Web Browser Programming Guide

for Version 5.1

Naurtech Terminal Emulation and Web Browser Smart Clients

for Windows CE Devices

CETerm | CE3270 | CE5250 | CEVT220
Copyright Notice

This document may not be reproduced in full, in part or in any form, without prior written permission of Naurtech Corporation.

Naurtech Corporation makes no warranties with respect to the contents of this document and specifically disclaims any implied warranties of merchantability or fitness for any particular purpose. Further, Naurtech Corporation, reserves the right to revise this publication and to make changes to it from time to time without any obligation to notify any person or organization of such revision or changes.

Trademarks

CETerm®, CE3270™, CE5250™, CEVT220™ are trademarks of Naurtech Corporation.

Other product names mentioned in this manual may be trademarks or registered trademarks of their respective companies and are hereby acknowledged.

Software Version

This document is for version 5.1 of Naurtech Web Browser smart clients.
**Table of Contents**

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Copyright Notice</td>
<td>2</td>
</tr>
<tr>
<td>Trademarks</td>
<td>2</td>
</tr>
<tr>
<td>Software Version</td>
<td>2</td>
</tr>
<tr>
<td>Table of Contents</td>
<td>3</td>
</tr>
<tr>
<td>Preface</td>
<td>5</td>
</tr>
<tr>
<td>Assumptions</td>
<td>5</td>
</tr>
<tr>
<td>Conventions used in this Manual</td>
<td>5</td>
</tr>
<tr>
<td>Additional Documentation</td>
<td>6</td>
</tr>
<tr>
<td>Online Knowledgebase</td>
<td>6</td>
</tr>
<tr>
<td>1.0 Introduction</td>
<td>7</td>
</tr>
<tr>
<td>1.1 Feature Highlights</td>
<td>7</td>
</tr>
<tr>
<td>2.0 Common Tasks</td>
<td>10</td>
</tr>
<tr>
<td>2.1 Scanner Input</td>
<td>10</td>
</tr>
<tr>
<td>2.2 Key Actions</td>
<td>11</td>
</tr>
<tr>
<td>2.3 Text Input Elements</td>
<td>12</td>
</tr>
<tr>
<td>2.4 IDA Action Codes</td>
<td>12</td>
</tr>
<tr>
<td>2.5 Device Control From JavaScript</td>
<td>13</td>
</tr>
<tr>
<td>2.6 Device Properties and CETerm Configuration</td>
<td>15</td>
</tr>
<tr>
<td>3.0 Special HTML META Tags</td>
<td>17</td>
</tr>
<tr>
<td>3.1 Application</td>
<td>18</td>
</tr>
<tr>
<td>3.2 Battery</td>
<td>19</td>
</tr>
<tr>
<td>3.3 BatteryNavigate</td>
<td>20</td>
</tr>
<tr>
<td>3.4 Command</td>
<td>21</td>
</tr>
<tr>
<td>3.5 CursorPos</td>
<td>22</td>
</tr>
<tr>
<td>3.6 ErrorNavigate</td>
<td>22</td>
</tr>
<tr>
<td>3.7 GetUnitInformation</td>
<td>23</td>
</tr>
<tr>
<td>3.8 HomeKey</td>
<td>24</td>
</tr>
<tr>
<td>3.9 IDA</td>
<td>24</td>
</tr>
<tr>
<td>3.10 MoveSIP</td>
<td>25</td>
</tr>
<tr>
<td>3.11 OnAllKeys</td>
<td>26</td>
</tr>
<tr>
<td>3.12 OnKey</td>
<td>27</td>
</tr>
<tr>
<td>3.13 PowerOn</td>
<td>29</td>
</tr>
<tr>
<td>3.14 Reboot</td>
<td>30</td>
</tr>
<tr>
<td>3.15 Scanner</td>
<td>30</td>
</tr>
<tr>
<td>3.16 ScannerNavigate</td>
<td>31</td>
</tr>
<tr>
<td>3.17 SetDate</td>
<td>33</td>
</tr>
<tr>
<td>3.18 SetTime</td>
<td>33</td>
</tr>
<tr>
<td>3.19 Signal</td>
<td>34</td>
</tr>
<tr>
<td>3.20 SignalNavigate</td>
<td>35</td>
</tr>
<tr>
<td>3.21 SIP</td>
<td>36</td>
</tr>
<tr>
<td>3.22 SIPUp</td>
<td>37</td>
</tr>
<tr>
<td>3.23 TextSize</td>
<td>37</td>
</tr>
<tr>
<td>3.24 TimerInterval</td>
<td>38</td>
</tr>
<tr>
<td>3.25 TimerNavigate</td>
<td>38</td>
</tr>
<tr>
<td>3.26 ZebraLabel_Complete or PLSeriesLabel_Complete</td>
<td>39</td>
</tr>
<tr>
<td>3.27 ZebraLabel_Print or PLSeriesLabel_Print</td>
<td>40</td>
</tr>
<tr>
<td>4.0 Advanced Topics</td>
<td>42</td>
</tr>
</tbody>
</table>
# 4.0 Browser Client Controls

## 4.1 Navigating to Pre-configured URLs

```c
void OnClick( int x, int y )
```

## 4.2 Controlling the Scanner

```c
void OnFocus()
```

## 4.3 Input Focus and the Tab Key

```c
void OnLostFocus()
```

## 4.4 Session Launcher

```c
void OnChange()
```

## 4.5 How to Identify the Current Browser

```c
objectname.SetFocus( select )
```

## 4.6 Device Information

```c
objectname.ShowSIP( visible )
```

## 4.7 Symbol Web Client

```c
void OnKeyPress( int akey )
```

## 5.0 Printing from HTML

### 5.1 Printing with a META Tag

```c
Status = PrintString( printData )
```

### 5.2 PrintString and Print Methods

```c
Status = Print( printData )
```

### 5.3 NAURTECH:PRINT Tag

```c
Value = GetProperty( propertyName )
```

### 5.4 ActiveX Printing Controls

```c
PlaySound( sound )
```

## 6.0 CEBrowseX Control

### 6.1 Syntax

```c
ClassID
```

### 6.2 Methods

```c
PostIDA( IDACode, session )
SendIDA( IDACode, session )
SetText( text, session )
```

### 6.3 Properties

```c
Value = GetProperty( propertyName )
```

### 6.4 EventHandlers

```c
PlayTone( volume, frequency, duration )
```

### 6.5 ActiveX Printing Controls

```c
Status = PrintString( printData ) or Status = Print( printData )
```

## 7.0 TextX Control

### 7.1 Syntax

```c
ClassID
```

### 7.2 Methods

```c
objectname.SetFocus( select )
```

### 7.3 Properties

```c
objectname.ShowSIP( visible )
```

### 7.4 EventHandlers

```c
void OnChange()
```

```c
void OnClick( int x, int y )
```

```c
void OnFocus()
```

```c
void OnLostFocus()
```

```c
void OnKeyDown( int vkey )
```

```c
int OnKeyPress( int akey )
```

```c
void OnKeyUp( int vkey )
```

# Appendix 1 - Properties

# Appendix 2 - IDA Action Codes

# Appendix 3 - Virtual Key Codes

# Glossary

# Index
Preface
All of us at Naurtech Corporation constantly strive to deliver the highest quality products and services to our customers. We are always looking for ways to improve our solutions. If you have comments or suggestions, please direct these to:

Naurtech Corporation
e-mail: contact@naurtech.com
Phone: +1 (425) 837.0800

Assumptions
This manual assumes you have working knowledge of:

• Microsoft Windows user interface metaphor and terminology.
• Stylus based touch screen navigation terminology.
• Dynamic HTML, the browser DOM, and JavaScript.
• Basic operations and requirements of the host applications you want to access with the Naurtech smart client.

Conventions used in this Manual
This manual uses the following typographical conventions:

• All user actions and interactions with the application are in bold, as in [Session] [Configure]

• Any precautionary notes or tips are presented as follows

  Tip: Text associated with a specific tip

• 

  represents new version specific information
• All text associated with samples is presented as follows. We use lower case in most samples for readability.

<html>
<head>
<title>Naurtech Web Browser</title>
<meta http-equiv="Scanner" content="Enabled">
Additional Documentation

The Naurtech Web Browser is an integral feature of Naurtech terminal emulation Smart Clients. Please refer to the User's Manual for these Smart Clients for detailed installation and configuration information. The User's Manual may be downloaded from the “Support” section of our website. You may also want to refer to the CETerm Scripting Guide for additional features to enhance the web browser.

Online Knowledgebase

Although we continually strive to keep this manual up to date, you may find our online support knowledgebase useful for the latest issues, troubleshooting tips and updates. You can access the support knowledgebase from our website at:

www.naurtech.com → Support → Knowledgebase
1.0 Introduction
The Naurtech Web Browser provides a robust and flexible environment for Web based applications which are accessed from a mobile device. This browser is available for Windows CE .NET, Windows CE 5.0, Windows Mobile 2003 and Windows Mobile 5.0 based devices.

Device tailored versions of the browser are available for most industrial terminals. These versions integrate with the peripherals on each device, such as barcode scanner, magnetic stripe reader, RFID reader and Bluetooth printer. The Naurtech Web Browser provides control of the peripherals and simplifies actions such as data collection, validation, and printing.

All Naurtech Web Browsers are integrated with one or more Terminal Emulations (TE) which allows a natural migration path from legacy text based TE applications to newer Web based applications. The Web based applications can be presented in a familiar single-purpose (locked down) configuration which uses keys, the touch screen, or both for user interactions.

The Naurtech Web Browser offers control of the device peripherals and settings via integrated JavaScript extensions, ActiveX controls, and special HTML META tags. This guide is written primarily to describe these extensions and custom features. Please consult the standard references for details on JavaScript, HTML syntax, the browser Document Object Model (DOM), and other aspects of Dynamic HTML. Please refer to the User’s Manual for details on basic usage and configuration of the Naurtech Web Browser clients.

There is no current standard for the browser extensions and META tags that have been added to Web browsers for industrial handhelds. Within the Naurtech Browser, we strive to support all the ad-hoc and de-facto extensions available in other products. In many cases, the Naurtech Web Browser is a "drop-in" replacement for these other products. In addition, we support nearly uniform behavior across a wide spectrum of devices from every major hardware device manufacturers.

Beyond these basic extensions, we have added many unique features to enable you to build more powerful business applications.

1.1 Feature Highlights
Following are some of the special features and extensions in the Naurtech Web Browser.

- **Access Control / Device Lockdown.** Access controls allow administrators to hide the start bar and to prevent users from exiting the Naurtech client. You
can hide the Windows CE “Start” button, the whole Start bar and/or the application menu bars and toolbars. Users are prevented from navigating to un-authorized Web sites as is possible with Pocket IE.

- **Multiple Browser Sessions.** All Naurtech clients allow up to 4 simultaneous sessions. Each session can be connected to a different Web application. This allows quick access to separate applications. Each session supports a unique scanner configuration.

- **Enhanced native HTML text INPUT.** Under Pocket PC 2002 and Windows Mobile 2003, the standard HTML text INPUT element will popup the SIP when it receives focus, and will not respond to the Tab key to advance the focus. The Naurtech Web Browser can prevent the SIP popup and enables Tab based navigation, without resorting to ActiveX input objects. Native HTML text INPUT elements are easier to use and help maintain consistency across different platforms.

- **Scanner Control via JavaScript.** Scanner input can be intercepted by JavaScript methods for data validation and editing. The scanner can also be enabled or disabled with META tags or from JavaScript.

- **Integration with CETerm Scripting and Automation Objects.** The web browser has access to the independent JavaScript engine running within CETerm and the associated Automation Objects. This independent engine can enhance legacy web pages and add functionality. The Automation Objects provide access to more features in CETerm and the underlying Windows OS. Please refer to the CETerm Scripting Guide for full details.

- **Key Driven Interaction.** Hardware keys can be re-assigned with HTML META tags to activate any desired action. Keys can be used to navigate to specific pages, to clear fields, submit forms, and even switch to different sessions.

- **On Screen Indicators.** Battery and WLAN (RF) strength can be displayed with on-screen indicators. The meters may also be displayed within KeyBar buttons. On some devices, keypad state indicators can also be displayed on-screen.

- **Printing from HTML.** Several techniques are available to send print content from an HTML page to a printer. The printer may be accessed via a serial port, IrDA, Bluetooth or the WLAN.
- **Notification Features.** Sound (.wav) files, tone generators (beepers), and vibrators can all be activated on devices which have these capabilities.

- **Context Menus.** Custom context menus can be defined to provide access to special actions without tying up valuable screen real-estate.
2.0 Common Tasks

This section describes some common ways that enhanced features can be used within a Web based application. Here we show how to manage scanner input, invoke actions via keys, and interact with the handheld device. Only small code “snippets” are shown. For complete details see the reference sections of this manual. These tasks help to illustrate the power of the Naurtech Web Browser for building Web based applications.

2.1 Scanner Input

The barcode scanner, and other readers such as magnetic card readers and RFID readers are typically integrated with the Naurtech client. The configuration of the reader is maintained by the Naurtech client and is managed to allow an independent configuration for each host session. When data is available from a reader, it is directed to the current host session. The data is typically inserted at the current cursor (or focus) location. For a Web page with multiple text input elements, this can be problematic. The focus may be in the wrong text element, or if the focus is not in a text element, the input may be lost.

A better way to receive scanner input is to use an extension, which directs the data to a JavaScript method or submits it to a URL. To enable this action, you must define a special HTML META tag and a JavaScript method to process the data. Here is a typical META tag:

```html
<meta http-equiv="ScannerNavigate" content="Javascript:onscan('%s','%s','%s','%s','%s');">  
```

When the scanner reads a barcode, each of the '%s' items will be filled with information from the scan and the onscan method will be invoked. Here is a typical method which shows what each argument contains.

```javascript
function onscan(data, source, type, time, length)  
{  
  alert("The barcode scanned was " + data + 
  "\nThe symbology was " + type + 
  "\nScanned at " + time + 
  "\nWith a length of " + length);  
}  

```

This method simply presents a popup message with the scanner data. More typically, the data would be validated and inserted into a text element. The HTML form containing that element may also be submitted:

```html
<script language=javascript>
```


function onscan(data, source, type, time, length)
{
    if (length > 5)
    {
        document.form[0].barcode.value = data;
        document.form[0].submit();
    }
}
</script>

2.2 KEY ACTIONS

In many situations, a Web application will be written to make special use of the keys on a handheld. The device may be used without a stylus or there may be function keys (Fx) which the application uses to perform special actions. The Naurtech Web Browser has several special features to make use of the hardware keys. These features are especially important on Windows Mobile based systems. The Windows Mobile browser does not have native support for special key actions, but the Naurtech Web Browser overcomes this limitation.

The simplest way to assign a key to a special action is to use a special HTML META tag. This will instruct the browser to execute a JavaScript method or navigate to a pre-specified URL when the key is pressed:

    <meta http-equiv="OnKey0x70" content="Javascript:onF1key()"></!

...  
    <script language=javascript>
        function onF1key()
        {
            // Clear entry
            document.form[0].barcode.value = "";
        }
    </script>

or

    <meta http-equiv="OnKey0x70" content="home.htm"></!

It is important to know that the OnKey META tag acts as a “hotkey” and it will activate the action even if the focus is in a text input element. If you assign an action to a normal key such as ‘1’ then you will be unable to enter the ‘1’ as a character anywhere on the page. Techniques are available to ignore the hotkey action within text input elements.
2.3 Text Input Elements

All Web applications will use text input elements at some point to collect information such as a barcode or count. The standard text input element is the HTML INPUT such as

```html
<input type="text" name="count" maxlength="10" size="10"/>
```

The capabilities of the text input element are different for different Windows CE platforms. The Windows Mobile platform has the most limited text input. For example, this input element does not support special event handlers such as “OnKeyPress”, does not respond to the Tab key to advance the focus, and will popup the Soft Input Panel (SIP) whenever focus is received by the element.

The Naurtech Web Browser corrects these deficiencies by adding Tab key support, and by locking down the SIP when desired. We can also provide an ActiveX control (TextX) which can be used in place of the native input element if it is essential to trap all keys with an “OnKeyPress” handler. However, with the unique behavior of the OnKey META tag within native text elements, along with the Tab key and SIP control, the TextX input control is not usually needed. Avoiding the TextX input control simplifies the HTML and accelerates application development. We describe the use of the TextX control in this manual for those situations where it is required.

Under the Windows CE .NET and CE 5.0 platform, the native text INPUT element fully supports the OnKeyPress handler as well as other events, and the TextX control is never needed.

