

Corkscrew Content Management

General Documentation

Copyright © 2016. Aquarius Operating Systems, India. All Rights Reserved.

Aquarius Operating Systems and Corkscrew are registered trademarks. Misuse and unauthorized usage of these names or branding will attract penal action and prosecution to the extent of applicable laws.

This is a guide for general users seeking to learn more about the Corkscrew Content Management System to create client, server and other applications. The information in this guide is provided on an as-is basis. Aquarius Operating Systems makes or implies no warranties or guarantees or assurances of any kind or assume any liabilities for damages arising out of usage of the content in this guide, including code examples.

Code examples and scripts provided in this guide may compile and function normally. But these are provided solely for illustrative purposes.

# Getting Started

Welcome to the general guide for Corkscrew Content Management System. We shall call this Corkscrew for short from here on. This guide describes the functionality provided by Corkscrew to general users to create enterprise-class content management applications. We also describe in detail and illustrate with examples the different features that Corkscrew supports.

## About Corkscrew

Corkscrew is a content management system. The features present in the application allow it to be used in a variety of scenarios by a wide range of users. Corkscrew allows its users to store “content” that may be any of –

* Documents (word processors, spreadsheets, presentations, etc)
* Multimedia (pictures, audio, video, etc)
* Archives (Zip files, CAB files, etc)
* Web pages (Html files, scripted files like ASP.NET files, etc)

## What makes Corkscrew special

Like traditional content management systems (“CMS” for short), Corkscrew also lets its users create “sites” and host content within these sites. In those systems, these sites are also “web” sites, having corresponding HTML pages viewable in a web browser. However, unlike those other CMS applications, Corkscrew does not require or mandate these sites to be web sites. In fact, Corkscrew can run without any frontend application at all. This is what makes Corkscrew so special and powerful.

## Scenarios that Corkscrew can be used in

Corkscrew can be used in a wide variety of scenarios. Some examples are given below; this is not an exhaustive list.

* **Corporate knowledge storage and management**

Businesses frequently need a centralized location to store and manage their knowledge in the form of documents, spreadsheets, presentations and other artifacts. Corkscrew can provide these services out of the box.

* **Web content and host**

It is possible to set up and manage multiple websites using Corkscrew as the file system host. Corkscrew provides HTTP modules out of the box that can intercept and serve content to web-based clients.

* **Business process host**

The key enabler for hosting business processes in any system is the ability to perform workflows. Out of the box, Corkscrew provides workflows written using Workflow Services 3.0 (WF3) and, Workflow Services 4.0 (WF4). In addition, Corkscrew also provides its own event-based workflow engine, called the Corkscrew Workflow Services. This makes Corkscrew an excellent host for business process applications.

## Backend scalability and reliability

Corkscrew has loose-coupling between its application and database layers. That is the platform of the application and database layers can both be varied independently of one another. The only limitation to this infinite variability is the support for the chosen database platform on the platform selected for the application layer. For example, if the chosen application platform is the Linux OS, the selected database platform must have drivers for the Linux OS otherwise there will be no connectivity.

Corkscrew’s data structures are very simple structures. The only structure with some complexity is the file system. The file system contains hierarchical data in a flattened structure. However, this table can be easily “replicated” by database and other storage systems that supports change-based replications. Developers and administrators can key off the Last Accessed or Last Modified date-time stamps for configuring such replication.

### Scalability

Corkscrew is a highly scalable application server. The application tier of Corkscrew is a tiny library that includes the object-model as well as the business layer code. Data is persisted in an extensible and variable backend system. The backend system can be any number of providers that can store information in a “tabular” structure --- this can be a database or a file-based storage or even the file system hosted by the operating system.

### Reliability

Corkscrew’s reliability is directly tied into the reliability features of its backend storage engine. For example, if Microsoft SQL Server 2016 is selected as the storage engine for a particular deployment, features like failover clustering, Always On and HADRON can be leveraged by Corkscrew as well. If the storage is persisted into a cloud-based database, like SQL Azure, all the reliability aspects of the cloud also apply to the Corkscrew deployment.

### Cloud readiness

Corkscrew is a cloud-ready application server. Its components are designed to operate independently of each other. This makes it easy to deploy components of Corkscrew applications on the cloud. One can create independent specifically rolled instances on the cloud and deploy its application layers and database layers to those rolls. Once deployed, it can transparently leverage the cloud provider’s built-in features for scalability and reliability.

## Security

Corkscrew security is multi-tier:

* **Operating system based security** – create user accounts in the host operating system and set access control permissions on the folders that contain the Corkscrew application.
* **Database security** – create user accounts in the database engine (SQL Server accounts, MySQL server accounts, Oracle accounts, etc) and use that to tighten permissions on the data.
* **Corkscrew accounts security** – create user accounts in the Corkscrew system and use that to control access to the data stored and accessed through the application.

