Chapter 2

CommVault® Data Management Concepts
The Simpana® product suite offers a wide range of features and options to provide great flexibility in configuring and managing protected data. Protection capabilities such as standard backup, snapshots, archiving and replication can all be incorporated in a single environment for a complete end-to-end data protection solution. No matter which methods are being used within a CommVault® environment, the concepts used to manage the data remain consistent. This chapter provides a basic overview of CommVault data management concepts.

**CommVault Data Management Concepts**

In traditional environments, storage administrators would create a backup of a server and then 'Clone Copy' data to tape for off-site storage. This method was adequate for linear backup strategies, but limited the ability to manage data based on business needs, instead binding the data to media based on which server it resided. Storage policies work differently by allowing the administrator to define protected data logically based on business requirements and not physical locations. This can also be thought of as *Three Dimensional data management* It allows for better media management, improved backup performance, easier recovery, and more flexible retention strategies.

*Traditional backups to tape and clone copy providing little granular management of data. In this case the data is simply treated as servers and no value is associated with business aspects of data protection.*
Logical management of business data using CommVault storage policies. Server data is grouped based on business value and associated with a policy. Based on business and technical reasons for protecting data, the data is placed in different copies to be stored and retained meeting protection requirements.

Three Dimensional Data Management

The concept of Three Dimensional Data Management allows for data to be protected, copied, and managed logically. Data is backed up from the production environment only once, and then additional copies can be created for off-site storage. This Copy Once Reuse Extensively (CORE) concept that the CommVault software uses provides more flexible protection strategies, more efficient media management, and lowers the total cost of ownership.

The primary backup of data from the production environment can be conducted during normal protection windows. This backup of data is considered the First Dimension. An additional copy of the data generated for off-site storage is considered the Second Dimension.

The Third Dimension takes traditional data storage to the next level. It provides the ability to logically manage data independent of its physical location. Logical management of business data is accomplished by grouping production data into logical units called subclients. Each subclient becomes a managed object within the CommVault protected environment allowing you to customize the protection of the subclient data regardless of which physical server it originated from.
Three dimensional storage policy design providing the logical management of data based on business retention and storage needs. Multiple storage policies and multiple copies within each policy can be used to group and manage data in any environment providing great flexibility and security while simplifying administration and minimizing media usage.

The power of three dimensional data protection and policy based data management allows data with like retention requirements to be grouped together. Sending journal Email, financial records, and legal documents off-site for 10 years consolidated on a single tape is much more efficient than sending an Email server, database server, and document server all on separate tapes off-site for 10 years. This concept will be discussed throughout this book.

Policy Based Data Management

CommVault software logically addresses data and data protection methods within a CommCell environment. Data in the production environment is defined in logical containers called subclients. Each of these subclients can be protected and managed independently. The subclients are then scheduled independently or associated with a schedule policy which determines when and how the subclients will be protected. Where the subclients will be protected to and how long they will be retained for is determined in the storage policy. All of these components
can be configured individually and then linked together through configuration options within the CommCell console.

The following diagram illustrates the method CommVault software uses to manage and protect data. Data is defined in containers at the logical level not the server level. The logical containers can all independently be associated with schedules and storage policies. Data containers can share schedule and storage policies or use dedicated policies.

**CommCell® Architecture**

CommVault software requires the coordination of the CommServe® server, Media Agents, Libraries, and Clients. It is important to understand what each of these components do and how they interact in order to gain an overall picture of how CommVault software works.

**CommServe® Server**

The CommServe server is the central management software component of a CommCell® environment. It is installed on Windows Server and will have an instance of Microsoft SQL server installed to hold the CommServe metadata database. The CommServe system is responsible for scheduling jobs, communicating with resources.
such as Clients and Media Agents, and maintaining a database of all activities. It is the essential component required for all functionality and communication must be established and maintained with other components for operations to work properly.

For CommServe server availability it can also be clustered or virtualized. CommServe server high availability is crucial when data archiving has been deployed in the CommCell environment. When using archiving objects are moved from the production environment into CommVault protected storage. A stub file is generated and placed in production storage. When a user goes to access the file a stub recaller redirects the recall to the CommServe server which will then locate the objects and communicate with the Media Agent to recover the object backup to the production environment. If the CommServe server is not available the object cannot be recovered.

