# Namespace Usage Guidelines for the Windows Vista File System

Anshul Rawat
Microsoft Corporation
June 2007

**Applies to:**

- Windows Vista®

**Summary:** This white paper is intended for application developers and IT professionals. It provides guidelines for using the Windows Vista User Profile namespace in your applications. Windows Vista has an improved file system namespace that enables users to find and organize data easily. The file system namespace also enables application developers to distinguish between application-managed data and end-user-managed data, between private data and shared data, and between computer-dependent data and computer-independent data. The new file system namespace provides a platform that leverages features already available in Windows (for example, folder redirection) and improves the functionality of features in user scenarios (for example, improved data roaming, data migration, and data backup).

# Legal Notice

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.

MICROSOFT MAKES NO WARRANTIES, EXPRESS OR IMPLIED, 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, e-mail address, logo, person, place or event is intended or should be inferred.

Microsoft does not make any representation or warranty regarding specifications in this document or any product or item developed based on these specifications. Microsoft disclaims all express and implied warranties, including but not limited to the implied warranties or merchantability, fitness for a particular purpose and freedom from infringement. Without limiting the generality of the foregoing, Microsoft does not make any warranty of any kind that any item developed based on these specifications, or any portion of a specification, will not infringe any copyright, patent, trade secret or other intellectual property right of any person or entity in any country. It is your responsibility to seek licenses for such intellectual property rights where appropriate. Microsoft shall not be liable for any damages arising out of or in connection with the use of these specifications, including liability for lost profit, business interruption, or any other damages whatsoever. Some states do not allow the exclusion or limitation of liability or consequential or incidental damages; the above limitation may not apply to you.

# Contents

# Introduction

Windows Vista® defines a User Profile namespace, which is a namespace of folders that end users and applications use for storing files. The structure of this namespace is intended to enable end users, applications, administrators, and the operating system to best use the storage features offered by the file system. To decide where files should be stored, you must consider several issues.

- Is the data primarily managed by the end user or through an application?
- Should the data be private to a particular user, shared across all users of the computer, or accessed through a network?
- Does the file need to be written to by the user or should it be read only?
- Should the data remain on a specific computer or should it be roamed to a network?

# Windows Vista User Profile

A user profile is a namespace of user-specific folders isolated for user and application data. In older versions of Windows®, Windows typically stored the user profiles in a root folder called Documents and Settings. This location has changed in Windows Vista® to a more intuitively named folder—the Users folder.

In addition to the root location, the names and locations of individual subfolders in the User Profile have also changed in Windows Vista. Previous versions of user profiles contained a complex folder structure, often nesting folders two to three layers deep. The new folder locations contain fewer nested folders, simplifying navigation, and the new names are reflective of the data contained within them. The following table displays the names of the folders in both Windows Vista and Windows XP. Additionally, the table shows the Windows XP folder locations. (In Windows Vista, all these folders are at Users\%username%\.)

| **Windows Vista folder name** | **Windows XP folder name** | **Description** | **Windows XP folder location** |
|---|---|---|---|
| Contacts | Not applicable | Default location for contacts | Not applicable |
| Desktop | Desktop | Contains desktop items, including files and shortcuts | Documents and Settings\%username%\Desktop |
| Documents | My Documents | Default location for documents | Documents and Settings\%username%\My Documents |
| Downloads | Not applicable | Default location to save all downloaded content | Not applicable |
| Favorites | Not applicable | Internet Explorer Favorites | Documents and Settings\%username%\Favorites |
| Links | Not applicable | Contains Windows Explorer Favorites | Not applicable |
| Music | My Music | Default location for user's music files | Documents and Settings\%username%\My Music |

| Windows Vista folder name | Windows XP folder name | Description | Windows XP folder location |
|---|---|---|---|
| Pictures | My Pictures | Default location for picture files | Documents and Settings\%username%\My Pictures |
| Saved Games | Not applicable | Default location for saved games | Not applicable |
| Searches | Not applicable | Default location for saved searches | Not applicable |
| Videos | My Videos | Default location for video files | Documents and Settings\%username%\My Videos |
| AppData | Not applicable | Default location for application data and binaries (hidden folder) | Not applicable |

