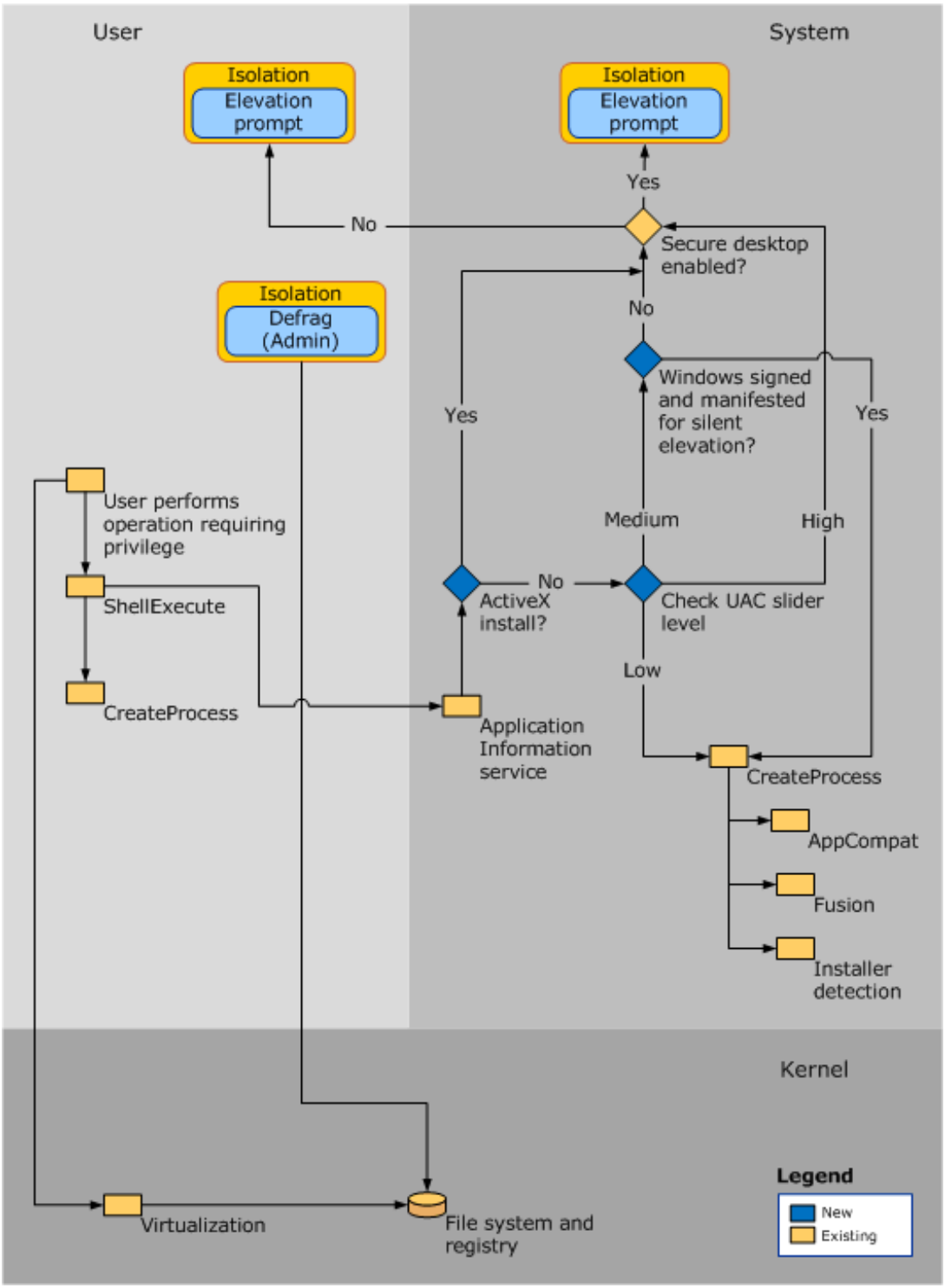
UAC Architecture

TN technet.microsoft.com/en-us/library/dd835540(v=ws.10).aspx

The following diagram details the UAC architecture.



