
Saxonite Limited

ArtiSan Mailbox Manager

Product Architecture

Author: Keith Young
Reference: ArtiSan/001
Date: 23 January 2004
Version: 1.0
Distribution: Company confidential

Abstract

This document describes the ArtiSan Mailbox Management product. The purpose of ArtiSan is to manage Exchange mailboxes and to provide Enterprise-wide archival for Exchange data.

© Saxonite Ltd 2004

Change History

- 1.0 Initial
- 2.0 Many corrections to the initial draft version

Contents

Introduction	2
ArtiSan Architecture	5
ArtiSan Services	7
ArtiSan Mailbox Management Services	7
ArtiSan Web Service	7
ArtiSan Administration	8
ArtiSan Components	10
ArtiSan Resilience	17
Product Installation and Support	19

Introduction

This document describes the structure and function of ArtiSan Mailbox Management. It is intended to describe what the product is intended to do, what it consists of and how the product achieves those functions.

Product Overview

ArtiSan is a mailbox management and archiving solution. ArtiSan allows customers to centrally manage user mailboxes, reduces the ongoing cost of email storage and ensures that valuable information can be retrieved quickly and efficiently for compliance and knowledge management use.

The main purpose of ArtiSan is to provide automatic archival of items located in the Exchange repository to long-term storage whilst still allowing access to those items from Outlook and OWA. This is not as a replacement for Exchange backup, but as a complement to it. In the first instance, ArtiSan archives all mail items and their attachments. This forms the vast bulk of material in the Exchange repository. ArtiSan can also archive other item types such as contacts, appointments and so on.

Exchange message store growth is a major problem for Exchange users both large and small. Administrators have tried to ease this growth by using Outlook Archiving to move old messages to personal folders (PST files).

Though this approach limits the growth of Exchange stores, it exposes corporate email content to new risks. PST files on user hard drives are not managed centrally and they may not be backed up and so valuable email content may be lost. In addition, PST files are prone to corruption as they grow in size, again resulting in the danger of lost email and knowledge. Storage costs explode as PSTs consume significantly more space than the equivalent email in Exchange.

ArtiSan adopts a different strategy. Instead of moving items out of the Exchange store, it simply archives the items and replaces them with a “link” to the archived content. These links are called placeholders (or stubs) and are ordinary short mail messages in their own right. There are a number of benefits to this approach. Firstly, these mail messages are structured in such a way as to allow existing users to access the archived content directly from the archive. Secondly, the replacement message is significantly smaller than the original and has the effect of limiting the growth of the Exchange store. Using ArtiSan means that Outlook Archiving is no longer required. Thirdly, the process is automatic, freeing up administration time for other more valuable tasks.

ArtiSan also provides tools to allow the administrator to import user PST files back into the Exchange store (PSTLoad.exe) and from there to ArtiSan remote storage. With PST content back under centralized control, email once more becomes an Enterprise Asset for knowledge and compliance use.

The ArtiSan interfaces to the repository through the ArtiSan Engine. The Engine constructs a standard identifier that uniquely identifies the content that allows it to optimise the storage requirement to single instance storage. ArtiSan can connect to one or more underlying storage systems. When it connects to storage, it is able to apply rules to how it uses that storage. Therefore, ArtiSan can move items, as they age, to cheaper, slower storage media; a mechanism called Retention Policy Management or RPM. This allows the storage administrator to control the cost of storage.

- Because of the archival operation, ArtiSan also provides a full repository of all user mail separate from Exchange storage for the purposes of compliance checking and knowledge management.
- As a side effect, ArtiSan only needs a very limited knowledge of the internal structure of mail content. This simplifies the design and development of the system as a whole and makes ArtiSan very efficient at handling data.
- Many of the design constraints are intended to minimise the cost of rollout of the product. Specifically, ArtiSan does not require any install at the client machine. All ArtiSan software is installed centrally at the server. As far as possible, ArtiSan

configures itself using knowledge queried directly from Exchange allowing it to support SME environments with little user input. However, ArtiSan also provides management tools to allow the larger customer to precisely tailor its behaviour to suit corporate needs.

Who Needs It?

ArtiSan was originally targeted to meet the needs of SMEs. The product is a shrink-wrapped solution that is designed to be easy to install and maintain with limited variation in configurations. A typical customer for the product would be running SBS or a single Exchange Server with a single storage group supporting up to a few thousand users. However, the product also supplies the means to scale the solution to larger organisations.

Requirements

ArtiSan runs on Intel boxes running Microsoft Windows 2000 Server (or above). The product is tightly integrated into standard Microsoft products including MS Exchange 2000. It also employs other standard Microsoft products such as SQL Server and IIS. It is assumed that the underlying Exchange storage mechanism is WSS. It is also assumed that the mail users are either using Outlook or OWA to access the mail repository.

Glossary

CDO

Standard set of objects designed by Microsoft in order to support collaborative processes. Actually, there are a number of CDO variants. The differences between them reflect whether they are targeted at Internet mail protocols such as SMTP or at corporate mail protocols such as MAPI and the specific products they are designed to support, e.g. Exchange 2000, Exchange 5.5, etc. In addition, there are a number of extensions to support external related functions such as HTML rendering and workflow.

Exchange

Exchange is Microsoft's flagship mail and groupware server product supporting corporate email and other collaborative processes such as workflow, appointments, journals and tasks. Older versions of Exchange were described according to a version number, e.g. Exchange 5.5. More recent versions are given the name of the year of issue, e.g. Exchange 2000 and Exchange 2003.

