

Inside this Issue

News Release - eventACTION.....	1
New Product - ussACTION	1
And another thing	1
Compare Utility	2
What's New: Compare.....	2
Ignore Defined Records	3
vertiCOMP	4
RTM Timing	4

Contact Information

Action Software ACTION news
June, 2004, Issue 09 is published by:
Action Software International
20 Valleywood Drive, Suite 107
Markham, Ontario L3R 6G1
Canada
Phone: 905-470-7113
Fax: 905-470-6507
E-Mail: sales@actionsoftware.com
Web Site: www.actionsoftware.com

Address correspondence to:
newsletter@actionsoftware.com

Copyright © 2004 Mazda Computer Corporation
Action Software International is a division of
Mazda Computer Corporation.

eventACTION, ussACTION and Changeplex are
all trademarks of Mazda Computer Corporation.

All other trademarks and registered trademarks
are the property of their respective owners.

May not be reproduced without permission. All
rights reserved.

Information subject to change without notice.

For reprints or back issues, contact Action
Software International.

And another thing

... As previously announced in our
October 2003 newsletter, support for
eventACTION releases up to and
including 6.05 will be withdrawn
effective June 30, 2004.

News Release - eventACTION

eventACTION is a new name for a proven quality z/OS systems management software product.

Change Action was originally designed as a product that manages, tracks, and controls all system changes in an OS/390 or z/OS computing environment. Over the years Change Action has evolved into more than just a change management tool. The name change from Change Action to **eventACTION** was made to reflect this evolution.

eventACTION is a comprehensive event tracking product that can help you proactively manage your z/OS environment without impacting the way that you already work.

Some of the events that can be tracked and controlled are:

- Program product execution
- Changes to data sets and members
- References to data sets and members
- System command usage
- SVC table updates

Some functions you can do from these events are:

- Keep statistics on changes and references to system data sets and members
- Take backups of data sets and members when changes occur
- Distribute changed data sets or members to other sites
- Control program products from executing on non-licensed systems
- Prevent changes to selected data without an approved change request
- Log operator commands to remove need to scan thru data in SYSLOG

eventACTION also provides a comprehensive reporting facility, a task scheduler and a unique side-by-side compare utility to allow you to identify changed data at a glance. *

New Product - ussACTION

Action Software International is pleased to announce the general availability of **ussACTION**, which helps installations manage their z/OS Unix System Services environments much as **eventACTION** assists with z/OS MVS systems.

ussACTION is comprised of the following major components:

- ussACTION Change Tracker
- ussACTION Change Manager
- ussACTION Reference Tracker

ussACTION also includes a full suite of supporting features such as reporting, external logging, excludes, and file and backup comparison. *

And another thing...continued

To upgrade to the current release of **eventACTION**, and to take advantage of all of the new features and enhancements that are included with release 6.06 to 7.00, contact your Action Software representative to order your tape.

... The **eventACTION** DFSMSHsm interface provides the ability to track and control changes made using the DFSMSHsm **HDELETE** and **HRECOVER** commands. When **eventACTION** tracks a change made using **HDELETE** or **HRECOVER**, the change is attributed to the user who issued the **HDELETE** or **HRECOVER** command rather than to DFSMSHsm. The Program will indicate that the action was done by DFSMSHsm.

The interface consists of a DFSMSHsm user exit program.

...The **eventACTION** DFSMSdss interface can track changes that are made using DFSMSdss, at both the volume and data set level. This interface requires `parm=dfdss` at **eventACTION** startup. For data sets currently defined to **eventACTION** for tracking, changes made by DFSMSdss will be tracked as soon as the interface has been activated.

To track volume level changes, pseudo data set names must be defined to 'Dataset Options'. The quick way to add these names is to add `.VOLUME.*`. Refer to the **eventACTION** Administration Guide for details.

... With **eventACTION** 7.00, VSAM change control is generally available. To activate, `parm=vsamctl` must be specified at **eventACTION** startup. *

What's New:Compare

Ignore IDR Date

When comparing load modules between two libraries, it is often the case that the only difference is the compile and link edit dates; you may not want to consider this a difference since execution of the module is not affected by this difference.

Compare Utility

The dataset Compare utility provides the capability to compare two files, whether live disk datasets or **eventACTION** backup copies. These files may reside either locally or at a remote location connected via **eventACTION**. The utility supports all file types that **eventACTION** supports for back up of changes.

Compare is comprised of two processes - the full data compare and the quick compare. The full compare is the default process which performs a record-level comparison of two files in order to determine, at a glance, the changes that have occurred in the data. The resulting output of the utility is a side-by-side display of the two files, using a vertically split screen, showing changed records via highlighting and separator markers. Alternatively, a quick compare may be performed to generate checksum values which are then compared to indicate simply whether or not the files are equal. This option uses far less CPU resources.

