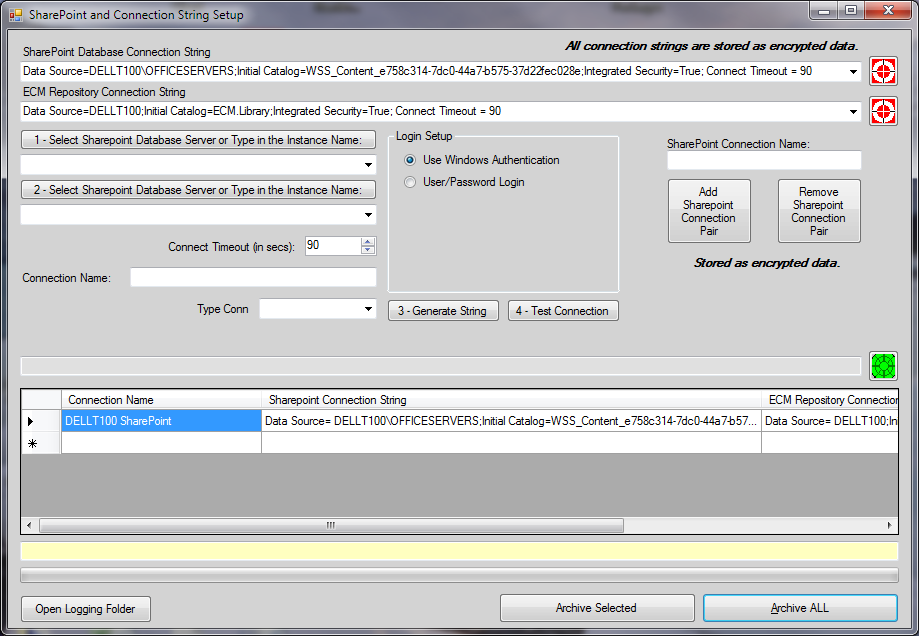
# ECM SharePoint Archiver (ESA)

At first glance, the ECM SharePoint Archiving Utility can appear to be a bit daunting. But when consideration is given to the fact that there is only one screen to master, it becomes far less formidable.

When looking at the screen, it is basically in three parts. The top shows the drop downs that display the “pair matches” between an ECM Repository and a SharePoint Server. Pairs can be created as easily as selecting a SharePoint server and a corresponding ECM Library repository.



The middle part of the screen allows the administrator to set up connections to SharePoint or an ECM Repository. It generates .Net connection strings. The buttons are numbered so it is much akin to a “paint by the numbers” painting. Press 1 and select the SharePoint server or the ECM Repository server. Select the desired server from the dropdown. Press 2 and select the ECM Repository database or the SharePoint database (it can be difficult to find the SharePoint database, please see below for help). Set the desired query timeout. Ninety seconds is the default, but can be anything you wish. Enter the “unique” connection name. Use one that means something as the SharePoint database names will not help you identify which one is currently in use. Select the type of connection. Press 3 and generate the connection string. Press 4 and test if it has been configured correctly or not.

If it is successful, the green button to the right of the string saves it to the ECM repository. This connection is now ready for use. Build a reciprocal connection to either ECM or SharePoint and now, the two can be paired together. Use top dropdown and select the desired SharePoint connection string. Use the next dropdown and select the desired ECM connection string. This will become the SharePoint / ECM connection pair. Name the pair so that it has meaning to you and the name must be unique to the ECM Repository. Press the “Add SharePoint Connection Pair” and the connection between ECM and SharePoint will be completed. Now, the defined SharePoint server will be automatically archived upon the press of a button. There are two buttons on the bottom of the screen. The “Archive Selected” does just that – it immediately archives only the selected pair. “Archive All”, when pressed, goes through the entire list of ECM/SharePoint pairs, archiving each in order of retrieval. In theory there is a limit to the number of pairs that can be constructed. It is somewhere around 2 billion. In common practices and for all practical purposes, it is unlimited. Of course, that number goes into the thousands of trillions if a 64 bit machine is utilized. Again, you will be hard pressed to run out of pair definitions.

SharePoint has turned into its own living beast. It is fast, more limited (much more limited) than ECM Library in its ability to capture content. For example, it does not deal with unknown file types nor does it process zipped files. Or maybe a better to state the obvious, “Not yet, but it will.” It is faster than ECM Library because of the direct pull from DB to DB. I see a way to write an SSIS routine and embed that into ECM Library. Even as fast as it is now, it will be 50 to 75 times faster. This is not needed now, but in 4 to 6 months, I see us moving terabytes of data every hour. So I am forced to think in terms of pure speed and efficiency.