2.4 IDA Action Codes

An IDA Action Code is a special value that is used to invoke a device action, program action, or emulator action within the Naurtech Smart Client. IDA Action Codes can invoke special keys under terminal emulation, sound a tone, connect a session, or show the SIP. There are many IDA codes and these are documented Appendix 2 in this manual. Almost any action which can be invoked by a KeyBar or assigned to a hardware key, can be invoked by an IDA code. Under the Web Browser, IDA codes can be sent to the program in several different ways. They can be in a special META tag, in an HTML link, or sent via JavaScript.

Here is a sample which pops up the SIP when a page loads:

```html
<meta http-equiv="IDA" content="IDA_SIP_SHOW"/>
```
Or, you can toggle the SIP visibility from a link:

<a href="ida:IDA_SIP_TOGGLEHIDE">Toggle Soft Input Panel (SIP) </a>

Or, you can perform the action from JavaScript by setting the document location:

```javascript
function togglehide()
{
    // Toggle the SIP visibility
    // This format may not work for Pocket PC
    location.href = "ida:IDA_SIP_TOGGLEHIDE";

    // Or (remove the comment characters)
    // document.location = "ida:IDA_SIP_TOGGLEHIDE";

    // Or (remove the comment characters)
    // window.navigate("ida:IDA_SIP_TOGGLEHIDE" );
}
```

The next section describes additional methods for invoking IDA Action Codes from within a JavaScript method.

### 2.5 Device Control From JavaScript

The Naurtech Web Browser is tailored to the features of most handheld devices. If the handheld has a vibrator or tone generator, we provide access to those features through JavaScript. We also provide access to most operations of the Browser client, such as switching to other sessions, or retrieving device or configuration information.

To access these features you can use an ActiveX control or for the CE .NET or Windows CE 5.0 platforms, you can use the Document Object Model (DOM) “external” object. There are slight differences between these two approaches, however they basically work the same.

**ALERT:** The CEBrowseX control has been updated in CETerm V5.1 to provide access to the new “CETerm” and “OS” automation objects used in the CETerm script engine. The new “CETerm” object provides all the features that were previously available from CEBrowseX. Please note the revised syntax. Existing web pages will continue to work but should be converted to the new syntax as they are updated.
The following example uses the ActiveX CEBrowseX control to sound a tone.

```
<html>
<head>
<meta http-equiv="OnKey0x31" content="javascript:mybeep();"> <!-- 1 -->
</head>
<body>
<object id="CEBrowseX" classid="clsid:D14943BD-4900-453E-8582-725F21A57E0C" height=0, width=0"></object>
<a href="Javascript:mybeep();">Tap me or press 1 for beep</a><br>
...
<script language="javascript">
function mybeep()
{
  document.CEBrowseX.CETerm.PostIDA( "IDA_BEEP_LOUD", 0 );
}
</script>
</body>
</html>
```

Within this manual, we often assign the name “CEBrowseX” to the CEBrowseX control when it is created. Usually this is followed by reference to the “CETerm” object which has the methods we wish to use.

The PostIDA method sends an IDA Action Code as described in the previous section. For example, action codes can activate the vibrator (for 500 milliseconds) (IDA_VIBRATE_500), switch to a different CETerm session (IDA_SESSION_S1) and many more actions.

The sample shown above can be used on both the Windows Mobile and Windows CE .NET and 5.0 platforms. Under Windows CE, the example could also be written as

```
<html>
<head>
<meta http-equiv="OnKey0x31" content="javascript:mybeep();"> <!-- 1 -->
</head>
<body>
<a href="Javascript:mybeep();">Tap me or press 1 for beep</a><br>
...
<script language="javascript">
function mybeep()
{
  external.CEBrowseX.CETerm.PostIDA( "IDA_BEEP_LOUD", 0 );
}
</script>
</body>
</html>
```
Note that we do not require the CEBrowseX object under Windows CE because we have direct access via the builtin "external" object.

2.6 DEVICE PROPERTIES AND CETERM CONFIGURATION

The CEBrowseX object or the "external" object can be used to access device properties and to read or set portions of the CETerm configuration. The following sample shows how this can be used

```javascript
<body onload="javascript:fetchvalues();">
<object id="CEBrowseX" classid="clsid:D14943BD-4900-453E-8582-725F21A57E0C" height=0, width=0></object>
<form id="form1" name="form1">
Property Features<br>
<input type="text" id="serialnumber" name="serialnumber" size="20"><br>
<input type="text" id="ipaddress" name="ipaddress" size="20"><br>
</form>

function fetchvalues()
{
    document.form1.serialnumber.value =
    document.CEBrowseX.CETerm.GetProperty( "device.serialnumber" );
    document.form1.ipaddress.value =
    document.CEBrowseX.CETerm.GetProperty( "device.ipaddress" );
}
</script>
```

There are several different properties currently accessible and more will be added in future versions. These are documented in Appendix 1. One other useful property gives access to the "User Text" area of CETerm. User Text are strings of characters which may contain IDA codes and which can be "sent" to an emulator. User Text strings are often tied to hardware keys to simplify text entry or to create "mini-macros" of IDA actions. From the browser, the User Text can be used as a general device-local persistent storage. They can also be used to send the user to special pre-configured URL’s.

Here is an example of using a User Text area for persistent storage:

```javascript
<body onload="javascript:loadfields();">
<object id="CEBrowseX" classid="clsid:D14943BD-4900-453E-8582-725F21A57E0C" height=0, width=0></object>
<form id="form1" name="form1">
Login Page<br>
<input type="text" id="user" name="user" size="20"><br>
<input type="password" id="password" name="password" size="20"><br>
```
<form>
... 
<script language=javascript>
function loadfields()
{
    var username = document.CEBrowseX.CETerm.GetProperty("app.usertext.1");

    if (username)
    {
        // Load with cached user name for this device
        document.form1.user.value = username;
    }
}
...
</script>

When the login is successful the application would store the current user name for the next login attempt:

... 
function onlogin( username )
{
    // Login was successful
    document.CEBrowseX.CETerm.SetProperty( "app.usertext.1", username );
}
...

There are some limitations with User Text. The values are shared among all emulator sessions and currently there are only 64 slots available.
3.0 Special HTML META Tags

This section describes the special META tags (or elements) that are recognized by the Naurtech Web Browser and are used to convey special instructions to the browser. The META tags use the standard HTML format but are not recognized by standard browsers. These special META tags assign hotkeys, control the scanner, configure the device, and other tasks.

A META tag has the form

```html
<meta http-equiv="identifier" content="valueString"/>
```

The Naurtech Browser supports a unique set of “identifier” values. However, because there is no current standard for these META tags, we also recognize most of the ad-hoc and de-facto extensions available in other products. In particular, we support most of the Symbol Pocket Browser and Intermec iBrowse META tag identifiers. For the iBrowse META tags, we recognize the identifiers with or without the leading “IBrowse_” text.

The following table summarizes the supported identifier values and their functions. Following the table are individual sections with detailed descriptions of the identifiers. All identifiers are effective only on the page that contains them unless otherwise specified in the description.

<table>
<thead>
<tr>
<th>Identifier</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application</td>
<td>Exit the program</td>
</tr>
<tr>
<td>Battery</td>
<td>Display on-screen battery information</td>
</tr>
<tr>
<td>BatteryNavigate</td>
<td>JavaScript or URL invoked with battery information</td>
</tr>
<tr>
<td>Command</td>
<td>Exit the program</td>
</tr>
<tr>
<td>CursorPos</td>
<td>Set location of wait cursor</td>
</tr>
<tr>
<td>ErrorNavigate</td>
<td>JavaScript or URL invoked on error</td>
</tr>
<tr>
<td>GetUnitInformation</td>
<td>JavaScript or URL invoked with device information</td>
</tr>
<tr>
<td>HomeKey</td>
<td>Enable/disable home key (F5)</td>
</tr>
<tr>
<td>IDA</td>
<td>Invoke IDA Action Code</td>
</tr>
<tr>
<td>MoveSIP</td>
<td>Set location of SIP input panel</td>
</tr>
<tr>
<td>OnAllKeys</td>
<td>Bind all keys to JavaScript or URL action</td>
</tr>
<tr>
<td>OnKey</td>
<td>Bind key to JavaScript or URL action</td>
</tr>
<tr>
<td>PowerOn</td>
<td>JavaScript or URL invoked on resume from suspend</td>
</tr>
<tr>
<td>Reboot</td>
<td>Invoke a device reset</td>
</tr>
<tr>
<td>Scanner</td>
<td>Enable/disable the scanner</td>
</tr>
<tr>
<td>ScannerNavigate</td>
<td>JavaScript or URL invoked on successful scan</td>
</tr>
<tr>
<td>SetDate</td>
<td>Set the system date</td>
</tr>
</tbody>
</table>
### Identifier | Description
--- | ---
SetTime | Set the system time
Signal | Display on-screen WLAN (RF) signal information
SignalNavigate | JavaScript or URL invoked with WLAN information
SIP | Control the SIP
SIPUp | Show the SIP
TextSize | Set text size factor (zoom)
TimerInterval | Set interval for TimerNavigate
TimerNavigate | JavaScript or URL invoked on TimerInterval
ZebraLabel_Print | Contents of data for printing
ZebraLabel_Complete | JavaScript or URL invoked to report print status

Several special tags contain a JavaScript statement or URL in the content. Depending on the tag, these may contain the text “%s” which is replaced by data which is unique to the tag action. You must have the correct number of replacement placeholders depending on the tag. If the number is incorrect, the tag will not be recognized.

When specifying a URL, it may be any standard form. The JavaScript: form is simply a type of URL. You may also use files local on the device:

```
"file:///Application Data/myapp/errorpage.htm?errno=%s&msg=%s"
```

You may also use the proprietary “ida:” type to invoke various actions.

### 3.1 APPLICATION

The Application tag performs actions which affect the client application. Currently this only exits the web browser.

**Syntax**

```
http-equiv="Application"
content="Quit"
```

**Comments**

This tag is supported for compatibility with other browsers. It is preferable to use the IDA Action Code “IDA_PROGRAM_EXIT” via the IDA META tag, an “ida:” URL or a PostIDA call.

**Example**

```html
<html>
```
3.2 Battery

The Battery tag is used to configure the on-screen battery strength meter. This meter overlays the HTML content and is updated at a specified interval. You can change the location and style of the meter. Available styles include a horizontal or vertical single bar meter or filled battery icon. The meter may be repositioned by a “touch and drag” stylus action if dragging is not disabled.

Syntax

http-equiv="Battery"
content="Show"
   "Hide"
"Right"
"Left"
"Top"
"Bottom"
"x=n"
"y=m"
"HBattery"
"VBattery"
"AllowDrag"
"NoDrag"

Where x and y are the screen coordinates of the upper left corner of the meter icon. The screen coordinate (0, 0) is in the upper left corner of the screen with x increasing to the right and y increasing downward. Right, Left, Top, and Bottom change the orientation of the meter and the placement of the icon. HBattery enables a horizontal battery icon filled according to strength. VBattery enables a vertical battery icon filled according to strength. AllowDrag will allow the user to drag the meter, whereas NoDrag will prevent dragging.

Comments

This tag is used to control the battery meter display. The meter may also be controlled within the CETerm configuration, independent of the Battery tag. The Battery tag will always override the internal configuration and will persist until changed by another Battery tag. Within the CETerm configuration you can
specify the update interval and a notification when the strength falls below a designated level.

**Example**

```html
<html>
<head>
<meta http-equiv="Battery" content="Show">
<meta http-equiv="Battery" content="HBattery">
<meta http-equiv="Battery" content="x=200">
<meta http-equiv="Battery" content="y=20">
</head>
<body>
The horizontal battery icon should be visible.
...
</body>
</html>
```

### 3.3 BatteryNavigate

The BatteryNavigate tag causes the specified JavaScript or URL to be invoked with battery information on a regular interval. The interval can be specified within a CETerm configuration dialog.

**Syntax**

```text
http-equiv="BatteryNavigate"
content="javascript:OnBattery(%s, %s, %s, %s);
<!- or ->
content="/battery.htm?AC=%s&strength=%s&backup=%s&chemistry=%s"
```

Where the "%s" are replaced with (1) AC line status, (2) main battery strength as a percentage, (3) backup battery strength as a percentage, and (3) the chemistry of the battery.

**Comments**

This tag will work with or without a visible battery meter. The strength normally ranges from 0 to 100. The special strength value of -1 indicates that the strength cannot be determined. If you use a URL for the action, in most cases, the URL will navigate away from the current page rather than repeatedly calling a JavaScript method.

**Example**

```html
<html>
```
<head>
<meta http-equiv="BatteryNavigate"
content="javascript:OnBattery(%s, %s, %s, %s);">
</head>
<body>
<script language=javascript>
function OnBattery(ACstate, strength, backup, chemistry)
{
  if (strength == -1)
  {
    alert("Unable to determine battery strength.");
  }
  else
  {
    alert("Battery strength = " + strength);
  }
}
</script>
...
</body>
</html>

3.4 COMMAND
The Command tag performs actions which affect the client application. Currently this only exits the web browser.

Syntax
http-equiv="Command"
content="exit"

Comments
This tag is supported for compatibility with other browsers. It is preferable to use the IDA Action Code “IDA_PROGRAM_EXIT” via the IDA META tag, an “ida.” URL or a PostIDA call.

Example
<html>
<head>
<meta http-equiv="Command" content="exit">
</head>
<body>
This message should not be visible.
</body>
</html>
3.5 CURSORPOS

The CursorPos tag is used to reposition the “busy-cursor” when it is visible. The busy cursor may appear while waiting for pages to load. Use CursorPos to move the busy-cursor from the default location at the center of the screen.

Syntax

```html
http-equiv="CursorPos"
content="x=n"
"y=m"
```

Where x and y are the screen coordinates with (0, 0) in the upper left corner of the screen and x increasing to the right and y increasing downward.

Comments

The new position only applies to the page loading busy-cursor; other busy-cursors will be unaffected. The new position will apply until changed with another CursorPos tag.

Example

```html
<html>
<head>
<meta http-equiv="CursorPos" content="x=0">
<meta http-equiv=" CursorPos" content="y=0">
</head>
<body>
Busy cursor in upper left corner.
...
</body>
</html>
```

3.6 ERRORNAVIGATE

The ErrorNavigate tag directs error messages to a JavaScript method or to a URL.

Syntax

```html
http-equiv="ErrorNavigate"
content="javascript:MyErrorHandler(%s, %s);
<!-- or -->
content="http://10.1.1.8/errorpage.htm?errno=%s&msg=%s"
<!- or to hide errors ->
content="javascript:var HideErrors='%s';;"
```

The first “%s” is replaced by an error number and the second “%s” by an error message.
Comments
This tag should be the first special META tag defined on the page, but not before any JavaScript methods that may be invoked by the content. If this tag is not specified, errors are reported via popup messages.

Example
<html>
<head>
<script language=javascript>
function MyErrorHandler(errno, msg)
{
  alert("Error Number=" + errno + "\nMessage=" + msg );
}
</script>
<meta http-equiv="ErrorNavigate"
  content="javascript:MyErrorHandler(%s, %s);">
</head>
<body>
...
</body>
</html>

3.7 **GetUnitInformation**
The GetUnitInformation tag reports device and client information to the host or user.

Syntax
http-equiv="GetUnitInformation"
content="javascript:ReportInfo(%s, %s, %s);"
<!- or ->
content="http://10.1.1.8/uinfo.htm?serial=%s&uuid=%s&version=%s"

The first "%s" is replaced by the device serial number, the second %s by the Windows CE device UUID, and third "%s" by the Web Browser version.

Comments
This tag is supported for compatibility with other browsers. It is preferable to use the “CETerm.GetProperty” method of the CEBrowseX control or the “external” object (under CE .NET or 5.0 only).