The Application Data folder structure, which is used by applications to store per-user application data, has changed in Windows Vista. Previous user profiles did not logically sort data stored in the Application Data folder, making it difficult to distinguish data that belonged to the computer from data that belonged to the user. Windows Vista addresses this issue by creating a single AppData folder under the Users profile (Users\%username%\AppData) which contains three subfolders: Roaming, Local, and LocalLow.

Windows Vista uses the Local and LocalLow folders for application data that should not roam with the user when they log onto another computer. Usually, this data is either computer-specific or too large to roam. The AppData\Local folder in Windows Vista is the same as the Documents and Settings\%username%\Local Settings\Application Data folder in Windows XP.

Windows Vista uses the Roaming folder for application-specific data, such as custom dictionaries, which are computer-independent and should roam with the user profile. The AppData\Roaming folder in Windows Vista is the same as the Documents and Settings\%username%\Application Data folder in Windows XP.

## Windows Vista User Profile vs. the Windows XP User Profile

It is important to understand how each new user-profile folder maps to its predecessor in Windows® XP, to ensure complete profile interoperability between Windows Vista® and Windows XP. The following tables map the new Windows Vista profile folder to its Windows XP predecessor, where applicable.

| Windows Vista profile location<br>Users\%username%\... | Windows XP profile location<br>Documents and Settings\%username%\... |
|---|---|
| …\AppData\Roaming | Application Data |
| …\AppData\Local | Local Settings |
| …\AppData\Local | Local Settings\Application Data |

| **Windows Vista profile location**<br>**Users\%username%\...** | **Windows XP profile location**<br>**Documents and Settings\%username%\...** |
|---|---|
| …\AppData\Local\Microsoft\Windows\History | Local Settings\History |
| …\AppData\Local\Temp | Local Settings\Temp |
| …\AppData\Local\Microsoft\Windows\Temporary Internet Files | Local Settings\Temporary Internet Files |
| …\AppData\Roaming\Microsoft \Windows\Cookies | Cookies |
| …\AppData\Roaming\Microsoft\Windows\Network Shortcuts | Nethood |
| …\AppData\Roaming\Microsoft\Windows\Printer Shortcuts | PrintHood |
| …\AppData\Roaming\Microsoft\Windows\Recent | Recent |
| …\AppData\Roaming\Microsoft\Windows\Send To | SendTo |
| …\AppData\Roaming\Microsoft\Windows\Start Menu | Start Menu |
| …\AppData\Roaming\Microsoft\Windows\Templates | Templates |
| …\Contacts | Not applicable |
| …\Desktop | Desktop |
| …\Documents | My Documents |
| …\Downloads | Not applicable |
| …\Favorites | Favorites |
| …\Music | My Music |
| …\Videos | My Videos |
| …\Pictures | My Pictures |
| …\Searches | Not applicable |
| …\Links | Not applicable |
| …\Saved Games | Not applicable |

| **Windows Vista Public profile location**<br>**Users\Public\** | **Windows XP profile location**<br>**Documents and Settings\All Users\...** |
|---|---|
| …\Desktop | Desktop |
| …\Documents | My Documents |
| …\Downloads | Not applicable |
| …\Favorites | Favorites |
| …\Music | My Music |
| …\Videos | My Videos |
| …\Pictures | My Pictures |

| **Windows Vista shared application data location** | **Windows XP shared application data location** |
|---|---|
| C:\ProgramData | C:\Document and Settings\All Users\Application Data |
| C:\ProgramData\Microsoft\Windows\Start Menu | C:\Document and Settings\All Users\Start Menu |
| C:\ProgramData\Microsoft\Windows\Templates | C:\Document and Settings\All Users\Templates |

## Windows Vista Public Profile and Shared Application Data Locations

Previous versions of Windows® had an All Users profile that enabled users to share data with other users on the computer or to add common user data to user profiles, without editing each user's profile independently. When the user logged on, Windows also merged the contents of the desktop and the Start Menu folder under the All Users profile with the contents of the user profile.