One can also use additional accounts (like service accounts in the Windows operating system) to control what accounts the application and database engines run under.

Corkscrew permissions system also works on a “deny first” principle where permissions that are ambiguous or where deny and allow permissions are set are evaluated as “deny”.

## Installation and removal

The process of installation and removal of Corkscrew itself is fairly straight-forward. You can use two methods for installation, repair and uninstall –

1. **Recommended** way is to use the provided installer. The installer provides options to install, repair and remove all of Corkscrew’s components. This is tested in various environments and is proven to work reliably.
2. You may also use “XCOPY” (copying the folder structure over to the target folder) deployment. Simply deleting the folder containing the files is sufficient to get rid of the application tier. Databases can be created using bundled scripts or tools, and dropped when not required.

## Data recovery

In case of unexpected failures or data corruption, recovering the system to the nearest “good” point is vital, especially to an organization. Since data is persisted to a backend or database engine or file system, data protection and recovery strategies provided by that storage layer can be leveraged to protect Corkscrew data as well. For example, a database engine’s failover, backup and restore, remote back up, shadow copies, change logs and such mechanisms can be used to recover data. The currency of the data post recovery depends on the aggressiveness of the backup strategy.

## Maintenance and management

Corkscrew is a very low maintenance application server. It does not require any dedicated administration or management effort. Once installed and configured correctly, it can run on its own without any further intervention. That said, associated host systems and software like the operating system, clustering and load balancing infrastructure, database systems etc. would need the requisite administration and maintenance as per their design.

## Editions and flavors of Corkscrew

Corkscrew comes in two editions:

* **Simple** – this edition of Corkscrew does not have Workflows enabled.
* **Enterprise** – this edition of Corkscrew has workflows enabled. However, if you try to install the database bundled with Enterprise edition installer onto a MySQL Server instance, a Simple edition database will be installed. This is because at this time, we do not support workflows when using the MySQL Server database.

## What you pay for

Every distribution of Corkscrew is free to download and use. This is because, once an entity downloads a copy of the software, nothing prevents them from providing copies to another entity. Corkscrew operates on similar principles as open source software:

* The software itself is free to download
* Customers pay for:
  + Technical operational support
  + Developer support
  + Access to the latest bits and patches before they are released to the general public
  + Ability to report specific bugs and get private patches or updates
  + Support for customizations to data structures, etc

## Hosted Corkscrew

Corkscrew is also available for use in a “hosted” mode. This is an instance of the Corkscrew application with all features enabled and hosted centrally by us. This environment can be used freely by developers and other users to try out various features of the application before deciding to host it on their infrastructure. This is a free to use system. However, users must note that data on this environment is not preserved. The environment is always maintained at the latest build level with all applicable patches and updates. At each update, any data (sites, files, etc) are wiped out.

## How to get

Corkscrew is available to download from the “Downloads” section of the Corkscrew website (link: <https://www.corkscrewcms.com>). Before you download, you may be asked to register.

Registration is a simple form that collects a few basic details. This helps us know who is downloading our software, and where and how it is used. This is not used to track your “copy” of the software, this is purely “CRM” data. It will not be used to spam you.

# Installation

The installation process of Corkscrew differs based on what you wish to install. The databases can be installed on any OS environment that support the compatible database engines. However, the other application-tier components can only be installed and used in a Windows environment.

The recommended method of installation is using the provided installer. This is called “CorkscrewSetup.exe” in the official distribution and can be downloaded from <https://www.corkscrewcms.com>.

## Pre-requisites:

The following programs and components must be installed on a target system before running the installer program. If the prerequisites are not found, the installer will warn you about the same. You may choose to pause the installer at that point, install the pre-requisite and then continue to install. Or, you may exit the installer, finish installing the pre-requisites (possibly reboot) and then restart the installer when ready.

### .NET Framework

In order to run the installer and then use the Corkscrew application, you will need to install .NET Framework 4.6.2 on the target systems. You can download the web installer from here: <https://www.microsoft.com/en-us/download/details.aspx?id=53345>

### IIS web server

Only required if you plan to install either the Control Center or the API Service applications. Corkscrew requires IIS web server to be present on the system for these components to be installed. When installing IIS, please ensure the following components are selected (the location of these components may vary depending on your Windows version):

|  |  |
| --- | --- |
|  | Components:   * WCF Services – HTTP Activation, TCP Port Sharing * ASP.NET * .NET Extensibility * HTTP Errors, HTTP Redirection * Static Content   Accept all dependency components that are added automatically. Make sure you select “Static Content” where available, without this stylesheets, pictures etc will not be delivered.  Where version numbers are listed for components, select the highest available version. |