Another method for providing CommServe availability is to install the CommServe software on a standby server. This server can be physical or virtual and will have the CommServe software preinstalled. A backup of the CommServe metadata database is conducted one or more times a day and the location of the backup database is directed to the standby CommServe server. In the event of the primary CommServe server being unavailable the standby server can quickly be brought online. If a standby CommServe server is going to be used it is important that the standby server be patched to the same level as the production CommServe server.

The following diagram shows a production CommServe server and different methods to provide high availability and failover. For high availability the CommServe server can be virtualized or clustered. For Failover, a standby CommServe server can be physical or virtual. If an active DR site is available it is strongly recommended to have a standby CommServe server at the DR location.
CommServe DR Backups

By default every morning at 10:00 AM a backup of the CommServe DR Database and CommVault registry hive is conducted. The backup can also be configured to protect important log files and can be scheduled to run multiple times a day if necessary. The backup will be used to restore the CommServe server if the metadata is lost or corrupt. It is important to consider the scheduling of this backup since if the database is restored to a point prior to jobs completing their will be no records of the jobs in the database. In this case the jobs will have to be cataloged back into the database after the CommServe is restored.

The backup process contains three parts:
1. Export
2. Backup
3. Post backup scripts (optional)

The first phase of the CommServe DR backup will dump the SQL metadata database to disk using a folder location or a UNC path. It is strongly recommended to place the export location on a disk separate from the production CommServe. If a standby server will be used set the export location to that server. By default five exports will be kept in the location. If there is adequate disk space available in the export location it is recommended to increase this number to equate to one week’s worth of exports.

The backup phase will use a dedicated DR storage policy or a standard backup storage policy to back up the metadata. To isolate the DR metadata on separate media, use a dedicated DR storage policy. To reduce the amount of media required to be sent off site you can associate the backup phase with a regular storage policy. It is important to note that any storage policy the DR backup is associated with should NOT have the Erase Data option enabled or the data will not be able to be recovered. See the Additional Storage Policy Features chapter for more information on the erase data option.

Another option when backing up the DR database is using post process scripts to copy the metadata to additional locations. This method is useful when multiple standby CommServe servers are being used such as an onsite and off-site CommServe system. The most recent DR dump is always kept in the <install drive>:\program files\commvault\simpana\commservedr folder. This folder can be used as the source data to be copied to additional locations.
The following diagram shows various methods for protecting the CommServe database. The metadata is exported, backed up and optional post scripts can be used to copy the metadata to additional locations.
Media Agent

The Media Agent is the high performance data mover. It is a software component that can be installed on most operating systems and platforms. All of its tasks are coordinated by the CommServe server. The Media Agent moves data from a Client to a Library during a data protection operation or vice-versa during data recovery. Media Agents are also used during auxiliary copy jobs when data is copied from a source library to a destination library.

There is a basic rule that all data must travel through a Media Agent to reach its destination. One exception to this rule is when conducting NDMP dumps direct to tape media. In this case a Media Agent can be used to execute the NDMP dump and no data will travel through the Media Agent. This rule is important to note as it will affect Media Agent placement.

Example: A Database server maintains several terabytes of data located in a Storage Area Network (SAN). The backup location for the data is also in the SAN. By placing a Media Agent module on the same host as the database server, the data can be processed internally within the server and written directly into the SAN. This is called a LAN free backup.

*Diagram of a LAN based backup and a LAN-Free backup. By placing a Media Agent locally on the database server the data path can avoid using the LAN network.*
Client

Client refers to production resources that require protection. A client can be a physical or virtual server, network storage, or end user workstation. A client will have an iDataAgent installed directly on the resource or on a proxy which has access to the resource. An iDataAgent is a software component which directly interacts with the file system or application requiring protection.

iDataAgent

Each Client server requiring protection will have at least one iDataAgent installed. All major operating systems and application are supported by CommVault.

iDataAgent software support:
- File system agents
- Application database agents
- Application object/document level agents

Note: In this book the terms iDataAgent and Agent will be used interchangeably.

