OPENTEXT

November 2014

OpenText Archive Server Functional Paper

Based on Archive Server Version 10.5

OpenText Archive Server is a core component of the OpenText EIM Suite and constitutes the foundation for enterprise-wide Enterprise Content Management solutions. Archive Server enables storage, retrieval and secure long-term retention of archived data and documents.

The OpenText Archive Server is a scalable and integrated service for archiving all of your enterprise content. This content is archived on a secure document repository, giving you the guarantee that all documents are safely stored for years, yet still instantly available when needed

This functional paper describes the architecture and main functionalities of the Archive Server.

Contents

What is OpenText Archive Server

OpenText Archive Server is a core component of the OpenText ECM Suite and constitutes the foundation for enterprise-wide ECM solutions. *Archive Server* enables storage, management and retrieval of archived data and documents. *Archive Server* is delivered as part of the OpenText Content Suite Platform.

OpenText offers customers several connectors to expand the functionality of *Archive Server*. These connectors allow you to manage business documents in different applications and to link them to the business processes. For example, with OpenText Suite for SAP, users can access all data and documents they need to process a business transaction in the SAP business suite. Furthermore, *Archive Server* provides general server interfaces for integrating new or customer specific applications.

Archive Server provides a complete set of services for content and documents. These services incorporate:

- Store and retrieve content
- Content lifecycle
- Storage virtualization
- Caching and Archive Cache Servers
- Single instance archiving
- Long-term preservation and readability
- secKeys and timestamps
- Compression and encryption
- Retention handling
- Backup and replication
- Disaster recovery
- High availability

Architecture

Archive Server provides storage capability for documents and data, and the central archiving functionality.

Archive Server comprises multiple services and processes, amongst which the Storage Manager, the Document Service and the Administration Server are the most important ones. The Storage Manager is responsible for managing external devices, whereas the document management functionality, the storage of technical meta data and other properties, and the entire communication is done by the Document Service..

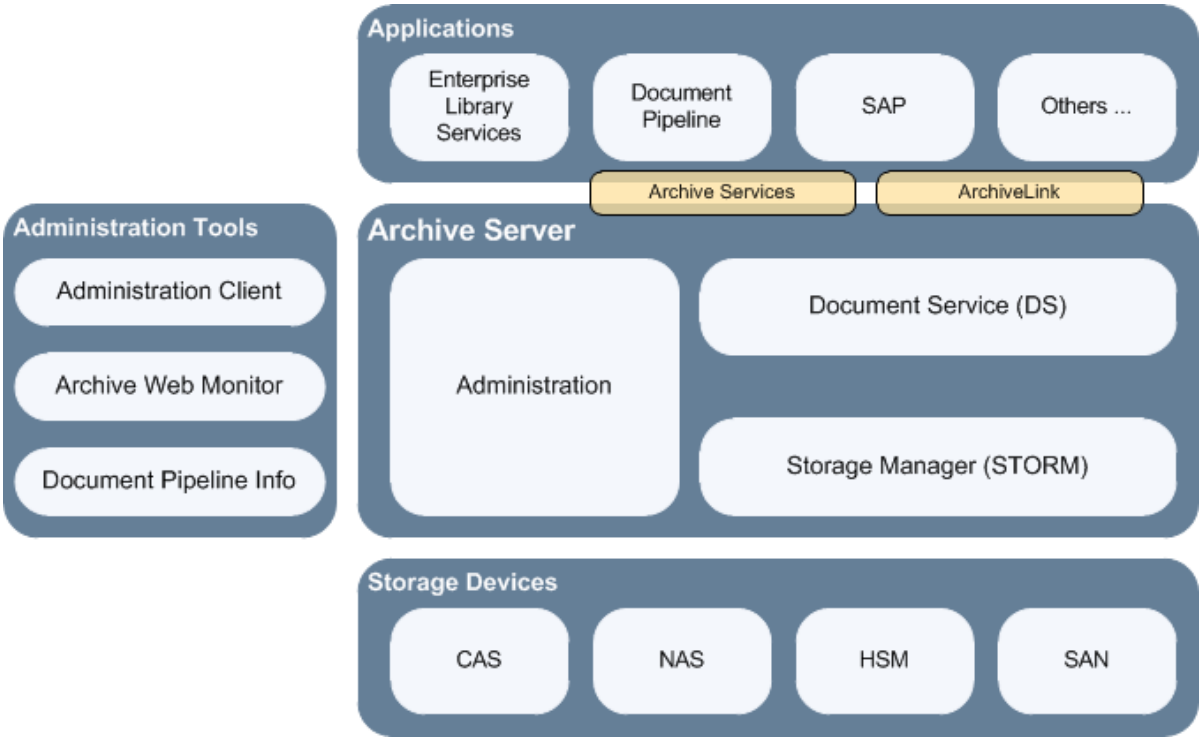
Client applications “talk” to the Document Service. (In the following, *Archive Server* is referred to as a whole.)

Depending on the business process, the document type and the storage media, *Archive Server* uses different techniques to store and access documents. This guarantees optimal data and storage resource management.. The Storage Manager provides access to ISO images within a physical or virtual jukebox. Content that is prone to change and has an individual lifecycle will be stored as single file.

More complex ECM implementations can consist of several *Archive Servers*, for example, to reduce access time in large—possibly worldwide—networks, or to improve reliability by mirroring an entire *Archive Server*. If an *Archive Server* acts as a mirroring system of another server, it is called a Replication Server. Additional Archive Cache Servers complement these servers to a complete, worldwide storage landscape.