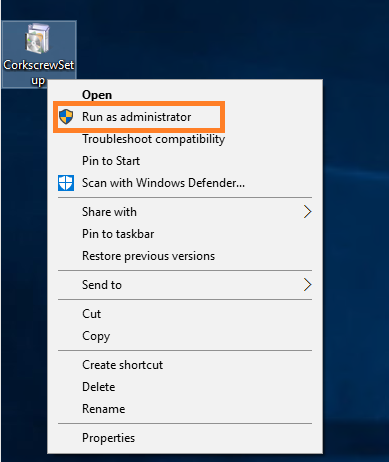
### Administrator privileges

The installer requires that you run it under administrative privileges. You must right-click on the installer program and select “Run as Administrator”. This is not required if you are already running under the built-in “Administrator” user account. Otherwise, ensure your account is added to the local system’s Administrator group (lusrmgr.msc, open “Groups”, select Administrators, ensure your user account is added there).

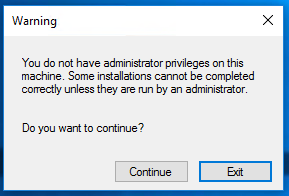
## Installation steps

The installer provided for Windows is GUI and wizard based. Each step asks you to perform selections. Note that existing settings are never displayed on these screens and you are required to enter them again on every run. Therefore, do this carefully.

1. Right-click the “CorkscrewSetup.exe” and select “Run as Administrator”. This launches the installer’s bootstrap. “CorkscrewSetup.exe” is actually an SFX (self-extracting executable) that contains the actual installer application and several CAB files (one for each component).

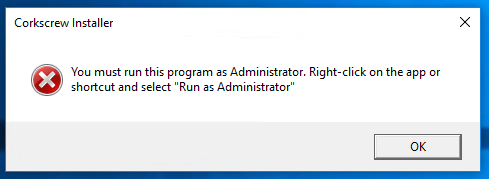


If you do not do this, you will get a warning:



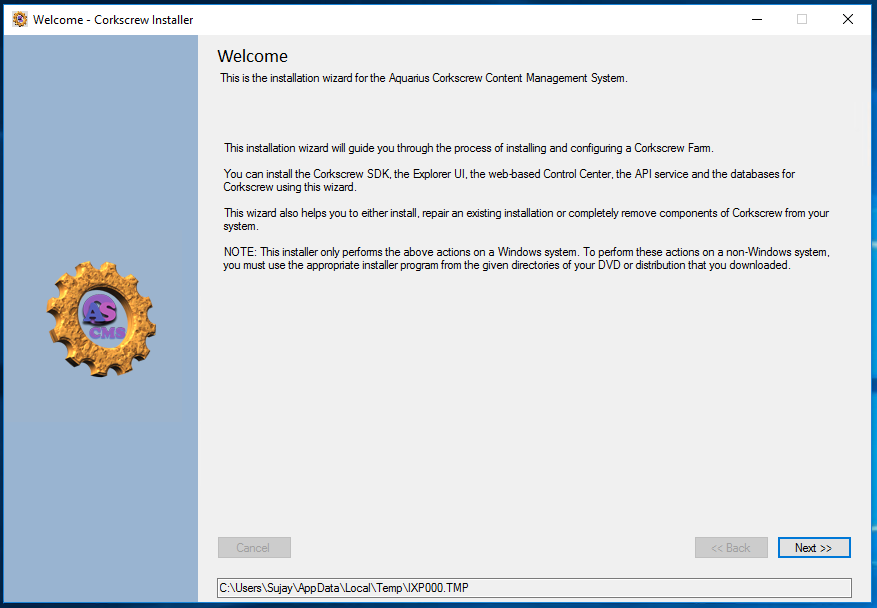
Corkscrew Installer requires elevated privileges in order to read and write to the Windows Registry and also if you are installing components into the Program Files folder tree.

1. You will see a brief “Extracting files” progress bar, for a length of time depending on your system’s performance. The actual installer program starts and checks if you are running the program in Administrator mode. If not, you will see the message:

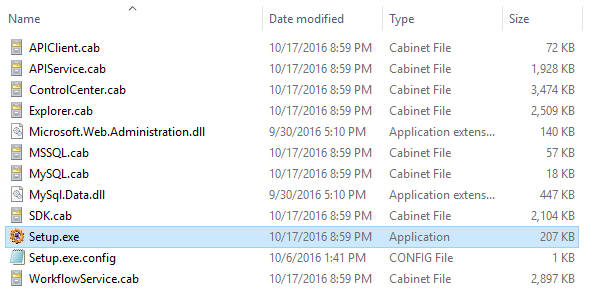


On clicking OK, the program will quit.