Folder

A "virtual" directory in the WSS used for storing mail and other content.

Mailbox

A mailbox is a collection of folders for a given user or group of users. Note that many of these folders are not used for mail, but for other things like appointments, tasks, journals and so on.

MAPI

MAPI is the Microsoft Messaging API.

Outlook

Desktop client software used to access mailboxes either in the MAPI domain including Exchange or with Internet or X.400 connectivity. Along with email, Outlook provides a rich set of related collaborative services such as journals, task lists, address books and a calendar. Outlook is typically installed as part of the Office package.

OWA

Outlook Web Access is an Exchange tool for browsing mail using a standard browser. OWA is commonly used in Enterprises for remote access to mail by road warriors and other employee who spend significant amounts of time away from the office. However,

OWA has a significantly lower deployment cost and, consequently, it is frequently used simply as an alternative to Outlook.

SharePoint

SharePoint is Microsoft Document Collaboration Portal software. Like Exchange, it uses the Web Storage System as its underlying repository.

WSS

The Web Storage System is the underlying store used by Exchange and SharePoint to store heterogeneous content. It features multiple access points and the ability to store large amounts of a wide variety of content types.

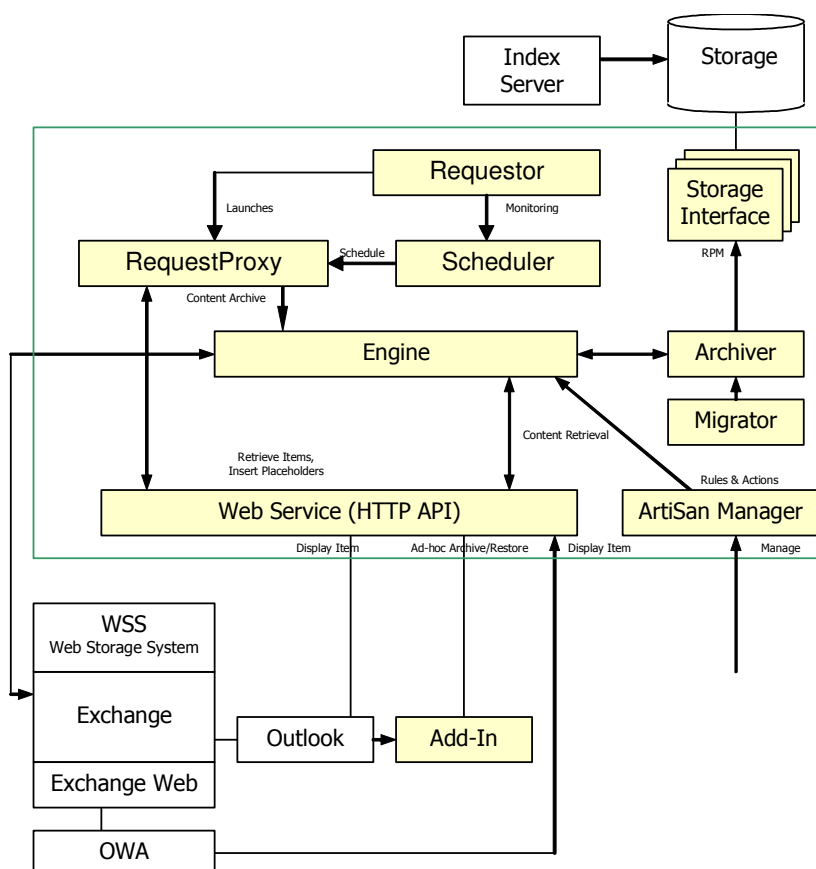
ArtiSan Architecture

ArtiSan employs state of the art techniques to provide enterprise-critical archival and retrieval of Microsoft Exchange content. The product is component-based and is tightly integrated into standard Microsoft products such as Exchange, Outlook and IIS.

As stated in the brief introduction, one of the requirements of ArtiSan is to minimise rollout costs of the product. Consequently, no software needs to be installed on client machines. Installation relies on little more than a simple setup program and customer configuration through the ArtiSan Manager interface.

Architectural Overview

The schematic below illustrates the basic architecture of the ArtiSan product and illustrates how it integrates with standard Microsoft products. All the components shown (with the exception of the Outlook Add-in) run on the ArtiSan server. The add-in communicates with the ArtiSan server through an HTTP API. Consequently, the Add-in is very lightweight.



The ArtiSan architecture can be sub-divided into three main areas: Mailbox Management, Web Service and Administration. The Mailbox Management sections of the system for the bulk of the system and are those components that create and manage archival and retrieval of mail items. The Web Service sections are responsible for access and display of archived items. The Administration section provides customer configuration, rules specification and control of the system as a whole.

Most of the interface boundaries between modules are implemented using COM. For instance, the Requestor, Scheduler and Migrator are COM Services; the RequestProxy is an out-of-process COM server, the Engine, Archiver and Storage Drivers in-process COM servers. Communications between the modules are implemented using standard COM method calls. Some of the interfaces are private (and completely hidden from the

customer) and others are publicly exposed. A key feature of this architecture is that the system is capable of being scripted by customers. The exception to this rule is the ArtiSan Web Service, which exposes its “API” as a series of HTTP requests. All desktop interaction with the ArtiSan engine is performed through this HTTP API.

