

SoftDate and Total Cost of Ownership

Softdate is a z/OS-based Date and Time Simulator (DTS), which simulates the system clock where applications function at a different time or date.

An overview of Total Cost of Ownership (TCO) with Softdate

The Total Cost of Ownership (TCO) of Softdate requires informed comparisons as one may be measuring in a very large centralised environment or a smaller environment where distributed systems are being consolidated with many variable factors; i.e. a dozen web apps with thousands of clients or some hundreds of web apps with millions of clients. Each case will differ, as date-dependent processing logic can be light through heavy.

Then there is the z-associated expenditure (increasing or declining), the number of associated personnel, their costs, outsourced or insourced. From a top-down viewpoint one should also determine how much enterprise data resides on the z/OS, how much data is off z/OS (e.g. Linux under VM), how many SQL inquiries are being tested from all sources under z/OS and how many Stored Procedures and Triggers are being executed. Much also depends on the current z/OS business logic and how much more the date dependant logic is going to be processed under SOA with Java, WebSphere Application Server, Java and UNIX System Services (USS). Date and time errors can extrapolate with SOA as all the business logic.

Some organisations have all of these figures in currency terms and others have sections of this information that can be deemed from a currency viewpoint. They have to be combined to obtain an actual currency value, per MIP, per MSU or some other means.

Although TCO is meaningful it is never straightforward requiring an open mind to produce. The economic environment of today is where organisations are looking very close at all costs and processes (no longer rely on the way they have always performed). SOA structures have assisted in reducing TCO by speeding up IT&C developments where the adoptions of different software development processes are still trying to match evolving business workloads. TCO is not easy, for example, costs may be increasing 5% per annum with efficiency increasing 10% so in theory TCO is reduced by 2.5%; very nice numbers for discussion, depending on what side of the fence you sit on. Today, are these percentages based on an IT budget only or the associated businesses? Are the Businesses increasing sales with larger margins of profit because of new technology?

The absolute TCO for Softdate is to minimise or remove all date dependent processing logic errors in z/OS-based apps. These Apps may be for example CICS, DB2, IMS or batch and/or SOA-enabled with z/OS Java logic, WebSphere Application Servers and UNIX System Services apps. In certain SOA environments SoftDate's virtual clock may be synchronizing any number of connected systems, e.g. iSeries, Linux, Unix, AIX, Windows to that same virtual clock for end-to-end testing of application behaviour; critical dates and times across a heterogeneous mix of systems.

The main TCO issue for any z/OS-based system is the actual costs of date and time errors which includes not only bug-fixing, the re-coding of inept code, re-release/upgrade of apps but also from an IT viewpoint the costs of emergencies and the crisis management with exponential growth of date and time errors. The actual costs to internal businesses, to external business partners and to clients are immeasurable by IT but for brand reputation, the customer-facing personnel and angry clients, it can be enormous.

Then, depending on what industry you are in or associated with, the strict compliance with tough standards that include SOX (Sarbanes-Oxley Act), HIPAA (Health Insurance Portability and Accountability Act), GLBA (Gramm-Leach-Bliley Act), DSS (Payment Card Industry Data Security Standard), PCI Security Standards Council and others around the globe. You can no longer rely upon a single page, double paragraph media

statement to save the day because social networking protests produce reams of statements. Date and Time perfection is a must as errors can be removed by DTS.

In the three TCO examples below we take into account a Greenfield organization that never date and time test, those who do date and time test manually (without a DTS product) and the comparison of Softdate against any of the DTS competitors.

How to TCO Softdate in a Greenfield site?

To TCO Softdate in a Greenfield site one must weigh up the balance on a 'have or have not basis'. The table below provides the more simple Steps where a date and time error has occurred where only 100 users are using the app online, which has arbitrary numbers that require organization-specific values. For example, one bug could involve 50 users where in another case it could be 5,000 users.

Critical Application Date and/or Time Error Table					
Steps		User Delays Hours	100 User Costs	IT Hours	IT Costs
1	Users identify a glitch	8	\$16,000	0	0
2	Chasing down date or time bug/glitch found by users	15	\$30,000	15	\$300
3	Identify bug in code	8	\$16,000	8	\$160
4	Fix bug	6	\$12,000	6	\$120
5	Test fix by IT	6	\$12,000	6	\$120
6	Implement fix for users	2	\$4,000	2	\$40
7	Upgrade fix at a source level into app	0	\$0	1	\$20
8	Test upgraded code by IT	0	\$0	2	\$40
9	Test upgraded code by 4 Business users	4	\$80	0	\$0
10	Misc collaboration, team work, documentation by management	2	\$40	2	\$40
TOTAL COSTS OF 1 DATE AND/OR TIME ERROR:			\$90,000		\$840
In this example only, the major cost of \$90,000 is the business folk with IT at \$840, less than 1% of total.					