1. You will be greeted by the Welcome screen:

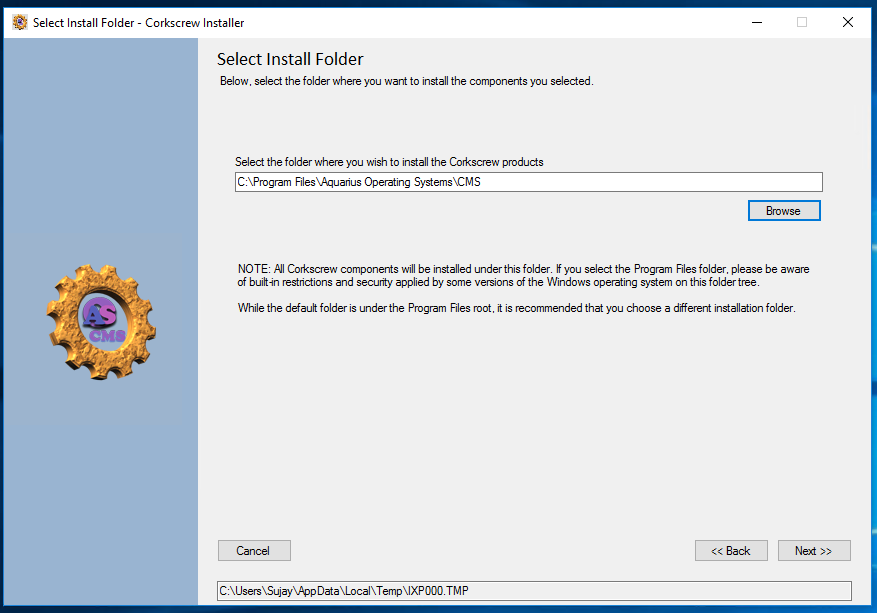


The textbox at the bottom of the screen will display the temporary path from where the installer is running. This where the original SFX application extracted its contents (the CAB files and the actual installer app) to. You can double-click this textbox to automatically open Windows Explorer at that location:



Each CAB file contains the files necessary for that component. No CAB file depends on another – they are independent of each other. “Setup.exe” at the location is what is displaying the wizard screen.

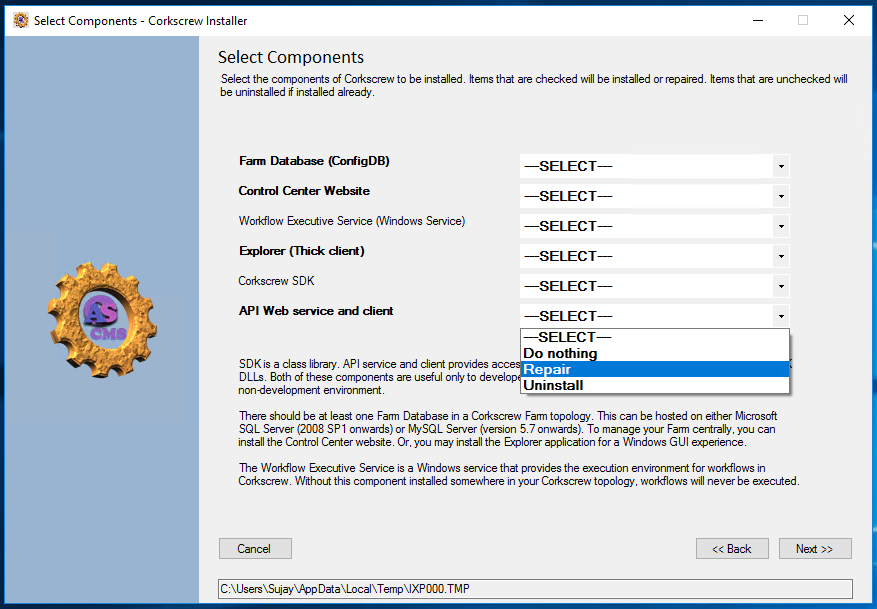
1. Click “Next” on the Welcome screen, you will be prompted to select the installation folder:



Browse to, or type in the folder where you wish the application to be installed. All components of the application will be installed in subdirectories under this folder.

NOTE: If components are already installed, the previously selected installation folder will be displayed here.