ArtiSan also has an integrated full text search service integrating with Microsoft Index Server. The Web Service provides a UI through which individuals can search the archive. This can be more tightly integrated into Outlook by creating a search service folder (the default view of this folder is the find.asp provided by the search service that is part of the Web Service).

Security Context

The security of the software as a whole is a key factor in the architectural design. During installation, a user account is created for the specific use of the ArtiSan software. This user account is given specialised administrative access to all mailboxes within the Exchange. Generally, all the ArtiSan services are executed in the context of this user account.

However, equally, most user-based ArtiSan activities are executed in the context of the executing user (i.e. the Web Service impersonates the client using the browser). ArtiSan leverages the native security subsystem presented by the Windows operating system.

The document “A Brief Guide to NT Security” provides more detail on the security requirements of the system.

ArtiSan Mailbox Management Services

The ArtiSan Mailbox Management services are responsible for the archival and retrieval of items from long-term storage.

Mailbox Management is driven either by:

- a) A background task or;
- b) As a result of ad-hoc actions of users interacting with the Outlook add-in;
- c) Customer scripting.

In each case, mailbox items are presented to the ArtiSan RequestProxy, which is responsible for the archive and restore operations.

Where the mailbox management process is scheduled as a background task, the Scheduler process is used to generate work for the RequestProxy. In essence, the Scheduler uses ADS to identify mailboxes that need to be managed. For each mailbox, the Scheduler enlists the help of the RequestProxy to manage the mailbox on its behalf. If the operation is driven through the add-in, the requests are sent through an HTTP API (the Web Service) and presented to a RequestProxy. Generally, customer scripts also invoke the RequestProxy to perform operations.

In essence, the central components of the archiving system are the RequestProxy, the Engine, the Rules System, the Archiver and the Storage Drivers.

The RequestProxy communicates with the Exchange using MAPI to identify individual folders and mail items. The RequestProxy then invokes the Engine to determine whether any work needs to be performed on the item given. The Engine evaluates a set of customer-specified rules (using a separate in-process COM server called Rules) to determine an appropriate action to take. If the action involves archiving the mail, the Engine passes the item to the Archiver, where it is recorded in the ArtiSan database and from there passed to the appropriate Storage Driver for archival. At this stage, ArtiSan also records the security context of the item and the retention period as specified in the rules. If the action also requires a stub to be created, the Engine creates the stub in the Exchange.

A key concept to understand is that the Archiver can manage many storage drivers and uses configured settings to determine which storage driver is responsible for maintaining a particular item instance. The Migrator process is responsible for managing the archive. It is able to move items from one storage driver to another and is also responsible for RPM (retention policy management).

User can elect to view or restore an item directly from their desktop. This is achieved through a call to the HTTP API (the Web Service). The Web Service communicates with the Engine to retrieve the content from the archive.

ArtiSan Web Service

The ArtiSan Web Service services leverage ubiquitous network services such as HTTP and HTML to integrate archived content into standard tools. In keeping with the rest of the product, the User interface components are installed on the server reducing the rollout costs of the product as a whole.

The ArtiSan Web Service

The purpose of the Web Service is to allow end-users access to archived mail content through the web interface provided by the stub content. The Web Service is implemented as standard ASP script as a back-end to an IIS Web Server running on the ArtiSan server. The main functions of this component are to retrieve archived mail content and to format

the main body of the content as HTML (including handling inline embedded attachments and HREF-style links to other attachments). This HTML is passed back to the user for display within a browser.

The Web Service is fully customizable to allow customers or third parties to configure the ArtiSan system whether that is for branding purposes or to provide value-added functionality. The source code for the Web Service is held in the VSS project called Files\Website.

The Web Service also contains code to communicate with the Index Server.

The ArtiSan Outlook Form

When Outlook attempts to display any item, it uses the message class of the item to determine the appropriate form to use. When an item is replaced by a stub, ArtiSan updates the message class of the item to "IPM.Note.ArtiSan", which causes Outlook to use the ArtiSan Outlook form.

The HTML is based on a template that the customer can amend for branding purposes. Typically, it simply consists of a link to view the item and a link to restore it. When a user clicks on the link a browser pops up displaying the message or a message indicating the success of failure of the operation.

The form is essentially a modified version of the standard mail reading form displaying the HTML of the stub. The form is modified to disable "Reply", "Forward" and "Reply All" functions since there would be no sense in forwarded a stub to another individual. In addition, the customised form has updated icons so that the user sees an indication in the folder that the item is stubbed and may take some time to view (Note: this is subject to some limitations).

The ArtiSan form is installed in the Exchange Organisational Forms Library so that any Outlook user can use it without installing any software on the client machine. It is assumed that the Forms Library is fully replicated between multiple Exchange Servers or that the form is installed on all instances.

The source code for the form is held in the VSS project called Files\Data.

OWA Support

ArtiSan supports OWA because the message stubs are visible to OWA users as standard Exchange messages. OWA users can click on stub links and, assuming, appropriate connectivity and DNS resolution can issues requests to view or restore items.

There is an associated Remote Access Proxy that allows external users to access the archive through a firewall. This applies to both OWA users and to Outlook Add-In users.