## In disassembling the insides of the SharePoint workings I found:

The database name differs based on how the portals have been named there. A few crucial databases and their SharePoint nomenclature are like this:

* Database ending with \_Config\_db: it stores the entire configuration database for the SharePoint server/farm. Anything pertaining to global configuration and which is set through the SharePoint central administration is stored in this database.
* Database ending with \_SITE: it stores all the content of a specific portal. It contains the files, web pages, webs, sites, and all inter-related SharePoint infrastructure.
* Database ending with \_PROF: it stores entire information about User Profiles; it also stores details of what data to gather about various User Profiles.
* Database ending with \_SERV: It stores data regarding search, notification, and indexing. It stores the gatherer log information as well as the text indexes for the content that is crawled. In fact, it can be easily checked up by right-clicking on any of the database and choosing Properties. In fact, that could be a way to know how much space each function/activity of your portal is taking up.

From a developer's point of view, the \_SERV and \_SITE databases are quite important. The \_SERV database has several very important tables - srch\_GathererLog\_\* (this could be many tables), srch\_docsspecialprops, and sub\_PropVersions. Each of these tables contains data about searching. The srch\_GathererLog\_\* tables contain all the log information from gatherer logs. SharePoint does not allow clearing the gatherer log from UI and this can be done only through the Enterprise Manager.

The srch\_docsspecialprops and sub\_PropVersions contain data about files that have been indexed. The srch\_docsspecialprops table includes information such as title, author, description, etc. for all files crawled as an external file source.

The tables in \_SITE database are really important tables. There are tables like Webs, Docs, DocVersions, and Lists. These tables have content defined by their names. The Webs table contains information about the webs on the portal and the Docs table contains the path, names and actual content of all the documents in your portal infrastructure.

One can look into the **Docs and DocVersions** tables to find out the number of documents and document versions in your portal quickly. In fact, the number of rows in the Docs table tells the number of documents in that particular table. The same is true for the DocVersions table. The number of files shown by the SharePoint object model will be different because it also counts all the aspx files and webparts (.dwp) files to tell you the number, and not just the user stored files.

The question, “Where does SharePoint hide its data?” With a lot of work and little perseverance, one can find:

SharePoint is confusing in that no one seems to know exactly where the data is held and how to get to it. I researched just that query on my ***MOSS 2007*** enterprise search this week - this is what I found.

#### For each shared service provider (SSP), SharePoint has three main locations for data storage:

The full text index catalogue is a flat file. Holy crap – so is ours! I modeled ours after Sybase, go imagine. It is created by index server(s), it consists of a list of keywords and document identifiers, along with mappings as to which keywords exist in which documents. **(Holy crap – so is ours! I modeled ours after Sybase)** This high level of abstraction between keywords and links allows the full text index catalog to typically be around 5-12% of the size of the data being indexed. The document identifiers in the full text index catalogue point to the document URLs and primary keys stored in the search database **(Holy crap – so is ours! I modeled ours after Sybase)**, a SQL Server database that is usually named sharedserviceprovidername\_Search\_DB.MDF. As its name suggests, this literally stores information relating to searchable content on a per-SSP basis, including URLs, access control lists (ACLs) and managed property information. On a WSS system this is named WSS\_Search\_computername.MDF.

The third location is the search configuration (or content) database, which contains configuration information relating to items such as crawl rules, content sources and the definition of managed properties. On a WSS system this is named WSS\_Content.MDF.

As you can see – we are far more compaible to SharePoint than , I am sure, Microsoft will find comfortable. This makes our job of being the ultimate Sharepoint Archiver very solid and very reachable. Hard work, perserverance, and a boot of “under the hood” investigation pays off again. If I could only get that across to the kids I mentor!

#### In addition to the above, there are a number of other elements to the SharePoint solution:

The SharePoint Central Administration content database (itself implemented as a WSS web application), named SharePoint\_AdminContent\_guid.MDF.

SharePoint\_Config\_guid.MDF, a database containing SharePoint configuration information.

The individual web (.aspx) pages and configuration files, along with XSLT transformations and other supporting files (generally XML-based).

Finally, the actual content that is indexed by SharePoint remains where it always was - i.e. in the file shares, document libraries, web sites, business systems, etc. that SharePoint is being used to search across.

(I find that this is true for WSS v3 and MOSS 2007 - other SharePoint versions may differ.)