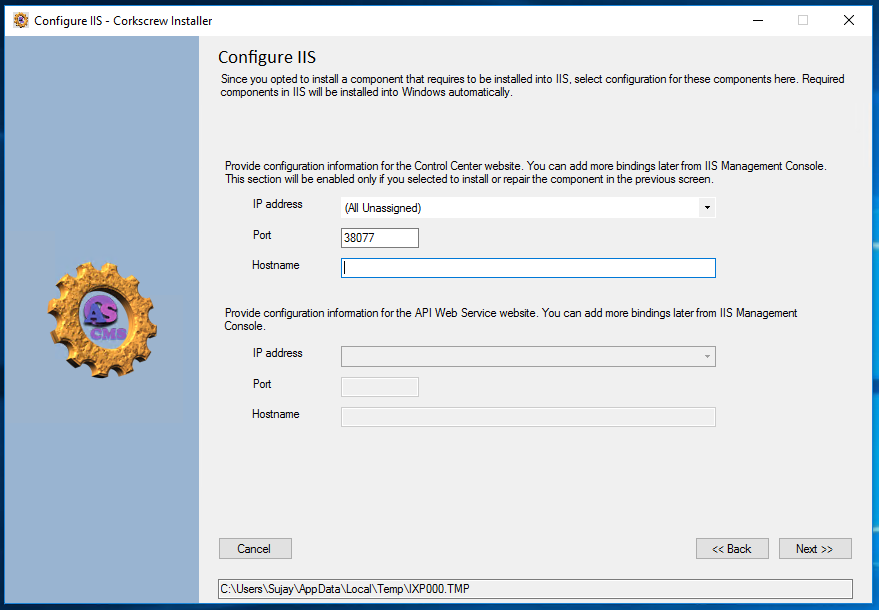
1. Click Next to proceed to the component selection screen.



This screen contains some cheats. Components that are already installed will have their names listed in **bold**. In the above screen, “Workflow Executive Service” and “Corkscrew SDK” are not installed. The dropdowns against each component will contain only the valid actions for each component. For components that are not present the dropdown will contain Install, the others will contain Repair and Uninstall.

Leaving the dropdown as “Select” or “Do Nothing” has the same impact – the current state of the component will remain untouched, no files or settings for that component will be modified.

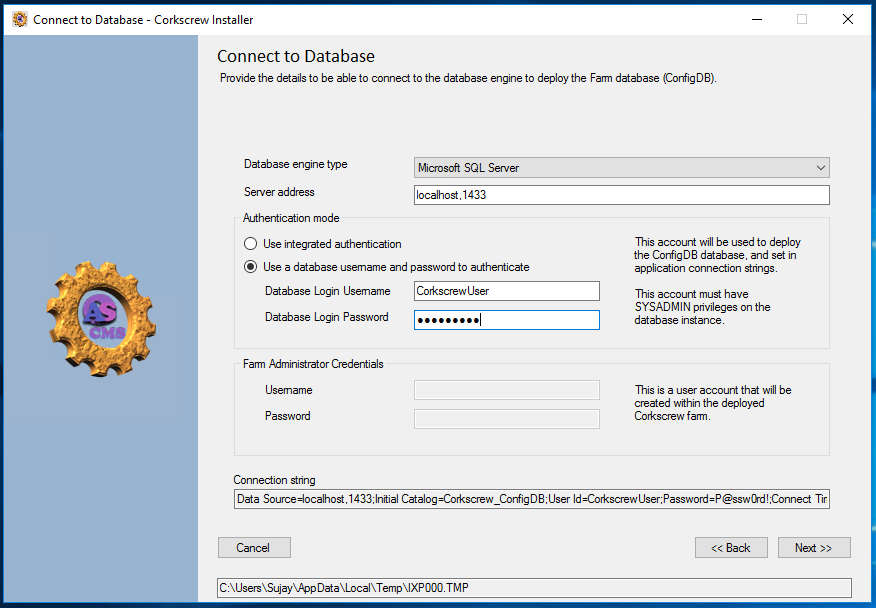
1. On clicking Next, if the Control Center or API Service components were selected for change, the Configure IIS screen is displayed.



This screen lets you select the IP address, port and hostname for each application. The upper set is for the Control Center website and the lower set is for the API Service website.

You can setup SSL bindings and other rules as per the needs of your topology and policy later through INETMGR console.

1. Click Next to display the Database settings page. You can also get here directly from step 5 if no web component is selected.



The database engine type dropdown lets you select either Microsoft SQL Server or MySQL Server. You can then specify the server address (for SQL Server, use the “host,port” format and for MySQL use the “host:port” format).

Select the authentication mechanism. Use the account here wisely – this information will also be embedded into the respective .CONFIG files to be used by the applications to connect to the ConfigDB database. If you need a distinct account for this, create it now (using the appropriate tools for the respective database engine) and enter the details into the screen.

If you selected to install the Farm Database in step 5, the “Farm Administrator Credentials” section would be enabled. This lets you create the first user account, as a Farm Administrator within the Corkscrew Farm you are deploying. Enter a suitable username and password. You may also leave it blank and do it manually later from Control Center (if you have a Control Center deployed in the topology). However, without at least one Farm Administrator account no sites or users can be created and the Farm will be practically useless.