For example, adding a shortcut to the desktop of the All Users profile resulted in every user receiving the shortcut on their desktop when they logged on.

In Windows Vista®, the All Users profile is renamed Public and has a folder structure similar to all other Windows Vista profiles except for the Public AppData folder.

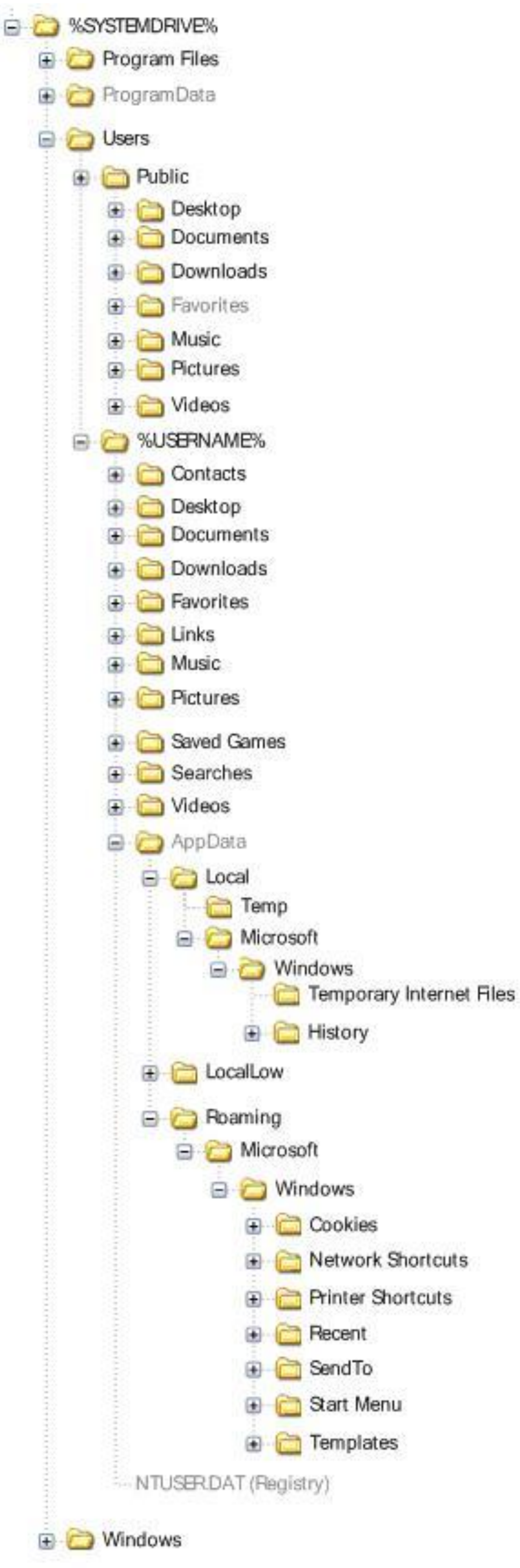
**Note** The All Users\Application Data portion of the All Users profile in Windows XP, which provided a location for applications to store shared application data common to all users, has also been moved in Windows Vista. It is now located at the root of the system drive as a hidden peer of the Program Files and the Users folders. This is the location where all shared application data should be stored across all user accounts.

Windows Explorer will continue to merge specific folders in the Public profile (such as the Desktop and Start Menu folders) with regular user profiles at logon. As in Windows XP, the Public profile does not have a user registry because Windows does not load this profile. Therefore, Windows writes all shared settings to the HKEY_LOCAL_MACHINE hive of the registry.