Archive Server incorporates the following components for storing, managing and retrieving your data and documents:

- Document Service, which controls the storage and retrieval of the individual components.
- Storage Manager (STORM), which transfers the storage archive to magnetic media and controls the storage devices.
- Document Pipeline, which is used to transport and process the data and documents to be archived. (The Document Pipeline is optional.)
- Archive Cache Server, which speeds up the access to the archived documents. The Cache Server is optional and used in ECM environments, mostly with worldwide, distributed departments and low network bandwidth. The Document Service itself contains a service to cache content from slow media.
- Administration Server, which allows the Administrator to create and maintain logical archives, physical devices, etc.
- In addition, *Archive Server* offers a COLD (Computer Output on Laser Disks) module, which archives COLD and spool data from host systems. The Document Pipeline controls data processing and archiving.



OpenText Archive Server Architecture

Functionality

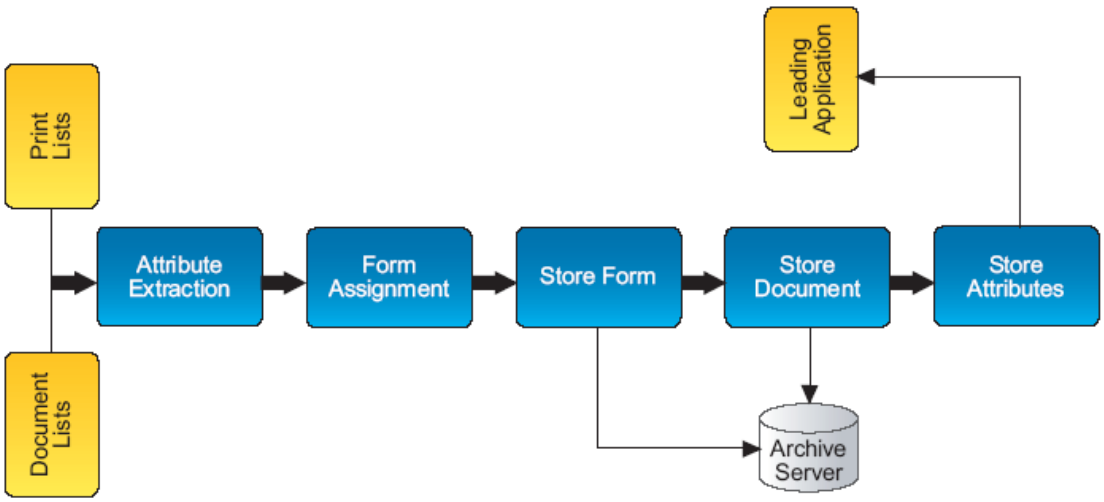
Document Pipeline

A Document Pipeline is the basic component in almost all document-processing software and is used, for instance, to transfer documents to a storage system or another application while performing certain additional tasks. Speaking figuratively, a Document Pipeline is the conveyor belt that transfers the documents through the software.

Individual tools (called DocTools) retrieve the documents from the conveyer belt, process them one by one, and then return them to be processed by the next tool. The last tool in the pipeline generally removes the document from the conveyor belt. Depending on the configuration, Document Pipelines can contain various different DocTools to implement all different kinds of document processing, and further tools can be added as required. An application called “Document Pipeline Info” displays the status of all document pipelines and their DocTools.

A scenario in which the document pipeline plays a central role is *Batch import of print lists or document lists with form overlay and attribute extraction*. On its way through the specifically configured document pipeline, each document has its attributes extracted. In the next step, a form is assigned to document lists.

After the form has been stored by *Archive Server*, the document list is stored together with a link to the form. Finally attributes are stored with the leading application. When users retrieve a document from a document list, it will be displayed together with the assigned form and dramatically improve usability.



Example of a Document Pipeline for batch import with attribute extraction

High Volume Filing

An important principle for all Document Pipelines is that processing is always transactional. That means the processing status of the document is always defined: either it has been processed by a

specific DocTool or not, and no documents can get lost. If for any reason the Document Pipeline is aborted or processing is cancelled at any time, the document is considered to be unprocessed by the last active DocTool. The current status is retained at all times. Therefore, when the Document Pipeline is started again, processing can continue at precisely the same step the document was at when the program was aborted. This re-entrance provides the security required for high-volume filing.

Based on defined Standards

Archive Server uses established standards to help protect your investment, running on various Windows Server versions and all major UNIX versions (including LINUX). The archive database can use an Oracle database or Microsoft's SQL Server. It also supports storage hardware from the leading storage vendors (e.g., EMC, HP, SUN, HDS, NetApp, etc.).

Archive Server stores any content, regardless of its format. Storage of some forms of content is trimmed to optimize the use of storage space or document access, e.g., outgoing invoices which may be numerous but very small. OpenText applications come with a set of clients for imaging and displaying documents.

These clients support existing imaging standards such as TIFF, JPEG, and PDF, as well as SAP formats such as OTF, ALF and ADK. All the desktop applications and the different Windows clients use ODMA (Open Document Management API) to communicate with the archive system. The ODMA interface also allows for seamlessly integrating most applications with the business document system. *OpenText DocuLink for SAP Solutions* offline deploys XML technology, the standard for Internet documents, to provide offline folders and documents to users.

Archive Server supports the SAP ArchiveLink protocol for communication, specifically with SAP systems but also with OpenText Imaging clients and Document Pipeline. The ArchiveLink protocol is based on HTTP resp. HTTPS standard with using SSL.