When you click Next, the Setup application will attempt to connect to the provided database with the provided credentials. If the host is down or otherwise unreachable or the address/credentials are incorrect, you will be shown an error. You can fix the problem and click Next again to continue.

1. Clicking Next will proceed with the installation. A progress bar and log of actions will be displayed. Several windows and screens may open and close on their own – do not close or attempt to perform any action within these windows!
2. When the progress screen displays “Completed successfully!”, click the Next button to go to the Finish screen and then click Finish to exit the installer.

The progress screen may also display a “Completed with errors” title. In this case, review the log text displayed for any error messages and attempt to fix the same.

## Uninstallation

To uninstall Corkscrew components, if you installed them by copying files and folders, then simply delete them. Databases can be uninstalled by (preferably first backing them up for safety and then) dropping the database. The Workflow Service can be uninstalled manually using the INSTALLUTIL tool.

Corkscrew is not listed in the “Add/Remove Programs” (or “Programs and Features”) applet. You must run the CorkscrewSetup.exe program again. On the component selection screen, select the “Uninstall” action for components you want to remove.

# Hierarchical object model

Corkscrew has a hierarchical object model for its core objects. The objects are structured thus:

The root of the hierarchy is the Farm. A farm is basically a particular single “deployment” with all of its configuration and data. Configuration settings, user accounts, workflow definitions and other global objects in the deployment are components of the farm. “Sites” are children of the farm. Sites contain the content of the farm.

Each farm contains a default virtual site. This site is called the “Configuration Site”. This site contains content that is available to all sites in the farm. If a request for some content in a site is made but it exists only in the configuration site, then the request is satisfied from the configuration site.

Content in sites are organized like a file system. This file system contains directories and files in a hierarchical fashion. Corkscrew does not set any limits on the “depth” of the file system or the number of files allowed in a directory. However, each file in the Corkscrew file system is limited to a maximum size of 2 GB. This limit is because the backing SQL Server or MySQL database engines impose the same limit on the size of an instance of binary data stored in it. The total content stored in a single file system is limited by the size of the disk storage the database is hosted on and the policies set in the database engine governing the size-limits and growth of the database over time.

# Workflows

Administrators and other users may use workflows in a content management system to provide business process and policy automation. Such processes may be approvals for new and changed files, restoration of unauthorized changes, automated duplication for safe keeping of critical content, notifications to pre-configured users for certain activity, etc. Typically, workflows in .NET are written using the Windows Workflow Foundation. This framework has had two revisions: v3 and v4. The v3 is now deprecated in the .NET Framework and new code may not compile without special configuration (disabling the preventing error code). Corkscrew supports workflows written in both v3 and v4 frameworks.

In addition, Corkscrew provides its own event-driven workflow infrastructure. Clients implementing this system are supported natively in the Corkscrew system.

NOTE: Corkscrew applications that depend only on the SDK can be implemented on MONO.NET under Linux OS as well. However, the Corkscrew Workflow Service has dependencies on the WF3 and WF4 .NET libraries and cannot be compiled or deployed on a Linux system at this time.

All three types of workflows are executed through the [Corkscrew Workflow Service](#_Corkscrew_Workflow_Service). If this service is not installed, misconfigured or stopped, then workflows will not execute. A single instance of the service can execute workflows from one farm. Since only one instance of a service may be installed on a system (Windows OS limitation), if you manage multiple farms, you will need to install the service on multiple systems to execute workflows on all of them.

NOTE: The 1:1 limitation of the service:farm is by design. A single farm may contain hundreds of active workflows that may take a long time each to do their job. If the service supported multiple farms then it is possible that some workflows are never executed.

Workflows are defined to describe what it can do and what events it can respond to. Workflow definitions are linked to Workflow Manifests. Each workflow definition can have one manifest attached and one manifest can be referenced by one workflow definition. The manifest is a list of all the code, resources, and dependencies of the workflow executable. At the time of execution of the workflow, the Workflow Service checks if the currently persisted executable is the latest. If there is newer code or dependencies available, the code is compiled and then executed.

Workflows are associated to different container objects in the [hierarchy](#_Hierarchical_object_model). Workflows can be associated only to container objects. A container object is one that can contain a child object in the hierarchy. In Corkscrew, the Farm, Site and Directory are containers. The File object is not a container. The association also subscribes to particular events of the container. For example, an association can subscribe to the event of a new site being created.

Create, Update and Delete operations in the object hierarchy cause the events to fire. This in turn creates an instance of the workflow using the association information. The instances are created centrally at the database layer.

The Workflow Service polls the database every 15 minutes for runnable workflows and executes them. Once created, each workflow goes through fixed states in its life cycle.