***Here is a questions for us all to ponder, “What (and where) is SharePoint Server 2009?” Products for 2010 are all over the place – what is up with this. We need to have a meeting soon to explore strategic possibilities for what this means. I think they are waiting to release a group of power products that will give us some serious competition.***

FYI: I found that SharePoint designer speeds up the process of checking in multiple files.

Configuration Database Tables

|  |  |
| --- | --- |
| **Name** | **Description** |
| [AntiVirusVendors](http://msdn.microsoft.com/en-us/library/dd587568(office.11).aspx) | Antivirus software vendors. |
| [CustomTemplates](http://msdn.microsoft.com/en-us/library/dd587580(office.11).aspx) | Templates that appear in the global templates list (template picker). |
| [Databases](http://msdn.microsoft.com/en-us/library/dd587585(office.11).aspx) | Content databases and associated settings. |
| [Globals](http://msdn.microsoft.com/en-us/library/dd585134(office.11).aspx) | Settings that apply to all sites on the physical server within a Windows SharePoint Services topology. |
| [InstalledWebPartPackages](http://msdn.microsoft.com/en-us/library/dd585163(office.11).aspx) | Web Part Packages that are available to the virtual server. |
| [Servers](http://msdn.microsoft.com/en-us/library/dd585192(office.11).aspx) | Physical computers in the Windows SharePoint Services topology. |
| [Services](http://msdn.microsoft.com/en-us/library/dd585200(office.11).aspx) | Services such as database, Web, and mail services that are included in the Windows SharePoint Services topology. |
| [Sites](http://msdn.microsoft.com/en-us/library/dd585220(office.11).aspx) | Sites in the Windows SharePoint Services topology. |
| [VirtualServers](http://msdn.microsoft.com/en-us/library/dd585251(office.11).aspx) | Virtual servers and their associated settings. |
| [WebPartPackages](http://msdn.microsoft.com/en-us/library/dd585278(office.11).aspx) | Web Part Packages that are installed in a topology. |

Content Database Tables

|  |  |
| --- | --- |
| **Name** | **Description** |
| [Categories](http://msdn.microsoft.com/en-us/library/dd587571(office.11).aspx) | Associations among documents and their assigned areas. |
| [ComMd](http://msdn.microsoft.com/en-us/library/dd587576(office.11).aspx) | Document discussions. |
| [Deps](http://msdn.microsoft.com/en-us/library/dd587590(office.11).aspx) | Document dependencies. |
| [DiskWarningDate](http://msdn.microsoft.com/en-us/library/dd587594(office.11).aspx) | Date and time of last warning about disk quota. |
| [Docs](http://msdn.microsoft.com/en-us/library/dd587598(office.11).aspx) | Documents and document folders. |
| [DocVersions](http://msdn.microsoft.com/en-us/library/dd585118(office.11).aspx) | Version history of documents from document libraries. |
| [EventCache](http://msdn.microsoft.com/en-us/library/dd585124(office.11).aspx) | Cache for alerts. |
| [EventLog](http://msdn.microsoft.com/en-us/library/dd585129(office.11).aspx) | Log for alerts. |
| [HT\_Cache](http://msdn.microsoft.com/en-us/library/dd585141(office.11).aspx) | Cache of documents transformed for viewing in a Web browser. |
| [HT\_Settings](http://msdn.microsoft.com/en-us/library/dd585147(office.11).aspx) | Settings for HTML viewing. |
| [ImmedSubscriptions](http://msdn.microsoft.com/en-us/library/dd585155(office.11).aspx) | Alerts that are set to be sent as soon as the changes occur. |
| [Links](http://msdn.microsoft.com/en-us/library/dd585169(office.11).aspx) | Links to be recalculated during a site recalculation. |
| [Lists](http://msdn.microsoft.com/en-us/library/dd585174(office.11).aspx) | Lists in the site. |
| [NavNodes](http://msdn.microsoft.com/en-us/library/dd585180(office.11).aspx) | Nodes that are displayed in the navigation bars. |
| [Personalization](http://msdn.microsoft.com/en-us/library/dd585188(office.11).aspx) | Personalization done to Web Parts. |
| [SchedSubscriptions](http://msdn.microsoft.com/en-us/library/dd585191(office.11).aspx) | Alerts that are set to be sent on a daily or weekly basis. |
| [SiteGroupMembership](http://msdn.microsoft.com/en-us/library/dd585209(office.11).aspx) | Members of cross-site groups. |
| [SiteGroups](http://msdn.microsoft.com/en-us/library/dd585214(office.11).aspx) | Cross-site groups. |
| [Sites](http://msdn.microsoft.com/en-us/library/dd585227(office.11).aspx) | Sites in the content database. |
| [TimerLock](http://msdn.microsoft.com/en-us/library/dd585233(office.11).aspx) | Locking scheme that determines which server locks which database. |
| [UserData](http://msdn.microsoft.com/en-us/library/dd585240(office.11).aspx) | List data. |
| [UserInfo](http://msdn.microsoft.com/en-us/library/dd585246(office.11).aspx) | Info about each user added to the site. |
| [WebCat](http://msdn.microsoft.com/en-us/library/dd585257(office.11).aspx) | Links between Web sites and areas. |
| [WebGroupMembership](http://msdn.microsoft.com/en-us/library/dd585263(office.11).aspx) | List of members of the site groups. |
| [WebGroups](http://msdn.microsoft.com/en-us/library/dd585267(office.11).aspx) | Site groups (such as Administrator, Web Designer) for the site. |
| [WebMembers](http://msdn.microsoft.com/en-us/library/dd585273(office.11).aspx) | Members of a site. |
| [WebParts](http://msdn.microsoft.com/en-us/library/dd585284(office.11).aspx) | Web Parts available for the sites. A view is a Web Part. |
| [Webs](http://msdn.microsoft.com/en-us/library/dd585290(office.11).aspx) | Subsites associated with a site. |
| [WelcomeNames](http://msdn.microsoft.com/en-us/library/dd585298(office.11).aspx) | Default documents for the site. |
|  |  |