## Complete Windows Vista File System Namespace

The following table and image show a full conceptual view of the top level of the Windows Vista® file system namespace.

| **Windows Vista folder name** | **Windows XP folder name** | **Description** |
|---|---|---|
| Program Files | Program Files | Default Location for Shared Application files and binaries |
| ProgramData | Documents and Settings\All Users\Application Data | Default Location for Shared Application data and settings |
| Users | Document and Settings | Default location for all User Account profiles |
| Users\Public | Documents and Settings\All Users | Default location for user data shared between all users |
| Users\%username% | Documents and Settings\%username% | Default location for each user's individual profile |
| Windows | Windows | Default Windows OS installation folder |


%SYSTEMDRIVE%
Program Files
ProgramData
Users
Public
Desktop
Documents
Downloads
Favorites
Music
Pictures
Videos
%USERNAME%
Contacts
Desktop
Documents
Downloads
Favorites
Links
Music
Pictures
Saved Games
Searches
Videos
AppData
Local
Temp
Microsoft
Windows
Temporary Internet Files
History
LocalLow
Roaming
Microsoft
Windows
Cookies
Network Shortcuts
Printer Shortcuts
Recent
SendTo
Start Menu
Templates
NTUSER.DAT (Registry)
Windows

# Defining Data Categories and Their Storage Locations in the File System

All the data on a computer falls into several categories. This section explains this categorization and clarifies where data from each category should be stored.

## End-User-Managed Data vs. Application-Managed Data

End-user-managed data is typically data that a user might interact with directly (for example, documents, music files, pictures, and so on) and should go in the storage locations that users consider “theirs”. Windows® makes these locations discoverable and efficient to access by providing entry points in the user interface (UI). For example, the Windows Explorer namespace (in the form of desktop icons, the folder tree, and the breadcrumb bar), the Start menu, and the File, Open, and Save dialog boxes are all designed to make it easy for the user to access these locations quickly. For the purpose of this document, we will call these folders user-data folders.

Visible folders in the user’s profile and the public profile exist to store end-user-managed data.

C:\Users\<*username*>\<ContentType> (for example, C:\Users\<*username*>\Documents)

C:\Users\Public\<ContentType> (for example, C:\Users\Public\ Documents)

Application-managed data, which, typically, a user never interacts with directly (for example, Microsoft Office Word toolbar settings), should go in folders that are less visible in the UI. These folders are hidden and placed in locations that users normally do not encounter. There are per-user and public forms of these folders as well as roaming, local, and low-rights forms.

C:\Users\<*username*>\AppData (hidden)

C:\ProgramData (hidden)

Because it is common for certain types of files to be end-user managed *and* application managed, consider the following characteristics for each of these types of data and select the most appropriate storage location for these files.

### Characteristics of End-User-Managed Data

Data that is managed by end users has the following characteristics:

- Users need to find this data easily, either by browsing or searching.
- Users use Windows Explorer to organize (store in folders), move, rename, and copy this data.
- Users open and save these items by using the File, Open, and Save dialog boxes.
- Users consider this data their own; something that they either created or are interested in.

Examples of end-user-managed files are:

- Microsoft Office documents such as Word documents, Excel spreadsheets, or PowerPoint® presentations.
- Media files such as photos, music, videos, and recorded television programs.

### Characteristics of Application-Managed Data

Application-managed data has the following characteristics:

- Applications create and manage the lifetime of this data.
- Users might not be aware that data in these files is created by the application.

- Users do not rename the files or the folders managed by the application because it might interfere with the correct operation of the application.
- The application provides the majority of the management functions for the data (backup, copy, delete).
- The application provides access to the storage location of the data, for example, an Open File command that helps users find the data.
- Storing this data with end-user-managed data would clutter that space, making users less efficient or annoyed by its presence because it is not “theirs”.