A workflow goes from Not Started > Started > Completed. After a workflow is in Started state, it may be paused and continued any number of times before it goes to Completed state. At any point, the workflow can also hit a problem and go straight to the Errored state. Once the workflow hits either Completed or Errored states, it cannot be restarted or resumed.

By design, an event will instantiate a workflow in Corkscrew only once. That is, if there is a workflow association that subscribes to a “site created” event, then when a site is created, it results only in one instance of the associated workflow. This means, if the associated workflow starts and then errors out, the workflow actions that the association was meant to perform (like perform some pre-determined automated configuration of the newly created site) will not be performed.

## Recovering from workflow failures

The official guidance is for the publishers of the workflows to handle errors that the workflow may encounter and ensure that the Corkscrew Farm is in a “sane” state. If there were changes in progress when the error occurred, then the workflow must take care of rolling them back. Corkscrew will not automatically rollback the changes made by a workflow.

# Security

Corkscrew has its own user account, permission and security system. In contrast to other content management systems however, Corkscrew does not have a “user group” concept. That is, users are managed and permissions attached individually. This is because user permission requirements can be classified into these broad categories:

|  |  |
| --- | --- |
| Category | Corkscrew Solution |
| Allow/deny access to a set of users, without access/change auditing | Share a user account among the users |
| Allow/deny access to a set of users, with access/change auditing | Each user whose access needs to be audited is given a unique user account. Other users may share one or more accounts |

## Access auditing

Corkscrew automatically audits access to Sites, Directories and Files. However, only the last accessed user Id and date/time stamp are recorded.

## Change Auditing

Corkscrew automatically change-tracks Sites, Directories and Files. Any changes to metadata and content is duly recorded. This includes additions, modifications and deletions. More specifically, the following are audited:

\* When a directory is deleted, all sub directories and files are deleted too. Metadata for all of these child elements are preserved as well.

## Permissions

Corkscrew allows administrators to set three hierarchical access control flags on any hierarchical object for a user account. These are:

* Read – Allows the holder to read the object’s metadata and content.
* Contribute – (Read +) allows the holder to modify the object’s metadata and content.
* Full Control – perform any action on the object.

When setting permissions, it is sufficient to set the highest required privilege. That is, to allow a user to modify an object, it is sufficient to grant “Contribute”. Granting both “Read” and “Contribute” is redundant.

Permissions granted at a level automatically apply down the hierarchy. Therefore, if a user has Contribute access on a particular directory, the user is automatically granted Contribute access to all subfolders and files in that directory.

## Selective Permissions

Unlike many CMS systems out there, Corkscrew does not practice any enforcement for setting permissions on an object. Look at the below table to understand what permissions are granted and how when objects are created or marshalled in the Corkscrew system.

|  |  |  |
| --- | --- | --- |
|  | Permissions granted | How |
| Object Created | Farm Admin – Full Control  Site Admin – Full Control  Users who have Read access to parent object inherit Read access to new object  User creating the object – Full Control to object and any child objects created later | Farm/Site admin – by virtue of their administrator privilege  Implicit inheritance  Because user “created” the object. |
| Object Copied | New object will behave the same as if it was just created. See Object Created above | |
| Object Moved | All permissions granted are retained as is | ACL key is object Id. This does not change when object is moved – only parent changes. Thus, all previous ACLs for the entire tree of objects is maintained as is. |
| Object Deleted | All permissions are dropped automatically | |

What this means is that, after a new object is created, copied or moved, the desired permissions must be adjusted accordingly.

## Deep Permissions

It is possible to grant a user access to a deep resource without granting access on its parent hierarchy. For example, a user may have no access to a Site or a tree of directories and yet have Read, Contribute or even Full Control access on a file deep in the hierarchy.

However, when the system tries to resolve access for the entire hierarchy to access the deep resource, it will encounter “Access denied” situations. To overcome this, when the system detects this to be the case, it will automatically add a non-permission “Child Access” to the exact parent hierarchy. The only privilege “Child Access” grants is to access and resolve a deep link accessor. It does not provide even Read privileges even to metadata of any element in the hierarchy.

Child access permission cannot be set through the API. This is an automatically managed attribute.

## Permission checking

The hierarchical permissions implementation in Corkscrew is a little complicated to look up and adhere to. Therefore, the implementation of permission checking is performed in the application layer and not in the database layer. This means that developers who wish to create their own backend systems for Corkscrew can do so without worrying about adding the permission-layer to their database side code, although they will need to duplicate the data structures used to store the permissions.

Permissions are checked only here!

Front-end or User Application

Corkscrew SDK

ConfigDB

SiteDB

SiteDB

Corkscrew Workflow Service

## Handling permissions when user account is deleted

When a user account is deleted from Corkscrew, any permissions granted to that account are deleted as well. However, if that user held the last permission on the object, the permissions are NOT granted to any other user. This is because the Farm Administrator and each site’s administrators automatically have full control access on all the objects lower in the hierarchy. They would then be able to grant access to a user of their choice on these objects.

