clsid:AD3C761C-4BCC-403D-A68D-128ED702A417>
   </OBJECT>

   <OBJECT id=”UltraServer”
   STYLE="LEFT: 0px; WIDTH: 0px; TOP: 0px; HEIGHT: 0px"
   CLASSID=clsid:19c1754d-ba97-43a1-a06d-496d2167400b>
   </OBJECT>

   <OBJECT id=”UPCEAN”
   STYLE="LEFT: 0px; WIDTH: 0px; TOP: 0px; HEIGHT: 0px"
   CLASSID=clsid:67A4B8A3-A9E5-4757-97EC-2D052AEDE521>
   </OBJECT>

   <OBJECT ID=”Codabar”
   STYLE="LEFT: 0px; WIDTH: 0px; TOP: 0px; HEIGHT: 0px"
   CLASSID=clsid:212E3EE7-C41A-44A4-A273-2535FA3921DA>
   </OBJECT>

   <OBJECT ID=”Device”
   STYLE="LEFT: 0px; WIDTH: 0px; TOP: 0px; HEIGHT: 0px"
   CLASSID=clsid:758B708D-4B0D-48FC-AC48-244773865BF9>
   </OBJECT>

   <FORM Name="frmMain" Action="javascript: print()">
      <INPUT TYPE=BUTTON VALUE="Submit" NAME="btnSubmit"
      onClick="print()">
      <INPUT TYPE=TEXT name="txtScan">
   </FORM>

<HR>
<A HREF="other.html">Additional</A>  |<A HREF="trigger.
html">OnDemand</A>

</BODY>
</HTML>
```

TRIGGER.HTML

```
<HTML>
   <HEAD>
   <TITLE>OnDemand</TITLE>
   <SCRIPT SRC="./jsUltra.js"></SCRIPT>
   <SCRIPT TYPE="text/javascript" LANGUAGE="javascript">

   function print() {
      var fmt = "{F,1,A,R,E,200,200,\"UPCA\"|
         C,150,49,0,50,8,8,A,L,0,0,\"M6039 Platinum\",1|
         B,1,12,F,25,28,1,4,100,7,L,0|
         T,2,22,V,133,1,0,1,1,1,O,C,0,0|}"
      var label = "{B,1,N,1|E,0,0,1,1,0,1|1,\"" +
         document.frmMain.txtUPC.value +
         "\"|2,\"Toothpaste\"|}"
      UltraServer.Print(fmt)
      UltraServer.Print(label)
   }

   function my_keydown() {
      if(window.event && window.event.keyCode == 115) {
         print();
      }
   }

   function my_load() {
      Scanner.TriggerMode = TM_FORWARD
   }

   function my_unload() {
      Scanner.TriggerMode = TM_SCAN
   }

   </SCRIPT>
   </HEAD>
```

```
<BODY onSubmit="print()"
   onKeyup="my_keydown()"
   onLoad="my_load()"
   onUnload="my_unload()">

   <OBJECT ID="UltraServer"
   STYLE="LEFT: 0px; WIDTH: 0px; TOP: 0px; HEIGHT: 0px"
   CLASSID="clsid:19c1754d-ba97-43a1-a06d-496d2167400b">
   </OBJECT>

   <OBJECT ID="Scanner"
   STYLE="LEFT: 0px; WIDTH: 0px; TOP: 0px; HEIGHT: 0px"
   CLASSID="clsid:AD3C761C-4BCC-403D-A68D-128ED702A417">
   </OBJECT>

   <FORM NAME="frmMain" ACTION="javascript: print()">
      <INPUT TYPE="text" NAME="txtUPC">
      <INPUT TYPE="button" VALUE="Print" NAME="btnSubmit"
      onClick="print()">
   </FORM>

   <HR>

   <A HREF="./Ultra.html">Back</A>

</BODY>
</HTML>
```

OTHER.HTML

```
<HTML>
   <HEAD>
      <TITLE>Other Functionality</TITLE>
      <SCRIPT src="./jsUltra.js"></SCRIPT>
      <SCRIPT type="text/javascript">

      function doFeed() {
         Printer.PaperFeed()
      }

      function trigger() {
         Scanner.DoTrigger()
      }

      function batteryLevel() {
         document.frmMain.txtOutput.value =
            rinter.BatteryLevel
      }

      function scannerMode() {
         document.frmMain.txtOutput.value =
            canner.ScannerMode
      }

      function lastPrintStatus() {
         document.frmMain.txtOutput.value = Printer.Status
      }

   </SCRIPT>

</HEAD>
```

```
<BODY>
   <OBJECT id=”Printer”
   style="LEFT: 0px; WIDTH: 0px; TOP: 0px; HEIGHT: 0px"
   classid=clsid:19c1754d-ba97-43a1-a06d-
   496d2167400b>
   </OBJECT>

   <OBJECT id=”Scanner”
   style="LEFT: 0px; WIDTH: 0px; TOP: 0px; HEIGHT: 0px"
   classid=clsid:AD3C761C-4BCC-403D-A68D-128ED702A417>
   </OBJECT>

   <OBJECT id=”General”
   style="LEFT: 0px; WIDTH: 0px; TOP: 0px; HEIGHT: 0px"
   classid=clsid:B1EED6A7-2259-442D-B273-71EBE93C8338>
   </OBJECT>

   <FORM Name="frmMain">
      <INPUT type=TEXT name="txtOutput">
      <INPUT type=BUTTON value="Feed" name="btnFeed"
         onClick="doFeed()">
      <INPUT type=BUTTON value="Trigger" name="btnTrigger"
         onClick="trigger()">
      <INPUT type=BUTTON value="Battery" name="btnBattery"
         onClick="batteryLevel()">
      <INPUT type=BUTTON value="ScannerMode"
         name="btnScannerMode" onClick="scannerMode()">
      <INPUT type=BUTTON value="Status" name="btnStatus"
         onClick="lastPrintStatus()">

   </FORM>

   <hr>
   <a href="Ultra.html">Back</a>

</BODY>
</HTML>
```

INDEX

D

E

F

G

H

S

T

U

W

Visit **www.paxar.com** for sales, service, supplies, information, and telephone numbers for our International locations.

TOLL FREE:
1-800-543-6650 (In the U.S.A.)
1-800-363-7525 (In Canada)