Component	Description
User	

User performs operation requiring privilege	If the operation changes the file system or registry, Virtualization is called. All other operations call ShellExecute.
ShellExecute	ShellExecute calls CreateProcess. ShellExecute looks for the ERROR_ELEVATION_REQUIRED error from CreateProcess. If it receives the error, ShellExecute calls the Application Information service to attempt to perform the requested task with the elevated prompt.
CreateProcess	If the application requires elevation, CreateProcess rejects the call with ERROR_ELEVATION_REQUIRED.
System	
Application Information service	A system service that helps start applications that require one or more elevated privileges or user rights to run, such as local administrative tasks, and applications that require higher integrity levels. The Application Information service helps start such applications by creating a new process for the application with an administrative user's full access token when elevation is required and (depending on Group Policy) consent is given by the user to do so.
ActiveX installed	If ActiveX is not installed, the system checks the UAC slider level. If ActiveX is installed, the **User Account Control: Switch to the secure desktop when prompting for elevation** Group Policy setting is checked.

Check UAC slider level	UAC now has four levels of notification to choose from and a slider to use to select the notification level: • High If the slider is set to **Always notify**, the system checks whether the secure desktop is enabled. • Medium If the slider is set to **Default-Notify me only when programs try to make changes to my computer**, the **User Account Control: Only elevate executable files that are signed and validated** policy setting is checked: ○ If the policy setting is enabled, the public key infrastructure (PKI) certification path validation is enforced for a given executable file before it is permitted to run. ○ If the policy setting is not enabled (default), the PKI certification path validation is not enforced before a given executable file is permitted to run. The **User Account Control: Switch to the secure desktop when prompting for elevation** Group Policy setting is checked. • Low If the slider is set to **Notify me only when programs try to make changes to my computer (do not dim by desktop)**, the CreateProcess is called. • Off If the slider is set to **Never notify**, UAC prompting is turned off. Software can be installed and Windows settings can be changed without a notification prompt. **Important** This setting is not recommended. This setting is not the same as setting the **User Account Control: Behavior of the elevation prompt for administrators in Admin Approval Mode** policy setting to **Elevate without prompting**. If you need to use programs that are not certified for Windows 7, see User Account Control: Deploying a Managed Desktop Environment in Windows 7 (http://go.microsoft.com/fwlink/?LinkID=148442) to determine if there are steps that you can take to run the non-compliant programs without turning UAC off.

Secure desktop enabled	The **User Account Control: Switch to the secure desktop when prompting for elevation** policy setting is checked: • If the secure desktop is enabled, all elevation requests go to the secure desktop regardless of prompt behavior policy settings for administrators and standard users. • If the secure desktop is not enabled, all elevation requests go to the interactive user's desktop, and the per-user settings for administrators and standard users are used.
CreateProcess	CreateProcess calls AppCompat, Fusion, and Installer detection to assess if the application requires elevation. The executable file is then inspected to determine its requested execution level, which is stored in the application manifest for the executable file. CreateProcess fails if the requested execution level specified in the manifest does not match the access token and returns an error (ERROR_ELEVATION_REQUIRED) to ShellExecute. For more information, see Requested execution levels later in this section.
AppCompat	The AppCompat database stores information in the application compatibility fix entries for an application.
Fusion	The Fusion database stores information from application manifests that describe the applications. The manifest schema is updated to add a new requested execution level field.
Installer detection	Installer detection detects setup executable files, which helps prevent installations from being run without the user's knowledge and consent. For more information, see Installer detection technology later in this section.
Kernel	
Virtualization	Virtualization technology ensures that non-compliant applications do not silently fail to run or fail in a way that the cause cannot be determined. UAC also provides file and registry virtualization and logging for applications that write to protected areas. For more information, see Virtualization later in this section.
File system and registry	The per-user file and registry virtualization redirects per-computer registry and file write requests to equivalent per-user locations. Read requests are redirected to the virtualized per-user location first and to the per-computer location second.

