DPI Configuration for Small PCs

A Guide for OEMs

December 3, 2009

Abstract

This paper provides information about the high-DPI (dot-per-inch) feature for the Windows® family of operating systems. This feature helps compensate for the small text size that results from high-resolution displays.

This paper is for display vendors and original equipment manufacturers (OEMs) of ultramobile PCs who must determine the best default screen resolution for their hardware. It explains the concepts that are involved in effective screen resolution, provides guidance for OEMs to determine the appropriate effective screen resolution, and includes a case study that shows the potential trade-offs that vendors might encounter.

This information applies to the Windows 7 operating system.

References and resources discussed here are listed at the end of this paper.

The current version of this paper is maintained on the Web at:   
 <http://www.microsoft.com/whdc/>device/display/DPIConfig\_SmallPCs.mspx

Disclaimer:

The information contained in this document represents the current view of Microsoft Corporation on the issues discussed as of the date of publication. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information presented after the date of publication.

This White Paper is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO THE INFORMATION IN THIS DOCUMENT.

Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

Unless otherwise noted, the example companies, organizations, products, domain names, e-mail addresses, logos, people, places and events depicted herein are fictitious, and no association with any real company, organization, product, domain name, email address, logo, person, place or event is intended or should be inferred.

© 2009 Microsoft Corporation. All rights reserved.

Microsoft, MSDN, Windows, and Windows Vista are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

The names of actual companies and products mentioned herein may be the trademarks of their respective owners.

Document History

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Date | Change |  |  |  |
| December 3, 2009 | Approved for public release | | | |
| June 5, 2009 | Preliminary version for publication on Connect | | | |

# Introduction

As technology advances, the number of dots per inch (DPI) in displays continues to increase. The Windows® High DPI feature compensates for the small text size that results from high-resolution displays.

This paper helps original equipment manufacturers (OEMs) of ultramobile PCs determine the best default screen resolution for their hardware. It explains the concepts that are involved in screen resolution, provides guidance for OEMs to determine the appropriate screen resolution, and includes a case study that shows the potential trade-offs that vendors might encounter.

The paper uses the following terminology:

native resolution

The number of dots per inch that the device hardware can display.

physical resolution

The current screen resolution, which can be less than or equal to the native resolution.

system DPI

The DPI setting that Windows uses. In Windows 7, system DPI is expressed as a percentage of 96. Therefore:

96 DPI = 100 percent

120 DPI = 125 percent

144 DPI = 150 percent

effective resolution

The apparent resolution that considers both the physical resolution and the OS DPI. As the system DPI setting increases, the effective resolution decreases.

effective font size

The apparent size of text on the screen when the ratio of the system DPI to the native screen resolution is considered.

# About DPI and Screen Resolution

By default, Windows renders text in a 9-point font to balance readability with screen real-estate usage. Point size is an absolute measure where 1point = 1/72 inch on a 96-DPI screen.On high-resolution displays, text that Windows renders in the default 9-point font appears smaller because the pixels are closer together. User interface elements similarly appear smaller.

Windows considers 96 DPI to be 100 percent. If the system DPI is greater than 96, text and other user interface (UI) elements appear larger. For example, at 125 percent DPI, Windows text and UI elements appear 25 percent larger than at 100 percent, but less of the UI fits on the screen. At the same time, this means that the effective resolution is reduced. It is important to understand this concept because Windows has minimum requirements for effective resolution.

## Effective Resolution

The effective resolution of a display considers both the physical resolution and the system DPI setting. As the DPI setting increases, the effective resolution decreases. The formula to calculate this is as follows:

Effective Resolution = Physical Resolution / DPI

Table 1 shows the effective resolution for some common displays at various DPI settings.

Table 1. Effective Resolution for Common Displays

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Description | Horizontal  pixels | Vertical  Pixels | Effective resolution at 100% DPI | Effective resolution at 125% DPI | Effective resolution at 150% DPI |
| 15.4” WXGA+ | 1440 | 900 | 1440x900 | 1152x720 | 960x600 |
| 14.1” WXGA | 1280 | 768 | 1280x768 | 1024x614 | 853x512 |
| 17” WUXGA | 1920 | 1200 | 1920x1200 | 1536x960 | 1280x800 |
| 15.4” WSXGA+ | 1680 | 1050 | 1680x1050 | 1344x840 | 1120x700 |
| 15.4” WUXGA | 1920 | 1200 | 1920x1200 | 1536x960 | 1280x800 |

For example, 1440 @ 150% DPI = 1440/ (150/100), which is equal to 960.