The ArtiSan Add-In

Associated with ArtiSan is a COM add-in used with Outlook (ArtiAddin.dll). This allows users to perform ad-hoc archive and restores. The add-in uses the Web Interface to perform those services, i.e. it constructs HTTP requests which it issues to the ArtiSan server. The only dependencies for the Add-In are the standard MSXML components and, obviously, the Outlook host.

The source code for the add-in is held in the VSS project called ArtiAddin.

ArtiSan Administration

Administration of the ArtiSan product focuses on the issues of installation, configuration, monitoring and maintenance.

The ArtiSan Manager

All the configurations of the system including scheduled periods, the various rules and actions are administered centrally through a Web Interface called the ArtiSan Manager.

This is an ASP.NET application written in C#. Note: individual mailbox users are not able to change the rules or actions applied to any folder within their or any other mailbox. This is a requirement of compliance.

The shipped product provides a set of default configurations to be used as a starting point for administrators. In many cases, very few modifications need to be made to these defaults. However, the rules system is provided completely empty; out of the box, ArtiSan has no rules and will not archive or stub any item in the Exchange.

The ArtiSan Manager also allows the administrator to monitor basic statistical information collected by the various ArtiSan components. This allows the administrator to be able to predict likely issues and to proactively resolve them.

The interface restricts its usage to members of the group “AMM Managers” and impersonates the AMM user login. It has pages for setting basic configuration, customer settings, rules editing, monitoring and logging.

The source code for the administration UI is held in the VSS project called ArtisanWeb.

The ArtiSan Scheduler

The ArtiSan Scheduler (Scheduler.exe) is an NT Service responsible for identifying mailboxes that need to be managed. The Scheduler maintains a thread pool, one thread for each currently processed mailbox. Note that the Scheduler only communicates with ADS to identify potential mailboxes to work on, it does not communicate with Exchange, this being the responsibility of the RequestProxy (see below).

As each potential mailbox is identified, the scheduler compares the mailbox to various include/exclude lists to decide whether the user is to be processed. If so, the scheduler allocates a thread from the thread pool to do the work. The maximum concurrent number of mailboxes being processed is a configuration equal to the number of threads in the pool.

For each mailbox found, it checks whether it is in a scheduled time. The schedule times are configured using the ArtiSan Management UI. If the current time is outside a schedule period, it waits for a schedule period to begin (or system shutdown). This allows the system to scan the Exchange storage even if the scan takes more than a complete scheduled period. If in a scheduled period, the worker thread creates a RequestProxy (via the Requestor) and passes the mailbox to the RequestProxy for evaluation.

During evaluation, the RequestProxy communicates with the worker thread giving it status updates. The worker thread monitors these status calls to determine if the RequestProxy is still operational and, if not, will automatically recover.

As each mailbox is complete, the scheduler reads the current CPU/IO utilisation of the host and uses this to calculate a per-user back-off period. The effect of this is to allow the scheduler to consume less CPU when the ArtiSan Server is busy and more during quieter periods. When a traversal is complete, the scheduler backs off for a configurable period and repeats the process. The product is shipped with defaults for all these configurations.

The Scheduler source code is held in the VSS project called Scheduler.

The ArtiSan RequestProxy

The main function of the RequestProxy (RequestProxy.exe) is to provide the interactions with the Exchange for a particular mailbox. It provides Exchange profile and session management as well as the ability to iterate mailbox folders and messages. Again it provides a COM interface to allow its functions to be exposed to the outside world. This is useful because it allows operations on the mailbox to be scripted. Each RequestProxy function is visible to scripting. For instance, it is possible with ArtiSan to create a script to archive mailboxes directly without using the Scheduler.

As each item is complete, the RequestProxy reads the current Exchange utilisation of the session and uses this to calculate a per-item back-off period. The effect of this is to allow the RequestProxy to consume less Exchange resource when the Exchange Server is busy and more during quieter periods.

The RequestProxy source code is held in the VSS project called RequestProxy.

The ArtiSan Requestor

The ArtiSan Requestor (Requestor.exe) is an NT Service responsible for:

- a) Ensuring that RequestProxy services are performed under the correct login account
- b) Provides some security services in delegated systems (systems where the archive is not local to the ArtiSan server).

If the Requestor is not performing functions it automatically closes itself down to preserve system CPU.

The Requestor source code is held in the VSS project called Requestor.

The main functionality of the software is contained in the ArtiSan Engine (Engine.DLL). The Engine implemented as a single ActiveX DLL. This approach simplifies the rollout of the product by reducing the number of elements in the system. COM simplifies the configuration of the product and enhances its flexibility through automation.

The Engine is invoked whenever access to ArtiSan functionality is required and is the single common interface to the system as a whole. Consequently, it normally runs in the context of the ArtiSan RequestProxy, in the ArtiSan Web Service and even within the ArtiSan Manager used for managing the system.

The Engine COM interface also provides an API of the product, though the main API is actually that of the RequestProxy (see below). These APIs are designed to allow third-party products to interface directly to ArtiSan allowing our partners and third parties to add value to the product and allowing the Enterprise customer to leverage corporate knowledge in an efficient manner.

In normal operation, the Engine may be invoked by:

- a) The RequestProxy, as a result of a scheduled scan.
- b) Customer scripts
- c) By the web service as a result of an add-in operation requested by the user.