Examples of application-managed files are:

- Outlook® PST files. Outlook offers a management UI for these files that handles the rare case when users must perform management tasks on the data.
- Windows registry (NTUSER.DAT)
- Temp/cache files and Temporary Internet Files
- Application settings stored in files like .ini files or databases
- Log files

## Per-User Data vs. Shared-User Data

All end-user-managed data can be further categorized into data that might belong to a specific user, which only they can directly interact with, or data that is shared with all users on a machine.

### Storing Per-User Data

Data that should be private to a user and maintained separately from other users of the same computer should be stored in user-data folders within the user’s own profile. The per-user folders are secured through Access Control Lists (ACLs) so that, under default circumstances, each user’s storage space is private.

### Storing Shared-User Data

Data that should be accessible to multiple users of the computer or other users or devices on the network should be stored within the public folders located at C:\Users\Public. This location uses a public/private sharing model, so only data that should be accessible to everyone should be placed there. Files that require discretionary access should remain in the users’ profiles and should have the appropriate ACL set on them.

**Note** The Sharing feature in Windows Vista also enables a user to share access to files and folders stored within a profile. The user can choose to share an individual file or an entire folder within their user profile without moving it to the public folder. When a file or a folder is shared to a user, only that file or that folder is shared while the rest of the profile is still secure. This functionality enables the user to control which users access specific shared data.

## Program Files vs. Per-User or Shared-Application Data

Windows Vista provides specific locations in the file system to store programs and software components, application data that is shared between all users on a computer, and application data that is specific to a user. These locations are, respectively:

- C:\Program Files:
  - Storage location for programs and software components.
  - Users should not store any user data or application data here because of the security permissions configured for this folder. Standard users do not have rights to write data to this location.

- C:\ProgramData:
    - Storage location for application data that is to be shared between all users on that computer.
- C:\Users\<*username*>\AppData:
    - Storage location for per-user application data that is exclusive to a user and should not be shared with any other user on that computer.
    - The AppData location itself has a further hierarchy below it to differentiate computer-dependent application data from computer-independent application data. This hierarchy is discussed in detail in the next section.

## Further Categorization of Application Data

All application-managed data can be further categorized into data that belongs to a specific user and is managed by an application exclusively for that user, or data that can be shared with all users of that computer.

### Storing Per-User Application Data

Windows Vista provides a specific hierarchy for application developers when storing application data within the user's profile. The C:\Users\<*username*>\AppData folder is further subdivided into the following three discrete folder paths:

- AppData\Local folder for storing computer-dependent data
- AppData\Roaming folder for storing computer-independent data
- AppData\LocalLow folder for storing low-rights integrity data

When storing per-user application data within the user profile, you can use the following procedure to determine where to store the data:

1. Determine the type of application data you are storing in the file system. Is it computer-dependent, computer-independent, or low-rights integrity data?

   Based on that information, use one of the following paths to store the data:

    - C:\Users\<*username*>\Appdata\**Roaming**
        - For per-user computer-independent application data.
        - Can be roamed or redirected within an organization by using folder redirection or roaming user profiles.
        - Computer-independent application data that contains a user's customizations and cannot be recreated without explicit user action.
    - C:\Users\<*username*>\Appdata\**Local**
        - For per-user computer-dependent application data.
        - Should not be roamed or redirected within an organization.
        - Application data that contains only temporary or non-critical data that can be discarded without affecting the application's functionality. (For example, log files for debugging purposes.)
        - Application data which, although machine independent, can easily be re-created without explicit user action, as it does not store any user-generated customizations or settings.
    - C:\Users\<*username*>\Appdata\**LocalLow**

        - For per-user low-rights application data.
        - Meant as a location to be used by low-rights applications to store their data.
        - Will not be roamed or redirected in an organization.