For more information on effective resolution and the problems that can occur with applications that do not correctly consider effective resolution, see “Writing High-DPI Win32 Applications” on the MSDN® Web site.

## Effective Font Size

The effective font size is the apparent size of text on the screen when the ratio of the system DPI to the native screen resolution is considered. If the system DPI and the native resolution are the same, the effective font size is the same as the absolute point size. Therefore, at 96 DPI on a screen that supports 96 pixels per inch, 9-point text renders at 9 points.

You can calculate the effective font size by using the following formula:

Effective Point Size = Point Size \* System DPI/Native DPI

For example, 9-point text on a 221-DPI display set that is set at 96 DPI (100 percent) is equivalent in size to 3.9-point text (9 \* 96/221) on a 96-DPI display.

Our research shows that users generally consider text too small if the effective font size is less than 5 points. This research is based on a reading distance of 18 inches and 20/20 visual acuity. OEMs should use this as a guideline to decide when to use optimal effective resolution versus minimum effective resolution. For example, if the effective font size would become 4 points when the DPI is set for optimal effective resolution, we recommend that OEMs override the default setting. In these cases, the DPI should be adjusted to increase the size of the text as long as the effective resolution does not fall below 800x600.

# Guidelines for DPI and Screen Resolution

The following are the guidelines for the optimal and minimum experiences.

##### Optimal Windows Effective Resolution

* Windows 7 is optimized for a minimum effective resolution of 1024x768.
* Windows user experience (UX) guidelines specify that applications should optimize for this resolution.
* System DPI should be configured based on the native DPI of the display.

##### Minimum Windows Requirement for Effective Resolution

* Windows minimum requirement for effective resolution is 800x600.
* Windows does not address issues that can occur if the effective resolution is lower than this.
* Windows UX guidelines specify that third-party applications should work at this resolution.

Small, high-density screens pose a challenge. When such a screen is configured to produce an optimal effective resolution, the text size can be too small for many people to read comfortably. In this situation, it is more important to increase the text size than to keep the optimal effective resolution for Windows. However, never configure the display below the minimum effective resolution.

##### When to Use Minimum Effective Resolution

Windows uses a default text size of 9 points at 96 DPI. When the effective font size is less than 5 points, many users have difficulty reading text, if you assume a viewing distance of 18 inches and 20/20 visual acuity. Therefore, we recommend that vendors increase the system DPI for displays if the default setting results in an effective font size that is less than 5 points. Vendors should increase the DPI to compensate as much as they can without going below the minimum effective resolution of 800x600.

# Configuring High DPI in Windows 7

Windows 7 provides the following three ways to set high DPI:

* DPI auto-configuration.
* OEM override.
* DPI setting in theControl Panel Display application.

The following sections discuss each method of changing DPI in more details.

## DPI Auto Configuration

The first time Windows 7 runs, it automatically configures the DPI based on the native DPI for the display, which is reported in the Extended Display Identification Data (EDID) of the display. The auto-configured resolution is never less than the optimal effective resolution of 1024x768.This resolution is required for the optimal Windows user experience and is recommended in the Windows User Experience Interaction Guidelines (the “UX Guide”) for independent software vendors (ISVs) to target for their optimal experience. For a link to the UX Guide, see “Resources” at the end of this paper.

Application compatibility testing shows that some third-party applications are incompatible with an effective resolution of less than 1024x768. We chose 1024x768 as the minimum for the auto-configuration feature to minimize these compatibility issues. However, most applications work correctly at an effective resolution of 800x600 because the UX Guide has specified this requirement for earlier Windows versions.

When a user runs Windows for the first time, the system calculates the native display DPI by using the following formula:

Native Display DPI = Native Horizontal Resolution / Screen Width in Inches

The information for the preceding formula is from the display EDID. If the EDID data is invalid, incomplete, or unavailable, the system DPI uses 100-percent DPI as the default. After Windows calculates the native DPI, the system sets the system DPI as follows:

|  |  |
| --- | --- |
| If the native display DPI is: | Windows sets the auto configuration DPI to: |
| <114.5 | 100 percent |
| >=114.5 - <137.5 | 125 percent |
| >=137.5 | 150 percent |

If the resulting effective resolution is lower than 1024x768, Windows sets the system DPI to the highest DPI that ensures an effective resolution of at least 1024x768.