Example
<html>
<head>
<meta http-equiv="GetUnitInformation"
  content="javascript:ReportInfo(%s, %s, %s);">
This page reports unit information.

```javascript
function ReportInfo(serial, uuid, version)
{
    alert( "Serial Number=" + serial +
            "\nUUID=" + uuid +
            "\nVersion=" + version );
}
</script>
</body>
</html>

3.8 HOMEKEY
The HomeKey tag enables the “home key” (F5) to navigate to the current home URL.

Syntax
http-equiv="HomeKey"
content="Enabled"
"Disabled"

Comments
This tag is supported for compatibility with other browsers. It is preferable to use the IDA Action Code “IDA_URL_HOME” via the IDA META tag, an “ida:” URL or a PostIDA call.

This action can also be achieved with the “OnKey” tag. This tag remains in effect until explicitly changed or the session ends.

Example
<html>
<head>
<meta http-equiv="HomeKey" content="Enabled">
</head>
<body>
Press F5 to navigate to the home URL.
</body>
</html>

3.9 IDA
The IDA tag performs a wide range of actions to control the device and the client.
Syntax

http-equiv="IDA"
content="IDA_symbolicname"

Comments

This tag offers rich functionality. It is used primarily when an action is needed upon loading a page. Alternatively, an “onload” handler can be used in the BODY element to perform actions via JavaScript.

A list of IDA Action Codes and their description can be found in the Appendix. All IDA symbolic names must be in upper case.

Any number of IDA meta tags may be specified on a single page. They are acted upon sequentially when parsed.

Example

<html>
<head>
<meta http-equiv="IDA" content="IDA_SIP_SHOW">
</head>
<body>
The Soft Input Panel (SIP) should be visible.
</body>
</html>

3.10 MoveSIP

The MoveSIP tag is used to reposition the Soft Input Panel (SIP). MoveSIP is not recommended for hiding the SIP off-screen. Use the SIP lockdown features within CETerm to prevent the SIP from popping up when not wanted.

Syntax

http-equiv="MoveSIP"
content="x=n"
"y=m"

Where x and y are the screen coordinates with (0,0) in the upper left corner of the screen and x increasing to the right and y increasing downward.

Comments
Use MoveSIP to move the default SIP location. **WARNING:** MoveSIP may move the SIP to a non-visible location. Usually, entering the CETerm configuration dialogs will temporarily restore the SIP to the default location. Also, window and scroll behavior may be erratic on Windows Mobile devices with the SIP in a non-standard location.

**Example**

```html
<html>
<head>
<meta http-equiv="MoveSIP" content="x=0">
<meta http-equiv="MoveSIP" content="y=240">
</head>
<body>
SIP is shifted down a bit.
...
</body>
</html>
```

### 3.11 **OnAllKeys**

The OnAllKeys tag assigns a single JavaScript action or URL to all hardware keys on the handheld device. The action will take place regardless of the focus location on the page.

**Syntax**

```
http-equiv="OnAllKeys"

content="javascript:myKeyAction(%s);"
<!- or ->
content="http://10.1.1.8/inventory.htm?key=%s"
```

The “%s” is replaced by the Windows CE “Virtual Key Code” (VK) value of the key pressed.

**Comments**

See the Appendix 3 for a list of Virtual Key Codes, their symbolic names, hexadecimal representation, and the typical keyboard name. If an OnKey tag has been specified for an individual key, that tag’s action will be invoked in place of the OnAllKeys action. See the OnKey tag for additional information.

**Example**

```html
<html>
```
Main Menu<br>1. Cycle Count<br>2. Inventory<br>3. Receiving<br>Select an action:<br>Press ‘A’ to check version.<br>...<br><script language=javascript>
function myKeyAction(vkcode)
{
  alert("Key pressed =" + vkcode);
}
</script>
</body>
</html>

3.12 ONKEY
The OnKey tag assigns a JavaScript action or URL to hardware keys on the handheld device. The action will take place regardless of the focus location on the page.

Syntax
http-equiv="OnKey0xZZ"
"OnKeyDDD"
"OnKeyIgnoreInText"
"OnKeyVK_name"  (version 4.5.3 or later)
"OnKey_name"    (version 4.5.3 or later)
content="javascript:myKeyAction();"
<!- or ->
content="http://10.1.1.8/inventory.htm"

Where ZZ is a two digit hexadecimal number that represents the Windows CE “Virtual Key Code” (VK) for a physical key, or “DDD” is up to 3 decimal digits representing the VK code, or “name” is the portion of the Virtual Key symbolic name after the underscore.

Comments
See the Appendix for a list of Virtual Key Codes, their symbolic names, hexadecimal representation, and the typical keyboard name. This tag allows several different formats to specify the VK value. Using the symbolic name
format yields HTML that is easier to read and maintain. See OnAllKeys to direct all key input to a single action.

Although most VK codes are uniform across devices, some devices can remap the keyboard at a driver level to change the VK codes. Consult your hardware documentation to understand what VK codes are generated by the keys on your device.

Please note that some keys may be tied to operating system actions and they may not be sent to the running applications, thus they cannot be used. Other times, they are tied to an action and will still be sent to the application, so you may see side-effects of their use.

All key shift states such as CTRL, ALT, and Shift are ignored by the OnKey action. However, it is possible to use shift states via key remapping as described below. We recommended that you do **not** assign OnKey actions to the CTRL, ALT, or SHIFT keys themselves.

The Naurtech Web Browser has very flexible key remapping features. In most cases, any user action which simulates a key, such as a KeyBar button or scanner post-amble will invoke the OnKey action. For example, you may remap Shift+"1" within the Web Browser key remapping to perform the F1 key action. When Shift+"1" is pressed in a browser session, the OnKey action for F1 will be invoked. Tapping on an “F1” KeyBar button will also invoke the OnKey action.

Unlike the Naurtech Web Browser, some other browsers do not act on OnKey assignments when the focus is in a native text INPUT element or in an ActiveX text component. Under Pocket PC and Windows Mobile 2003, the “OnKeyIgnoreInText” tag can be used to ignore the OnKey assignments when the focus is in a native text input element or in a TextX input object. The content is ignored for “OnKeyIgnoreInText”

There is no limit to the number of OnKey assignments within a page.

### Example

```html
<html>
<head>
<meta http-equiv="OnKey0x1B" content="javascript:document.form[0].clearbutton.click();"> <!-- ESC -->
<meta http-equiv="OnKey0x31" content="/cyclecount.htm"> <!-- 1 -->
<meta http-equiv="OnKey0x32" content="/inventory.htm"> <!-- 2 -->
<meta http-equiv="OnKey0x33" content="/receiving.htm"> <!-- 3 -->
<meta http-equiv="OnKey0x41" content="ida:IDA_PROGRAM_ABOUT"> <!-- A -->
</head>
```
<body>
Main Menu<br> 4. Cycle Count<br> 5. Inventory<br> 6. Receiving<br> Select an action:<br> Press ‘A’ to check version.<br> ...
<form name="form1">
<input type=text name="scan" value="" size=30><br>
<input type=button name="clearbutton" value="Clear Scanned Data"
onclick="javascript:document.form1.scan.value='';" >
</form>
</body>
<html>
<p>3.13 POWERON</p>
The PowerOn tag specifies an action that will occur when the handheld device resumes operation after a power suspend.

**Syntax**
```html
http-equiv="PowerOn"
content="javascript:PowerOnAction();"
<!- or ->
content="http://10.1.1.8/login.htm?mode=resume"
```

**Comments**
This tag is useful to set the browser to a consistent URL or state after a suspend/resume cycle. For example, a user authentication can be required to maintain security.

**Example**
```html
<html>
<head>
<meta http-equiv="PowerOn"
content="http://10.1.1.8/login.htm?mode=resume">
</head>
<body>
...
</body>
</html>
```
3.14 REBOOT
The Reboot tag will invoke a warm (soft) or cold (hard) reset of the device.
WARNING: Do not perform the cold reset unless you are prepared to lose all current settings, data, and add-on programs on the device.

Syntax

```html
http-equiv="Reboot"
content="Warm"
"Cold"
```

Comments
This tag is supported for compatibility with other browsers. It is preferable to use the IDA Action Code “IDA_WARMBOOT” via the IDA META tag, an “ida:” URL or a PostIDA call.

A warm reset will cause all un-saved work-in-progress to be lost. A cold boot will reset all RAM to factory original configurations. A cold boot will typically clear all network and device settings, user data files, and add-on programs that have not been placed in non-volatile memory.

The reset will occur as soon as the tag is parsed.

Example

```html
<html>
<head>
<meta http-equiv="Reboot" content="Warm">
</head>
<body>
This message should not be visible.
</body>
</html>
```

3.15 SCANNER
The Scanner tag is used to enable or disable the barcode scanner.

Syntax

```html
http-equiv="Scanner"
content="Enabled"
"Disabled"
```

Comments
This tag is used to enable or disable the scanner when a page is first loaded. You can also use the IDA codes IDA_SCAN_RESUME and IDA_SCAN_SUSPEND to change the state from an IDA META tag, an “ida:” URL or dynamically via a PostIDA call.

We do not support the “AutoEnter” or “AutoTab” content values. These values are un-needed because these and more complex scanner post-ambles can be configured within the Naurtech Web Browser. See the User’s Manual for more information about the scanner configuration.

Under Windows Mobile 2003, the Naurtech Web Browser will advance focus within native HTML text input objects when the Tab entered via a key or post-amble, without any extra handlers.

All other scanner configurations are maintained within the Web Browser configuration dialogs. These settings are session dependent and may be different for different Web browser or TE sessions.

Example
<html>
<head>
<meta http-equiv="Scanner" content="Enabled">
</head>
<body>
The scanner will work on this page.
<form name="form1">
<input type=text name="scan" value="" size=30><br>
...
</form>
</body>
</html>

3.16 SCANNERNAVIGATE
The ScannerNavigate tag directs scanner input to the specified JavaScript method or URL. This tag should be used whenever possible to provide the most robust control of scanner input.

Syntax
http-equiv="ScannerNavigate"
content="javascript:OnScan('%s', '%s', '%s', '%s', '%s');"
content="/scan.htm?data=%s&src=%s&type=%s&time=%s&length=%s"
There are two variants of the ScannerNavigate command for compatibility with other browsers. The first uses 5 parameters and the second uses 3 parameters. For the 5 parameter version, the “%s” are replaced with (1) barcode data, (2) source scanner name, (3) symbology type, (4) timestamp, and (5) barcode length. For the 3 parameter version, the “%s” are replaced with (1) barcode data, (2) symbology type, and (3) timestamp.

The variant type is determined automatically. If there are other that 3 or 5 substitution parameters, the ScannerNavigate tag is ignored.

Comments
Only one ScannerNavigate tag is permitted on a page. The ScannerNavigate tag should be used to ensure that the scanner data is inserted into the correct input element, or is submitted directly via the URL. The barcode data can be examined, validated and/or edited prior to use.

Example
<html>
<head>
<meta http-equiv="ScannerNavigate"
content="javascript:OnScan('%s', '%s', '%s', '%s', '%s');">"
</head>
<body>
<object id="CEBrowseX"
classid="clsid:D14943BD-4900-453E-8582-725F21A57E0C"
height=0, width=0></object>
<form name=form1>
Fill With First Scan<br>
<input type=text name="scan1" value="" size=30><br>
Fill With Second Scan<br>
<input type=text name="scan2" value="" size=30><br>
<script language=javascript>
function onscan(data, source, type, time, length)
{
    if (document.form1.scan1.value == "]
    {
        document.form1.scan1.value = data;
    }
    else if (document.form1.scan2.value == "]
    {
        document.form1.scan2.value = data;
    }
    else
    {
        document.CEBrowseX.CETerm.PostIda("IDA_VIBRATE_500", 0);
    }
</script>
</form>
</body>
</html>
alert("Form full, scan discarded.");
}
</script>
</form>
</body>
</html>

3.17 **SetDate**
The SetDate tag is used to set the current system date on the handheld.

**Syntax**

```html
http-equiv="SetDate"
content="mm/dd/yyyy"
"mm,dd,yyyy"
"mm-dd-yyyy"
```

Where `mm` is the month, `dd` is the day, and `yyyy` is the year. The date must be expressed in Coordinated Universal Time (UTC). You must use leading zeros for the day and month if less than 10.

**Comments**

If the date format is invalid, this tag is ignored.

**Example**

```html
<html>
<head>
<meta http-equiv="SetDate" content="04/15/2004">
</head>
<body>
...
</body>
</html>
```

3.18 **SetTime**
The SetTime tag is used to set the current system time on the handheld.

**Syntax**

```html
http-equiv="SetTime"
content="hh:mm"
"hh:mm"
```

Where `hh` is the hour, and `mm` is the minute. The time must be expressed in Coordinated Universal Time (UTC). You must use leading zeros if less than 10.
Comments
If the time format is invalid, this tag is ignored.

Example
<html>
<head>
<meta http-equiv="SetTime" content="12:01">
</head>
<body>
...
</body>
</html>

3.19 SIGNAL
The Signal tag is used to configure the on-screen WLAN (RF) signal strength meter. This meter overlays the HTML content and is updated at a specified interval. You can change the location and style of the meter. Available styles include a horizontal or vertical single bar meter and a stepped bar meter. The meter may be repositioned by a “touch and drag” stylus action if dragging is not disabled.

Syntax
http-equiv="Signal"
content="Show"
"Hide"
"Right"
"Left"
"Top"
"Bottom"
"x=n"
"y=m"
"Steps"
"AllowDrag"
"NoDrag"

Where x and y are the screen coordinates of the upper left corner of the meter icon. The screen coordinate (0,0) is in the upper left corner of the screen with x increasing to the right and y increasing downward. Right, Left, Top, and Bottom change the orientation of the meter and the placement of the WLAN icon. Steps enables the step style meter. AllowDrag will allow the user to drag the meter, whereas NoDrag will prevent dragging.

Comments
This tag is used to control the WLAN (RF) meter display. The meter may also be controlled within the CETerm configuration, independent of the Signal tag. The Signal tag will always override the internal configuration and will persist until changed by another Signal tag. Within the CETerm configuration you can specify the update interval and a notification when the strength falls below a designated level.

Example
<html>
<head>
<meta http-equiv="Signal" content="Show">
<meta http-equiv="Signal" content="Steps">
</head>
<body>
The RF meter should be visible.
...
</body>
</html>

3.20 SIGNALNAVIGATE
The SignalNavigate tag causes the specified JavaScript or URL to be invoked with WLAN signal information on a regular interval. The interval can be specified within a CETerm configuration dialog.

Syntax
http-equiv="SignalNavigate"
content="javascript:OnSignal('%s', '%s', '%s');"
<!- or ->
content="/signal.htm?strength=%s&ESSID=%s&MAC=%s"

Where the "%s" are replaced with (1) signal strength, (2) ESSID, and (3) the MAC address of the device.

Comments
This tag will work with or without a visible WLAN signal meter. The strength normally ranges from 0 to 100. The special strength value of -2 indicates that the device is not associated with any access point. The special strength value of -1 indicates that the strength cannot be determined. If you use a URL for the action, in most cases, the URL will navigate away from the current page rather than repeatedly calling a JavaScript method.
Example

```html
<html>
<head>
<meta http-equiv="SignalNavigate"
    content="javascript:OnSignal('%s', '%s', '%s');">  
</head>
<body>
<script language=javascript>
function OnSignal(strength, ESSID, MACAddress)  
{
    if (strength == -2)  
    {
        alert("RF radio is not associated.");
    }  
    else if (strength == -1)  
    {
        alert("Unable to determine RF signal strength.");
    }  
    else  
    {
        alert("RF Signal strength = " + strength);
    }  
}
</script>
...
</body>
</html>
```

### 3.21 SIP

The SIP tag controls the visibility of the Soft Input Panel (SIP).

**Syntax**