The main role of the ArtiSan Engine during scheduled archiving is to take an Exchange item and evaluate a set of rules for that item. Rules are only evaluated on rule candidates. A candidate is an item that is either currently un-archived or an item that has been modified subsequent to archival. ArtiSan identifies un-archived items by the use of a set of private properties applied to all ArtiSan archived items. These properties contain the timestamp of the ArtiSan change process. This allows the system to identify items that have been modified after the archival process and allow the system to re-archive those items and re-apply the rules.

Rules are actually implemented in the Rules object, again with a COM interface to allow it to be scripted for test purposes; see below for more on the rules engine.

ArtiSan placeholders are, to all intents and purposes, mail items that contain a copy of the basic header information from the original with details such as subject, recipient lists, sender and so on. They also contain a reference to the storage handle for the item they represent. This is ultimately used by the Web Service to retrieve the item for display in Outlook and by OWA scripts. Since the item is a standard mail item, it continues to appear in the folder. The placeholder is also marked as Archived so that ArtiSan never attempts to archive a placeholder.

The placeholder is also given a specific ArtiSan Message Class and Content Class. The ArtiSan Outlook Form and OWA scripts use this to display the item.

ArtiSan placeholders are significantly smaller than the original item reducing Exchange store overhead. A typical placeholder is only a few hundred bytes whereas a typical message may be more than a factor of ten larger. In addition, placeholders tend to vary less in size, reducing the overhead on the Web Storage System.

If the item has been deleted through retention management or through specific administrative delete, ArtiSan creates a special “deleted item” placeholder to replace the existing placeholder for the item. This placeholder describes the reason for the deletion.

From the user perspective, the item is deleted, though the data is still available from the storage via the ArtiSan interface and therefore still available to compliance and knowledge management tools. The end-user is allowed to delete the placeholder if they so desire, since the actual content is stored on the backing storage. The effect is simply to make that item invisible to the user, as they would expect.

Note that the above operation is implemented in a transactional manner. If any part of the process fails, the item is restored to its original state. This allows ArtiSan to handle storage outages in a customer-friendly manner.

The Engine also has entry points for ad-hoc archives and restores or to copy items from the archive into specific mailbox folders.

The Engine source code is held in the VSS project called Engine.

ArtiSan Rules

The Rules engine is a VBScript hosted rules system. Rules are maintained in a hierarchical form within a standard XML file format.

The customer configures ArtiSan rules and actions on the basis of groups, users and mailbox folders. Generally, the system works from the very specific (an individual mailbox folder) the more general (a mailbox) to the very general (a group or global). ArtiSan automatically applies a default “inherit” rule to all nodes, which can be overridden by the customer. The system searches for all rules related to mail message, checking inheritance until it has collected all rules for that item. It then evaluates the VB Script code associated with each rule in turn until it find a rule stating that the item should be archived. The rules also specify the retention period and whether the item should also be stubbed.

To simplify the creation of rules, the system provides a graphical interface that allows the user to build simple, but powerful rules. This includes:

- Candidate is older than a specific date.
- Candidate is bigger than a given size.
- Folder is bigger than a given size.
- Always (i.e. all defined actions are always performed for candidates).

If the Rules system has determined that a candidate matches a rule configured for the folder, the Rules determines the actions associated with that rule. The first rule match takes precedence. The possible actions include:

- Archive the Item (and mark as such).
- Stub the item

In addition, rules can specify the required retention period.

Rules can also be stored as templates to allow the user to apply a common set of rules to a range of folders. In addition, customers can script rules themselves creating very complex rule sets. The rules can be made much more sophisticated than this. For more information, see the separate document “Introduction to ArtiSan Rules”.

Customers would configure the parent folder of user mailboxes to have rules/actions that apply by default to new users. Equally, if a user creates a new folder, the rules in their parent folder are automatically applied to the child.

Although a particular set of rules can be specified to apply to an individual user folder which can be useful in a compliance environment, generally, in most cases, rules are applied to groups making management very much simpler. Occasionally, it is also useful to be able to have special rules that apply to a journal account.

The ArtiSan Migrator

The ArtiSan Migrator (Migrator.exe) is an NT Service responsible for RPM (Retention Policy Management) and for migrating items between the various storage drivers attached to the system. The Archiver carries the migration process on the Migrator’s behalf; the Migrator is simply used to schedule the actions the Archiver carries out.

The Migrator source code is held in the VSS project called Migrator.

The ArtiSan Archiver

The ArtiSan Archiver (Archiver.dll) is responsible for managing remote storage through one or more storage interfaces and for accessing the ArtiSan database. Like the other modules, the Archiver provides a COM interface that is useful in writing test scripts and similar functions.

On initialisation, the Archiver queries the storage interfaces for their capabilities. The Archiver uses these capabilities to determine how to use the storage. Sophisticated media may well internally support notions such as aging. In this case, the Archiver delegates the decision making process to the storage itself. However, where the media is simpler in implementation, the Archiver provides the necessary functionality. The main capabilities of all storage media are “cost” and “capacity”. ArtiSan seeks to minimise the cost of retrieval of old items. It takes the view that more expensive media tend to have faster access times and should be used in preference to slower cheaper media. Two storage interfaces with the same “cost” should be used equally. Consequently, ArtiSan attempts to fill higher cost media before attempting to use lower cost ones. As a medium gets close to capacity, the Archiver migrates the oldest items to lower cost media on the assumption that the oldest items are the least commonly used. The Archiver retests its assumptions on a regular basis allowing the underlying storage to be configured on the fly.