In the above table the costs of one error are born 99% by the businesses and only 1% by IT. This example has only 10 basic steps but in our complex day-to-day world it is often far more complicated as often the date and time bug will not be discovered before true damage is done. For example, if batch runs are completed before the identity of the bug, exponential logic problems occur. Or, the batch run itself could be in peril. In electric commerce the business costs will extrapolate with logic errors going external, or the non-compliance of a Standard or Act. There are so many possible errors with date-dependent processing but understandable types are for example, missing an insurance policy renewal or the wrong pension rollover.

How to TCO Softdate when you date and time test manually?

Some organisations have no DTS testing product and change date and time by re-IPLing one or more LPARs and resetting the system clock. Some just do this month after month in set processes. It is tedious, difficult, disruptive, time consuming and expensive. Resetting the system clock can cause a number of problems such as software expiry problems (i.e. products can believe their licenses have expired).

It is not easy to truly test different times and days manually as the system clock has to be modified every time a different date or time is required where with Softdate it is a simple user change at the user level so multiple testers can be very granular, performing extensive tests at the app level.

Softdate provides the **maximum granularity in activation for any job step** via JCL or a simple but very powerful ISPF rule definition facility, which allows specification of eligible workloads via any combination of:

- job name
- step name
- procedure step name
- program name
- user id
- job class
- subsystem

It is virtually impossible do perform high quality QA of dates and times with manual changing of the system clock unless the date and time logic is very well known, minimal and not open to change.

Originally these organisations did it manually due to the high cost or unsuitability of simulation software to their own site but now Softdate works across all modern workloads.

The TCO for this manual intervention of resetting the system clock will vary depending on the overall cost of the specific LPARs being used, the amount of time spent testing in comparison to a DTS product and the complications of fixing date and time error bugs with such steps as briefly described in the table above. It is definitely less expensive to use Softdate, especially in a much more complex world.

How to TCO Softdate against an incumbent competition?

One can obtain clear TCO indications from a side-by-side viewpoint. That is, the end results are to drive even more efficiency in TCO with customers proceeding at the individual competitive product level. It isn't just about the software price; it is also about technical innovation, better automation of the processes, increased staff efficiency, reduced MIPS, better support and more.

The competitive products were all developed and written in the old century where Softdate was provisioned in the 21st Century. In IT organisations where an incumbent DTS product is used the Softdate TCO is therefore initially compared against the TCO of the old century product. Then, the unique Softdate DTS in the SOA world of z/OS-based Java, WebSphere Application Server, UNIX System Services (USS), CICS MRO and other unique facilities.

•

The table below provides estimates that can be altered to be site specific.

Softdate Function	Competition	Softdate TCO Savings
<p>“minimum impact” design; essentially no impact on jobs that are not using Softdate. This means minimal usage of processing power.</p>	<p>All other products intercept every system date or time request that occurs in the LPAR on which they are installed.</p> <p>Since typically only a tiny percentage of the total workload on the LPAR actually requires date simulation services, this can be a significant hidden overhead and CPU cost.</p>	<p>Date-dependent processing logic can be light through heavy.</p> <p>CPU savings could possibly range from well less than 1% to 2%.</p> <p>Find TCO of every LPAR being used to obtain a TCO value, e.g. 1% of \$100,000 is \$1,000.</p>

<p>No permanent modifications to any IBM system software. No need for separate libraries with patched copies of key system and subsystem routines. This feature alone sets Softdate apart from other z/OS and older MVS products on the market, and completely avoids several significant potential maintenance and performance headaches.</p>	<p>Make permanent modifications to IBM system software. Must have separate libraries with patched copies of key system and subsystem routines.</p> <p>Several significant potential maintenance and performance headaches.</p> <p>Difficult in system software upgrades and compliance with 3rd party software.</p>	<p>Greatly reduces the Maintenance costs and much easier.</p> <p>Faster upgrades and much less work.</p> <p>80% reduction in Maintenance costs, e.g. 20% of \$10,000 is \$2,000, an \$8,000 saving.</p>
<p>Maximum granularity in activation for any job step via JCL or a simple but very powerful ISPF rule definition facility which allows specification of eligible workloads via any combination of: job name, user id, step name, job class, procedure step name, subsystem, program name</p>	<p>Limited granularity depending on product.</p>	<p>With maximum granularity more precise testing is provided.</p> <p>Speeds up testing with better results.</p> <p>Reduces testing time by at least 1%.</p>
<p>The most powerful, flexible and intuitive ISPF interface for specifying workloads to use Softdate.</p>	<p>Less flexible user interfaces.</p>	<p>Increases quality and reduces testing time by at least 1%.</p>
<p>Extensive Job rule enhancements</p>	<p>Less powerful Job Rules</p>	<p>Increases quality and reduces testing time by at least 1% even up to 5% depending on environments.</p>
<p>The most powerful set of CICS features available: users can set a clock value for all work run under their user id, a clock value can be set for groups of users, transactions and/or terminals, rules can be updated, activated or inactivated via batch (for increased control), rules can be automatically activated when CICS starts, comprehensive optional security controls</p>	<p>Less powerful CICS features.</p>	<p>Increases quality and reduces testing time by at least 1%.</p>
<p>Unique CICS MRO (multiregion operation) support means setting a simulated date or time under CICS is exactly as simple under MRO as it is in a single region. CICS regions are grouped by the name of the rule set that activated Softdate for them,</p>	<p>No support for CICS MRO.</p>	<p>TCO benefits are strictly with organisations who use MRO.</p> <p>As no competitors have MRO the TCO can be calculated on a Need Basis, where the benefits of testing apps under MRO are made.</p>

