**GINA**

A Graphical Identification and Authentication dynamic-link library (DLL). The GINA is a replaceable DLL component that is loaded by the *[Winlogon](https://msdn.microsoft.com/en-us/library/windows/desktop/ms721635(v=vs.85).aspx" \l "_security_winlogon_gly)* executable. The GINA implements the authentication policy of the interactive logon model and is expected to perform all identification and authentication user interactions

**Winlogon**

A part of the Windows operating system that provides interactive logon support. Winlogon is designed around an interactive logon model that consists of three parts: the Winlogon executable, a Graphical Identification and Authentication dynamic-link library (DLL) referred to as the GINA, and any number of network providers.

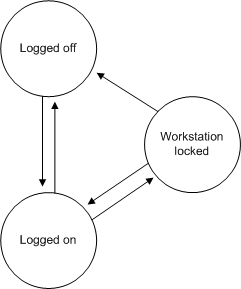
Winlogon States

[*Winlogon*](https://msdn.microsoft.com/en-us/library/windows/desktop/ms721635(v=vs.85).aspx#_security_winlogon_gly) maintains the workstation state that is used by the [*GINA*](https://msdn.microsoft.com/en-us/library/windows/desktop/ms721584(v=vs.85).aspx#_security_gina_gly) to determine what authentication actions are required.

At any point in time, Winlogon is in one of three states:

* [Logged-Off State](https://msdn.microsoft.com/en-us/library/windows/desktop/aa380547(v=vs.85).aspx#logged_off_state)
* [Logged-On State](https://msdn.microsoft.com/en-us/library/windows/desktop/aa380547(v=vs.85).aspx#logged_on_state)
* [Workstation-Locked State](https://msdn.microsoft.com/en-us/library/windows/desktop/aa380547(v=vs.85).aspx#workstation_locked_state)

These three states are shown in the following illustration.



Logged-Off State

When Winlogon is in the logged-off state, users are prompted to identify themselves and provide authentication information. If a user provides correct user account information and no restrictions prevent it, the user is logged on and a shell program (such as Windows Explorer) is executed in the application desktop. Winlogon changes to the logged-on state.

Logged-On State

When Winlogon is in the logged-on state, users can interact with the shell, activate additional applications, and do their work. From the logged-on state, users can either stop all work and log off, or lock their workstations (leaving all work in place). If the user decides to log off, Winlogon will terminate all processes associated with that [*logon session*](https://msdn.microsoft.com/en-us/library/windows/desktop/ms721592(v=vs.85).aspx#_security_logon_session_gly) and the workstation will be available for another user. If, instead, the user decides to lock the workstation, Winlogon changes to the workstation-locked state.

Workstation-Locked State

When Winlogon is in the workstation-locked state, a secure desktop is displayed until the user unlocks the workstation by providing the same identification and authentication information as the user who originally logged on, or until an administrator forces a logoff. If the workstation is unlocked, the application desktop is displayed, and work can resume. If, however, an administrator unlocks the workstation (by providing the identification and authentication information of an administrator account), the processes of the logged-on user are terminated, and Winlogon changes to the logged-off state.

A number of different actions can be performed in each of the Winlogon states. A GINA DLL may implement actions that are not part of the standard Windows operating system. For example, a high security system could automatically lock a workstation every 10 minutes and force users to reauthenticate themselves.

For information about creating desktops and registering a [*secure attention sequence*](https://msdn.microsoft.com/en-us/library/windows/desktop/ms721625(v=vs.85).aspx#_security_secure_attention_sequence_gly) (SAS), see [Initializing Winlogon](https://msdn.microsoft.com/en-us/library/windows/desktop/aa375994(v=vs.85).aspx). For information about time-out operations, see [Supported Dialog Box Service Time-out Operations](https://msdn.microsoft.com/en-us/library/windows/desktop/bb540760(v=vs.85).aspx). For information about sending messages to the GINA while a dialog box is displayed, see [Sending Messages to the GINA](https://msdn.microsoft.com/en-us/library/windows/desktop/aa380133(v=vs.85).aspx). For information about support functions, see [Winlogon Support Functions](https://msdn.microsoft.com/en-us/library/windows/desktop/aa374731(v=vs.85).aspx" \l "winlogon_support_functions)

# Initializing Winlogon

When [Winlogon](https://msdn.microsoft.com/en-us/library/windows/desktop/aa380542(v=vs.85).aspx) initializes, it registers the CTRL+ALT+DEL [secure attention sequence](https://msdn.microsoft.com/en-us/library/windows/desktop/ms721625(v=vs.85).aspx#_security_secure_attention_sequence_gly) (SAS) with the system, and then creates three desktops within the WinSta0(window station object) window station.