Another important capability of the storage media is its ability to compress data. Most storage systems provide compression internally. As a result, ArtiSan does not provide a compression function.

The Archiver is also responsible for maintaining the mapping between the storage handle and the stored content. The Archiver can move content from one store to another and this has no effect on the storage handle. This feature allows the Archiver to move content to progressively longer-term storage as items age. The Archiver has a local cache of the details of all archived items (maintained in the ArtiSan database) that it uses to both maintain these mapping and to perform decisions on migration.

The process of archiving an item involves:

- 1) Extracting a unique message ID for the item and checking if the item has already been archived. If so, the current storage handle is returned. This mechanism allows the system to maintain “single instance storage” for an item. In other words, where there are copies of a given item in the Web Storage System, for instance, a mail message copied to more than one recipient, ArtiSan only maintains one copy in long-term storage saving large amounts of valuable storage space. NOTE: the engine does not maintain positional information about an item to identify it. An item in one folder can be moved to another without needing an update to the ArtiSan system. Because of this, ArtiSan never stores or uses information about folders.
- 2) If the item is currently un-archived, the Archiver generates a new storage handle and sends the entire item content as a blob of information to long-term storage via the storage interface. On completion, the interface returns its identifier for the item in remote storage to the Archiver. Nothing is either added or removed from the item at this stage. This ensures the complete integrity of the data.
- 3) Updating the ArtiSan local database with details of the new item in long-term storage. The purpose of this database is to provide the cross-mapping function between items and their long-term storage using the storage handles.

An ArtiSan Archiver may also operate in a “delegation” mode. They are able to supply as well as consume an ArtiSan Storage Interfaces. This allows Archivers to be “daisy-chained” together and allows for the remoting of storage interfaces (see below).

The Archiver source code is held in the VSS project called Archiver.

ArtiSan Storage Interfaces

Customers may have widely differing requirements of the underlying storage provider. In order to provide the archival/retrieval function, ArtiSan employs an ArtiSan Storage Interface. This ActiveX DLL protects the ArtiSan product from the vagaries of particular long-term storage products. The storage interface provides APIs used by the Archiver to query for storage capabilities (see above) and to store and retrieve blobs of data.

Storage interfaces are developed as necessary to support particular customers as additional custom development tasks independent of the licensing of the ArtiSan product. In the first

instance, ArtiSan has a simple file-based provider (located in Archiver.dll) that can be integrated with RSS or a generic storage system.

There is also a demo driver that can be supplied on demand in source code to illustrate how a storage driver is written. The system also has an alpha status (complete but untested) “Hydra” driver (also located in Archiver.dll) that allows multiple storages to appear as a single store to the system.

Other Components

There are a number of other components within the source code tree. The text below gives a brief description. NOTE: some of these components are at various stages of completion.

ArtiSan Archive Indexing and the Permissioning EML Filter

ArtiSan supports full-text indexing through the Index Server. In order to make this work on remote systems there is an indexing service add-in (PermEML.DLL) that allows the index server to control access to files according to permissions of the files (by default, IS does not do this). There are detailed instructions on how to install this add-in.

The source code for the Permissioning EML filter is contained in the VSS Project PermEML.

Remote Access Proxy

The Remote Access Proxy is software that allows external OWA users access to the ArtiSan archive from outside a corporate firewall in a secure manner. In reality, it is a cut-down version of separate Saxonite software called the Remote Access Proxy. The idea is that the software is installed in a corporate DMZ and is used as a web proxy to the archive running on an ArtiSan server. Each connection is secure by user using their internal corporate credentials. In operation, it is very similar to the web aspects of the Exchange front-end server and ISA Server.

Detailed information on the operation of the RProxy is held in separate documentation.

The software is held in the VSS project Files/Optional/RProxy.

Demo Driver

The Demo Driver is a source code project designed to illustrate how to construct a storage driver. The intent is to allow this code to be passed to third parties to allow them to build storage drivers for specialised devices.

The software for the demo driver is held in the VSS project Storage Drivers/DemoDriver.

AMMDiagnostics

The AMMDiagnostics program is a UI tool designed to test and report problem accessing items within an ArtiSan system. The software allows the user to input an ArtiSan identifier for a message and tests each layer of access to identify whether the item is accessible and, if not, the reason for the failure. If the software can retrieve the item, it provides the facility to forward the item to a user specified mailbox.

The software for the AMMDiagnostics program is in the VSS Project AMMDiagnostics.

CreateLicence

The CreateLicence program is the program used to generate end-user licence keys.

The software for the CreateLicence program is in the VSS Project CreateLicence.

LogView

The LogView program is a UI tool used to display ArtiSan log and to be able to start and stop the scheduler service. It is largely subsumed by the Management UI.

The software for the LogView is in the VSS Project LogView.

Mailbox Analysis

Mailbox Analysis is an ROI tool for estimating the storage savings resulting from using ArtiSan. The tool is an Outlook add-in (with its own installation). It walks the entire tree of the mailbox it is used in and calculates the size of saving that would result from using ArtiSan on that mailbox.

The software for the MailboxAnalysis is in the VSS project MailboxAnalysis.

PSTLoad

As stated above ArtiSan provides a facility to allow PSTs to be migrated to the ArtiSan archive. This is achieved via the Exchange. The PSTLoad program provides options for discovery of PSTs (disk searches, network searches, recovery for profiles and so on). Once it has isolated a PST, the program provides options to allow the PST to be imported into the ArtiSan archive. Among the options are:

- a) To move the content (i.e. removing from the PST) or to copy it
- b) To apply rules or to automatically archive and stub each item as it is copied
- c) To delete the PST on successful completion.