```html
http-equiv="SIP"
    content="Show"
    "Hide"
    "Locked"
```

**Comments**

This tag is used primarily when the SIP should be displayed upon loading a page. Alternatively, you can use an IDA Action Code via the IDA META tag, an “ida:” URL or a PostIDA call. These latter techniques can be used to change the SIP visibility during the user interaction with the page. The “Locked” value will prevent the SIP from popping up when focus is set to a text input element under Pocket PC 2002 or Windows Mobile 2003. The locked state can also be set in the CETerm configuration dialogs.
3.22 SIPUp
The SIPUp tag makes the Soft Input Panel (SIP) visible.

Syntax
http-equiv="SIPUp"
content=""

Comments
This tag is supported for compatibility with other browsers. It is preferable to use the SIP META tag or the IDA Action Code “IDA_SIP_SHOW” via the IDA META tag, an “ida:” URL or a PostIDA call. The content is ignored, but it must be present.

Example
<html>
<head>
<meta http-equiv="SIPUp" content="" />
</head>
<body>
The Soft Input Panel (SIP) should be visible.
</body>
</html>

3.23 TextSize
The TextSize tag is used to set an overall text zoom level. The zoom level can also be changed manually with font sizing buttons on the Toolbar or KeyBar.

Syntax
http-equiv="TextSize"
content="Smallest"
"Smaller"
"Medium"
"Larger"
"Largest"

**Comments**
The TextSize tag is somewhat misnamed. Relative text sizes are determined by the HTML content. TextSize applies an overall zoom factor to the sizes set in the HTML. The initial zoom value will be the last value set for the session whether by a TextSize tag or by manual actions.

**Example**
```html
<html>
<head>
<meta http-equiv="TextSize" content="Largest">
</head>
<body>
This is the Largest TextSize zoom.
...
</body>
</html>
```

### 3.24 TimerInterval

The TimerInterval tag is used to specify the interval between activations of the TimerNavigate action.

**Syntax**
```
http-equiv="TimerInterval"
content="milliseconds"
```

Where milliseconds is the number of milliseconds between activations.

**Comments**
The effect of TimerInterval and TimerNavigate tags can be accomplished using the Javascript `setInterval()` and `setTimeout()` methods on the window DOM object. Using the Javascript methods is recommended.

**Example**
See example under TimerNavigate.

### 3.25 TimerNavigate

The TimerNavigate tag causes the specified JavaScript or URL to be invoked on a regular interval. The interval is specified with the TimerInterval tag.
Syntax

http-equiv="TimerNavigate"
content="javascript:OnTimer('%s');"
<!- or ->
content="/timer.htm?time=%s"

Where the "%s" is replaced with the current time in the form hh:mm:ss.

Comments

The effect of TimerInterval and TimerNavigate tags can be accomplished using the Javascript setInterval() and setTimeout() methods on the window DOM object. Using the Javascript methods is recommended.

Example

<html>
<head>
<meta http-equiv="TimerInterval" content="1000">
<meta http-equiv="TimerNavigate"
content="javascript:OnTimer('%s');">
</head>
<body>
Current time: <div id="timerDiv">junk</div>
<script language=javascript>
function onTimer(time)
{
    timerDiv.innerHTML = time;
}
</script>
... 
</body>
</html>

3.26 ZEBRALABEL_COMPLETE OR PLSeriesLabel_Complete

The ZebraLabel_Complete tag is used to report the status of a print from the ZebraLabel_Print tag. The alternative identifier PLSeriesLabel_Complete will also be recognized.

Syntax

http-equiv="ZebraLabel_Complete"
content="javascript:PrintStatus('%ld');"
<!- or ->
content="http://10.1.1.8/print.htm?status=%ld"
Comments
This tag is supported for compatibility with other browsers. It is preferable to use other techniques for sending print content to a printer. See the advanced topic “Printing from HTML”.

The status code of the print action will be substituted into the “%ld” location. A status of 0 indicates success, 1 indicates failure. See ZebraLabel_Print for more details.

Example
<html>
<head>
<meta http-equiv="ZebraLabel_Complete"
      content= "http://10.1.1.8/print.htm?status=%ld"
      <meta http-equiv="ZebraLabel_Print"
             content="! 0 200 200 581 1\r\nLABEL ... PRINT\r\n">
</head>
<body>
The print should be produced.
This message should not be visible.
</body>
</html>

3.27 ZEBRALABEL_PRINT OR PLSERIESLABEL_PRINT
The ZebraLabel_Print tag contains data that is sent to a Zebra printer. The ZebraLabel_Complete tag must also be present to report the status of the print. The alternative identifier PLSeriesLabel_Print will also be recognized.

Syntax
http-equiv="ZebraLabel_Print"
content="! 0 200 200 581 1\r\nLABEL ... PRINT\r\n">

Comments
This tag is supported for compatibility with other browsers. It is preferable to use other techniques for sending print content to a printer. See the advanced topic “Printing from HTML”.

The contents must be less than 1024 characters. Any “\r” sequences are replaced by the CR (0x0d) character. Any “\n” sequences are replaced by the NL
(0x0a) character. The sequence “\xXX” is replaced by a byte with the hexadecimal value XX. Use “\" for a literal backslash.

The content is sent to the printer that is currently configured under the CETerm Printer tab. This may be a serial attached printer, Bluetooth, or network accessible printer.

Example
```
<html>
<head>
<!- NOTE – The content must be on one line ->
<!- NOTE – It is split here for readability ->
<meta http-equiv=" ZebraLabel_Print "
    content="! 100 200 200 1225 1 \r
    TEXT 0 2 1 0 Vendor\n    TEXT 5 2 1 20 Adidas®\n    TEXT 5 2 1 135 Storm\n    TEXT 0 1 225 Cross-Training\n    TEXT 5 0 1 250 Arch Support\n    TEXT 5 0 1 275 Leather Upper\n    TEXT 5 0 1 300 Cushioned Insole\n    TEXT 4 1 100 435 $61.99\n    TEXT 0 2 1 550 ______________\n    TEXT180 4 1 250 700 $61.99\n    TEXT180 5 0 270 740 000000292818\n    BARCODE UPCA 1 1 50 100 750 000000292818\n    TEXT180 5 0 320 910 WHITE/RED\n    TEXT180 5 0 320 935 Available in:\n    TEXT180 0 2 320 1025 ______________\n    JOURNAL\nPRINT\r">
<meta http-equiv="ZebraLabel_Complete"
    content= "http://10.1.1.8/print.htm?status=%ld"
</head>
<body>
The print should be produced.
This message should not be visible.
</body>
</html>
```
4.0 Advanced Topics

In this chapter, we discuss several common, but advanced, topics for creating robust Web based applications. We discuss how to overcome some limitations of the Pocket IE browser behavior.

4.1 Navigating to Pre-configured URLs

The Naurtech Web Browser allows you to configure pre-defined URL's which can be associated with a hardware key, a KeyBar button, or activated from the within the browser via a link or via JavaScript.

The URL is defined within a User Text string and this text can be submitted for navigation by any of the ways listed above. For example, with the following content in User Text 1:

\IDA_URL\file:///Program Files/myhelppage.htm\r

You can navigate to this local help page by activating the IDA code IDA_USTRING_1 which has the friendly name “Text 1”. You can associate a hardware key or a KeyBar button to this IDA code. Alternatively, you could use any of the ways listed in the IDA Action Codes section to trigger the navigation. For example the following link will send you to the help page:

<a href="ida:IDA_USTRING_1">Show Help Page</a>

Be sure to provide a way to navigate back to the application from your help page. Note the required “\IDA_URL\” at the beginning of the URL and the “\r” at the end. If your URL contains a literal backslash, it must be escaped with a second backslash “\".

It is possible to change the contents of User Text from within JavaScript with the “SetProperty” method of the CEBrowseX or “external” objects. Thus, this storage can be used to maintain some persistent, device-local, information.

4.2 Controlling the Scanner

We have shown how the Scanner and ScannerNavigate META tag identifiers can be used to control the scanner. You can also use IDA codes to provide additional control.

The scanner can be enabled or disabled via an IDA code. You can also activate a “soft trigger” on most scanner equipped handhelds. This will start the scanner without the user needing to hold the trigger. This mode is useful if the application is driving the scanner in a “read loop” until a desired number of scans are completed.
<meta http-equiv="Scanner" content="Enabled">
<meta http-equiv="ScannerNavigate"

ccontent="Javascript:onscan('%s','%s','%s','%s','%s');">...
<object id="CEBrowseX"
classid="clsid:D14943BD-4900-453E-8582-725F21A57E0C"
height=0, width=0>
</object>
<input type=button name="autoscan" value="Auto Scan"
onclick="autoscan();">...
<script language=javascript>
function autoscan()
{
    if (autoscantrigger)
    {
        autoscantrigger = 0;
    }
    else
    {
        autoscantrigger = 1;
        document.CEBrowseX.CETerm.PostIda( "IDA_SCAN_TRIGGER", 0 );
    }
}

function onscan(data, source, type, time, length)
{
    if (document.form1.scan1.value == ")
    {
        document.form1.scan1.value = data;
    }
    else if (document.form1.scan2.value == ")
    {
        document.form1.scan2.value = data;
    }
    else if (document.form1.scan3.value == ")
    {
        document.form1.scan3.value = data;
        autoscantrigger = 0;
    }
    else
    {
        autoscantrigger = 0;
        document.CEBrowseX.CETerm.PostIda( "IDA_VIBRATE_500", 0 );
        alert("Form full, scan discarded.");
    }

    if (autoscantrigger)
    {
        document.CEBrowseX.CETerm.PostIda( "IDA_SCAN_TRIGGER", 0 );
    }
}
4.3 INPUT FOCUS AND THE TAB KEY

One of the first limitations you are likely to notice about Pocket Internet Explorer is that the Tab key does not always move the input focus as you expect from using the desktop Internet Explorer. Pocket IE on the Pocket PC 2002 and Windows Mobile platforms has the most limited behavior.

The Naurtech Web Browser provides enhanced focus control on the Pocket PC platforms. If you only need to move between native HTML text input elements, then the Tab key should perform as you expect from the desktop. The Tab key should also work as expected on Windows CE .NET or 5.0 platforms. However, currently a Tab post-amble will not advance the focus on all platforms.

Some other vendors provide a text input control like Naurtech’s TextX to manage the focus under Pocket PC 2002 or Windows Mobile. TextX will perform similarly, but we recommend you avoid using ActiveX controls for input to simplify and accelerate development. If you must use the ActiveX text input, then you can provide additional focus management.

The following example shows how to track the focus and advance the focus when the Tab key is pressed. These techniques will work for the native HTML text input elements on Windows CE platforms or with the ActiveX text input on Pocket PC 2002 and Windows Mobile. Both support the OnKeyPress event handlers.

This first example uses native HTML text input on Windows CE .NET. Note the use of “document.activeElement” in this example:

```
<html>
<head>
<title>Naurtech Tab Demo Page</title>
<meta http-equiv="Scanner" content="Enabled">
<meta http-equiv="ScannerNavigate" content="Javascript:onscan('%s','%s','%s','%s','%s');">
<meta http-equiv="OnKey0x0D" content="Javascript:document.form1.clear.click();">
</head>
<body scroll=no onload="Javascript:document.form1.scan1.focus();">
<form name=form1>
<center>
<font size=+2>
Naurtech Tab Demo<br>
</font>
```
<br>
Focus starts in first input
<input type=text name="scan1" value="" size=30 onkeypress="myonkey();"/>
<input type=text name="scan2" value="" size=30 onkeypress="myonkey();"/>
<input type=text name="scan3" value="" size=30 onkeypress="myonkey();"/>
<input type=button name="clear" value="Clear Data" onclick="myclear();" onkeypress="myonkey();">
</center>
</form>

<script for="document" event=onkeypress>
// IMPORTANT: This handler is used when focus is not already in an input object
myonkey();
</script>

<script language=javascript>
// Handle the key event
function myonkey()
{
  if (window.event.keyCode == 9) // look for tab key
  {
    nextfield( document.activeElement );

    window.event.cancelBubble = true;
  }
}

// Move from the current field to the next field
function nextfield( current )
{
  if (current == document.form1.scan1)
  {
    document.form1.scan2.focus();
  }
  else if (current == document.form1.scan2)
  {
    document.form1.scan3.focus();
  }
  else
  {
    document.form1.scan1.focus();
  }
}

// Clear the fields
function myclear()
{
  document.form1.scan1.value = "";
}
Here is a similar Tab navigation example using the TextX input object. This example does not use the ScannerNavigate tag. A Tab in the scanner postamble "\t" will advance the focus. However, a scan when focus is not in an input field will be lost. Note the use of hidden fields to pass the TextX contents back to the host.

```html
<html>
<head>
<title>Naurtech TextX</title>
<meta http-equiv="Scanner" content="Enabled">
</head>
<body onload="javascript:document.myform1.textx1.SetFocus(1);">
<form name=myform1
```
Naurtech TextX Tab Test
(Windows Mobile 2003)

TextX1 object id="textx1"
classid="clsid:6402E27B-CD4F-448C-BAEA-F3558242459D"
height=22, width=150>
-- Please check browser configuration --</span>
</object>

TextX2 object id="textx2"
classid="clsid:6402E27B-CD4F-448C-BAEA-F3558242459D"
height=22, width=150>

<input type=button name="submitbutton" value="Submit"
OnClick="mysubmit()" />
<input type=hidden name="hiddentextx1">
<input type=hidden name="hiddentextx2">

<script language=javascript for="textx1" event="OnKeyPress(key)"/>
return MyKeyPress(key, document.myform1.textx2);
</script>

<script language=javascript for="textx2" event="OnKeyPress(key)"/>
return MyKeyPress(key, document.myform1.textx1);
</script>

<script language=javascript>
function mysubmit()
{

// Put values of TextX into hidden input fields for host
// Otherwise, host won’t see control contents
document.myform1.hiddentextx1.value = document.myform1.textx1.value;
document.myform1.hiddentextx2.value = document.myform1.textx2.value;

// Submit the form
document.myform1.submit();
}

function MyKeyPress(key, control)
{
if (key == 9)
{
// Advance the focus
control.SetFocus(1);
4.4 Session Launcher

One interesting use of the Web Browser is as a launcher for other browser or terminal emulation sessions. The following example shows how a static web page on the device could be used to activate other sessions. This example assumes that this web page is displayed under Session 4. Each of the other sessions must be configured for the desired activity:

```html
<html>
<head>
<title>Naurtech Launch Page</title>
<meta http-equiv="PowerOn" content="Javascript:poweron();">  
<meta http-equiv="Scanner" content="Disabled">  
<meta http-equiv="OnKey0x70" content="Javascript:startsession(1);"/><! F1>
<meta http-equiv="OnKey0x71" content="Javascript:startsession(2);"/><! F2>
<meta http-equiv="OnKey0x72" content="Javascript:startsession(3);"/><! F3>
</head>
<body>
<object id="CEBrowseX"
  classid="clsid:D14943BD-4900-453E-8582-725F21A57E0C"
  height=0, width=0>
</object>
<form name=form1>
  <font size=9>
    <center>
    Main Menu<br>
    <a href="javascript:startsession(1);">F1. Pick</a><br>
    <a href="javascript:startsession(2);">F2. Cycle Count</a><br>
    <a href="javascript:startsession(3);">F3. Receive</a><br>
    </font>
    </center>
  <script language=javascript>
    function poweron()
    {
      // Navigate to Main Menu on resume
    }
  </script>
</form>
</body>
</html>
```
sendida( "IDA_SESSION_S4", 0 );
}

function startsession(id)
{
    if (id >= 1 && id <=4)
    {
        // Switch to session
        sendida("IDA_SESSION_S" + id, 0);
        // Connect if not connected
        sendida("IDA_SESSION_CONNECT", id);
        return;
    }
    else
    {
        alert("Unsupported session.");
    }
}

function sendida(ida, s)
{
    // PocketPC 2002
    //CEBrowseX.CETerm.SendIDA(ida, s);
    // Windows Mobile
    document.CEBrowseX.CETerm.SendIDA(ida, s);
}
</script>
</body>
</html>

4.5 HOW TO IDENTIFY THE CURRENT BROWSER
The Naurtech Web Browser returns the same User-Agent HTTP value as does the standard Windows Pocket IE browser. This is necessary to indicate the fundamental capabilities of the browser to the Web server. There are other ways for the Web application to determine if the Naurtech Web Browser is the client.

You can use the GetUnitInformation META tag identifier to retrieve the version number and send this to the host. As of this writing, the current version number is “5.1.0”. Most other browsers will not support the GetUnitInformation feature and your HTML must be prepared to return an empty value.

<html>
<head>
<meta http-equiv="GetUnitInformation"
    content="Javascript:saveunitinfo('%s','%s','%s');">>
<script language=javascript>
var serial;
var ceuuid;
var version;

function saveunitinfo(serialarg, ceuuidarg, versionarg)
{
    serial = serialarg;
    ceuuid = ceuuidarg;
    version = versionarg;
}
</script>
</head>
<body onload="javascript:checkversion();">
....check version variable and return info to host....
</body>

Alternatively, you can use a CEBrowseX or “external” object to retrieve the version in a CETerm.GetProperty call. Other browsers will be able to create a CEBrowseX object if it is installed, but CETerm.GetProperty will not return a valid result.

<object id="CEBrowseX"
classid="clsid:D14943BD-4900-453E-8582-725F21A57E0C"
height=0, width=0>
</object>
...
<script language=javascript>
function checkversion()
{
    version = document.CEBrowseX.CETerm.GetProperty( "app.version" );
    // check the version value, return info to host
}

4.6 DEVICE INFORMATION
You can retrieve information about the device configuration using the CETerm.GetProperty method. See the CEBrowseX section for details

4.7 SYMBOL WEB CLIENT
An early browser developed for Symbol devices running the Palm OS had some extensions for controlling the scanner and printing. These extensions were implemented through custom attributes on standard HTML tags and custom HTML tags. The Naurtech Web Browser running on Windows CE .NET or CE 5.0 platforms supports these extensions. These extensions are not supported on Pocket PC 2002 or Windows Mobile based devices. These extensions are supported for compatibility with existing Web based applications. New Web applications should use the META tags.
On the text INPUT element we support the “stiscan” and “stisubmit” attributes. If present, the stiscan attribute controls which symbologies may be inserted into the input element. If no value is specified, all symbologies are allowed. If an empty string is specified, no symbologies are allowed. When the “stisubmit” attribute is present, the enclosing form will be submitted after scanned data is inserted in the element.

