

Description of updates to APIs that enable Windows-based applications to retrieve historically accurate time stamps

- This article describes updates to APIs that enable Windows-based applications to retrieve historically accurate time stamps.

Go to Microsoft's website for complete details: <http://support.microsoft.com/kb/960417>

Hotfix Download Available

[View and request hotfix downloads](#)

What time functions are updated? `loadTOCNode(2, 'moreinformation')`; The following time functions are updated by this update:
`SystemTimeToTzSpecificLocalTime()` Converts a time in Coordinated Universal Time (UTC) to a specified time zone's corresponding local time.
`TzSpecificTimeToSystemTime()` Converts a local time to a time in Coordinated Universal Time (UTC).

Note The functions that are listed here use a `TIME_ZONE_INFORMATION` structure that specifies the `START DATE` and the `END DATE` for daylight saving time (DST). By default, the current DST rules are used when no such structure is provided. **When to use these functions in an application** `loadTOCNode(2, 'moreinformation')`; Most date and time stamps that are created and displayed in Windows and in many applications are stored as UTC. Then, they are rendered in local time by using system APIs. Examples of these include Windows file time stamps, Outlook sent and received dates, and event logging time stamps. Not all built-in, Windows-based applications use these newly updated APIs. However, they are available to all applications, including third-party applications. These functions are useful in many scenarios, such as a user auditing scenario in which the ability to use DST rules for previous years on a current Windows service pack is important.

Generally, if an application requires historically accurate time stamps, these functions should be used.

Go to Microsoft's website for complete details: <http://support.microsoft.com/kb/960417>

<http://kb.actforadvisors.com/questions.php?questionid=171>