

## **NonStop AutoSYNC – Eliminating Configuration Drift**

August 2011

A major problem in active/passive systems is *configuration drift*. Typically, both the active production system and the passive standby system must be identical – identical hardware, operating system version, database management system, and application versions. Should the production system fail, the applications must be started on the standby system to continue service to the users. However, if some component – hardware or software – is different from the production system, the applications may not execute properly or may not run at all on the standby system. A failover fault has occurred.

Configuration drift is one of the major causes of failover faults. Because testing failover is so risky and expensive for many applications, exercising failover is often given short shrift by companies. It is all too common for some version of a software application or a script to be changed on the production system but not on the backup system.

An important tool for ensuring that this situation does not occur is to use a facility that maintains the software configuration of the two systems in synchronism. Such a tool is HP's NonStop [AutoSYNC™](#) Software built for HP by Carr Scott Software, Inc. ([www.carrscott.com](http://www.carrscott.com)).

### **HP NonStop AutoSYNC**

#### ***File and Table Synchronization***

AutoSYNC automatically replicates NonStop Enscribe files and SQL tables from a source system to a target system to maintain the two systems in synchronization. Files and tables to be replicated are specified by file-set patterns. The file-set specifications may contain wild cards. Both NonStop Guardian and OSS (Open System Services) operating-system personalities are supported. File copying can be to a remote system, to the local system, or to multiple systems.

AutoSYNC file replication is similar to data replication, which synchronizes two copies of a database; but AutoSYNC focuses on files instead. Data-replication engines replicate individual row or record updates to a database as they are made. AutoSYNC periodically copies to a remote system the files that have been added to or changed on the production system.

AutoSYNC is also useful when an application is distributed across several systems. It ensures that application objects are the same version on each system in the application network.

AutoSYNC allows individual users to manage their own synchronizations independently.

## ***Synchronization Modes***

AutoSYNC can be used in one of two modes – one-time synchronization and scheduled synchronization.

### One-Time Synchronization

Any user can use the SYNCHRONIZE command to copy a file set to which he has been granted access. The file copy happens interactively, and the results are reported to the user. One-time synchronization requires appropriate passwords for the remote system.

### Scheduled Synchronization

A user can schedule file sets to be synchronized according to a time schedule or at specified intervals. He does this by adding a file set to the AutoSYNC schedule via the ADD SYNCFILESET command. He must have been granted access to the files that he is scheduling.

The results of scheduled file synchronizations are reported to the AutoSYNC log file.

### User Authorization

Any user can use AutoSYNC to copy a single file set to which he has been granted access. An AutoSYNC administrator must authorize users to schedule synchronizations. Such authorization can include a maximum priority and the CPUs that can be used. The administrator can also suspend and activate users.

## ***Synchronizing Executing Programs***

If an executable is currently running as one or more processes, it can still be optionally synchronized. The existing executable is renamed, and the new executable is copied. Existing processes are unaffected, but any new process instantiation will use the new executable. When the old executable is closed, AutoSYNC deletes the renamed file.

The same procedure is used for any other nonaudited file type that is currently open.

## ***Synchronizing Files Being Updated***

A file that is open for infrequent updates can be synchronized. AutoSYNC has the ability to synchronize only during certain time periods, like midnight to 8AM. If AutoSYNC determines that a file did, in fact, change during its replication, AutoSYNC will discard the copy and will try again at its next synchronization interval.

## ***AutoSYNC Features***

In many installations, file synchronization is performed by operators using manual or semi-automated methods. This can be a primary source of configuration drift. In contrast, AutoSYNC provides several benefits:

- AutoSYNC synchronizes files and tables without operator intervention.
- AutoSYNC is implemented on top of the HP NonStop fault-tolerant, scalable architecture. It continues to perform even in the presence of processor or process failures

- AutoSYNC error reporting is integrated with the NonStop operating system. Synchronization errors are logged to EMS (Event Management System) or to any other specified reporting facility.
- AutoSYNC is managed from a single command-line interface and maintains an historical record of all synchronization activity.
- AutoSYNC is secure. Users can synchronize only those files to which they have access rights.
- Subvolumes and files can be mapped to a different name on the destination system.
- OSS files can be synchronized by directory.
- Synchronization may be parallelized by running multiple synchronization batches simultaneously.
- Security provisions prevent the unauthorized replication, purging, or replacement of files and tables.