For more information on EDID, see the “VESA EDID Implementation Guide,” which is listed in “Resources.”

## Overriding DPI Auto Configuration

Windows 7 enables OEMs to override the DPI auto configuration setting by using System Image Manager (SIM) or by specifying the DPI in an Unattend.xml file. This is useful for scenarios where it is more important to increase the size of the text and UI elements than to achieve the optimal effective resolution of 1024x768. Consider overriding the automatic configuration in the following key scenarios:

* If the effective font size is less than 5 points (common for small PCs that have high-resolution screens).
* For machines that are optimized for Windows Touch input, where target size is important.

For more information on how to optimize machines for Windows Touch, see the OEM Preinstallation Kit (OPK).

Follow these guidelines when you override the automatically configured settings:

* Set the DPI value to 96, 120, or 144.
* Ensure that the Windows effective resolution is greater than 800x600.

SIM is part of the Windows Automated Installation Kit (WAIK), which is available for download from the Microsoft Web site. For more information, see the “System Image Manager Technical Reference,” the WAIK, and the “Unattended Windows Setup Reference,” all of which are listed in “Resources.” You can also find details about overriding the DPI value in the OPK documentation.

For more information about the trade-offs in changing DPI settings, see “Analysis of DPI Settings for Small PC Configurations” later in this paper.

## DPI Configuration through the Control Panel Display Application

A user can change the DPI to suit personal preference by using the Control Panel Displayapplication. To start the application, users right-click the Desktop and select **Screen Resolution**. Figure 1 shows the dialog box that the application displays. As this figure shows, the application warns the user if the resulting effective resolution is less than 1024x768.

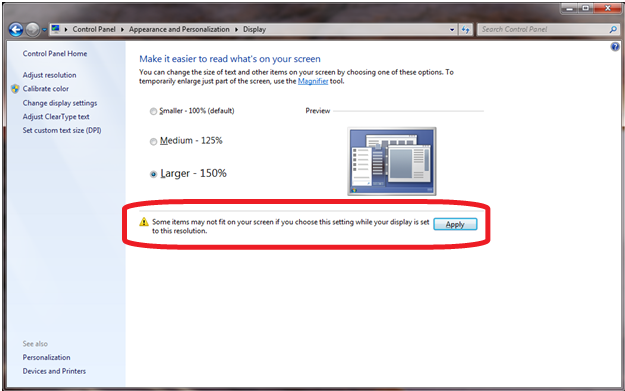


Figure 1. Low-resolution warning

The Display application disables the DPI selection when the physical resolution is less than 800x600, as Figure 2 shows.

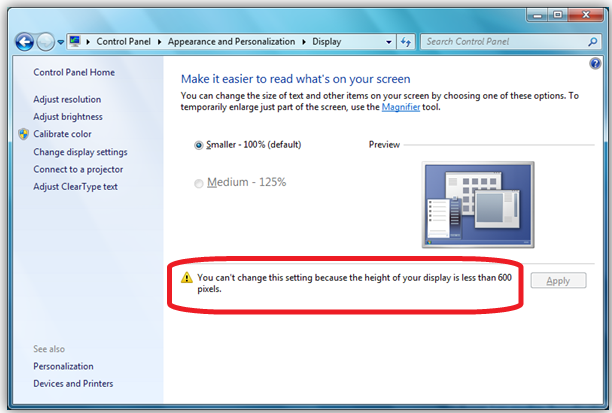


Figure 2. Disabled DPI selections on a low-resolution monitor

The policy warnings that appear in the figures are intended to discourage users from accidentally selecting a configuration that results in a low effective resolution.

# Analysis of DPI Settings for Small PC Configurations

To assess the effect of the high-DPI feature, we looked at small PCs that are currently on the market. Table 2 on the following page lists the small PC configurations on the market at the time of our analysis.

The table indicates whether the configuration meets Windows requirements. For configurations that do not meet requirements for both effective font size and effective resolution, three conditions are possible:

* Does not meet minimum Windows requirements.
* Should not override: Text size is small, but acceptable.
* Recommend overriding: Text size is too small.