2. Create a subfolder under the appropriate AppData folder above using the following convention:
    - Step A: ..\Vendor Name\Major Product Name (Recommended)
        - For example: AppData\Roaming\**Microsoft\Office**
    - Step B: ..\Minor Product Name (Optional)
        - For example: AppData\Roaming\Microsoft\Office\**Groove**
    - Step C: ..\Version + SKU Information (Optional)
        - For example: AppData\Roaming\Microsoft\Office\Groove\**2007 Ultimate Edition**

### Storing Shared Application Data

All application data that must be shared among users on the computer should be stored within the C:\ProgramData folder location. To store data for your application, create a subfolder under the C:\ProgramData\ folder by using the following convention:

- Step A: ..\Vendor Name\Major Product Name (Recommended)
    - For example: C:\ProgramData\**Microsoft\Office**
- Step B: ..\Minor Product Name (Optional)
    - For example: C:\ProgramData\Microsoft\Office\**Groove**
- Step C: ..\Version + SKU Information (Optional)
    - For example: C:\ProgramData\Microsoft\Office\Groove\**2007 Ultimate Edition**

## Using and Extending the User Profile Namespace

By default, the user profile root (C:\Users\<*username*>) contains folders that help users better organize their data (such as documents, pictures, and music files) without cluttering their root profile folder. This organization of data also helps enable features within Windows Vista® that operate based on folders and the items stored within them. Examples include:

- Folder redirection. You can relocate a folder from within the user's profile to another hard disk location or to a network location. For example, if you run out of disk space, you can redirect your Pictures folder to D:\Pictures, D being a hard disk drive that has the storage capacity you need.
- Search scopes. You can search in a specific folder or set of folders. Indexer crawl scopes are based on a set of folders and on rules that can be applied to items in those folders.
- Security permissions. These are typically applied using inheritance. This enables the items in the folder to set their security based on the security configuration of the folder.
- File sharing. This enables users to share a specific folder and the files within that folder.
- File syncing features. You can synchronize files stored in multiple locations. For example: through Foldershare.com™ and Microsoft® Office Groove™.
- Folder UI customization. This includes view modes (thumbnails versus details), visible columns, command, and pane visibility. For example, custom templates can be applied to folders for storing music and pictures to create a better experience with the items within those folders.

To enable such features, application developers should make an effort to reuse the existing folder structure provided by Windows Vista where possible. If you cannot use the existing defined structure and need to extend it with your own folders, you should:

- Extend the folder structure in a manner that matches the user’s need for the data that will be stored in them, as defined in earlier sections (user-managed data vs. application-managed data or per-user data vs. public data).
- Take into account the relevant features of both Windows Vista and the application that might be applicable to that data as defined in earlier sections of this document.

The following sections contain detailed guidelines for using the namespace and descriptions of ways to extend it.

## User Profile Namespace Guidelines

When possible, applications should use an existing folder that Windows Vista has already defined in the User Profile namespace that matches the file types that the application supports. For example:

- For document files, like .docx, .pps, .xls, and .txt files, you should use the Documents folder.
- For music files, like .mp3, .wma, and .rp, you should use the Music folder.
- For video files, like .wmv, .mov, .mpg, and .avi, you should use the Videos folder.

To store a new data type that does not map to any of the existing folders in a user's profile, applications should create a new data folder either at the root of the user's profile (for example, C:\Users\<*username*>\<*new datatype*>) or at the root of the Public profile (for example, C:\Users\Public\<*new datatype*>) The choice of location will depend on whether the data is per-user data or should be shared among all users on the computer. This will also ensure that it is a peer of other data folders like Documents and easily discoverable and accessible by the user.

Applications should use the extensible Known Folder system to define their folder as a well known Windows folder to enable features like redirection or to enable other applications to find that folder. For more details on the Known Folder mechanism, please see the Known Folder API documentation on MSDN at .

Application creators should publish a specification that explains the intended use of their folder and publish the KNOWNFOLDERID in a Software Development Kit (SDK) to enable other applications to access it programmatically if appropriate.

When reviewing all the data your application creates, you should have a clear separation between user-managed data and application-managed data. Applications should not create application-managed files in the profile root as that can cause unnecessary clutter. This type of data should be stored below the AppData folder within the user’s profile as described earlier.

