the Availability Digest

www.availabilitydigest.com

FileSync and CSR Synchronize NonStop Systems
Part 1 - FileSync

October 2011

A common technique for guaranteeing system availability is to configure a backup system that can take over processing should a production system fail. A major hurdle to achieving high availability with active/backup architectures is *configuration drift*. If the software versions of programs, scripts, and configuration files resident on the backup system are not up-to-date, version conflicts may prevent the backup system from operating properly. Failing over to the backup system following a production system failure may be unsuccessful, resulting in a *failover fault*. Equally important, testing failover is made much more complex if version errors must be tracked down and corrected in order to successfully pass a test.

To ensure that a failover will be successful, it is important that all versions of software running on the backup system match that of the properly operating production system. Tools are available to compare the production system software modules to those on the backup system to detect version errors. If such errors are found, operations staff must take steps to correct them.

A more advanced solution is to have a facility that not only will detect version errors on the backup system but that also will automatically correct such errors. Such a facility is FileSync from TANDsoft, Inc. Coupled with TANDsoft's Command Stream Replicator, FileSync relieves the operations staff from having to continually monitor and correct backup software versions.

Synchronizing the Backup System

Three classes of objects are involved in system synchronization to ensure proper failover:

- *Audited Databases*: Several products are available for replicating changes in real time to a NonStop audited database, whether it be Enscribe or SQL. They include RDF from HP, Shadowbase from Gravic, Inc., DRNet from Network Technologies, Replicate from Attunity, and GoldenGate from Oracle. These products read changes from the TMF source audit trail and replicate them to the target database.

- *Unaudited Files*: The primary function of FileSync is to synchronize unaudited files. An unaudited file may contain several types of data, such as program source code and executables, scripts, configuration files, or application data. FileSync ensures synchronization of unaudited files by replicating the entire file or by replicating only the changes to the file (FileSync Audit).

- *Configuration Changes*: Various NonStop utilities are provided to change the configuration of the processing environment, such as FUP and SQLCI. These configuration changes must be made to the backup system as well. This is the job of TANDsoft's Command Stream Replicator.

In Part 1 of this series, we describe FileSync and FileSync Audit. In Part 2, we introduce CSR.

FileSync

FileSync synchronizes application environments and unaudited files across one or more NonStop servers. The target servers may be disaster-recovery systems or other production systems (for instance, servers in an active/active network). Bidirectional replication in an active/active environment is possible by configuring a FileSync subsystem for each direction.

Multiple copies of FileSync can run on the same server to support, for instance, different business units.

FileSync can replicate files across either Expand or TCP/IP links, though the use of Expand is the more efficient option. However, TCP/IP is required if the node names of the source and target servers are the same.

File Types

FileSync can replicate any Guardian or OSS file on a NonStop system. These files include:

SQL tables and partitions*
Enscribe files and partitions*
OSS files and directories
Audited and unaudited files or tables
Backup and Restore files
Edit files
Configuration files
TACL scripts
Batch files
Program source and object files
PAX and PAK files

*partitioned tables and files are replicated in their entirety.

Invoking FileSync

FileSync is invoked via a TACL (Tandem Advanced Control Language) script. The script includes many parameters and options that are provided to control FileSync. They include:

- *Job ID*: Each FileSync job is given a descriptive ID.
- *From file lists*: Files that are to be replicated are included in one or more From file lists. These are standard NonStop qualified lists. File sets can be specified with wild cards and further qualified with a WHERE clause that selects files based on multiple file attributes.
- *To file lists*: The target files to which the source files are to be replicated are given in one or more qualified To file lists.
- *Interval*: The times at which FileSync is to be invoked.
- *Rename*: A target file that is open for read-only access or execution can be synchronized by renaming the currently open file.
- *Open*: A source file that is open for write access can be synchronized so long as it is not opened exclusively. This option should be used with great caution as it can cause file corruption.
- *Purge*: Delete target files or subvolumes that are not present in the source system file set.
- *Priority*: The priority at which FileSync processes are to be run.
- *Window*: The amount of time allocated to a FileSync replication run. If FileSync exceeds this, it terminates; and the files that were not replicated will be replicated on the next FileSync run.

Replication Interval

FileSync replication is batched. The following schedule alternatives are supported:

- *Periodic*: Files may be replicated at fixed intervals in increments of minutes. The shortest interval is one minute.
- *Specified Times*: Replication can occur at specified times throughout the day.
- *Interactive*: An operator can initiate FileSync replication at any time by invoking a TACL script.

- *Event*: Replication can be invoked by a trigger generated upon the completion of some external event.

Also, NETBATCH can be used to schedule FileSync synchronization.

File Selection

Several criteria govern file replication. A file is replicated if;

- The source file's last-modified time stamp is later than the destination file's time stamp (that is, the target file is stale).
- The destination file does not exist.
- The destination file is corrupt.
- The destination file's last-modified time stamp is later than the source timestamp, and the TimeExact option is specified.
- The source and target files have a security-attribute mismatch.