Table 2. Current Small PC Configurations (May 2009)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Screen size  (inch) | Horizontal resolution | Vertical resolution | Panel DPI | System DPI OOBE | Scale level  (%) | Effective Windows font size | Comments |
| 8.9 | 1024 | 600 | 133 | 96 | 100 | 6.4 | Meets requirements. |
| 10 | 1024 | 600 | 118 | 96 | 100 | 7.3 | Meets requirements. |
| 10.1 | 1024 | 576 | 116 | 96 | 100 | 7.4 | Does not meet Windows minimum requirements because vertical resolution is less than 600. |
| 8 | 1600 | 768 | 221 | 96 | 100 | 3.9 | Recommend overriding because text size is too small. Windows Auto Configuration does not set High DPI because effective screen resolution height will fall below 768. |
| 12.1 | 1280 | 800 | 124 | 96 | 100 | 6.9 | Meets requirements. |
| 8.9 | 1280 | 768 | 168 | 96 | 100 | 5.9 | Should not override because text is acceptable, although small. Windows auto configuration does not set High DPI because effective screen resolution will fall below 768 thresholds. |
| 10.2 | 1024 | 600 | 116 | 96 | 100 | 7.4 | Meets requirements. |

##### Does not meet Windows minimum requirements

In the table, notice the row that shows a 10.1-inch display at 1025x576. The Vertical Resolution column is highlighted in red because Windows requires a minimum vertical resolution of 600 and this system has only 576. Therefore, this display does not meet the minimum requirements for Windows, as described in the UX Guide. Because the screen height is less than 600 pixels, users cannot change the DPI setting.

Windows does not address issues that might occur if the effective resolution is less than 800x600.

##### Should not override: Text size is small, but acceptable

In the table, notice the row that shows an 8.9-inch display at 1280x768. Although the native DPI of the display is 146, the DPI auto configuration does not set a DPI value that causes an effective resolution below 1024x768 and therefore the default DPI is 96, or 100 percent, highlighted in yellow. At this system DPI setting, the effective Windows font size is 5.9 points, which is acceptable. Vendors should not override the automatically configured default on such a system.

##### Recommend overriding DPI auto configuration

When the effective font size falls below 5 points, the OEM should override the automatically configured value to increase the effective size of text. The PC with an 8–inch display that has a native resolution at 1600x768 shows this situation. The case study in the following section discusses this example in more detail.

## Case Study: Trade-offs of Low Effective Resolution

To better show whether to accept the automatic DPI configuration in Windows 7 or to force high-DPI settings by using the Unattend.xml file, we present a case study. In this example, the PC has a native resolution of 1600x768 and a diagonal screen size of 8 inches and therefore provides 221 DPI horizontally.

By default, the Windows auto configuration feature sets the physical resolution for the example PC at 96 DPI. Windows uses this setting because the next DPI setting of 125 percent results in an effective resolution of 1280x614 and the vertical resolution of 614 DPI is below the optimal height of 768.

However, OEMs might consider 100-percent DPI less than optimal for a 221-DPI screen because text and icons look very small when they are rendered at this size. For this case study, we configured high DPI at 125 percent (120 DPI) and 150 percent (144 DPI), as described in the following two sections.

### 125-Percent DPI Case Analysis

For the machine in this case study, the effective screen resolution at the 125-percent setting is 1280x614. The ratio of system DPI to native DPI is 54 percent, which makes text appear at 54 percent of its actual size. Therefore, the Windows effective text size is 4.8 points (9 points x 0.54), which is below the minimum suggested text size.

The Windows UI looks best at a minimum effective resolution of 1024x768, although it will work at an effective resolution of 800x600. Therefore, the layout of some Windows UI components might be affected and in some cases the UI is visible only if the user uses the scroll bars.

When effective resolution falls below 1024x768, third-party applications are the main concern. Some applications warn users but continue to run. Users can dismiss the warning, but the usability of the application might be compromised because it is designed for an effective resolution of 1024x768.

Some other applications, however, do not run when the resolution is too low. End users have no workaround except lowering the DPI to obtain the full desktop resolution (in this case study, it is 1600x768).

Most applications work correctly at an effective resolution of 800x600 because the UX Guidelines have specified this requirement for previous Windows versions. We recommend that OEMs that choose an effective resolution of 800x600 carefully test any bundled applications to ensure a high-quality out-of-box experience (OOBE).