Scalability and Distribution

Archive Server is built for enterprise-wide deployments. This, in turn, means *Archive Server* has:

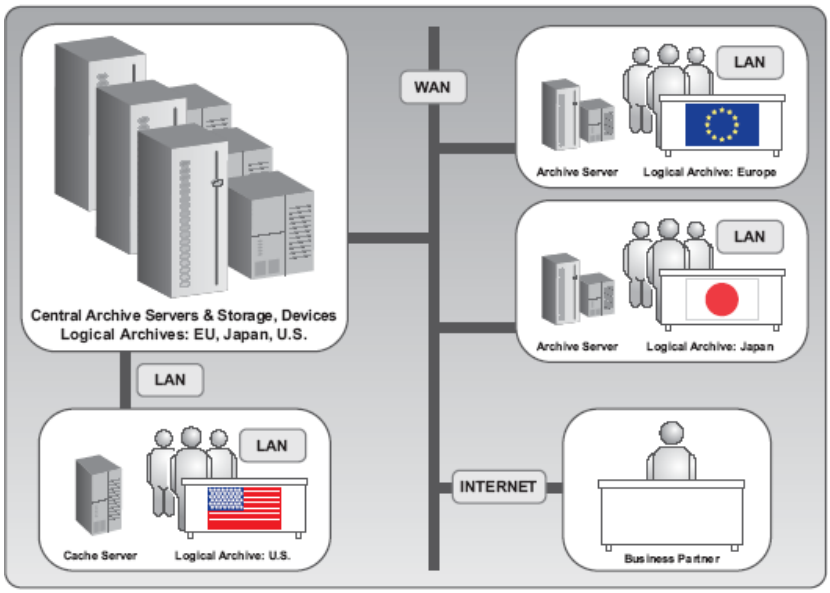
- Strong capabilities in the sense of scalability in document volumes and number of requests.
- Strong capabilities in the ability to distribute the system to all business regions.
- Flexibility to run the system on existing databases and operation systems.
- Flexibility to connect the system to existing or new storage hardware.

Archive Server client/server architecture provides versatile options for configuring, scaling and distributing the archive system. For example, it is equally possible to install multiple Archive Servers to form a large, distributed archive system, as it is to manage multiple logical archives on a single archive server. In addition, no matter how large the distribution, it is possible to centralize system administration and management.

A logical archive can represent an individual content lifecycle with individual storing properties like compression, single-instance archiving and the possibility to set a fixed retention period for this logical archive. The pools attached to a logical archive can represent different storage media, which accommodate the storage requirements of individual content. This allows for implementing customer-specific storage hierarchies and content lifecycle management. For instance, two logical archives are created, one to store contracts and another one for personal signatures. Personal signatures need to be accessed very fast, thus the logical archive should be attached to a HD. Contracts need to be stored on a save medium which ensures they cannot be manipulated. Thus, the logical archive for

contracts can be attached to media with WORM feature. Furthermore, retention periods can be different for individual document types. By assigning and explicitly naming logical archives or pools to individual fiscal years, the administrator has an immediate overview on retentions.

Archive Server can adapt to changing business needs flexibly and cost-effectively. *Archive Server* scales within an Archive Server instance by adding additional worker processes and with adding additional Archive Servers with the “Known Server Concept”, where an assembly of Archive Servers forms one big virtual archive. As the number of users grows, it is possible to connect new clients to *Archive Server*, or to install additional Archive Servers or Cache Servers.



A Distributed Archive Server System

High Performance

Archive Server grants fast (within seconds) and efficient access to very large document volumes and for a large number of users. A single Archive Server can handle billions of business documents and thousands of users.

This has been proved by performance benchmarks in IBM and HP laboratories as well as with the SAP ArchiveLink Load Test. The strong performance results underline the strength and scalability of a single Archive Server instance and give OpenText customers the confidence of deploying an SAP certified enterprise-ready solution. Customers can store ~10 million documents a day (at 10 hours/day operations) on a single Archive Server – and in addition scale linearly with the know server concept.

Scalability is implemented by a configurable number of threads and connections. In addition, various caching mechanisms, designed to suit different business scenarios and system configurations, provide speedy access to documents—such as network and media caching, SSL session caching, and attribute caching.

However, speed of access depends on the underlying storage. It also depends on the access technology used. The archive system provides a technical metadata layer that is used by leading applications to efficiently retrieve documents. The metadata layer is also used to organize the storage

location by logical information (document lifecycle class) rather than only technical information (such as file size or last accessed date, like HSM systems).

Fast access is among the tasks of *Archive Server*, which require high performance. The *Archive Server* fulfills performance requirements for filing (store), backup, replication, migration, deletion and administration.

The Archive Server Release Notes provide a minimal sizing recommendation for productive installations, which already fulfills performance requirements for a large range of archiving scenarios.

Support of Operating Systems and Database Systems

Archive Server is designed for use in heterogeneous IT landscapes, and runs under Windows Server operating systems, all major UNIX versions, and hybrid Windows Server/UNIX environments. It also runs under virtualization software environments.

Operating Systems:

- IBM AIX
- SUN Solaris
- HP HP-UX
- Novel Suse Linux
- Red Hat Linux
- Microsoft Windows

Database Systems

- Oracle
- Microsoft SQL Server
- SAP HANA database

For details, please see the Archive Server Release Notes.

Support of Storage Systems

Archive Server virtualizes the storage layer and, because of this, *Archive Server* supports a wide range of different storage technologies and storage vendors. *Archive Server* supports:

Storage hardware

- Hard disk Write Once media
- CAS – content addressed storage
- SAN – storage area networks
- NAS – network attached storage
- HSM – hierarchical storage management systems
- Cloud storage (e.g. Amazon S3 or Windows Azure)
- Tape devices as nearline storage

For details, please see the Archive Server Storage Platforms Release Notes.