```xml
<input type=text name=upc size=12 maxlength=16 stiscan="ABCDEFGHIJKLMNOPQRSTUVWXYZ" stisubmit>
```

The custom element BEEP is supported to sound a tone:

```xml
<NAURTECH:beep frequency="f" repeat="n" duration="d">
```

where f is in Hz, n is a count, and d is in milliseconds. The handheld device must support a tone generator. The “NAURTECH:” namespace label is required.

The custom element PRINT is supported to send print content to the printer.

```xml
<NAURTECH:print name="printer" type="N" announce="yes" retries=3>
Print content here
</print>
```

where “printer” is the destination name of the printer, and type is the connection type. The “NAURTECH:” namespace label is required. The following connection types are supported:

<table>
<thead>
<tr>
<th>Connection Type</th>
<th>Name Values</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q</td>
<td>Name is the Windows print queue name</td>
<td>Print to Windows print queue</td>
</tr>
<tr>
<td>N</td>
<td>Printer IP address or hostname and port. For example, “192.168.1.101:6101”</td>
<td>Print directly to port</td>
</tr>
<tr>
<td>dddd</td>
<td>Same as ‘N’ but port specified as dddd.</td>
<td>Print directly to port</td>
</tr>
<tr>
<td>I, B, S, (other)</td>
<td>(ignored)</td>
<td>Print to configured printer for current session.</td>
</tr>
</tbody>
</table>

Standard escape characters such as \r, \n, and \xXX” will be substituted in the print contents. Literal line breaks in the content are ignored, you must use \n if the printer language requires a line break.
You can set the values of the currently configured network printer using the CETerm.SetProperty method on the CEBrowseX or “external” objects.
5.0 Printing from HTML

There are numerous ways to print from the Naurtech Web Browser. Once a printer is configured, print content may be specified via a special META tag, the CEBrowseX or "external" objects, the custom PRINT tag, or with any ActiveX control designed to print from a browser.

The Naurtech Web Browser maintains the printer configuration within the Session configuration dialogs. Here you may specify a serial attached, Bluetooth, IrDA, or network attached printer. We support both Windows Print Queues and direct-to-port printing for network printers. See the CETerm User's Manual for more details on configuring a printer.

All of the techniques for specifying print content allow common escape sequences to be embedded which will be converted to non-printable characters. These include the carriage return (CR – “\r”), linefeed (LF – “\n”), and general hexadecimal bytes (0xXX).

5.1 PRINTING WITH A META TAG

See the “ZebraLabel_Print” and “ZebraLabel_Complete” identifiers in the META tag section for details on initiating a print from a META tag.

5.2 PRINTSTRING AND PRINT METHODS

The PrintString method on the CEBrowseX control and the Print method on the "external" object behave similarly. In both cases, a string is constructed and sent to the current printer.

```html
<a href="Javascript:myprint();">Test External Print</a>
<script language=javascript>
function myprint()
{
    external.Print(
    "! 100 200 200 1225 1\r\n"
    +
    "TEXT 0 2 1 0 Vendor\r\n"
    +
    ...
    "JOURNAL\r\n"
    +
    "PRINT\r\n"
    );
}
</script>
```

5.3 NAURTECH:PRINT Tag

The custom PRINT tag is supported for compatibility with other browsers. See the Symbol Web Client section for a discussion of this tag.
5.4 **ACTIVEX PRINTING CONTROLS**

There are numerous ActiveX controls available to send print content to a printer. These are available from handheld manufacturers, printer manufacturers, and third-party sources. All controls which follow accepted standards will work within the Naurtech Browser. Currently we do not recommend any specific control.

6.0 CEBrowseX Control

The Naurtech CEBrowseX control is used to send commands to the Naurtech Web Browser and to access device and application information. This control is installed when you install the Naurtech Web Browser.

On Windows CE .NET or CE 5.0 based devices, most of the CEBrowseX methods are available from the Document Object Model (DOM) “external” object. In this case you do not need to instantiate a CEBrowseX object.

**ALERT:** The CEBrowseX control has been updated in CETerm V5.1 to provide access to the new “CETerm” and “OS” automation objects used in the CETerm script engine. The new “CETerm” object provides all the features that were previously available from CEBrowseX. The CEBrowseX object should now only be used to access the CETerm or OS objects. Existing web pages will continue to work but should be converted to the new syntax as they are updated.

**ALERT:** See the CETerm Scripting Guide for full details on the CETerm and OS objects and their methods and properties. Scripting in CETerm offers additional capabilities to enhance your web based applications.

**SYNTAX**

```xml
<object id="CEBrowseX"
classid="clsid:D14943BD-4900-453E-8582-725F21A57E0C"
height=0, width=0>
```

The CEBrowseX control has no visible display, but you must specify height and width of zero to prevent the object from consuming space on the page.
**CLASSID**

The CEBrowseX CLASSID is:

```
CLASSID="clsid:D14943BD-4900-453E-8582-725F21A57E0C"
```

**METHODS**

The following methods are available. Note that most are marked as deprecated (DEP) in favor of using the CETerm or OS object and their associated methods. The CETerm and OS objects are obtained from the corresponding CEBrowseX properties.

<table>
<thead>
<tr>
<th>Method</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>PostIDA( IDACode, session )</td>
<td>Send a command to a session (asynchronous) (DEP)</td>
</tr>
<tr>
<td>SendIDA( IDACode, session )</td>
<td>Send a command to a session (synchronous) (DEP)</td>
</tr>
<tr>
<td>SendText( session )</td>
<td>Send text content to a session (DEP)</td>
</tr>
<tr>
<td>GetProperty</td>
<td>Get a property value from the Web Browser (DEP)</td>
</tr>
<tr>
<td>SetProperty</td>
<td>Set a property value in the Web Browser (DEP)</td>
</tr>
<tr>
<td>PlaySound( session )</td>
<td>Play a tone or wave file on the device (CEBrowseX only) (DEP)</td>
</tr>
<tr>
<td>PlayTone( session )</td>
<td>Play a tone on the device (external only) (DEP)</td>
</tr>
<tr>
<td>PrintString</td>
<td>Send content to a printer (CEBrowseX only)</td>
</tr>
<tr>
<td>Print</td>
<td>Send content to a printer (external only)</td>
</tr>
</tbody>
</table>

*PostIDA( IDACode, session )*

PostIDA sends an IDA action command to the Web Browser and directs it to the specified session. Valid session values are 1-4. The special session value of 0 will send the command to the current session. See the Appendix for IDA values.

The PostIDA command will return before the Web Browser acts on the command. We recommend using PostIDA rather than SendIDA. There are only rare situations when SendIDA must be used.

*SendIDA( IDACode, session )*

SendIDA sends an IDA action command to the Web Browser and directs it to the specified session. Valid session values are 1-4. The special session value of 0 will send the command to the current session. See the Appendix for IDA values.
The SendIDA method will attempt to complete the action before returning. We recommend using PostIDA rather than SendIDA. There are only rare situations when SendIDA must be used.

\[\text{SendText}(\text{text, session})\]
SendText sends a text string to the Web Browser and directs it to the specified session. Valid session values are 1-4. The special session value of 0 will send the command to the current session.

The text string may include IDA symbolic names between backslash characters ‘\’. The IDA codes will be interpolated as the text is sent.

\[\text{Value} = \text{GetProperty}(\text{propertyName})\]
GetProperty will return the named property value. This may be a device property, application property, or session property. See the Appendix for a list of available properties.

\[\text{Status} = \text{SetProperty}(\text{propertyName, propertyValue})\]
SetProperty will assign the given value to the named property. See the Appendix for a list of available properties.

\[\text{PlaySound}(\text{sound})\]
PlaySound will play a tone or wave file depending on the specified sound. Any wave file on the device may be specified. Use the complete file path if the file is not in the \Windows directory. PlaySound is not available on the “external” object (see PlayTone).

If the handheld device contains a programmable tone generator, the sound parameter may also be a string which defines a sequence of tones to play. The syntax is given below:

“vvfffddd” – where
vv – is the volume 01-10
fff – is the frequency in 10’s of MHz, 000-999
ddd – is the duration in 10’s of millisec, 000-999
PlayTone( volume, frequency, duration )

PlayTone will play a tone if supported by the handheld hardware. PlayTone is not available on the CEBrowseX object (see PlaySound).

volume – is the volume 00 -10  (0 is off, 10 is loudest)
frequency – is the frequency in Hz.
duration – is the duration in millisec.

Status = PrintString( printData ) or Status = Print( printData)

PrintString and Print will send the printData to the currently configured printer. PrintString is only available on the CEBrowseX object. Print is only available on the “external” object.

The printData string may contain escape characters for CR (\r), Newline (\n), and hexadecimal bytes (\xXX).

**PROPERTIES**

The CEBrowseX properties are used to return the CETerm or OS objects. Both are read-only.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CETerm</td>
<td>CETerm automation object</td>
</tr>
<tr>
<td>OS</td>
<td>OS automation object</td>
</tr>
</tbody>
</table>

**EVENTHANDLERS**

The CEBrowseX control has no event handlers.
7.0 TextX Control

The Naurtech TextX control is a replacement for the HTML text input element on Pocket PC 2002 and Windows Mobile 2003 devices. The native HTML text input element does not support most events and many properties. Using the TextX object gives much greater control over user input. However, there are drawbacks to using the control, such as increased HTML complexity and no desktop equivalent to simplify development. This control is not installed when you install the Naurtech Web Browser. Please contact Naurtech Support if you need this control.

You may create as many instances of this control, each as a separate input element, as you need on a page. Each instance must have a unique ID.

Some other browsers must use an ActiveX control for text input because they are not able to prevent the auto-popup of the SIP when focus is placed in a native HTML text input element. The Naurtech Web Browser controls the SIP behavior independently and thus does not require the TextX control for this purpose. Also, the Naurtech Web Browser implements Tab key behavior between native HTML text input elements, eliminating another primary reason to use TextX. The third most common reason to use and ActiveX text object is to submit a form when the Enter key is pressed or received as a scanner post-amble. We recommend using the ScannerNavigate META tag for robust scanner input but we also support the OnKey META tag within native HTML text input elements. This eliminates the third common reason for the TextX control.

There is no need to use the TextX control on Windows CE .NET platforms because the native HTML text input element has equivalent features.

In summary, we recommend against using the TextX control, but we provide it for those situations where the features are required.

Syntax

```xml
<OBJECT ID="textx"
CLASSID="clsid:6402E27B-CD4F-448C-BAEA-F3558242459D"
HEIGHT=20,
WIDTH=200>
<PARAM NAME=ALIGNMENT VALUE="left">
<PARAM NAME=AUTOHSCROLL VALUE="true">
<PARAM NAME=AUTOVSCROLL VALUE="false">
<PARAM NAME=BORDER VALUE="true">
<PARAM NAME=FONTBOLD VALUE="false">
<PARAM NAME=FONTPITCH VALUE="false">
<PARAM NAME=FONTITALIC VALUE="false">
<PARAM NAME=FONTNAME VALUE="Tahoma">
<PARAM NAME=FONTSIZE VALUE=10 >
```
CLASSID
The TextX CLASSID is:
CLASSID="clsid:6402E27B-CD4F-448C-BAEA-F3558242459D"

METHODS
Two methods are available on the TextX object.
**objectname.SetFocus( select )**
SetFocus will set the current focus to this TextX object. If select is true, then all of the text in the control will be selected.

**objectname.ShowSIP( visible )**
ShowSIP will cause the SIP to be visible if the argument is true. It will hide the SIP if the argument is false. This may also be accomplished via an IDA code (see IDA Action Codes).

**PROPERTIES**
The TextX control has several properties to control the appearance and behavior of the object. All properties may be set in an initial “PARAM” statement or set dynamically via JavaScript, unless described otherwise.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Values</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alignment</td>
<td>Text justification</td>
<td>LEFT, RIGHT, CENTER</td>
<td>LEFT</td>
</tr>
<tr>
<td>AutoHScroll</td>
<td>Text will autoscroll horizontally</td>
<td>TRUE, FALSE</td>
<td>TRUE</td>
</tr>
<tr>
<td>AutoVScroll</td>
<td>Text will autoscroll vertically on Enter</td>
<td>TRUE, FALSE</td>
<td>FALSE</td>
</tr>
<tr>
<td>Border</td>
<td>Show border lines</td>
<td>TRUE, FALSE</td>
<td>TRUE</td>
</tr>
<tr>
<td>FontBold</td>
<td>Use bold font</td>
<td>TRUE, FALSE</td>
<td>FALSE</td>
</tr>
<tr>
<td>FontFixedPitch</td>
<td>Use fixed pitch font</td>
<td>TRUE, FALSE</td>
<td>FALSE</td>
</tr>
<tr>
<td>FontItalic</td>
<td>Use italic font</td>
<td>TRUE, FALSE</td>
<td>FALSE</td>
</tr>
<tr>
<td>FontName</td>
<td>Font for object</td>
<td>(any on device)</td>
<td>System</td>
</tr>
<tr>
<td>FontSize</td>
<td>Size of font</td>
<td>6-28</td>
<td>10</td>
</tr>
<tr>
<td>FontUnderline</td>
<td>Use underlined font</td>
<td>TRUE, FALSE</td>
<td>FALSE</td>
</tr>
<tr>
<td>FontWeight</td>
<td>Weight of font</td>
<td>300=light 400=normal 700=bold 900=heavy</td>
<td>400</td>
</tr>
<tr>
<td>MaxLength</td>
<td>Maximum text length</td>
<td>0-8192 (0 – no limit)</td>
<td>0</td>
</tr>
<tr>
<td>NumberOnly</td>
<td>Only accept digits</td>
<td>TRUE, FALSE</td>
<td>FALSE</td>
</tr>
<tr>
<td>Password</td>
<td>Display text as *’s</td>
<td>TRUE, FALSE</td>
<td>FALSE</td>
</tr>
<tr>
<td>Value</td>
<td>Text content</td>
<td>(any text)</td>
<td>(empty)</td>
</tr>
<tr>
<td>WantReturn</td>
<td>Enter gives a newline in Multiline control</td>
<td>TRUE, FALSE</td>
<td>FALSE</td>
</tr>
</tbody>
</table>
Some font properties may conflict or may not yield visible differences on all devices. FontBold is equivalent to FontWeight=700. Specifying a font name will override some other attributes.

To create a multi-line control, simply use a large HEIGHT. Multiline content with embedded “\r\n” cannot be set via a PARAM element. You can set such content dynamically in JavaScript via the Value property.

More than MaxLength characters can be inserted in the text box via the Value property.

**EVENTHANDLERS**

The TextX control supports event handlers which are invoked when the user interacts with the control. Only OnKeyPress returns a value. This value can be set to cancel (or ignore) the event. The return values of all other handlers are ignored and the associated events cannot be canceled.

```c
void OnChange()
```

The OnChange handler is called when the contents of the control have been changed and focus is lost from the control.

```c
void OnClick( int x, int y )
```

The OnClick handler is called when the user taps the stylus on the control. The x and y coordinates of the tap are provided in the arguments.

```c
void OnFocus()
```

The OnFocus handler is called when the control receives input focus.

```c
void OnLostFocus()
```

The OnLostFocus handler is called when the control loses input focus.