Virtualization

Because system administrators in enterprise environments attempt to secure systems, many line-of-business (LOB) applications are designed to use only a standard user access token. As a result, IT administrators do not need to replace the majority of applications when running Windows 7 with UAC enabled.

Windows 7 includes file and registry virtualization technology for applications that are not UAC compliant and that require an administrator's access token to run correctly. Virtualization ensures that even applications that are not UAC compliant are compatible with Windows 7. When an administrative application that is not UAC compliant attempts to write to a protected directory, such as Program Files, UAC gives the application its own virtualized view of the resource it is attempting to change. The virtualized copy is maintained in the user's profile. This strategy

creates a separate copy of the virtualized file for each user that runs the non-compliant application.

Most application tasks operate properly by using virtualization features. Although virtualization allows a majority of applications to run, it is a short-term fix and not a long-term solution. Application developers should modify their applications to be compliant with the Windows 7 logo program as soon as possible, rather than relying on file, folder, and registry virtualization.

Virtualization is not in option in the following scenarios:

- Virtualization does not apply to applications that are elevated and run with a full administrative access token.
- Virtualization supports only 32-bit applications. Non-elevated 64-bit applications simply receive an access denied message when they attempt to acquire a handle (a unique identifier) to a Windows object. Native Windows 64-bit applications are required to be compatible with UAC and to write data into the correct locations.
- Virtualization is disabled for an application if the application includes an application manifest with a requested execution level attribute.

Requested execution levels

An application manifest is an XML file that describes and identifies the shared and private side-by-side assemblies that an application should bind to at run time. In Windows 7, the application manifest includes entries for UAC application compatibility purposes. Administrative applications that include an entry in the application manifest prompt the user for permission to access the user's access token. Although they lack an entry in the application manifest, most administrative applications can run without modification by using application compatibility fixes. Application compatibility fixes are database entries that enable applications that are not UAC compliant to work properly with Windows 7.

All UAC-compliant applications should have a requested execution level added to the application manifest. If the application requires administrative access to the system, then marking the application with a requested execution level of "require administrator" ensures that the system identifies this program as an administrative application and performs the necessary elevation steps. Requested execution levels specify the privileges required for an application.

Installer detection technology

Installation programs are applications designed to deploy software. Most installation programs write to system directories and registry keys. These protected system locations are typically writeable only by an administrator in Installer detection technology, which means that standard users do not have sufficient access to install programs. Windows 7 heuristically detects installation programs and requests administrator credentials or approval from the administrator user in order to run with access privileges. Windows 7 also heuristically detects updates and programs that uninstall applications. One of the design goals of UAC is to prevent installations from being run without the user's knowledge and consent because installation programs write to protected areas of the file system and registry.

Installer detection only applies to:

- 32-bit executable files.
- Applications without a requested execution level attribute.
- Interactive processes running as a standard user with UAC enabled.

Before a 32-bit process is created, the following attributes are checked to determine whether it is an installer:

- The file name includes keywords such as "install," "setup," or "update."
- Versioning Resource fields contain the following keywords: Vendor, Company Name, Product Name, File Description, Original Filename, Internal Name, and Export Name.
- Keywords in the side-by-side manifest are embedded in the executable file.
- Keywords in specific StringTable entries are linked in the executable file.
- Key attributes in the resource script data are linked in the executable file.
- There are targeted sequences of bytes within the executable file.

Note

The keywords and sequences of bytes were derived from common characteristics observed from various installer technologies.

Note

The **User Account Control: Detect application installations and prompt for elevation** policy setting must be enabled for installer detection to detect installation programs. This setting is enabled by default and can be configured locally by using the Local Security Policy snap-in (Secpol.msc) or configured for the domain, OU, or specific groups by Group Policy (Gpedit.msc).

For general information and an overview of the Windows Installer, see Windows Installer in the MSDN Library (http://go.microsoft.com/fwlink/?LinkId=120410).