The software for the PSTLoad is in the VSS project PSTLoad.

ArtiSan Remoting

NOTE: this software is alpha status (i.e. complete, but untested)

A common requirement is multiple site support. Consequently, the ArtiSan product also supplies secure “remoting” components that allows the storage to be maintained off site or

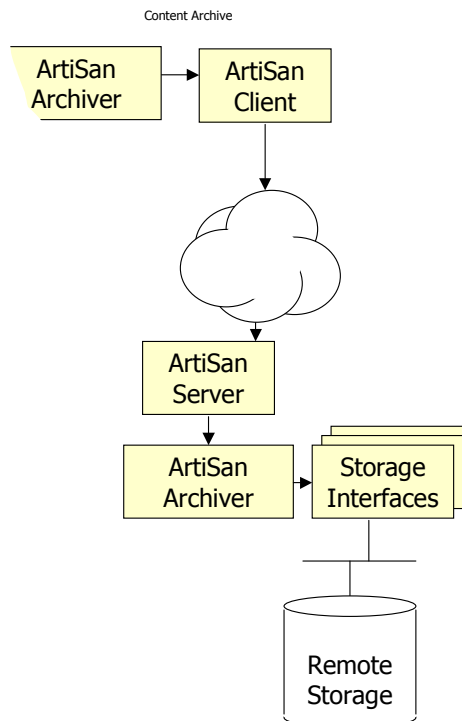


Figure2: Accessing Remote Storage via the ArtiSan Client and Server

by a third party. The ArtiSan Client (ArtClient.dll) is an instance of a Storage Interface that communicates using a virtual circuit to an ArtiSan Server (ArtServer.exe). The ArtiSan Server uses a local Storage Interface to archive or retrieve data from the storage. The ArtiSan client-server uses a specialised transaction protocol to ensure the integrity of remote content. Note: ArtiSan employs a proprietary protocol rather than MSMQ or DCOM due to the poor performance and management restrictions of the latter. This protocol can employ any network virtual circuit protocol. Secured circuits can be achieved using VPNs or private physical links.

In addition, the Archiver may be using a combination of both local and remote storage interfaces. This allows a wide variety of configurations to be built.

The source code for these components is held in the VSS project called ArtClient and ArtServer.

The need for resilience is a function of the requirements of the end-user. Some customers may have regulatory reasons for requiring very high degree of resilience and pay a premium for it. It may be less of an issue for others. ArtiSan caters for this variety of needs through its provision of a variety of underlying storage systems. Some customers may require redundancy through multiple levels of RAID whilst others may simply need some cheap external storage media. As far as the ArtiSan product is concerned, the main core code of the product is shared across all customers and must provide for the most stringent requirements in this area.

Consequently, resilience is central to the ArtiSan product design. In principle, it should be possible for any part of the system to go down without affecting the overall performance of the product. If ArtiSan goes down, the mail system continues to run as before. ArtiSan is optimising the system, not replacing it. When ArtiSan comes up, the system recovers as normal.

A number of types of system outages are possible. ArtiSan employs a variety of techniques to handle each of these.

There are two basic workflows in ArtiSan, scheduled work and on-the-fly work, the latter falling into two categories deletion processing and retrieval. Scheduled work represents the processing of items in the background and applying rules such as migration to those items. This is dependent on access to Exchange, the Archiver and the storage interfaces. Retrieval is the process of restoring items from long-term storage and making them available to users. Deletion Processing is work scheduled by the Migrator where an item is beyond its retention period.

Exchange Outage

In the event that the Exchange Server fails, the scheduler detects the outage through its COM interface and backs off scheduled operations. Items left in the work queue are processed as normal.

ArtiSan waits until the server comes online again, prior to continuing operations. This allows administrators to perform maintenance functions on the Exchange product as a whole.

Retrieval requests for items through the Web Service continue to work, even though no user could actually be logged in to the server to issue those requests. This is a side effect of the design.

Storage Group Outage

Many of the administrative functions of Exchange operate on a per storage group basis. Operators are able to mount and un-mount storage groups.

The Scheduler operates on a per storage group basis, so if an SG is un-mounted, the scheduler waits until that storage group becomes available again. Again, items in the work queue are processed as normal.

Retrieval requests for items through the Web Service continue to work for the un-mounted storage group, even though no user could actually be logged in to the Exchange Server to issue those requests. This is a side effect of the design.

ArtiSan Outages

ArtiSan outages can happen because of bugs and other general exceptions. ArtiSan is coded in a manner that all operations are transactions provided for rollback in the event of an outage. Most outages are handled by simply allowing the schedule function to operate on the next pass minimising the impact of the outage.

Unexpected errors that occur during ingestion of an item are handled by amending the item in the Web Store and marking it “archived” and “in error”. This stops the scheduler

attempting to re-archive the offending item in the future. The benefit of this strategy is that ArtiSan is very resilient at the cost of less overall optimisation.

ArtiSan depends on services such as the ArtiSan database and the storage interfaces that can go offline. In the event of an outage at this level, ArtiSan simply leaves the Web Store unchanged, allowing the item to be archived on the next pass.