Data Set

A Data Set is a logical view of all protected data for which an iDataAgent is responsible. For instance; a data set for a file system iDataAgent will represent every drive, folder, and file on a server. The term data set is used as a generic term to describe backup sets, archive sets or replication sets which are the terms used in the GUI interface. Most iDataAgents will have a Default Data Set. Additional backup sets can be configured if needed, but may result in production data being backed up multiple times.

Subclient

A subclient is the smallest logical management container representing production data. Each backup set will have at least one subclient (default) preconfigured. The default subclient will represent all data within a file system or application that is not otherwise defined within another subclient. This means that data contained in subclients within a backup set will not be backed up more than once using normal schedule settings.
In the following diagram a client has an iDataAgent installed. A data set manages all data the agent is responsible to protect. Subclients are configured which defines the actual content that will be protected.
Libraries

Removable Media Library

A removable media library is any library where media can be moved between compatible libraries within a CommCell environment. Removable media libraries will be divided into the following components:

- **Library** – Is the logical representation of a library within a CommCell environment. A library can be dedicated to a Media Agent or shared between multiple Media Agents. Sharing of removable media libraries can be static or dynamic depending on the library type and the network connection method between the Media Agents and the library.

- **Master drive pool** – is a physical representation of drives within a library. An example of master drive pools would be a tape library with different drive types like LTO4 and LTO5 drives within the same library.

- **Drive pool** – can be used to logically divide drives within a library. The drives can then be assigned to protect different jobs.

- **Scratch pool** – can be defined to manage media which can then be assigned to different data protection jobs. Custom scratch pools can be defined and media can be assigned to each pool. Custom barcode patterns can be defined to automatically assign specific media to different scratch pools or media can manually be moved between scratch pools in the library.

Disk library

A disk library is a logical container which is used to define one or more paths to storage called mount paths. These paths are defined explicitly to the location of the storage and can be defined as a drive letter or a UNC path. Within each mount path writers can be allocated which defines the total number of concurrent streams for the mount path.

Stream management for disk libraries is an important aspect of overall CommCell performance. Depending on the disk’s capabilities, network capacity and Media Agent power, the number of writers can be increased to allow more streams to run concurrently during protection periods. When implementing Simpana client side deduplication the number of disk library streams can be set as high as 50. Stream management will be covered in detail in the *Data Movement* chapter.

CommVault Indexing Methods

CommVault software uses a distributed indexing structure that provides for enterprise level scalability and automated index management. This works by using the CommServe database to only retain job based metadata which will keep the database relatively small. Job and detailed index information will be kept on the Media Agent...
protecting the job, automatically copied to media containing the job and optionally copied to an index cache server.

Job summary data maintained in the CommServe database will keep track of all data chunks being written to media. As each chunk completes it is logged in the CommServe database. This information will also maintain media identities where the job was written to which can be used when recalling off site media back for restores. This data will be held in the database for as long as the job exists. This means even if the data has exceeded defined retention rules, the summary information will still remain in the database until the job has been overwritten. An option to browse aged data can be used to browse and recover data on media that has exceeded retention but has not been overwritten.

The detailed index information for jobs is maintained in the Media Agent’s Index Cache. This information will contain each object protected, what chunk the data is in, and the chunk offset defining the exact location of the data within the chunk. The index files are stored in the index cache and after the data is protected to media, an archive index operation is conducted to write the index to the media. This method automatically protects the index information eliminating the need to perform separate index backup operations. The archived index can also be used if the index cache is not available, when restoring the data at alternate locations, or if the indexes have been pruned from the index cache location.

**Indexed and Non-Indexed Jobs**

CommVault software defines data protection jobs as indexed or non-indexed job types. Indexes are used when data protection jobs require indexing information for granular level recovery. Non-indexed jobs are database jobs where recovery can only be performed at the database level. Indexed based operations will require access to the index cache for creating or updating index files. Non-indexed based jobs do not require index cache access as the backup jobs use the CommServe database to update job summary information.

The following lists the types of indexed and non-indexed jobs:

**Indexed Based Jobs:**

- File system backup and archive operations.
- Exchange or Lotus Notes Domino mailbox level backup and archive operations.
- SharePoint document level backup and archive operations.

**Non-Indexed Based Jobs:**

- All database jobs protected at the database level.