The purpose of the Full Compare is to provide a means of determining data changes via a process of file comparison. But the key objective is to present the results of such a process in such a way that you can quickly and easily identify the changes that have occurred within the data contents. Thus, flexibility in what is able to be compared is a primary goal of the utility. The greatest benefit is achieved though by being able to view changes in the context of the entire file or in relation to only what data is significant in a particular instance, therefore, this flexibility is extended to include the presentation of the comparison results. Extensive options are available to allow you to tailor specifically what data you wish to compare and how you want to view the results. In this way you can avoid wading through an unnecessary, and often confusing, amount of output to find what you need to know.

The purpose of the Quikcompare process is to minimize the utilization of CPU and network resources to determine only if two data sets or a set of their members are different. This process is especially valuable when comparing information between two sites.

The utility compares real sequential or VSAM data sets. Partitioned data sets (PDS) are compared at a member level; other libraries in the form of Panvalet or Librarian are also compared at a member level. For such libraries, the compare may be limited to only member lists in order to determine if all members exist in both libraries. From such results, data contents can then be compared for a selected pair of members. For any of these types of data sets, **eventACTION** backup copies can also be compared. Any type of data set supported by **eventACTION** is likewise supported in the Compare Utility. Thus, different backup copies of a member of a PDS or library or versions of data sets can be compared to determine how the file has changed over time. Data Sets of like organizations are typically compared but those of different organizations may also be compared. Backup copies may also be compared to real data sets or members. A variety of record formats are supported and need not necessarily be the same for the two files being compared, except for load modules. Record sizes may also be different for the two files.

Data sets and **eventACTION** backup copies can also be compared across multiple sites, that is, the two files being compared may reside in two different locations. This can only be done if **eventACTION** and its LULU component are installed and running at the local and remote sites. It is LULU which performs the networking functions. Each site is identified by a Changeplex name where a Changeplex is defined as a group of hosts that physically share a common database.

Some of the types of data that you may wish to compare are: source files, object code, JCL files, Clist members, REXX members, load modules and so on.

Ignore IDR Date...continued

There is now an option on the main Compare Utility panel called "Ignore IDR Date Info" which, when set to YES, will ignore all IDR records in load modules (these records contain the date information related to compile and link) and also ignore the eye-catcher area at the beginning of each CSECT; the eye-catcher area must be located immediately following the first instruction which must be a branch around the eye-catcher area. On the Data Output display you will see the eye-catcher area denoted with the Ignored indicators (round brackets) in the separator fields.

When using the quick compare with this new option, the CRC's generated will exclude the IDR records and the eye-catcher area. Also now included in the CRC values is linkage editor information obtained from the directory entry for a load module so that if link attributes are changed then these will generate a different CRC value because they can affect the execution of the load module.

PDSE Support

PDSE load modules are now handled in the compare utility. The output hexdump format is slightly different from non-PDSE load modules because of the extra information that may exist within a PDSE load module. As such, non-PDSE load modules cannot be compared with PDSE load modules.

Commands to Alter Options

Commands to alter comparison options have been added to the line-by-line Data Output results display so that you do not have to return to the main compare panel to change these options. There is now a LDREF command to turn this option off and on, there is now a COLS command to alter the columns being compared, and there is an IGNORE command to alter the options of ignoring line numbers, IDR dates for load modules, and ignoring certain records within the files using a table of record definitions. Please refer to the help panels.

Compare Utility continued

The following are a few examples of using the utility to compare data sets.

- A sequential data set with a member of a PDS.
- A VSAM data set with a sequential data set (i.e. a true VSAM file with a REPRO or EXPORT version of a VSAM file).
- A Panvalet member with a member of a PDS (i.e. a production source member with a development member).
- A member list, and possibly all their contents, of a system library on one volume of an operating image with the same library on another volume for another system image (i.e. SYS1.PARMLIB(*) on RES001 with SYS1.PARMLIB(*) on RES002).
- A real member, or the most recent backup copy of that member, with the previous backup copy (i.e. backup copy 1 with backup copy 2).
- A Clist / REXX member with a variable format (RECFM=V) with a fixed Clist / REXX member (RECFM=F).
- A load module with itself to obtain a reference map, determine what zaps have been applied, or to see the linkedit date.

There is no limit to the size of the data records that may be compared hence all output is made fully scrollable while viewing sizes are also modifiable. There is no practical limit to the size of the files to be compared. However, the utility makes extensive use of the user's region above the 16Mb line. If the installation limits such usage then the compare utility will, in turn, be limited. Both data sets being compared must be on volumes that are physically accessible to the system on which the utility is invoked except when using a remote CPX. *

Ignore Defined Records

Files that are to be compared may contain records which are not significant in the compare process and/or may cause the side-by-side view of the results to be skewed because there are so many of them. Such records might be blank lines or comment cards. These records may be ignored by defining them in a table and using that table during the compare process. You may define different tables for unique types of data.

