3.0 Disk Safe

3.0 Disk Safe

3.0 Disk Safe Limits

Disk Safe Page Size = 32,768 bytes

One Disk Safe Storage File (.db) for each Disk (e.g. C: drive and D: drive have their own storage files)

Max Pages per Storage File = 2,147,483,648

Storage Limit per Disk per Agent = 2,147,483,648 * 32,768 = 64 Terabytes

Max Storage Files per CDP Server = no limit

2.0 Disk Safe V.S. 3.0 Disk Safe

	2.0 Disk Safe	3.0 Disk Safe
Disk Safe	Block Deltas	ALL data needed for any restore for an Agent in Disk Safe Directory Defined by User
Block Store	Proprietary .data, .idx, .map files	Proprietary .db file with Atomic Journaling using B-Trees stores block deltas
Scalability	8 TB per Disk per Agent (as little 1 TB with 512-byte blocks)	64 TB per Disk per Agent
Crash Recovery	Time Consuming Integrity Check (not always able to recover from all crashes or power failures)	ACID Transactions Atomic Commit Rollback Journal File Recovers Automatically from Crashes or Power Failures Highly Reliable & Robust
Location on Disk	Fixed, determined by CDP Server	The Disk Safe location is completely user-selectable

Can Be Moved?	No	Can be Opened and Closed Moved from one location to another on a CDP Server Copied from one CDP Server to another Copied across platforms (Windows and Linux) Simple tools like Windows Explorer drag-and-drop, cp, xcopy, ftp, scp, etc. can be used to move Disk Safes

	2.0 Disk Safe	3.0 Disk Safe
Memory used During Backup, Replication, Synchronization	Fixed 144 MB / TB / of Disk Safe Size (much more if file system block size is < 4KB) memory used only when Disk Safe is opened	Configurable page cache buffer (default 60 MB) Additional 8 MB memory used / TB of data being added to Disk Safe in single transaction memory used only when Disk Safe is open
Memory used During Browse or Restore	Fixed 144 MB / TB / of Disk Safe Size (much more if file system block size is < 4KB) memory used only when Disk Safe is opened	Configurable page cache buffer (default 60 MB) memory used only when Disk Safe is open
Backup Performance	Fair	Fast
Bare-Metal Restore Performance	Fast	Faster
File Restore Performance	Slow	Fast
Delete (Merge) Recovery Point	Slow	Faster When Deleting Multiple Recovery Points
On Disk Size Can be Shrunk After Large Deletes	Yes- uses "defragment task"	Yes Block Stores are Vacuumed in-place when vacuum now is selected by user

Blocks can be re-organized on disk and optimized	No	Yes block stores can be completely rewritten and freed of internal fragmentation
Data De-Duplication Possible	Not Possible	Very Feasible in near-Future

Atomic Commit and ACID Transactions

With Atomic Commit either all Disk Safe changes in a transaction happen or they don't occur at all. With Atomic Commit in 3.0 CDP Disk Safes small changes and writes can be made all over the Disk Safe files and these changes either appear to happen all at once or not at all. CDP 3.0 Disk Safe transactions appear atomic even if the transaction is interrupted because of an operating system crash or power failure.

Example Disk Safe Transactions:

- Creation of a New Recovery Point
- Merging (Deleting) a Recovery Point
- Creating an Archive of an Existing Recovery Point
- MySQL Recovery Points
- Storage Configuration Backup e.g. Partition Tables, LVM Configuration etc.

Hardware and Operating System Assumptions

1. Your Hardware and Operating System Do Not Corrupt Data
   The Disk Safe assumes that the detection and or correction of hardware bit errors caused by faulty hardware, device driver bugs, bad cables, bad memory, electro-magnetic interference and other such malfunctions are the responsibility of the hardware and operating system. The 3.0 Disk Safe does not add any data redundancy to the Disk Safe files for the purpose of detecting or correcting hardware errors and it is assumed that data can be read later exactly as it was written.

2. File Deletion is Atomic
   The 3.0 Disk Safe assumes that If the Disk Safe deletes a file and power is lost the file either does not exist at all or exists with all of its contents.

3. Operating Systems Buffer Writes
   The Disk Safe expects that the file system will buffer writes and that the write operation will return before the actual data is written to disk. The Disk Safe periodically does a flush of

file write buffers at key points and the Disk Safe assumes that the flush operation does not return until all data has been written through safely to disk. We have heard that some versions of Linux and Windows do not correctly implement fsync or flush. In particular if Linux NFS is used it is highly recommended that Linux NFS Version 4 over TCP is configured. We must assume that a flush or fsync force data through to the disk. If there is no way to fully flush data through to physical media then the Disk Safe is opened up to corruption in the event of a operating system crash or power loss.

Rollback Journal File

Before any changes are made to a Disk Safe file a new on-disk journal file is created. Before any Disk Safe pages are altered the original content of the Disk Safe pages are written to the journal file. This allows any transaction to be completely rolled back. The roll back journal is always flushed completely through to storage medium before any changes are made to the Disk Safe file. This ensures that the roll back journal is always intact in case of a power failure or crash.