Integration and APIs

Integration with SAP

The integration between Enterprise Library with *Archive Server* and SAP is based on and certified for various standard SAP interfaces:

- SAP ArchiveLink Interface
- SAP HTTP Content Server Interface
- SAP ILM WebDAV Interface (together with other components of *Enterprise Library*)

The SAP ArchiveLink interface — developed in 1992 by SAP and IXOS, an OpenText company — is the most important communication interface between SAP and an external archive system. This standard SAP component allows for linking documents that Archive Server manages with SAP business processes, and provides retrieval through SAP transactions.

The SAP HTTP Content Server Interface is the newest version of the ArchiveLink interface. In addition to accommodate ArchiveLink documents, it allows for connection to the SAP Knowledge Provider which is used e.g., for SAP PLM and SAP DMS.

The SAP ILM WebDAV interface is the successor of the SAP WebDAV XML Data Archiving Interface. The ILM WebDAV interface is used to manage the complete lifecycle of archived SAP data. Enterprise Library together with the Archive Server enforces the retention periods and holds, which are transmitted by SAP for the data archiving files and ArchiveLink attachments.

All these integrations into standard SAP interfaces allow customers to leverage the document functionality of SAP in each and every SAP module. Also through the usage of these standard interfaces, *Archive Server* can be rapidly connected to SAP.

Archiving from Customer Solutions

OpenText provides a general Server API, which enables customers to develop their own archiving solutions. The Server API is available for all supported server platforms and sold as separate license.

Single Instance Archiving

Especially in groupware scenarios, identical documents can be a risk of being stored several times, if emails with attachments are sent to hundreds of recipients and all of them want to archive this email. *Archive Server* enables single instance archiving (SIA), keeping the same document only once in the connected storage devices. Dependent on the amount of expected redundancy of email attachments, SIA can reduce required storage space significantly. SIA can be configured for logical archives and can be restricted to application types or mime types.

Compression

In order to save storage space, content can be compressed before writing to storage system. Compression can be activated for each individual logical archive or content type. All important formats including e-mail and office formats are compressed by default. Compression rates depend on file format and content and correspond roughly to gzip level 6.

Retentions Handling

Archive Server implements retention handling, not retention management. Retention handling enables a leading application to implement retention management. A retention period of a document defines a period, in which it has to be impossible to delete or modify this document. For compliance reasons, it is not enough to set a flag, which enables the software to reject any deletion request against the document. The content of the document needs to be physically protected instead. (e.g. by a storage system with WORM capability) as far as possible.

Archive Server supports fixed retention and variable retention for documents. With fixed retention, an incoming document either has a retention period passed along by the leading application, or it inherits default configuration per logical archive. The retention period is stored on the *Archive Server* and is passed to the storage platform, as far as the storage platform supports the notion of retention. Variable retention is fully managed by the leading application and may not be mapped to the storage layer due to limitations of the storage platform to handle variable retention.

When the retention period of a document expires, it occupies not only wasted space in a companies' content store, but also its can become a liability. In this case, the leading application such as SAP or the OpenText Content Server may sent a deletion request.

As physical storage may not allow immediate physical deletion, or even physical destruction of documents, Archive Server logically deletes a document immediately on request (depending on capabilities of physical storage), and does the physical deletion or destruction asynchronously. The Archive Server will logically or physically delete documents only if no retention period is applied or the retention period has expired and only upon request of the leading application.

Retention handling in *Archive Server* is designed as a top-down concept: A leading application sets the retention period in *Archive Server*. *Archive Server*, in turn, sets the retention period on the storage system. After the retention expired, the leading application has to trigger the purge of the content. Then, *Archive Server* triggers the purge of the files on the storage system.

A leading application can specify a retention period (and a retention behavior) during the creation and migration of a document. If nothing is specified, a default period and behavior is used, configured per logical archive within the administration client. A leading application can prolong the retention period and this will be propagated down to the storage level by Archive Server.

Retention management

Retention management is performed by the leading application, which accesses *Archive Server*'s Retention Handling functionality. For instance, *Records Management* requires classification, retention management, audit trails and deletion of documents. Though most of these requirements have to be met by a records management application, *Archive Server* handles retention periods and keeps track of all changes on document content.

Furthermore, *Archive Server* supports to configure a fixed retention period for a logical archive. Thus all documents written to this logical archive inherit the retention period configured, which is set to start with the date of archival. Documents with the same retention requirements can be sent to this logical archive, for example invoices and other tax related documents.

Volume Migration

Volume Migration is a very important function needed for a long-term ECM strategy and to assure compliance. Compliance requires not only the storage of documents in a safe place, but also the need

to purge them once the retention period has been expired. Therefore, *Volume Migration* is important to retention handling if documents are stored on WORM media.

For this purpose, *Archive Server* administration compiles a list with all volumes containing mostly expired documents. Numerous volumes with mainly expired documents can be reduced to a handful via automatic migration. When the migration is completed, the expired volumes can be removed or purged, thus saving jukebox slots or storage space, depending on the media.

Volume Migration also provides the flexibility to adjust the storage strategy, or to move from outdated storage media/devices to recent technology with more capacity.

Authorization and Authentication

Various laws and regulations require document and data retention to prove services rendered, orders placed and so on. Moreover, many documents and forms are crucial to the company's success, so it is vital to protect and secure these documents against unauthorized access and alteration throughout creation, transmission, long-term archiving and retrieval. The following sections contain information on how *Archive Server* handles security issues.