A source file will not be replicated if it is corrupt.

Replication can be made to a target file opened for read or execute access (typically a program object file) if the Rename option is specified. In this case, the open file is renamed; and the source file is replicated to the file name. On the next open of that file, the new file will be opened.

A source file that is open for write access (providing that it is not opened exclusively) can be replicated if the Open option is used. This option should be used with great caution. If the source file is modified during replication, the destination file may be corrupt and unusable.

A file will only be replicated if the user (or program) initiating the request is authorized to access both the source and destination files.

Triggers

FileSync can generate a trigger either before a synchronization job begins or when it ends. The trigger invokes a TACL script, a TACL macro, a TACL obey file, a TACL command, or a TACL RUN command, which can perform any desired function. For instance, programs can be SQL-compiled after a FileSync job completes. A batch program might be launched. Node names can be modified.

Other Features

Fault Tolerance

FileSync can be optionally configured to be persistent in the event of a fault that takes down any of the FileSync processes in either the source or the target systems. This configuration can take two forms:

- The FileSync processes can be configured as checkpointed process pairs so that a backup process will take over should its primary process fail
- The FileSync processes can be started by SCF (Subsystem Control Facility), which will automatically spawn a replacement process in a surviving processor should an executing process fail.

In either case, should a process fail, the current FileSync job that it is processing is terminated. Remaining files will be replicated on the next invocation of the FileSync job.

Parallel Processing

If a FileSync command includes multiple file lists, each file list can run in a different CPU. If multiple instantiations of FileSync are running (for instance, to serve different business units), they can run in different CPUs.

FileSync uses the From/To list specifying the files to be replicated to build a *file synchronization package*. It passes this to other utilities to build an *archive* file, as described later under FileSync Architecture. An archive file has a configured maximum number of replicated files that it can hold. If the number of files in the file synchronization package is too large, the archive file will spawn multiple sub-files. Each of these archive sub-files can be replicated by different FileSync instantiations running in different processors.

Compression

Files to be replicated can optionally be compressed.

Security

FileSync complies with all NonStop security protocols.

System Comparison

FileSync can list all files that are in sync, out of sync, or designated for purging.

Reporting

FileSync provides several statistical options. Every step is logged in a TACL Log file that can be followed to determine the status of a running FileSync job. The completion statistics for each FileSync job are recorded in the Log file and in a History File for later reference and analysis.

The FileSync monitor, FSYNcom, provides a monitoring interface to FileSync. FSYNcom will report the status of a currently running job using the Log file and the completion statistics based on the History File. FSYNcom provides one control function, and that is to stop a currently executing job.

FileSync Architecture

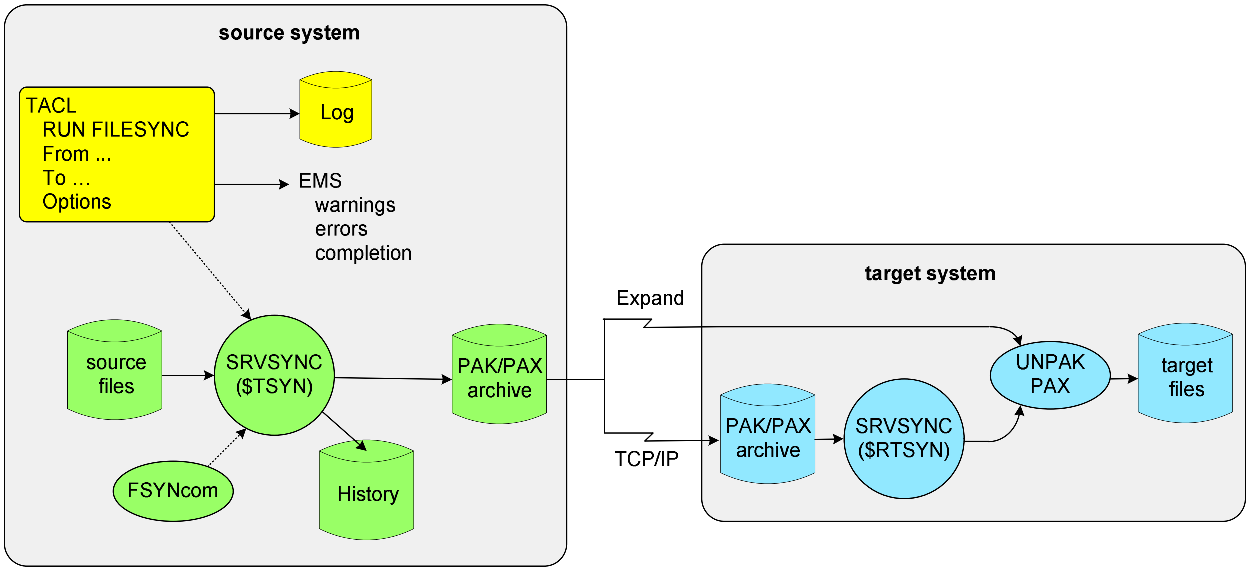
The heart of FileSync is the SRVSYNC process. A copy of SRVSYNC must run on the source system. It must also run on each target system if TCP/IP is used as an interconnect.