Database Outage

If the ArtiSan database goes offline for any reason, Archiver transactions would fail. The result would be that items marked for archiving would never be deleted from the Exchange store and mail usage would continue as normal. Items being queue by the event handler would remain in the queue until the database comes back online. During these periods of outage, old archived items may become temporarily available to Exchange users.

Archiver Outage

If the Archiver fails for any reason, then migration is abandoned and the process is similar to the case for Database outage above.

Storage Outage

If particular back-end storage fails, then items within that storage would become temporarily unavailable to Exchange users. As soon as the storage has recovered this situation would be resolved. The Archiver would either elect another storage for the period of the failure or if none are available fail the operation causing the same effect as above. The Archiver only elects alternate storages that have a retrieval time less than or equal to the failed storage. Exchange users should see no other side effect of this operation.

Network Outage

If there is a network outage, ArtiSan Client and Server components could fail. However, these operate a transaction-based protocol that allows the ArtiSan system to detect the failure and treat it identically to a failure on a local storage system.

Exchange Maintenance Cycles

As stated in the introduction, ArtiSan is not conceived as a replacement to standard Exchange maintenance cycles, but as a complement to them. In the normal run of things, Exchange Storage Groups are backed up. The Exchange backup includes the placeholders inserted by the ArtiSan product. These placeholders are not transient and so are able to be backed up and restored just like any other mail message. However, in order to function correctly, the local ArtiSan database also needs to be backed up in line with the Exchange Storage Groups in order to maintain a consistent image of the current state of the stores.

Exchange Servers and the database can be restored from backup (plus both transaction logs) and the ArtiSan product functions as normal.

Product Installation and Support

Installation

The ArtiSan product is installed through a standard setup program built using InstallShield 10.5. At setup time, ArtiSan collects information about the customer installation and uses that information to create defaults for configuration, etc. This information is logged for support purposes.

The source code for the installation software is held in the VSS project called ArtiSanInstall.

Monitoring ArtiSan

Monitoring information is provided through the Manager as described above.

Operation Logs

The ArtiSan product communicates problems and other informational messages through the standard event reporting mechanism provided by Windows 2000. In addition, ArtiSan maintains more detailed information can be obtained through review of the ArtiSan internal logs – files that are generated during normal operation.

Testing ArtiSan

The system is supplied with a number of test and monitoring programs. These include:

RuleChecker.EXE – A UI tool that allows end users to test the effects of rules

AMMDiagnostics.EXE – A UI tool used to diagnose faults in the ArtiSan system

IDGenerator – A UI tool for testing the hash generation for items in the store.

LogView.EXE – A UI tool for viewing the logs.

As well as a set of test scripts that are described in the installation manual. All the programs are located in the Bin directory of the install.

The source code for the scripts is held in the VSS project called Files\Scripts.

Versioning

All components of the product are versioned. The product is always installed complete to simplify ongoing configuration of the product. Product Updates are only supplied as complete systems (not as patches). This reduces the chance of running mismatched versions of software and simplifies the verification of the software to maintain a high degree of Quality Assurance.

Third-Party Dependencies

The normal run-time requirements for the product are listed in the release notes (i.e. versions of the OS and other software such as IIS, IS and SQL Server 2000) with their appropriate versions. This also includes the DotNet framework version 1.1. The ArtiSan installer installs all other required software.

The product has no additional third-party run-time dependencies.

Product Support

As far as possible, the ArtiSan product maintains itself. In the event of a problem, the product is supported through the supplier chain as first tier support. Saxonite does not, normally, provide direct support to customers for this product. In the event that the problem cannot be resolved through their supplier or their agents, support issues are relayed from the supplier to us as second-tier customer support. This is done through

email. In some special circumstances we may seek, with the approval of the supplier, to contact the end-user in order to expedite resolution of the problem.

Licensing

Typically, the product is licensed on a per mailbox/date/server basis. This allows licensing costs to rise in inline with customer needs and allows us to supply entry-level product at a low cost to SMEs. It also reflects the cost of installing multiple servers as opposed to a single server. However, for larger customers, we are flexible as to these licensing arrangements and may enter into site or corporate level licensing agreements with these customers.

In order to control the licensing, the product is licensed to run on specific hosts. Again, this restriction may be relaxed for larger customers with site or corporate level agreements. In the latter cases, the software is “burnt” with a key indicating that the software is supplied on the basis of that agreement.

Product Organisation

The product is installed to a directory tree on the customers system along with a registry area for the basic configurations of the system such as the location of the installed code.

The directory tree is based on a pair of target directories selectable at install time though these may be the same location. These locations are placed in the registry and used as the primary lookup locations for all files. The directories represent read-only install files and read-writable data areas. No files are installed into any other directory including Windows system directory.

The directory tree has the following sub directories:

- AMManager – The ArtiSan manager UI.
- Bin – All ArtiSan binaries and related resources are held in the Bin directory. Read Only.
- Data – General short-term data storage is held in the data directory such as the logs, the rules configuration file and so on.
- Data\Optional – contains optional components such as the Remote Access Proxy and the Index Server add-in PermEML.
- Tools – Some tools used during the installation of the software.
- Website – The standard web site used by end-users to access the archive

The registry key for the ArtiSan product is contained in the HKLM sub-tree as most of the software will be running in the context of NT services. The main configurations are held in sub-keys on a per module basis. These configurations are described in the file Configurations.txt, which in turn is derived from the Configurations.src file in the Configurations project.