```c
void OnKeyDown( int vkey )
```

The OnKeyDown handler is called when the user presses a key with focus in the control. The vkey argument is the Windows VK Code for the key. This handler
may also be called when keys are simulated within the Web Browser via KeyBar buttons or remapped physical keys.

int OnKeyPress( int akey )
The OnKeyPress handler is called when a key is pressed which generates an ASCII character. The akey argument contains the ASCII character which is generated. The return value from this handler controls the event propagation. A return value of 0 will continue the event, a value of -1 will cancel the event. Any positive value will replace the akey in this event. This handler is called before the character is placed in the input object, so changing the return value will change the character placed in the object.

void OnKeyUp( int vkey )
The OnKeyUp handler is called when the user releases a key with focus in the control. The vkey argument is the Windows VK Code for the key. This handler is not called when keys are simulated from within the Web Browser.
Appendix 1 - Properties

The properties listed in this appendix may be accessed via the GetProperty and SetProperty methods on the CEBrowseX ActiveX control or via the DOM “external” object under Windows CE .NET or 5.0. Properties marked (RO) are read-only and may not be set with SetProperty.

ALERT: Use the CETerm object methods GetProperty and SetProperty to access properties. There are many additional properties described in the CETerm Scripting Guide. The following list is included for older applications that used the deprecated GetProperty and SetProperty methods of the CEBrowseX control.

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Application Properties</strong></td>
<td></td>
</tr>
<tr>
<td>app.buildid (RO)</td>
<td>Program build identifier</td>
</tr>
<tr>
<td>app.name (RO)</td>
<td>Program name</td>
</tr>
<tr>
<td>app.session.active (RO)</td>
<td>Currently active session</td>
</tr>
<tr>
<td>app.usertext.NN</td>
<td>User text # NN contents, NN is 1-64</td>
</tr>
<tr>
<td>app.version (RO)</td>
<td>Program version</td>
</tr>
<tr>
<td><strong>Device Properties</strong></td>
<td></td>
</tr>
<tr>
<td>device.batterystatus (RO)</td>
<td>Current battery status</td>
</tr>
<tr>
<td>device.battery.statustext (RO)</td>
<td>Current battery status string</td>
</tr>
<tr>
<td>device.battery.status (RO)</td>
<td>Current battery status</td>
</tr>
<tr>
<td></td>
<td>-1 – unknown, 0 – critical, 1 – warning,</td>
</tr>
<tr>
<td></td>
<td>2 – low, 3 – medium, 4 – high, 5 - charging</td>
</tr>
<tr>
<td>device.battery.level (RO)</td>
<td>Current battery strength - 0 – 100</td>
</tr>
<tr>
<td></td>
<td>-1 – unknown</td>
</tr>
<tr>
<td>device.deviceid (RO)</td>
<td>Device ID string</td>
</tr>
<tr>
<td>device.ipaddress (RO)</td>
<td>IP Address of handheld</td>
</tr>
<tr>
<td>device.macaddress (RO)</td>
<td>MAC Address of handheld</td>
</tr>
<tr>
<td>device.platformid (RO)</td>
<td>Windows CE Platform ID</td>
</tr>
<tr>
<td>device.presetid (RO)</td>
<td>Windows CE Preset ID</td>
</tr>
<tr>
<td>device.rf.strength (RO)</td>
<td>RF signal strength 0-100,</td>
</tr>
<tr>
<td></td>
<td>-2 – not associated with AP,</td>
</tr>
<tr>
<td></td>
<td>-1 – unknown</td>
</tr>
<tr>
<td>device.rf.status (RO)</td>
<td>RF status</td>
</tr>
<tr>
<td></td>
<td>-1 – unknown, 0 – unassociated, 1 – poor,</td>
</tr>
<tr>
<td></td>
<td>2 – fair, 3 – good, 4 – very good, 5 – excellent</td>
</tr>
<tr>
<td>device.serialnumber (RO)</td>
<td>Device serialnumber</td>
</tr>
<tr>
<td>Property Name</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>session.connection.host</td>
<td>Session host (or home URL)</td>
</tr>
<tr>
<td>session.connection.port</td>
<td>TE session port</td>
</tr>
<tr>
<td>session.connection.type</td>
<td>Session type</td>
</tr>
<tr>
<td></td>
<td>3270, 5250, VT220, HTML</td>
</tr>
<tr>
<td>session.printer.network.queue</td>
<td></td>
</tr>
</tbody>
</table>
Appendix 2 - IDA Action Codes

Many IDA codes apply only to a Terminal Emulation session but are listed here for completeness. Some IDA codes can only be used in restricted circumstances, such as IDA_URL.

<table>
<thead>
<tr>
<th>Symbolic Name</th>
<th>Friendly Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDA_BEL</td>
<td>Bell</td>
<td></td>
</tr>
<tr>
<td>IDA_BS</td>
<td>Backspace</td>
<td></td>
</tr>
<tr>
<td>IDA_HT</td>
<td>Horizontal Tab</td>
<td></td>
</tr>
<tr>
<td>IDA_TAB</td>
<td>Tab</td>
<td></td>
</tr>
<tr>
<td>IDA_LF</td>
<td>Linefeed</td>
<td></td>
</tr>
<tr>
<td>IDA_VT</td>
<td>Vertical Tab</td>
<td></td>
</tr>
<tr>
<td>IDA_FF</td>
<td>Form Feed</td>
<td></td>
</tr>
<tr>
<td>IDA_CR</td>
<td>Carriage Return</td>
<td></td>
</tr>
</tbody>
</table>

Printable ASCII

<table>
<thead>
<tr>
<th>Symbolic Name</th>
<th>Friendly Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDA_SPACE</td>
<td>&lt;Space&gt;</td>
</tr>
<tr>
<td>IDA_EXCLAMATION_MARK</td>
<td>!</td>
</tr>
<tr>
<td>IDA_DOUBLE_QUOTE</td>
<td>&quot;</td>
</tr>
<tr>
<td>IDA_NUMBER_SIGN</td>
<td>#</td>
</tr>
<tr>
<td>IDA_DOLLAR_SIGN</td>
<td>$</td>
</tr>
<tr>
<td>IDA_PERCENT</td>
<td>%</td>
</tr>
<tr>
<td>IDA_AMPERSAND</td>
<td>&amp;</td>
</tr>
<tr>
<td>IDA_SINGLE_QUOTE</td>
<td>'</td>
</tr>
<tr>
<td>IDA_LEFT_PAREN</td>
<td>(</td>
</tr>
<tr>
<td>IDA_RIGHT_PAREN</td>
<td>)</td>
</tr>
<tr>
<td>IDA_ASTERISK</td>
<td>*</td>
</tr>
<tr>
<td>IDA_PLUS</td>
<td>+</td>
</tr>
<tr>
<td>IDA_COMMA</td>
<td>,</td>
</tr>
<tr>
<td>IDA_HYPHEN</td>
<td>-</td>
</tr>
<tr>
<td>IDA_PERIOD</td>
<td>.</td>
</tr>
<tr>
<td>IDA_SLASH</td>
<td>/</td>
</tr>
<tr>
<td>IDA_0</td>
<td>0</td>
</tr>
<tr>
<td>IDA_1</td>
<td>1</td>
</tr>
<tr>
<td>…</td>
<td>…</td>
</tr>
<tr>
<td>IDA_9</td>
<td>9</td>
</tr>
<tr>
<td>IDA_COLON</td>
<td>:</td>
</tr>
<tr>
<td>IDA_SEMICOLON</td>
<td>;</td>
</tr>
<tr>
<td>IDA_LESS_THAN</td>
<td>&lt;</td>
</tr>
<tr>
<td>IDA_EQUAL</td>
<td>=</td>
</tr>
<tr>
<td>IDA_GREATER_THAN</td>
<td>&gt;</td>
</tr>
<tr>
<td>Symbolic Name</td>
<td>Friendly Name</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>---------------</td>
</tr>
<tr>
<td>IDA_QUESTION_MARK</td>
<td>?</td>
</tr>
<tr>
<td>IDA_AT</td>
<td>@</td>
</tr>
<tr>
<td>IDA_A</td>
<td>A</td>
</tr>
<tr>
<td>IDA_B</td>
<td>B</td>
</tr>
<tr>
<td>…</td>
<td>…</td>
</tr>
<tr>
<td>IDA_Z</td>
<td>Z</td>
</tr>
<tr>
<td>IDA_LEFT_BRACKET</td>
<td>{</td>
</tr>
<tr>
<td>IDA_BACKSLASH</td>
<td>\</td>
</tr>
<tr>
<td>IDA_RIGHT_BRACKET</td>
<td>}</td>
</tr>
<tr>
<td>IDA_CARET</td>
<td>^</td>
</tr>
<tr>
<td>IDA_UNDERSCORE</td>
<td>~</td>
</tr>
<tr>
<td>IDA_BACKTICK</td>
<td>`</td>
</tr>
<tr>
<td>IDA_a</td>
<td>a</td>
</tr>
<tr>
<td>IDA_b</td>
<td>b</td>
</tr>
<tr>
<td>…</td>
<td>…</td>
</tr>
<tr>
<td>IDA_z</td>
<td>Z</td>
</tr>
<tr>
<td>IDA_LEFT_BRACE</td>
<td>{</td>
</tr>
<tr>
<td>IDA_PIPE</td>
<td></td>
</tr>
<tr>
<td>IDA_RIGHT_BRACE</td>
<td>}</td>
</tr>
<tr>
<td>IDA_TILDE</td>
<td>~</td>
</tr>
<tr>
<td>IDA_DEL</td>
<td>DEL</td>
</tr>
<tr>
<td>C1 ASCII Controls</td>
<td></td>
</tr>
<tr>
<td>IDA_IND</td>
<td>Index</td>
</tr>
<tr>
<td>IDA_NEL</td>
<td>Next Line</td>
</tr>
<tr>
<td>IDA_HTS</td>
<td>Horiz Tab Set</td>
</tr>
<tr>
<td>IDA_RI</td>
<td>Reverse Index</td>
</tr>
<tr>
<td>IDA_SS2</td>
<td>Single Shift 2</td>
</tr>
<tr>
<td>IDA_SS3</td>
<td>Single Shift 3</td>
</tr>
<tr>
<td>IDA_DCS</td>
<td>Device Ctrl Str</td>
</tr>
<tr>
<td>IDA_PU1</td>
<td>Private Use One</td>
</tr>
<tr>
<td>IDA_PU2</td>
<td>Private Use Two</td>
</tr>
<tr>
<td>IDA_CSI</td>
<td>Ctrl Seq Intro</td>
</tr>
<tr>
<td>IDA_ST</td>
<td>String Term</td>
</tr>
<tr>
<td>IDA_OSC</td>
<td>OS Command</td>
</tr>
<tr>
<td>IDA_PM</td>
<td>Private Msg</td>
</tr>
<tr>
<td>IDA_APC</td>
<td>App Prog Cmd</td>
</tr>
</tbody>
</table>

Internal Actions (TE only)
<table>
<thead>
<tr>
<th>Symbolic Name</th>
<th>Friendly Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDA_UPDATE_CURSOR</td>
<td>Update Cursor</td>
<td></td>
</tr>
<tr>
<td>IDA_INHIBIT_UPDATE</td>
<td>Inhibit Update</td>
<td>Don’t update display</td>
</tr>
<tr>
<td>IDA_UNINHIBIT_UPDATE</td>
<td>Uninhibit Update</td>
<td>Allow display update</td>
</tr>
<tr>
<td>IDA_UPDATE</td>
<td>Update</td>
<td>Force display update</td>
</tr>
<tr>
<td>IDA_INHIBIT_SEND</td>
<td>Inhibit Send</td>
<td>VT buffer characters</td>
</tr>
<tr>
<td>IDA_UNINHIBIT_SEND</td>
<td>Uninhibit Send</td>
<td>VT stop buffering</td>
</tr>
<tr>
<td>IDA_SEND_PENDING</td>
<td>Send Pending Chars</td>
<td>VT send buffered chars</td>
</tr>
</tbody>
</table>

**Program Actions**

<table>
<thead>
<tr>
<th>Symbolic Name</th>
<th>Friendly Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDA_PROGRAM_ABOUT</td>
<td>Program About</td>
<td>Display About dialog</td>
</tr>
<tr>
<td>IDA_PROGRAM_EXIT</td>
<td>Program Exit</td>
<td>Exit program</td>
</tr>
<tr>
<td>IDA_PROGRAM_EXITSILENT</td>
<td>Program Exit Silent</td>
<td>Exit program silently</td>
</tr>
<tr>
<td>IDA_PROGRAM_HELP</td>
<td>Program Help</td>
<td>Display Help</td>
</tr>
<tr>
<td>IDA_SUSPEND_DEVICE</td>
<td>Suspend Device</td>
<td>Enter suspend state</td>
</tr>
<tr>
<td>IDA_BLUETOOTH_DISCOVERY</td>
<td>Bluetooth Discovery</td>
<td>Start discovery</td>
</tr>
<tr>
<td>IDA_WARMBOOT</td>
<td>Warm Boot</td>
<td>Warm boot device</td>
</tr>
<tr>
<td>IDA_COLDBOOT</td>
<td>Cold Boot</td>
<td>Cold boot device</td>
</tr>
<tr>
<td>IDA_MENU_TOPBOTTOM</td>
<td>Menu Top/Bot</td>
<td>Toggle menu location</td>
</tr>
<tr>
<td>IDA_MENU_TOGGLEHIDE</td>
<td>Menu Toggle</td>
<td>Toggle menu visibility</td>
</tr>
<tr>
<td>IDA_TOOLBAR_TOGGLE</td>
<td>ToolBar Toggle</td>
<td>Toggle toolbar visibility</td>
</tr>
<tr>
<td>IDA_START_TOGGLEHIDE</td>
<td>Start Menu Toggle</td>
<td>Toggle Start visibility</td>
</tr>
<tr>
<td>IDA_MENUBAR_TOGGLEHIDE</td>
<td>MenuBar Toggle</td>
<td>Toggle menubar visibility</td>
</tr>
<tr>
<td>IDA_SESSION_TOGGLECON</td>
<td>Connect/Discon</td>
<td>Toggle session connection</td>
</tr>
<tr>
<td>IDA_SESSION_CONFIGURE</td>
<td>Configure</td>
<td>Configure session</td>
</tr>
<tr>
<td>IDA_SESSION_CONNECT</td>
<td>Connect</td>
<td>Connect session</td>
</tr>
<tr>
<td>IDA_SESSION_DISCONNECT</td>
<td>Disconnect</td>
<td>Disconnect session</td>
</tr>
<tr>
<td>IDA_SESSION_NEXT_LIVE</td>
<td>Next Live Session</td>
<td>Switch to next live session</td>
</tr>
<tr>
<td>IDA_SESSION_PASSWORD</td>
<td>Password</td>
<td>Session password dialog</td>
</tr>
<tr>
<td>IDA_SESSION_PREV</td>
<td>Prev Session</td>
<td>Switch to previous session</td>
</tr>
<tr>
<td>IDA_SESSION_NEXT</td>
<td>Next Session</td>
<td>Switch to next session</td>
</tr>
<tr>
<td>IDA_SESSION_DISCONNECT_ALL</td>
<td>Disconnect All</td>
<td>Disconnect all sessions</td>
</tr>
<tr>
<td>IDA_SESSION_S1</td>
<td>Session 1</td>
<td>Switch to session 1</td>
</tr>
<tr>
<td>Symbolic Name</td>
<td>Friendly Name</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>---------------</td>
<td>------------------------------------</td>
</tr>
<tr>
<td>IDA_SESSION_S2</td>
<td>Session 2</td>
<td>Switch to session 2</td>
</tr>
<tr>
<td>IDA_SESSION_S3</td>
<td>Session 3</td>
<td>Switch to session 3</td>
</tr>
<tr>
<td>IDA_SESSION_S4</td>
<td>Session 4</td>
<td>Switch to session 4</td>
</tr>
<tr>
<td>IDA_TOOLBAND_HIDE</td>
<td>Hide ToolBar</td>
<td>Hide full Toolbar</td>
</tr>
<tr>
<td>IDA_TOOLBAND_TOGGLEHIDE</td>
<td>Toggle ToolBar</td>
<td>Toggle Toolbar visibility</td>
</tr>
<tr>
<td>IDA_KEYBAR_HIDE</td>
<td>Hide KeyBar</td>
<td>Hide KeyBar</td>
</tr>
<tr>
<td>IDA_KEYBAR_TOGGLEHIDE</td>
<td>KeyBar Toggle</td>
<td>Toggle KeyBar visibility</td>
</tr>
<tr>
<td>IDA_KEYBAR_LEFT</td>
<td>Prev KeyBar</td>
<td>Switch to previous KeyBar</td>
</tr>
<tr>
<td>IDA_KEYBAR_RIGHT</td>
<td>Next KeyBar</td>
<td>Switch to next KeyBar</td>
</tr>
<tr>
<td>IDA_KEYBAR_SEPARATOR</td>
<td>--Separator--</td>
<td>Separator for KeyBar</td>