### The DOCS Table (for example – and the links above provide us with this kind of detail)

Docs Table

The Docs table stores metadata about documents.

**Warning**  Modifying the database schema or database structures is not supported. Changes that you make to the database contents may be overwritten when you install updates or service packs for Windows SharePoint Services, or when you upgrade an installation to the next product version.

## Columns

|  |  |  |
| --- | --- | --- |
| **Column name** | **Data type** | **Description** |
| Id | **uniqueidentifier** | GUID of the document. |
| SiteId | **uniqueidentifier** | GUID of the site to which the document belongs. |
| DirName | **nvarchar** | Directory in which the file is stored. |
| LeafName | **nvarchar** | Name of the file or folder. |
| WebId | **uniqueidentifier** | GUID of the subsite to which the document belongs. Found in the Webs table. |
| ListId | **uniqueidentifier** | GUID of the list with which the document is associated. |
| DoclibRowId | **int** | Row ID of the document. Documents stored in a document library are numbered in the order in which they are uploaded. |
| Type | **tinyint** | 1 for a folder; 0 for a file. |
| Size | **int** | Size of the document. |
| MetaInfoSize | **int** | Size of the metadata. |
| Version | **int** | Current version number of the document. Initial version is 1. |
| UIVersion | **int** | Document version number that is displayed to users. |
| Dirty | **bit** | Value that indicates whether a document needs to be reparsed. When **true**, it is used to clean up links inside the document after a document rename. |
| CacheParseId | **uniqueidentifier** | GUID value used for optimistic concurrency checking. |
| DocFlags | **tinyint** | Bit field used for various settings. |
| ThicketFlag | **bit** | Value that indicates whether the document has additional supporting files. |
| CharSet | **int** | Assumed Microsoft Windows character set of the document. |
| TimeCreated | **datetime** | Date and time the file was created. |
| TimeLastModified | **datetime** | Date and time the file was modified. |
| NextToLastTimeModified | **datetime** | Date and time the file was modified previous to TimeLastModified. |
| MetaInfoTimeLastModified | **datetime** | Date and time the metadata was updated. |
| TimeLastWritten | **datetime** | Date and time the document was written. |
| SetupPath | **nvarchar** | Path from which the file was originally installed. |
| CheckoutUserId | **int** | ID of the user in the UserInfo table who has the file checked out. |
| CheckoutDate | **datetime** | Date and time that the file was checked out. |
| CheckoutExpires | **datetime** | Date and time that the checkout expires. Does not get cleared when the checkout expires. |
| CheckoutSize | **int** | Size of the file when it was checked out. |
| VersionCreatedSinceSTCheckout | **bit** | Flag that prevents multiple document versions from being created on a single short-term checkout. |
| LTCheckoutUserId | **int** | ID of the user in the UserInfo table who has the file checked out. |
| VirusVendorID | **int** | ID of the antivirus vendor. |
| VirusStatus | **int** | Status of the virus scan. |
| VirusInfo | **nvarchar** | Information about a virus, if detected. |
| MetaInfo | **image** | Metadata in a binary field. |
| Content | **image** | Content of the file in a binary field. |
| CheckoutContent | **image** | Field that overwrites the Content field when the file is checked in. |