**How the Index Cache Works**

Indexes are generated and maintained at a job level in an index cache on the Media Agent. It is important to note that a job can be an entire server or just portions of the server. CommVault uses subclients to define actual data that is being protected. When a subclient job runs, all indexing information will be kept in index files specific to
that subclient. This means if a server has four subclients defined, there will be four separate indexes maintained for the data.

When a full data protection job runs, by default a new index file will be generate. This means that if weekly full backup jobs are being conducted, each week a new index will be generated when a full backup runs for the subclient. When dependent jobs run (differential or incremental) indexing information will be appended to the index files in the cache. At the completion of each job the updated index will be copied to media. By automatically copying the index to media, the latest index will always be available regardless of index cache availability.

Since the indexes are job based and new indexes are created when full backups run, the index files will not grow very large. The size of the index will depend on how many objects are being protected in the subclient and how often the objects are modified throughout the cycle.

_The following diagram shows the CommVault indexing structure. Job summary data is maintained in the CommServe database. Index files are maintained in the index cache and copied to media after each job._
**Self-Maintaining Index Cache**

Since the index cache contains many small index files it will automatically maintain index files based on the following settings in the **Catalog** tab of the Media Agent properties:

- Index retention time – This determines the number of days index files will be retained for.
- Index Cleanup percent – This determines the maximum size the index cache will consume in the cache location.

It is important to note that these settings use OR logic to determine how long indexes will be maintained in the cache. If either one of these criteria are met index files will be pruned from the cache location. When files are pruned from the cache they will be deleted based on access time deleting the least frequently accessed files first. This means that older index files that have been more recently accessed may be kept in the cache location while newer index files that have not been accessed will be deleted.

*The following diagram illustrates index cache pruning based on retention OR index cleanup percent. These parameters are configured in the Catalog tab of the Media Agent’s properties.*
Shared Access to Indexes

One of the powerful features of the Simpana product suite is the ability to pool storage resources with multiple Media Agents for scalability. Multiple Media Agent paths can be defined to a library and the paths can be configured to load balance or failover. This will require index access to be shared between the Media Agents. When a Media Agent runs a full data protection job it, by default generates a new index. When dependent jobs run, Media Agents will require access to the index files. If multiple Media Agents are being used to run protection jobs they will all need shared access to the index location. This can be accomplished using two different methods:

- Index Cache Server
- Shared Index Cache

Index Cache Server

The Index Cache Server (ICS) is a Simpana v9 feature that uses a dedicated Media Agent to hold copies of index files. Each Media Agent will be configured with a local cache and then log ship index files to the Index Cache Server. By default, log shipping is performed after each chunk is written to media during indexed based data protection operations.

There are several advantages to using an Index Cache Server:

- In a shared library configuration using multiple Media Agents it allows for job continuation in the event that a Media Agent goes off-line. When the CommServe server detects that the Media Agent has gone off line it will redirect the job another available Media Agent. The Media Agent will request the index from the Index Cache Server and continue the job from the most recent chunk update.

- Since index files are being stored in two locations it provides high availability of index information in cache. In this case if a Media Agent goes off line, if the index cache is unavailable or if the index cache server is unavailable, index information will still be accessible from a cache location.

- Media Agents can keep local indexes for shorter periods of time reducing the size of the index cache folder structure and the overall disk space required for the index cache. By using high speed dedicated disks for index cache locations on each Media Agent and keeping the cache folder structure smaller data protection performance will be better.

When indexes are required for data protection or recovery operations the indexes will be retrieved in the following order:

1. Media Agent Index Cache if available
2. Index Cache Server if available
3. Media containing indexes
Diagram showing three Media Agents with local index caches and an Index Cache Server. This configuration will log ship index files to the ICS as each chunk completes successfully.

**Shared Index Cache**

Prior to Simpana v9 the method to allow multiple media agents access to indexes was using a Shared Index Cache. One Media Agent will host the cache and other Media Agents would connect to the cache through a UNC path. If any of the Media Agents not hosting the cache went off-line jobs could continue. If the Media Agent hosting the index cache went off-line then none of the other Media Agents would have access to the cache. This created a single point of failure.
Configuring the Index Cache

The index cache is configured in the Media Agent properties in the Catalog tab. There are several key aspects that should be considered when configuring the index cache:

**Location of index cache** – By default the index cache location will be on the system drive which is not recommended. To change the index cache location, use the Index Cache Directory box to specify a location where you want the index cache to reside. It is recommended to use high speed dedicated disks with adequate space to hold the indexes based on the estimated size the index cache will grow to.