</tr>
<tr>
<td>IDA_KEYBAR_NONE</td>
<td>(Empty)</td>
<td>No action placeholder</td>
</tr>
<tr>
<td>IDA_HSCROLL_HIDE</td>
<td>HScroll Hide</td>
<td></td>
</tr>
<tr>
<td>IDA_HSCROLL_VISIBLE</td>
<td>HScroll Show</td>
<td></td>
</tr>
<tr>
<td>IDA_HSCROLL_TOGGLEHIDE</td>
<td>HScroll Toggle</td>
<td></td>
</tr>
<tr>
<td>IDA_HSCROLL_PLUSONE</td>
<td>HScroll Right One</td>
<td></td>
</tr>
<tr>
<td>IDA_HSCROLL_MINUSONE</td>
<td>HScroll Left One</td>
<td></td>
</tr>
<tr>
<td>IDA_HSCROLL_PLUSHALF</td>
<td>HScroll Right Page</td>
<td></td>
</tr>
<tr>
<td>IDA_HSCROLL_MINUSHALF</td>
<td>HScroll Left Page</td>
<td></td>
</tr>
<tr>
<td>IDA_HSCROLL_PLUSEND</td>
<td>HScroll Right End</td>
<td></td>
</tr>
<tr>
<td>IDA_HSCROLL_MINUSEND</td>
<td>HScroll Left End</td>
<td></td>
</tr>
<tr>
<td>IDA_VSCROLL_HIDE</td>
<td>VScroll Hide</td>
<td></td>
</tr>
<tr>
<td>IDA_VSCROLL_VISIBLE</td>
<td>VScroll Show</td>
<td></td>
</tr>
<tr>
<td>IDA_VSCROLL_TOGGLEHIDE</td>
<td>VScroll Toggle</td>
<td></td>
</tr>
<tr>
<td>IDA_VSCROLL_PLUSONE</td>
<td>VScroll Up One</td>
<td></td>
</tr>
<tr>
<td>IDA_VSCROLL_MINUSONE</td>
<td>VScroll Down One</td>
<td></td>
</tr>
<tr>
<td>IDA_VSCROLL_PLUSHALF</td>
<td>VScroll Up Page</td>
<td></td>
</tr>
<tr>
<td>IDA_VSCROLL_MINUSHALF</td>
<td>VScroll Down Page</td>
<td></td>
</tr>
<tr>
<td>IDA_VSCROLL_PLUSEND</td>
<td>VScroll Up End</td>
<td></td>
</tr>
<tr>
<td>IDA_VSCROLL_MINUSEND</td>
<td>VScroll Down End</td>
<td></td>
</tr>
<tr>
<td>IDA_FONT_PLUS</td>
<td>Font Inc</td>
<td>Increase font size</td>
</tr>
<tr>
<td>IDA_FONT_MINUS</td>
<td>Font Dec</td>
<td>Decrease font size</td>
</tr>
<tr>
<td>IDA_TOGGLE_FONT_BOLD</td>
<td>Font Bold</td>
<td></td>
</tr>
<tr>
<td>IDA_SMARTPAD_OPEN</td>
<td>SmartPad Show</td>
<td></td>
</tr>
<tr>
<td>IDA_SMARTPAD_CLOSE</td>
<td>SmartPad Hide</td>
<td></td>
</tr>
<tr>
<td>IDA_SMARTPAD_TOGGLEHIDE</td>
<td>SmartPad Toggle</td>
<td></td>
</tr>
<tr>
<td>Symbolic Name</td>
<td>Friendly Name</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>---------------------</td>
<td>------------------------------</td>
</tr>
<tr>
<td>IDA_SLEEP_10</td>
<td>Sleep 10msec</td>
<td></td>
</tr>
<tr>
<td>IDA_SLEEP_50</td>
<td>Sleep 50msec</td>
<td></td>
</tr>
<tr>
<td>IDA_SLEEP_200</td>
<td>Sleep 200msec</td>
<td></td>
</tr>
<tr>
<td>IDA_SLEEP_1000</td>
<td>Sleep 1sec</td>
<td></td>
</tr>
<tr>
<td>IDA_SLEEP_5000</td>
<td>Sleep 5sec</td>
<td></td>
</tr>
<tr>
<td>IDA_SLEEP_20000</td>
<td>Sleep 20sec</td>
<td></td>
</tr>
<tr>
<td>IDA_SLEEP_100000</td>
<td>Sleep 100sec</td>
<td></td>
</tr>
<tr>
<td>IDA_SCAN_TRIGGER</td>
<td>Scan Trigger</td>
<td>Soft trigger scanner</td>
</tr>
<tr>
<td>IDA_MACRO_OPEN</td>
<td>Macro Show</td>
<td>Show Macro Tool</td>
</tr>
<tr>
<td>IDA_MACRO_CLOSE</td>
<td>Macro Hide</td>
<td>Hide Macro Tool</td>
</tr>
<tr>
<td>IDA_MACRO_TOGGLEHIDE</td>
<td>Macro Toggle</td>
<td>Toggle Macro Tool</td>
</tr>
<tr>
<td>IDA_MACRO_RECORD</td>
<td>Macro Record</td>
<td>Start Macro record</td>
</tr>
<tr>
<td>IDA_MACRO_STOP</td>
<td>Macro Stop</td>
<td>Stop Macro record</td>
</tr>
<tr>
<td>IDA_MACRO_PLAY</td>
<td>Macro Play</td>
<td>Replay Macro</td>
</tr>
<tr>
<td>IDA_PRINT_SCREEN</td>
<td>Print Screen</td>
<td>Print current screen</td>
</tr>
<tr>
<td>IDA_OIA_HIDE</td>
<td>OIA Hide</td>
<td>Hide IBM OIA bar</td>
</tr>
<tr>
<td>IDA_OIA_VISIBLE</td>
<td>OIA Show</td>
<td>Show IBM OIA bar</td>
</tr>
<tr>
<td>IDA_OIA_TOGGLEHIDE</td>
<td>OIA Toggle</td>
<td>Toggle OIA bar visibility</td>
</tr>
</tbody>
</table>

**General IBM and VT Actions**

<table>
<thead>
<tr>
<th>Symbolic Name</th>
<th>Friendly Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDA_PF1</td>
<td>F1</td>
<td>(Not VT PF1)</td>
</tr>
<tr>
<td>IDA_PF2</td>
<td>F2</td>
<td>(Not VT PF2)</td>
</tr>
<tr>
<td>IDA_PF3</td>
<td>F3</td>
<td>(Not VT PF3)</td>
</tr>
<tr>
<td>IDA_PF4</td>
<td>F4</td>
<td>(Not VT PF4)</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>IDA_PF24</td>
<td>F24</td>
<td></td>
</tr>
<tr>
<td>IDA_HOME</td>
<td>Home</td>
<td></td>
</tr>
<tr>
<td>IDA_DOWN</td>
<td>Down</td>
<td></td>
</tr>
<tr>
<td>IDA_UP</td>
<td>Up</td>
<td></td>
</tr>
<tr>
<td>IDA_LEFT</td>
<td>Left</td>
<td></td>
</tr>
<tr>
<td>IDA_RIGHT</td>
<td>Right</td>
<td></td>
</tr>
<tr>
<td>IDA_ENTER</td>
<td>Enter</td>
<td></td>
</tr>
</tbody>
</table>

**IBM Actions**

---

Web Browser Programming Guide  Page 69
<table>
<thead>
<tr>
<th>Symbolic Name</th>
<th>Friendly Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDA_IBM_HOME</td>
<td>IBM Home</td>
<td></td>
</tr>
<tr>
<td>IDA_DELETE</td>
<td>Delete</td>
<td></td>
</tr>
<tr>
<td>IDA_INSERT_ON</td>
<td>Insert On</td>
<td></td>
</tr>
<tr>
<td>IDA_INSERT_OFF</td>
<td>Insert Off</td>
<td></td>
</tr>
<tr>
<td>IDA_INSERT_TOGGLE</td>
<td>Insert Toggle</td>
<td></td>
</tr>
<tr>
<td>IDA_ATTN</td>
<td>Attn</td>
<td></td>
</tr>
<tr>
<td>IDA_CLEAR</td>
<td>Clear</td>
<td></td>
</tr>
<tr>
<td>IDA_CURSOR_SELECT</td>
<td>Cursor Select</td>
<td></td>
</tr>
<tr>
<td>IDA_DUP</td>
<td>DUP</td>
<td></td>
</tr>
<tr>
<td>IDA_ERASE_EOF</td>
<td>Erase EOF</td>
<td></td>
</tr>
<tr>
<td>IDA_ERASE_INPUT</td>
<td>Erase Input</td>
<td></td>
</tr>
<tr>
<td>IDA_FIELD_MARK</td>
<td>Field Mark</td>
<td></td>
</tr>
<tr>
<td>IDA_NEWLINE</td>
<td>Newline</td>
<td></td>
</tr>
<tr>
<td>IDA_PA1</td>
<td>PA1</td>
<td></td>
</tr>
<tr>
<td>IDA_PA2</td>
<td>PA2</td>
<td></td>
</tr>
<tr>
<td>IDA_PA3</td>
<td>PA3</td>
<td></td>
</tr>
<tr>
<td>IDA_RESET</td>
<td>Reset</td>
<td></td>
</tr>
<tr>
<td>IDA_SYSREQ</td>
<td>Sys Request</td>
<td></td>
</tr>
</tbody>
</table>

**5250 Specific Actions**

<table>
<thead>
<tr>
<th>Symbolic Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDA_FIELD_EXIT</td>
<td>Field Exit</td>
</tr>
<tr>
<td>IDA_FIELD_PLUS</td>
<td>Field +</td>
</tr>
<tr>
<td>IDA_FIELD_MINUS</td>
<td>Field -</td>
</tr>
<tr>
<td>IDA_FIELD_ADVANCE</td>
<td>Field Advance</td>
</tr>
<tr>
<td>IDA_FIELD_BACKSPACE</td>
<td>Field Backspace</td>
</tr>
<tr>
<td>IDA_FIELD_SUB</td>
<td>Field SUB</td>
</tr>
<tr>
<td>IDA_HELP</td>
<td>IBM Help</td>
</tr>
<tr>
<td>IDA_ROLL_DOWN</td>
<td>Roll Down</td>
</tr>
<tr>
<td>IDA_ROLL_UP</td>
<td>Roll Up</td>
</tr>
<tr>
<td>IDA_ROLL_LEFT</td>
<td>Roll Left</td>
</tr>
<tr>
<td>IDA_ROLL_RIGHT</td>
<td>Roll Right</td>
</tr>
<tr>
<td>IDA_BACKSPACE</td>
<td>Backspace</td>
</tr>
<tr>
<td>IDA_PRINT</td>
<td>IBM Print</td>
</tr>
</tbody>
</table>

**VT Actions**

<table>
<thead>
<tr>
<th>Symbolic Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDA_ANSWERBACK</td>
<td>Answerback</td>
</tr>
<tr>
<td>IDA_FIND</td>
<td>Find</td>
</tr>
<tr>
<td>IDA_INSERT_HERE</td>
<td>Insert Here</td>
</tr>
<tr>
<td>IDA_NEXT</td>
<td>Next</td>
</tr>
<tr>
<td>IDA_PREVIOUS</td>
<td>Previous</td>
</tr>
<tr>
<td>Symbolic Name</td>
<td>Friendly Name</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------------</td>
</tr>
<tr>
<td>IDA_REMOVE</td>
<td>Remove</td>
</tr>
<tr>
<td>IDA_SELECT</td>
<td>Select</td>
</tr>
<tr>
<td>IDA_VT_PF1</td>
<td>VT PF1</td>
</tr>
<tr>
<td>IDA_VT_PF2</td>
<td>VT PF2</td>
</tr>
<tr>
<td>IDA_VT_PF3</td>
<td>VT PF3</td>
</tr>
<tr>
<td>IDA_VT_PF4</td>
<td>VT PF4</td>
</tr>
<tr>
<td>IDA_VT_COMMA</td>
<td>Numpad Comma</td>
</tr>
<tr>
<td>IDA_NUMPAD_0</td>
<td>Numpad 0</td>
</tr>
<tr>
<td>IDA_NUMPAD_1</td>
<td>Numpad 1</td>
</tr>
<tr>
<td>IDA_NUMPAD_2</td>
<td>Numpad 2</td>
</tr>
<tr>
<td>IDA_NUMPAD_3</td>
<td>Numpad 3</td>
</tr>
<tr>
<td>IDA_NUMPAD_4</td>
<td>Numpad 4</td>
</tr>
<tr>
<td>IDA_NUMPAD_5</td>
<td>Numpad 5</td>
</tr>
<tr>
<td>IDA_NUMPAD_6</td>
<td>Numpad 6</td>
</tr>
<tr>
<td>IDA_NUMPAD_7</td>
<td>Numpad 7</td>
</tr>
<tr>
<td>IDA_NUMPAD_8</td>
<td>Numpad 8</td>
</tr>
<tr>
<td>IDA_NUMPAD_9</td>
<td>Numpad 9</td>
</tr>
<tr>
<td>IDA_VT_ENTER</td>
<td>Numpad Enter</td>
</tr>
<tr>
<td>IDA_VT_MINUS</td>
<td>Numpad Minus</td>
</tr>
<tr>
<td>IDA_VT_PERIOD</td>
<td>Numpad Period</td>
</tr>
<tr>
<td>IDA_UDK_F6</td>
<td>UDK F6</td>
</tr>
<tr>
<td>IDA_UDK_F7</td>
<td>UDK F7</td>
</tr>
<tr>
<td></td>
<td>...</td>
</tr>
<tr>
<td>IDA_UDK_F20</td>
<td>UDK F20</td>
</tr>
<tr>
<td>IDA_VT_HELP</td>
<td>VT Help</td>
</tr>
<tr>
<td>IDA_VT_DO</td>
<td>VT Do</td>
</tr>
<tr>
<td>IDA_ADD</td>
<td>Add</td>
</tr>
<tr>
<td>IDA_MULTIPLY</td>
<td>Multiply</td>
</tr>
<tr>
<td>IDA_DIVIDE</td>
<td>Divide</td>
</tr>
</tbody>
</table>

**Custom VT Sequences**

<table>
<thead>
<tr>
<th>Symbolic Name</th>
<th>Friendly Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDA_VT_SAP0135</td>
<td>VT SAP0135</td>
<td>0x00 0x35</td>
</tr>
<tr>
<td>IDA_VT_CSI_M</td>
<td>VT CSI M</td>
<td>ESC [ M</td>
</tr>
<tr>
<td>IDA_VT_CSI_N</td>
<td>VT CSI N</td>
<td>ESC [ N</td>
</tr>
<tr>
<td>IDA_VT_CSI_O</td>
<td>VT CSI O</td>
<td></td>
</tr>
<tr>
<td>IDA_VT_CSI_P</td>
<td>VT CSI P</td>
<td></td>
</tr>
<tr>
<td>Symbolic Name</td>
<td>Friendly Name</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-----------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>IDA_VT_CSI_Q</td>
<td>VT CSI Q</td>
<td></td>
</tr>
<tr>
<td>IDA_VT_CSI_R</td>
<td>VT CSI R</td>
<td></td>
</tr>
<tr>
<td>IDA_VT_CSI_S</td>
<td>VT CSI S</td>
<td></td>
</tr>
<tr>
<td>IDA_VT_CSI_T</td>
<td>VT CSI T</td>
<td></td>
</tr>
</tbody>
</table>

**Windows App Keys**

| IDA_APPKEY_K1                | App Key 1       |                                   |
| IDA_APPKEY_K2                | App Key 2       |                                   |
| ...                          | ...             |                                   |
| IDA_APPKEY_K16               | App Key 16      |                                   |

| IDA_SCROLL_UPPERLEFT         | Scroll Upper Left|                                   |
| IDA_SCROLL_UPPERRIGHT        | Scroll Upper Right|                                 |
| IDA_SCROLL_LOWERLEFT         | Scroll Lower Left|                                |
| IDA_SCROLL_LOWERRIGHT        | Scroll Lower Right|                               |
| IDA_SCROLL_CENTER            | Scroll Center   |                                   |
| IDA_SCROLL_CURSOR_CENTER    | Scroll Cursor Center|                             |
| IDA_SCROLL_CURSOR_VISIBLE   | Scroll Cursor Visible|                          |

| IDA_COPYALL                  | Copy All        | Copy screen to clipboard         |
| IDA_PASTE                    | Paste           | Past clipboard                    |

| IDA_USTRING_0                | Text 1          | Send user text 1                  |
| IDA_USTRING_1                | Text 2          | Send user text 2                  |
| ...                          | ...             |                                   |
| IDA_USTRING_63               | Text 64         | Send user text 64                 |

| IDA_SCRIPT_1                 | Script 1        | Run Script 1                      |
| IDA_SCRIPT_2                 | Script 2        | Run Script 2                      |
| ...                          | ...             |                                   |
| IDA_SCRIPT_64                | Script 64       | Run Script 64                     |

<p>| IDA_SIP_HIDE                 | SIP Hide        |                                   |
| IDA_SIP_SHOW                 | SIP Show        |                                   |
| IDA_SIP_TOGGLEHIDE           | SIP Toggle      |                                   |
| IDA_SIP_LOCKDOWN             | SIP Lockdown    |                                   |
| IDA_SIP_UNLOCK               | SIP Unlock      |                                   |
| IDA_SIP_UP                   | SIP Up          |                                   |
| IDA_SIP_DOWN                 | SIP Down        |                                   |
| IDA_SIP_FORCEDOWN            | SIP Forcedown   |                                   |</p>
<table>
<thead>
<tr>
<th>Symbolic Name</th>
<th>Friendly Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDA_IM_KEYBOARD</td>