Secure user authorization

It is essential to protect business documents against unauthorized access. However, that is not always easy or efficient when managing billions of documents over decades. Access control to documents via users, groups and access control lists (ACL) can create high administrative efforts as users leave the company, move and others join.

Since business documents are always accessed by business applications (and are mostly worthless without their business context), *Archive Server* follows a different concept. The business application itself (e.g. SAP, TCP)—and not single users—authenticates at *Archive Server* (signed URL resp., SecKey, Certificates). *Archive Server* expects that the business application has authorized the user of the corresponding request and grants access to documents.

Authentication with SecKey

A very effective mechanism in identifying unknown and unauthorized requests is using access with signed URLs. In this case, *Archive Server* accepts only those requests that were signed by a trusted source (e.g., a special application server). The signature from this trusted source guarantees that the request was initiated by an authorized user.

When a client sends a request to the application server, the trusted source checks the access rights and if they exist, signs the URL and sends it to the client. The client can then access archive Server with this URL. The signed URL contains an expiry time, after which it is no longer valid, for instance two hours.

Within *Archive Server*, the URL signature is called a SecKey, which is part of the Server API and used by all leading OpenText applications, such as Exchange Archive and TCP. *Archive Server* can be configured so that unsigned requests are rejected; i.e. only requests from the explicitly authorized SAP application server are accepted. Thus, even if an attacker obtains a document ID, unauthorized access to the document will be denied.

Secure Data Transport

SSL “Secure Sockets Layer” Communication

In a client-server scenario, authentication and key exchange are performed using asymmetric cryptographic algorithms. These algorithms always need two different keys: a private key and a public key. An SSL server needs such a key-pair. The server’s private key is kept on the server and must not be visible to anyone. However, to exchange a key or to authenticate a server, every client needs the server’s public key. The public key is kept in a certificate, which is usually issued and signed by a certification authority. The digital signature on the certificate makes it impossible to manipulate it. This way, the public key is strongly associated with the name in the certificate. A client, which connects to a server, via SSL, compares the name in *Archive Server*’s certificate with the hostname. If those names don't match, the user should get a warning. By using SSL, authorized and encrypted access to all or individual logical archives is ensured.

Client-server transport secured with checksums

Checksums allow recognizing and revealing of unwanted modifications to the documents on their way through the archive. When clients archive or display documents, checksums are used to identify whether transmission was complete and error-free. The checksums are not signed, as the methods used to reveal modifications are directed towards technical failures and not malicious attacks.

OpenText Imaging Enterprise Scan generates checksums for all scanned documents and passes them on to the Document Service. The Document Service verifies the checksums and reports errors. On the way from the Document Service to the Storage Manager, the documents are provided with checksums as well, in order to recognize errors when writing to the media.

How does the client know that a document is authentic and has been sent by *Archive Server*? OpenText Imaging clients can check the document’s timestamp in order to prove data integrity and authenticity of the document.

Digital signatures

We distinguish two types of digital signatures: personal signatures to handle authentication; and timestamp signatures to ensure data integrity. Although personal signatures are stored with *Archive Server*, the handling is controlled by the leading application. Timestamp signatures provided by *Archive Server* are described below.

Secure, long-term archiving and data integrity

Generally, *Archive Server* archives documents on non-changeable media with WORM feature. These can only be written once, providing excellent security against accidental as well as intentional deletion or alteration. However if an additional level of security is required to ensure data integrity of documents, timestamp signatures can be used.

Timestamps

In order to avoid any unnoticed data loss, even the transmission of a document is secured on its way with the help of checksums. From there, the integrity is secured with the help of signed timestamps. Timestamps ensure that document components can no longer be modified unnoticed after they have

been archived. Timestamps guarantee the authenticity of archived business documents. When tax auditors examine a document several years later, the company can prove that it was saved at a certain time and has not been changed since.

A timestamp is a signed datagram containing an external document's hash value, the current time and date, and additional information. *Timestamp Server* creates timestamps, which means that it digitally signs such datagrams. It has been built to be compatible with timestamp service providers like timeproof, Authentidate, Quovadis, I.CA and Signtrust (see release notes for details). The algorithms supported by OpenText Archive Server are: RIPEMD-160, SHA-256, SHA-512.

To put a timestamp on every document, *Archive Server* needs a service to request timestamps for a document. This can be a special hardware device or timestamp service providers. *OpenText Timestamp Server* allows you to use the time stamping features at no cost. However, it does not provide the same high level of security. *Timestamp Server* consists of two separate programs. One program handles the incoming requests, creates the timestamps and sends the reply. It runs as an *Archive Server* component on any framework supported by *Archive Server*. The other program is the administration component, which offers an interface for the initialization and configuration of the service.

A timestamp is valid for about up to eight years. After a certain time, it loses its security because it is based on a hash algorithm, which may be identified by hackers. Thus, after a certain period of time, signature renewal must be performed.

ArchiSig – Signature renewal for long-term digital signature

In contrast to paper-based documents, the value of digitally signed documents as legal evidence decreases over the course of time. This is particularly due to the following reasons: the employed cryptographic algorithms and the keys lose their security qualification over time. It also cannot be guaranteed that the directories and documents needed for the verification of certificates are available for 30 years or more. In addition, the use of digital signing procedures is often insecure, and information for the subsequent evaluation of the actual security is missing. Concepts to solve these problems have only been developed to a certain extent.