To avoid cluttering the user namespace, applications should not encourage the creation of user-managed files in the root of the user's profile folder (C:\Users\<*username*>). They should encourage storing the data within the subfolders as mentioned earlier. There are group policies in Windows Vista that can prevent the user from creating items at the root of the user profile in Windows Explorer and your application might break if you encourage that behavior.

## Creating Folders at the Drive Root

An application should not create files or folders at the root of the system drive and should use the folder structure as described in previous sections to store their program and software components, application data, and user data.

An application can enable relocation of folders it manages by using the Folder Redirection feature in the Windows® operating system. This enables the end user to make the decision to relocate that data if they want to, such as to the root of the disk, to another local disk, or to a network location. Folder redirection was part of Windows 2000 and Windows XP, but was limited to only the My Documents, My Pictures, Desktop, Application Data, and Start Menu folders. Windows Vista allows a larger selection of known folders to be redirected. Applications can also enable a custom folder to be redirected in their folder definition when they create the folder using the Known Folder APIs as described on MSDN at http://msdn2.microsoft.com/en-us/library/ms645247.aspx.

## Accessing Well Known Folder Locations

Windows Vista® provides application developers a programmatic way to access well known Windows® locations by using the Known Folder APIs and KNOWNFOLDERIDs. KNOWNFOLDERIDs are unique to each well known Windows folder location and to the SHGetKnownFolderPath function as defined on MSDN at http://msdn2.microsoft.com/en-us/library/ms647783.aspx. This API support enables an application to get the current path for a specific well known location, because the paths might have changed since the last time the folder was accessed by the application (for example, due to folder redirection).

To see a list of all well known locations that can be accessed using the Known Folder system, review the KNOWNFOLDERID topic on MSDN at http://msdn2.microsoft.com/en-us/library/bb407328.aspx.

**Important** An application should never try to hard code or save an absolute path to a well known Windows location or to files stored within. The path to the data can change over time due to features like folder redirection or may vary from one computer to another. As a result, the application might break or its data might become inaccessible. Using the Known Folder APIs ensures that you are always able to get to your data.

If, for some reason, you are unable to use the APIs to get programmatic access to a folder, you can use predefined environment variables in Windows to build a relative path to the data you are accessing. The following is a list of variables and the default locations they point to, which you can use to build a relative path:

- Legacy variables supported in Windows Vista:
    - %USERPROFILE% => ..\Users\<*username*>\
    - %APPDATA% => ..\Users\<*username*>\AppData\Roaming\
    - %TEMP% => ..\Users\<*username*>\AppData\Local\Temp\
- New variables added for Windows Vista:
    - %LOCALAPPDATA% => ..\Users\<*username*>\AppData\Local
    - %PUBLIC% => ..\Users\Public
    - %PROGRAMDATA% => %systemdrive%\ProgramData

## SHGetKnownFolderPath Function vs. the SHGetFolderPath Function

SHGetKnownFolderPath (http://msdn2.microsoft.com/en-us/library/ms647783.aspx) is a new function in Windows Vista that is a superset of SHGetFolderPath (http://msdn2.microsoft.com/en-us/library/ms647764.aspx), which was included in earlier versions of the Windows Shell. The SHGetFolderPath function, although a legacy function, is still supported in Windows Vista. SHGetFolderPath provides programmatic access to a subset of the well known folder locations in Windows Vista through a set of constants called constant special item ID list (CSIDL) constants. These constants are documented on MSDN at http://msdn2.microsoft.com/en-us/library/ms649274.aspx.

To programmatically access all Windows Vista well known folder locations, including well known folders that your application created, use the SHGetKnownFolderPath function.

You can use the SHGetFolderPath function if all of the following are true:

- You are building an application that must work on both Windows Vista and Windows XP.
- You do not need to define a new data folder type for your application.
- To store data that is accessible on both Windows Vista and Windows XP, you are using only the well known folder locations that are supported on Windows XP.