Many times, it is desired to ignore data records with specific data occurring in variable positions but either before or after specific locations. Therefore defining the actual records that are to be ignored during the compare process involves defining a set of criteria which make up strings of data that must be found, possibly in a certain area, of each data record. Once a match has been found between the definition and a portion of a data record, then that record is marked to be ignored.

eventACTION allows you to define the records that you wish to ignore and group them together in a table. This table is stored in your ISPF table dataset and is available to you any time you wish to compare similar types of files. You may create as many tables as you need according to the variety of types of data with which you work.

For example, comment cards and blank lines are very often unnecessary in comparing files, but comments must follow various rules depending on the type of data. PL1, COBOL, Assembler and other programming languages each have their specific format for comments. Parameter files have their own requirements also. You may then define different tables to be used when processing different types of data. You may have tables such as PLICODE, ASMCODE, and PARMFILE to deal with your own requirements.

Use of Pop-ups

Pop-up windows are being used to display warning and results messages to give a more complete view of the compare processing. When comparing a set of members using a member mask, a results summary will be displayed. When using Quick compare, a results summary window will also be displayed. *

vertiCOMP

Have you heard about vertiCOMP? **vertiCOMP** is an independent copy of **eventACTION**'s Compare utility. It is initiated from a clist and can run completely independent.

Your application development staff may have a requirement for a comprehensive compare utility, but do not have a requirement for the other features of **eventACTION**. With **vertiCOMP** you can give them access to Compare without logging into **eventACTION**.

For further information contact us at vertiComp@actionsoftware.com. *

RTM Timing

RTM definitions & deletions in a multi cpu environment can have delays with some of the sharing cpu's receiving the host records. If an RTM entry is deleted shortly after creation, timing problems can occur. If you must delete a new RTM entry shortly after creation you should wait 2-3 minutes per cpu before deletion. *

Survey Enclosed

With this newsletter is a survey to get client feedback for our products. We aim to provide our customers with highly functional and usable software along with exceptional customer service. We have developed this survey to this end. To fill out the survey online email us at

Survey@actionsoftware.com. *

Ignore Defined Records continued

Upon using an ignore table, the comparison results will align in a more productive manner where the changes can be more precisely highlighted.

During compare processing, each data record is checked against the records in the Ignore Table. If the data record meets the requirements as defined by one of the generic definitions in the table, then that data record is ignored and marked as such. In the output display, ignored records are typically coloured green and the associated separator column contains a round bracket. Each generic ignore record definition is comprised of a primary character string and two optional strings, each with optional locations and repetition factors.

Here is an example of a definition for PLI blank comment lines:

```
.... Ignore => ' '      Blank PL1 Comments in Col => >3      Occurs => >30
Preceded By => /*     At least 30 blanks      in Col =>      Occurs =>
Followed By => */     between /* */D      in Col =>      Occurs =>
```

By using just the **Ignore** field for the character string definition, or a combination of the **Ignore**, **Preceded By** and **Followed By** fields, you are able to define what characters must be contained in a data record and in what order they must occur. Quotes can be used to imbed blanks within a string. The **Ignore** field is the only mandatory field and is checked first with its location and repeat factor to determine if the data record matches it. If this field matches then the **Preceded By** and **Followed By** requirements are checked.

By using the **In Col** field, you can be more precise about where the characters must be found within the data record. Without this field, the character string may be located anywhere in the data record. You may specify this number as a range by using <tt for any column up to tt or >ff for any column greater than ff or ff:tt for any column from ff to tt inclusive.

By using the **Occurs** field, you can indicate that the character string is a repeated pattern and by how much it must be repeated to generate a match with a data record. This number is a multiplication factor of the length of its corresponding string. As with the **In Col** field, this number too may be a range by using <t for any factor less than t (cannot be more) or >ff for any factor greater than ff (must be at least this many) or ff:tt for any factor from ff to tt.

Ignore Defined Record Tables are ISPF tables in that they are created and stored according to the standard TSO/ISPF table format. A sample table (MZSAMPLE) with various unrelated examples of record definitions is supplied with **eventACTION** to give you an idea of how these various fields fit together. The primary command **Ignore** will present you with a list of all the tables defined within your TSO/ISPF user environment. As with any ISPF table, if you create or update an Ignore Table, then the new copy is saved in the table library allocated for update in the user's session. Such tables can then be copied to the read-only table data set if you want to make these tables available to others so that each user doesn't have to define their own. But they still may customize one of their own if they want to.

To invoke the use of an **Ignore Defined Records** table in the online environment simply set the **Ignore Defined Records** option to **YES**. If you know which table you want to use, specify the name in the **Using Table** field. If you specify **YES** in the **Ignore Defined Records** option but leave the table name blank, you will be prompted to select a name from a list of the tables available to your session. In the batch environment, each record entry of a table must be individually defined by using the control card format. This is because TSO/ISPF table data sets are not usable in batch mode. By indicating that you want to use an **Ignore Defined Records** table on the online panel and issuing the command **B** for **Batch** instead of invoking the compare, you can then use the command **EDIT** on the subsequent panel to generate the control cards for the specified table. *