

# SOFTDATE

## Test z/OS-based Programs with a Simulated Past or Future System Date

SoftDate™ was developed to assist in the increasing need to have e-Application 21st Century Calendar Compliance, easily integrating as a standard step in the final stages of program testing.

Many programs have features that are activated only under certain date and time conditions, such as:

- End of month, end of quarter or end of year
- Policy anniversary
- Bond maturity
- Overdue payment
- License expiry, and so on

This type of date-dependent processing is virtually universal across a wide variety of applications found in all industry segments. Such programs cannot be fully tested without running them under simulated current date and time conditions that will trigger these special processing paths.

SoftDate also allows you to cater for special requirements such as:

- Re-running failed production jobs on the same apparent system date
- Simulating correct local date and time for users in widely dispersed geographic locations and time zones, independent of the actual physical location of their data center – a valuable feature in data center consolidation

SoftDate gives you a simple mechanism to test date-dependent logic in your programs without requiring cumbersome and expensive procedures such as 'IPL'ing a test LPAR with the required date. Applications that contain time-sensitive logic also require testing for all possible date and time conditions. SoftDate provides the ability to correctly test against these predicted time events.

SoftDate can be turned on at the individual job step, TSO, CICS or IMS user level without impacting other jobs or users running in the same system.

### How Does SoftDate Work?

SoftDate works by dynamically front-ending system routines used to obtain the current date and time. This front-ending occurs only in regions where it is required. Other regions and system processes that do not require SoftDate are completely unaffected.

In cases where it is needed, SoftDate lets the system routine return the current date and time and then applies the adjustment factor requested by the user.

### **The SoftDate “Minimum Impact” Approach**

Past approaches to date simulation have invariably involved creating modified versions of key system routines. These patched routines then allow the simulation product to gain control whenever any program running in the z/OS® system requests the system date or time. The product then typically scans its own internal tables to determine if the caller requires a simulated date, and returns the simulated value if so, otherwise passing the call off to the standard system routine. This is a high impact approach because:

- Every date request issued by any program running in the z/OS system will incur overhead. Date requests are issued by many system components as well as application programs and there can easily be thousands or more of these per second in a busy z/OS system. The additional overhead introduced by the simulator will be a substantial cost, particularly when you consider at times there could be no or few actual users of it, but yet every process running on the z/OS system is still penalized.
- The entire z/OS system will be exposed to any stability problems (bugs) that may exist in the product.

By contrast, the SoftDate approach confines its impact to only those regions and users that actually require it. Apart from two trivial exceptions, there is zero impact from SoftDate on regions which are not actively using it. There are no permanent changes to any system routines, and any region that is running without SoftDate active will execute only standard, unmodified system code.

### **Other SoftDate Benefits**

There are a variety of unique SoftDate features and options that supply benefits relating to increased productivity and the reduction of risk:

- Supports Java workloads running under z/OS Unix System Services and WebSphere® Application Server
- Has a powerful and intuitive ISPF front end for defining rules to specify jobs that require SoftDate, meaning you can test with SoftDate without having to make JCL changes
- Offers unrivalled flexibility in rules specification and security controls
- Assists Sarbanes-Oxley compliance for testing date compliance and to protect profit by pre-empting risks of date errors and related issues
- Assists Basel II compliance with risk management testing, protecting your data assets and those transactions running with them

- Does not make permanent changes to critical system routines
- Is under a program of active support and development