Registering CTRL+ALT+DEL makes this initialization the first process, thus ensuring that no other application has hooked that key sequence.

WinSta0 is the name of the window station object that represents the physical screen, keyboard and mouse. Winlogon creates the following desktops in the WinSta0 object.

|  |  |
| --- | --- |
| **Desktop** | **Description** |
| Winlogon desktop | This is the desktop that Winlogon and [GINA](https://msdn.microsoft.com/en-us/library/windows/desktop/ms721584(v=vs.85).aspx#_security_gina_gly) use for interactive identification and authentication, and other secure dialog boxes. Winlogon automatically switches to this desktop when it receives SAS event notification. |
| Application desktop | Each time a user successfully logs on, an application desktop is created for that [logon session](https://msdn.microsoft.com/en-us/library/windows/desktop/ms721592(v=vs.85).aspx#_security_logon_session_gly). The application desktop is also known as the default or user desktop. This desktop is where all user activity takes place. The application desktop is protected; only the system and the interactive logon session have access to it. Note that only a particular instance of the logged-on user has access to the desktop. If the interactive user activates a process using the service controller, that service application will not have access to the application desktop. |
| Screen saver desktop | This is the current desktop when a screen saver is running. If a user is logged on, both the system and the interactive logon session have access to the desktop. Otherwise, only the system has access to the desktop. |

As the owner of these desktops, Winlogon can switch the current, or visible, desktop to any of the three desktops and allow the GINA access to this functionality. In general, GINA developers will not change the current desktop because Winlogon sets the desktop appropriately before communicating with the GINA. The description of each GINA function indicates which desktop is current for that call.

**Boot Phase Explorer Initialization – Phase Activity**

The list of activities for this phase are –

* Explorer.exe starts
* Desktop Window Manager starts
* The desktop is displayed for the first time
* Auto-start (Run-keys) applications are launched

**Boot Phase Explorer Initialization – Potential Issues**

Explorer Initialization is usually a quick phase, taking only a couple of seconds but it may become longer with high disk contention. As services, antivirus, and auto-start applications compete, the system disk I/O becomes important.

Additionally, Explorer Initialization is a CPU intensive phase. CPU competition with other processes may delay the user experience. Run and RunOnce registry keys may introduce some of these competing processes.

**Conclusion**

Explorer Initialization is a fairly simple phase where you main goal is to reduce CPU and Disk utilization so that the shell can be started as soon as possible.

**Different source**

**Explorer Installation:**

This metric captures the time from the end of the Winlogon phase until the Windows Shell (Explorer.exe) reports an initialized Start Screen. It includes the start of the userinit.exe process, which in turn starts Explorer.exe. It also includes the initiation of startup applications that are stored in **RunOnce** registry keys.

**Typical Influencing Factors**

In the absence of **RunOnce** applications, most of the time in this phase should be spent initializing the Explorer process. Explorer reads a number of libraries and data files into memory during its initialization process. Contending I/O from other running components can interfere with and delay this activity.

**Analysis and Remediation Steps**

Specific issues are usually generated for longer durations of the Explorer Initialization phase. You can gain more insight by opening WPA to the time interval of the Explorer Initialization activity. Avoid placing applications in the **RunOnce** key on a recurring basis because it delays Explorer Initialization.

**Different source**

## ****ExplorerInit: Explorer Initialization****

The ExplorerInit subphase begins when Explorer.exe starts. During ExplorerInit, the system creates the desktop window manager (DWM) process, which initializes the desktop and displays it for the first time.

### Winlogon Initialization [WinLogonInit]

* Winlogon.exe start with User Logon Screen .
* Service control manager starts services.
* Group policy script run during winlogon initialization.

### Explorer Initialization [ExplorerInit]

* This sub-phase starts when Winlogon process passes control to Explorer process [Explorer.exe].
* Subsystem creates Desktop Windows Manager [DWM] process.
* DWM initializes desktop and display for the first time on screen