### 150-Percent DPI Case Analysis

At the 150-percent setting, the effective screen resolution for the machine in this case study is 1066x512. The ratio of system DPI to native DPI is therefore 64 percent, so that text appears at 64 percent of its actual size and the effective font size is 5.8 points (9 points x 0.64). In our analysis, text size appeared acceptable when we compared it to the 100-percent and 125-percent DPI settings even though the effective screen resolution falls below the 800x600 threshold.

If users encounter issues in this configuration, their only workaround is to lower the DPI settings. However, many users are unaware of this workaround, so this issue could generate support calls. Third-party applications might also exhibit problems with the 150-percent DPI setting because the vertical resolution of 512 is less than the minimum configuration that the Windows UI guidelines specify.

## Summary of Analysis

Table 3 summarizes the effective resolution and effective font size for each DPI setting for the sample 8‑inch PC that has a native resolution of 1600x768.

Table 3. Effective Resolution and Text Size for Sample PC

|  |  |  |
| --- | --- | --- |
| DPI setting | Effective resolution | Height of 9-point text |
| 100% (96 DPI) | 1600x768 | 3.9/72 inch |
| 125% (120 DPI) | 1280x614 | 4.8/72 inch |
| 150% (144 DPI) | 1066x512 | 5.8/72 inch |
| 200% (198 DPI) | 800x384 | 7.8/72 inch |

The table clearly shows the trade-off—optimal text size causes unacceptable effective screen resolution. Specifically:

* 100 percent is optimal for Windows and application compatibility but unacceptable for text size.
* 125 percent is not optimal for Windows, but Windows will work at this setting. Some third-party applications might have compatibility issues, and text size is smaller than on typical systems.
* 150 percent is good for text size but unacceptable for Windows compatibility.

Based on the data in the table, 125 percent is the most reasonable choice because the effective resolution should never be less than 800x600.

# Recommendations and Guidelines

On some small PCs, manufacturers face trade-offs of text size versus application compatibility when they configure high DPI.

If effective resolution falls below 1024x768, the Windows auto-configuration feature ensures application compatibility first and therefore opts for smaller text. However, OEMs can opt for larger text size over application compatibility. If so, they should test preinstalled software for incompatibilities and inform end users about the trade-offs.

At effective resolutions below 1024x768, the user experience becomes less optimal, but is still acceptable. We do not recommend display configurations that result in an effective resolution below 800x600 in either direction. These parameters are the required minimum for Windows and for Windows UI guidelines.

On small PCs that have such issues, OEMs should carefully consider the effects of their chosen defaults and inform end users of the trade-offs so that users can determine whether to change these defaults. In Windows 7, users can change DPI by right-clicking the screen, selecting **Screen resolution**, and logging off, instead of restarting the system as Windows Vista® requires.

# Resources

#### MSDN

High DPI

<http://msdn.microsoft.com/en-us/library/dd464646(VS.85).aspx>

Tutorial: Writing High-DPI Win32 Applications

<http://msdn.microsoft.com/en-us/library/dd464659(VS.85).aspx>

Writing High-DPI Win32 Applications

<http://msdn.microsoft.com/en-us/library/dd464660(VS.85).aspx>

Windows User Experience Interaction Guidelines

<http://msdn.microsoft.com/en-us/library/aa511258.aspx>

#### WHDC Web site

High DPI and Windows 7 (WinHEC 2008 Conference Presentation)

<http://download.microsoft.com/download/5/E/6/5E66B27B-988B-4F50-AF3A-C2FF1E62180F/GRA-T582_WH08.pptx>

#### Microsoft Web site

Microsoft OEM Preinstallation Kits

<http://www.microsoft.com/oem/sblicense/OPK/default.mspx>

#### Specifications

VESA E-EDID Implementation Guide

<http://www.vesa.org>

#### Engineering Windows 7 Blogs

Follow-up on High DPI Resolution

<http://blogs.msdn.com/e7/archive/2008/09/13/follow-up-on-high-dpi-resolution.aspx>

More Follow-up to discussion about High DPI

<http://blogs.msdn.com/e7/archive/2008/09/16/more-follow-up-to-discussion-about-high-dpi.aspx>

#### Microsoft TechNet

Windows Automated Installation Kit (Windows AIK)  
<http://technet.microsoft.com/en-us/library/cc748933.aspx>

Windows System Image Manager Technical Reference  
<http://technet.microsoft.com/en-us/library/cc722301.aspx>

Unattended Windows Setup Reference  
<http://technet.microsoft.com/en-us/library/cc722187.aspx>