# Recommended Topologies

Corkscrew can be deployed in a variety of topologies. Here we detail a few typical scenarios that we recommend. Note that in all of the below cases, the terms “machines”, “boxes” etc can be either a physical system, a virtual machine or a cloud-based entity.

NOTE: The Front-end, User application and Corkscrew Workflow service depend on the Corkscrew SDK. The SDK can be installed and used only on a Windows OS machine or role. The databases can be hosted on any supported database platform.

## Single Server, Everything on one box

This is a “dev box” type scenario. This topology can be used typically in the following scenarios:

1. Developer desktop to create, test, change the application.
2. Trial – to deploy the application and try it out before a purchase.
3. Very small and small centers where dedication multiple server-class boxes to a light-weight application like Corkscrew is not cost-effective.

Front-end or User Application

Corkscrew SDK

ConfigDB

SiteDB

SiteDB

Corkscrew Workflow Service

## Two servers, separation of application and storage roles

In this scenario, we use two boxes. One will host the application layer along with the Corkscrew Workflow Service and the other will host the databases. This is the simplest typical scenario for usage in a mid-sized (SMB) corporate environment.

Front-end or User Application

Corkscrew SDK

ConfigDB

SiteDB

SiteDB

Corkscrew Workflow Service

## Three or more servers, with database redundancy

Mid-sized deployments may also prefer to plan in some database redundancy so that there is recovery in case of failures. This scenario can use three or more hosts with multiple hosts performing database redundancy.

Front-end   
or   
Application

Corkscrew SDK

ConfigDB

SiteDB

SiteDB

ConfigDB

SiteDB

SiteDB

Corkscrew SDK

Corkscrew Workflow Service

REPLICATION

In this scenario, the Corkscrew Workflow Service and its dependency (the SDK) are installed on a separate system.

The ConfigDB and SiteDB databases can be deployed and replicated on different servers. However, we recommend all the databases for a farm to reside under a single database instance.

For large farms, where the size of a single site is so large that its database files can run out of space, there are a few options:

1. Modify the SiteDB definition to use multiple files to host the database content. This is possible through the use of File Groups in Microsoft SQL Server.
2. Host the SiteDB’s files on a SAN storage with sufficient space.
3. Split the farm in such a way that the load of the database file sizes is distributed as per your needs.

If an existing farm grows too big to continue performing well in the current topology, Corkscrew provides a tool to split the farm.

## Cloud-based topology

A cloud has a role-based topology. Instead of machines, it has roles. Of course, to support deployments that require a full machine for some reason, most cloud providers also provide virtual machines hosted on the cloud. But this is not an efficient use of the cloud-principle since VMs cannot utilize the cloud’s features for scalability, reliability and geo-redundancy as easily as roles can. Corkscrew supports deployment in roles in the cloud.

APP + SDK

APP + SDK

SiteDB

SiteDB

ConfigDB

SiteDB

SiteDB

ConfigDB

SiteDB

SiteDB

ConfigDB

SiteDB

SiteDB

ConfigDB

APP + SDK

WFSVC + SDK

WFSVC + SDK

WFSVC + SDK

The application layer with the SDK should be installed in App or Web roles. The databases are obviously deployed on Database roles. The Corkscrew Workflow Service can be installed in a separate set of App or Web role instances or share the same instances as the main app.

## Cloud and On-Premise Mixed Topology

When opting for this hybrid topology, it is essential to decide what features of the cloud you want for which layer of the application you are hosting at each end. You also need to consider security and data-protection aspects – for example, certain types of data should never be hosted in the cloud or they have country/region restrictions.

# How to use change-track and audit data

As mentioned [above](#_Security), Corkscrew automatically logs access to and changes to data hosted in it. It records both metadata and content changes for Site and file system objects.

This audit data is not accessible through the object-model. It can be accessed by directly querying the databases. Site changes are recorded in the SitesChangeLog table in the ConfigDB. Changes to file system objects are recorded in the FileSystemChangeLog table in the relevant database (ConfigDB or SiteDB) that is hosting the content.

This data can be used to rollback changes done by the system. The Corkscrew SDK offers no direct way to roll back changes, because the application may depend on the data state in ways that the SDK cannot determine or control. Therefore, it is left to the implementing application to perform roll backs.

The recommended usages for this data are:

1. Rolling back corrupted, incomplete or broken changes
2. Forensics – to identify who changed what data when
3. Backup – the ChangeLog tables can be separately pulled to a backup database for safe keeping

Any other usage of this data is not supported.