### ***Some Limitations***

There are certain files that AutoSYNC cannot replicate. Files that are opened exclusively cannot be replicated as AutoSYNC cannot gain access to them. Likewise, corrupted source files are not replicated. However, a corrupted target file is always replaced regardless of its timestamp.

AutoSYNC does not coordinate the copying of an Enscribe file with its alternate key files. Therefore, there may be a brief period in which one exists without the other. However, SQL index tables are coordinated with their base tables.

AutoSYNC does not directly support the copying of file partitions. To accomplish this, source partitions must be mapped to target volumes. However, if the primary partition is specified in the source file set, the entire file is synchronized.

### **Triggers**

Triggers allow a user to run a TACL or an OSS shell command after every successful file copy or after every successful file-set copy. A TACL command may invoke a TACL macro, a script, or an executable. An OSS shell command may invoke a script or an OSS executable.

The trigger action may be as simple as altering file attributes or as complex as language or SQL compilation. For instance, a trigger can be used to customize a file that has been replicated to the destination system by changing hard-coded system and device names.

### **Reporting**

It can be a daunting task to determine how frequently files should be synchronized. There is a balance between communication efficiency and the need for current information on the target system.

AutoSYNC can prepare reports showing the change activity of files on the production system. It provides the information necessary for evaluating the amount of change that file sets undergo before they are synchronized. This is done without actually moving any data. The reporting

function allows system administrators to configure efficient synchronization schedules for disaster recovery or for data distribution.

AutoSYNC reports also compare source and destination file sets and indicate their status. They are useful to check the results of scheduled or one-time synchronizations. The reports can also aid in system migrations by providing ad-hoc status on migrated volumes and subvolumes during the migration process.

## **Synchronization Options**

AutoSYNC provides many additional functional options that affect file synchronization:

- Target system files or subvolumes that do not exist on the production system may either be kept or may be purged.
- New files being synchronized can carry the name of the owner and the file security parameters.
- SQL tables can be synchronized if their structures have been changed via DDL commands.
- Temporary files (type ZZ) can be included or excluded in the synchronization process.
- Audited files can be included or excluded in the synchronization process.
- Certain files can be excluded from the synchronization process.
- Synchronization can take place over a NonStop Expand link or via TCP/IP. When using TCP/IP, the production and target systems can have the same name, which is not possible when using Enscribe.
- Data can be optionally compressed by 40% to 50% of its original size to be more efficient with bandwidth utilization. However, this requires more CPU resources.
- To control its operational footprint, AutoSYNC can run at a specified priority or can be throttled to a specified maximum CPU usage percentage.

## **AutoSYNC Architecture**

AutoSYNC comprises three primary components – a command interpreter, a monitor process, and the synchronization processes.

### ***AutoSYNC Command Interpreter***

The AutoSYNC command interpreter is the user interface to AutoSYNC for configuration and administration. It can also be used to perform other functions, such as one-time synchronization of selected file sets. Its syntax is similar to the NonStop system's SQLCI (SQL Command Interface).

### ***AutoSYNC Monitor***

The AutoSYNC Monitor manages synchronization and is driven by the synchronization schedules. It invokes the proper synchronization processes to execute the scheduled synchronization tasks. It distributes the synchronization tasks in the primary and backup systems

across all available systems to ensure scalability. Should there be any problem that prevents a synchronization task from being completed, the Monitor reports this to the EMS event-reporting system.

### **Synchronization Processes**

Two programs are provided to execute the synchronization tasks. The ASYSYNC program executes a one-time synchronization command. ASYSYNCP programs are invoked by the Monitor when it is time to perform a scheduled synchronization. The Monitor starts enough ASYSYNCP processes to handle all of the active synchronizations.

An ASYSYNC synchronization process is started on the target system for each ASYSYNCP process (by the Monitor if TCP/IP is used for communication, or by the ASYSYNCP process if Expand is being used). Using the synchronization schedule, the ASYSYNCP and ASYSYNC processes coordinate with each other to determine the files that have changed by comparing their time stamps. The time-stamp comparison takes into account time zones.