The solution to meet these shortcomings is the *ArchiSig* concept. OpenText is a member of the consortium that elaborates the legally compliant and long-term valid electronic signature. *Archive Server* today supports the *ArchiSig* concept. An *ArchiSig*-generated timestamp with renewal is valid for an unlimited period of time.

Auditing – long-term traceability

All actions of *Archive Server* are monitored in audit trails. Audits are enforced for compliant retention classes. Typical actions to be audited are: create, copy, migrate, timestamp and delete. Administrative changes will always be audited. To access audit information, *Archive Server* provides a tool to view reports, as well as http-based calls for leading applications to display audit information documents.

Encryption of the stored data

Document data, in particular critical data, can be stored on the storage device in an encrypted manner. Thus, the documents cannot be read without an archive system and a key for decryption.

For the document encryption, a symmetric key (system key) is used. The administrator creates this system key and stores it in the Archive Server's keystore. The system key itself is encrypted on the Archive Server with the Archive Server's public key and can then only be read with the help of the Archive Server's private key. RSA (asymmetric encryption) is used to exchange the system key between the Archive Server and the remote standby server.

Document encryption can be activated per logical archive. It is performed when the documents are transferred to the buffer of the logical archive for temporary storage.

Backup, Replication, High Availability and Disaster Recovery

Backup

Power outages, physical damage, outdated media, hardware faults or usage errors can unexpectedly shut down IT operations at any time. *Archive Server* provides a variety of options to optimize the availability of the business documents.

Archive Server backup concept provides maximum reliability. This includes backing up all the hard disk partitions that contain archived documents before they are stored in the optical archive, as well as the operating system and the application software. The system can also generate backups of all the entries in the archive database, and duplicate the optical media, largely as automated functions. Furthermore,

Archive Server can create copies of volumes as backups. The copies may be produced on the local archive server or on a remote backup or standby server. To avoid losing data in the event of a hard disk failure and resume using *Archive Server* immediately, we recommend using RAID (Redundant Array of Independent Disks) technology as an additional data backup mechanism.

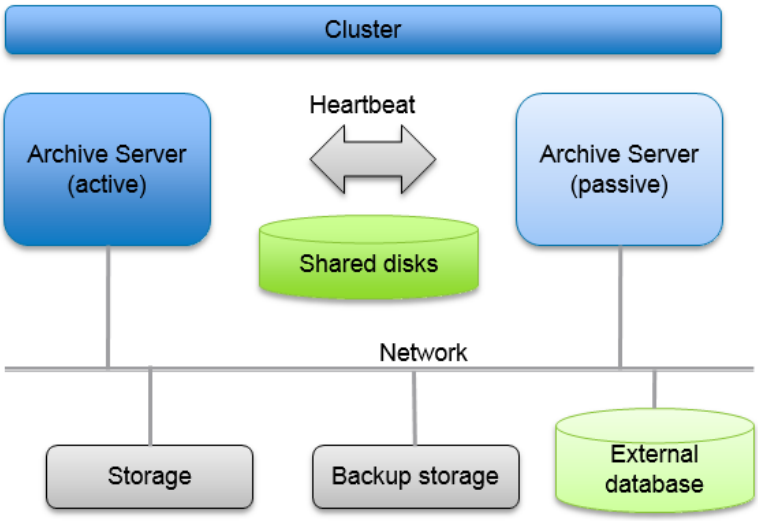
In addition to document content, administrative information is synchronized between original and backup systems.

High Availability

To eliminate long downtimes, *Archive Server* offers high availability by "Hot Standby Server".

The hot standby server is a two-node cluster solution, in which a fully equipped *Archive Server* node monitors the current production system. If a node fails, the other node automatically assumes all activities, with full transparency for end users. *Archive Server* clusters run through a fast LAN, and respond to end users in the same way as a single *Archive Server*.

If the production system fails, users can continue to work normally on the secondary archive system. In contrast to the remote standby server scenario, both read (retrieval) and write (archiving) access to documents is possible in this configuration.



High availability scenario

Remote standby

With a remote standby server, all the documents in the configured logical archive are duplicated on a second *Archive Server* —the remote standby server—via a WAN connection for geographic separation. The remote standby server's configuration is identical to that of the original *Archive Server*. The logical archives and hard disk buffers of the original server are replicated asynchronously. If the production *Archive Server* fails, the remote standby server continues to provide read-access to all the documents. Physically separating the two servers also provides optimal protection against fire, flood and other catastrophic loss.

Disaster recovery

The *Archive Server* stores the available meta data together with content on the storage media (e.g. DocId, aid, timestamp, …). This allows *Archive Server* to completely restore access to archived documents in case the *Archive Server* hardware has a major breakdown or has been destroyed. Technically, the entire database can be restored from the information that is stored on the media. Consistency checks are supplied to check database versus volumes and volumes versus database. In addition, support for a fast delta import after server crash is provided.

Storage Management

Logical archives

A logical archive is an area on *Archive Server* in which documents can be stored. *Archive Server* may contain many logical archives. Each logical archive may be configured to represent a different archiving strategy appropriate to the types of documents archived exclusively there. A logical archive may contain one or more storage pools. Each logical archive is assigned its own exclusive set of

partitions, which make up the actual storage capacity of that archive. Documents are related to a business process, which is handled by a leading application. For example,

- All invoices from the current year are grouped together, so that they can be easily deleted after the retention period has expired.
- HR documents have to be kept separate from financial documents, and special treatment such as encryption can apply.