<td>IM Keyboard</td>
<td></td>
</tr>
<tr>
<td>IDA_IM_LOCKED</td>
<td>IM Locked</td>
<td></td>
</tr>
</tbody>
</table>

**HTML Actions**

| IDA_URL_HOME                | URL Home          |                                   |
| IDA_URL_BACK                | URL Back          |                                   |
| IDA_URL                     | URL               | Defines start of URL              |

**Special Actions**

| IDA_VIBRATE_100             | Vibrate 100ms     |                                   |
| IDA_VIBRATE_200             | Vibrate 200ms     |                                   |
| IDA_VIBRATE_500             | Vibrate 500ms     |                                   |
| IDA_VIBRATE_1000            | Vibrate 1sec      |                                   |
| IDA_VIBRATE_2000            | Vibrate 2sec      |                                   |
| IDA_VIBRATE_5000            | Vibrate 5sec      |                                   |
| IDA_BEEP_OK                 | Beep              |                                   |
| IDA_BEEP_WARN               | Beep Warn         |                                   |
| IDA_BEEPLOUD                | Beep Loud         |                                   |

| IDA_KBD_ALPHA               | KeyMode Alpha     |                                   |
| IDA_KBD_NUMERIC             | KeyMode Numeric   |                                   |
| IDA_KBD_ALPHANUM            | KeyMode AlphaNum  |                                   |
| IDA_KBD_UPPERALPHA          | KeyMode Upper Alpha|                                 |
| IDA_KBD_LOWERALPHA          | Keymode Lower Alpha|                                |
| IDA_KBD_FUNCMODE            | KeyMode Func      |                                   |
| IDA_KBD_CYCLEMODE           | KeyMode Cycle     | Cycle to next mode               |

| IDA_POPUP_IPADDRESS         | Show IP Address   |                                   |
| IDA_POPUP_MACADDRESS        | Show MAC Address  |                                   |
| IDA_POPUP_BATTERY           | Show Battery      |                                   |
| IDA_POPUP_TIME              | Show Time         |                                   |
| IDA_POPUP_SERIALNUMBER      | Show Serial #     |                                   |
| IDA_POPUP_DEVICEID          | Show Device ID    |                                   |
| IDA_POPUP_RFINFO            | Show RF info      |                                   |
## Appendix 3 - Virtual Key Codes

This appendix contains a list of Windows CE Virtual Key Codes (VK) which are used with the OnKey META tag.

Notice that there is no case distinction of the alphabetic keys. Also, note that the symbols on the tops of the digit keys are not listed because they are a shift state of the digit keys.

<table>
<thead>
<tr>
<th>Symbolic Name</th>
<th>Hexadecimal Value</th>
<th>Keyboard Equivalent</th>
</tr>
</thead>
<tbody>
<tr>
<td>VK_BACK</td>
<td>08</td>
<td>BACKSPACE key</td>
</tr>
<tr>
<td>VK_TAB</td>
<td>09</td>
<td>TAB key</td>
</tr>
<tr>
<td>VK_CLEAR</td>
<td>0C</td>
<td>CLEAR key</td>
</tr>
<tr>
<td>VK_RETURN</td>
<td>0D</td>
<td>ENTER key</td>
</tr>
<tr>
<td>VK_SHIFT</td>
<td>10</td>
<td>SHIFT key</td>
</tr>
<tr>
<td>VK_CONTROL</td>
<td>11</td>
<td>CTRL key</td>
</tr>
<tr>
<td>VK_MENU</td>
<td>12</td>
<td>ALT key</td>
</tr>
<tr>
<td>VK_PAUSE</td>
<td>13</td>
<td>PAUSE key</td>
</tr>
<tr>
<td>VK_CAPITAL</td>
<td>14</td>
<td>CAPS LOCK key</td>
</tr>
<tr>
<td>VK_ESCAPE</td>
<td>1B</td>
<td>ESC key</td>
</tr>
<tr>
<td>VK_SPACE</td>
<td>20</td>
<td>SPACEBAR</td>
</tr>
<tr>
<td>VK_PRIOR</td>
<td>21</td>
<td>PAGE UP key</td>
</tr>
<tr>
<td>VK_NEXT</td>
<td>22</td>
<td>PAGE DOWN key</td>
</tr>
<tr>
<td>VK_END</td>
<td>23</td>
<td>END key</td>
</tr>
<tr>
<td>VK_HOME</td>
<td>24</td>
<td>HOME key</td>
</tr>
<tr>
<td>VK_LEFT</td>
<td>25</td>
<td>LEFT ARROW key</td>
</tr>
<tr>
<td>VK_UP</td>
<td>26</td>
<td>UP ARROW key</td>
</tr>
<tr>
<td>VK_RIGHT</td>
<td>27</td>
<td>RIGHT ARROW key</td>
</tr>
<tr>
<td>VK_DOWN</td>
<td>28</td>
<td>DOWN ARROW key</td>
</tr>
<tr>
<td>VK_SELECT</td>
<td>29</td>
<td>SELECT key</td>
</tr>
<tr>
<td>VK_EXECUTE</td>
<td>2B</td>
<td>EXECUTE key</td>
</tr>
<tr>
<td>VK_SNAPSHOT</td>
<td>2C</td>
<td>PRINT SCREEN key</td>
</tr>
<tr>
<td>VK_INSERT</td>
<td>2D</td>
<td>INS key</td>
</tr>
<tr>
<td>VK_DELETE</td>
<td>2E</td>
<td>DEL key</td>
</tr>
<tr>
<td>VK_HELP</td>
<td>2F</td>
<td>HELP key</td>
</tr>
<tr>
<td>VK_0</td>
<td>30</td>
<td>0 key</td>
</tr>
<tr>
<td>VK_1</td>
<td>31</td>
<td>1 key</td>
</tr>
<tr>
<td>Symbolic Name</td>
<td>Hexadecimal Value</td>
<td>Keyboard Equivalent</td>
</tr>
<tr>
<td>---------------</td>
<td>------------------</td>
<td>---------------------</td>
</tr>
<tr>
<td>VK_2</td>
<td>32</td>
<td>2 key</td>
</tr>
<tr>
<td>VK_3</td>
<td>33</td>
<td>3 key</td>
</tr>
<tr>
<td>VK_4</td>
<td>34</td>
<td>4 key</td>
</tr>
<tr>
<td>VK_5</td>
<td>35</td>
<td>5 key</td>
</tr>
<tr>
<td>VK_6</td>
<td>36</td>
<td>6 key</td>
</tr>
<tr>
<td>VK_7</td>
<td>37</td>
<td>7 key</td>
</tr>
<tr>
<td>VK_8</td>
<td>38</td>
<td>8 key</td>
</tr>
<tr>
<td>VK_9</td>
<td>39</td>
<td>9 key</td>
</tr>
<tr>
<td>VK_A</td>
<td>41</td>
<td>A key</td>
</tr>
<tr>
<td>VK_B</td>
<td>42</td>
<td>B key</td>
</tr>
<tr>
<td>VK_C</td>
<td>43</td>
<td>C key</td>
</tr>
<tr>
<td>VK_D</td>
<td>44</td>
<td>D key</td>
</tr>
<tr>
<td>VK_E</td>
<td>45</td>
<td>E key</td>
</tr>
<tr>
<td>VK_F</td>
<td>46</td>
<td>F key</td>
</tr>
<tr>
<td>VK_G</td>
<td>47</td>
<td>G key</td>
</tr>
<tr>
<td>VK_H</td>
<td>48</td>
<td>H key</td>
</tr>
<tr>
<td>VK_I</td>
<td>49</td>
<td>I key</td>
</tr>
<tr>
<td>VK_J</td>
<td>4A</td>
<td>J key</td>
</tr>
<tr>
<td>VK_K</td>
<td>4B</td>
<td>K key</td>
</tr>
<tr>
<td>VK_L</td>
<td>4C</td>
<td>L key</td>
</tr>
<tr>
<td>VK_M</td>
<td>4D</td>
<td>M key</td>
</tr>
<tr>
<td>VK_N</td>
<td>4E</td>
<td>N key</td>
</tr>
<tr>
<td>VK_O</td>
<td>4F</td>
<td>O key</td>
</tr>
<tr>
<td>VK_P</td>
<td>50</td>
<td>P key</td>
</tr>
<tr>
<td>VK_Q</td>
<td>51</td>
<td>Q key</td>
</tr>
<tr>
<td>VK_R</td>
<td>52</td>
<td>R key</td>
</tr>
<tr>
<td>VK_S</td>
<td>53</td>
<td>S key</td>
</tr>
<tr>
<td>VK_T</td>
<td>54</td>
<td>T key</td>
</tr>
<tr>
<td>VK_U</td>
<td>55</td>
<td>U key</td>
</tr>
<tr>
<td>VK_V</td>
<td>56</td>
<td>V key</td>
</tr>
<tr>
<td>VK_W</td>
<td>57</td>
<td>W key</td>
</tr>
<tr>
<td>VK_X</td>
<td>58</td>
<td>X key</td>
</tr>
<tr>
<td>VK_Y</td>
<td>59</td>
<td>Y key</td>
</tr>
<tr>
<td>Symbolic Name</td>
<td>Hexadecimal Value</td>
<td>Keyboard Equivalent</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------------</td>
<td>--------------------------------</td>
</tr>
<tr>
<td>VK_Z</td>
<td>5A</td>
<td>Z key</td>
</tr>
<tr>
<td>VK_NUMPAD0</td>
<td>60</td>
<td>Numeric keypad 0 key</td>
</tr>
<tr>
<td>VK_NUMPAD1</td>
<td>61</td>
<td>Numeric keypad 1 key</td>
</tr>
<tr>
<td>VK_NUMPAD2</td>
<td>62</td>
<td>Numeric keypad 2 key</td>
</tr>
<tr>
<td>VK_NUMPAD3</td>
<td>63</td>
<td>Numeric keypad 3 key</td>
</tr>
<tr>
<td>VK_NUMPAD4</td>
<td>64</td>
<td>Numeric keypad 4 key</td>
</tr>
<tr>
<td>VK_NUMPAD5</td>
<td>65</td>
<td>Numeric keypad 5 key</td>
</tr>
<tr>
<td>VK_NUMPAD6</td>
<td>66</td>
<td>Numeric keypad 6 key</td>
</tr>
<tr>
<td>VK_NUMPAD7</td>
<td>67</td>
<td>Numeric keypad 7 key</td>
</tr>
<tr>
<td>VK_NUMPAD8</td>
<td>68</td>
<td>Numeric keypad 8 key</td>
</tr>
<tr>
<td>VK_NUMPAD9</td>
<td>69</td>
<td>Numeric keypad 9 key</td>
</tr>
<tr>
<td>VK_MULTIPLY</td>
<td>6A</td>
<td>Multiply key</td>
</tr>
<tr>
<td>VK_ADD</td>
<td>6B</td>
<td>Add key</td>
</tr>
<tr>
<td>VK_SEPARATOR</td>
<td>6C</td>
<td>Separator key</td>
</tr>
<tr>
<td>VK_SUBTRACT</td>
<td>6D</td>
<td>Subtract key</td>
</tr>
<tr>
<td>VK_DECIMAL</td>
<td>6E</td>
<td>Decimal key</td>
</tr>
<tr>
<td>VK_DIVIDE</td>
<td>6F</td>
<td>Divide key</td>
</tr>
<tr>
<td>VK_F1</td>
<td>70</td>
<td>F1 key</td>
</tr>
<tr>
<td>VK_F2</td>
<td>71</td>
<td>F2 key</td>
</tr>
<tr>
<td>VK_F3</td>
<td>72</td>
<td>F3 key</td>
</tr>
<tr>
<td>VK_F4</td>
<td>73</td>
<td>F4 key</td>
</tr>
<tr>
<td>VK_F5</td>
<td>74</td>
<td>F5 key</td>
</tr>
<tr>
<td>VK_F6</td>
<td>75</td>
<td>F6 key</td>
</tr>
<tr>
<td>VK_F7</td>
<td>76</td>
<td>F7 key</td>
</tr>
<tr>
<td>VK_F8</td>
<td>77</td>
<td>F8 key</td>
</tr>
<tr>
<td>VK_F9</td>
<td>78</td>
<td>F9 key</td>
</tr>
<tr>
<td>VK_F10</td>
<td>79</td>
<td>F10 key</td>
</tr>
<tr>
<td>VK_F11</td>
<td>7A</td>
<td>F11 key</td>
</tr>
<tr>
<td>VK_F12</td>
<td>7B</td>
<td>F12 key</td>
</tr>
<tr>
<td>VK_F13</td>
<td>7C</td>
<td>F13 key</td>
</tr>
<tr>
<td>VK_F14</td>
<td>7D</td>
<td>F14 key</td>
</tr>
<tr>
<td>VK_F15</td>
<td>7E</td>
<td>F15 key</td>
</tr>
<tr>
<td>Symbolic Name</td>
<td>Hexadecimal Value</td>
<td>Keyboard Equivalent</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------------</td>
<td>---------------------</td>
</tr>
<tr>
<td>VK_F16</td>
<td>7F</td>
<td>F16 key</td>
</tr>
<tr>
<td>VK_F17</td>
<td>80</td>
<td>F17 key</td>
</tr>
<tr>
<td>VK_F18</td>
<td>81</td>
<td>F18 key</td>
</tr>
<tr>
<td>VK_F19</td>
<td>82</td>
<td>F19 key</td>
</tr>
<tr>
<td>VK_F20</td>
<td>83</td>
<td>F20 key</td>
</tr>
<tr>
<td>VK_F21</td>
<td>84</td>
<td>F21 key</td>
</tr>
<tr>
<td>VK_F22</td>
<td>85</td>
<td>F22 key</td>
</tr>
<tr>
<td>VK_F23</td>
<td>86</td>
<td>F23 key</td>
</tr>
<tr>
<td>VK_F24</td>
<td>87</td>
<td>F24 key</td>
</tr>
</tbody>
</table>
Glossary

**CEBrowseX**
A Naurtech ActiveX control which provides access to the Naurtech Web Browser configuration and actions.

**external**
This is the name of an implied object in the DOM of the Windows CE .NET and CE 5.0 browser that gives access to special non-standard features. See the CEBrowseX documentation for details.

**GetProperty**
A method available via the CEBrowseX ActiveX object or the CE .NET “external” object to get a configuration value from the Web Browser.

**IDA Action Code**
An IDA Action Code defines a special device, application, or emulation action within the Naurtech Smart Clients. IDA codes can be tied to keys, or KeyBars, and invoked via META tags or JavaScript. See the Appendix for a list of values.

**PostIDA**
A method available via the CEBrowseX ActiveX object or the CE .NET “external” object to post an IDA Action Code to the Web Browser.

**SendIDA**
A method available via the CEBrowseX ActiveX object or the CE .NET “external” object to send an IDA Action Code to the Web Browser. The SendIDA is a synchronous activation of the action.

**SetProperty**
A method available via the CEBrowseX ActiveX object or the CE .NET “external” object to set a configuration value in the Web Browser.

**TextX**
A Naurtech ActiveX control that is used with Pocket PC or Windows Mobile 2003 platforms to provide a fully featured text input element which supports event handlers.
Index

C

CEBrowseX CLASSID · 53
CEBrowseX Control · 53

E

external object · 53

G

GetProperty · 54, 62

H

hidden fields · 45

I

iBrowse META tags · 16
IDA Action Codes · 11, 63

M

META Tag Identifiers · 16
  Application · 17
  Battery · 18
  BatteryNavigate · 19
  Command · 20
  CursorPos · 21
  ErrorNavigate · 21
  GetUnitInformation · 22
  HomeKey · 23
  IDA · 23
  MoveSIP · 24
  OnAllKeys · 25
  OnKey · 26
  PLSeriesLabel_Complete · 38
  PLSeriesLabel_Print · 39
  PowerOn · 28
  Reboot · 29
  Scanner · 29
  ScannerNavigate · 30
  SetDate · 32
  SetTime · 32
  Signal · 33
  SignalNavigate · 34
  SIP · 35
  SIPUp · 36
  TextSize · 36
  TimerInterval · 37
  TimerNavigate · 37
  ZebraLabel_Complete · 38
  ZebraLabel_Print · 39

O

OnChange · 60
OnClick · 60
OnFocus · 60
OnKeyPress · 61
OnKeyUp · 61
OnLostFocus · 60

P

PlaySound · 55
PlayTone · 55
PostIDA · 54
Print · 55
PrintString · 55

S

SendIDA · 54
SetFocus · 59
SetProperty · 55, 62
ShowSIP · 59
stiscan · 50
stisubmit · 50

T

TextX CLASSID · 58
TextX Control · 57