<p>and users will see the same virtual date no matter in which region in the group their transaction runs. This support is a significant enabler for accurate application testing and certification in an SOA-driven world where CICS is being opened up to other platforms, and MRO configurations are the norm.</p>		
<p>Online users in the same CICS or IMS region can independently set dates and times without impacting one other.</p>	<p>No support known for this functionality with any competitor.</p>	<p>Depending on the working environment it can Increase quality and reduces testing time by at least 1%.</p>
<p>Advanced IMS/TM support, including: user setting of test dates and times from within IMS (no need to exit to ISPF), each user can independently set their own test dates and times, enhanced support for Fast Path message regions, automatic IOPCB date support</p>	<p>Less powerful IMS features, certain competitors do not fully support IMS.</p>	<p>Increases quality and reduces testing time by at least 1%.</p>
<p>Configurable IMS region sharing type means that the scope of Softdate user rule sharing across IMS regions can be configured as: within regions sharing the same IMS subsystem, or within regions grouped by the same job rule set (the default), or within all IMS regions on the LPAR.</p>	<p>Less powerful IMS features, certain competitors do not fully support IMS.</p>	<p>Increases quality and reduces testing time by at least 1%.</p>
<p>DB2 Fully Nested Stored Procedures, Triggered SQL and SQL support means Softdate will correctly shift current date and time requests.</p>	<p>Less powerful and limited support.</p>	<p>The TCO will increase by the complexity of the organisation and could be 5% or more.</p>
<p>Parallel Sysplex-ready support allows automatic synchronisation of shared rules and clocks across a Sysplex. For example if a CICS user sets a virtual date in one region of a CICS MRO that date can be propagated automatically to all other participating regions, even if spread across multiple LPARs.</p>	<p>Parallel Sysplex not fully support or no support.</p>	<p>Depending on the working environment it can Increase quality and reduces testing time by at least 1%.</p>

<p>64-bit support to applications specifically exploiting 64-bit storage above the 2-gig bar and those that exploit it indirectly via facilities in the latest levels of systems software (such as Language Environment, DB2, Java and WebSphere Application Server).</p>	<p>No support for 64-bit</p>	<p>This is very important for organisations utilising 64-bit as without testing the 64-bit apps, serious problems could occur.</p> <p>TCO is difficult to calculate as site specific and there is no current testing.</p>
<p>Unique, Optional support for Java apps running on the z/OS mainframe. This includes Java applications running under batch, CICS, IMS and WebSphere Application Server for z/OS (WAS for z/OS). This option provides simulated date and time testing from the z/OS side for web-enabled applications and enterprise-class transaction applications.</p>	<p>No support.</p>	<p>The TCO is very high as the only way to DTS Java logic and WebSphere Application Servers and their integration with CICS, batch, DB2 and IMS is with Softdate.</p> <p>Today, DTS is very expensive and time consuming by manual methods.</p> <p>Any site with Java and its integration with z/OS apps will see at least a 25% TCO saving</p>
<p>TCO Savings of Softdate to any competitor for user efficiencies, reduction of missed errors and reduced Maintenance on all features supported by the competition is between 10% and 30%, site specific.</p> <p>With sites that use CICS MRO, Parallel Sysplex, Java, WAS for z, UNIX Systems Services and other unique features TCO would see a major reduction in TCO.</p>		

For those sites who have different time zone LPARS to match different time zone businesses, the costs can be huge. Softdate *GTZV (Global Time Zone Virtualisation)* can reduce TCO significantly. Softdate makes cloud computing an easier experience.

To deploy SoftDate please call your local representative or go here and send an email:

To deploy Softdate please call your local representative or go here and send an email:

[mailto:<Softdate@ddvtechnologies.com.au>](mailto:Softdate@ddvtechnologies.com.au)