A FileSync instantiation is started either manually or by SCF by invoking a TACL script on the source and target systems. The script specifies whether FileSync is to be run at scheduled times, at intervals (in increments of minutes), or once interactively by an operator. Alternatively, FileSync can be started by NETBATCH

SRVSYNC is started as process $TSYN (the default name) on the source system. If TCP/IP is being used to connect the sites, SRVSYNC is also started on the target system with the name $RTSYN (default name). If there are multiple instantiations of FileSync, the source and target SRVSYNC processes will have different names.

$TSYN will parse its FROM file list and will check the files on the target system to see which files must be synchronized. If Expand is being used, $TSYN can access the target files directly to make this determination. If TCP/IP is being used, $TSYN checks the target files via $RTSYN.

FileSync then builds a *file synchronization package*, which is a list of files to be synchronized. Guardian and OSS files are placed in different file synchronization packages. Guardian file synchronization packages are sent to the NonStop backup/restore utility PAK, which will package the files into an optionally compressed backup archive. FileSync passes OSS file synchronization packages to the open utility PAX (portable archive exchange) utility, which builds a similar backup archive. HP's BR2 is used to archive SQL/MX tables for replication.



If Expand is being used, the PAK archives are read by the UNPAK process on the target system. PAX archives are read by PAX on the target system. These utilities access each file from the archive stored on the source system and write them to the target database. When UNPAK (or its OSS equivalent PAX) completes, the FileSync job is complete.

If the link between the systems is TCP/IP, then the archived file sets are sent to the target system over the TCP/IP channel. $RTSYN passes the archive to the UNPAK or PAX process, as appropriate, which writes the files to the target database. When UNPAK or PAX has finished, $RTSYN notifies $TSYN of the completion.

Upon completion, $TSYN writes the job statistics to the History File and notifies the TACL that the job is complete.

As each step in the process completes, the TACL is notified and writes an event to the Log file. It also generates EMS (Event Management Service) messages for warnings, errors, and completions. EMS messages are sent to $0 by default, but FileSync can be directed to send the messages to another collector if desired.

Operations personnel can use the monitor process, FSYNcom, to monitor the progress of a job by inspecting the Log file. The completion statistics in the History File also can be viewed. In addition, a job can be stopped by an FSYNcom command.

FileSync Audit

FileSync replicates an entire file when something – even one byte – is changed. This is appropriate for small files that seldom change, such as configuration files. However, an Enscribe database file can be very large and very active. Replication of it in its entirety can be prohibitive.

FileSync Audit is a FileSync extension that replicates only changes to an Enscribe file (audited or not). FileSync Audit takes advantage of TANDsoft's OPTA (Online Process Tracer and Analyzer) intercept technology. The OPTA library configured for the interception of Enscribe file operations is bound into a user program that is updating the file to be replicated. No application changes are needed. Each file modification is intercepted and passed by OPTA to FileSync Audit, which captures the change and places it into an audited or unaudited change log.

The change log is periodically replicated by FileSync to the target system. FileSync Audit on the target system updates the target copy of the Enscribe file with the changes in the change log.

FileSync in Action

Lehigh Valley Hospital in Allentown, Pennsylvania, serves as a good example of a FileSync user. The hospital purchased two NonStop Blade systems to use in a production/disaster recovery architecture. They understood from the beginning the importance of keeping these two systems in synchronization.

FileSync was recommended to them by another NonStop shop that had been using FileSync for years. LVH tested it and installed it in April, 2011. Since then, FileSync has performed trouble-free.

LVH uses FileSync to synchronize its unaudited files every four hours and its system pack twice a day over an Expand link. FileSync scheduling is accomplished via NetBatch. Unaudited files include those in LVH's GE Healthcare system as well as a variety of other files used by other applications. Audited files are replicated by Shadowbase from Gravic, Inc.

Rich Karluk, Lead Subject Matter Expert at LVH, observes that "It is pretty impressive that FileSync can check 60,000 files and synchronize 600 changed files in sixteen minutes with very little CPU overhead."

Summary

In many active/backup architectures, ensuring that backup system software versions correspond to those on the production system is imperative to prevent failover faults. FileSync periodically compares the source and target files of importance and updates out-of-date target files.

This still leaves configuration changes that have been made by system utilities such as FUP and SQLCI but that are not reflected in the target system. TANDsoft's Command Stream Replicator satisfies this need. It replicates to the target system changes made to the source-system configuration by NonStop utilities, thus completing the synchronization cycle. CSR is described in the second part of this series.