Logical archives make it possible to store documents in a structured way. You can organize archived documents in different logical archives according to the following criteria:

- The leading application and the module to which it belongs
- The contents of the document
- The retention period
- The archiving and cache strategy
- Storage media types
- Customer relations (for ASPs)
- Text versus productive context
- Protection of documents (authentication certificates per archive)

Hardware abstraction

Key tasks of *Archive Server* include hiding specific hardware characteristics to leading applications, providing transparent access, and optimizing storage resources.

Archive Server like a “Janus”—on the one side, it can handle complex hardware; on the other side, it provides hardware abstraction by offering a unified storage. If a hardware vendor’s storage API changes, or if new versions come up, it’s not necessary to change all the leading applications using the hardware, but only the Archive Server’s interface to the storage device.

Storage reorganization

Content lifecycle may be different depending on the document type, thus imposing different requirements on the storage sub-system. For example, many working copies will be created until a conceptual document (such as a product specification or contractual work) is finalized. Often it is not necessary to store working copies in a long-term archive; sometimes they even can be deleted once the content has been finalized. The finalized version, however, needs to be stored on a save, non-alterable, long-term storage medium.

Another example is incoming invoices. They must be immediately filed on a non-alterable medium. Only during invoice processing, the documents are cached on high-speed storage in order to guarantee very fast access. If retention periods change for existing, archived documents in regulated scenarios, storage needs to be reorganized. Other causes for storage reorganizations are changes in storage strategy, organizational changes or legacy decommissioning.

Storing Files of Any Size

The size of business documents can vary from a few kilobytes up to several gigabytes, and both sizes challenge storage systems. Very small documents may waste much space due to big block size of storage media, and decrease filing performance. Archive Server addresses these limitations with a

special container file technology. Depending on the document type, the business scenario and the storage media, Archive Server supports several types of container files.

Very large documents may exceed physical partition limits. To overcome partition limitations, Archive Server stores big documents (up to 100 GB tested) in several chunks.

Supported storage media

Archive Server supports a wide range of different storage media and devices. Supported storage media are normal hard disk drive storage, hard disk write-once media, tape. *Archive Server* connects to Hard Disk Write-Once media devices from different vendors, e.g., EMC, Hitachi, IBM, NetApp, Sun and HP. Furthermore, various Hierarchical Storage Management (HSM). (

For most recent information, see the Storage Platform Release Information of the OpenText *Archive Server*.

Caching and Cache Server

Local Cache Scenarios on the Archive Server

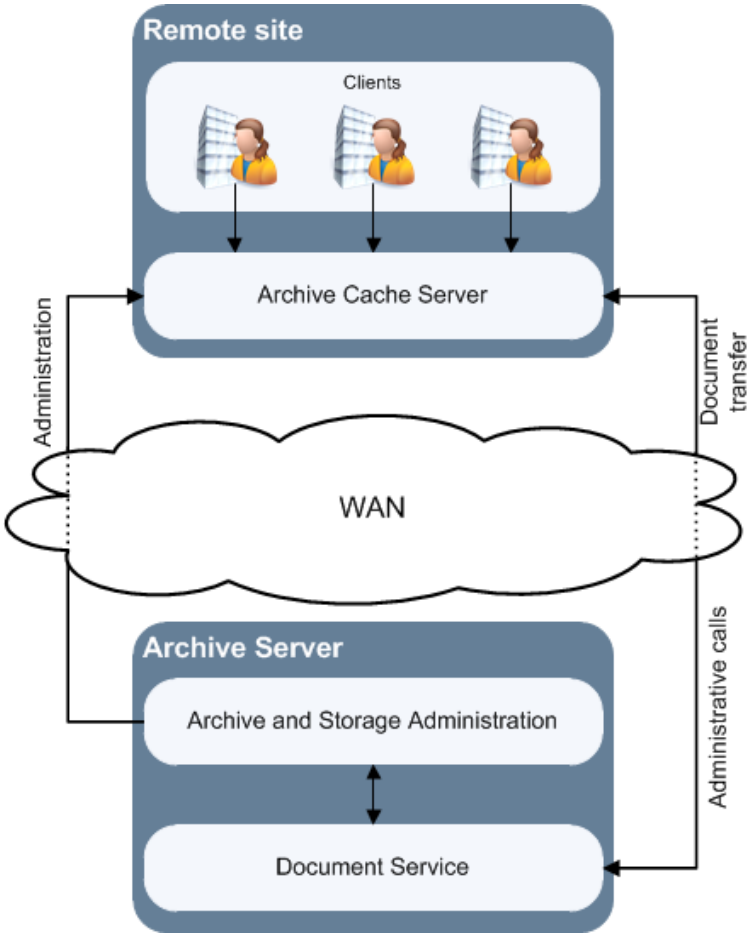
On the *Archive Server*, cache areas can be assigned to logical archives. These caches can be filled upon purging Disk Buffer and by read requests following the FIFO rule (First in – first out). Old documents are removed from cache is cache area is full. Disk Buffers are also used as read cache as long as document copies are in Disk Buffer.

Cache Server

Archive Server supports caching via the Cache Server. It gives users fast access to archived documents.

This is especially important in distributed network environments (such as WAN) because it greatly reduces the network load. It stores locally all the recently read documents and displays them on the client on request. When displaying documents, the Cache Server ensures that the document in the cache is the most current archived original. If several Cache Servers are used, even the logical archives and subnets of the network can be individually configured.

The Cache-Server normally operates in a write-through mode, where all documents that are created locally are stored on the Cache Server and at the same time directly written trough to the *Archive Server*. The Cache Server can be switched into a write-back mode. In this mode all the documents are cached in the local store of the Archive Cache Server only. An administrative job will later transfer these documents to the central *Archive Server*. This mode is intended for architectures with low network bandwidth.