**Size of Index** – There are basic guidelines of how large an index cache should be. However regardless of how large or small the index cache is the indexes will only be retained based on the following criteria:

- **Job retention** – Once a job ages and is deleted all corresponding index files in the cache will also be deleted.
- **Days Retention** – Regardless of how long the job is being retained for once the days retention time expires the indexes will be deleted from the cache.
- **Index Cleanup Percent** – Regardless of how long the job is being retained for if disk usage reaches the Index Cleanup Percent defined threshold indexes will be deleted from the cache.
Since the indexes are automatically written to media, if the index cache does not contain the index it will be read from media and restored to the cache when needed. This may result in a delay before browse results are displayed. The larger the size of the index cache, the longer index files will be retained in the cache and browse results will be returned quicker. This is especially important when browsing data on tape media since the tape must be mounted and the indexes restored from the tape if not in cache which can be time consuming.

As a general best practice CommVault recommends sizing the index cache location to be approximately 4% of the estimated size of all data being protected by the Media Agent. However the index size is determined by the number of objects being protected and not the total size of the data. Large media files will require much less index space than small document files.

Another aspect of sizing the index is how long data will be retained for. If an index cache is managing jobs containing approximately one million objects and retaining the data for two cycles a total of two million index records will be required. Incremental rate of change should also be factored into this calculation which will make this number higher. Technically you can estimate each object entry in an index will require 150 bytes of space over the course of a cycle. One million objects being retained for two cycles will not require too much index space but if the same number of objects was being retained for 26 cycles the index cache will be significantly larger.

The final aspect of index cache sizing and probably the most important is how far back in time browse operations are typically conducted. The farther back in time a browse may need to be performed the more of a chance the index file was deleted from the cache requiring indexes to be restored from media. This means in environments where recoveries are typically performed only within a short period after the data was protected, index cache sizing might not be critical. If recovery requests may potentially be for older data then larger caches should be considered to provide for quicker browse and recovery operations. If browses may be needed for data for extended periods potentially dating back years then consider using an index cache server where inexpensive high capacity disks can be used to retain indexes for long periods.

**Index Files Effect on Browsing**

By default a new index file is created every time a full backup runs and completes successfully. This establishes a browse boundary. This means that data can only be browsed back to the point in which an index file was created. When configuring data protection jobs an option **Create New Index** is enabled by default in the **Advanced** job options. By disabling this option existing indexes will be appended to when full backups run instead of creating a new index file. This will extend the time and date range in which data can be browsed in a single browse operation. This configuration method is referred to as *transparent browse*. It is important to note that de-selecting this option will cause indexes to grow large over time and that indexes will still be copied to media at the conclusion of a job. You can use the Simpana job scheduler to customize schedules to generate new indexes on monthly or even quarterly basis. This method will extend the browse range that can be conducted while preventing the indexes from growing too large.
The following diagrams illustrate browse boundaries established when new index files are generated. By selecting to not create a new index, the existing index file will be appended to extending the browse window.

Transparent Browse Feature
CommCell Architecture

A CommCell® deployment defines the management boundaries of all CommVault components under the control of a single CommServe server. The CommServe system will coordinate all tasks and data movement with the CommCell environment. When agents are deployed they will be joined to the CommCell environment either by specifying the name of the CommServe server at time of install or by registering the agent through the CommCell console after the agent has been installed using the de-coupled install method.

Some environments may require multiple CommCell environments. There is a upper limit of 5000 clients within a single Simpana v9 CommCell environment. Environments larger than this will require multiple CommCell deployments. For geographically dispersed environments multiple CommCell deployments may be used to allow each environment to operate autonomously. Though there is no method for creating a shared CommCell infrastructure, the use of Global Repository Cells can be used to replicate CommCell environment information back to a master cell. This is typically used where remote offices need to function independently of one another but data must be retained and managed at a main data center. Pod Cells are created at each remote location and the Global Repository Cell is set up at the main data center location. The Pod Cells log ship SQL metadata to the repository cell where the metadata is merged into the master CommServe server.