Integrity Journal on the Web

The first **Integrity Journal** was mailed in September of 2004, almost one year after **Integrity Services** was started. There have been three issues a year (January, May, September) since that time. Several years ago the Journal was placed on our web page dating back to 1999. The current issue and previous issues are available under the Technical Info tab. The web page is www.integrity-services-inc.com. Because this is a very convenient media, it is our intention to continue offering the complimentary Journal only on the web page. This issue as well as the next issue (January 2008) will be mailed in written form as all the others have been.

Beginning in May 2008 the **Integrity Journal** will only be available on the web page. Feedback is always welcome.   For anyone desiring an email to tell you there is a new Journal on the web page, we will be happy to do that.   Send us your email address with the request.   You will then get all **Integrity Services'** correspondence of updates, changes, and announcements.   As always, our list of email addresses is not sold or given to others.

> *Courage is very important. Like a muscle, it is strengthened by use.*
>
> *Ruth Gordon*

Pricing of Doctor Software

In January, 2008 the pricing of Doctor software will be increased for the first time in several years. All maintenance prices have been placed at the 20% line for the first time. Software ordered on January 1, 2008 or later will reflect the new prices.   This does not affect maintenance fees for current users. These fees are fixed for your protection by contract. Order now and save money (10 – 15%).

New Password Management Controls

A couple of years ago Unisys implemented additions to the character set allowed in passwords.   The **Integrity Password Management**, which may be purchased separately, but is included with the **Security Doctor** and **Super Doctor Software**, now takes advantage of the character extensions.   This character set now includes upper case letters, lower case letters, numbers, and special characters.   These may be included in a password up to 15 characters long.

These complex passwords meet industry standards and are available with the MCP. The Doctor software takes advantage of these extensions. This password aging is a must in today's imposing on-line environment.

Tracking Source File Changes

There are several methods that may be used in tracking changes to source files. It is certainly a good idea to track changes to Cobol, WFL, Algol, Fortran, etc. The purposes to keeping records are to be able to restore to a point in time, know what changed when unusual things happen on the computer, and to audit changes for internal use.

One age-old method is to keep a list of changes and dates as they occur. This method should be adopted even with other methods since it can be used by operations personnel when dealing with unusual situations in the computer environment. It also lets others see what changes have been made so that they can integrate their changes within the framework of all software changes.

Another method to track source changes is to use the Markid function in Cande. This will work with all source files except WFL. What an oversight! The Markid function allows a ten character literal to be place in each line of source code that is added, changed, or resequenced. There is benefit in using the Markid even if other measures are used to track changes. It provides documentation in the source code. I prefer to use a Markid with the date in it (08/24/2007), others use patch levels. This does not replace internal documentation, but it does document when a change was made.

Unisys provides software to allow another method of tracking changes. If each change is coded as a patch file, then System/Patch may be used to gather all patches, merge them together, and then you may fire off the appropriate compiler. You still must document each patch file so that you know when and what it is doing. System/Patch allows you to do this in header records that begin with a $. A $# is the first line in the code file and each additional comment is preceded by a $:.

Again, System/Patch may be used with Markid for further amplification of changes. I use both of these methods as well as placing documentation comments in a patch list at the beginning of each source file. This way I know what changes were made and on what date for a source file or a patch file.

For auditing source file changes and to control changes, special software is needed. Although there is at least one other product on the market, I have developed software to address this issue, and I will use it, **SourceTracker**, as an example in demonstrating how to use this method of tracking source file changes. **SourceTracker** was developed as a cost effective method of tracking and controlling source file changes.

First only one person may be working on a particular source file at any given time. Others may view it but only one may make changes. This is done by having a user, who is in the qualified list, check out a particular source file. Until this user checks this source file back

in, no one else may check it out. Even if one user were viewing the source, he or she must still check it out before making changes. This ensures integrity in each source file. Only one user may make changes, and all other attempts will be disallowed.

If someone decides to circumvent the system and instead of checking out a source file the user copies the source under his or her own usercode, there is no way to stop this without watchdog software on the system. With **SourceTracker** this is done with the **System Doctor** software. For those who do not have the **System Doctor**, a restricted **System Doctor** is provided at a modest cost for this function. Alarms will be generated when this happens. This software has satisfied very strict auditors. **SourceTracker** maintains each previous source file with a date in the title. Reinstating these files is very easy.