The names of files that have changed and are to be synchronized are returned to ASYSYNCP by ASYSYNC. ASYSYNCP sends these files to ASYSYNC, which creates or replaces the files.

When a replacement is received, ASYSYNC saves the existing file to be replaced by renaming it before writing the new file. If the new file is replaced successfully, the saved file is purged. Should the synchronization task fail for any reason, the original file is restored from the renamed file.

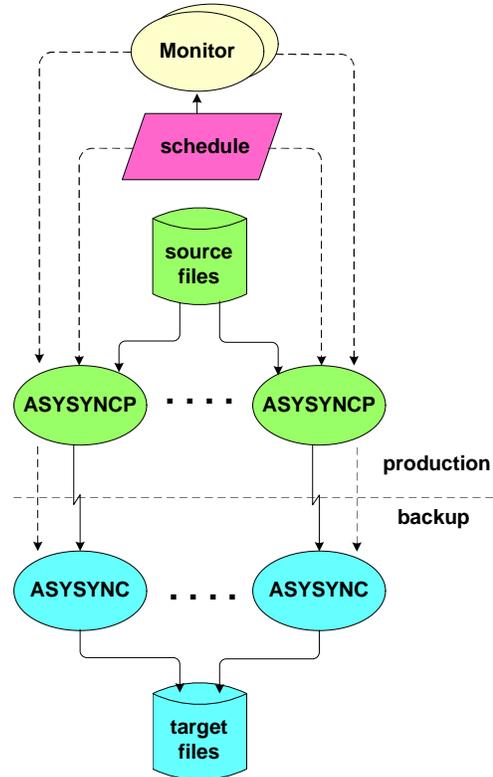
For OSS files, the replaced files are first written under different names. If the synchronization is successful, the current file or directory is purged; and the new file or directory is renamed to the target name.

### **Fault Tolerance**

#### **Architecture**

The AutoSYNC Monitor is a checkpointed, fault-tolerant process pair that survives any single processor failure. It is responsible for correcting faults that affect the synchronization processes. Should a process fail either due to a processor failure or to a process fault, the Monitor restarts the failed process in a surviving processor. The Monitor makes three successive attempts to restart the process and thereafter will try every 30 minutes. Upon recovery, the synchronization task continues from its point of interruption.

If any fault should interrupt synchronization, AutoSYNC will restore the previous version of any file whose copy was interrupted. Once the fault has been corrected, file synchronization continues where it left off.



## **Failover**

AutoSYNC is often used in conjunction with database-replication products to provide business continuity and disaster recovery. As do the replication products, AutoSYNC must also have provisions to bring the database into a consistent state following a failover to the backup system.

### Planned Outages

To take the primary system down for maintenance, all that is required is to gracefully terminate AutoSYNC. First, the scheduled synchronization file sets are inspected to see if there are any outstanding errors. If so, these errors should first be resolved. AutoSYNC is then suspended to prevent any further synchronization.

When the primary system is returned to service, the backup system is paused; and AutoSYNC is used to resynchronize the primary system with the backup system's file changes that occurred while the primary system was down. Synchronization from the primary system to the backup system is then restarted.

### Unplanned Outages

Should the primary system cease functioning and is no longer synchronizing files, the backup system's synchronizing processes perform the necessary file-recovery actions and then terminate themselves. The backup system can be put into service at this point.

When the primary system is returned to service, AutoSYNC must first be disabled on the primary system to prevent old files from being replicated to the backup system. Recovery then proceeds as described above for planned outages.

## **Summary**

AutoSYNC is an important tool for managing the configurations of a primary and backup system in order to eliminate configuration drift. It provides a rich feature set that allows or excludes the synchronization of NonStop Guardian and OSS files whether they are quiescent or active. AutoSYNC is extensible via the use of triggers that can invoke scripts or programs following a successful synchronization task.

AutoSYNC is built on the availability and scalability capabilities of the HP NonStop system. It supports planned downtime for maintenance. It continues synchronization uninterrupted following any single fault and recovers gracefully from disastrous faults. It uses all available resources in the system to provide scalability and parallel execution of synchronizing tasks.