Cache Server Scenario

The cache of the Cache Server is filled upon reading of documents and writing documents, for example when scanning with Enterprise Scan or importing documents via the Document Pipeline. Also all applications using *Archive Server* API will make use of the Cache Server scenarios.

Administration and Monitoring

Archive Server offers the option of centralizing enterprise-wide system administration and monitoring. The SNMP protocol means the entire archive system can be easily monitored from a single control station. Connecting control station software helps make for seamless *Archive Server* operation in a professional data center with specific alert thresholds, refined alert scenarios and active component monitoring. Problem identification in an early stage will avoid time-consuming and expensive downtimes.

Administration Server

The Administration server of *Archive Server* is used to manage and configure the following system components:

- The logical archive, which can be used to group the documents by department, physical location, document type, etc. A retention period can be specified for each logical archive.
- *Archive Server*-Cluster, in which several *Archive Server*s (possibly in different locations) are combined to function as one system for high-availability scenarios.
- The optical media and media pools (for example, automatic WORM finalization)
- Archive Server users
- The timestamp certificate
- The definition and scheduling of the archive jobs

The entire archiving system can be managed either locally or remotely using the Administration Client of the Enterprise Library.

Server Monitoring

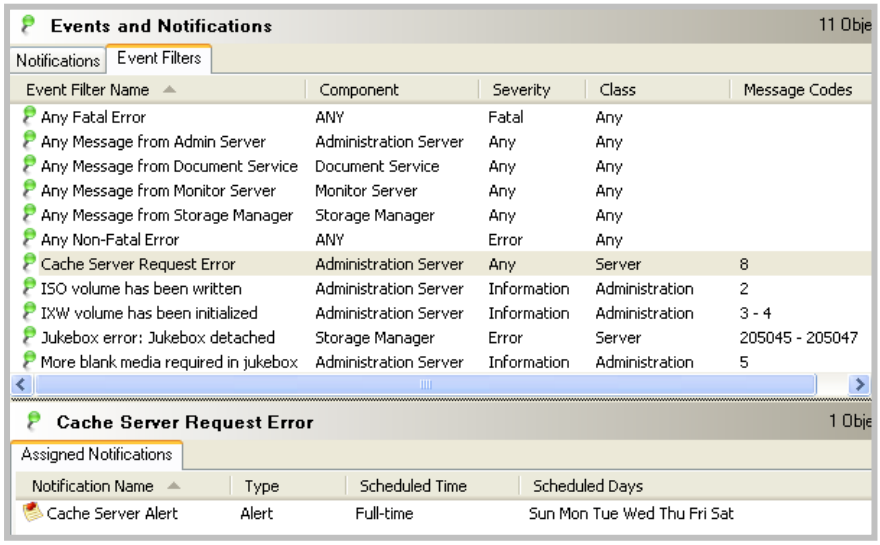
Monitoring ongoing processes helps maintain optimal system performance. For this reason, *Archive Server* includes various monitoring systems that help control the overall system—from the resources for the storage hardware to the individual archiving components' processes.

- Monitor Server with Web Monitor client
- Notification Server
- SNMP Integration
- Log Files
- JMX Monitoring
- CA Wily Introscope Monitoring

Via SNMP standards, it is possible to integrate the monitoring of *Archive Server* with management or monitoring systems from other vendors, such as BMC Patrol, Tivoli or MMS. This lets administrators manage *Archive Server* from within familiar management and monitoring systems. Moreover, log files offer another powerful method for diagnosing *Archive Server*. All the archive components generate log files, which record the activities of the different processes. The log levels' default setting records a minimum of information. If the administrator suspects a problem with a certain component, however, he/she can increase the log level for that component.

The **Monitor Server** helps administrators locate and correct potential problems by using remote procedure calls, SQL queries, and operating system calls to collect and monitor data from the individual components. It continuously saves data about the components' status and the available storage space. The Monitor Server has a web-based Monitor Client that lets the administrator monitor processes. The individual components' processes appear in an intuitive graphical user interface. System resource status and availability appear as symbols.

The **Notification Server** sends notifications, via mail or message, when certain server events (errors, access violations, etc.) occur. You can define these notifications in the Archive Administration.



Events and Notifications

For long-term monitoring, you can have performance data written to **log files**. Logging for each component of *Archive Server* can be individually switched on/off within the Server Administration.

JMX is as specification from the Java Community to administer and monitor Java applications. JMX clients allow displaying monitor information. The Archive Server provides server object monitoring with JMX. The information of the Archive Server Web Monitor is in addition exposed via JMX. JMX monitoring can be displayed for example in Jconsole or SAP NetWeaver CE Administration.

With **CA Wily Introscope**, Single Java methods can be “watched”. Methods of interest for monitoring purpose are listed within a probe directive file. As a result, Introscope agent sends information to Introscope Manager that collects the monitored information. Introscope is available to SAP customers as part of SAP Solution Manager Diagnostics.

About OpenText

OpenText provides Enterprise Information Management software that enables companies of all sizes and industries to manage, secure and leverage their unstructured business information, either in their data center or in the cloud. Over 50,000 companies already use OpenText solutions to unleash the power of their information.

To learn more about OpenText (NASDAQ: OTEX; TSX: OTC), please visit www.opentext.com.

www.opentext.com

NORTH AMERICA +800 499 6544 • UNITED STATES +1 847 267 9330 • GERMANY +49 89 4629 0

UNITED KINGDOM +44 0 1189 848 000 • AUSTRALIA +61 2 9026 3400