While there are many ways to track software changes, each one must decide upon the local needs to establish the methods employed. Sooner or later a need arises to go back to a previous version of software or, at least, to know what changes were made at a certain point in time. This discussion only addresses maintaining source files. Some may want to control object files in a production environment. This is another discussion. Maintaining source files is a higher priority than object files since object files can always be reproduced from source files.

> *To me, life is like the back nine in golf. Sometimes you play better on the back nine. You may not be stronger, but hopefully you're wiser. And if you keep most of your marbles intact, you can add a note of wisdom to the coming generation.*
>
> *Clint Eastwood*

MCP Uniqueness

There are so many features about the MCP that no other operating system has. I doubt that there is a user out there that uses everything. The main reason that the MCP has developed so many features over the years is that Unisys has kept the same operating system since its inception in 1969 on the B6500. In the early 1980s, with help of a consultant (Mr. Worth who gave us Pascal), at that time, Burroughs moved the MCP source to a new compiler, NEWP. All functionality of the MCP was retained.

Burroughs also modularized the MCP so that all of the written code developed over the years could be retained for each generation of computers. That has existed to this day. The benefit of modularization,

in addition to ease of functional use, is that every new computer hardware interface can be placed as a module in the MCP without disturbing anything that has been developed over the years. No other operating system has had this advantage.

Another outstanding benefit to the MCP is that, because of the modular development retaining all functionality, there are no conversions needed when moving to the next level or generation of computers. Other computer manufacturers develop a new operating system with the introduction of new hardware. This means that a user must go through a conversion when moving to a new computer. In the mainframe arena this is a big deal. Although Unisys has only one competitor, Unisys alone does not put their customers through a conversion when introducing new hardware. This concept may be downplayed by those promoting another computer, but nevertheless, it remains a fact.

Others may tell you that it is simple to move to their computers, but someone has to do a conversion. It may be done for free initially, but after they have you, it will cost you. Remember, the functions of the MCP have been debugged over the years and we take for granted that they will work on new systems. This is not true on other computers. Functions that used to work make not on new computers. Unisys is alone in this superiority of design and function. Long live the MCP! There is nothing else like it.

MARC – FN command

A little known Marc command will show internal file names of programs. It is the FN command. The syntax is FN<program file name>.

WFL Wait Statement

I have noticed many users using the WFL wait statement to pause for a given number of seconds (time). This statement works fine, but a number of them would prefer to wait on a condition. The WFL statement, Wait(FILE TEST/DATA IS RESIDENT), will stop execution until the file, TEST/DATA exists. Then it will proceed to the next statement. This avoids guessing a time interval and wasting time.

I am ready to meet my Maker. Whether my Maker is ready for the ordeal of meeting me is another matter.

Sir Winston Churchill

New Marc Directive

When trying to tighten security many stations must be made non-commandcapable. This reduces the available commands to users of the stations. They still have windowing capability, but very few commands. Even 'Hello' is taken away so that if the station is automatically logged on then the user cannot use 'Hello' to change usercodes. This can be very helpful, but sometimes there are one or more commands that you would like to give them. Hence the new Marc Directive for Doctor users. Others may purchase it, but it is without cost to Doctor users.

The new Marc Directive allows the user to specify a station list and a command list. This is preferred over the old way of making commands available to all non-commandcapable users. The station and command lists are simple Cande created files and may be maintained as all security-relevent files should be. The command list or station list may be altered at any time and put into use by using a Replace Directive function.

Other Marc Directives available give the ability to add or delete commands from non-system users. These include:
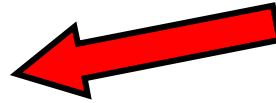
QUIT COMS
GO
PA
CC
RUN
START
NEWS
INFO
MORE
ODT

Marc Directives may be contained in one or more libraries and turned on and off at anytime with a Directive function on the DIRECT Window. To check for your use of Marc Directives issue a 'Go Direct' command on the action line of Marc. You will observe several options including the option to list current directives. In addition to adding and deleting Directives you may also disable and enable any Directive to make commands temporarily unavailable.

Marc Directives remain in use until disabled or deleted. They have the same function after a halt/load, power off and on, or an MCP upgrade. Once set you do not have to concern yourself about reactivating them.